IEC 60839-11-33

Edition 1.0 2021-08

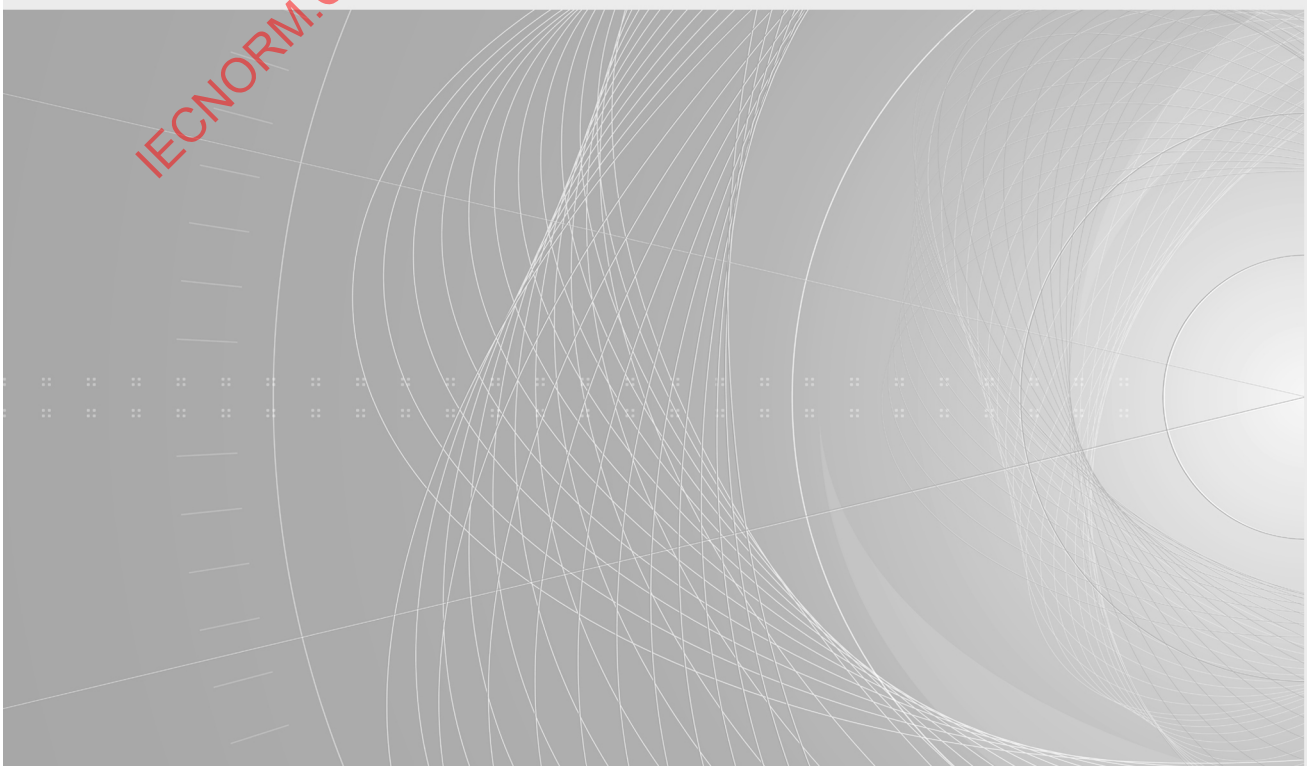
INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Alarm and electronic security systems – Part 11-33: Electronic access control systems – Access control configuration based on Web services

Systèmes d'alarme et de sécurité électroniques – Partie 11-33: Systèmes de contrôle d'accès électronique – Configuration du contrôle d'accès en fonction des services Web



IEC 60839-11-33:2021-08(en-fr)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

IEC 60839-11-33

Edition 1.0 2021-08

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Alarm and electronic security systems – Part 11-33: Electronic access control systems – Access control configuration based on Web services

Systèmes d'alarme et de sécurité électroniques – Partie 11-33: Systèmes de contrôle d'accès électronique – Configuration du contrôle d'accès en fonction des services Web

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 13.320

ISBN 978-2-8322-1011-4

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

ALARM AND ELECTRONIC SECURITY SYSTEMS –

Part 11-33: Electronic access control systems – Access control configuration based on Web services

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 60839-11-33 has been prepared by IEC technical committee 79: Alarm and electronic security systems. It is an International Standard.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
79/646/FDIS	79/648/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

A list of all parts in the IEC 60839 series, published under the general title *Alarm and electronic security systems*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This document makes it possible to build an alarm and electronic security system with clients, typically a monitoring console, and devices, typically an access control unit, from different manufacturers using common and well defined interfaces.

The document specifies only the data and control flow between a client and the services without reference to any physical device as the services required to implement a compliant electronic access control system (EACS) are not necessarily implemented on a single device, i.e. all services can be run on a control panel, event aggregator software on PC, etc.

This document does not define internal communication between an access control unit and its components if they are implemented on a single device.

This document is based upon work done by the ONVIF open industry forum. The ONVIF Credential specification, ONVIF Access Rules specification, ONVIF Authentication Behaviour specification and ONVIF Schedule specification are compatible with this document.

This document is accompanied by a set of computer readable interface definitions (see Annex A):

- credential service WSDL, see Clause A.1;
- access rules service WSDL, see Clause A.2;
- authentication behaviour service WSDL, see Clause A.3;
- schedule service WSDL, see Clause A.4.

Due to the differences in terminology used in IEC 60839-11-1:2013 and IEC 60839-11-2:2014 and the ONVIF specification that this part of IEC 60839 is based on, a reader should take special notice of the terms and definitions clause.

Additional services needed for monitoring of doors and access points (portal sides) are outside the scope of this document. These services are covered by IEC 60839-11-32.

ALARM AND ELECTRONIC SECURITY SYSTEMS –

Part 11-33: Electronic access control systems – Access control configuration based on Web services

1 Scope

This part of IEC 60839 defines the Web services interface for electronic access control systems. This includes listing electronic access control system components, their logical composition, monitoring their states and controlling them. It also includes a mapping of mandatory and optional requirements in accordance with IEC 60839-11-1:2013, as covered by Annex B.

This document applies to physical security only. Physical security prevents unauthorized personnel, attackers or accidental intruders from physically accessing a building, room, etc.

Web services usage and device management functionality are outside the scope of this document. Refer to IEC 60839-11-31:2016 for more information.

This document does not in any way limit a manufacturer to add other protocols or extend the protocol defined here. For rules on how to accomplish this, refer to IEC 60839-11-31:2016.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60839-11-1:2013, *Alarm and electronic security systems – Part 11-1: Electronic access control systems – System and components requirements*

IEC 60839-11-2:2014, *Alarm and electronic security systems – Part 11-2: Electronic access control systems – Application guidelines*

IEC 60839-11-31:2016, *Alarm and electronic security systems – Part 11-31: Electronic access control systems – Core interoperability protocol based on Web services*

IEC 60839-11-32:2016, *Alarm and electronic security systems – Part 11-32: Electronic access control systems – Access control monitoring based on Web services*

ISO 16484-5:2017, *Building automation and control systems (BACS) – Part 5: Data communication protocol*

RFC 5545, *Internet Calendaring and Scheduling Core Object Specification (iCalendar),* (availabe at https://tools.ietf.org/html/rfc5545)

RFC 5234, *Augmented BNF for Syntax Specifications: ABNF,* (available at https://tools.ietf.org/html/rfc5234)

3 Terms and definitions

For the purposes of this document the terms and definitions given in IEC 60839-11-1, IEC 60839-11-2, IEC 60839-11-31 and IEC 60839-11-32 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

NOTE When the IEC term defined in IEC 60839-11-1 and IEC 60839-11-2 differs from the terms used in this document, the IEC term will be given within parentheses in the section headers.

3.1
access level
set of rules used to determine where and when a credential has authorized access to one or more portals and which may include special passage conditions such as specific portal allowed open times

Note 1 to entry: In this document, the term "access rules" is used.

3.2
access point
portal
physical entrance/exit at which access can be controlled by a door, turnstile or other secure barrier

Note 1 to entry: For the purposes of this document the access point is considered to be a logical composition of a physical door and reader(s) controlling access in one direction.

Note 2 to entry: In this document, the term "door" has the same meaning as "access point" or "portal".

3.3
access policy
association of an access point (portal side) and a schedule (time zone)

Note 1 to entry: An access policy defines when an access point (portal side) can be accessed.

3.4
access profile
collection of access policies

Note 1 to entry: Access profiles are used to define "when" and "where" access will be granted.

3.5
anti-passback violation state
state indicating if the anti-passback rules have been violated for a credential

3.6
authentication policy
association of a security level and a schedule (time zone)

Note 1 to entry: An authentication policy defines when a security level is required at an access point (portal side).

3.7
authentication profile
collection of authentication policies

EXAMPLE All entrance access points (portal sides) are configured to require card access during office hours, card and PIN access during nighttime, and no access during holidays.

Note 1 to entry: Authentication profiles are used to define authentication behaviour for a type of access point (portal sides).

3.8
credential
information either memorized or held within a token with information either memorized, held within a token or uniquely identified biological (person) or physical (vehicle) trait of the user or credential holder

EXAMPLE A credential number can be stored in the physical credential (token). A PIN code is associated with the token, but will not necessarily be stored in the token. Both the credential number and the PIN code are credential identifiers that can be contained in the same credential.

Note 1 to entry: In this document, the term "credential" represents information that is used to identify a credential holder (user) at an access point (portal side). This information is not necessarily held within a token.

Note 2 to entry: In this document, several pieces of information (credential identifiers) can be part of the same credential if they belong together.

3.9
credential identifier
piece of data that identifies a credential holder (user) at an access point

EXAMPLE Card number, unique card information, PIN, fingerprint, or other biometric information.

Note 1 to entry: A credential identifier also specifies the type of identifier (recognition method) and the format of the value used in the Web service interface.

3.10
credential state
state indicating if a credential is enabled or disabled, or if anti-passback rules have been violated or not

Note 1 to entry: The state may also contain a reason why the credential was disabled.

3.11
format type
format of a credential identifier value in the Web services interface

Note 1 to entry: The format is not necessarily the same as when it is stored in a physical credential.

Note 2 to entry: ONVIF supports the BACnet format types listed in ISO 16484-5:2017.

3.12
iCalendar
industry standard format for exchanging scheduling and activity-recording information electronically

3.13
recognition group
logical OR between the recognition methods in a security level

EXAMPLE One recognition group contains the recognition methods pt:Card and pt:Fingerprint. Another group contains the recognition methods pt:Card and pt:Face. The resulting effect is that the access point will require either Card+Fingerprint, or Card+Face.

3.14
recognition method
method of recognizing a credential holder (user) at an access point (portal side)

EXAMPLE Information held within a physical credential (token) such as a credential number, memorized information such as a PIN, fingerprint, or other biometric information.

3.15
recognition type
recognition method or anonymous entry

3.16
security level
defined level of assurance to be granted access to an access point (portal side)

Note 1 to entry: A security level may be defined as individual recognition methods, combinations of recognition methods (using logical AND or OR), or no recognition methods (open).

3.17
special day group
set of days (or parts of days) that require the regular schedule (time zone) to be overridden

EXAMPLE Holidays, half-days or working Sundays.

3.18
special days' schedule
set of overriding time periods (time slots) for a special day group

3.19
time slot
interval of time between two given moments indicating the beginning and the end of a valid period within a time zone

Note 1 to entry: In this document, the term "time period" is used.

Note 2 to entry: In this document, time periods are only used in special days' schedules, i.e. the time slots that override a regular schedule during special days.

Note 3 to entry: Time periods do not have a date part.

3.20
time zone
one or more time slots combined with calendar information

EXAMPLE The schedule (time zone) "Working hours" can be defined as two time periods (time slots) valid on working days, one from 08:00 to 11:30, and one from 12:30 to 18:00. A set of special days' schedules can be defined for dates the regular time periods do not apply. Each special day's schedule can contain another set of time periods that will override the regular time periods for the special days.

Note 1 to entry: In this document, the term "schedule" is used.

3.21
token
portable device containing a readable unique identifier (credential) that can be associated with a user's data and access rights stored within the electronic access control system

Note 1 to entry: In this document, the term "physical credential" is used.

Note 2 to entry: In this document, the term "token" is the unique identifier of any managed entity.

3.22
user
person requesting access through an access point

Note 1 to entry: In this document, the term "credential holder" is used.

Note 2 to entry: A credential holder is not necessarily a person; a credential may for instance be assigned to a vehicle; the vehicle credential shall be associated with a person.

3.23
vEvent
component in iCalendar, specifying the properties of an event

4 Overview

4.1 General

This document defines four Web service interfaces:

- credential service (see Clause 5);
- access rules service (see Clause 6);
- authentication behaviour service (see Clause 7);
- schedule service (see Clause 8).

The services above relate to the two following Web service interfaces defined in IEC 60839-11-32:2016:

- access control service (see IEC 60839-11-32:2016, Clause 5);
- door control service (see IEC 60839-11-32:2016, Clause 6).

Figure 1 shows how the six Web service interfaces relate to each other.

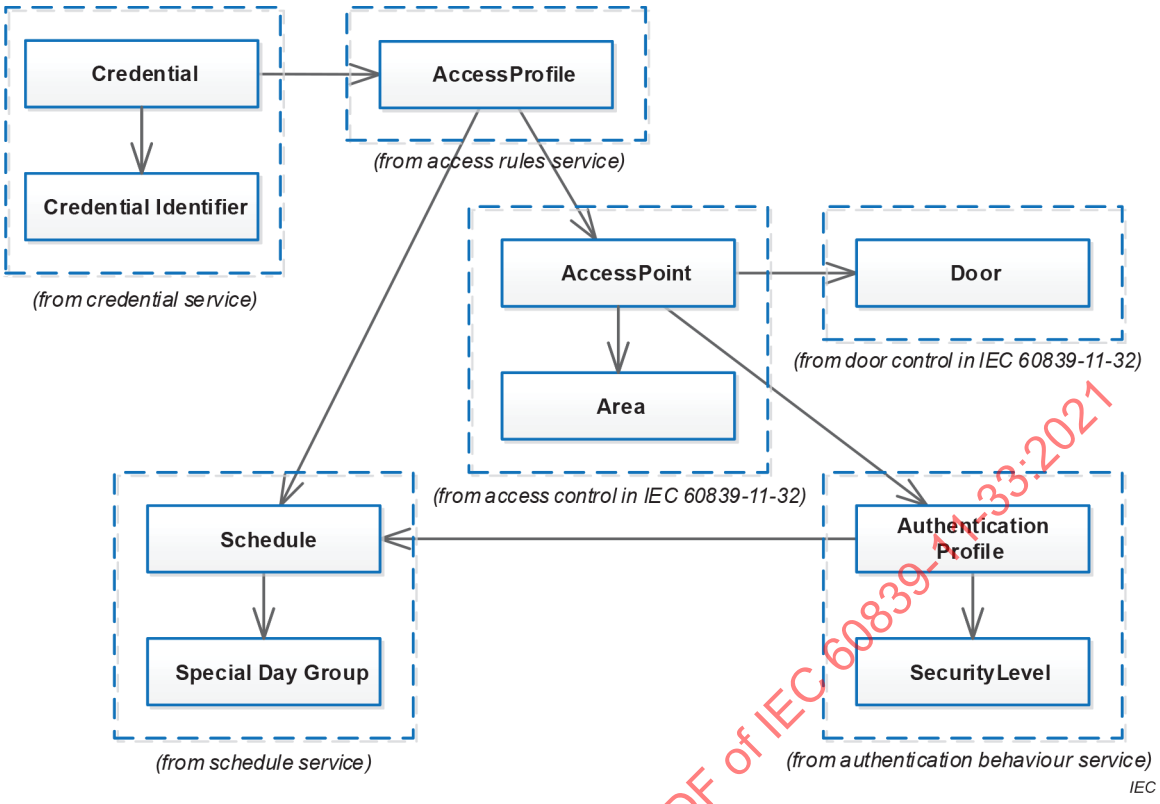


Figure 1 – Overview of service dependencies

Also refer to IEC 60839-11-32:2016, Clause 4, which is relevant for this document.

4.2 Namespaces

In addition to namespaces listed in IEC 60839-11-31:2016, 5.4, this document defines the namespaces listed in Table 1.

Table 1 – Defined namespaces in this document

Prefix	Namespace URI	Description
pt	https://www.onvif.org/ver10/pacs/types.xsd	XML schema descriptions in this document
tac	http://www.onvif.org/ver10/accesscontrol/wsdl	Namespace for the access control service
tdc	http://www.onvif.org/ver10/doorcontrol/wsdl	Namespace for the door control service
tcr	http://www.onvif.org/ver10/credential/wsdl	Namespace for the credential service
tar	http://www.onvif.org/ver10/accessrules/wsdl	Namespace for the access rules service
tab	http://www.onvif.org/ver10/authenticationbehaviour/wsdl	Namespace for the authentication behaviour service.
tsc	http://www.onvif.org/ver10/schedule/wsdl	Namespace for the schedule service

The namespaces listed in Table 2 are referenced by this document.

Table 2 – Referenced namespaces (with prefix)

Prefix	Namespace URI	Description
env	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [W3C SOAP12-PART11]

4.3 Error handling

Faults listed in this document are specific to each command. Each command may also return generic faults listed in IEC 60839-11-31:2016, 5.12.3.2.

5 Credential service

5.1 General

The credential service specification defines the commands to configure credentials.

A credential holds information that can be validated in an access point, such as unique card information, PIN, biometric information. A credential also holds information on what the credential can access via credential access profiles, which ties the credential to access profiles as described in Clause 6.

A credential is assigned to a person (called credential holder) and has a validity that specifies the period during which the credential can be used to get access.

Consider the following example:

> A credential is assigned to a consultant to be temporarily used during one week (April 2nd to April 6th). The start date/time of the validity is set to the morning of April 2nd and the end date/time of the validity is set to the evening of April 6th.
>
> This particular credential is a card with number 987654321 and the consultant is given the personal PIN code 1234. Both pieces of information are stored in the credential. The card number is a credential identifier of type pt:Card and the PIN code is a credential identifier of type pt:PIN (see 5.2.2 for supported identifier types).
>
> The consultant will help out with the installation of a server so he needs access to the server room, and of course also access to the other common facilities at the office. The access profile "IT support access" gives access to the server room during office hours, and the access profile "Staff access" gives access to the rest of the office. The references to both access profiles are stored on the credential in the CredentialAccessProfile structure.

A credential service object model representation is shown in Figure 2.

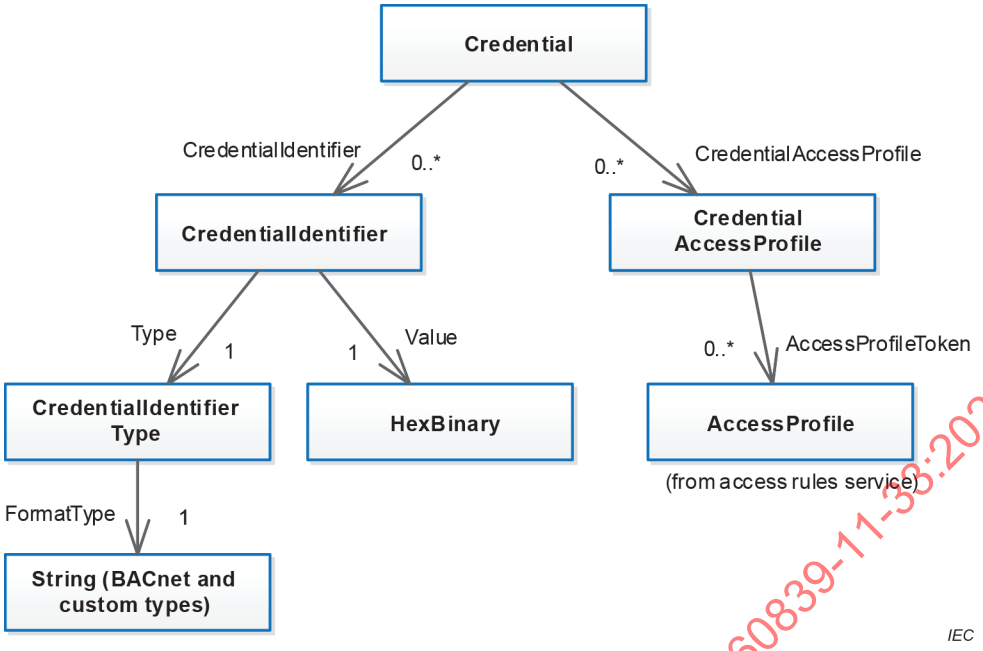


Figure 2 – Main data structures in the credential service

5.2 Service capabilities

5.2.1 General

The device shall provide service capabilities in two ways:

1) With the GetServices method of Device service when IncludeCapability is true. Refer to IEC 60839-11-31:2016, 8.1.
2) With the GetServiceCapabilities method.

5.2.2 ServiceCapabilities data structure

The service capabilities reflect the optional functionalities of a service. The information is static and does not change during the device operation. The following capabilities are available:

- **MaxLimit**
  The maximum number of entries returned by a single Get<Entity>List or Get<Entity> request. The device shall never return more than this number of entities in a single response.
- **CredentialValiditySupported**
  Indicates that the device supports credential validity.
- **CredentialAccessProfileValiditySupported**
  Indicates that the device supports validity on the association between a credential and an access profile.
- **ValiditySupportsTimeValue**
  Indicates that the device supports both date and time value for validity. If set to false, then the time value is ignored.
- **MaxCredentials**
  The maximum number of credentials supported by the device.
- **MaxAccessProfilesPerCredential**
  The maximum number of access profiles for a credential.
- **ResetAntipassbackSupported**
  Indicates the device supports resetting of anti-passback violations and notifying on anti-passback violations.

- **SupportedIdentifierType**
  A list of identifier types that the device supports. Identifiers types starting with the prefix pt: are reserved to define ONVIF-specific types. For custom defined identifier types, free text can be used. The following types are defined by ONVIF:
  - pt:Card Supports Card identifier type (including fob, tag and similar).
  - pt:PIN Supports PIN identifier type.
  - pt:Fingerprint Supports Fingerprint biometric identifier type.
  - pt:Face Supports Face biometric identifier type.
  - pt:Iris Supports Iris biometric identifier type.
  - pt:Vein Supports Vein biometric identifier type.
  - pt:Palm Supports Palm biometric identifier type.
  - pt:Retina Supports Retina biometric identifier type.
- **SupportedExemptionType**
  A list of exemptions that the device supports. Exemption types starting with the prefix pt: are reserved to define ONVIF-specific types. For custom defined exemption types, free text can be used. The following types are defined by ONVIF:
  - pt:ExemptFromAuthentication Supports ExemptedFromAuthentication in 5.3.2.3.
- **ClientSuppliedTokenSupported**
  Indicates that the client is allowed to supply the token when creating credentials. To enable the use of the command SetCredential, the value shall be set to true.
- **DefaultCredentialSuspensionDuration**
  The default time period that the credential will temporary be suspended (e.g. by using the wrong PIN a predetermined number of times).

The time period is defined as an [ISO 8601] duration string (e.g. "PT5M").

5.2.3 GetServiceCapabilities command

This operation returns the capabilities of the credential service.

A device shall support this command as described in Table 3.

Table 3 – GetServiceCapabilities command

GetServiceCapabilities	Access class: PRE_AUTH
Message name	**Description**
GetServiceCapabilitiesRequest	*This message shall be empty.*
GetServiceCapabilitiesResponse	*This message contains:* • *"Capabilities": The capability response message contains the requested credential service capabilities using a hierarchical XML capability structure.* tcr:ServiceCapabilities **Capabilities [1][1]**
Fault codes	**Description**
	No command specific faults.

5.3 Credential information

5.3.1 General

A credential holds information that can be validated in an access point (card information, PIN, etc.) and what the credential can access (access profiles).

5.3.2 Data structures

5.3.2.1 CredentialInfo

The CredentialInfo type represents the credential as a logical object. The structure contains the basic information of a specific credential instance. The device shall provide the following fields for each credential.

- **token**
  A service unique identifier of the credential.
- **CredentialHolderReference**
  An external reference to a person holding this credential. The reference is a username or used ID in an external system, such as a directory service.

To provide more information, the device may include the following optional fields:

- **Description**
  User readable description for the credential. It shall be up to 1 024 characters.
- **ValidFrom**
  The start date/time validity of the credential. If the ValiditySupportsTimeValue capability is set to false, then only the date is supported (the time is ignored).
- **ValidTo**
  The expiry date/time validity of the credential. If the ValiditySupportsTimeValue capability is set to false, then only the date is supported (the time is ignored).

5.3.2.2 Credential

A credential is a physical/tangible object, a piece of knowledge, or a facet of a person's physical being that enables an individual access to a given physical facility or computer-based information system. A credential holds one or more credential identifiers. To gain access one or more identifiers may be required.

The device shall include all properties of the CredentialInfo structure and also a list of credential identifiers and a list of credential access profiles.

- **CredentialIdentifier**
  A list of credential identifier structures. At least one credential identifier is required. Maximum one credential identifier structure per type is allowed.

To provide more information, the device may include the following optional fields:

- **CredentialAccessProfile**
  A list of credential access profile structures.
- **Attribute**
  A list of credential attributes as name value pairs. Key name types starting with the prefix pt: are reserved to define ONVIF-specific types. For custom defined key name types, free text can be used.
- **ExtendedGrantTime**
  A boolean indicating that the credential holder needs extra time to get through the door. ExtendedReleaseTime will be added to ReleaseTime, and ExtendedOpenTime will be added to OpenTime.

5.3.2.3 CredentialIdentifier

A credential identifier is a card number, unique card information, PIN or biometric information such as fingerprint, iris, vein, face recognition, that can be validated in an access point.

- **Type**
  Contains the details of the credential identifier type. Is of type CredentialIdentifierType.
- **ExemptedFromAuthentication**
  If set to true, this credential identifier is not considered for authentication. For example if the access point requests Card plus PIN, and the credential identifier of type PIN is exempted from authentication, then the access point will not prompt for the PIN.
- **Value**
  The value of the identifier in hexadecimal representation.

5.3.2.4 CredentialIdentifierType

Specifies the name of the credential identifier type and its format for the credential value.

- **Name**
  The name of the credential identifier type, such as pt:Card, pt:PIN, etc. (see 5.2.2 for supported credential identifier types).
- **FormatType**
  Specifies the format of the credential value for the specified identifier type name. See 5.3.2.5 below.

5.3.2.5 CredentialIdentifierFormatTypeInfo

Contains information about a format type.

- **FormatType**
  A format type supported by the device. A list of supported format types is provided in ISO 16484-5:2017. The BACnet type "CUSTOM" is not used in this document. Instead device manufacturers can define their own format types.
- **Description**
  User readable description of the credential identifier format type. It shall be up to 1 024 characters. For custom types, it is recommended to describe how the octet string is encoded (following the structure given in the column Authentication Factor Value Encoding of ISO 16484-5:2017, Table P-1).

5.3.2.6 CredentialAccessProfile

The association between a credential and an access profile. Refer to 6.3.2.2.

- **AccessProfileToken**
  The reference token of the associated access profile.

The device may include the following optional fields:

- **ValidFrom**
  The start date/time of the validity for the association between the credential and the access profile. If the ValiditySupportsTimeValue capability is set to false, then only the date is supported (the time is ignored).
- **ValidTo**
  The end date/time of the validity for the association between the credential and the access profile. If the ValiditySupportsTimeValue capability is set to false, then only the date is supported (the time is ignored).

5.3.2.7 CredentialState

The CredentialState structure contains information about the state of the credential and optionally the reason why the credential was disabled.

- **Enabled**
  True if the credential is enabled or false if the credential is disabled.

The device may include the following optional fields:

- **Reason**
  The reason for disabling the credential. Reason types starting with the prefix pt: are reserved to define ONVIF-specific types. For custom defined reason types, free text can be used. The following types are defined by ONVIF:
  - pt:CredentialLockedOut — Access is denied due to credential locked out.
  - pt:CredentialBlocked — Access is denied because the credential has deliberately been blocked by the operator.
  - pt:CredentialLost — Access is denied due to the credential being reported as lost.
  - pt:CredentialStolen — Access is denied due to the credential being reported as stolen.
  - pt:CredentialDamaged — Access is denied due to the credential being reported as damaged.
  - pt:CredentialDestroyed — Access is denied due to the credential being reported as destroyed.
  - pt:CredentialInactivity — Access is denied due to credential inactivity.
  - pt:CredentialExpired — Access is denied because the credential has expired.
  - pt:CredentialRenewalNeeded — Access is denied because the credential requires a renewal (e.g. new PIN or fingerprint enrollment).
- **AntipassbackState**
  A structure indicating the anti-passback state. This field shall be supported if the ResetAntipassbackSupported capability is set to true.

5.3.2.8 Anti-passback State

A structure containing anti-passback related state information.

- **AntipassbackViolated**
  Indicates if anti-passback is violated for the credential.

5.3.2.9 GetCredentialInfo command

This operation requests a list of CredentialInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 4.

Table 4 – GetCredentialInfo command

GetCredentialInfo	Access class: READ_SYSTEM
Message name	**Description**
GetCredentialInfoRequest	*This message contains:* • *"Token": Tokens of CredentialInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetCredentialInfoResponse	*This message contains:* • *"CredentialInfo": List of CredentialInfo items.* tcr:CredentialInfo **CredentialInfo [0][unbounded]**
Fault codes	***Description***
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

5.3.3 GetCredentialInfoList command

This operation requests a list of all CredentialInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 5.

Table 5 – GetCredentialInfoList command

GetCredentialInfoList	Access class: READ_SYSTEM
Message name	**Description**
GetCredentialInfoListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetCredentialInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"CredentialInfo": List of CredentialInfo items.* xs:string **NextStartReference [0][1]** tcr:CredentialInfo **CredentialInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

5.3.4 GetCredentials command

This operation requests a list of Credential items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 6.

Table 6 – GetCredentials command

GetCredentials	Access class: READ_SYSTEM_SECRET
Message name	**Description**
GetCredentialsRequest	*This message contains:* • *"Token": Tokens of Credentials to get.* pt:ReferenceToken **Token [1][unbounded]**
GetCredentialsResponse	*This message contains:* • *"Credential": List of Credential items.* tcr:Credential **Credential [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

5.3.5 GetCredentialList command

This operation requests a list of all Credential items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 7.

Table 7 – GetCredentialList command

GetCredentialList	Access class: READ_SYSTEM_SECRET
Message name	**Description**
GetCredentialListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetCredentialListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"Credential": List of Credential items.* xs:string **NextStartReference [0][1]** tcr:Credential **Credential [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

5.3.6 CreateCredential command

This operation creates the specified credential in the device.

A call to this method takes a Credential structure and a CredentialState structure as input parameters. The credential state can be created in disabled or enabled state.

The token field of the Credential structure shall be empty and the device shall allocate a token for the credential. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

A device shall support this command as described in Table 8.

Table 8 – CreateCredential command

CreateCredential	Access class: WRITE_SYSTEM
Message name	**Description**
CreateCredentialRequest	*This message contains:* • *"Credential": The credential to create.* • *"State": The state of the credential.* tcr:Credential **Credential [1][1]** tcr:CredentialState **State [1][1]**
CreateCredentialResponse	*This message contains:* • *"Token": The token of the created credential.* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
env:Sender ter:CapabilityViolated ter:MaxCredentials	*There is not enough space to create a new credential, see the MaxCredentials capability.*
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	*There are too many access profiles per credential, see the MaxAccessProfilesPerCredential capability.*
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	*Credential validity is not supported by the device, see the CredentialValiditySupported capability.*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*Credential access profile validity is not supported by the device, see the CredentialAccessProfileValiditySupported capability.*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Specified identifier type is not supported by the device, see the SupportedIdentifierType capability.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	*The same identifier type was used more than once.*
env:Sender ter:InvalidArgVal ter:InvalidFormatType	*Specified identifier format type is not supported by the device.*
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	*Specified identifier value is not in accordance with the FormatType definition.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	*The same combination of identifier type, format and value was used more than once (some devices may not support duplicate identifier values).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	*Specified exemption type is not supported by the device, see the SupportedExemptionType capability.*

5.3.7 SetCredential command

This method is used to synchronize a credential in a client with the device.

A call to this method takes a Credential structure and a CredentialState structure as input parameters. The token field of the credential shall not be empty.

If a credential with the specified token does not exist in the device, the credential is created. The credential state can be created in disabled or enabled state.

If a credential with the specified token exists in the device, then the credential is modified. The credential state will be set according to the specified state (this behaviour differs from the ModifyCredential command). All existing credential identifiers and credential access profiles are removed and replaced with the specified entities.

A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 9.

Table 9 – SetCredential command

SetCredential	Access class: WRITE_SYSTEM
Message name	**Description**
SetCredentialRequest	*This message contains:* • *"Credential": The credential to create.* • *"State": The state of the credential.* tcr:Credential **Credential [1][1]** tcr:CredentialState **State [1][1]**
SetCredentialResponse	*This message shall be empty.*
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*The device does not support that the client supplies the token.*
env:Sender ter:CapabilityViolated ter:MaxCredentials	*There is not enough space to create a new credential, see the MaxCredentials capability.*
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	*There are too many access profiles per credential, see the MaxAccessProfilesPerCredential capability.*
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	*Credential validity is not supported by the device, see the CredentialValiditySupported capability.*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*Credential access profile validity is not supported by the device, see the CredentialAccessProfileValiditySupported capability.*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Specified identifier type is not supported by the device, see the SupportedIdentifierType capability.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	*The same identifier type was used more than once.*
env:Sender ter:InvalidArgVal ter:InvalidFormatType	*Specified identifier format type is not supported by the device.*
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	*Specified identifier value is not in accordance with the FormatType definition.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	*The same combination of identifier type, format and value was used more than once (some devices may not support duplicate identifier values).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	*Specified exemption type is not supported by the device, see the SupportedExemptionType capability.*

5.3.8 ModifyCredential command

This operation modifies the specified credential.

The token of the credential to modify is specified in the token field of the Credential structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified credential.

When an existing credential is modified, the state is not modified explicitly. The only way for a client to change the state of a credential is to explicitly call the EnableCredential, DisableCredential or ResetAntipassback command.

All existing credential identifiers and credential access profiles are removed and replaced with the specified entities.

A device shall support this command as described in Table 10.

Table 10 – ModifyCredential command

ModifyCredential	Access class: WRITE_SYSTEM
Message name	**Description**
ModifyCredentialRequest	*This message contains:* • *"Credential": Details of the credential.* tcr:Credential **Credential [1][1]**
ModifyCredentialResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	*There are too many access profiles per credential, see the MaxAccessProfilesPerCredential capability.*
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	*Credential validity is not supported by the device, see the CredentialValiditySupported capability.*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*Credential access profile validity is not supported by the device, see the CredentialAccessProfileValiditySupported capability.*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Specified identifier type not is supported by the device, see the SupportedIdentifierType capability.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	*The same identifier type was used more than once.*
env:Sender ter:InvalidArgVal ter:InvalidFormatType	*Specified identifier format type is not supported by the device.*
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	*Specified identifier value is not in accordance with the FormatType definition.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	*The same combination of identifier type, format and value was used more than once (some devices may not support duplicate identifier values).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	*Specified exemption type is not supported by the device, see the SupportedExemptionType capability.*

5.3.9 DeleteCredential command

This method deletes the specified credential.

If it is associated with one or more entities some devices may not be able to delete the credential, and consequently a ReferenceInUse fault shall be generated.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 11.

Table 11 – DeleteCredential command

DeleteCredential	Access class: WRITE_SYSTEM
Message name	**Description**
DeleteCredentialRequest	*This message contains:* • *"Token": The token of the credential to delete.* pt:ReferenceToken **Token [1][1]**
DeleteCredentialResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

5.3.10 GetCredentialState command

This method returns the state for the specified credential.

If the capability ResetAntipassbackSupported is set to true, then the device shall supply the anti-passback state in the returned CredentialState structure.

A device shall support this command as described in Table 12.

Table 12 – GetCredentialState command

GetCredentialState	Access class: READ_SYSTEM_SENSITIVE
Message name	**Description**
GetCredentialStateRequest	*This message contains:* • *"Token": Token of Credential.* pt:ReferenceToken **Token [1][1]**
GetCredentialStateResponse	*This message contains:* • *"State": State of the credential.* tcr:CredentialState **State [1][1]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

5.3.11 EnableCredential command

This method is used to enable a credential.

A device shall support this command as described in Table 13.

Table 13 – EnableCredential command

EnableCredential	Access class: ACTUATE
Message name	**Description**
EnableCredentialRequest	*This message contains:* • *"Token": The token of the credential.* • *"Reason": Reason for enabling the credential.* pt:ReferenceToken **Token [1][1]** pt:Name **Reason [0][1]**
EnableCredentialResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

5.3.12 DisableCredential command

This method is used to disable a credential.

A device shall support this command as described in Table 14.

Table 14 – DisableCredential command

DisableCredential	Access class: ACTUATE
Message name	**Description**
DisableCredentialRequest	*This message contains:* • *"Token": The token of the credential.* • *"Reason": Reason for disabling the credential.* pt:ReferenceToken **Token [1][1]** pt:Name **Reason [0][1]**
DisableCredentialResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

5.3.13 ResetAntipassbackViolation command

This method is used to reset anti-passback violations for a specified credential.

A device that signals support for the ResetAntipassbackSupported capability shall support this method as described in Table 15.

Table 15 – ResetAntipassbackViolation command

ResetAntipassbackViolation	Access class: ACTUATE
Message name	**Description**
ResetAntipassbackViolationRequest	*This message contains:* • *"CredentialToken": Token of Credential.* pt:ReferenceToken **CredentialToken [1][1]**
ResetAntipassbackViolationResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

5.3.14 GetSupportedFormatTypes command

This method returns all the supported format types of a specified identifier type that is supported by the device.

A device shall support this command as described in Table 16.

Table 16 – GetSupportedFormatTypes command

GetSupportedFormatTypes	Access class: READ_SYSTEM
Message name	**Description**
GetSupportedFormatTypesRequest	*This message contains:* • *"CredentialIdentifierTypeName": Name of the credential identifier type.* xs:string **CredentialIdentifierTypeName**
GetSupportedFormatTypesResponse	*This message contains:* • *"FormatTypeInfo": Identifier format types.* tcr:CredentialIdentifierFormatTypeInfo **FormatTypeInfo [1][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Identifier type is not found.*

5.3.15 GetCredentialIdentifiers command

This method returns all the credential identifiers for a credential.

A device shall support this command as described in Table 17.

Table 17 – GetCredentialIdentifiers command

GetCredentialIdentifiers	Access class: READ_SYSTEM_SECRET
Message name	**Description**
GetCredentialIdentifersRequest	*This message contains:* • *"CredentialToken": Token of Credential.* pt:ReferenceToken **CredentialToken [1][1]**
GetCredentialIdentifersResponse	*This message contains:* • *"CredentialIdentifier": Identifiers of the credential* tcr:CredentialIdentifier **CredentialIdentifier [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

5.3.16 SetCredentialIdentifier command

This operation creates or updates a credential identifier for a credential.

If the type of specified credential identifier already exists, the current credential identifier of that type is replaced. Otherwise the credential identifier is added.

A device shall support this command as described in Table 18.

Table 18 – SetCredentialIdentifier command

SetCredentialIdentifier	Access class: WRITE_SYSTEM
Message name	**Description**
SetCredentialIdentiferRequest	*This message contains:* • *"CredentialToken": Token of Credential.* • *"CredentialIdentifier": Identifier of the credential.* pt:ReferenceToken **CredentialToken [1][1]** tcr:CredentialIdentifier **CredentialIdentifier [1][1]**
SetCredentialIdentiferResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Specified identifier type is not supported by the device, see the SupportedIdentifierType capability.*
env:Sender ter:InvalidArgVal ter:InvalidFormatType	*Specified identifier format type is not supported by the device.*
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	*Credential identifier value is not in accordance with the FormatType definition.*

5.3.17 DeleteCredentialIdentifier command

This method deletes all the identifier values for the specified type. However, if the identifier type name does not exist in the device, it will be silently ignored without any response.

A device shall support this command as described in Table 19.

Table 19 – DeleteCredentialIdentifier command

DeleteCredentialIdentifier	Access class: WRITE_SYSTEM
Message name	**Description**
DeleteCredentialIdentiferRequest	*This message contains:* • *"CredentialToken": Token of Credential.* • *"CredentialIdentifierTypeName": Identifier type name of a credential.* pt:ReferenceToken **CredentialToken [1][1]** pt:Name **CredentialIdentifierTypeName [1][1]**
DeleteCredentialIdentiferResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*
env:Receiver ter:ConstraintViolated ter:MinIdentifiersPerCredential	*At least one credential identifier is required.*

5.3.18 GetCredentialAccessProfiles command

This method returns all the credential access profiles for a credential.

A device shall support this command as described in Table 20.

Table 20 – GetCredentialAccessProfiles command

GetCredentialAccessProfiles	Access class: READ_SYSTEM
Message name	**Description**
GetCredentialAccessProfilesRequest	*This message contains:* • *"CredentialToken": Token of Credential.* pt:ReferenceToken **CredentialToken [1][1]**
GetCredentialAccessProfilesResponse	*This message contains:* • *"CredentialAccessProfile": Access profiles of the credential.* tcr:CredentialAccessProfile **CredentialAccessProfile [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

5.3.19 SetCredentialAccessProfiles command

This operation adds or updates the credential access profiles for a credential.

The device shall update the credential access profile if the access profile token in the specified credential access profile matches. Otherwise the credential access profile is added.

A device shall support this command as described in Table 21.

Table 21 – SetCredentialAccessProfiles command

SetCredentialAccessProfiles	Access class: ACTUATE
Message name	**Description**
SetCredentialAccessProfilesRequest	*This message contains:* • *"CredentialToken": Token of Credential.* • *"CredentialAccessProfile": Access profiles of the credential.* pt:ReferenceToken **CredentialToken [1][1]** tcr:CredentialAccessProfile **CredentialAccessProfile [1][unbounded]**
SetCredentialAccessProfilesResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	*There are too many access profiles per credential, see the MaxAccessProfilesPerCredential capability.*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*Credential access profile validity is not supported by the device, see the CredentialAccessProfileValiditySupported capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

5.3.20 DeleteCredentialAccessProfiles command

This method deletes credential access profiles for the specified credential token.

However, if no matching credential access profiles are found, the corresponding access profile tokens are silently ignored without any response.

A device shall support this command as described in Table 22.

Table 22 – DeleteCredentialAccessProfiles command

DeleteCredentialAccessProfiles	Access class: ACTUATE
Message name	**Description**
DeleteCredentialAccessProfilesRequest	*This message contains:* • *"CredentialToken": Token of Credential.* • *"AccessProfileToken": Tokens of Access Profiles.* pt:ReferenceToken **CredentialToken [1][1]** pt:ReferenceToken **AccessProfileToken [1][unbounded]**
DeleteCredentialAccessProfilesResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

5.4 Notification topics

5.4.1 General

Subclause 5.4 defines notification topics specific to the credential service.

5.4.2 Event overview (informative)

The credential service specifies events when the credential state changes and when credentials are changed.

The main topics for status changes are:

- tns1:Credential/State/Enabled
- tns1:Credential/State/AppViolation

The main topics for configuration change notifications are:

- tns1:Configuration/Credential/Changed
- tns1:Configuration/Credential/Removed

5.4.3 Status changes

5.4.3.1 General

The device shall provide the status change events to inform subscribed clients when credential entity status is changed. The device shall use the topics defined in 5.4.3 associated with the respective message description.

5.4.3.2 Credential

Whenever the credential state (enabled or disabled) is changed, the device shall provide the following event (the Reason field is one of the fields listed in 5.3.2.7, e.g. "pt:CredentialLost"):

```
Topic: tns1:Credential/State/Enabled
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="State"
                              Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="Reason"
                              Type="xs:string"/>
    <tt:SimpleItemDescription Name="ClientUpdated"
                              Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

ClientUpdated is set to true if the state change was initiated by the client. ClientUpdated is set to false if the device initiated the state change (e.g. because the wrong PIN was entered three times in a row).

The device shall provide the following event whenever there is any anti-passback violation:

```
Topic: tns1:Credential/State/ApbViolation
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="ApbViolation"
                              Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="ClientUpdated"
                              Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

5.4.4 Configuration changes

5.4.4.1 General

Whenever configuration data has been changed, added or removed, the device shall provide these events to inform subscribed clients.

5.4.4.2 Credential

Whenever configuration data for a credential (including credential identifiers and credential access profiles) is changed, or if a credential is added, the device shall provide the following event:

```
Topic: tns1:Configuration/Credential/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Whenever a credential is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/Credential/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

6 Access rules service

6.1 General

The access rules service defines the access profile and its access policies. The credentials are associated with an access profile for access authorization to a facility.

The access rules service defines WHEN and WHERE credentials have access. Each credential is associated with access profiles, where each access profile defines the access for a group of people. For example, employees will have access to office doors during office hours. Another example is access to an apartment by one family during all hours. Each access profile consists of a number of access policies, where each access policy defines when access is possible to an access point.

The service is flexible in such a way that it is possible to give access to something other than an access point (by setting EntityType to a QName other than AccessPointInfo).

Figure 3 shows the main data structures involved in the access rules service.

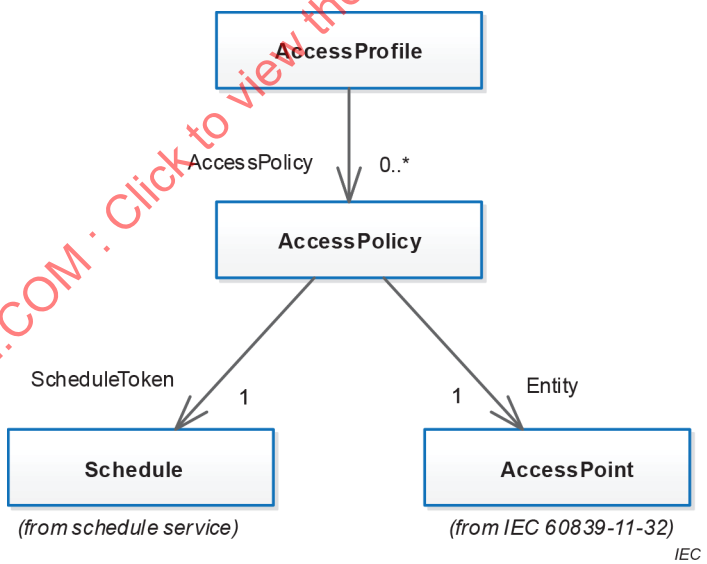


Figure 3 – Main data structures in the access rules service

6.2 Service capabilities

6.2.1 General

An ONVIF compliant device shall provide service capabilities in two ways:

1) With the GetServices method of Device service when IncludeCapability is true. Refer to the ONVIF Core Specification for more details.
2) With the GetServiceCapabilities method.

6.2.2 ServiceCapabilities data structure

The service capabilities reflect the optional functionalities of a service. The information is static and does not change during the device operation. The following capabilities are available:

- **MaxLimit**
  The maximum number of entries returned by a single Get<Entity>List or Get<Entity> request. The device shall never return more than this number of entities in a single response.
- **MaxAccessProfiles**
  Indicates the maximum number of access profiles supported by the device.
- **MaxAccessPoliciesPerAccessProfile**
  Indicates the maximum number of access policies per access profile supported by the device.
- **MultipleSchedulesPerAccessPointSupported**
  Indicates whether or not several access policies can refer to the same access point in an access profile.
- **ClientSuppliedTokenSupported**
  Indicates that the client is allowed to supply the token when creating access profiles. To enable the use of the command SetAccessProfile, the value shall be set to true.

6.2.3 GetServiceCapabilities command

This operation returns the capabilities of the access rules service.

A device shall support this command as described in Table 23.

Table 23 – GetServiceCapabilities command

GetServiceCapabilities	Access class: PRE_AUTH
Message name	**Description**
GetServiceCapabilitiesRequest	*This message shall be empty.*
GetServiceCapabilitiesResponse	*This message contains:* • *"Capabilities": The capability response message contains the requested access rules service capabilities using a hierarchical XML capability structure.* tar:ServiceCapabilities **Capabilities [1][1]**
Fault codes	**Description**
	No command specific faults.

6.3 Access profile information

6.3.1 General

Access profiles define who can access what and when.

6.3.2 Data structures

6.3.2.1 AccessProfileInfo

The AccessProfileInfo structure contains basic information about an access profile. The device shall provide the following fields for each access profile instance.

- **token**
  A service unique identifier of the access profile.
- **Name**
  A user readable name. It shall be up to 64 characters.

To provide more information, the device may include the following optional field:

- **Description**
  User readable description for the access profile. It shall be up to 1 024 characters.

6.3.2.2 AccessProfile

The AccessProfile structure contains information about the collection of access policies. The device shall include all properties of the AccessProfileInfo structure and also a list of access policies.

- **AccessPolicy**
  A list of access policy structures, where each access policy defines during which schedule an access point can be accessed.

6.3.2.3 AccessPolicy

The access policy is an association of an access point and a schedule. It defines when an access point can be accessed using an access profile which contains this access policy. If an access profile contains several access policies specifying different schedules for the same access point, this will result in a union of the schedules.

The device shall provide the following fields for each access policy instance.

- **ScheduleToken**
  Reference to the schedule used by the access policy.
- **Entity**
  Reference to the entity used by the rule engine. The entity type may be specified by the optional EntityType field explained below but is typically an access point.

To provide more information, the device may include the following optional field:

- **EntityType**
  Optional entity type; if missing, an access point type as defined by the ONVIF Access Control service should be assumed. This can also be represented by the QName value "tac:AccessPoint" where tac is the namespace of [Access Control Service Specification]. This field is provided for future extensions; it will allow an access policy being extended to cover entity types other than access points as well.

6.3.3 GetAccessProfileInfo command

This operation requests a list of AccessProfileInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 24.

Table 24 – GetAccessProfileInfo command

GetAccessProfileInfo	Access class: READ_SYSTEM
Message name	**Description**
GetAccessProfileInfoRequest	*This message contains:* • *"Token": Tokens of AccessProfileInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetAccessProfileInfoResponse	*This message contains:* • *"AccessProfileInfo": List of AccessProfileInfo items.* tar:AccessProfileInfo **AccessProfileInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

6.3.4 GetAccessProfileInfoList command

This operation requests a list of all AccessProfileInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 25.

Table 25 – GetAccessProfileInfoList command

GetAccessProfileInfoList	Access class: READ_SYSTEM
Message name	**Description**
GetAccessProfileInfoListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAccessProfileInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"AccessProfileInfo": List of AccessProfileInfo items.* xs:string **NextStartReference [0][1]** tar:AccessProfileInfo **AccessProfileInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

6.3.5 GetAccessProfiles command

This operation requests a list of AccessProfile items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 26.

Table 26 – GetAccessProfiles command

GetAccessProfiles	Access class: READ_SYSTEM
Message name	**Description**
GetAccessProfileRequest	*This message contains:* • *"Token": Tokens of Access Profile items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetAccessProfileResponse	*This message contains:* • *"AccessProfile": List of Access Profile items.* tar:AccessProfile **AccessProfile [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

6.3.6 GetAccessProfileList command

This operation requests a list of all AccessProfile items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 27.

Table 27 – GetAccessProfileList command

GetAccessProfileList	Access class: READ_SYSTEM
Message name	**Description**
GetAccessProfileListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAccessProfileListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"AccessProfile": List of Access Profile items.* xs:string **NextStartReference [0][1]** tar:AccessProfile **AccessProfile [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

6.3.7 CreateAccessProfile command

This operation creates the specified access profile in the device.

The token field of the AccessProfile structure shall be empty and the device shall allocate a token for the access profile. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

If several access policies in one access profile are specifying different schedules for the same access point, then it will result in a union of the schedules.

Figure 4 shows an example of several schedules used at one access point. Each row in the figure below corresponds to one access policy. The first and second lines are two access policies using different schedules for the same access point.

Time periods of the individual schedule					**Access policies**	
08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	**Schedule**	**Access point**
					Schedule 1	Room A Entry
					Schedule 2	Room A Entry
					Schedule 1	Room B Entry

Figure 4 – Multiple schedules per access point

Since both Schedule 1 and Schedule 2 define when Room A Entry can be accessed, a credential holder will experience a union of Schedule 1 and Schedule 2, as seen in the first row in Figure 5 below.

Time periods of the schedule union					Access policies	
08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	**Schedule**	**Access point**
					Schedule 1 and Schedule 2 union	Room A Entry
					Schedule 1	Room B Entry

Figure 5 – Result of schedule union

A device shall support this command as described in Table 28.

Table 28 – CreateAccessProfile command

CreateAccessProfile	Access class: WRITE_SYSTEM
Message name	**Description**
CreateAccessProfileRequest	*This message contains:* • *"AccessProfile": The AccessProfile to create.* tar:AccessProfile **AccessProfile [1][1]**
CreateAccessProfileResponse	*This message contains:* • *"Token": The Token of created AccessProfile.* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:MaxAccessProfiles	*There is not enough space to add a new AccessProfile, see the MaxAccessProfiles capability.*
env:Sender ter:CapabilityViolated ter:MaxAccessPoliciesPerAccessProfile	*There are too many AccessPolicies in an AccessProfile, see MaxAccessPoliciesPerAccessProfile capability.*
env:Sender ter:CapabilityViolated ter:MultipleSchedulesPerAccessPointSupported	*Multiple AccessPoints are not supported for the same schedule, see MultipleSchedulesPerAccessPointSupported capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

6.3.8 SetAccessProfile command

This method is used to synchronize an access profile in a client with the device.

If an access profile with the specified token does not exist in the device, the access profile is created. If an access profile with the specified token exists, then the access profile is modified.

A call to this method takes an access profile structure as input parameter. The token field of the access profile shall not be empty.

A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 29.

Table 29 – SetAccessProfile command

SetAccessProfile	Access class: WRITE_SYSTEM
Message name	**Description**
SetAccessProfileRequest	*This message contains:* • *"AccessProfile": The AccessProfile item to create or modify.* tar:AccessProfile **AccessProfile [1][1]**
SetAccessProfileResponse	*This message shall be empty.*
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*The device does not support that the client supplies the token.*
env:Receiver ter:CapabilityViolated ter:MaxAccessProfiles	*There is not enough space to add a new AccessProfile, see the MaxAccessProfiles capability.*
env:Sender ter:CapabilityViolated ter:MaxAccessPoliciesPerAccessProfile	*There are too many AccessPolicies in an AccessProfile, see MaxAccessPoliciesPerAccessProfile capability.*
env:Sender ter:CapabilityViolated ter:MultipleSchedulesPerAccessPointSupported	*Multiple AccessPoints are not supported for the same schedule, see MultipleSchedulesPerAccessPoint-Supported capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

6.3.9 ModifyAccessProfile command

This operation modifies the specified access profile.

The token of the access profile to modify is specified in the token field of the AccessProfile structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified access profile.

If several access policies specify different schedules for the same access point, this will result in a union of the schedules. See Figure 4 and Figure 5 above.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 30.

Table 30 – ModifyAccessProfile command

ModifyAccessProfile	Access class: WRITE_SYSTEM
Message name	**Description**
ModifyAccessProfileRequest	*This message contains:* • *"AccessProfile": The details of Access Profile.* tar:AccessProfile **AccessProfile[1][1]**
ModifyAccessProfileResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Access profile token is not found.*
env:Sender ter:CapabilityViolated ter:MaxAccessPoliciesPerAccessProfile	*There are too many AccessPolicies in an AccessProfile, see MaxAccessPoliciesPerAccessProfile capability.*
env:Sender ter:CapabilityViolated ter:MultipleSchedulesPerAccessPointSupported	*Multiple AccessPoints are not supported for the same schedule, see MultipleSchedulesPerAccessPointSupported capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

6.3.10 DeleteAccessProfile command

This operation will delete the specified access profile.

If the access profile is deleted, all access policies associated with the access profile will also be deleted.

If it is associated with one or more entities some devices may not be able to delete the access profile, and consequently a ReferenceInUse fault shall be generated.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 31.

Table 31 – DeleteAccessProfile command

DeleteAccessProfile	Access class: WRITE_SYSTEM
Message name	**Description**
DeleteAccessProfileRequest	*This message contains:* • *"Token": The token of the access profile to delete.* pt:ReferenceToken **Token [1][1]**
DeleteAccessProfileResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Access profile token is not found.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Failed to delete. Access profile token is in use.*

6.4 Notification topics

6.4.1 General

Subclause 6.4 defines notification topics specific to the access rules service.

6.4.2 Event overview (informative)

The access rules service specifies events when access profiles are changed.

The main topics for configuration change notifications are:

- tns1:Configuration/AccessProfile/Changed
- tns1:Configuration/AccessProfile/Removed

6.4.3 Configuration changes

6.4.3.1 General

Whenever configuration data has been changed, added or removed, the device shall provide these events to inform subscribed clients.

6.4.3.2 Access profile

Whenever configuration data for an access profile is changed or an access profile is added, the device shall provide the following event:

```
Topic: tns1:Configuration/AccessProfile/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AccessProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Whenever an access profile is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/AccessProfile/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AccessProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

7 Authentication behaviour service

7.1 General

This service offers commands to manage authentication behaviour and security levels.

Authentication profiles are used to define how credential holders can be granted access to an access point by defining when different security levels are required.

7.2 Example

The following example uses a schedule (refer to 8.4.2.2) that defines the following (refer to Figure 6):

- time range of 09:00 to 17:00 during regular Mondays-Fridays;
- no time ranges are defined for regular Saturdays and Sundays;
- a special day group for half-working days, with a time range of 09:00 to 13:00;
- a special day group for bank holidays, with a time range of 09:00 to 17:00.

Additionally, this service defines four security levels: "Card", "Card+PIN", "Dual Card" and "No access" (see 7.5.2.2).

By using security level constraints (see 7.4.2.4), we can map the four different schedule states to a security level and an authentication mode, as seen in Table 32:

Table 32 – Example of schedule state to security level mapping

Special Day?	Time range active?	Resulting security level	Resulting authentication mode
No/Regular day	No	"Card+PIN"	pt:SingleCredential
No/Regular day	Yes	"Card"	pt:SingleCredential
"Half-working days"	No	"Card+PIN"	pt:SingleCredential
"Half-working days"	Yes	"Card"	pt:SingleCredential
"Bank holidays"	No	"No access"	(irrelevant)
"Bank holidays"	Yes	"Card"	pt:DualCredential

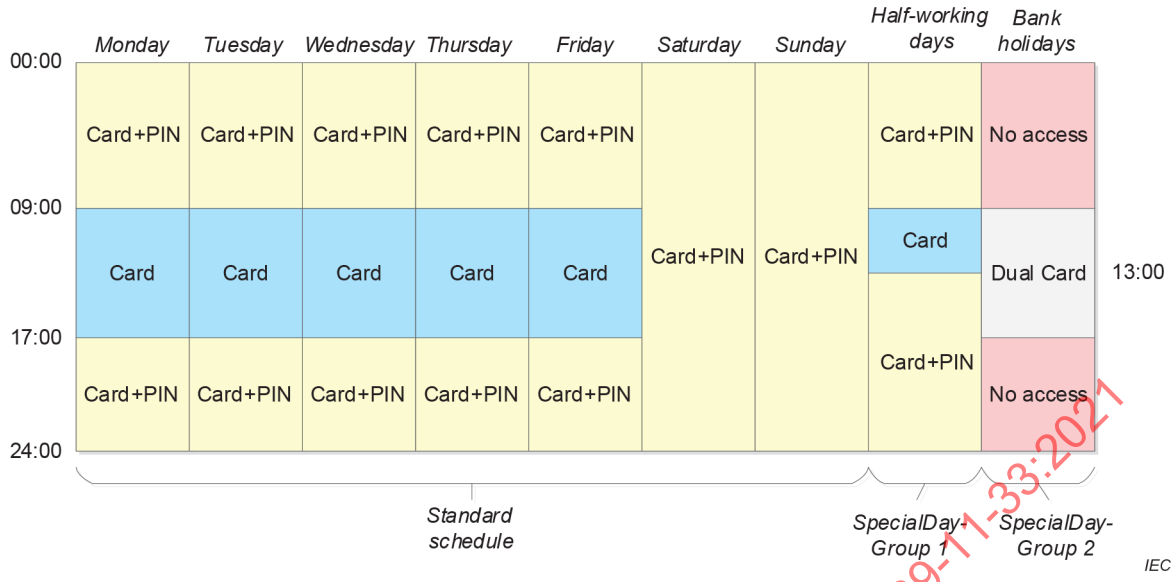


Figure 6 – Authentication behaviour example

7.3 Service capabilities

7.3.1 General

The device shall provide service capabilities in two ways:

1) With the GetServices method of Device service when IncludeCapability is true. Refer to [Core Specification] for more details.
2) With the GetServiceCapabilities method.

7.3.2 ServiceCapabilities data structure

The service capabilities reflect the optional functionalities of a service. The information is static and does not change during the device operation. The following capabilities are available:

- **MaxLimit**
  The maximum number of entries returned by a single Get<Entity>List or Get<Entity> request. The device shall never return more than this number of entities in a single response.
- **MaxAuthenticationProfiles**
  Indicates the maximum number of authentication profiles the device supports. The device shall support at least one authentication profile.
- **MaxPoliciesPerAuthenticationProfile**
  Indicates the maximum number of authentication policies per authentication profile supported by the device.
- **MaxSecurityLevels**
  Indicates the maximum number of security levels the device supports. The device shall support at least one security level.
- **MaxRecognitionGroupsPerSecurityLevel**
  Indicates the maximum number of recognition groups per security level supported by the device.
- **MaxRecognitionMethodsPerRecognitionGroup**
  Indicates the maximum number of recognition methods per recognition group supported by the device.

- **ClientSuppliedTokenSupported**
  Indicates that the client is allowed to supply the token when creating authentication profiles and security levels. To enable the use of the commands SetAuthenticationProfile and SetSecurityLevel, the value shall be set to true.

7.3.3 GetServiceCapabilities command

This operation returns the capabilities of the authentication behaviour service.

A device shall support this command as described in Table 33.

Table 33 – GetServiceCapabilities command

GetServiceCapabilities	Access class: PRE_AUTH
Message name	**Description**
GetServiceCapabilitiesRequest	*This message shall be empty.*
GetServiceCapabilitiesResponse	*This message contains:* • *"Capabilities": The capability response message contains the requested authentication behaviour service capabilities using a hierarchical XML capability structure.* tab:ServiceCapabilities **Capabilities [1][1]**
Fault codes	**Description**
	No command specific faults.

7.4 Authentication profile information

7.4.1 General

Authentication profiles are used to define authentication behaviour for a type of access point. For instance, all entrance access points are configured to require Card access during office hours, Card+PIN access during nighttime, and no access during holidays.

The authentication behaviour of an access point type is defined by associating security levels with schedules. When the schedule is active, the specified security level is required.

If a certain point in time is not covered by any schedule, then the access point is set to the default security level.

Figure 7 shows an overview of the related objects of an authentication profile.

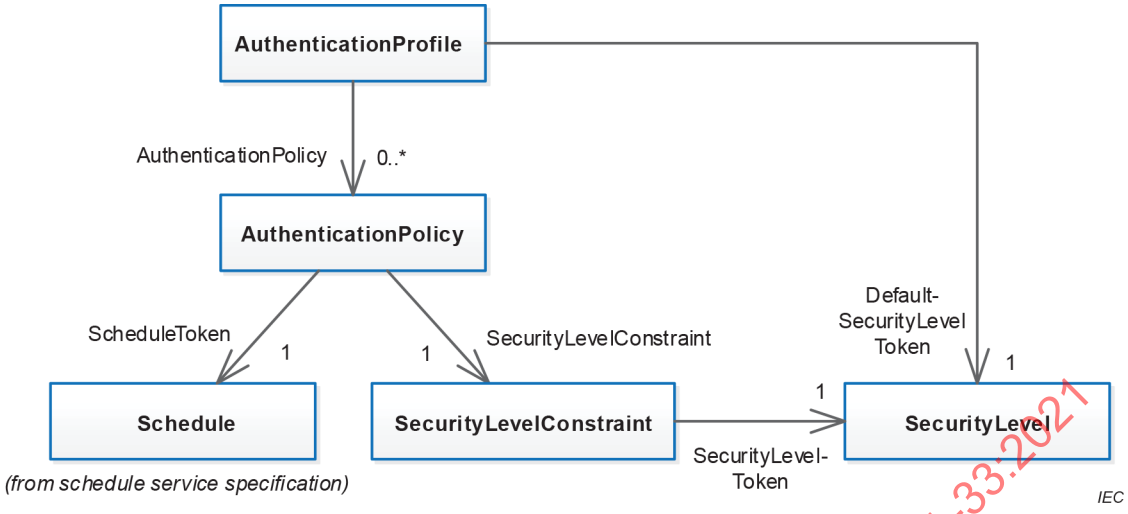


Figure 7 – Related objects of an authentication profile

7.4.2 Data structures

7.4.2.1 AuthenticationProfileInfo

The AuthenticationProfileInfo structure contains information of a specific authentication profile instance.

The device shall provide the following fields for each AuthenticationProfileInfo instance:

- **token**
  A service unique identifier of the authentication profile.
- **Name**
  A descriptive name, such as "Entrance doors – entry". It shall be up to 64 characters.

To provide more information, the device may include the following optional field:

- **Description**
  User readable description for the authentication profile. It shall be up to 1 024 characters.

7.4.2.2 AuthenticationProfile

The AuthenticationProfile structure shall include all properties of the AuthenticationProfileInfo structure and also a default security level, an authentication mode, and a list of AuthenticationPolicy instances.

The device shall provide the following fields for each AuthenticationProfile instance:

- **DefaultSecurityLevelToken**
  The default security level is used if none of the authentication policies has a schedule covering the time of access (or if no authentication policies are defined).
- **AuthenticationPolicy**
  Each authentication policy associates a security level with a schedule (during which the specified security level will be required at the access point).

7.4.2.3 AuthenticationPolicy

The authentication policy is an association of a security level and a schedule. It defines when a certain security level is required to grant access to a credential holder. Each security level is given a unique priority. If authentication policies have overlapping schedules, the security level with the highest priority is used.

The device shall provide the following fields for each authentication policy instance:

- **ScheduleToken**
  Reference to the schedule used by the authentication policy.
- **SecurityLevelConstraint**
  A list of security level constraint structures defining the conditions for what security level to use.

  Minimum one security level constraint shall be specified.

7.4.2.4 SecurityLevelConstraint

This structure defines what security level should be active depending on the state of the schedule. The state of a schedule has two boolean values corresponding to four different states:

- the standard schedule is active, and it is currently no special day;
- the standard schedule is inactive, and it is currently no special day;
- it is a special day, and a time period defined in the special days' schedule is active;
- it is a special day, but no time periods defined in the special days' schedule are active.

If the state of the schedule corresponds to the ActiveRegularSchedule and ActiveSpecialDaySchedule settings in this structure, then the specified security level will be used.

Note that if the device does not support special days, the value of the field ActiveSpecialDay-Schedule will be ignored.

The device shall provide the following fields for each security level constraint instance:

- **ActiveRegularSchedule**
  Corresponds to the Active field in the ScheduleState structure in [Schedule Service Specification].
- **ActiveSpecialDaySchedule**
  Corresponds to the SpecialDay field in the ScheduleState structure in [Schedule Service Specification].
  This field will be ignored if the device does not support special days.
- **AuthenticationMode**
  Defines the mode of authentication. Authentication modes starting with the prefix pt: are reserved to define ONVIF-specific authentication modes. For custom defined authentication modes, free text can be used. The following authentication modes are defined by ONVIF:
  - pt:SingleCredential Normal mode where only one credential holder is required to be granted access.
  - pt:DualCredential Two credential holders are required to be granted access.
- **SecurityLevelToken**
  Reference to the security level used by the authentication policy.

7.4.3 GetAuthenticationProfileInfo command

This operation requests a list of AuthenticationProfileInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 34.

Table 34 – GetAuthenticationProfileInfo command

GetAuthenticationProfileInfo	Access class: READ_SYSTEM
Message name	**Description**
GetAuthenticationProfileInfoRequest	*This message contains:* • *"Token": Tokens of AuthenticationProfileInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetAuthenticationProfileInfoResponse	*This message contains:* • *"AuthenticationProfileInfo": List of AuthenticationProfileInfo items.* tab:AuthenticationProfileInfo **AuthenticationProfileInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

7.4.4 GetAuthenticationProfileInfoList command

This operation requests a list of all AuthenticationProfileInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 35.

Table 35 – GetAuthenticationProfileInfoList command

GetAuthenticationProfileInfoList	Access class: READ_SYSTEM
Message name	**Description**
GetAuthenticationProfileInfoListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAuthenticationProfileInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"AuthenticationProfileInfo": List of AuthenticationProfileInfo items.* xs:string **NextStartReference [0][1]** tab:AuthenticationProfileInfo **AuthenticationProfileInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

7.4.5 GetAuthenticationProfiles command

This operation requests a list of AuthenticationProfile items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 36.

Table 36 – GetAuthenticationProfiles command

GetAuthenticationProfiles	Access class: READ_SYSTEM
Message name	**Description**
GetAuthenticationProfileRequest	*This message contains:* • *"Token": Tokens of AuthenticationProfile items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetAuthenticationProfileResponse	*This message contains:* • *"AuthenticationProfile": List of AuthenticationProfile items.* tab:AuthenticationProfile **AuthenticationProfile [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

7.4.6 GetAuthenticationProfileList command

This operation requests a list of all AuthenticationProfile items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 37.

Table 37 – GetAuthenticationProfileList command

GetAuthenticationProfileList	Access class: READ_SYSTEM
Message name	**Description**
GetAuthenticationProfileListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAuthenticationProfileListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"AuthenticationProfile": List of AuthenticationProfile items.* xs:string **NextStartReference [0][1]** tab:AuthenticationProfile **AuthenticationProfile [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

7.4.7 CreateAuthenticationProfile command

This operation creates the specified authentication profile in the device.

The token field of the AuthenticationProfile structure shall be empty and the device shall allocate a token for the authentication profile. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

A device shall support this command as described in Table 38.

Table 38 – CreateAuthenticationProfile command

CreateAuthenticationProfile	Access class: WRITE_SYSTEM
Message name	**Description**
CreateAuthenticationProfileRequest	*This message contains:* • *"AuthenticationProfile": The authentication profile to create.* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
CreateAuthenticationProfileResponse	*This message contains:* • *"Token": The token of the created authentication profile.* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:MaxAuthenticationProfiles	*There is not enough space to add a new authentication profile, see MaxAuthenticationProfiles capability.*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*There are too many AuthenticationPolicy entities referred in this AuthenticationProfile, see MaxPoliciesPerAuthenticationProfile capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

7.4.8 SetAuthenticationProfile command

This method is used to synchronize an authentication profile in a client with the device.

If an authentication profile with the specified token does not exist in the device, the authentication profile is created. If an authentication profile with the specified token exists, then the authentication profile is modified.

A call to this method takes an AuthenticationProfile structure as input parameter. The token field of the AuthenticationProfile shall not be empty.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device that signals support for the ClientSuppliedTokenSupported capability shall support this method as described in Table 39.

Table 39 – SetAuthenticationProfile command

SetAuthenticationProfile	Access class: WRITE_SYSTEM
Message name	**Description**
SetAuthenticationProfileRequest	*This message contains:* • *"AuthenticationProfile": The AuthenticationProfile to create or modify.* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
SetAuthenticationProfileResponse	*This message shall be empty.*
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*The device does not support that the client supplies the token.*
env:Receiver ter:CapabilityViolated ter:MaxAuthenticationProfiles	*There is not enough space to add a new AuthenticationProfile, see MaxAuthenticationProfiles capability.*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*There are too many AuthenticationPolicy entities referred in this AuthenticationProfile, see MaxPoliciesPerAuthenticationProfile capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*
env:Sender ter:CapabilityViolated ter:SupportedSecurityLevels	*The updated authentication profile contains security levels that are not supported by an access point that is referring to this authentication profile (some devices may not validate this).*

7.4.9 ModifyAuthenticationProfile command

This operation modifies the specified authentication profile.

The token of the authentication profile to modify is specified in the token field of the AuthenticationProfile structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified authentication profile.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 40.

Table 40 – ModifyAuthenticationProfile command

ModifyAuthenticationProfile	Access class: WRITE_SYSTEM
Message name	**Description**
ModifyAuthenticationProfileRequest	*This message contains:* • *"AuthenticationProfile": The AuthenticationProfile to modify.* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
ModifyAuthenticationProfileResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*The specified token is not found.*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*There are too many AuthenticationPolicy entities referred in this AuthenticationProfile, see MaxPoliciesPerAuthenticationProfile capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*
env:Sender ter:CapabilityViolated ter:SupportedSecurityLevels	*The updated authentication profile contains security levels that are not supported by an access point that is referring to this authentication profile (some devices may not validate this).*

7.4.10 DeleteAuthenticationProfile command

This operation deletes the specified authentication profile.

If the authentication profile is deleted, all authentication policies associated with the authentication profile will also be deleted.

If it is associated with one or more entities some devices may not be able to delete the authentication profile, and consequently a ReferenceInUse fault shall be generated.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 41.

Table 41 – DeleteAuthenticationProfile command

DeleteAuthenticationProfile	Access class: WRITE_SYSTEM
Message name	**Description**
DeleteAuthenticationProfileRequest	*This message contains:* • *"Token": The token of the AuthenticationProfile to delete.* pt:ReferenceToken **Token [1][1]**
DeleteAuthenticationProfileResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*AuthenticationProfile token is not found.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Failed to delete, AuthenticationProfile token is in use.*

7.5 Security level information

7.5.1 General

Security levels are defined as individual recognition methods, combinations of recognition methods (using logical AND or OR), or no recognition methods (open). Security levels are given explanatory names, such as "Card", "Card+ PIN", "Fingerprint or Iris", "Open".

Possible authentication factors are Card, PIN, Fingerprint, Face, Iris, Vein, Palm and REX (that is not hardwired to directly unlock, but where the device can do some action, such as taking a decision).

Figure 8 shows an overview of the related objects of a security level.

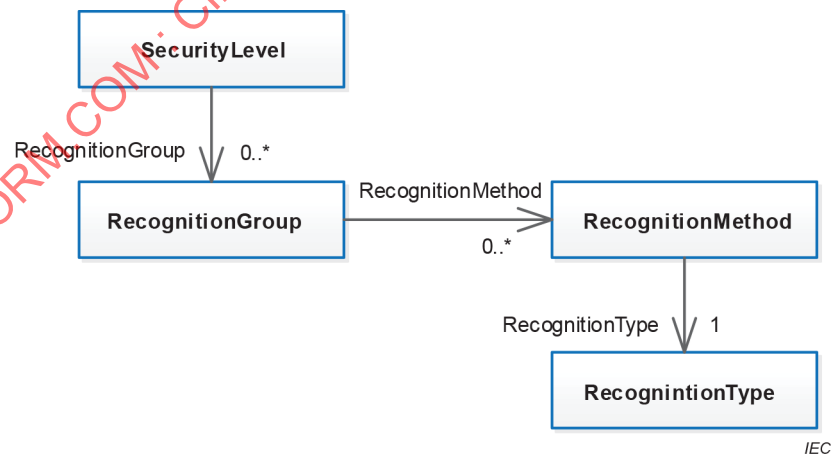


Figure 8 – Related objects of a security level

7.5.2 Data structures

7.5.2.1 SecurityLevelInfo

The SecurityLevelInfo structure contains information of a specific security level instance.

The device shall provide the following fields for each SecurityLevelInfo instance:

- **token**
  A service-unique identifier of the security level.
- **Name**
  User readable name. It shall be up to 64 characters.
- **Priority**
  A higher number indicates that the security level is considered more secure than security levels with lower priorities. The priority is used when an authentication profile has overlapping schedules with different security levels. When an access point is accessed, the authentication policies are walked through in priority order (highest priority first). When a schedule is found covering the time of access, the associated security level is used and processing stops. Two security levels cannot have the same priority.

To provide more information, the device may include the following optional field:

- **Description**
  User readable description for the security level. It shall be up to 1 024 characters.

7.5.2.2 SecurityLevel

The SecurityLevel structure shall include all properties of the SecurityLevelInfo structure and also a set of recognition groups.

The recognition groups are used to define a logical OR between the groups. Each recognition group consists of one or more recognition methods.

Example: One recognition group contains the recognition methods pt:Card and pt:Fingerprint. Another group contains the recognition methods pt:Card and pt:Face. The resulting effect is that the access point will require either Card+Fingerprint, or Card+Face.

The order of the requested recognition methods can be defined at the access point using the order field in the recognition method structure.

The example in Figure 9 shows three different security levels. The first has no recognition groups and the access point will not require any authentication when active. The second is described above. The third defines one recognition group but with no recognition methods, which has the same effect as if the access point is disabled.

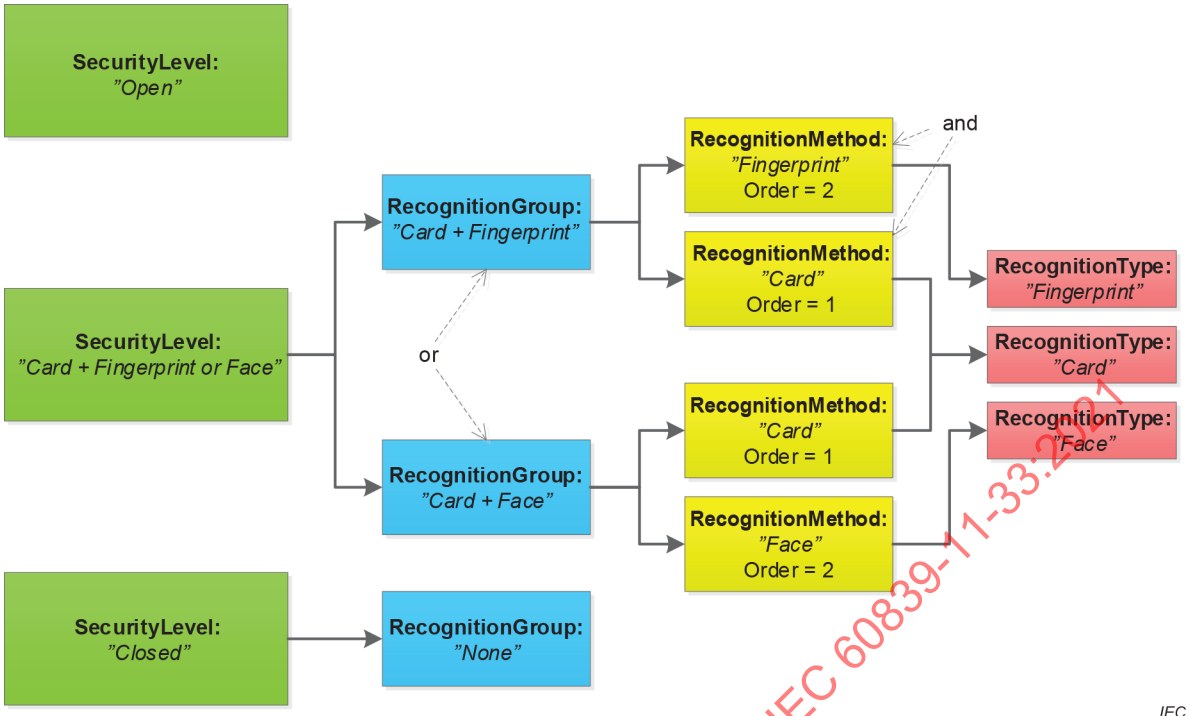


Figure 9 – Security level examples

The device shall provide the following field for each SecurityLevel instance:

- **RecognitionGroup**
  The recognition groups are used to define a logical OR between the groups. Each recognition group consists of one or more recognition methods.

  No recognition groups mean that the access point is open.

7.5.2.3 RecognitionGroup

The device shall provide the following field for each RecognitionGroup instance:

- **RecognitionMethod**
  A list of recognition methods to request for at the access point.

7.5.2.4 RecognitionMethod

Recognition is the action of identifying authorized users requesting access by the comparison of presented credential data with recorded credential data. A recognition method is either memorized, biometric or held within a physical credential. A recognition type is either a recognition method or a physical input such as a request-to-exit button.

The device shall provide the following fields for each RecognitionMethod instance:

- **RecognitionType**
  The requested type of recognition. Is of type text. Can be either one of the following reserved ONVIF types, or a custom type:
  - pt:Card Card is used as recognition method.
  - pt:PIN PIN is used as recognition method.
  - pt:Fingerprint A fingerprint scan is used as recognition method.
  - pt:Face A facial scan is used as recognition method.
  - pt:Iris An iris scan is used as recognition method.

- pt:Vein A vein scan is used as recognition method.
- pt:Palm A palm scan is used as recognition method.
- pt:REX A request-to-exit button is used to be granted access.

- **Order**
  The order value defines when this recognition method will be requested in relation to the other recognition methods in the same security level. A lower number indicates that the recognition method will be requested before recognition methods with a higher number.

7.5.3 GetSecurityLevelInfo command

This operation requests a list of SecurityLevelInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 42.

Table 42 – GetSecurityLevelInfo command

GetSecurityLevelInfo	Access class: READ_SYSTEM
Message name	**Description**
GetSecurityLevelInfoRequest	*This message contains:* • *"Token": Tokens of SecurityLevelInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetSecurityLevelInfoResponse	*This message contains:* • *"SecurityLevelInfo": List of SecurityLevelInfo items.* tab:SecurityLevelInfo **SecurityLevelInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

7.5.4 GetSecurityLevelInfoList command

This operation requests a list of all SecurityLevelInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 43.

Table 43 – GetSecurityLevelInfoList command

GetSecurityLevelInfoList	Access class: READ_SYSTEM
Message name	**Description**
GetSecurityLevelInfoListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSecurityLevelInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"SecurityLevelInfo": List of SecurityLevelInfo items.* xs:string **NextStartReference [0][1]** tab:SecurityLevelInfo **SecurityLevelInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

7.5.5 GetSecurityLevels command

This operation requests a list of SecurityLevel items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 44.

Table 44 – GetSecurityLevels command

GetSecurityLevels	Access class: READ_SYSTEM
Message name	**Description**
GetSecurityLevelsRequest	*This message contains:* • *"Token": Tokens of the SecurityLevel items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetSecurityLevelsResponse	*This message contains:* • *"SecurityLevel": List of SecurityLevel items.* tab:SecurityLevel **SecurityLevel [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

7.5.6 GetSecurityLevelList command

This operation requests a list of all SecurityLevel items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 45.

Table 45 – GetSecurityLevelList command

GetSecurityLevelList	Access class: READ_SYSTEM
Message name	**Description**
GetSecurityLevelListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSecurityLevelListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"SecurityLevel": List of SecurityLevel items.* xs:string **NextStartReference [0][1]** tab:SecurityLevel **SecurityLevel [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

7.5.7 CreateSecurityLevel command

This operation creates the specified security level in the device.

The token field of the SecurityLevel structure shall be empty and the device shall allocate a token for the security level. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

A device shall support this command as described in Table 46.

Table 46 – CreateSecurityLevel command

CreateSecurityLevel	Access class: WRITE_SYSTEM
Message name	**Description**
CreateSecurityLevelRequest	*This message contains:* • *"SecurityLevel": The security level to create.* tab:SecurityLevel **SecurityLevel [1][1]**
CreateSecurityLevelResponse	*This message contains:* • *"Token": The token of created security level.* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:MaxSecurityLevels	*There is not enough space to add new SecurityLevel items, see the MaxSecurityLevels capability.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*There are too many recognition method groups in a SecurityLevel, see MaxRecognitionGroupsPerSecurityLevel capability.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognitionGroup	*There are too many recognition methods in a recognition group, see MaxRecognitionMethodsPerRecognitionGroup capability.*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*The priority of the security level is already used by another security level.*

7.5.8 SetSecurityLevel command

This method is used to synchronize a security level in a client with the device.

If a security level with the specified token does not exist in the device, the security level is created. If a security level with the specified token exists, then the security level is modified.

A call to this method takes a SecurityLevel structure as input parameter. The token field of the SecurityLevel shall not be empty.

A device that signals support for the ClientSuppliedTokenSupported capability shall support this method as described in Table 47.

Table 47 – SetSecurityLevel command

SetSecurityLevel	Access class: WRITE_SYSTEM
Message name	**Description**
SetSecurityLevelRequest	*This message contains:* • *"SecurityLevel": The security level to create or modify.* tab:SecurityLevel **SecurityLevel [1][1]**
SetSecurityLevelResponse	*This message shall be empty.*
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*The device does not support that the client supplies the token.*
env:Receiver ter:CapabilityViolated ter:MaxSecurityLevels	*There is not enough space to add new SecurityLevel items, see the MaxSecurityLevels capability.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*There are too many recognition method groups in a SecurityLevel, see MaxRecognitionGroupsPerSecurityLevel capability.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognitionGroup	*There are too many recognition methods in a recognition method group, see MaxRecognitionMethodsPerRecognition-Group capability.*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*The priority of the security level is already used by another security level.*
env:Sender ter:InvalidArgVal ter:MissingToken	*The token of the security level item shall be specified.*

7.5.9 ModifySecurityLevel command

This operation modifies the specified security level.

The token of the security level to modify is specified in the token field of the SecurityLevel structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified security level.

A device shall support this command as described in Table 48.

Table 48 – ModifySecurityLevel command

ModifySecurityLevel	Access class: WRITE_SYSTEM
Message name	**Description**
ModifySecurityLevelRequest	*This message contains:* • *"SecurityLevel": The security level to modify.* tab:SecurityLevel **SecurityLevel [1][1]**
ModifySecurityLevelResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*The specified token is not found.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*There are too many recognition method groups in a SecurityLevel, see MaxRecognitionGroupsPerSecurityLevel capability.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognitionGroup	*There are too many recognition methods in a recognition method group, see MaxRecognitionMethodsPerRecognition-Group capability.*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*The priority of the security level is already used by another security level.*

7.5.10 DeleteSecurityLevel command

This method deletes the specified security level.

If it is associated with one or more entities some devices may not be able to delete the security level, and consequently a ReferenceInUse fault shall be generated.

A device shall support this command as described in Table 49.

Table 49 – DeleteSecurityLevel command

DeleteSecurityLevel	Access class: WRITE_SYSTEM
Message name	**Description**
DeleteSecurityLevelRequest	*This message contains:* • *"Token": The token of the security level item to delete.* pt:ReferenceToken **Token [1][1]**
DeleteSecurityLevelResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Security level token is not found.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Failed to delete, security level token is in use.*

7.6 Notification topics

7.6.1 General

Subclause 7.6 defines notification topics specific to the authentication behaviour service.

7.6.2 Event overview (informative)

The authentication behaviour service specifies events when authentication profiles or security levels are changed.

The main topics for configuration change notifications are:

- tns1:Configuration/AuthenticationProfile/Changed
- tns1:Configuration/AuthenticationProfile/Removed
- tns1:Configuration/SecurityLevel/Changed
- tns1:Configuration/SecurityLevel/Removed

7.6.3 Configuration changes

7.6.3.1 General

Whenever configuration data has been changed, added or removed, the device shall provide these events to inform subscribed clients.

7.6.3.2 Authentication profile

Whenever configuration data for an authentication profile is changed or an authentication profile is added, the device shall provide the following event:

```
Topic: tns1:Configuration/AuthenticationProfile/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AuthenticationProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Whenever an authentication profile is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/AuthenticationProfile/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AuthenticationProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

7.6.3.3 Security level

Whenever configuration data for a security level is changed or a security level is added, the device shall provide the following event:

```
Topic: tns1:Configuration/SecurityLevel/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SecurityLevelToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Whenever a security level is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/SecurityLevel/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SecurityLevelToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

8 Schedule service

8.1 General

The schedule service provides functions to manage schedules and special days.

Schedules can, for example, be used to define when access is granted or when a video camera should record.

The schedule service supports schedules in both local time and UTC time. Typically a schedule is in local time.

Figure 10 shows the main data structures involved in the schedule service.

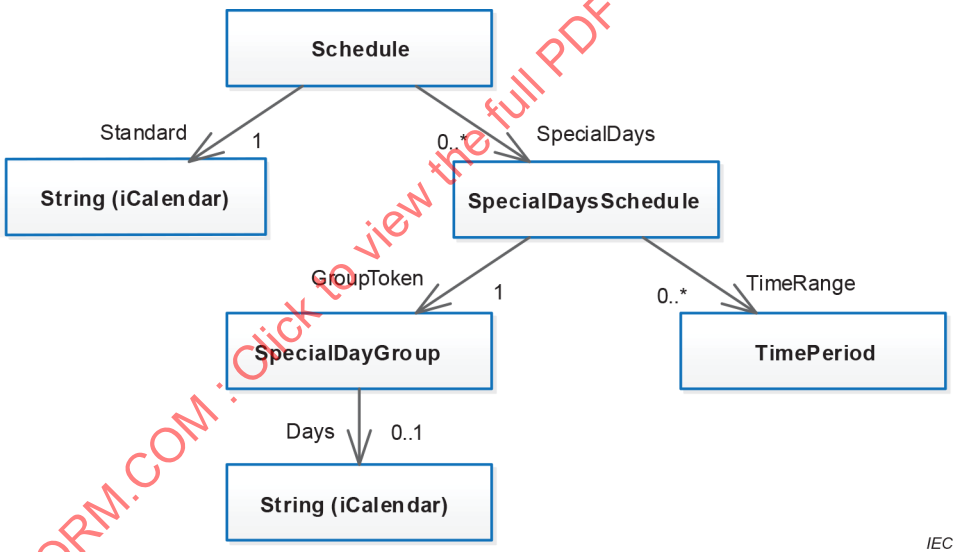


Figure 10 – Main data structures in the schedule service

A group of special days may be used to define what times the facility is accessible during holidays. Another group of special days may be used to define what times the facility is accessible during half-working days.

Special day groups are typically reused by different schedules. Consider the following example:

A EACS system is shared by two sites; one in Sweden, and one in Poland. All Polish holidays are grouped in a special day group called "Polish holidays". All Swedish holidays are grouped in a special day group called "Swedish holidays". In addition Sweden has half-working days before some of the bank holidays, so another group is defined and called "Swedish half-days".

A schedule called "Polish employee schedule" is created, defining access to the office for Polish employees during all days of the week, including the weekend (e.g. access on Monday-Friday from 07:00 to 20:00, and access on Saturday-Sunday from 10:00 to 14:00). The schedule is linked to the "Polish holidays" group defining the access to work during these days (e.g. no access during holidays).

Similarly, a schedule called "Swedish employee schedule" is created, defining access to the office for all Swedish employees during all days of the week, including the weekend. The schedule is linked to both the "Swedish holidays" group (e.g. no access) and the "Swedish half-days" group (e.g. access from 07:00 to 13:00).

Schedules and special days are defined by iCalendar vEvents. iCalendar is a very flexible format where recurrence types such as yearly, monthly, weekly, daily, can be used. For more details, see RFC 5545.

To briefly explain the iCalendar format, consider the following examples (only parts of the iCalendar structures are shown below):

An iCalendar event starting at 20:00 on November 25, 2014, and ending at 02:00 on November 26, 2014:

```
BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART:20141125T200000
DTEND:20141126T020000
END:VEVENT
END:VCALENDAR
```

A weekly recurring iCalendar event each Tuesday and Thursday from 14:00 to 16:00, where the first occurrence is on November 25, 2014, and the last occurrence is on December 9, 2014 (5 occurrences):

```
BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART:20141125T140000
DTEND:20141125T160000
RRULE:FREQ=WEEKLY;COUNT=5;BYDAY=TU,TH
END:VEVENT
END:VCALENDAR
```

The following example has exactly the same effect as the previous example, but the end date of the series is specified instead of the number of occurrences:

```
BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART:20141125T140000
DTEND:20141125T160000
RRULE:FREQ=WEEKLY;UNTIL:20141209T140000;BYDAY=TU,TH
END:VEVENT
END:VCALENDAR
```

Only a certain subset of the iCalendar specification is required to be supported to get a usable service. Support for additional features within the iCalendar specification is reflected in the ServiceCapabilities.

8.2 Recurrence

8.2.1 General

When the ExtendedRecurrenceSupported capability is set to true then extended iCalendar recurrence format is supported. Special days can still be used but may not be needed.

For devices not supporting extended iCalendar recurrence, the ExtendedRecurrenceSupported capability should be set to false. The iCalendar "FREQ" and "BYDAY" keywords are used to define weekly recurring events per day in the week (see iCalendar examples above). When the ExtendedRecurrenceSupported capability is set to false, then the year of the start date of recurring events shall be set to 1970, and the occurrence count (optionally the until date) shall be omitted (i.e. forever recurring events).

Figure 11 shows the effect of the ExtendedRecurrenceSupported capability settings for schedule and special days respectively.

	ExtendedRecurrenceSupported	
	FALSE	**TRUE**
Schedules	Weekly recurrence	Extended recurrence
Special days	No recurrence allowed	Extended recurrence

Figure 11 – Recurrence support matrix

In 8.2.2 and 8.2.3, the ABNF syntax shall be used (see RFC 5234). The rule names to the left of the equal sign (dtstart, dtend and recur-rule-part) are redefining the rules in RFC 5545. The rule names to the right of the equal sign (date-month, date-mday, time-hour, time-minute, time-second, bymolist, bywdaylist, bymodaylist, weekday and enddate) shall be defined in accordance with RFC 5545.

8.2.2 Weekly recurrence

Only the weekly iCalendar recurrence type shall be supported. Non-recurring events and other recurring types are not supported.

The device shall only accept recurring events starting with the year '1970' (the month and day is needed to match the week day of the recurrence).

```
dtstart = "DTSTART" ":" "1970" date-month date-mday
          "T" time-hour time-minute time-second
```

The device shall only allow end dates no later than midnight of the date specified in the start date,

```
dtend = "DTEND" ":" "1970" date-month date-mday
        "T" time-hour time-minute time-second
```

where date-month and date-mday shall be either the same as for dtstart or one day later but with time-hour = "00", time-minute = "00" and time-second = "00".

Only local times are supported.

The device shall not accept an occurrence count or until date in recurring events. The following recurrence pattern is allowed:

```
recur-rule-part = "WEEKLY" [ ";" "BYDAY" "=" (weekday *("," weekday)) ]
```

8.2.3 Extended recurrence

The device shall support the following recurrence pattern:

```
recur-rule-part =
       (  ( "YEARLY" )
        / ( "YEARLY" ";" "BYMONTH" "=" bymolist )
        / ( "MONTHLY" )
        / ( "MONTHLY" ";" "BYDAY" "=" bywdaylist )
        / ( "MONTHLY" ";" "BYMONTHDAY" "=" bymodaylist )
        / ( "WEEKLY" )
        / ( "WEEKLY" ";" "BYDAY" "=" (weekday *("," weekday)) )
        / ( "DAILY" )
        / ( "HOURLY" )
        / ( "MINUTELY" )
        / ( "SECONDLY" ) )
       [ ";" "INTERVAL" "=" 1*DIGIT ]
       [ ";" "COUNT" "=" 1*DIGIT / ";" "UNTIL" "=" enddate ]
```

8.2.4 Standard schedule recurrence

Single occurrences in a series of recurring events can be deleted to create exceptions from the recurring events.

Figure 12 illustrates a recurring series of events. Each vertical bar represents the 24-hour day, and the green parts represent events (time periods). Events occur each Monday, but there is an exception on the fourth Monday, when no event occurs.

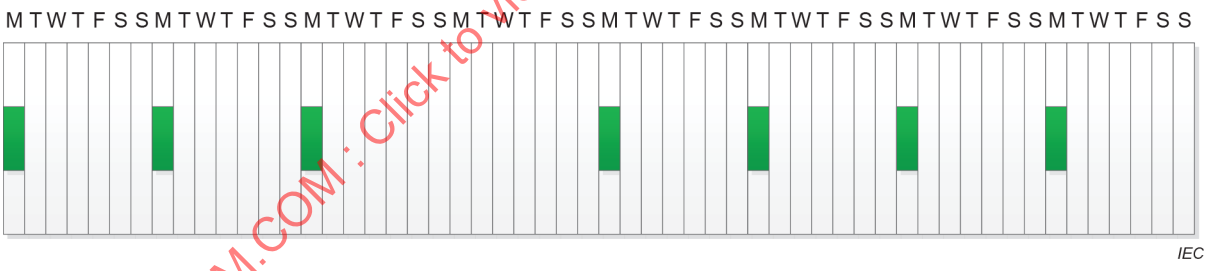


Figure 12 – Recurring events with an exception

8.2.5 Special day recurrence

In contrast to exceptions, the schedule service also supports special days. Special days are used to define different behaviour during for instance holidays or half-working days. Examples of different behaviour includes video recording with a different frame rate or different access times to a facility. Special days are grouped together and given names such as "US holidays", "UK holidays", "Swedish half-days".

Figure 13 illustrates the same series of recurring events, but instead of an exception, the fourth Monday is treated as a special day with different behaviour.

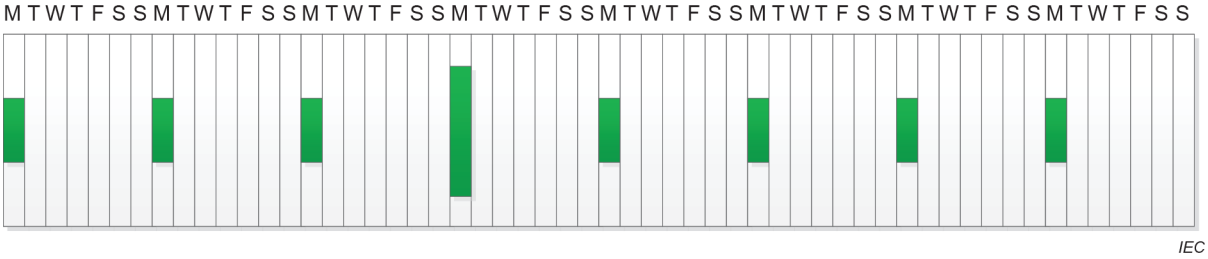


Figure 13 – Recurring events with a special day

In the EACS domain, supporting different behaviour during special days is essential. Most traditional access control units on the market today only handle weekly recurrence. Generally they do not support exceptions, but they usually support special days.

8.3 Service capabilities

8.3.1 General

The device shall provide service capabilities in two ways:

1) With the GetServices method of Device service when IncludeCapability is true. Refer to [Core Specification] for more details.
2) With the GetServiceCapabilities method.

8.3.2 ServiceCapabilities data structure

The service capabilities reflect the optional functionalities of a service. The information is static and does not change during the device operation. The following capabilities are available:

- **MaxLimit**
  The maximum number of entries returned by a single Get<Entity>List or Get<Entity> request. The device shall never return more than this number of entities in a single response.
- **MaxSchedules**
  Indicates the maximum number of schedules the device supports. The device shall support at least one schedule.
- **MaxTimePeriodsPerDay**
  Indicates the maximum number of time periods per day the device supports in a schedule including special days' schedule. The device shall support at least one time period per day.
- **SpecialDaysSupported**
  If this capability is supported, the device shall support special days.
- **MaxSpecialDayGroups**
  Indicates the maximum number of special day group entities the device supports. The device shall support at least one SpecialDayGroup entity if SpecialDaysSupported is set to true. This value is ignored if SpecialDaysSupported is set to false.
- **MaxDaysInSpecialDayGroup**
  Indicates the maximum number of days per SpecialDayGroup entity the device supports. The device shall support at least one day per SpecialDayGroup entity if SpecialDaysSupported is set to true. This value is ignored if SpecialDaysSupported is set to false.
- **MaxSpecialDaysSchedules**
  Indicates the maximum number of SpecialDaysSchedule entities referred by a schedule that the device supports. This value is ignored if SpecialDaysSupported is set to false.

- **ExtendedRecurrenceSupported**
  If ExtendedRecurrenceSupported = true, schedules shall support extended recurrence in accordance with 8.2.3.

  If ExtendedRecurrenceSupported = true and SpecialDaysSupported = true, then special days supports extended recurrence in accordance with 8.2.3.

  If ExtendedRecurrenceSupported = false, only weekly recurrence shall be supported for schedules (in accordance with 8.2.2) and no recurrence shall be supported for special days.
- **StateReportingSupported**
  If this capability is set to true, the device shall implement the GetScheduleState command, and shall notify subscribing clients whenever schedules become active or inactive.
- **ClientSuppliedTokenSupported**
  Indicates that the client is allowed to supply the token when creating schedules and special day groups. To enable the use of the commands SetSchedule and SetSpecialDayGroup, the value shall be set to true.

8.3.3 GetServiceCapabilities command

This operation returns the capabilities of the schedule service.

A device shall support this command as described in Table 50.

Table 50 – GetServiceCapabilities command

GetServiceCapabilities	Access class: PRE_AUTH
Message name	**Description**
GetServiceCapabilitiesRequest	*This message shall be empty.*
GetServiceCapabilitiesResponse	*This message contains:* • *"Capabilities": The capability response message contains the requested schedule service capabilities using a hierarchical XML capability structure.* tsc:ServiceCapabilities **Capabilities [1][1]**
Fault codes	**Description**
	No command specific faults.

8.4 Schedule information

8.4.1 General

A schedule is a set of time periods which may also include one or more special days' schedules.

8.4.2 Data structures

8.4.2.1 ScheduleInfo

The ScheduleInfo type represents the schedule as a physical object. The structure contains information of a specific schedule instance.

The device shall provide the following fields for each ScheduleInfo instance:

- **token**
  A service unique identifier of the schedule.
- **Name**
  A user readable name. It shall be up to 64 characters.

To provide more information, the device may include the following optional field:

- **Description**
  User readable description for the schedule. It shall be up to 1 024 characters.

8.4.2.2 Schedule

The schedule structure shall include all properties of the ScheduleInfo structure and also the standard events (iCalendar format) and a list of SpecialDaysSchedule instances.

The device shall provide the following fields for each schedule instance:

- **Standard**
  An iCalendar structure that defines a number of events. Events can be recurring or non-recurring. The events can, for instance, be used to control when a camera should record or when a facility is accessible.

  Some devices might not be able to fully support all the features of iCalendar. Setting the service capability ExtendedRecurrenceSupported to false will enable more devices to be ONVIF compliant. Is of type string (but contains an iCalendar structure).
- **SpecialDays**
  For devices that are not able to support all the features of iCalendar, supporting special days is essential. Each SpecialDaysSchedule instance defines an alternate set of time periods that overrides the regular schedule for a specified list of special days. Is of type SpecialDaysSchedule.

8.4.2.3 SpecialDaysSchedule

An override schedule that defines alternate time periods for a group of special days.

If multiple SpecialDaysSchedule structures contain conflicting time periods for the same special day, it will result in a union of the time periods.

Figure 14 shows the use of SpecialDaysSchedule instances. A Schedule instance called "Employee schedule" holds an iCalendar structure defining the regular access, and also holds a number of SpecialDaysSchedule instances. Each SpecialDaysSchedule holds a token to a SpecialDayGroup and zero or more TimePeriod instances.

In this example, the first SpecialDaysSchedule instance holds the token to the "Holidays" special day group, but holds no time periods. The result is that no access is granted on the special days listed in the "Holidays" special day group.

The second SpecialDaysSchedule instance holds the token to the "Half-days" special day group, and also holds one time period "07:00 to 13:00", resulting in access granted during those hours for special days listed in the "Half-days" special day group.

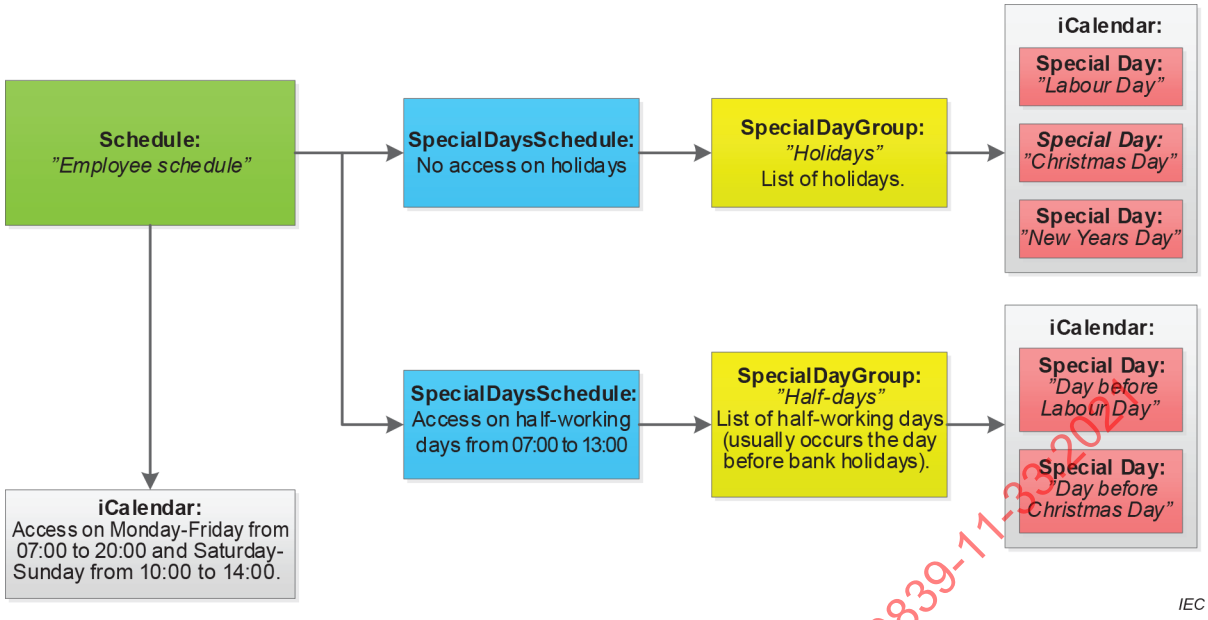


Figure 14 – SpecialDaysSchedule example

If a special day is defined as part of a day (requires ExtendedRecurrenceSupported = true), then access shall only be granted when the time period and the part of the special day overlaps.

Consider the example depicted in Figure 15. Assume a standard schedule giving access on weekdays from 06:00 to 18:00 (marked in green in Figure 15), but not on weekends. Then assume a special day starting on Friday, June 10, at 14:00 and ending on Saturday, June 11, at 10:00 (marked in yellow in Figure 15). If the special days' schedule defines a time period of 08:00 to 16:00 (marked by dashed red lines in Figure 15), then access will be granted from 06:00 to 16:00 on Friday, June 10, and from 08:00 to 10:00 on Saturday, June 11. See detailed example in 8.8.9.

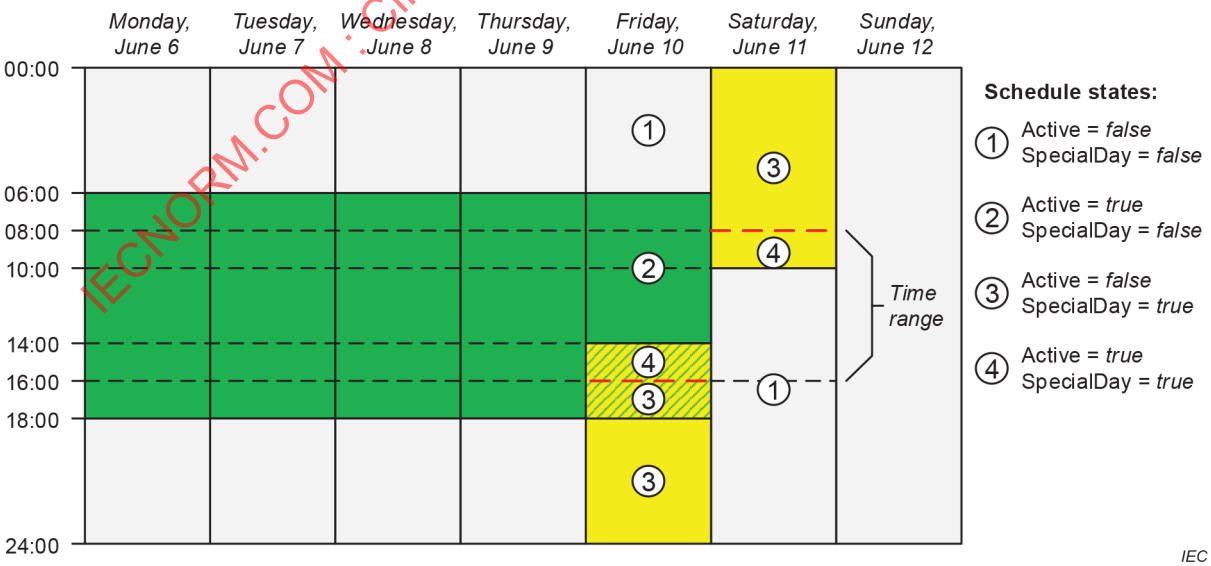


Figure 15 – Example of special day with time part

The device shall provide the following fields for each SpecialDaysSchedule instance:

- **GroupToken**
  Indicates the list of special days in a schedule.
- **TimeRange**
  Indicates the alternate time periods for the list of special days (overrides the regular schedule). For example, the regular schedule indicates that it is active from 08:00 to 17:00 on Mondays. However, this particular Monday is a special day, and the alternate time periods state that the schedule is active from 09:00 to 11:00 and 13:00 to 16:00.

  If no time periods are defined, then no access is allowed. Is of type TimePeriod.

8.4.2.4 TimePeriod

A time period defines a start and end time. For full day access, the start time = "00:00:00" with no defined end time. For a time period with no end time, the schedule runs until midnight. If an end time is specified, it shall always be greater than the start time, otherwise an InvalidArgVal fault shall be generated by the device.

The device shall provide the following fields for each TimePeriod instance:

- **From**
  Indicates the start time.

To provide more information, the device may include the following optional field:

- **Until**
  Indicates the end time. Is optional, if omitted, the period ends at midnight. The end time is exclusive, meaning that that exact moment in time is not part of the period. To determine if a moment in time (t) is part of a time period, the formula StartTime ≤ t < EndTime is used.

8.4.3 GetScheduleInfo command

This operation requests a list of ScheduleInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 51.

Table 51 – GetScheduleInfo command

GetScheduleInfo	Access class: READ_SYSTEM
Message name	**Description**
GetScheduleInfoRequest	*This message contains:* • *"Token": Tokens of ScheduleInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetScheduleInfoResponse	*This message contains:* • *"ScheduleInfo": List of ScheduleInfo items.* tsc:ScheduleInfo **ScheduleInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

8.4.4 GetScheduleInfoList command

This operation requests a list of all ScheduleInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 52.

Table 52 – GetScheduleInfoList command

GetScheduleInfoList	Access class: READ_SYSTEM
Message name	**Description**
GetScheduleInfoListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetScheduleInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"ScheduleInfo": List of ScheduleInfo items.* xs:string **NextStartReference [0][1]** tsc:ScheduleInfo **ScheduleInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

8.4.5 GetSchedules command

This operation requests a list of Schedule items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 53.

Table 53 – GetSchedules command

GetSchedules	Access class: READ_SYSTEM
Message name	**Description**
GetScheduleRequest	*This message contains:* • *"Token": Tokens of Schedule items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetScheduleResponse	*This message contains:* • *"Schedule": List of schedule items.* tsc:Schedule **Schedule [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

8.4.6 GetScheduleList command

This operation requests a list of all Schedule items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 54.

Table 54 – GetScheduleList command

GetScheduleList	Access class: READ_SYSTEM
Message name	**Description**
GetScheduleListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetScheduleListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"Schedule": List of Schedule items.* xs:string **NextStartReference [0][1]** tsc:Schedule **Schedule [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

8.4.7 CreateSchedule command

This operation creates the specified schedule in the device.

The token field of the Schedule structure shall be empty and the device shall allocate a token for the schedule. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

A device shall support this command as described in Table 55.

Table 55 – CreateSchedule command

CreateSchedule	Access class: WRITE_SYSTEM
Message name	**Description**
CreateScheduleRequest	*This message contains:* • *"Schedule": The Schedule to create.* tsc:Schedule **Schedule [1][1]**
CreateScheduleResponse	*This message contains:* • *"Token": The token of created Schedule.* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:MaxSchedules	*There is not enough space to add a new schedule, see MaxSchedules capability.*
env:Sender ter: CapabilityViolated ter:MaxSpecialDaysSchedules	*There are too many SpecialDaysSchedule entities referred in this schedule, see MaxSpecialDaysSchedules capability.*
env:Sender ter: CapabilityViolated ter:MaxTimePeriodsPerDay	*There are too many time periods in a day schedule, see MaxTimePeriodsPerDay capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

8.4.8 SetSchedule command

This method is used to synchronize a schedule in a client with the device.

If a schedule with the specified token does not exist in the device, the schedule is created. If a schedule with the specified token exists, then the schedule is modified.

A call to this method takes a Schedule structure as input parameter. The token field of the Schedule structure shall not be empty.

A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 56.

Table 56 – SetSchedule command

SetSchedule	Access class: WRITE_SYSTEM
Message name	**Description**
SetScheduleRequest	*This message contains:* • *"Schedule": The Schedule item to create or modify* tsc:Schedule **Schedule [1][1]**
SetScheduleResponse	*This message shall be empty.*
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*The device does not support that the client supplies the token.*
env:Receiver ter:CapabilityViolated ter:MaxSchedules	*There is not enough space to add a new schedule, see MaxSchedules capability.*
env:Sender ter:CapabilityViolated ter:MaxSpecialDaysSchedules	*There are too many SpecialDaysSchedule entities referred in this schedule, see MaxSpecialDaysSchedules capability.*
env:Sender ter:CapabilityViolated ter:MaxTimePeriodsPerDay	*There are too many time periods in a day schedule, see MaxTimePeriodsPerDay capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

8.4.9 ModifySchedule command

This operation modifies the specified schedule.

The token of the schedule to modify is specified in the token field of the Schedule structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified schedule.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 57.

Table 57 – ModifySchedule command

ModifySchedule	Access class: WRITE_SYSTEM
Message name	**Description**
ModifyScheduleRequest	*This message contains:* • *"Schedule": The Schedule to modify/update.* tsc:Schedule **Schedule [1][1]**
ModifyScheduleResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*The specified token is not found.*
env:Sender ter:CapabilityViolated ter:MaxSpecialDaysSchedules	*There are too many SpecialDaysSchedule entities referred in this schedule, see MaxSpecialDaysSchedules capability.*
env:Sender ter: CapabilityViolated ter:MaxTimePeriodsPerDay	*There are too many time periods in a day schedule, see MaxTimePeriodsPerDay capability.*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*A referred entity token is not found (some devices may not validate referred entities).*

8.4.10 DeleteSchedule command

This operation will delete the specified schedule.

If it is associated with one or more entities some devices may not be able to delete the schedule, and consequently a ReferenceInUse fault shall be generated.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 58.

Table 58 – DeleteSchedule command

DeleteSchedule	Access class: WRITE_SYSTEM
Message name	**Description**
DeleteScheduleRequest	*This message contains:* • *"Token": The token of the schedule to delete.* pt:ReferenceToken **Token [1][1]**
DeleteScheduleResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Schedule token is not found.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Failed to delete, Schedule token is in use.*

8.5 Special day group information

8.5.1 General

A special day group are days (or parts of days) that require the regular schedule to be overridden with an alternate schedule, for example holidays, half-days, working Sundays.

8.5.2 Data structures

8.5.2.1 SpecialDayGroupInfo

The SpecialDayGroupInfo structure contains the basic information about the special days list.

The device shall provide the following fields for each SpecialDayGroupInfo instance:

- **token**
  A service-unique identifier of the special day group.
- **Name**
  User readable name. It shall be up to 64 characters.

To provide more information, the device may include the following optional field:

- **Description**
  User readable description for the special days. It shall be up to 1 024 characters.

8.5.2.2 SpecialDayGroup

The special day group structure shall include all properties of the SpecialDayGroupInfo structure and also a set of special days. A special day group are days (or parts of days) that require the regular schedule to be overridden with an alternate schedule, for example holidays, half-days, working Sundays.

The device shall provide the following field for each SpecialDayGroup instance:

- **Days**
  An iCalendar structure that contains a group of special days. Is of type string (containing an iCalendar structure).

8.5.3 GetSpecialDayGroupInfo command

This operation requests a list of SpecialDayGroupInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device that signals support for the SpecialDaysSupported capability shall support this method as described in Table 59.

Table 59 – GetSpecialDayGroupInfo command

GetSpecialDayGroupInfo	Access class: READ_SYSTEM
Message name	**Description**
GetSpecialDayGroupInfoRequest	*This message contains:* • *"Token": Tokens of SpecialDayGroupInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetSpecialDayGroupInfoResponse	*This message contains:* • *"SpecialDayGroupInfo": List of SpecialDayGroupInfo items.* tsc:SpecialDayGroupInfo **SpecialDayGroupInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

8.5.4 GetSpecialDayGroupInfoList command

This operation requests a list of all SpecialDayGroupInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device that signals support for the SpecialDaysSupported capability shall support this method as described in Table 60.

Table 60 – GetSpecialDayGroupInfoList command

GetSpecialDayGroupInfoList	Access class: READ_SYSTEM
Message name	**Description**
GetSpecialDayGroupInfoListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSpecialDayGroupInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"SpecialDayGroupInfo": List of SpecialDayGroupInfo items.* xs:string **NextStartReference [0][1]** tsc:SpecialDayGroupInfo **SpecialDayGroupInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

8.5.5 GetSpecialDayGroups command

This operation requests a list of SpecialDayGroup items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device that signals support for the SpecialDaysSupported capability shall support this method as described in Table 61.

Table 61 – GetSpecialDayGroups command

GetSpecialDayGroups	Access class: READ_SYSTEM
Message name	**Description**
GetSpecialDayGroupsRequest	*This message contains:* • *"Token": Tokens of the SpecialDayGroup items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetSpecialDayGroupsResponse	*This message contains:* • *"SpecialDayGroup": List of SpecialDayGroup items.* tsc:SpecialDayGroup **SpecialDayGroup [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Too many items were requested, see MaxLimit capability.*

8.5.6 GetSpecialDayGroupList command

This operation requests a list of all SpecialDayGroupList items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device that signals support for the SpecialDaysSupported capability shall support this method as described in Table 62.

Table 62 – GetSpecialDayGroupList command

GetSpecialDayGroupList	Access class: READ_SYSTEM
Message name	**Description**
GetSpecialDayGroupListRequest	*This message contains:* • *"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSpecialDayGroupListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"SpecialDayGroup": List of SpecialDayGroup items.* xs:string **NextStartReference [0][1]** tsc:**SpecialDayGroup SpecialDayGroup [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

8.5.7 CreateSpecialDayGroup command

This operation creates the specified special day group in the device.

The token field of the SpecialDayGroup structure shall be empty and the device shall allocate a token for the special day group. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

A device that signals support for the SpecialDaysSupported capability shall support this method as described in Table 63.

Table 63 – CreateSpecialDayGroup command

CreateSpecialDayGroup	Access class: WRITE_SYSTEM
Message name	**Description**
CreateSpecialDayGroupRequest	*This message contains:* • *"SpecialDayGroup": The special day group to create.* tsc:SpecialDayGroup **SpecialDayGroup [1][1]**
CreateSpecialDayGroupResponse	*This message contains:* • *"Token": The token of created special day group.* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
env:Receiver ter: CapabilityViolated ter:MaxSpecialDayGroups	*There is not enough space to add new SpecialDayGroup items, see the MaxSpecialDayGroups capability.*
env:Sender ter: CapabilityViolated ter:MaxDaysInSpecialDayGroup	*There are too many special days in a SpecialDayGroup, see MaxDaysInSpecialDayGroup capability.*

8.5.8 SetSpecialDayGroup command

This method is used to synchronize a special day group in a client with the device.

If a special day group with the specified token does not exist in the device, the special day group is created. If a special day group with the specified token exists, then the special day group is modified.

A call to this method takes a special day group structure as input parameter. The token field of the SpecialDayGroup structure shall not be empty.

A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command as described in Table 64.

Table 64 – SetSpecialDayGroup command

SetSpecialDayGroup	Access class: WRITE_SYSTEM
Message name	**Description**
SetSpecialDayGroupRequest	*This message contains:* • *"SpecialDayGroup": The SpecialDayGroup item to create or modify.* tsc:SpecialDayGroup **SpecialDayGroup [1][1]**
SetSpecialDayGroupResponse	*This message shall be empty.*
Fault codes	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*The device does not support that the client supplies the token.*
env:Receiver ter:CapabilityViolated ter:MaxSpecialDayGroups	*There is not enough space to add new SpecialDayGroup items, see the MaxSpecialDayGroups capability.*
env:Sender ter:CapabilityViolated ter:MaxDaysInSpecialDayGroup	*There are too many special days in a SpecialDayGroup, see MaxDaysInSpecialDayGroup capability.*
env:Sender ter:InvalidArgVal ter:MissingToken	*The token of the special day group item shall be specified.*

8.5.9 ModifySpecialDayGroup command

This operation modifies the specified special day group.

The token of the special day group to modify is specified in the token field of the SpecialDayGroup structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified special day group.

A device that signals support for the SpecialDaysSupported capability shall support this method as described in Table 65.

Table 65 – ModifySpecialDayGroup command

ModifySpecialDayGroup	Access class: WRITE_SYSTEM
Message name	**Description**
ModifSpecialDayGroupRequest	*This message contains:* • *"SpecialDayGroup": The SpecialDayGroup item to modify/update.* tsc:SpecialDayGroup **SpecialDayGroup [1][1]**
ModifySpecialDayGroupResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*The specified token is not found.*
env:Sender ter: CapabilityViolated ter:MaxDaysInSpecialDayGroup	*There are too many special days in a SpecialDayGroup, see MaxDaysInSpecialDayGroup capability.*

8.5.10 DeleteSpecialDayGroup command

This method deletes the specified special day group.

If it is associated with one or more schedules some devices may not be able to delete the special day group, and consequently a ReferenceInUse fault shall be generated.

A device that signals support for the SpecialDaysSupported capability shall support this method as described in Table 66.

Table 66 – DeleteSpecialDayGroup command

DeleteSpecialDayGroup	Access class: WRITE_SYSTEM
Message name	**Description**
DeleteSpecialDayGroupRequest	*This message contains:* • *"Token": The token of the SpecialDayGroup item to delete.* pt:ReferenceToken **Token [1][1]**
DeleteSpecialDayGroupResponse	*This message shall be empty.*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Special day group token is not found.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Failed to delete, Special day group token is in use.*

8.6 Schedule status

8.6.1 ScheduleState data structure

The ScheduleState contains state information for a schedule. The two fields below forms four different states that a schedule can have.

The device shall provide the following fields for each ScheduleState instance:

- **Active**
  Indicates that the current time is within the boundaries of the schedule or its special days schedules' time periods. For example, if this schedule is being used for triggering automatic recording on a video source, the Active flag will be true when the schedule-based recording is supposed to record.

To provide more information, the device may include the following optional field:

- **SpecialDay**
  Indicates that the current time is within the boundaries of its special days schedules' time periods. For example, if this schedule is being used for recording at a lower frame rate on a video source during special days, the SpecialDay flag will be true. If special days are not supported by the device, this field may be omitted and interpreted as false by the client.

Example

Consider a schedule defining a standard schedule that is active from 09:00 to 17:00 during week days. There are two different special day groups: one defining half-working days, and one defining bank holidays. The special days schedule defines a time period from 09:00 to 13:00 on half-working days.

Figure 16 depicts when the four different states occur:

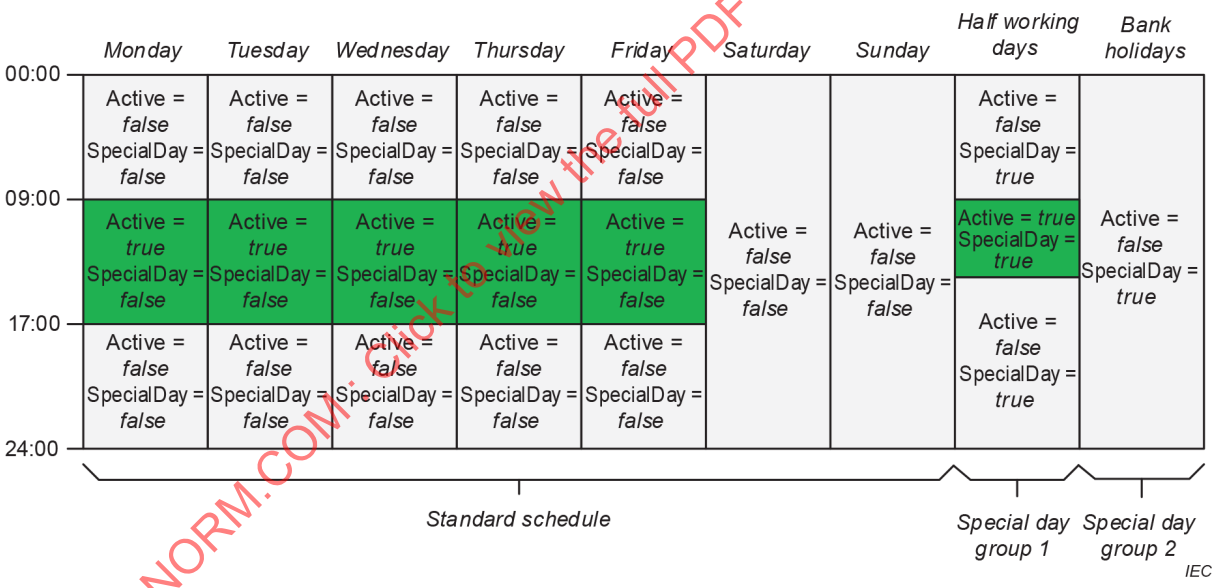


Figure 16 – Schedule states

8.6.2 GetScheduleState command

This operation requests the ScheduleState for the schedule instance specified by the given token.

A device that signals support for the StateReportingSupported capability shall support this method as described in Table 67.

Table 67 – GetScheduleState command

GetScheduleState	Access class: READ_SYSTEM_SENSITIVE
Message name	**Description**
GetScheduleStateRequest	*This message contains:* • *"Token": Token of schedule instance to get ScheduleState.* pt:ReferenceToken **Token [1][1]**
GetScheduleStateResponse	*This message contains:* • *"ScheduleState": ScheduleState item.* tsc:ScheduleState **ScheduleState [1][1]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Schedule token is not found.*

8.7 Notification topics

8.7.1 General

Subclause 8.7 defines notification topics specific to the schedule service.

8.7.2 Event overview (informative)

The schedule service specifies events when schedules or its special days are active and when schedules or special days are changed.

The main topic for status changes is:

- tns1:Schedule/State/Active

The main topics for configuration change notifications are:

- tns1:Configuration/Schedule/Changed
- tns1:Configuration/Schedule/Removed
- tns1:Configuration/SpecialDays/Changed
- tns1:Configuration/SpecialDays/Removed

8.7.3 Status changes

8.7.3.1 General

The device shall provide the status change events to inform subscribed clients when schedule entity status is changed. The device shall use the topics defined in 8.7.3 associated with the respective message description.

8.7.3.2 Schedule

If the StateReportingSupported capability is set to true then the service/device shall be capable of generating the following event whenever a schedule or its special days becomes active or inactive based on the device time. It is a property event that indicates if the schedule is active or not.

```
Topic: tns1:Schedule/State/Active
<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="ScheduleToken"
                              Type="pt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="Name"
                              Type="xs:string"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="Active"
                              Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="SpecialDay"
                              Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

8.7.4 Configuration changes

8.7.4.1 General

Whenever configuration data has been changed, added or removed a compliant device shall provide these events to inform subscribed clients.

8.7.4.2 Schedule

Whenever the configuration data for a schedule is changed (including SpecialDaysSchedule) or if a schedule is added, the device shall provide the following event:

```
Topic: tns1:Configuration/Schedule/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="ScheduleToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt: MessageDescription>
```

Whenever a schedule is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/Schedule/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="ScheduleToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

8.7.4.3 SpecialDayGroup

Whenever the configuration data for a SpecialDayGroup item is changed or added, the device shall provide the following event:

```
Topic: tns1:Configuration/SpecialDays/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SpecialDaysToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Whenever a SpecialDayGroup item is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/SpecialDays/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SpecialDaysToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

8.8 Examples

8.8.1 General

Note that all iCalendar structures in the examples below are not complete. Only the parts that are explanatory are included.

8.8.2 Access 24 × 7 for admin staff

In the following example, the values of SpecialDaysSupported and ExtendedRecurrenceSupported are irrelevant.

```
<Schedule token="402">
      <Name>Access 24*7</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access 24*7
            DTSTART:19700101T000000
            DTEND:19700102T000000
            RRULE:FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

8.8.3 Access on Monday and Wednesday from 06:00 to 20:00 for cleaning staff

In the following example, the values of SpecialDaysSupported and ExtendedRecurrenceSupported are irrelevant.

```
<Schedule token="402">
      <Name>Access on Monday and Wednesday from 06:00 to 20:00</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access on Monday and Wednesday from 06:00 to 20:00
            DTSTART:19700105T060000
            DTEND:19700105T200000
            RRULE:FREQ=WEEKLY;BYDAY=MO,WE
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

8.8.4 Access from Friday 18:00 to 07:00 for maintenance staff

This example can be realized in two different ways depending on the capabilities of the schedule service exposed by the device.

Case A

The following example is valid for ExtendedRecurrenceSupported=true. The value of SpecialDaysSupported is irrelevant.

Note that if ExtendedRecurrenceSupported=false for the example below, the device will throw a fault because the year in DTSTART is not '1970'.

```
<Schedule token="402">
      <Name>Access from Friday 18:00 to 07:00 for maintenance staff</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access from Friday 18:00 to 07:00 for maintenance staff
            DTSTART:20140523T180000
            DTEND:20140526T070000
            RRULE:FREQ=WEEKLY;BYDAY=FR
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

Case B

The following example is valid for SpecialDaysSupported=false. The value of ExtendedRecurrenceSupported is irrelevant.

```
<Schedule token="402">
      <Name>Access from Friday 18:00 to 07:00 for maintenance staff</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access from Friday 18:00 to Friday midnight
            DTSTART:19700102T180000
            DTEND:19700103T000000
            RRULE:FREQ=WEEKLY;BYDAY=FR
            END:VEVENT
            BEGIN:VEVENT
            SUMMARY:Access for all Saturdays and all Sundays
            DTSTART:19700103T000000
            DTEND:19700104T000000
            RRULE:FREQ=WEEKLY;BYDAY=SA,SU
            END:VEVENT
            BEGIN:VEVENT
            SUMMARY:Access from Sunday midnight to Monday 07:00
            DTSTART:19700105T000000
            DTEND:19700105T070000
            RRULE:FREQ=WEEKLY;BYDAY=MO
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

8.8.5 Access on weekdays from 08:00 to 17:00 for employees

In the following example, the values of SpecialDaysSupported and ExtendedRecurrenceSupported are irrelevant.

```
<Schedule token="402">
      <Name>Access on weekdays from 08:00 to 17:00 for employees</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access on weekdays from 08:00 to 17:00 for employees
            DTSTART:19700101T080000
            DTEND:19700101T170000
            RRULE:FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

8.8.6 Access from January 15, 2014, to January 14, 2015, from 09:00 to 18:00

This example can be realized in two different ways depending on the capabilities of the schedule service exposed by the device.

Case A

The following example is valid for ExtendedRecurrenceSupported=true. The value of SpecialDaysSupported is irrelevant.

```
<Schedule token="402">
      <Name>Access from Jan 15, 2014, to Jan 14, 2015, from 09:00 to 18:00</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access from Jan 15, 2014, to Jan 14, 2015, from 09:00 to 18:00
            DTSTART:20140115T090000
            DTEND:20150114T180000
            RRULE:FREQ=DAILY
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

Case B

The following example is valid for SpecialDaysSupported=false and ExtendedRecurrenceSupported=false.

Note that this scenario cannot be directly implemented in the device using only the schedule if ExtendedRecurrenceSupported capability is not supported. EACS uses start and end dates in access policies which refer to schedules. Alternately, the client or the host would need to break the schedule into a weekly schedule (whatever recurrence the device supports) and download those schedules to the device whenever the schedule changes (January 15, 2014 and January 14, 2015).

8.8.7 Special days example 1

An employee has access defined in accordance with the schedule given in the example in 8.8.5. However the site administrator wants to restrict access to employees on Christmas day (Thursday) as it is a national holiday. The site administrator has a special days entity called 'National Holidays' with December 25 added to it. The site administrator modifies the schedule to add a special days' schedule for the special days entity 'National Holidays' to override the access and disable employees from accessing the facility.

The following example is valid for SpecialDaysSupported=true. The value of ExtendedRecurrenceSupported is irrelevant.

```
<SpecialDayGroup token="8765">
      <Name>National Holidays</Name>
      <Days>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Christmas day
            DTSTART:20141225T000000
            DTEND:20141226T000000
            UID:Holiday@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Days>
<SpecialDayGroup>

<Schedule token="402">
      <Name>Access on Monday and Wednesday from 08:00 to 17:00. No access on Christmas
day</Name>
      <Standard>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Access on Monday and Wednesday from 08:00 to 17:00
            DTSTART:19700105T080000
            DTEND:19700105T170000
            RRULE:FREQ=WEEKLY;BYDAY=MO,WE
            UID:Event@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Standard>
      <SpecialDaysSchedule>
            <GroupToken>8765</GroupToken>
            <TimeRange></TimeRange>
            <!-- No time period indicates no access on referred special days -->
      </SpecialDaysSchedule>
</Schedule>
```

8.8.8 Special days example 2

A janitor has access defined in accordance with the schedule given in the example in 8.8.5, however the site administrator wants to allow access to cleaning staff on Christmas day (Thursday) as it is a national holiday. The site administrator has a special days entity called 'National Holidays' with December 25 added to it. The site administrator modifies the schedule to add a special day schedule for the special days entity 'National Holidays' to override the access and allow the janitor to access the facility.

The following example is valid for SpecialDaysSupported=true. The value of ExtendedRecurrenceSupported is irrelevant.

```
<SpecialDayGroup token="8765">
      <Name>National Holidays</Name>
      <Days>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Christmas day
            DTSTART:20141225T000000
            DTEND:20141226T000000
            UID:Holiday@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Days>
<SpecialDayGroup>

<Schedule token="402">
      <Name>Access on Monday and Wednesday from 08:00 to 17:00. No access on Christmas
day</Name>
      <Standard>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Access on Monday and Wednesday from 08:00 to 17:00
            DTSTART:19700105T080000
            DTEND:19700105T170000
            RRULE:FREQ=WEEKLY;BYDAY=MO,WE
            UID:Event@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Standard>
      <SpecialDaysSchedule>
            <GroupToken>8765</GroupToken>
            <TimeRange>
                  <From>00:00:00</From>
            </TimeRange>
      </SpecialDaysSchedule>
</Schedule>
```

8.8.9 Special days example 3

A standard schedule gives access on weekdays from 07:00 to 18:00, but not on weekends. A special day starts on Friday, June 10, at 14:00 and ends on Saturday, June 11, at 10:00. The special days' schedule defines a time range of 08:00 to 16:00.

```
<SpecialDayGroup token="7312">
      <Name>Team building days</Name>
      <Days>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Boot camp
            DTSTART:20160610T140000
            DTEND:20160611T100000
            UID:TeamBuilding@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Days>
<SpecialDayGroup>

<Schedule token="245">
       <Name>Employee schedule 2016</Name>
      <Description>Access on weekdays from 07:00 to 18:00.
             Access from 08:00 to 16:00 on Team building days</Description>
      <Standard>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Access on weekdays from 07:00 to 18:00
            DTSTART:20160101T070000
            DTEND:20161231T180000
            RRULE:FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR
            UID:Workhours@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Standard>
      <SpecialDaysSchedule>
            <GroupToken>7312</GroupToken>
            <StartTime>08:00:00</StartTime>
            <Until>16:00:00</Until>
      </SpecialDaysSchedule>
</Schedule>
```

Annex A
(normative)

Access control interface XML schemata

A.1 Credential service WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:tcr="http://www.onvif.org/ver10/credential/wsdl"
    name="CredentialService"
    targetNamespace="http://www.onvif.org/ver10/credential/wsdl">
  <!-- The data types definition for the 'Credential Service' -->
  <wsdl:types>
    <xs:schema targetNamespace="http://www.onvif.org/ver10/credential/wsdl"
               xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:pt="http://www.onvif.org/ver10/pacs"
               xmlns:tcr="http://www.onvif.org/ver10/credential/wsdl"
               elementFormDefault="qualified"
               version="18.06">
      <xs:import namespace="http://www.onvif.org/ver10/pacs"
                 schemaLocation="../../pacs/types.xsd"/>
      <!--ServiceCapabilities definitions-->
      <xs:complexType name="ServiceCapabilities">
        <xs:annotation>
          <xs:documentation>
            The service capabilities reflect optional functionality of a service. The
            information is static and does not change during device operation.
          </xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:element name="SupportedIdentifierType" type="pt:Name" minOccurs="1"
                      maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>
                A list of identifier types that the device supports. Identifier types starting
                with the prefix pt: are reserved to define ONVIF specific types. For custom
                defined identifier types shall all share the "pt:&lt;Name&gt;" syntax.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Extension" type="tcr:ServiceCapabilitiesExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="MaxLimit" type="pt:PositiveInteger" use="required">
          <xs:annotation>
            <xs:documentation>
              The maximum number of entries returned by a single Get&lt;Entity&gt;List or
              Get&lt;Entity&gt; request. The device shall never return more than this number of
              entities in a single response.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="CredentialValiditySupported" type="xs:boolean" use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates that the device supports credential validity.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="CredentialAccessProfileValiditySupported" type="xs:boolean"
                      use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates that the device supports validity on the association between a
              credential and an access profile.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="ValiditySupportsTimeValue" type="xs:boolean" use="required">
          <xs:annotation>
            <xs:documentation>
```

```
        Indicates that the device supports both date and time value for validity. If set
        to false, then the time value is ignored.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="MaxCredentials" type="pt:PositiveInteger" use="required">
    <xs:annotation>
      <xs:documentation>
        The maximum number of credential supported by the device.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="MaxAccessProfilesPerCredential" type="pt:PositiveInteger"
                use="required">
    <xs:annotation>
      <xs:documentation>
        The maximum number of access profiles for a credential.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ResetAntipassbackSupported" type="xs:boolean" use="required">
    <xs:annotation>
      <xs:documentation>
        Indicates the device supports resetting of anti-passback violations and notifying
        on anti-passback violations.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ClientSuppliedTokenSupported" type="xs:boolean" default="false">
    <xs:annotation>
      <xs:documentation>
        Indicates that the client is allowed to supply the token when creating
        credentials. To enable the use of the command SetCredential, the value must be set
        to true.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="DefaultCredentialSuspensionDuration" type="xs:string"
                default="PT5M">
    <xs:annotation>
      <xs:documentation>
        The default time period that the credential will temporary be suspended (e.g. by
        using the wrong PIN a predetermined number of times). The time period is defined
        as an [ISO 8601] duration string (e.g. "PT5M").
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ServiceCapabilitiesExtension">
  <xs:sequence>
    <xs:element name="SupportedExemptionType" type="pt:Name" minOccurs="0"
                maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>
          A list of exemptions that the device supports. Supported exemptions starting
          with the prefix pt: are reserved to define ONVIF specific exemption types and
          these reserved exemption types shall all share "pt:&lt;Name&gt;" syntax.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--CredentialInfo definitions-->
<xs:complexType name="CredentialInfo">
  <xs:annotation>
    <xs:documentation>
      The CredentialInfo type represents the credential as a logical object. The structure
      contains the basic information of a specific credential instance. The device shall
      provide the following fields for each credential.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="pt:DataEntity">
      <xs:sequence>
```

```
            <xs:element name="Description" type="pt:Description" minOccurs="0">
              <xs:annotation>
                <xs:documentation>
                  User readable description for the credential. It shall be up to 1024
                  characters.
                </xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="CredentialHolderReference">
              <xs:annotation>
                <xs:documentation>
                  An external reference to a person holding this credential. The reference is
                  a username or used ID in an external system, such as a directory service.
                </xs:documentation>
              </xs:annotation>
              <xs:simpleType base="xs:string">
                <xs:annotation>
                  <xs:documentation></xs:documentation>
                </xs:annotation>
                <xs:restriction base="xs:string">
                  <xs:maxLength value="64"/>
                  <xs:minLength value="0"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element name="ValidFrom" type="xs:dateTime" minOccurs="0">
              <xs:annotation>
                <xs:documentation>
                  The start date/time validity of the credential. If the
                  ValiditySupportsTimeValue capability is set to false, then only date is
                  supported (time is ignored).
                </xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="ValidTo" type="xs:dateTime" minOccurs="0">
              <xs:annotation>
                <xs:documentation>
                  The expiration date/time validity of the credential. If the
                  ValiditySupportsTimeValue capability is set to false, then only date is
                  supported (time is ignored).
                </xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
          <xs:anyAttribute processContents="lax"/>
        </xs:extension>
      </xs:complexContent>
    </xs:complexType>
    <!-- End of definition -->
    <!--Credential definitions-->
    <xs:complexType name="Credential">
      <xs:annotation>
        <xs:documentation>
          A Credential is a physical/tangible object, a piece of knowledge, or a facet of a
          person's physical being, that enables an individual access to a given physical
          facility or computer-based information system. A credential holds one or more
          credential identifiers. To gain access one or more identifiers may be required.
        </xs:documentation>
      </xs:annotation>
      <xs:complexContent>
        <xs:extension base="tcr:CredentialInfo">
          <xs:sequence>
            <xs:element name="CredentialIdentifier" type="tcr:CredentialIdentifier"
                        minOccurs="1" maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>
                  A list of credential identifier structures. At least one credential
                  identifier is required. Maximum one credential identifier structure per type
                  is allowed.
                </xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element name="CredentialAccessProfile" type="tcr:CredentialAccessProfile"
                        minOccurs="0" maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>
                  A list of credential access profile structures.
                </xs:documentation>
```

```
        </xs:annotation>
      </xs:element>
      <xs:element name="ExtendedGrantTime" type="xs:boolean" default="false">
        <xs:annotation>
          <xs:documentation>
            A boolean indicating that the credential holder needs extra time to get
            through the door. ExtendedReleaseTime will be added to ReleaseTime, and
            ExtendedOpenTime will be added to OpenTime
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Attribute" type="pt:Attribute" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            A list of credential attributes as name value pairs. Key names starting with
            the prefix pt: are reserved to define PACS specific attributes following the
            "pt:&lt;Name&gt;" syntax.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Extension" type="tcr:CredentialExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<xs:complexType name="CredentialExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of data types definition -->
<!--CredentialIdentifier definitions-->
<xs:complexType name="CredentialIdentifier">
  <xs:annotation>
    <xs:documentation>
      A credential identifier is a card number, unique card information, PIN or biometric
      information such as fingerprint, iris, vein, face recognition, that can be validated
      in an access point.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Type" type="tcr:CredentialIdentifierType">
      <xs:annotation>
        <xs:documentation>Contains the details of the credential identifier type. Is of
          type CredentialIdentifierType.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ExemptedFromAuthentication" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          If set to true, this credential identifier is not considered for authentication.
          For example if the access point requests Card plus PIN, and the credential
          identifier of type PIN is exempted from authentication, then the access point
          will not prompt for the PIN.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Value" type="xs:hexBinary">
      <xs:annotation>
        <xs:documentation>
          The value of the identifier in hexadecimal representation.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="CredentialIdentifierType">
  <xs:annotation>
    <xs:documentation>
      Specifies the name of credential identifier type and its format for the credential
      value.
```

```
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Name" type="pt:Name">
        <xs:annotation>
          <xs:documentation>
            The name of the credential identifier type, such as pt:Card, pt:PIN, etc.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="FormatType" type="xs:string">
        <xs:annotation>
          <xs:documentation>
            Specifies the format of the credential value for the specified identifier type
            name.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
      <!-- first ONVIF then Vendor -->
    </xs:sequence>
  </xs:complexType>
  <!-- End of definition -->
  <!--CredentialAccessProfile definitions-->
  <xs:complexType name="CredentialAccessProfile">
    <xs:annotation>
      <xs:documentation>
        The association between a credential and an access profile.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="AccessProfileToken" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>
            The reference token of the associated access profile.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ValidFrom" type="xs:dateTime" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            The start date/time of the validity for the association between the credential
            and the access profile. If the ValiditySupportsTimeValue capability is set to
            false, then only date is supported (time is ignored).
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ValidTo" type="xs:dateTime" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            The end date/time of the validity for the association between the credential and
            the access profile. If the ValiditySupportsTimeValue capability is set to false,
            then only date is supported (time is ignored).
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <!-- End of definition -->
  <!--CredentialState definitions-->
  <xs:complexType name="CredentialState">
    <xs:annotation>
      <xs:documentation>
        The CredentialState structure contains information about the state of the credential
        and optionally the reason of why the credential was disabled.
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Enabled" type="xs:boolean">
        <xs:annotation>
          <xs:documentation>
            True if the credential is enabled or false if the credential is disabled.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Reason" type="pt:Name" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
```

```
          Predefined ONVIF reasons as mentioned in the section 5.4.2.7 of credential
          service specification document. For any other reason, free text can be used.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="AntipassbackState" type="tcr:AntipassbackState" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          A structure indicating the anti-passback state. This field shall be supported if
          the ResetAntipassbackSupported capability is set to true.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Extension" type="tcr:CredentialStateExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="CredentialStateExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--AntipassbackState definitions-->
<xs:complexType name="AntipassbackState">
  <xs:annotation>
    <xs:documentation>
      A structure containing anti-passback related state information.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="AntipassbackViolated" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          Indicates if anti-passback is violated for the credential.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<!-- End of definition -->
<!-- CredentialIdentifierFormatTypeInfo definitions-->
<xs:complexType name="CredentialIdentifierFormatTypeInfo">
  <xs:annotation>
    <xs:documentation>Contains information about a format type.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="FormatType" type="xs:string">
      <xs:annotation>
        <xs:documentation>
          A format type supported by the device. A list of supported format types is
          provided in [ISO 16484-5:2014-09 Annex P]. The BACnet type "CUSTOM" is not used
          in this specification. Instead device manufacturers can define their own format
          types.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Description" type="pt:Description">
      <xs:annotation>
        <xs:documentation>
          User readable description of the credential identifier format type. It shall be
          up to 1024 characters. For custom types, it is recommended to describe how the
          octet string is encoded (following the structure in column Authentication Factor
          Value Encoding of [ISO 16484-5:2014-09 Annex P]).
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Extension" type="tcr:CredentialIdentifierFormatTypeInfoExtension"
                minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="CredentialIdentifierFormatTypeInfoExtension">
  <xs:sequence>
```

```
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      <!-- first ONVIF then Vendor -->
    </xs:sequence>
  </xs:complexType>
  <!-- End of definition -->
  <!-- CredentialData definitions-->
  <xs:complexType name="CredentialData">
    <xs:annotation>
      <xs:documentation>Contains information about a format type.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Credential" type="tcr:Credential">
        <xs:annotation>
          <xs:documentation>
            A format type supported by the device. A list of supported format types is
            provided in [ISO 16484-5:2014-09 Annex P]. The BACnet type "CUSTOM" is not used
            in this specification. Instead device manufacturers can define their own format
            types.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="CredentialState" type="tcr:CredentialState">
        <xs:annotation>
          <xs:documentation>
            User readable description of the credential identifier format type. It shall be
            up to 1024 characters. For custom types, it is recommended to describe how the
            octet string is encoded (following the structure in column Authentication Factor
            Value Encoding of [ISO 16484-5:2014-09 Annex P]).
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Extension" type="tcr:CredentialDataExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="CredentialDataExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      <!-- first ONVIF then Vendor -->
    </xs:sequence>
  </xs:complexType>
  <!-- End of definition -->
  <!-- CredentialData definitions-->
  <xs:complexType name="FaultResponse">
    <xs:annotation>
      <xs:documentation>Contains information about a format type.</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>
            A format type supported by the device. A list of supported format types is
            provided in [ISO 16484-5:2014-09 Annex P]. The BACnet type "CUSTOM" is not used
            in this specification. Instead device manufacturers can define their own format
            types.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Fault" type="xs:string">
        <xs:annotation>
          <xs:documentation>
            User readable description of the credential identifier format type. It shall be
            up to 1024 characters. For custom types, it is recommended to describe how the
            octet string is encoded (following the structure in column Authentication Factor
            Value Encoding of [ISO 16484-5:2014-09 Annex P]).
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Extension" type="tcr:FaultResponseExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="FaultResponseExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      <!-- first ONVIF then Vendor -->
    </xs:sequence>
  </xs:complexType>
```

```
<!-- End of definition -->
<!-- ##################-->
<!-- Message request / response elements-->
<xs:element name="GetServiceCapabilities">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetServiceCapabilitiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Capabilities" type="tcr:ServiceCapabilities">
        <xs:annotation>
          <xs:documentation>
            The capability response message contains the requested credential service
            capabilities using a hierarchical XML capability structure.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSupportedFormatTypes">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialIdentifierTypeName" type="xs:string">
        <xs:annotation>
          <xs:documentation>Name of the credential identifier type</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSupportedFormatTypesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="FormatTypeInfo" type="tcr:CredentialIdentifierFormatTypeInfo"
                  minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Identifier format type</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Tokens of CredentialInfo items to get.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialInfo" type="tcr:CredentialInfo" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of CredentialInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialInfoList">
  <xs:complexType>
    <xs:sequence>
```

```
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialInfoListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items. If absent, no
            more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="CredentialInfo" type="tcr:CredentialInfo" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of CredentialInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentials">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Token of Credentials to get</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Credential" type="tcr:Credential" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of Credential items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
```

```
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items. If absent, no
            more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Credential" type="tcr:Credential" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of Credential items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateCredential">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Credential" type="tcr:Credential">
        <xs:annotation>
          <xs:documentation>The credential to create.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="State" type="tcr:CredentialState">
        <xs:annotation>
          <xs:documentation>The state of the credential.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateCredentialResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>The token of the created credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ModifyCredential">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Credential" type="tcr:Credential">
        <xs:annotation>
          <xs:documentation>Details of the credential.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ModifyCredentialResponse">
  <xs:complexType>
```

```
      <xs:sequence>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="SetCredential">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="CredentialData" type="tcr:CredentialData">
          <xs:annotation>
            <xs:documentation>Details of the credential.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="SetCredentialResponse">
    <xs:complexType>
      <xs:sequence>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="DeleteCredential">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Token" type="pt:ReferenceToken">
          <xs:annotation>
            <xs:documentation>The token of the credential to delete.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="DeleteCredentialResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="GetCredentialState">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Token" type="pt:ReferenceToken">
          <xs:annotation>
            <xs:documentation>Token of Credential</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="GetCredentialStateResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="State" type="tcr:CredentialState">
          <xs:annotation>
            <xs:documentation>State of the credential.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="EnableCredential">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Token" type="pt:ReferenceToken">
          <xs:annotation>
            <xs:documentation>The token of the credential</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Reason" type="pt:Name" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Reason for enabling the credential.</xs:documentation>
          </xs:annotation>
```

```
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="EnableCredentialResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="DisableCredential">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>Token of the Credential</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Reason" type="pt:Name" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Reason for disabling the credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="DisableCredentialResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ResetAntipassbackViolation">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialToken" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>Token of the Credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ResetAntipassbackViolationResponse">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialIdentifiers">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialToken" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>Token of the Credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialIdentifiersResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialIdentifier" type="tcr:CredentialIdentifier"
                  minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Identifier of the credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
```

```
<xs:element name="SetCredentialIdentifier">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialToken" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>Token of the Credential</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="CredentialIdentifier" type="tcr:CredentialIdentifier">
        <xs:annotation>
          <xs:documentation>Identifier of the credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetCredentialIdentifierResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="DeleteCredentialIdentifier">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialToken" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>Token of the Credential</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="CredentialIdentifierTypeName" type="pt:Name">
        <xs:annotation>
          <xs:documentation>Identifier type name of a credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="DeleteCredentialIdentifierResponse">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialAccessProfiles">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialToken" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>Token of the Credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetCredentialAccessProfilesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialAccessProfile" type="tcr:CredentialAccessProfile"
                  minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Access Profiles of the credential</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetCredentialAccessProfiles">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CredentialToken" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>Token of the Credential</xs:documentation>
        </xs:annotation>
```

```
          </xs:element>
          <xs:element name="CredentialAccessProfile" type="tcr:CredentialAccessProfile"
                      minOccurs="1" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>Access Profiles of the credential</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="SetCredentialAccessProfilesResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="DeleteCredentialAccessProfiles">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="CredentialToken" type="pt:ReferenceToken">
            <xs:annotation>
              <xs:documentation>Token of the Credential</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="AccessProfileToken" type="pt:ReferenceToken" minOccurs="1"
                      maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>Tokens of Access Profiles</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="DeleteCredentialAccessProfilesResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <!-- ##################-->
  </xs:schema>
</wsdl:types>
<!--Definition of 'GetServiceCapabilities' message-->
<wsdl:message name="GetServiceCapabilitiesRequest">
  <wsdl:part name="parameters" element="tcr:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
  <wsdl:part name="parameters" element="tcr:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSupportedFormatTypes' message-->
<wsdl:message name="GetSupportedFormatTypesRequest">
  <wsdl:part name="parameters" element="tcr:GetSupportedFormatTypes"/>
</wsdl:message>
<wsdl:message name="GetSupportedFormatTypesResponse">
  <wsdl:part name="parameters" element="tcr:GetSupportedFormatTypesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetCredentialInfo' message-->
<wsdl:message name="GetCredentialInfoRequest">
  <wsdl:part name="parameters" element="tcr:GetCredentialInfo"/>
</wsdl:message>
<wsdl:message name="GetCredentialInfoResponse">
  <wsdl:part name="parameters" element="tcr:GetCredentialInfoResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetCredentialInfoList' message-->
<wsdl:message name="GetCredentialInfoListRequest">
  <wsdl:part name="parameters" element="tcr:GetCredentialInfoList"/>
</wsdl:message>
<wsdl:message name="GetCredentialInfoListResponse">
  <wsdl:part name="parameters" element="tcr:GetCredentialInfoListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetCredentials' message-->
<wsdl:message name="GetCredentialsRequest">
  <wsdl:part name="parameters" element="tcr:GetCredentials"/>
```

```
</wsdl:message>
<wsdl:message name="GetCredentialsResponse">
  <wsdl:part name="parameters" element="tcr:GetCredentialsResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetCredentialList' message-->
<wsdl:message name="GetCredentialListRequest">
  <wsdl:part name="parameters" element="tcr:GetCredentialList"/>
</wsdl:message>
<wsdl:message name="GetCredentialListResponse">
  <wsdl:part name="parameters" element="tcr:GetCredentialListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'CreateCredential' message-->
<wsdl:message name="CreateCredentialRequest">
  <wsdl:part name="parameters" element="tcr:CreateCredential"/>
</wsdl:message>
<wsdl:message name="CreateCredentialResponse">
  <wsdl:part name="parameters" element="tcr:CreateCredentialResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'ModifyCredential' message-->
<wsdl:message name="ModifyCredentialRequest">
  <wsdl:part name="parameters" element="tcr:ModifyCredential"/>
</wsdl:message>
<wsdl:message name="ModifyCredentialResponse">
  <wsdl:part name="parameters" element="tcr:ModifyCredentialResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'SetCredential' message-->
<wsdl:message name="SetCredentialRequest">
  <wsdl:part name="parameters" element="tcr:SetCredential"/>
</wsdl:message>
<wsdl:message name="SetCredentialResponse">
  <wsdl:part name="parameters" element="tcr:SetCredentialResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DeleteCredential' message-->
<wsdl:message name="DeleteCredentialRequest">
  <wsdl:part name="parameters" element="tcr:DeleteCredential"/>
</wsdl:message>
<wsdl:message name="DeleteCredentialResponse">
  <wsdl:part name="parameters" element="tcr:DeleteCredentialResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetCredentialState' message-->
<wsdl:message name="GetCredentialStateRequest">
  <wsdl:part name="parameters" element="tcr:GetCredentialState"/>
</wsdl:message>
<wsdl:message name="GetCredentialStateResponse">
  <wsdl:part name="parameters" element="tcr:GetCredentialStateResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'EnableCredential' message-->
<wsdl:message name="EnableCredentialRequest">
  <wsdl:part name="parameters" element="tcr:EnableCredential"/>
</wsdl:message>
<wsdl:message name="EnableCredentialResponse">
  <wsdl:part name="parameters" element="tcr:EnableCredentialResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DisableCredential' message-->
<wsdl:message name="DisableCredentialRequest">
  <wsdl:part name="parameters" element="tcr:DisableCredential"/>
</wsdl:message>
<wsdl:message name="DisableCredentialResponse">
  <wsdl:part name="parameters" element="tcr:DisableCredentialResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'ResetAntipassbackViolation' message-->
<wsdl:message name="ResetAntipassbackViolationRequest">
  <wsdl:part name="parameters" element="tcr:ResetAntipassbackViolation"/>
</wsdl:message>
<wsdl:message name="ResetAntipassbackViolationResponse">
  <wsdl:part name="parameters" element="tcr:ResetAntipassbackViolationResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetCredentialIdentifiers' message-->
```

```
<wsdl:message name="GetCredentialIdentifiersRequest">
  <wsdl:part name="parameters" element="tcr:GetCredentialIdentifiers"/>
</wsdl:message>
<wsdl:message name="GetCredentialIdentifiersResponse">
  <wsdl:part name="parameters" element="tcr:GetCredentialIdentifiersResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'SetCredentialIdentifier' message-->
<wsdl:message name="SetCredentialIdentifierRequest">
  <wsdl:part name="parameters" element="tcr:SetCredentialIdentifier"/>
</wsdl:message>
<wsdl:message name="SetCredentialIdentifierResponse">
  <wsdl:part name="parameters" element="tcr:SetCredentialIdentifierResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DeleteCredentialIdentifier' message-->
<wsdl:message name="DeleteCredentialIdentifierRequest">
  <wsdl:part name="parameters" element="tcr:DeleteCredentialIdentifier"/>
</wsdl:message>
<wsdl:message name="DeleteCredentialIdentifierResponse">
  <wsdl:part name="parameters" element="tcr:DeleteCredentialIdentifierResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetCredentialAccessProfiles' message-->
<wsdl:message name="GetCredentialAccessProfilesRequest">
  <wsdl:part name="parameters" element="tcr:GetCredentialAccessProfiles"/>
</wsdl:message>
<wsdl:message name="GetCredentialAccessProfilesResponse">
  <wsdl:part name="parameters" element="tcr:GetCredentialAccessProfilesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'SetCredentialAccessProfiles' message-->
<wsdl:message name="SetCredentialAccessProfilesRequest">
  <wsdl:part name="parameters" element="tcr:SetCredentialAccessProfiles"/>
</wsdl:message>
<wsdl:message name="SetCredentialAccessProfilesResponse">
  <wsdl:part name="parameters" element="tcr:SetCredentialAccessProfilesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DeleteCredentialAccessProfiles' message-->
<wsdl:message name="DeleteCredentialAccessProfilesRequest">
  <wsdl:part name="parameters" element="tcr:DeleteCredentialAccessProfiles"/>
</wsdl:message>
<wsdl:message name="DeleteCredentialAccessProfilesResponse">
  <wsdl:part name="parameters" element="tcr:DeleteCredentialAccessProfilesResponse"/>
</wsdl:message>
<!--================================-->
<wsdl:portType name="CredentialPort">
  <wsdl:operation name="GetServiceCapabilities">
    <wsdl:documentation>
      This operation returns the capabilities of the credential service.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetServiceCapabilitiesRequest"/>
    <wsdl:output message="tcr:GetServiceCapabilitiesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetSupportedFormatTypes">
    <wsdl:documentation>
      This method returns all the supported format types of a specified identifier type that
      is supported by the device.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetSupportedFormatTypesRequest"/>
    <wsdl:output message="tcr:GetSupportedFormatTypesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialInfo">
    <wsdl:documentation>
      This operation requests a list of CredentialInfo items matching the given tokens.
      The device shall ignore tokens it cannot resolve and shall return an empty list if there
      are no items matching the specified tokens. The device shall not return a fault in this
      case.
      If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be
      returned.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetCredentialInfoRequest"/>
    <wsdl:output message="tcr:GetCredentialInfoResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialInfoList">
    <wsdl:documentation>
      This operation requests a list of all CredentialInfo items provided by the device.
```

```
      A call to this method shall return a StartReference when not all data is returned and
      more data is available. The reference shall be valid for retrieving the next set of
      data. Please refer to section 4.8.3 in [ONVIF Access Control Service Specification] for
      more details.
      The number of items returned shall not be greater than the Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetCredentialInfoListRequest"/>
    <wsdl:output message="tcr:GetCredentialInfoListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetCredentials">
    <wsdl:documentation>
      This operation returns the specified credential items matching the given tokens.
      The device shall ignore tokens it cannot resolve and shall return an empty list if there
      are no items matching specified tokens. The device shall not return a fault in this
      case.
      If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be
      returned.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetCredentialsRequest"/>
    <wsdl:output message="tcr:GetCredentialsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialList">
    <wsdl:documentation>
      This operation requests a list of all credential items provided by the device.
      A call to this method shall return a StartReference when not all data is returned and
      more data is available. The reference shall be valid for retrieving the next set of
      data.
      Please refer section 4.8.3 in [Access Control Service Specification] for more details.
      The number of items returned shall not be greater the Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetCredentialListRequest"/>
    <wsdl:output message="tcr:GetCredentialListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="CreateCredential">
    <wsdl:documentation>
      This operation creates a credential. A call to this method takes a credential structure
      and a credential state structure as input parameters. The credential state can be
      created in disabled or enabled state. The token field of the credential shall be empty,
      the device shall allocate a token for the credential. The allocated token shall be
      returned in the response. If the client sends any value in the token field, the device
      shall return InvalidArgVal as generic fault code.
    </wsdl:documentation>
    <wsdl:input message="tcr:CreateCredentialRequest"/>
    <wsdl:output message="tcr:CreateCredentialResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SetCredential">
    <wsdl:documentation>
      This method is used to synchronize a credential in a client with the device.
    </wsdl:documentation>
    <wsdl:input message="tcr:SetCredentialRequest"/>
    <wsdl:output message="tcr:SetCredentialResponse"/>
  </wsdl:operation>
  <wsdl:operation name="ModifyCredential">
    <wsdl:documentation>
      This operation modifies the specified credential. The token of the credential to modify
      is specified in the token field of the Credential structure and shall not be empty. All
      other fields in the structure shall overwrite the fields in the specified credential.
      When an existing credential is modified, the state is not modified explicitly. The only
      way for a client to change the state of a credential is to explicitly call the
      EnableCredential, DisableCredential or ResetAntipassback command. All existing
      credential identifiers and credential access profiles are removed and replaced with the
      specified entities.
    </wsdl:documentation>
    <wsdl:input message="tcr:ModifyCredentialRequest"/>
    <wsdl:output message="tcr:ModifyCredentialResponse"/>
  </wsdl:operation>
  <wsdl:operation name="DeleteCredential">
    <wsdl:documentation>
      This method deletes the specified credential. If it is associated with one or more
      entities some devices may not be able to delete the credential, and consequently a
      ReferenceInUse fault shall be generated.
    </wsdl:documentation>
    <wsdl:input message="tcr:DeleteCredentialRequest"/>
    <wsdl:output message="tcr:DeleteCredentialResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialState">
    <wsdl:documentation>
      This method returns the state for the specified credential. If the capability
```

```
      ResetAntipassbackSupported is set to true, then the device shall supply the
      anti-passback state in the returned credential state structure.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetCredentialStateRequest"/>
    <wsdl:output message="tcr:GetCredentialStateResponse"/>
  </wsdl:operation>
  <wsdl:operation name="EnableCredential">
    <wsdl:documentation>
      This method is used to enable a credential.
    </wsdl:documentation>
    <wsdl:input message="tcr:EnableCredentialRequest"/>
    <wsdl:output message="tcr:EnableCredentialResponse"/>
  </wsdl:operation>
  <wsdl:operation name="DisableCredential">
    <wsdl:documentation>
      This method is used to disable a credential.
    </wsdl:documentation>
    <wsdl:input message="tcr:DisableCredentialRequest"/>
    <wsdl:output message="tcr:DisableCredentialResponse"/>
  </wsdl:operation>
  <wsdl:operation name="ResetAntipassbackViolation">
    <wsdl:documentation>
      This method is used to reset anti-passback violations for a specified credential.
    </wsdl:documentation>
    <wsdl:input message="tcr:ResetAntipassbackViolationRequest"/>
    <wsdl:output message="tcr:ResetAntipassbackViolationResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialIdentifiers">
    <wsdl:documentation>
      This method returns all the credential identifiers for a credential.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetCredentialIdentifiersRequest"/>
    <wsdl:output message="tcr:GetCredentialIdentifiersResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SetCredentialIdentifier">
    <wsdl:documentation>
      This operation creates or updates a credential identifier for a credential. If the type
      of specified credential identifier already exists, the current credential identifier of
      that type is replaced. Otherwise the credential identifier is added.
    </wsdl:documentation>
    <wsdl:input message="tcr:SetCredentialIdentifierRequest"/>
    <wsdl:output message="tcr:SetCredentialIdentifierResponse"/>
  </wsdl:operation>
  <wsdl:operation name="DeleteCredentialIdentifier">
    <wsdl:documentation>
      This method deletes all the identifier values for the specified type. However, if the
      identifier type name doesn't exist in the device, it will be silently ignored without
      any response.
    </wsdl:documentation>
    <wsdl:input message="tcr:DeleteCredentialIdentifierRequest"/>
    <wsdl:output message="tcr:DeleteCredentialIdentifierResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialAccessProfiles">
    <wsdl:documentation>
      This method returns all the credential access profiles for a credential.
    </wsdl:documentation>
    <wsdl:input message="tcr:GetCredentialAccessProfilesRequest"/>
    <wsdl:output message="tcr:GetCredentialAccessProfilesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SetCredentialAccessProfiles">
    <wsdl:documentation>
      This operation add or updates the credential access profiles for a credential. The
      device shall update the credential access profile if the access profile token in the
      specified credential access profile matches. Otherwise the credential access profile is
      added.
    </wsdl:documentation>
    <wsdl:input message="tcr:SetCredentialAccessProfilesRequest"/>
    <wsdl:output message="tcr:SetCredentialAccessProfilesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="DeleteCredentialAccessProfiles">
    <wsdl:documentation>
      This method deletes all the credential access profiles for the specified tokens.
      However, if no matching credential access profiles are found, the corresponding access
      profile tokens are silently ignored without any response.
    </wsdl:documentation>
    <wsdl:input message="tcr:DeleteCredentialAccessProfilesRequest"/>
    <wsdl:output message="tcr:DeleteCredentialAccessProfilesResponse"/>
  </wsdl:operation>
```

```
</wsdl:portType>
<!--Protocol & data format for the operations and messages for the port type
'credentialPort'-->
<wsdl:binding name="CredentialBinding" type="tcr:CredentialPort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetServiceCapabilities">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/GetServiceCapabilities"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetSupportedFormatTypes">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/GetSupportedFormatTypes"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialInfo">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/GetCredentialInfo"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialInfoList">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/GetCredentialInfoList"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetCredentials">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/GetCredentials"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialList">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/GetCredentialList"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CreateCredential">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/CreateCredential"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SetCredential">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/SetCredential"/>
```

```
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ModifyCredential">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/ModifyCredential"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DeleteCredential">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/DeleteCredential"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialState">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/GetCredentialState"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="EnableCredential">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/EnableCredential"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DisableCredential">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/DisableCredential"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ResetAntipassbackViolation">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/ResetAntipassbackViolation"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetCredentialIdentifiers">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/credential/wsdl/GetCredentialIdentifiers"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SetCredentialIdentifier">
```

```
      <soap:operation
          soapAction="http://www.onvif.org/ver10/credential/wsdl/SetCredentialIdentifier"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteCredentialIdentifier">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/credential/wsdl/DeleteCredentialIdentifier"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetCredentialAccessProfiles">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/credential/wsdl/GetCredentialAccessProfiles"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetCredentialAccessProfiles">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/credential/wsdl/SetCredentialAccessProfiles"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteCredentialAccessProfiles">
      <soap:operation soapAction=
          "http://www.onvif.org/ver10/credential/wsdl/DeleteCredentialAccessProfiles"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

A.2 Access rules service WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:tar="http://www.onvif.org/ver10/accessrules/wsdl"
    name="AccessRulesService"
    targetNamespace="http://www.onvif.org/ver10/accessrules/wsdl">
  <!-- The data types definition for the 'Access Rules Service' -->
  <wsdl:types>
    <xs:schema targetNamespace="http://www.onvif.org/ver10/accessrules/wsdl"
               xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:pt="http://www.onvif.org/ver10/pacs"
               xmlns:tar="http://www.onvif.org/ver10/accessrules/wsdl"
               elementFormDefault="qualified"
               version="18.06">
      <xs:import namespace="http://www.onvif.org/ver10/pacs"
                 schemaLocation="../../pacs/types.xsd"/>
      <!--ServiceCapabilities definitions-->
      <xs:complexType name="ServiceCapabilities">
        <xs:annotation>
          <xs:documentation>
```

```
            The service capabilities reflect optional functionality of a service. The
            information is static and does not change during device operation.
          </xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
          <!-- first ONVIF then Vendor -->
        </xs:sequence>
        <xs:attribute name="MaxLimit" use="required">
          <xs:annotation>
            <xs:documentation>
              The maximum number of entries returned by a single Get&lt;Entity&gt;List or
              Get&lt;Entity&gt; request. The device shall never return more than this number of
              entities in a single response.
            </xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:unsignedInt">
              <xs:minInclusive value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="MaxAccessProfiles" use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates the maximum number of access profiles supported by the device.
            </xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:unsignedInt">
              <xs:minInclusive value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="MaxAccessPoliciesPerAccessProfile" use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates the maximum number of access policies per access profile supported by
              the device.
            </xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:unsignedInt">
              <xs:minInclusive value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="MultipleSchedulesPerAccessPointSupported" type="xs:boolean"
                      use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates whether or not several access policies can refer to the same access
              point in an access profile.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="ClientSuppliedTokenSupported" type="xs:boolean" default="false">
          <xs:annotation>
            <xs:documentation>
              Indicates that the client is allowed to supply the token when creating access
              profiles. To enable the use of the command SetAccessProfile, the value shall be
set
              to true.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <!-- End of definition -->
      <!--AccessPolicy definitions-->
      <xs:complexType name="AccessPolicy">
        <xs:annotation>
          <xs:documentation>
            The access policy is an association of an access point and a schedule. It defines
            when an access point can be accessed using an access profile which contains this
            access policy. If an access profile contains several access policies specifying
            different schedules for the same access point will result in a union of the
            schedules.
```

```
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="ScheduleToken" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>
            Reference to the schedule used by the access policy.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Entity" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>
            Reference to the entity used by the rule engine, the entity type may be
            specified by the optional EntityType field explained below but is typically an
            access point.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="EntityType" type="xs:QName" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Optional entity type; if missing, an access point type as defined by the ONVIF
            Access Control Service Specification should be assumed. This can also be
            represented by the QName value "tac:AccessPoint" where tac is the namespace of
            ONVIF Access Control Service Specification. This field is provided for future
            extensions; it will allow an access policy being extended to cover entity types
            other than access points as well.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Extension" type="tar:AccessPolicyExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="AccessPolicyExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      <!-- first ONVIF then Vendor -->
    </xs:sequence>
  </xs:complexType>
  <!-- End of definition -->
  <!--AccessProfileInfo definitions-->
  <xs:complexType name="AccessProfileInfo">
    <xs:annotation>
      <xs:documentation>
        The AccessProfileInfo structure contains basic information about an access profile.
        The device shall provide the following fields for each access profile instance.
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="pt:DataEntity">
        <xs:sequence>
          <xs:element name="Name" type="pt:Name">
            <xs:annotation>
              <xs:documentation>
                A user readable name. It shall be up to 64 characters.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Description" type="pt:Description" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                User readable description for the access profile. It shall be up to 1024
                characters.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!-- End of definition -->
  <!--AccessProfileInfo definitions-->
  <xs:complexType name="AccessProfile">
    <xs:annotation>
      <xs:documentation>
        The access profile structure contains information about the collection of access
```

```
      policies. The device shall include all properties of the AccessProfileInfo structure
      and also a list of access policies.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="tar:AccessProfileInfo">
      <xs:sequence>
        <xs:element name="AccessPolicy" type="tar:AccessPolicy" minOccurs="0"
                    maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>
              A list of access policy structures, where each access policy defines during
              which schedule an access point can be accessed.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Extension" type="tar:AccessProfileExtension" minOccurs="0"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AccessProfileExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of data types definition -->
<!-- ##################-->
<!-- Message request / response elements-->
<xs:element name="GetServiceCapabilities">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetServiceCapabilitiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Capabilities" type="tar:ServiceCapabilities">
        <xs:annotation>
          <xs:documentation>
            The capability response message contains the requested access rules service
            capabilities using a hierarchical XML capability structure.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAccessProfileInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Tokens of AccessProfileInfo items to get.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAccessProfileInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AccessProfileInfo" type="tar:AccessProfileInfo" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of AccessProfileInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
```

```
<xs:element name="GetAccessProfileInfoList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAccessProfileInfoListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items. If absent, no
            more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="AccessProfileInfo" type="tar:AccessProfileInfo" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of AccessProfileInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAccessProfiles">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Tokens of AccessProfile items to get.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAccessProfilesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AccessProfile" type="tar:AccessProfile" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of Access Profile items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAccessProfileList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
```

```
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAccessProfileListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items. If absent, no
            more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="AccessProfile" type="tar:AccessProfile" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of Access Profile items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateAccessProfile">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AccessProfile" type="tar:AccessProfile">
        <xs:annotation>
          <xs:documentation>The AccessProfile to create.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateAccessProfileResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>The Token of created AccessProfile.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ModifyAccessProfile">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AccessProfile" type="tar:AccessProfile">
        <xs:annotation>
          <xs:documentation>The details of Access Profile</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ModifyAccessProfileResponse">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
```

```
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="SetAccessProfile">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="AccessProfile" type="tar:AccessProfile">
            <xs:annotation>
              <xs:documentation>The AccessProfile item to create or modify</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="SetAccessProfileResponse">
      <xs:complexType>
        <xs:sequence>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="DeleteAccessProfile">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Token" type="pt:ReferenceToken">
            <xs:annotation>
              <xs:documentation>The token of the access profile to delete.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="DeleteAccessProfileResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <!-- ##################-->
  </xs:schema>
</wsdl:types>
<!--Definition of 'GetServiceCapabilities' message-->
<wsdl:message name="GetServiceCapabilitiesRequest">
  <wsdl:part name="parameters" element="tar:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
  <wsdl:part name="parameters" element="tar:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetAccessProfileInfo' message-->
<wsdl:message name="GetAccessProfileInfoRequest">
  <wsdl:part name="parameters" element="tar:GetAccessProfileInfo"/>
</wsdl:message>
<wsdl:message name="GetAccessProfileInfoResponse">
  <wsdl:part name="parameters" element="tar:GetAccessProfileInfoResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetAccessProfileInfoList' message-->
<wsdl:message name="GetAccessProfileInfoListRequest">
  <wsdl:part name="parameters" element="tar:GetAccessProfileInfoList"/>
</wsdl:message>
<wsdl:message name="GetAccessProfileInfoListResponse">
  <wsdl:part name="parameters" element="tar:GetAccessProfileInfoListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetAccessProfiles' message-->
<wsdl:message name="GetAccessProfilesRequest">
  <wsdl:part name="parameters" element="tar:GetAccessProfiles"/>
</wsdl:message>
<wsdl:message name="GetAccessProfilesResponse">
  <wsdl:part name="parameters" element="tar:GetAccessProfilesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetAccessProfileList' message-->
<wsdl:message name="GetAccessProfileListRequest">
  <wsdl:part name="parameters" element="tar:GetAccessProfileList"/>
</wsdl:message>
```

```
<wsdl:message name="GetAccessProfileListResponse">
  <wsdl:part name="parameters" element="tar:GetAccessProfileListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'CreateAccessProfile' message-->
<wsdl:message name="CreateAccessProfileRequest">
  <wsdl:part name="parameters" element="tar:CreateAccessProfile"/>
</wsdl:message>
<wsdl:message name="CreateAccessProfileResponse">
  <wsdl:part name="parameters" element="tar:CreateAccessProfileResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'ModifyAccessProfile' message-->
<wsdl:message name="ModifyAccessProfileRequest">
  <wsdl:part name="parameters" element="tar:ModifyAccessProfile"/>
</wsdl:message>
<wsdl:message name="ModifyAccessProfileResponse">
  <wsdl:part name="parameters" element="tar:ModifyAccessProfileResponse"/>
</wsdl:message>
<!--Definition of 'SetAccessProfile' message-->
<wsdl:message name="SetAccessProfileRequest">
  <wsdl:part name="parameters" element="tar:SetAccessProfile"/>
</wsdl:message>
<wsdl:message name="SetAccessProfileResponse">
  <wsdl:part name="parameters" element="tar:SetAccessProfileResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DeleteAccessProfile' message-->
<wsdl:message name="DeleteAccessProfileRequest">
  <wsdl:part name="parameters" element="tar:DeleteAccessProfile"/>
</wsdl:message>
<wsdl:message name="DeleteAccessProfileResponse">
  <wsdl:part name="parameters" element="tar:DeleteAccessProfileResponse"/>
</wsdl:message>
<!--================================-->
<wsdl:portType name="AccessRulesPort">
  <wsdl:operation name="GetServiceCapabilities">
    <wsdl:documentation>
      This operation returns the capabilities of the access rules service.
    </wsdl:documentation>
    <wsdl:input message="tar:GetServiceCapabilitiesRequest"/>
    <wsdl:output message="tar:GetServiceCapabilitiesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetAccessProfileInfo">
    <wsdl:documentation>
      This operation requests a list of AccessProfileInfo items matching the given tokens. The
      device shall ignore tokens it cannot resolve and shall return an empty list if there are
      no items matching the specified tokens. The device shall not return a fault in this
      case. If the number of requested items is greater than MaxLimit, a TooManyItems fault
      shall be returned.
    </wsdl:documentation>
    <wsdl:input message="tar:GetAccessProfileInfoRequest"/>
    <wsdl:output message="tar:GetAccessProfileInfoResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetAccessProfileInfoList">
    <wsdl:documentation>
      This operation requests a list of all of AccessProfileInfo items provided by the device.
      A call to this method shall return a StartReference when not all data is returned and
      more data is available. The reference shall be valid for retrieving the next set of
      data. The number of items returned shall not be greater than the Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tar:GetAccessProfileInfoListRequest"/>
    <wsdl:output message="tar:GetAccessProfileInfoListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetAccessProfiles">
    <wsdl:documentation>
      This operation returns the specified access profile item matching the given tokens.
      The device shall ignore tokens it cannot resolve and shall return an empty list if there
      are no items matching specified tokens. The device shall not return a fault in this
      case. If the number of requested items is greater than MaxLimit, a TooManyItems fault
      shall be returned.
    </wsdl:documentation>
    <wsdl:input message="tar:GetAccessProfilesRequest"/>
    <wsdl:output message="tar:GetAccessProfilesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetAccessProfileList">
    <wsdl:documentation>
      This operation requests a list of all of access profile items provided by the device.
```

```
        A call to this method shall return a StartReference when not all data is returned and
        more data is available. The reference shall be valid for retrieving the next set of
        data. The number of items returned shall not be greater than the Limit parameter.
      </wsdl:documentation>
      <wsdl:input message="tar:GetAccessProfileListRequest"/>
      <wsdl:output message="tar:GetAccessProfileListResponse"/>
    </wsdl:operation>
    <wsdl:operation name="CreateAccessProfile">
      <wsdl:documentation>
        This operation creates the specified access profile in the device. The token field of
        the access profile shall be empty, the service shall allocate a token for the access
        profile. The allocated token shall be returned in the response. If the client sends any
        value in the token field, the device shall return InvalidArgVal as generic fault code.
        In an access profile, if several access policies specifying different schedules for the
        same access point will result in a union of the schedules.
      </wsdl:documentation>
      <wsdl:input message="tar:CreateAccessProfileRequest"/>
      <wsdl:output message="tar:CreateAccessProfileResponse"/>
    </wsdl:operation>
    <wsdl:operation name="ModifyAccessProfile">
      <wsdl:documentation>
        This operation will modify the access profile for the specified access profile token.
        The token of the access profile to modify is specified in the token field of the
        AccessProile structure and shall not be empty. All other fields in the structure shall
        overwrite the fields in the specified access profile. If several access policies
        specifying different schedules for the same access point will result in a union of the
        schedules. If the device could not store the access profile information then a fault
        will be generated.
      </wsdl:documentation>
      <wsdl:input message="tar:ModifyAccessProfileRequest"/>
      <wsdl:output message="tar:ModifyAccessProfileResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetAccessProfile">
      <wsdl:documentation>
        This operation will synchronize an access profile in a client with the device. If an
        access profile with the specified token does not exist in the device, the access profile
        is created. If an access profile with the specified token exists, then the access
        profile is modified. A call to this method takes an access profile structure as input
        parameter. The token field of the access profile must not be empty.
        A device that signals support for the ClientSuppliedTokenSupported capability shall
        implement this command.
      </wsdl:documentation>
      <wsdl:input message="tar:SetAccessProfileRequest"/>
      <wsdl:output message="tar:SetAccessProfileResponse"/>
    </wsdl:operation>
    <wsdl:operation name="DeleteAccessProfile">
      <wsdl:documentation>
        This operation will delete the specified access profile. If the access profile is
        deleted, all access policies associated to the access profile will also be deleted.
        If it is associated with one or more entities some devices may not be able to delete the
        access profile, and consequently a ReferenceInUse fault shall be generated.
      </wsdl:documentation>
      <wsdl:input message="tar:DeleteAccessProfileRequest"/>
      <wsdl:output message="tar:DeleteAccessProfileResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <!--Protocol & data format for the operations and messages for the port type
  'AccessRulesPort'-->
  <wsdl:binding name="AccessRulesBinding" type="tar:AccessRulesPort">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GetServiceCapabilities">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/GetServiceCapabilities"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAccessProfileInfo">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/GetAccessProfileInfo"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
```

```
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAccessProfileInfoList">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/GetAccessProfileInfoList"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAccessProfiles">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/GetAccessProfiles"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetAccessProfileList">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/GetAccessProfileList"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="CreateAccessProfile">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/CreateAccessProfile"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ModifyAccessProfile">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/ModifyAccessProfile"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="SetAccessProfile">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/SetAccessProfile"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="DeleteAccessProfile">
      <soap:operation
          soapAction="http://www.onvif.org/ver10/accessrules/wsdl/DeleteAccessProfile"/>
      <wsdl:input>
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

A.3 Authentication behaviour service WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="AuthenticationBehaviorService"
                  targetNamespace="http://www.onvif.org/ver10/authenticationbehavior/wsdl"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:tab="http://www.onvif.org/ver10/authenticationbehavior/wsdl">
<!-- The data types definition for the 'Authentication Behavior Service' -->
<wsdl:types>
  <xs:schema targetNamespace="http://www.onvif.org/ver10/authenticationbehavior/wsdl"
             xmlns:xs="http://www.w3.org/2001/XMLSchema"
             xmlns:pt="http://www.onvif.org/ver10/pacs"
             xmlns:tab="http://www.onvif.org/ver10/authenticationbehaviour/wsdl"
             elementFormDefault="qualified"
             version="18.06">
    <xs:import namespace="http://www.onvif.org/ver10/pacs"
               schemaLocation="../../pacs/types.xsd"/>
    <!--ServiceCapabilities definitions-->
    <xs:complexType name="ServiceCapabilities">
      <xs:annotation>
        <xs:documentation>
          The service capabilities reflect optional functionality of a service. The
          information is static and does not change during device operation.
        </xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        <!-- first ONVIF then Vendor -->
      </xs:sequence>
      <xs:attribute name="MaxLimit" type="pt:PositiveInteger" use="required">
        <xs:annotation>
          <xs:documentation>
            The maximum number of entries returned by a single Get&lt;Entity&gt;List or
            Get&lt;Entity&gt; request. The device shall never return more than this number of
            entities in a single response.
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="MaxAuthenticationProfiles" type="pt:PositiveInteger" use="required">
        <xs:annotation>
          <xs:documentation>
            Indicates the maximum number of authentication profiles the device supports. The
            device shall support at least one authentication profile.
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="MaxPoliciesPerAuthenticationProfile" type="pt:PositiveInteger"
                    use="required">
        <xs:annotation>
          <xs:documentation>
            Indicates the maximum number of authentication policies per authentication profile
            supported by the device.
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="MaxSecurityLevels" type="pt:PositiveInteger" use="required">
        <xs:annotation>
          <xs:documentation>
            Indicates the maximum number of security levels the device supports.
            The device shall support at least one security level.
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="MaxRecognitionGroupsPerSecurityLevel" type="pt:PositiveInteger"
                    use="required">
        <xs:annotation>
          <xs:documentation>
            Indicates the maximum number of recognition groups per security level supported by
            the device.
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="MaxRecognitionMethodsPerRecognitionGroup" type="pt:PositiveInteger"
                    use="required">
        <xs:annotation>
```

```
      <xs:documentation>
        Indicates the maximum number of recognition methods per recognition group
        supported by the device.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ClientSuppliedTokenSupported" type="xs:boolean" default="false">
    <xs:annotation>
      <xs:documentation>
        Indicates that the client is allowed to supply the token when creating
        authentication profiles and security levels. To enable the use of the commands
        SetAuthenticationProfile and SetSecurityLevel, the value must be set to true.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<!-- End of definition -->
<!--AuthenticationProfileInfo definitions-->
<xs:complexType name="AuthenticationProfileInfo">
  <xs:annotation>
    <xs:documentation>
      The AuthenticationProfileInfo structure contains information of a specific
      authentication profile instance.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="pt:DataEntity">
      <xs:sequence>
        <xs:element name="Name" type="pt:Name">
          <xs:annotation>
            <xs:documentation>
              A user readable name. It shall be up to 64 characters.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Description" type="pt:Description" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              User readable description for the access profile. It shall be up to 1024
              characters.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!-- End of definition -->
<!--AuthenticationProfile definitions-->
<xs:complexType name="AuthenticationProfile">
  <xs:annotation>
    <xs:documentation>
      The AuthenticationProfile structure shall include all properties of the
      AuthenticationProfileInfo structure and also a default security level, an
      authentication mode, and a list of AuthenticationProfile instances.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="tab:AuthenticationProfileInfo">
      <xs:sequence>
        <xs:element name="DefaultSecurityLevelToken" type="pt:ReferenceToken">
          <xs:annotation>
            <xs:documentation>
              The default security level is used if none of the authentication policies
              has a schedule covering the time of access (or if no authentication policies
              are defined).
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="AuthenticationPolicy" type="tab:AuthenticationPolicy"
                    minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>
              Each authentication policy associates a security level with a schedule
              (during which the specified security level will be required at the access
              point).
            </xs:documentation>
```

```
          </xs:annotation>
        </xs:element>
        <xs:element name="Extension" type="tab:AuthenticationProfileExtension"
                    minOccurs="0"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="AuthenticationProfileExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--AuthenticationPolicy definitions-->
<xs:complexType name="AuthenticationPolicy">
  <xs:annotation>
    <xs:documentation>
      The authentication policy is an association of a security level and a schedule. It
      defines when a certain security level is required to grant access to a credential
      holder. Each security level is given a unique priority. If authentication policies
      have overlapping schedules, the security level with the highest priority is used.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ScheduleToken" type="pt:ReferenceToken">
      <xs:annotation>
        <xs:documentation>
          Reference to the schedule used by the authentication policy.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SecurityLevelConstraint" type="tab:SecurityLevelConstraint"
                minOccurs="1" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>
          A list of security level constraint structures defining the conditions for what
          security level to use. Minimum one security level constraint must be specified.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Extension" type="tab:AuthenticationPolicyExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="AuthenticationPolicyExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--SecurityLevelConstraint definitions-->
<xs:complexType name="SecurityLevelConstraint">
  <xs:annotation>
    <xs:documentation>
      This structure defines what security level should be active depending on the state
      of the schedule.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="ActiveRegularSchedule" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          Corresponds to the Active field in the ScheduleState structure in [ONVIF
          Schedule Service Specification].
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ActiveSpecialDaySchedule" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          Corresponds to the SpecialDay field in the ScheduleState structure in [ONVIF
          Schedule Service Specification]. This field will be ignored if the device do not
          support special days.
        </xs:documentation>
```

```
      </xs:annotation>
    </xs:element>
    <xs:element name="AuthenticationMode" type="pt:Name" minOccurs="0"
                maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>
          Defines the mode of authentication. Authentication modes starting with the
          prefix pt: are reserved to define ONVIF-specific authentication modes.
          For custom defined authentication modes, free text can be used.
          The following authentication modes are defined by ONVIF:
          pt:SingleCredential – Normal mode where only one credential holder is required
          to be granted access.
          pt:DualCredential – Two credential holders are required to be granted access
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="SecurityLevelToken" type="pt:ReferenceToken">
      <xs:annotation>
        <xs:documentation>
          Reference to the security level used by the authentication policy.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Extension" type="tab:SecurityLevelConstraintExtension"
                minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="SecurityLevelConstraintExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--RecognitionMethod definitions-->
<xs:complexType name="RecognitionMethod">
  <xs:annotation>
    <xs:documentation>
      Recognition is the action of identifying authorized users requesting access by the
      comparison of presented credential data with recorded credential data. A recognition
      method is either memorized, biometric or held within a physical credential. A
      recognition type is either a recognition method or a physical input such as a
      request-to-exit button.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="RecognitionType" type="xs:string">
      <xs:annotation>
        <xs:documentation>The requested type of recognition.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Order" type="xs:int">
      <xs:annotation>
        <xs:documentation>
          The order value defines when this recognition method will be requested in
          relation to the other recognition methods in the same security level. A lower
          number indicates that the recognition method will be requested before
          recognition methods with a higher number.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Extension" type="tab:RecognitionMethodExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="RecognitionMethodExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--RecognitionGroup definitions-->
<xs:complexType name="RecognitionGroup">
  <xs:annotation>
    <xs:documentation>
    </xs:documentation>
```

```
    </xs:annotation>
    <xs:sequence>
      <xs:element name="RecognitionMethod" type="tab:RecognitionMethod" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            A list of recognition methods to request for at the access point.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Extension" type="tab:RecognitionGroupExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="RecognitionGroupExtension">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      <!-- first ONVIF then Vendor -->
    </xs:sequence>
  </xs:complexType>
  <!-- End of definition -->
  <!--SecurityLevelInfo definitions-->
  <xs:complexType name="SecurityLevelInfo">
    <xs:annotation>
      <xs:documentation>
        The SecurityLevelInfo structure contains information of a specific security level
        instance.
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="pt:DataEntity">
        <xs:sequence>
          <xs:element name="Name" type="pt:Name">
            <xs:annotation>
              <xs:documentation>
                A user readable name. It shall be up to 64 characters.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Priority" type="xs:int">
            <xs:annotation>
              <xs:documentation>
                A higher number indicates that the security level is considered more secure
                than security levels with lower priorities. The priority is used when an
                authentication profile have overlapping schedules with different security
                levels. When an access point is accessed, the authentication policies are
                walked through in priority order (highest priority first).
                When a schedule is found covering the time of access, the associated
                security level is used and processing stops. Two security levels cannot have
                the same priority.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Description" type="pt:Description" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                User readable description for the access profile. It shall be up to 1024
                characters.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!-- End of definition -->
  <!--SecurityLevel definitions-->
  <xs:complexType name="SecurityLevel">
    <xs:annotation>
      <xs:documentation>
        The SecurityLevel structure shall include all properties of the SecurityLevelInfo
        structure and also a set of recognition groups.
        The recognition groups are used to define a logical OR between the groups.
        Each recognition group consists of one or more recognition methods.
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="tab:SecurityLevelInfo">
```

```
      <xs:sequence>
        <xs:element name="RecognitionGroup" type="tab:RecognitionGroup" minOccurs="0"
                    maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>
              The recognition groups are used to define a logical OR between the groups.
              Each recognition group consists of one or more recognition methods.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Extension" type="tab:SecurityLevelExtension" minOccurs="0"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SecurityLevelExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of data types definition -->
<!-- ##################-->
<!-- Message request / response elements-->
<xs:element name="GetServiceCapabilities">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetServiceCapabilitiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Capabilities" type="tab:ServiceCapabilities">
        <xs:annotation>
          <xs:documentation>
            The capability response message contains the requested access rules service
            capabilities using a hierarchical XML capability structure.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAuthenticationProfileInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            Tokens of AuthenticationProfileInfo items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAuthenticationProfileInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AuthenticationProfileInfo" type="tab:AuthenticationProfileInfo"
                  minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of AuthenticationProfileInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAuthenticationProfileInfoList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
```

```
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAuthenticationProfileInfoListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items. If absent, no
            more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="AuthenticationProfileInfo" type="tab:AuthenticationProfileInfo"
                  minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of AuthenticationProfileInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAuthenticationProfiles">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            Tokens of AuthenticationProfile items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAuthenticationProfilesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AuthenticationProfile" type="tab:AuthenticationProfile"
                  minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of AuthenticationProfile items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAuthenticationProfileList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
```

```
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetAuthenticationProfileListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items. If absent, no
            more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="AuthenticationProfile" type="tab:AuthenticationProfile"
                  minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of AuthenticationProfile items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateAuthenticationProfile">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AuthenticationProfile" type="tab:AuthenticationProfile">
        <xs:annotation>
          <xs:documentation>The AuthenticationProfile to create.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateAuthenticationProfileResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>The Token of created AuthenticationProfile.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetAuthenticationProfile">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="AuthenticationProfile" type="tab:AuthenticationProfile">
        <xs:annotation>
          <xs:documentation>
            The AuthenticationProfile to create or modify.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetAuthenticationProfileResponse">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
```

```
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="ModifyAuthenticationProfile">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="AuthenticationProfile" type="tab:AuthenticationProfile">
          <xs:annotation>
            <xs:documentation>The AuthenticationProfile to modify.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="ModifyAuthenticationProfileResponse">
    <xs:complexType>
      <xs:sequence>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="DeleteAuthenticationProfile">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Token" type="pt:ReferenceToken">
          <xs:annotation>
            <xs:documentation>
              The token of the AuthenticationProfile to delete.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="DeleteAuthenticationProfileResponse">
    <xs:complexType>
      <xs:sequence/>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="GetSecurityLevelInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                    maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Tokens of SecurityLevelInfo items to get.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="GetSecurityLevelInfoResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SecurityLevelInfo" type="tab:SecurityLevelInfo" minOccurs="0"
                    maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>List of SecurityLevelInfo items.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--================================-->
  <xs:element name="GetSecurityLevelInfoList">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Limit" type="xs:int" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              Maximum number of entries to return. If not specified, less than one or higher
              than what the device supports, the number of items is determined by the
              device.
            </xs:documentation>
```

```
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSecurityLevelInfoListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items. If absent, no
            more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="SecurityLevelInfo" type="tab:SecurityLevelInfo" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of SecurityLevelInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSecurityLevels">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Tokens of SecurityLevel items to get.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSecurityLevelsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SecurityLevel" type="tab:SecurityLevel" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of SecurityLevel items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSecurityLevelList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
```

```
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSecurityLevelListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items. If absent, no
            more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="SecurityLevel" type="tab:SecurityLevel" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of SecurityLevel items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateSecurityLevel">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SecurityLevel" type="tab:SecurityLevel">
        <xs:annotation>
          <xs:documentation>The SecurityLevel to create.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateSecurityLevelResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>The Token of created SecurityLevel.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetSecurityLevel">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SecurityLevel" type="tab:SecurityLevel">
        <xs:annotation>
          <xs:documentation>The SecurityLevel to create or modify.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetSecurityLevelResponse">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ModifySecurityLevel">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SecurityLevel" type="tab:SecurityLevel">
        <xs:annotation>
          <xs:documentation>The SecurityLevel to modify.</xs:documentation>
        </xs:annotation>
```

```
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="ModifySecurityLevelResponse">
      <xs:complexType>
        <xs:sequence>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="DeleteSecurityLevel">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Token" type="pt:ReferenceToken">
            <xs:annotation>
              <xs:documentation>The token of the SecurityLevel to delete.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="DeleteSecurityLevelResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <!-- ##################-->
  </xs:schema>
</wsdl:types>
<!--Definition of 'GetServiceCapabilities' message-->
<wsdl:message name="GetServiceCapabilitiesRequest">
  <wsdl:part name="parameters" element="tab:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
  <wsdl:part name="parameters" element="tab:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetAuthenticationProfileInfo' message-->
<wsdl:message name="GetAuthenticationProfileInfoRequest">
  <wsdl:part name="parameters" element="tab:GetAuthenticationProfileInfo"/>
</wsdl:message>
<wsdl:message name="GetAuthenticationProfileInfoResponse">
  <wsdl:part name="parameters" element="tab:GetAuthenticationProfileInfoResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetAuthenticationProfileInfoList' message-->
<wsdl:message name="GetAuthenticationProfileInfoListRequest">
  <wsdl:part name="parameters" element="tab:GetAuthenticationProfileInfoList"/>
</wsdl:message>
<wsdl:message name="GetAuthenticationProfileInfoListResponse">
  <wsdl:part name="parameters" element="tab:GetAuthenticationProfileInfoListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetAuthenticationProfiles' message-->
<wsdl:message name="GetAuthenticationProfilesRequest">
  <wsdl:part name="parameters" element="tab:GetAuthenticationProfiles"/>
</wsdl:message>
<wsdl:message name="GetAuthenticationProfilesResponse">
  <wsdl:part name="parameters" element="tab:GetAuthenticationProfilesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetAuthenticationProfileList' message-->
<wsdl:message name="GetAuthenticationProfileListRequest">
  <wsdl:part name="parameters" element="tab:GetAuthenticationProfileList"/>
</wsdl:message>
<wsdl:message name="GetAuthenticationProfileListResponse">
  <wsdl:part name="parameters" element="tab:GetAuthenticationProfileListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'CreateAuthenticationProfile' message-->
<wsdl:message name="CreateAuthenticationProfileRequest">
  <wsdl:part name="parameters" element="tab:CreateAuthenticationProfile"/>
</wsdl:message>
<wsdl:message name="CreateAuthenticationProfileResponse">
  <wsdl:part name="parameters" element="tab:CreateAuthenticationProfileResponse"/>
```

```
</wsdl:message>
<!--================================-->
<!--Definition of 'SetAuthenticationProfile' message-->
<wsdl:message name="SetAuthenticationProfileRequest">
  <wsdl:part name="parameters" element="tab:SetAuthenticationProfile"/>
</wsdl:message>
<wsdl:message name="SetAuthenticationProfileResponse">
  <wsdl:part name="parameters" element="tab:SetAuthenticationProfileResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'ModifyAuthenticationProfile' message-->
<wsdl:message name="ModifyAuthenticationProfileRequest">
  <wsdl:part name="parameters" element="tab:ModifyAuthenticationProfile"/>
</wsdl:message>
<wsdl:message name="ModifyAuthenticationProfileResponse">
  <wsdl:part name="parameters" element="tab:ModifyAuthenticationProfileResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DeleteAuthenticationProfile' message-->
<wsdl:message name="DeleteAuthenticationProfileRequest">
  <wsdl:part name="parameters" element="tab:DeleteAuthenticationProfile"/>
</wsdl:message>
<wsdl:message name="DeleteAuthenticationProfileResponse">
  <wsdl:part name="parameters" element="tab:DeleteAuthenticationProfileResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSecurityLevelInfo' message-->
<wsdl:message name="GetSecurityLevelInfoRequest">
  <wsdl:part name="parameters" element="tab:GetSecurityLevelInfo"/>
</wsdl:message>
<wsdl:message name="GetSecurityLevelInfoResponse">
  <wsdl:part name="parameters" element="tab:GetSecurityLevelInfoResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSecurityLevelInfoList' message-->
<wsdl:message name="GetSecurityLevelInfoListRequest">
  <wsdl:part name="parameters" element="tab:GetSecurityLevelInfoList"/>
</wsdl:message>
<wsdl:message name="GetSecurityLevelInfoListResponse">
  <wsdl:part name="parameters" element="tab:GetSecurityLevelInfoListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSecurityLevels' message-->
<wsdl:message name="GetSecurityLevelsRequest">
  <wsdl:part name="parameters" element="tab:GetSecurityLevels"/>
</wsdl:message>
<wsdl:message name="GetSecurityLevelsResponse">
  <wsdl:part name="parameters" element="tab:GetSecurityLevelsResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSecurityLevelList' message-->
<wsdl:message name="GetSecurityLevelListRequest">
  <wsdl:part name="parameters" element="tab:GetSecurityLevelList"/>
</wsdl:message>
<wsdl:message name="GetSecurityLevelListResponse">
  <wsdl:part name="parameters" element="tab:GetSecurityLevelListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'CreateSecurityLevel' message-->
<wsdl:message name="CreateSecurityLevelRequest">
  <wsdl:part name="parameters" element="tab:CreateSecurityLevel"/>
</wsdl:message>
<wsdl:message name="CreateSecurityLevelResponse">
  <wsdl:part name="parameters" element="tab:CreateSecurityLevelResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'SetSecurityLevel' message-->
<wsdl:message name="SetSecurityLevelRequest">
  <wsdl:part name="parameters" element="tab:SetSecurityLevel"/>
</wsdl:message>
<wsdl:message name="SetSecurityLevelResponse">
  <wsdl:part name="parameters" element="tab:SetSecurityLevelResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'ModifySecurityLevel' message-->
<wsdl:message name="ModifySecurityLevelRequest">
  <wsdl:part name="parameters" element="tab:ModifySecurityLevel"/>
</wsdl:message>
```

```
<wsdl:message name="ModifySecurityLevelResponse">
  <wsdl:part name="parameters" element="tab:ModifySecurityLevelResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DeleteSecurityLevel' message-->
<wsdl:message name="DeleteSecurityLevelRequest">
  <wsdl:part name="parameters" element="tab:DeleteSecurityLevel"/>
</wsdl:message>
<wsdl:message name="DeleteSecurityLevelResponse">
  <wsdl:part name="parameters" element="tab:DeleteSecurityLevelResponse"/>
</wsdl:message>
<!--================================-->
<wsdl:portType name="AuthenticationBehaviorPort">
  <wsdl:operation name="GetServiceCapabilities">
    <wsdl:documentation>
      This operation returns the capabilities of the authentication behavior service.
    </wsdl:documentation>
    <wsdl:input message="tab:GetServiceCapabilitiesRequest"/>
    <wsdl:output message="tab:GetServiceCapabilitiesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationProfileInfo">
    <wsdl:documentation>
      This operation requests a list of AuthenticationProfileInfo items matching the given
      tokens. The device shall ignore tokens it cannot resolve and shall return an empty list
      if there are no items matching the specified tokens. The device shall not return a fault
      in this case.
    </wsdl:documentation>
    <wsdl:input message="tab:GetAuthenticationProfileInfoRequest"/>
    <wsdl:output message="tab:GetAuthenticationProfileInfoResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationProfileInfoList">
    <wsdl:documentation>
      This operation requests a list of all of AuthenticationProfileInfo items provided by the
      device. A call to this method shall return a StartReference when not all data is
      returned and more data is available. The reference shall be valid for retrieving the
      next set of data.
      Please refer Access Control Service Specification for more details.
      The number of items returned shall not be greater than Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tab:GetAuthenticationProfileInfoListRequest"/>
    <wsdl:output message="tab:GetAuthenticationProfileInfoListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationProfiles">
    <wsdl:documentation>
      This operation returns the specified AuthenticationProfile item matching the given
      tokens. The device shall ignore tokens it cannot resolve and shall return an empty list
      if there are no items matching specified tokens. The device shall not return a fault in
      this case.
    </wsdl:documentation>
    <wsdl:input message="tab:GetAuthenticationProfilesRequest"/>
    <wsdl:output message="tab:GetAuthenticationProfilesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationProfileList">
    <wsdl:documentation>
      This operation requests a list of all of AuthenticationProfile items provided by the
      device. A call to this method shall return a StartReference when not all data is
      returned and more data is available. The reference shall be valid for retrieving the
      next set of data.
      Please refer Access Control Service Specification for more details.
      The number of items returned shall not be greater the Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tab:GetAuthenticationProfileListRequest"/>
    <wsdl:output message="tab:GetAuthenticationProfileListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="CreateAuthenticationProfile">
    <wsdl:documentation>
      This operation creates the specified authentication profile in the device. The token
      field of the AuthenticationProfile structure shall be empty and the device shall
      allocate a token for the authentication profile. The allocated token shall be returned
      in the response. If the client sends any value in the token field, the device shall
      return InvalidArgVal as a generic fault code.
    </wsdl:documentation>
    <wsdl:input message="tab:CreateAuthenticationProfileRequest"/>
    <wsdl:output message="tab:CreateAuthenticationProfileResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SetAuthenticationProfile">
    <wsdl:documentation>
      This method is used to synchronize an authentication profile in a client with the
```

```
      device. If an authentication profile with the specified token does not exist in the
      device, the authentication profile is created. If an authentication profile with the
      specified token exists, then the authentication profile is modified.
      A call to this method takes an AuthenticationProfile structure as input parameter.
      The token field of the AuthenticationProfile shall not be empty.
      A device that signals support for the ClientSuppliedTokenSupported capability
      shall implement this command.
    </wsdl:documentation>
    <wsdl:input message="tab:SetAuthenticationProfileRequest"/>
    <wsdl:output message="tab:SetAuthenticationProfileResponse"/>
  </wsdl:operation>
  <wsdl:operation name="ModifyAuthenticationProfile">
    <wsdl:documentation>
      This operation modifies the specified authentication profile. The token of the
      authentication profile to modify is specified in the token field of the
      AuthenticationProfile structure and shall not be empty. All other fields in the
      structure shall overwrite the fields in the specified authentication profile.
    </wsdl:documentation>
    <wsdl:input message="tab:ModifyAuthenticationProfileRequest"/>
    <wsdl:output message="tab:ModifyAuthenticationProfileResponse"/>
  </wsdl:operation>
  <wsdl:operation name="DeleteAuthenticationProfile">
    <wsdl:documentation>
      This operation deletes the specified authentication profile. If the authentication
      profile is deleted, all authentication policies associated with the authentication
      profile will also be deleted. If it is associated with one or more entities some devices
      may not be able to delete the authentication profile, and consequently a ReferenceInUse
      fault shall be generated.
    </wsdl:documentation>
    <wsdl:input message="tab:DeleteAuthenticationProfileRequest"/>
    <wsdl:output message="tab:DeleteAuthenticationProfileResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetSecurityLevelInfo">
    <wsdl:documentation>
      This operation requests a list of SecurityLevelInfo items matching the given tokens.
      The device shall ignore tokens it cannot resolve and shall return an empty list if there
      are no items matching the specified tokens. The device shall not return a fault in this
      case.
    </wsdl:documentation>
    <wsdl:input message="tab:GetSecurityLevelInfoRequest"/>
    <wsdl:output message="tab:GetSecurityLevelInfoResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetSecurityLevelInfoList">
    <wsdl:documentation>
      This operation requests a list of all of SecurityLevelInfo items provided by the device.
      A call to this method shall return a StartReference when not all data is returned and
      more data is available. The reference shall be valid for retrieving the next set of
      data.
      Please refer Access Control Service Specification for more details.
      The number of items returned shall not be greater than Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tab:GetSecurityLevelInfoListRequest"/>
    <wsdl:output message="tab:GetSecurityLevelInfoListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetSecurityLevels">
    <wsdl:documentation>
      This operation returns the specified SecurityLevel item matching the given tokens. The
      device shall ignore tokens it cannot resolve and shall return an empty list if there are
      no items matching specified tokens. The device shall not return a fault in this case.
    </wsdl:documentation>
    <wsdl:input message="tab:GetSecurityLevelsRequest"/>
    <wsdl:output message="tab:GetSecurityLevelsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetSecurityLevelList">
    <wsdl:documentation>
      This operation requests a list of all of SecurityLevel items provided by the device.
      A call to this method shall return a StartReference when not all data is returned and
      more data is available. The reference shall be valid for retrieving the next set of
      data.
      Please refer Access Control Service Specification for more details.
      The number of items returned shall not be greater the Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tab:GetSecurityLevelListRequest"/>
    <wsdl:output message="tab:GetSecurityLevelListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="CreateSecurityLevel">
    <wsdl:documentation>
      This operation creates the specified security level in the device. The token field of
```

```
      the SecurityLevel structure shall be empty and the device shall allocate a token for the
      security level. The allocated token shall be returned in the response. If the client
      sends any value in the token field, the device shall return InvalidArgVal as a generic
      fault code.
    </wsdl:documentation>
    <wsdl:input message="tab:CreateSecurityLevelRequest"/>
    <wsdl:output message="tab:CreateSecurityLevelResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SetSecurityLevel">
    <wsdl:documentation>
      This method is used to synchronize an security level in a client with the device.
      If an security level with the specified token does not exist in the device, the security
      level is created. If an security level with the specified token exists, then the
      security level is modified. A call to this method takes an SecurityLevel structure as
      input parameter. The token field of the SecurityLevel shall not be empty.
      A device that signals support for the ClientSuppliedTokenSupported capability shall
      implement this command.
    </wsdl:documentation>
    <wsdl:input message="tab:SetSecurityLevelRequest"/>
    <wsdl:output message="tab:SetSecurityLevelResponse"/>
  </wsdl:operation>
  <wsdl:operation name="ModifySecurityLevel">
    <wsdl:documentation>
      This operation modifies the specified security level. The token of the security level to
      modify is specified in the token field of the SecurityLevel structure and shall not be
      empty. All other fields in the structure shall overwrite the fields in the specified
      security level.
    </wsdl:documentation>
    <wsdl:input message="tab:ModifySecurityLevelRequest"/>
    <wsdl:output message="tab:ModifySecurityLevelResponse"/>
  </wsdl:operation>
  <wsdl:operation name="DeleteSecurityLevel">
    <wsdl:documentation>
      This operation deletes the specified security level. If the security level is deleted,
      all authentication policies associated with the security level will also be deleted.
      If it is associated with one or more entities some devices may not be able to delete the
      security level, and consequently a ReferenceInUse fault shall be generated.
    </wsdl:documentation>
    <wsdl:input message="tab:DeleteSecurityLevelRequest"/>
    <wsdl:output message="tab:DeleteSecurityLevelResponse"/>
  </wsdl:operation>
</wsdl:portType>
<!--Protocol & data format for the operations and messages for the port type
'AuthenticationBehaviorPort'-->
<wsdl:binding name="AuthenticationBehaviorBinding" type="tab:AuthenticationBehaviorPort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetServiceCapabilities">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetServiceCapabilities"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationProfileInfo">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetAuthenticationProfileInfo"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationProfileInfoList">
    <soap:operation soapAction=
     "http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetAuthenticationProfileInfoList"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationProfiles">
    <soap:operation soapAction=
```

```
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetAuthenticationProfiles"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetAuthenticationProfileList">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetAuthenticationProfileList"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CreateAuthenticationProfile">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/CreateAuthenticationProfile"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SetAuthenticationProfile">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/SetAuthenticationProfile"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ModifyAuthenticationProfile">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/ModifyAuthenticationProfile"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DeleteAuthenticationProfile">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/DeleteAuthenticationProfile"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetSecurityLevelInfo">
    <soap:operation

soapAction="http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetSecurityLevelInfo"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetSecurityLevelInfoList">
    <soap:operation

soapAction="http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetSecurityLevelInfoList"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
```

```
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetSecurityLevels">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetSecurityLevels"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetSecurityLevelList">
    <soap:operation

soapAction="http://www.onvif.org/ver10/authenticationbehavior/wsdl/GetSecurityLevelList"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="CreateSecurityLevel">
    <soap:operation

soapAction="http://www.onvif.org/ver10/authenticationbehavior/wsdl/CreateSecurityLevel"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="SetSecurityLevel">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/authenticationbehavior/wsdl/SetSecurityLevel"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="ModifySecurityLevel">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/ModifySecurityLevel"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="DeleteSecurityLevel">
    <soap:operation soapAction=
        "http://www.onvif.org/ver10/authenticationbehavior/wsdl/DeleteSecurityLevel"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
</wsdl:definitions>
```

A.4 Schedule service WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:tsc="http://www.onvif.org/ver10/schedule/wsdl"
    name="ScheduleService"
    targetNamespace="http://www.onvif.org/ver10/schedule/wsdl">
  <!-- The data types definition for the 'Schedule Service' -->
  <wsdl:types>
    <xs:schema targetNamespace="http://www.onvif.org/ver10/schedule/wsdl"
               xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:pt="http://www.onvif.org/ver10/pacs"
               xmlns:tsc="http://www.onvif.org/ver10/schedule/wsdl"
               elementFormDefault="qualified"
               version="18.06">
      <xs:import namespace="http://www.onvif.org/ver10/pacs"
                 schemaLocation="../../pacs/types.xsd"/>
      <!--ServiceCapabilities definitions-->
      <xs:complexType name="ServiceCapabilities">
        <xs:annotation>
          <xs:documentation>
            The service capabilities reflect optional functionality of a service.
            The information is static and does not change during device operation.
          </xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
          <!-- first ONVIF then Vendor -->
        </xs:sequence>
        <xs:attribute name="MaxLimit" type="pt:PositiveInteger" use="required">
          <xs:annotation>
            <xs:documentation>
              The maximum number of entries returned by a single Get&lt;Entity&gt;List or
              Get&lt;Entity&gt; request. The device shall never return more than this number of
              entities in a single response.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="MaxSchedules" type="pt:PositiveInteger" use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates the maximum number of schedules the device supports.
              The device shall support at least one schedule.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="MaxTimePeriodsPerDay" type="pt:PositiveInteger" use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates the maximum number of time periods per day the device supports in a
              schedule including special days schedule. The device shall support at least one
              time period per day.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="MaxSpecialDayGroups" type="pt:PositiveInteger" use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates the maximum number of special day group entities the device supports.
              The device shall support at least one 'SpecialDayGroup' entity.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="MaxDaysInSpecialDayGroup" type="pt:PositiveInteger" use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates the maximum number of days per 'SpecialDayGroup' entity the device
              supports. The device shall support at least one day per 'SpecialDayGroup' entity.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="MaxSpecialDaysSchedules" type="pt:PositiveInteger" use="required">
          <xs:annotation>
            <xs:documentation>
```

```
        Indicates the maximum number of 'SpecialDaysSchedule' entities referred by a
        schedule that the device supports.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ExtendedRecurrenceSupported" type="xs:boolean" use="required">
    <xs:annotation>
      <xs:documentation>
        For schedules:
        If this capability is supported, then all iCalendar recurrence types shall be
        supported by the device. The device shall also support the start and end dates (or
        iCalendar occurrence count) in recurring events.
        If this capability is not supported, then only the weekly iCalendar recurrence
        type shall be supported. Non-recurring events and other recurring types are not
        supported. The device shall only accept a start date with the year '1970' (the
        month and day is needed to reflect the week day of the recurrence) and will not
        accept an occurrence count (or iCalendar until date) in recurring events.
        For special days (only applicable if SpecialDaysSupported is set to true):
        If this capability is supported, then all iCalendar recurrence types shall be
        supported by the device. The device shall also support the start and end dates (or
        occurrence count) in recurring events. If this capability is not supported, then
        only non-recurring special days are supported.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="SpecialDaysSupported" type="xs:boolean" use="required">
    <xs:annotation>
      <xs:documentation>
        If this capability is supported, then the device shall support special days.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="StateReportingSupported" type="xs:boolean" use="required">
    <xs:annotation>
      <xs:documentation>
        If this capability is set to true, the device shall implement the GetScheduleState
        command, and shall notify subscribing clients whenever schedules become active or
        inactive.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ClientSuppliedTokenSupported" type="xs:boolean" default="false">
    <xs:annotation>
      <xs:documentation>
        Indicates that the client is allowed to supply the token when creating schedules
        and special day groups. To enable the use of the commands SetSchedule and
        SetSpecialDayGroup, the value must be set to true.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<!-- End of definition -->
<!--ScheduleInfo definitions-->
<xs:complexType name="ScheduleInfo">
  <xs:annotation>
    <xs:documentation>
      The ScheduleInfo type represents the schedule as a physical object.
      The structure contains information of a specific schedule instance.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="pt:DataEntity">
      <xs:sequence>
        <xs:element name="Name" type="pt:Name">
          <xs:annotation>
            <xs:documentation>
              A user readable name. It shall be up to 64 characters.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Description" type="pt:Description" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              User readable description for the schedule. It shall be up to 1024
              characters.
            </xs:documentation>
          </xs:annotation>
```

```
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!-- End of definition -->
<!--Schedule definitions-->
<xs:complexType name="Schedule">
  <xs:annotation>
    <xs:documentation>
      The schedule structure shall include all properties of the ScheduleInfo structure
      and also the standard events (iCalendar format) and a list of SpecialDaysSchedule
      instances.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="tsc:ScheduleInfo">
      <xs:sequence>
        <xs:element name="Standard" type="xs:string">
          <xs:annotation>
            <xs:documentation>
              An iCalendar structure that defines a number of events. Events can be
              recurring or non-recurring. The events can, for instance, be used to control
              when a camera should record or when a facility is accessible.
              Some devices might not be able to fully support all the features of
              iCalendar.
              Setting the service capability ExtendedRecurrenceSupported to false will
              enable more devices to be ONVIF compliant.
              Is of type string (but contains an iCalendar structure).
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="SpecialDays" type="tsc:SpecialDaysSchedule" minOccurs="0"
                    maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>
              For devices that are not able to support all the features of iCalendar,
              supporting special days is essential. Each SpecialDaysSchedule instance
              defines an alternate set of time periods that overrides the regular schedule
              for a specified list of special days. Is of type SpecialDaysSchedule.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Extension" type="tsc:ScheduleExtension" minOccurs="0"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ScheduleExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--SpecialDaysSchedule definitions-->
<xs:complexType name="SpecialDaysSchedule">
  <xs:annotation>
    <xs:documentation>
      A override schedule that defines alternate time periods for a group of special days.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="GroupToken" type="pt:ReferenceToken">
      <xs:annotation>
        <xs:documentation>
          Indicates the list of special days in a schedule.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="TimeRange" type="tsc:TimePeriod" minOccurs="0"
                maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>
          Indicates the alternate time periods for the list of special days (overrides the
          regular schedule). For example, the regular schedule indicates that it is active
```

```
                from 08:00 to 17:00 on Mondays. However, this particular Monday is a special
day,
                and the alternate time periods state that the schedule is active from 09:00 to
11:00 and 13:00 to 16:00. If no time periods are defined, then no access is allowed.
                Is of type TimePeriod.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Extension" type="tsc:SpecialDaysScheduleExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:complexType name="SpecialDaysScheduleExtension">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
          <!-- first ONVIF then Vendor -->
        </xs:sequence>
      </xs:complexType>
      <!-- End of definition -->
      <!-- ScheduleState definitions-->
      <xs:complexType name="ScheduleState">
        <xs:annotation>
          <xs:documentation>
            The ScheduleState contains state information for a schedule.
          </xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:element name="Active" type="xs:boolean">
            <xs:annotation>
              <xs:documentation>
                Indicates that the current time is within the boundaries of the schedule or its
                special days schedules' time periods. For example, if this schedule is being
                used for triggering automatic recording of a video source, the Active flag will
                be true when the schedule-based recording is supposed to record.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="SpecialDay" type="xs:boolean" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                Indicates that the current time is within the boundaries of its special days
                schedules' time periods. For example, if this schedule is being used for
                recording at a lower frame rate on a video source during special days, the
                SpecialDay flag will be true. If special days are not supported by the device,
                this field may be omitted and interpreted as false by the client.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Extension" type="tsc:ScheduleStateExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
      </xs:complexType>
      <xs:complexType name="ScheduleStateExtension">
        <xs:sequence>
          <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
          <!-- first ONVIF then Vendor -->
        </xs:sequence>
      </xs:complexType>
      <!-- End of definition -->
      <!--TimePeriod definitions-->
      <xs:complexType name="TimePeriod">
        <xs:annotation>
          <xs:documentation>
            A time period defines a start and end time. For full day access, the start time
            ="00:00:00" with no defined end time. For a time period with no end time, the
            schedule runs until midnight. The end time must always be greater than the start
            time, otherwise an InvalidArgVal error messages is generated by the device.
          </xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:element name="From" type="xs:time">
            <xs:annotation>
              <xs:documentation>
                Indicates the start time.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Until" type="xs:time" minOccurs="0">
```

```
      <xs:annotation>
        <xs:documentation>
          Indicates the end time. Is optional, if omitted, the period ends at midnight.
          The end time is exclusive, meaning that that exact moment in time is not part of
          the period. To determine if a moment in time (t) is part of a time period, the
          formula StartTime &#8804; t &lt; EndTime is used.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Extension" type="tsc:TimePeriodExtension" minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="TimePeriodExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--SpecialDaysInfo definitions-->
<xs:complexType name="SpecialDayGroupInfo">
  <xs:annotation>
    <xs:documentation>
      The SpecialDayGroupInfo structure contains the basic information about the special
      days list.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="pt:DataEntity">
      <xs:sequence>
        <xs:element name="Name" type="pt:Name">
          <xs:annotation>
            <xs:documentation>
              User readable name. It shall be up to 64 characters.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Description" type="pt:Description" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              User readable description for the special days. It shall be up to 1024
              characters.
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<!-- End of definition -->
<!--SpecialDays definitions-->
<xs:complexType name="SpecialDayGroup">
  <xs:annotation>
    <xs:documentation>
      The special day group structure shall include all properties of the
      SpecialDayGroupInfo structure and also a set of special days. A special day group
      are days (or parts of days) that require the regular schedule to be overridden with
      an alternate schedule. For example holidays, half-days, working Sundays, etc.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="tsc:SpecialDayGroupInfo">
      <xs:sequence>
        <xs:element name="Days" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              An iCalendar structure that contains a group of special days.
              Is of type string (containing an iCalendar structure).
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="Extension" type="tsc:SpecialDayGroupExtension" minOccurs="0"/>
      </xs:sequence>
      <xs:anyAttribute processContents="lax"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="SpecialDayGroupExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!-- ##################-->
<!-- Message request / response elements-->
<xs:element name="GetServiceCapabilities">
  <xs:complexType>
    <xs:sequence/>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetServiceCapabilitiesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Capabilities" type="tsc:ServiceCapabilities">
        <xs:annotation>
          <xs:documentation>
            The capability response message contains the requested schedule service
            capabilities using a hierarchical XML capability structure.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetScheduleState">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>
            Token of schedule instance to get ScheduleState.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetScheduleStateResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ScheduleState" type="tsc:ScheduleState">
        <xs:annotation>
          <xs:documentation>
            ScheduleState item.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetScheduleInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            Tokens of ScheduleInfo items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetScheduleInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ScheduleInfo" type="tsc:ScheduleInfo" minOccurs="0"
                  maxOccurs="unbounded">
```

```
        <xs:annotation>
          <xs:documentation>
            List of ScheduleInfo items.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetScheduleInfoList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetScheduleInfoListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items.
            If absent, no more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="ScheduleInfo" type="tsc:ScheduleInfo" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            List of ScheduleInfo items.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSchedules">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            Tokens of Schedule items to get
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSchedulesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Schedule" type="tsc:Schedule" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of schedule items.</xs:documentation>
```

```
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetScheduleList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetScheduleListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items.
            If absent, no more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Schedule" type="tsc:Schedule" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of Schedule items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateSchedule">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Schedule" type="tsc:Schedule" minOccurs="1">
        <xs:annotation>
          <xs:documentation>The Schedule to create</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateScheduleResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken">
        <xs:annotation>
          <xs:documentation>The token of created Schedule</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetSchedule">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Schedule" type="tsc:Schedule" minOccurs="1">
```

```
        <xs:annotation>
          <xs:documentation>The Schedule to modify/create</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetScheduleResponse">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ModifySchedule">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Schedule" type="tsc:Schedule" minOccurs="1">
        <xs:annotation>
          <xs:documentation>The Schedule to modify/update</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="ModifyScheduleResponse">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="DeleteSchedule">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1">
        <xs:annotation>
          <xs:documentation>The token of the schedule to delete.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="DeleteScheduleResponse">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSpecialDayGroupInfo">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Tokens of SpecialDayGroupInfo items to get.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSpecialDayGroupInfoResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SpecialDayGroupInfo" type="tsc:SpecialDayGroupInfo" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of SpecialDayGroupInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<!--================================-->
<xs:element name="GetSpecialDayGroupInfoList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSpecialDayGroupInfoListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items.
            If absent, no more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="SpecialDayGroupInfo" type="tsc:SpecialDayGroupInfo" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of SpecialDayGroupInfo items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSpecialDayGroups">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Tokens of the SpecialDayGroup items to get</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSpecialDayGroupsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SpecialDayGroup" type="tsc:SpecialDayGroup" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of SpecialDayGroup items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSpecialDayGroupList">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Limit" type="xs:int" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
```

```
            Maximum number of entries to return. If not specified, less than one or higher
            than what the device supports, the number of items is determined by the
            device.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="StartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            Start returning entries from this start reference. If not specified, entries
            shall start from the beginning of the dataset.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="GetSpecialDayGroupListResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="NextStartReference" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            StartReference to use in next call to get the following items.
            If absent, no more items to get.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="SpecialDayGroup" type="tsc:SpecialDayGroup" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>List of SpecialDayGroup items.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateSpecialDayGroup">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SpecialDayGroup" type="tsc:SpecialDayGroup" minOccurs="1">
        <xs:annotation>
          <xs:documentation>
            The special day group to create.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="CreateSpecialDayGroupResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Token" type="pt:ReferenceToken" minOccurs="1">
        <xs:annotation>
          <xs:documentation>The token of created special day group.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetSpecialDayGroup">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SpecialDayGroup" type="tsc:SpecialDayGroup" minOccurs="1">
        <xs:annotation>
          <xs:documentation>The SpecialDayGroup to modify/create</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--================================-->
<xs:element name="SetSpecialDayGroupResponse">
```

```
      <xs:complexType>
        <xs:sequence>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="ModifySpecialDayGroup">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="SpecialDayGroup" type="tsc:SpecialDayGroup" minOccurs="1">
            <xs:annotation>
              <xs:documentation>The special day group to modify/update.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="ModifySpecialDayGroupResponse">
      <xs:complexType>
        <xs:sequence>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="DeleteSpecialDayGroup">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Token" type="pt:ReferenceToken">
            <xs:annotation>
              <xs:documentation>
                The token of the special day group item to delete.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <xs:element name="DeleteSpecialDayGroupResponse">
      <xs:complexType>
        <xs:sequence/>
      </xs:complexType>
    </xs:element>
    <!--================================-->
    <!-- ##################-->
  </xs:schema>
</wsdl:types>
<!--Definition of 'GetServiceCapabilities' message-->
<wsdl:message name="GetServiceCapabilitiesRequest">
  <wsdl:part name="parameters" element="tsc:GetServiceCapabilities"/>
</wsdl:message>
<wsdl:message name="GetServiceCapabilitiesResponse">
  <wsdl:part name="parameters" element="tsc:GetServiceCapabilitiesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetScheduleState' message-->
<wsdl:message name="GetScheduleStateRequest">
  <wsdl:part name="parameters" element="tsc:GetScheduleState"/>
</wsdl:message>
<wsdl:message name="GetScheduleStateResponse">
  <wsdl:part name="parameters" element="tsc:GetScheduleStateResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetScheduleInfo' message-->
<wsdl:message name="GetScheduleInfoRequest">
  <wsdl:part name="parameters" element="tsc:GetScheduleInfo"/>
</wsdl:message>
<wsdl:message name="GetScheduleInfoResponse">
  <wsdl:part name="parameters" element="tsc:GetScheduleInfoResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetScheduleInfoList' message-->
<wsdl:message name="GetScheduleInfoListRequest">
  <wsdl:part name="parameters" element="tsc:GetScheduleInfoList"/>
</wsdl:message>
<wsdl:message name="GetScheduleInfoListResponse">
  <wsdl:part name="parameters" element="tsc:GetScheduleInfoListResponse"/>
```

```
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSchedules' message-->
<wsdl:message name="GetSchedulesRequest">
  <wsdl:part name="parameters" element="tsc:GetSchedules"/>
</wsdl:message>
<wsdl:message name="GetSchedulesResponse">
  <wsdl:part name="parameters" element="tsc:GetSchedulesResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetScheduleList' message-->
<wsdl:message name="GetScheduleListRequest">
  <wsdl:part name="parameters" element="tsc:GetScheduleList"/>
</wsdl:message>
<wsdl:message name="GetScheduleListResponse">
  <wsdl:part name="parameters" element="tsc:GetScheduleListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'CreateSchedule' message-->
<wsdl:message name="CreateScheduleRequest">
  <wsdl:part name="parameters" element="tsc:CreateSchedule"/>
</wsdl:message>
<wsdl:message name="CreateScheduleResponse">
  <wsdl:part name="parameters" element="tsc:CreateScheduleResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'SetSchedule' message-->
<wsdl:message name="SetScheduleRequest">
  <wsdl:part name="parameters" element="tsc:SetSchedule"/>
</wsdl:message>
<wsdl:message name="SetScheduleResponse">
  <wsdl:part name="parameters" element="tsc:SetScheduleResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'ModifySchedule' message-->
<wsdl:message name="ModifyScheduleRequest">
  <wsdl:part name="parameters" element="tsc:ModifySchedule"/>
</wsdl:message>
<wsdl:message name="ModifyScheduleResponse">
  <wsdl:part name="parameters" element="tsc:ModifyScheduleResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DeleteSchedule' message-->
<wsdl:message name="DeleteScheduleRequest">
  <wsdl:part name="parameters" element="tsc:DeleteSchedule"/>
</wsdl:message>
<wsdl:message name="DeleteScheduleResponse">
  <wsdl:part name="parameters" element="tsc:DeleteScheduleResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSpecialDayGroupInfo' message-->
<wsdl:message name="GetSpecialDayGroupInfoRequest">
  <wsdl:part name="parameters" element="tsc:GetSpecialDayGroupInfo"/>
</wsdl:message>
<wsdl:message name="GetSpecialDayGroupInfoResponse">
  <wsdl:part name="parameters" element="tsc:GetSpecialDayGroupInfoResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSpecialDayGroupInfoList' message-->
<wsdl:message name="GetSpecialDayGroupInfoListRequest">
  <wsdl:part name="parameters" element="tsc:GetSpecialDayGroupInfoList"/>
</wsdl:message>
<wsdl:message name="GetSpecialDayGroupInfoListResponse">
  <wsdl:part name="parameters" element="tsc:GetSpecialDayGroupInfoListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSpecialDayGroups' message-->
<wsdl:message name="GetSpecialDayGroupsRequest">
  <wsdl:part name="parameters" element="tsc:GetSpecialDayGroups"/>
</wsdl:message>
<wsdl:message name="GetSpecialDayGroupsResponse">
  <wsdl:part name="parameters" element="tsc:GetSpecialDayGroupsResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'GetSpecialDayGroupList' message-->
<wsdl:message name="GetSpecialDayGroupListRequest">
  <wsdl:part name="parameters" element="tsc:GetSpecialDayGroupList"/>
</wsdl:message>
```

```
<wsdl:message name="GetSpecialDayGroupListResponse">
  <wsdl:part name="parameters" element="tsc:GetSpecialDayGroupListResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'CreateSpecialDayGroup' message-->
<wsdl:message name="CreateSpecialDayGroupRequest">
  <wsdl:part name="parameters" element="tsc:CreateSpecialDayGroup"/>
</wsdl:message>
<wsdl:message name="CreateSpecialDayGroupResponse">
  <wsdl:part name="parameters" element="tsc:CreateSpecialDayGroupResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'SetSpecialDayGroup' message-->
<wsdl:message name="SetSpecialDayGroupRequest">
  <wsdl:part name="parameters" element="tsc:SetSpecialDayGroup"/>
</wsdl:message>
<wsdl:message name="SetSpecialDayGroupResponse">
  <wsdl:part name="parameters" element="tsc:SetSpecialDayGroupResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'ModifySpecialDayGroup' message-->
<wsdl:message name="ModifySpecialDayGroupRequest">
  <wsdl:part name="parameters" element="tsc:ModifySpecialDayGroup"/>
</wsdl:message>
<wsdl:message name="ModifySpecialDayGroupResponse">
  <wsdl:part name="parameters" element="tsc:ModifySpecialDayGroupResponse"/>
</wsdl:message>
<!--================================-->
<!--Definition of 'DeleteSpecialDayGroup' message-->
<wsdl:message name="DeleteSpecialDayGroupRequest">
  <wsdl:part name="parameters" element="tsc:DeleteSpecialDayGroup"/>
</wsdl:message>
<wsdl:message name="DeleteSpecialDayGroupResponse">
  <wsdl:part name="parameters" element="tsc:DeleteSpecialDayGroupResponse"/>
</wsdl:message>
<!--================================-->
<wsdl:portType name="SchedulePort">
  <wsdl:operation name="GetServiceCapabilities">
    <wsdl:documentation>
      This operation returns the capabilities of the schedule service.
    </wsdl:documentation>
    <wsdl:input message="tsc:GetServiceCapabilitiesRequest"/>
    <wsdl:output message="tsc:GetServiceCapabilitiesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetScheduleState">
    <wsdl:documentation>
      This operation requests the ScheduleState for the schedule instance specified by the
      given token.
    </wsdl:documentation>
    <wsdl:input message="tsc:GetScheduleStateRequest"/>
    <wsdl:output message="tsc:GetScheduleStateResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetScheduleInfo">
    <wsdl:documentation>
      This method returns a list of schedule info items, specified in the request.
      Only found schedules shall be returned, i.e., the returned numbers of elements can
      differ from the requested element.
      The device shall ignore tokens it cannot resolve and shall return an empty list if there
      are no items matching the specified tokens.
      If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be
      returned.
    </wsdl:documentation>
    <wsdl:input message="tsc:GetScheduleInfoRequest"/>
    <wsdl:output message="tsc:GetScheduleInfoResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetScheduleInfoList">
    <wsdl:documentation>
      This operation requests a list of all of ScheduleInfo items provided by the device.
      A call to this method shall return a StartReference when not all data is returned and
      more data is available. The reference shall be valid for retrieving the next set of
      data.
      Please refer Access Control Service Specification for more details.
      The number of items returned shall not be greater the Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tsc:GetScheduleInfoListRequest"/>
    <wsdl:output message="tsc:GetScheduleInfoListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetSchedules">
```

```
      <wsdl:documentation>
        This operation returns the specified schedule item matching the given tokens.
        The device shall ignore tokens it cannot resolve and shall return an empty list if there
        are no items matching the specified tokens.
        If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be
        returned
      </wsdl:documentation>
      <wsdl:input message="tsc:GetSchedulesRequest"/>
      <wsdl:output message="tsc:GetSchedulesResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetScheduleList">
      <wsdl:documentation>
        This operation requests a list of all of Schedule items provided by the device.
        A call to this method shall return a StartReference when not all data is returned and
        more data is available. The reference shall be valid for retrieving the next set of
        data.
        Please refer Access Control Service Specification for more details.
        The number of items returned shall not be greater the Limit parameter.
      </wsdl:documentation>
      <wsdl:input message="tsc:GetScheduleListRequest"/>
      <wsdl:output message="tsc:GetScheduleListResponse"/>
    </wsdl:operation>
    <wsdl:operation name="CreateSchedule">
      <wsdl:documentation>
        This operation creates the specified schedule. The token field of the schedule structure
        shall be empty, the device shall allocate a token for the schedule. The allocated token
        shall be returned in the response. If the client sends any value in the token field,
        the device shall return InvalidArgVal as generic fault code.
      </wsdl:documentation>
      <wsdl:input message="tsc:CreateScheduleRequest"/>
      <wsdl:output message="tsc:CreateScheduleResponse"/>
    </wsdl:operation>
    <wsdl:operation name="SetSchedule">
      <wsdl:documentation>
        This operation modifies or creates the specified schedule.
      </wsdl:documentation>
      <wsdl:input message="tsc:SetScheduleRequest"/>
      <wsdl:output message="tsc:SetScheduleResponse"/>
    </wsdl:operation>
    <wsdl:operation name="ModifySchedule">
      <wsdl:documentation>
        This operation modifies or updates the specified schedule.
      </wsdl:documentation>
      <wsdl:input message="tsc:ModifyScheduleRequest"/>
      <wsdl:output message="tsc:ModifyScheduleResponse"/>
    </wsdl:operation>
    <wsdl:operation name="DeleteSchedule">
      <wsdl:documentation>
        This operation will delete the specified schedule.
        If it is associated with one or more entities some devices may not be able to delete the
        schedule, and consequently a ReferenceInUse fault shall be generated.
      </wsdl:documentation>
      <wsdl:input message="tsc:DeleteScheduleRequest"/>
      <wsdl:output message="tsc:DeleteScheduleResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetSpecialDayGroupInfo">
      <wsdl:documentation>
        This operation requests a list of SpecialDayGroupInfo items matching the given tokens.
        The device shall ignore tokens it cannot resolve and shall return an empty list if there
        are no items matching specified tokens. The device shall not return a fault in this
        case.
        If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be
        returned.
      </wsdl:documentation>
      <wsdl:input message="tsc:GetSpecialDayGroupInfoRequest"/>
      <wsdl:output message="tsc:GetSpecialDayGroupInfoResponse"/>
    </wsdl:operation>
    <wsdl:operation name="GetSpecialDayGroupInfoList">
      <wsdl:documentation>
        This operation requests a list of all of SpecialDayGroupInfo items provided by the
        device.
        A call to this method shall return a StartReference when not all data is returned and
        more data is available. The reference shall be valid for retrieving the next set of
        data.
        The number of items returned shall not be greater than Limit parameter.
      </wsdl:documentation>
      <wsdl:input message="tsc:GetSpecialDayGroupInfoListRequest"/>
      <wsdl:output message="tsc:GetSpecialDayGroupInfoListResponse"/>
```

```
  </wsdl:operation>
  <wsdl:operation name="GetSpecialDayGroups">
    <wsdl:documentation>
      This operation returns the specified special day group item matching the given token.
    </wsdl:documentation>
    <wsdl:input message="tsc:GetSpecialDayGroupsRequest"/>
    <wsdl:output message="tsc:GetSpecialDayGroupsResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetSpecialDayGroupList">
    <wsdl:documentation>
      This operation requests a list of all of SpecialDayGroupList items provided by the
      device.
      A call to this method shall return a StartReference when not all data is returned and
      more data is available. The reference shall be valid for retrieving the next set of
      data.
      Refer Access Control Service Specification for more details.
      The number of items returned shall not be greater the Limit parameter.
    </wsdl:documentation>
    <wsdl:input message="tsc:GetSpecialDayGroupListRequest"/>
    <wsdl:output message="tsc:GetSpecialDayGroupListResponse"/>
  </wsdl:operation>
  <wsdl:operation name="CreateSpecialDayGroup">
    <wsdl:documentation>
      This operation creates the specified special day group. The token field of the
      SpecialDayGroup structure shall be empty, the device shall allocate a token for the
      special day group. The allocated token shall be returned in the response.
      If there is any value in the token field, the device shall return InvalidArgVal as
      generic fault code.
    </wsdl:documentation>
    <wsdl:input message="tsc:CreateSpecialDayGroupRequest"/>
    <wsdl:output message="tsc:CreateSpecialDayGroupResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SetSpecialDayGroup">
    <wsdl:documentation>
      This operation modifies or creates the specified special day group.
    </wsdl:documentation>
    <wsdl:input message="tsc:SetSpecialDayGroupRequest"/>
    <wsdl:output message="tsc:SetSpecialDayGroupResponse"/>
  </wsdl:operation>
  <wsdl:operation name="ModifySpecialDayGroup">
    <wsdl:documentation>
      This operation updates the specified special day group.
    </wsdl:documentation>
    <wsdl:input message="tsc:ModifySpecialDayGroupRequest"/>
    <wsdl:output message="tsc:ModifySpecialDayGroupResponse"/>
  </wsdl:operation>
  <wsdl:operation name="DeleteSpecialDayGroup">
    <wsdl:documentation>
      This method deletes the specified special day group.
      If it is associated with one or more schedules some devices may not be able to delete
      the special day group, and consequently a ReferenceInUse fault must be generated.
    </wsdl:documentation>
    <wsdl:input message="tsc:DeleteSpecialDayGroupRequest"/>
    <wsdl:output message="tsc:DeleteSpecialDayGroupResponse"/>
  </wsdl:operation>
</wsdl:portType>
<!--Protocol & data format for the operations and messages for the port type 'SchedulePort'-->
<wsdl:binding name="ScheduleBinding" type="tsc:SchedulePort">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="GetServiceCapabilities">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetServiceCapabilities"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="GetScheduleState">
    <soap:operation
        soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetScheduleState"/>
    <wsdl:input>
      <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"/>
    </wsdl:output>
```

```
</wsdl:operation>
<wsdl:operation name="GetScheduleInfo">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetScheduleInfo"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetScheduleInfoList">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetScheduleInfoList"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSchedules">
  <soap:operation soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetSchedules"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetScheduleList">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetScheduleList"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreateSchedule">
  <soap:operation soapAction="http://www.onvif.org/ver10/schedule/wsdl/CreateSchedule"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetSchedule">
  <soap:operation soapAction="http://www.onvif.org/ver10/schedule/wsdl/SetSchedule"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ModifySchedule">
  <soap:operation soapAction="http://www.onvif.org/ver10/schedule/wsdl/ModifySchedule"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DeleteSchedule">
  <soap:operation soapAction="http://www.onvif.org/ver10/schedule/wsdl/DeleteSchedule"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSpecialDayGroupInfo">
  <soap:operation
```

```
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetSpecialDayGroupInfo"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSpecialDayGroupInfoList">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetSpecialDayGroupInfoList"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSpecialDayGroups">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetSpecialDayGroups"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetSpecialDayGroupList">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/GetSpecialDayGroupList"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="CreateSpecialDayGroup">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/CreateSpecialDayGroup"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="SetSpecialDayGroup">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/SetSpecialDayGroup"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ModifySpecialDayGroup">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/ModifySpecialDayGroup"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="DeleteSpecialDayGroup">
  <soap:operation
      soapAction="http://www.onvif.org/ver10/schedule/wsdl/DeleteSpecialDayGroup"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
```

```
  </wsdl:binding>
</wsdl:definitions>
```

Annex B
(informative)

Mapping of mandatory functions in IEC 60839-11-1

The following Table B.1 to Table B.6 provide mapping of mandatory and optional requirements in IEC 60839-11-1 in accordance with Web services specifications. The monitoring (Mon column in the tables) and administration (Admin column in the tables) are typically done by a monitoring console.

NOTE Abbreviations used in Table B.1 to Table B.6 are the following:

NP = not permitted

OP = optional

M = mandatory

OP* = one of the options in the identified grouping (gray area) shall be implemented

M* = mandatory only if optional functionality is supported for the specified grade

N/A = not applicable

NR = Not relevant to the protocol defined by this document

NS = Not supported by this document

S = Supported (31 refers to IEC 60839-1-31, 32 refers to IEC 60839-11-32 and 33 refers to IEC 60839-11-33 followed by clause reference in the respective document)

Table B.1 – Access point interface requirements

Access point interface requirements		Grade assignment				Mon	Admin
		1	2	3	4		
	A – Release timing						
1	The release time shall be system-defined	OP*	OP*	NP	NP	NR	NR
2	The release time shall be configurable per portal	OP*	OP*	M	M	NR	S (32: 6.3.2.3)
3	When the release time is system-defined, the permitted value shall not to be less than 3 s	M	M	N/A	N/A	NR	NR
4	When the release time is configurable, several permitted values can be associated to access rights per portal	OP	OP	OP	OP	NR	NS
	B – Access control						
5	Provide access control for entry into a protected (controlled) area	M	M	M	M	S (32:5.7)	S (33: 6.3.2.2)
6	Provide access control for exit from a protected (controlled) area	OP	M	M	M	S (32:5.7)	S (33: 6.3.2.2)
7	Hard anti-passback	OP	OP	M	M	NS	NS
8	Soft anti-passback	OP	OP	OP	OP	NS	NS
9	Global anti-passback	OP	OP	OP	M	NS	NS
10	Anti-passback override/disabling	OP	OP	OP	M	NS	NS
11	Timed anti-passback	OP	OP	OP	M	NS	NS
12	Access granted conditional upon effective/expiry date	OP	OP	M	M	S (32:5.7)	S (33:5.3.2.1 & 33:5.3.2.6)
13	Access granted conditional upon credential validity (blocked, suspended, invalid)	M	M	M	M	S (32:5.7)	S (33: 5.3.2.7)
14	Visitor escorted access	OP	OP	OP	OP	NS	NS
15	Supervisor mode	OP	OP	OP	OP	NS	NS
16	Dual occupancy (two or more persons presence check)	OP	OP	OP	OP	NS	NS
17	Dual access (two-person access)	OP	OP	OP	M	S (32:5.7)	S (33: 7.4.2.4)
18	Singularization/ anti-tailgating	OP	OP	OP	OP	NS	NS
19	Elevator control	OP	OP	OP	OP	NS	NS
	C – Access point status monitoring						
20	Access point/status shall be monitored	OP	M	M	M	S (32:6.6.2)	S (32:6.5)
21	Access point permitted open time shall be system-defined (recommended open time to be not less than 10 s)	OP	OP*	NP	NP	NR	NR
22	Access point open time shall be configurable per portal	OP	OP*	M	M	NR	S (32: 6.3.2.3)
23	When configurable, several permitted open times may be associated with access rights per access point	OP	OP	OP	OP	NR	NS
	D – Input signals						
24	Digital input signals (i.e. other than communication signals) with an active period exceeding 400 ms shall be processed	OP	M	M	M	S (31:9)	NR

Table B.2 – Indication and annunciation requirements

Indication and annunciation requirements		Indication			Grade assignment				Mon	Admin
		Display	Alert	Logging	1	2	3	4		
	B – Monitoring console (annunciation)									
5	Visual annunciation is required when access is granted	•			OP	OP	OP	OP	S (32:5.7.3)	NR
6	Logging is required when access is granted			•	OP	OP	M	M	S (32:5.7.3)	NR
7	Visual annunciation, alert and logging is required for duress conditions	•	•	•	OP	OP	OP	M	S (32:5.7.8)	NR
8	Card usage counter	•		•	OP	OP	OP	OP	S (32:5.7)	NR
9	Visual annunciation, alert and logging required for denial of access due to an attempt to use a token with expired validity	•	•	•	OP	OP	OP	M	S (32:5.7.6)	NR
10	Visual annunciation, alert and logging required for denial of access due to a configurable number of attempts to use a valid token with invalid memorized information. Where the number of attempts is not configurable it shall be limited to 5	•	•	•	OP	OP	OP	M	S (32:5.7.6)	NR
11	Visual annunciation, alert and logging required for denial of access due to a configurable number of sequential attempts to use invalid memorized information (use PIN only for recognition). Where the number of attempts is not configurable it shall be limited to 5 subsequent attempts within 30 s each	•	•	•	OP	OP	NP	NP	S (32:5.7.6)	NR
12	Visual indication of access points alerts on the floor plan of the controlled areas	•			OP	OP	OP	M	S (32:6.6.2)	NR
13	Instructions shall be displayed following alerts	•			OP	OP	OP	M	S (32:6.6.2)	NR
14	Transactions			•	OP	M	M	M	S (32:5.7)	NR
15	Visual annunciation and logging for portal open status following access granted. It may be configurable by portal	•		•	OP	OP	M	M	S (32:5.7.4)	NR
16	Visual annunciation, alert and logging for portal remain closed status following access granted. It may be configurable by portal	•	•	•	OP	OP	OP	M	S (32:5.7.5)	NR
17	Access denied. It may be configurable by portal	•	•	•	OP	OP	M	M	S (32:5.7.6)	NR

Indication and annunciation requirements		Indication			Grade assignment				Mon	Admin
		Display	Alert	Logging	1	2	3	4		
18	Cause of access denial. It may be configurable by portal and/or cause of denial	•	•	•	OP	OP	OP	M	S (32:5.7.6)	NR
19	Scheduled or manual portal status change			•	OP	OP	M	M	S (32:6.6.2)	S (32:6.5)
20	Primary power failure	•	•	•	OP	OP	M	M	S (31:8.6.7)	NR
21	Primary power restoration	•		•	OP	OP	M	M	NS	NR
22	Standby power supply trouble condition (low battery voltage level and no battery present)	•	•	•	OP	OP	M	M	S (31:8.6.6)	NR
23	Entering and leaving configuration mode	•		•	OP	OP	M	M	NR	NR
24	Loss of communication between access control unit and monitoring console	•	•	•	OP	M	M	M	S (31:8.6.2)	NR
25	Roll call	•		•	OP	OP	M	M	S (32:5.7.4)	NS
26	Portal closed following portal forced open or portal opened too long	•		•	OP	OP	M	M	S (32:6.6.2)	NR
27	All events shall be identified by type, location, time and date of occurrence	•		•	OP	OP	M	M	S (31:10)	NR
28	Alerts shall contain an indication of their respective priority level if the system allows assigning of such priority levels	•		•	OP	OP	M	M	NS	NS
29	Concurrently received alerts shall be displayed by order of priority if the system allows assigning of such priority levels	•			OP	OP	M	M	NS	NS
30	Tamper detection	•	•	•	OP	M	M	M	S (32:6.6.2)	NR
31	Portal forced open	•	•	•	OP	M	M	M	S (32:6.6.2)	NR
32	Visual annunciation, alert and logging for expiry of portal allowed open time (portal opened too long)	•	•	•	OP	M	M	M	S (32:6.6.2)	NR
33	Card trace	•		•	OP	OP	OP	M	S (32:5.7)	NR
34	Reader trace	•		•	OP	OP	OP	M	S (32:5.7)	NR
35	Reader condition off-line	•	•	•	OP	OP	OP	M	NS	NR
36	Locking device abnormal status	•	•	•	OP	OP	OP	M	S (32:6.6.2)	NR
37	Annunciation of reaching the limit of 90 % from maximum logging capacity	•	•	•	OP	OP	M	M	NR	NR

Indication and annunciation requirements		Indication			Grade assignment				Mon	Admin
		Display	Alert	Logging	1	2	3	4		
38	Maximum delay time for signals reaching the monitoring console (90 s, 45 s and 15 s)	•	•	•	OP	90 s	45 s	15 s	NR	NR
39	Maximum delay time for displaying text instructions following alert reaching the monitoring console (5 s)	•	•		OP	OP	OP	M	NR	NR
40	Maximum delay time for displaying image and graphics following alert reaching the monitoring console (6 s)	•	•		OP	OP	OP	6 s	NR	NR
41	System shall be capable to assign priority levels to specific alert events	•			OP	OP	M	M	NR	NR
42	Alerts received at the monitoring console require acknowledgement by the operator	•	•	•	OP	OP	M	M	NR	NR
43	Visual annunciation, alert and logging are required when dual/multiple occupancy conditions are not respected (minimum number of persons not present)	•	•	•	OP	OP	OP	OP	NS	NS
44	All operator initiated changes shall be logged with type, operator ID, time and date of the occurrence			•	OP	OP	OP	M	NR	NR
45	Operator comments to alerts shall be logged with operator ID, time and date of entering the comment. The specific alert covered by the comments shall be identified	•		•	OP	OP	OP	M	NR	NR
46	Accessing logged information for retrieving (e.g. displaying, printing, exporting) events shall be logged with operator ID, time and date of occurrence			•	OP	OP	M	M	NR	NR
47	Minimum number of system events logging capacity on average per reader			•	OP	200	500	1000	NR	NR

Table B.3 – Recognition requirements

Recognition requirements		Grade assignment				Mon	Admin
		1	2	3	4		
	A – Access levels						
1	The built-in real time clock shall have an accuracy of ±10 s a week and be capable of adjusting to daylight saving time, leap year	OP	M	M	M	NR	NR
2	The system shall be capable of managing multiple time zones	OP	OP	OP	OP	NR	S (33:8.4.2.2)
3	For systems with multiple interconnected control units, the clocks shall be synchronized with the master clock or other reliable synchronization source, at least once every 24 h	OP	OP	M	M	NR	S (31:8.3.7)
4	Synchronize the master clock of the system to the official time	OP	OP	OP	M	NR	S (31:8.3.7)
5	Real time clock shall be kept for the indicated minimum period of time in case of total power loss (except for loss of data retention battery)	OP	24 h	120 h	120 h	NR	NR
6	Minimum number of user access levels	1	8	16	64	NR	S (33:6.2.2)
7	Minimum number of configurable time periods	0	4	8	16	NR	S (33:8.3.2)
8	Minimum resolution for time within access level includes day of week, hours and minutes of day	N/A	M	M	M	NR	S (33:8.4.2.2)
9	Minimum resolution for time within access level includes day of month, month and year	N/A	OP	OP	M	NR	S (33:8.4.2.2)
10	System shall be capable of handling a number of configurable days (e.g. statutory holidays, special business days and non-business days)	N/A	2	16	24	NR	S (33:8.3.2)
11	System should be capable of assigning access rights to a group of credentials	OP	OP	OP	OP	NR	S (33:5.3.2.6)
12	System should be capable of changing access rights to a group of credentials in response to emergency conditions	OP	OP	OP	OP	NR	NS
	B – Equipment and methods of recognition						
13	The system shall assign a unique identity to each authorized user	OP	M	M	M	NR	S (33:5.3.2.1)
14	The system shall use memorized information only	OP*	OP*	NP	NP	NR	S (33:7.5.2.2)
15	The system shall use biometrics alone or in combination with other recognition methods	OP*	OP*	OP*	OP*	NR	S (33: 7.5.2.2)
16	The system shall use token	OP*	OP*	OP*	OP*	NR	S (33: 7.5.2.2)
17	The system shall use memorized information and token	OP*	OP*	OP*	OP*	NR	S (33: 7.5.2.2)
18	Access shall be denied after each attempt to gain access using a valid token with invalid memorized information, and after a predetermined number of unsuccessful attempts the access rights for that token shall be suspended for a pre-set duration The number of attempts can be configurable. Where it is not configurable the number of attempts shall be limited to 5	OP	M	M	M	S (32:5.7.6)	NR
19	Access shall be denied after each attempt to gain access with invalid memorized information only. The access shall be suspended after 5 sequential incorrect inputs within a pre-set period of time	OP	OP	N/A	N/A	S (32:5.7.6)	NR

Recognition requirements		Grade assignment				Mon	Admin
		1	2	3	4		
20	When using biometrics, FAR_{eff} shall not exceed limits shown for each grade. NOTE 1 FAR_{eff} = FAR (false acceptance rate) when 1:1 comparison is performed (e.g. biometric verification of an identity claimed by memorized information or token) or $FAR_{eff} = FAR \times n$ when 1:n comparison is performed and n = number of stored templates (e.g. biometric identification without using memorized information or token). NOTE 2 The FAR values are based on the review of the supplied manufacturer's documentation.	1 %	0,3 %	0,3 %	0,1 %	NR	NR
21	The minimum ratio between number of possible user codes and number of allocated codes shall be at least 1 000 to 1 when the system is using recognition of a valid user by memorized information only e.g.: up to 10 users: 4 digits, up to 100 users: 5 digits, up to 1 000 users: 6 digits, etc	M	M	N/A	N/A	NR	NR
22	For systems using recognition by memorized information combined with token or biometrics the memorized information requires 4 digits minimum	OP	OP	M	M	NR	NS
23	In normal mode of operation the system shall use complete token information (facility code and card number, or unique card number) for recognition	M	M	M	M	NR	S (33:5.3.2.5)
24	System utilizing facility coding to support multiple facility codes	OP	OP	OP	M	NR	S (33:5.3.2.5)
25	In degraded mode of operation the system may use partial token information (e.g. facility code only) for recognition	OP	OP	OP	NP	NR	NR
26	Tokens with a coding system structure visible to unaided human eye shall not be used	M	M	M	M	NR	NR
27	The token identity number readable on the token not to be a direct representation of the entire coding	M	M	M	M	NR	NR

Table B.4 – Duress signalling requirements

Duress signalling requirements		Grade assignment				Mon	Admin
		1	2	3	4		
1	Enabling of the duress functionality shall be configurable	OP	OP	OP	M	NR	S (32:5.3.1.3)
2	The duress alert at the monitoring console to be distinct from other alerts	M*	M*	M*	M	S (32:5.7.8)	NR
3	The operation of the duress initiating device shall not produce a signal which may be audible or visible at the location where the duress has been initiated	M*	M*	M*	M	NR	NR

Table B.5 – Overriding requirements

Overriding requirements		Grade assignment				Mon	Admin
		1	2	3	4		
1	Single free access granting, single portal	OP	OP	M	M	S (32:6.6.2)	S (32:6.5.2)
2	System-wide free access granting	OP	OP	OP	OP	S (32:6.6.2)	S (32:6.5.2)
3	Free access granting until further system command, single portal or group of portals	OP	OP	OP	OP	S (32:6.6.2)	S (32:6.5.4)
4	Scheduled/timed free access granting, single portal or group of portals	OP	OP	OP	OP	S (32:6.6.2)	NS
5	The electronic access control system shall not prohibit the free exit granted by other emergency systems (e.g. fire, environmental)	M	M	M	M	NR	NR
6	Blocking of portal until further system command, single portal or group of portals	OP	OP	OP	OP	S (32:6.6.2)	S (32.6.5.5)
7	Scheduled/timed blocking of portal, single portal or group of portals	N/A	OP	OP	OP	S (32:6.6.2)	NS

Table B.6 – System self protection requirements

Overriding requirements		Grade assignment				Mon	Admin
		1	2	3	4		
18	Encryption required for communication signals between components of the EAC system when using publicly shared networks (e.g. the Internet)	OP	OP	M	M	S (31:11)	S (31:11)

Bibliography

ONVIF Access Control Specification, *ONVIF Access Control Service Specification Version 18.06 June, 2018*
<http://www.onvif.org/specs/srv/access/ONVIF-AccessControl-Service-Spec-v1806.pdf>

ONVIF Door Control Specification, *ONVIF Door Control Service Specification Version 18.06 June, 2018*
<http://www.onvif.org/specs/srv/door/ONVIF-DoorControl-Service-Spec-v1806.pdf>

ONVIF Credential Specification, *ONVIF Credential Service Specification Version 18.06 June, 2018*
<http://www.onvif.org/specs/srv/door/ONVIF-DoorControl-Service-Spec-v1806.pdf>

ONVIF Access Rules Specification, *ONVIF Access Rules Service Specification Version 18.06 June, 2018*
<http://www.onvif.org/specs/srv/door/ONVIF-DoorControl-Service-Spec-v1806.pdf>

ONVIF Authentication Behavior Specification, *ONVIF Authentication Behavior Service Specification Version 18.06 June, 2018*
<http://www.onvif.org/specs/srv/door/ONVIF-DoorControl-Service-Spec-v1806.pdf>

ONVIF Schedule Specification, *ONVIF Schedule Service Specification Version 18.06 June, 2018*
<http://www.onvif.org/specs/srv/door/ONVIF-DoorControl-Service-Spec-v1806.pdf>

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

SYSTÈMES D'ALARME ET DE SÉCURITÉ ÉLECTRONIQUES –

Partie 11-33: Systèmes de contrôle d'accès électronique – Configuration du contrôle d'accès en fonction des services Web

AVANT-PROPOS

1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets.

L'IEC 60839-11-33 a été établie par le comité d'études 79 de l'IEC: Systèmes d'alarme et de sécurité électroniques. Il s'agit d'une Norme internationale.

Le texte de cette Norme internationale est issu des documents suivants:

FDIS	Rapport de vote
79/646/FDIS	79/648/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Le présent document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/standardsdev/publications.

Une liste de toutes les parties de la série IEC 60839, publiées sous le titre général *Systèmes d'alarme et de sécurité électroniques*, se trouve sur le site web de l'IEC.

Le comité a décidé que le contenu du présent document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. A cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture du présent document indique qu'il contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

Le présent document permet d'établir un système d'alarme et de sécurité électroniques avec les clients (généralement une console de commande) ainsi que les dispositifs (généralement une unité de contrôle d'accès) de différents fabricants qui utilisent des interfaces courantes et bien définies.

Le présent document spécifie uniquement les flux de données et de commande entre un client et les services sans aucune référence aux dispositifs physiques, car les services exigés pour la mise en œuvre d'un système de contrôle d'accès électronique (EACS, *Electronic Access Control System*) conforme peuvent ne pas être nécessairement mis en œuvre sur un dispositif unique, c'est-à-dire que l'ensemble des services peuvent être exécutés sur un panneau de commande, un logiciel agrégateur d'événements sur PC, etc.

Le présent document ne définit pas la communication interne entre une unité de contrôle d'accès et ses composants s'ils sont mis en œuvre sur un dispositif unique.

Le présent document a été élaboré sur la base des travaux réalisés par le forum de l'industrie ONVIF (Open Network Video Interface Forum). La spécification Credential de l'ONVIF, la spécification Access Rules de l'ONVIF, la spécification Authentication Behavior de l'ONVIF et la spécification Schedule de l'ONVIF sont compatibles avec le présent document.

Le présent document s'accompagne d'un ensemble de définitions d'interfaces informatiques (voir Annexe A):

- WSDL du service de l'identifiant, voir Article A.1;
- WSDL du service des règles d'accès, voir Article A.2;
- WSDL du service de comportement d'authentification, voir Article A.3;
- WSDL du service de programmation, voir Article A.4.

En raison des différences terminologiques qui existent entre l'IEC 60839-11-1:2013, l'IEC 60839-11-2:2014 et les spécifications de l'ONVIF sur lesquelles repose la présente partie de l'IEC 60839, il convient que les lecteurs lisent l'article Termes et définitions avec attention.

Les services supplémentaires nécessaires à la commande des portes et des points d'accès (côtés accès contrôlé) ne relèvent pas du domaine d'application du présent document. Ces services sont couverts par l'IEC 60839-11-32.

SYSTÈMES D'ALARME ET DE SÉCURITÉ ÉLECTRONIQUES –

Partie 11-33: Systèmes de contrôle d'accès électronique – Configuration du contrôle d'accès en fonction des services Web

1 Domaine d'application

La présente partie de l'IEC 60839 définit l'interface de services Web pour les systèmes de contrôle d'accès électronique. Elle répertorie notamment les composants des systèmes de contrôle d'accès électronique et spécifie leur composition logique, leur contrôle et celui de leurs états. Elle définit également le mapping des exigences obligatoires et facultatives conformément à l'IEC 60839-11-1:2013 à l'Annexe B.

Le présent document s'applique uniquement à la sécurité physique. La sécurité physique empêche l'accès physique à un bâtiment, un local, etc., à tout personnel non autorisé, à des agresseurs ou à des intrus occasionnels.

L'utilisation des services Web et la fonctionnalité de gestion de dispositif ne relèvent pas du domaine d'application du présent document. Pour plus d'informations, se reporter à l'IEC 60839-11-31:2016.

Le présent document n'empêche en aucune façon un fabricant d'ajouter d'autres protocoles ou de développer le protocole défini ci-après. Pour plus d'informations sur les règles d'ajout ou d'extension, se reporter à l'IEC 60839-11-31:2016.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 60839-11-1:2013, *Systèmes d'alarme et de sécurité électroniques – Partie 11-1: Systèmes de contrôle d'accès électronique – Exigences système et exigences concernant les composants*

IEC 60839-11-2:2014, *Systèmes d'alarme et de sécurité électroniques – Partie 11-2: Systèmes de contrôle d'accès électronique – Lignes directrices d'application*

IEC 60839-11-31:2016, *Systèmes d'alarme et de sécurité électroniques – Partie 11-31: Systèmes de contrôle d'accès électronique – Protocole de base d'interopérabilité en fonction des services Web*

IEC 60839-11-32:2016, *Systèmes d'alarme et de sécurité électroniques – Partie 11-32: Systèmes de contrôle d'accès électronique – Commande de contrôle d'accès en fonction des services Web*

ISO 16484-5:2017, *Building automation and control systems (BACS) – Part 5: Data communication protocol* (disponible en anglais seulement)

RFC 5545, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)* (disponible en anglais seulement)
(disponible à l'adresse https://tools.ietf.org/html/rfc5545)

RFC 5234, *Augmented BNF for Syntax Specifications: ABNF* (disponible en anglais seulement)
(disponible à l'adresse https://tools.ietf.org/html/rfc5234)

3 Termes et définitions

Pour les besoins du présent document, les termes et définitions de l'IEC 60839-11-1, l'IEC 60839-11-2, l'IEC 60839-11-31 et l'IEC 60839-11-32 ainsi que les suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse http://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse http://www.iso.org/obp

NOTE Lorsque le terme IEC défini dans l'IEC 60839-11-1 et l'IEC 60839-11-2 est différent du terme utilisé dans le présent document, le terme IEC est indiqué entre parenthèses dans les en-têtes.

3.1
niveau d'accès
ensemble de règles qui permettent de déterminer les conditions d'accès (où et quand) d'un identifiant à un ou plusieurs accès contrôlés, et qui peuvent comporter des conditions de passage particulières telles que les périodes spécifiques d'ouverture autorisée de l'accès ou des accès contrôlés

Note 1 à l'article: Dans le présent document, le terme "règles d'accès" est utilisé.

3.2
point d'accès
accès contrôlé
entrée/sortie physique à laquelle l'accès peut être contrôlé par une porte, un tourniquet ou toute autre barrière de sécurité

Note 1 à l'article: Pour les besoins du présent document, le point d'accès est considéré comme une composition logique constituée d'une porte physique et d'un ou de plusieurs lecteurs chargés de contrôler l'accès dans une direction.

Note 2 à l'article: Dans le présent document, le terme "porte" a la même signification que les termes "point d'accès" et "accès contrôlé".

3.3
politique d'accès
association d'un point d'accès (côté accès contrôlé) et d'un programme (zone de temps)

Note 1 à l'article: Une politique d'accès définit quand un point d'accès (côté accès contrôlé) peut être accessible.

3.4
profil d'accès
ensemble de politiques d'accès

Note 1 à l'article: Les profils d'accès sont utilisés pour définir où et quand l'accès est octroyé.

3.5
état de violation du dispositif antiretour
état qui indique si les règles antiretour ont été violées pour un identifiant

3.6
politique d'authentification
association d'un niveau de sécurité et d'un programme (zone de temps)

Note 1 à l'article: Une politique d'authentification définit quand un point d'accès (côté accès contrôlé) exige un niveau de sécurité.

3.7
profil d'authentification
ensemble de politiques d'authentification

EXEMPLE Tous les points d'accès d'entrée (côtés accès contrôlé) sont configurés pour exiger un accès par carte pendant les heures de bureau, un accès par carte et par code PIN la nuit, et aucun accès pendant les vacances.

Note 1 à l'article: Les profils d'authentification sont utilisés pour définir le comportement d'authentification d'un type de point d'accès (côtés accès contrôlé).

3.8
identifiant
informations mémorisées, contenues dans un jeton, ou caractéristiques biologiques (personne) ou physiques (véhicule) qui permettent d'identifier de manière unique l'utilisateur ou le détenteur de l'identifiant

EXEMPLE Un numéro d'identifiant peut être stocké dans l'identifiant physique (jeton). Un code PIN est associé au jeton, mais n'est pas nécessairement stocké dans le jeton. Le numéro d'identifiant et le code PIN sont des informations d'identification qui peuvent être contenues dans le même identifiant.

Note 1 à l'article: Dans le présent document, le terme "identifiant" désigne les informations utilisées pour identifier le détenteur d'un identifiant (utilisateur) au niveau d'un point d'accès (côté accès contrôlé). Ces informations ne sont pas nécessairement contenues dans un jeton.

Note 2 à l'article: Dans le présent document, plusieurs éléments d'information (informations d'identification) peuvent appartenir au même identifiant s'ils sont liés.

3.9
informations d'identification
élément de donnée qui permet d'identifier le détenteur d'un identifiant (utilisateur) au niveau d'un point d'accès

EXEMPLE Numéro de carte, informations de carte uniques, code PIN, empreinte digitale ou autres informations biométriques.

Note 1 à l'article: Les informations d'identification spécifient également le type d'identificateur (méthode de reconnaissance), ainsi que le format de la valeur utilisée dans l'interface du service Web.

3.10
état de l'identifiant
état qui indique si un identifiant est activé ou désactivé, ou si les règles antiretour ont été violées

Note 1 à l'article: L'état peut également préciser la raison pour laquelle l'identifiant a été désactivé.

3.11
type de format
format de la valeur des informations d'identification au sein de l'interface des services Web

Note 1 à l'article: Le format n'est pas nécessairement le même lorsque les informations sont stockées dans un identifiant physique.

Note 2 à l'article: L'ONVIF prend en charge les types de formats BACnet répertoriés dans l'ISO 16484-5:2017.

3.12
iCalendar
format normalisé de l'industrie pour l'échange électronique des informations de calendrier et de planification

3.13
groupe de reconnaissance
opérateur logique OR entre les méthodes de reconnaissance dans un niveau de sécurité

EXEMPLE Un groupe de reconnaissance contient les méthodes de reconnaissance pt:Card et pt:Fingerprint. Un autre groupe contient les méthodes de reconnaissance pt:Card et pt:Face. Le point d'accès nécessite alors Card+Fingerprint (Carte+Empreinte digitale) ou Card+Face (Carte+Visage).

3.14
méthode de reconnaissance
méthode de reconnaissance d'un détenteur d'identifiant (utilisateur) au niveau d'un point d'accès (côté accès contrôlé)

EXEMPLE Les informations contenues dans un identifiant physique (jeton) comme un numéro d'identifiant, des informations mémorisées (code PIN), une empreinte digitale ou toute autre information biométrique.

3.15
type de reconnaissance
méthode de reconnaissance ou entrée anonyme

3.16
niveau de sécurité
niveau d'assurance défini pour bénéficier d'un accès à un point d'accès (côté accès contrôlé)

Note 1 à l'article: Les niveaux de sécurité peuvent être définis comme des méthodes de reconnaissance individuelles, des combinaisons de méthodes de reconnaissance (à l'aide de l'opérateur logique AND ou OR) voire aucune méthode de reconnaissance (ouvert).

3.17
groupe de jours particuliers
ensemble de jours (ou parties de la journée) qui nécessitent une neutralisation du programme normal (zone de temps)

EXEMPLE Vacances, demi-journées ou dimanches travaillés.

3.18
programme de jours particuliers
ensemble de périodes (plages horaires) de neutralisation concernant un groupe de jours particuliers

3.19
plage horaire
intervalle de temps entre deux moments donnés qui indique le début et la fin d'une période valide dans une zone de temps

Note 1 à l'article: Dans le présent document, le terme "période" est utilisé.

Note 2 à l'article: Dans le présent document, les périodes ne sont utilisées que dans les programmes de jours particuliers, c'est-à-dire les plages horaires qui neutralisent le programme normal pendant les jours particuliers.

Note 3 à l'article: Les périodes ne comportent pas de partie date.

3.20
zone de temps
une ou plusieurs plages horaires combinées à des informations calendaires

EXEMPLE Le programme (zone de temps) "Heures de travail" peut être défini comme deux périodes (plages horaires) valides les jours ouvrables, une période s'étendant de 08:00 à 11:30 et l'autre de 12:30 à 18:00. Un ensemble de programmes de jours particuliers peut être défini pour les dates auxquelles les périodes normales ne s'appliquent pas. Chaque programme de jours particuliers peut contenir un autre ensemble de périodes qui neutralisent les périodes normales pour les besoins des jours particuliers.

Note 1 à l'article: Dans le présent document, le terme "programme" est utilisé.

3.21
jeton
dispositif portable qui contient un identifiant unique lisible qui peut être associé aux données et droits d'accès d'un utilisateur mémorisés dans le système de contrôle d'accès électronique

Note 1 à l'article: Dans le présent document, le terme "identifiant physique" est utilisé.

Note 2 à l'article: Dans le présent document, le terme "jeton" est l'identificateur unique d'une entité gérée.

3.22
utilisateur
personne qui demande à franchir un point d'accès

Note 1 à l'article: Dans le présent document, le terme "détenteur d'identifiant" est utilisé.

Note 2 à l'article: Un détenteur d'identifiant n'est pas nécessairement une personne; même si un détenteur d'identifiant peut par exemple être attribué à un véhicule, l'identifiant du véhicule doit être associé à une personne.

3.23
vEvent
composant dans iCalendar qui spécifie les propriétés d'un événement

4 Vue d'ensemble

4.1 Généralités

Le présent document définit quatre interfaces de services Web:

- service de l'identifiant (voir Article 5);
- service des règles d'accès (voir Article 6);
- service de comportement d'authentification (voir Article 7);
- service de programmation (voir Article 8).

Les services ci-dessus se rapportent aux deux interfaces de services Web suivantes définies dans l'IEC 60839-11-32:2016:

- service de contrôle d'accès (voir Article 5 de l'IEC 60839-11-32:2016);
- service de commande de portes (voir Article 6 de l'IEC 60839-11-32:2016).

La Figure 1 montre comment les six interfaces de services Web s'interconnectent.

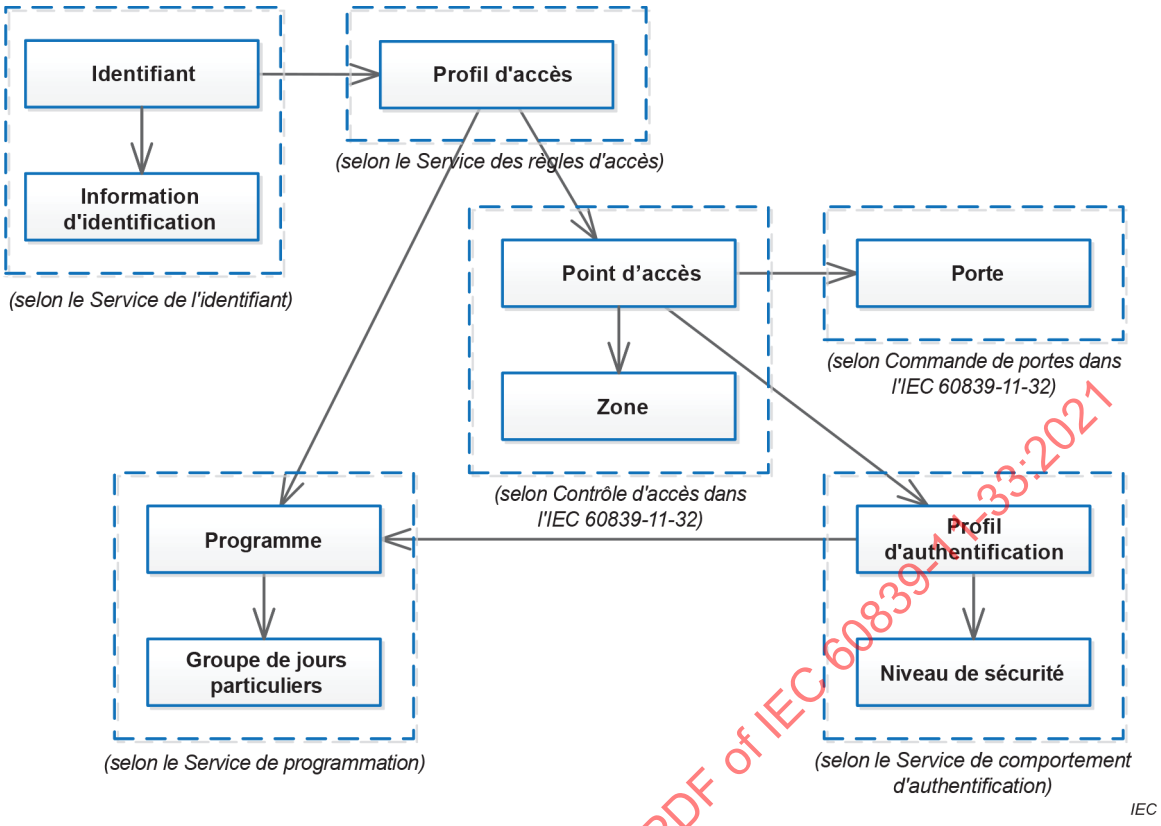


Figure 1 – Présentation des dépendances de services

Se référer également à l'Article 4 de l'IEC 60839-11-32:2016, qui est pertinent pour le présent document.

4.2 Espaces de nommage

Outre les espaces de nommage spécifiés en 5.4 de l'IEC 60839-11-31:2016, le présent document définit également les espaces de nommage répertoriés dans le Tableau 1.

Tableau 1 – Espaces de nommage définis dans le présent document

Préfixe	URI d'espace de nommage	Description
pt	https://www.onvif.org/ver10/pacs/types.xsd	Descriptions de schémas XML du présent document
tac	http://www.onvif.org/ver10/accesscontrol/wsdl	Espace de nommage pour le service de contrôle d'accès
tdc	http://www.onvif.org/ver10/doorcontrol/wsdl	Espace de nommage pour le service de commande de portes
tcr	http://www.onvif.org/ver10/credential/wsdl	Espace de nommage pour le service d'identifiant
tar	http://www.onvif.org/ver10/accessrules/wsdl	Espace de nommage pour le service de règles d'accès
tab	http://www.onvif.org/ver10/authenticationbehaviour/wsdl	Espace de nommage pour le service de comportement d'authentification
tsc	http://www.onvif.org/ver10/schedule/wsdl	Espace de nommage pour le service de programmation

Les espaces de nommage répertoriés dans le Tableau 2 sont référencés par le présent document.

Tableau 2 – Espaces de nommage référencés (avec préfixe)

Préfixe	URI d'espace de nommage	Description
env	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [W3C SOAP12-PART11]

4.3 Traitement des erreurs

Les défauts répertoriés dans le présent document sont spécifiques à chaque commande. Chaque commande peut également renvoyer les défauts génériques répertoriés en 5.12.3.2 de l'IEC 60839-11-31:2016.

5 Service de l'identifiant

5.1 Généralités

La spécification du service de l'identifiant définit les commandes qui permettent de configurer les identifiants.

Un identifiant contient des informations qui peuvent être validées au niveau d'un point d'accès, notamment des informations de carte unique, un code PIN ou des informations biométriques. Un identifiant contient également des informations concernant les éléments auxquels peut accéder l'identifiant par le biais des profils d'accès de l'identifiant, ce qui lie l'identifiant aux profils d'accès décrits à l'Article 6.

Un identifiant est attribué à une personne (appelée détenteur d'identifiant) et comporte une durée de validité qui spécifie la période au cours de laquelle l'identifiant peut être utilisé afin d'obtenir un accès.

Par exemple:

un identifiant est attribué à un consultant pour une utilisation temporaire d'une semaine (du 2 avril au 6 avril). La date/l'heure de début de validité est fixée au matin du 2 avril et la date/l'heure de fin de validité est fixée au soir du 6 avril;

cet identifiant particulier est une carte qui porte le numéro 987654321 et le consultant reçoit le code PIN personnel 1234. Les deux éléments d'information sont stockés dans l'identifiant. Le numéro de carte est une information d'identification de type pt:Card et le code PIN est une information d'identification de type pt:PIN (voir 5.2.2 pour les types d'identificateurs pris en charge);

comme le consultant contribue à l'installation du serveur, il est nécessaire qu'il ait accès à la salle des serveurs, mais également à d'autres installations communes du bureau. Le profil d'accès "Accès au support informatique" octroie un accès à la salle des serveurs pendant les heures de bureau tandis que le profil d'accès "Accès du personnel" octroie un accès au reste du bureau. Les références aux deux profils d'accès sont stockées dans l'identifiant au sein de la structure CredentialAccessProfile.

La Figure 2 représente un modèle d'objet de service de l'identifiant.

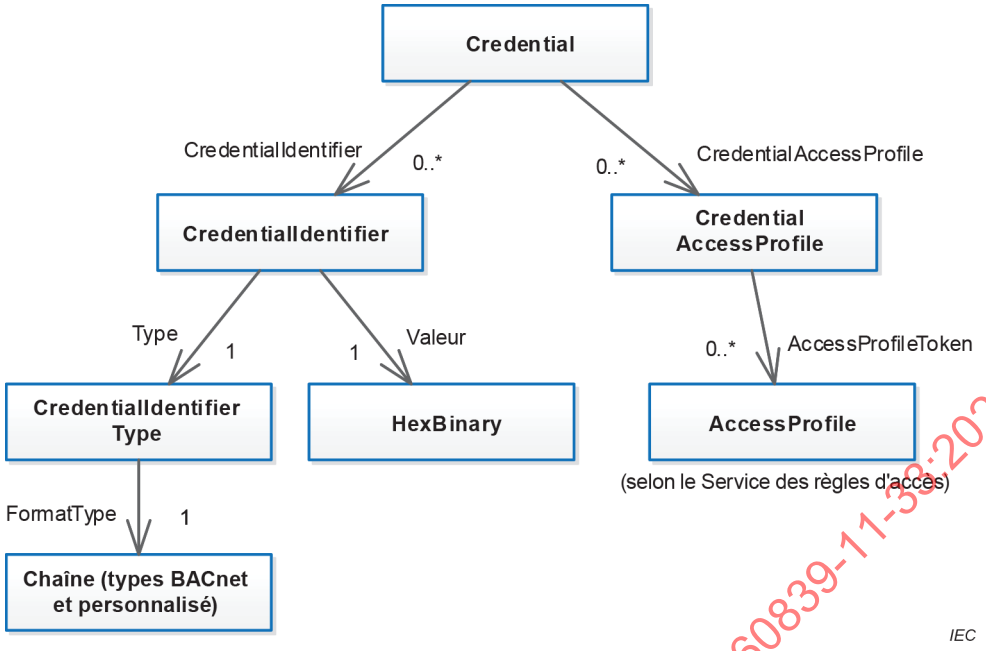


Figure 2 – Structures de données principales dans le service de l'identifiant

5.2 Fonctionnalités de service

5.2.1 Généralités

Le dispositif doit fournir les fonctionnalités de service de deux manières:

1) avec la méthode GetServices du service Dispositif (Device) lorsque IncludeCapability est défini sur true. Se reporter au 8.1 de l'IEC 60839-11-31:2016;
2) avec la méthode GetServiceCapabilities.

5.2.2 Structure de données ServiceCapabilities

Les fonctionnalités de service reflètent les fonctionnalités facultatives d'un service. Les informations sont statiques et ne changent pas pendant le fonctionnement du dispositif. Les fonctionnalités suivantes sont disponibles:

- **MaxLimit**
  indique le nombre maximal d'entités renvoyées par une demande Get<Entity>List ou Get<Entity>Request unique. Le dispositif ne doit jamais renvoyer dans une même réponse un nombre d'entités supérieur à ce nombre;
- **CredentialValiditySupported**
  indique que le dispositif prend en charge la validité de l'identifiant;
- **CredentialAccessProfileValiditySupported**
  indique que le dispositif prend en charge la validité de l'association entre un identifiant et un profil d'accès;
- **ValiditySupportsTimeValue**
  indique que le dispositif prend en charge les valeurs de date et d'heure pour la validité. Si la valeur est définie sur false, l'heure n'est pas prise en compte;
- **MaxCredentials**
  indique le nombre maximal d'identifiants pris en charge par le dispositif;
- **MaxAccessProfilesPerCredential**
  indique le nombre maximal de profils d'accès associés à un identifiant;

- **ResetAntipassbackSupported**
  indique que le dispositif prend en charge la réinitialisation et la notification des violations du dispositif antiretour;
- **SupportedIdentifierType**
  liste des types d'identificateurs pris en charge par le dispositif. Les types d'identificateurs qui commencent par le préfixe pt: sont réservés à la définition des types d'identificateurs spécifiques à l'ONVIF. Pour les types d'identificateurs personnalisés, il est permis d'utiliser du texte libre. Les types d'identificateurs suivants sont définis par l'ONVIF:
  - pt:Card prend en charge le type d'identificateur Card (y compris fob, tag et analogues);
  - pt:PIN prend en charge le type d'identificateur PIN;
  - pt:Fingerprint prend en charge le type d'identificateur biométrique Empreinte digitale (Fingerprint);
  - pt:Face prend en charge le type d'identificateur biométrique Visage (Face);
  - pt:Iris prend en charge le type d'identificateur biométrique Iris (Iris);
  - pt:Vein prend en charge le type d'identificateur biométrique Veine (Vein);
  - pt:Palm prend en charge le type d'identificateur biométrique Paume (Palm);
  - pt:Retina prend en charge le type d'identificateur biométrique Rétine (Retina).
- **SupportedExemptionType**
  liste des exemptions prises en charge par le dispositif. Les types d'exemptions qui commencent par le préfixe pt: sont réservés à la définition des types d'exemptions spécifiques à l'ONVIF. Pour les types d'exemptions personnalisés, il est permis d'utiliser du texte libre. Les types d'identificateurs suivants sont définis par l'ONVIF:
  - pt:ExemptFromAuthentication Prend en charge le champ ExemptedFromAuthentication spécifié en 5.3.2.3;
- **ClientSuppliedTokenSupported**
  indique que le client est autorisé à fournir le jeton lors de la création des identifiants. Pour permettre l'utilisation de la commande SetCredential, la valeur doit être définie sur true;
- **DefaultCredentialSuspensionDuration**
  période par défaut pendant laquelle l'identifiant est suspendu (utilisation d'un code PIN erroné pendant un nombre de fois prédéterminé, par exemple).

La période est définie sous la forme d'une chaîne de durée [ISO 8601] ("PT5M", par exemple).

5.2.3 Commande GetServiceCapabilities

Cette opération renvoie les fonctionnalités du service de l'identifiant.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 3.

Tableau 3 – Commande GetServiceCapabilities

GetServiceCapabilities	Classe d'accès: PRE_AUTH
Nom de message	**Description**
GetServiceCapabilitiesRequest	*Ce message doit être vide.*
GetServiceCapabilitiesResponse	*Ce message contient:* • *"Capabilities": le message de réponse de la fonctionnalité répertorie les fonctionnalités du service de l'identifiant demandées, organisées selon une structure de fonctionnalités XML hiérarchique.* tcr:ServiceCapabilities **Capabilities [1][1]**
Codes de défaut	**Description**
	Aucun défaut spécifique à cette commande.

5.3 Informations sur l'identifiant

5.3.1 Généralités

Un identifiant regroupe les informations qui peuvent être validées au niveau d'un point d'accès (informations de carte, code PIN, etc.) et les éléments auxquels l'identifiant peut avoir accès (profils d'accès).

5.3.2 Structures de données

5.3.2.1 CredentialInfo

Le type CredentialInfo représente l'identifiant comme un objet logique. La structure contient les informations de base concernant une instance d'identifiant spécifique. Le dispositif doit fournir les champs suivants pour chaque identifiant.

- **token**
  Identificateur unique au service de l'identifiant.
- **CredentialHolderReference**
  Référence externe au détenteur de cet identifiant. La référence est un nom d'utilisateur ou un ID utilisé dans un système externe, comme un service d'annuaire.

Afin de fournir davantage d'informations, le dispositif peut inclure les champs facultatifs suivants:

- **Description**
  description de l'identifiant lisible par l'utilisateur. Elle doit comporter 1 024 caractères au maximum;
- **ValidFrom**
  date/heure de début de validité de l'identifiant. Si la fonctionnalité ValiditySupportsTimeValue est définie sur false, seule la date est prise en charge (l'heure n'est pas prise en compte);
- **ValidTo**
  date/heure d'expiration de validité de l'identifiant. Si la fonctionnalité ValiditySupportsTimeValue est définie sur false, seule la date est prise en charge (l'heure n'est pas prise en compte).

5.3.2.2 Identifiant

Un identifiant est un objet physique/tangible, un élément de connaissance ou une caractéristique physique d'une personne, qui permet d'octroyer un accès individuel à une fonctionnalité physique donnée ou à un système d'information informatique. Un identifiant contient une ou plusieurs informations d'identification. Pour obtenir un accès, un ou plusieurs identificateurs peuvent être exigés.

Le dispositif doit inclure l'ensemble des propriétés de la structure CredentialInfo, ainsi qu'une liste des informations d'identification et une liste des profils d'accès d'identifiants.

- **CredentialIdentifier**
  Liste des structures d'informations d'identification. Au moins une information d'identification est exigée. Une structure d'informations d'identification maximale par type est autorisée.

Afin de fournir davantage d'informations, le dispositif peut inclure les champs facultatifs suivants:

- **CredentialAccessProfile**
  liste des structures de profils d'accès des identifiants;
- **Attribute**
  liste des attributs Identifiant tels que les paires nom-valeur. Les types de noms de clés qui commencent par le préfixe pt: sont réservés à la définition des types de noms de clés spécifiques à l'ONVIF. Pour les types de noms de clés personnalisés, il est permis d'utiliser du texte libre;

- **ExtendedGrantTime**
  valeur booléenne qui indique que le détenteur de l'identifiant nécessite plus de temps pour franchir la porte. ExtendedReleaseTime est ajouté à ReleaseTime et ExtendedOpenTime à OpenTime.

5.3.2.3 CredentialIdentifier

Une information d'identification représente un numéro de carte, des informations de carte uniques, un code PIN ou des informations biométriques (telles que l'empreinte digitale, la reconnaissance faciale, de l'iris et des veines) qui peuvent être validées au niveau d'un point d'accès.

- **Type**
  Contient les informations relatives à un type d'information d'identification. Le type est CredentialIdentifierType.
- **ExemptedFromAuthentication**
  Si la valeur est définie sur true, cette information d'identification n'est pas prise en compte pour l'authentification. Par exemple, si le point d'accès exige un niveau de sécurité "Card+PIN" et que les informations d'identification de type code PIN sont dispensées d'authentification, alors le code PIN n'est pas demandé au point d'accès.
- **Value**
  Valeur de l'identificateur au format hexadécimal.

5.3.2.4 CredentialIdentifierType

Spécifie le nom du type d'information d'identification et son format pour la valeur de l'identifiant.

- **Name**
  Spécifie le nom du type d'information d'identification tel que pt:Card, pt:PIN, etc. (voir 5.2.2 pour les types d'informations d'identification pris en charge).
- **FormatType**
  Spécifie le format de la valeur Identifiant du nom du type d'identificateur spécifié. Voir 5.3.2.5 ci-dessous.

5.3.2.5 CredentialIdentifierFormatTypeInfo

Contient les informations relatives au type de format.

- **FormatType**
  Type de format pris en charge par le dispositif. L'ISO 16484-5:2017 fournit une liste des types de formats pris en charge. Le type BACnet "CUSTOM" n'est pas utilisé dans le présent document. A défaut, les fabricants de dispositifs peuvent définir leurs propres types de formats.
- **Description**
  Description du type de format des informations d'identification lisible par l'utilisateur. Elle doit comporter 1 024 caractères au maximum. Pour les types personnalisés, il est recommandé de décrire la manière dont est codée la chaîne d'octets (en respectant la structure donnée dans la colonne "Authentification Factor Value Encoding" du Tableau P-1 de l'ISO 16484-5:2017).

5.3.2.6 CredentialAccessProfile

Association entre un identifiant et un profil d'accès. Voir 6.3.2.2.

- **AccessProfileToken**
  Jeton de référence du profil d'accès associé.

Le dispositif peut inclure les champs facultatifs suivants:

- **ValidFrom**
  date et heure de début de validité de l'association entre l'identifiant et le profil d'accès. Si la fonctionnalité ValiditySupportsTimeValue est définie sur false, seule la date est prise en charge (l'heure n'est pas prise en compte);
- **ValidTo**
  date et heure de fin de validité de l'association entre l'identifiant et le profil d'accès. Si la fonctionnalité ValiditySupportsTimeValue est définie sur false, seule la date est prise en charge (l'heure n'est pas prise en compte).

5.3.2.7 CredentialState

La structure CredentialState contient des informations sur l'état de l'identifiant et éventuellement la raison pour laquelle l'identifiant a été désactivé.

- **Enabled**
  Cette valeur est définie sur true si l'identifiant est activé ou sur false s'il est désactivé.

Le dispositif peut inclure les champs facultatifs suivants:

- **Reason**
  indique la raison pour laquelle l'identifiant a été désactivé. Les types de raisons qui commencent par le préfixe pt: sont réservés à la définition des types de raisons spécifiques à l'ONVIF. Pour les types de raisons personnalisés, il est permis d'utiliser du texte libre. Les types d'identificateurs suivants sont définis par l'ONVIF:
  - pt:CredentialLockedOut — l'accès est refusé, car l'identifiant est verrouillé;
  - pt:CredentialBlocked — l'accès est refusé, car l'identifiant a été bloqué délibérément par l'opérateur;
  - pt:CredentialLost — l'accès est refusé, car l'identifiant a été déclaré perdu;
  - pt:CredentialStolen — l'accès est refusé, car l'identifiant a été déclaré volé;
  - pt:CredentialDamaged — l'accès est refusé, car l'identifiant a été déclaré endommagé;
  - pt:CredentialDestroyed — l'accès est refusé, car l'identifiant a été déclaré détruit;
  - pt:CredentialInactivity — l'accès est refusé, car l'identifiant est inactif;
  - pt:CredentialExpired — l'accès est refusé, car l'identifiant a expiré;
  - pt:CredentialRenewalNeeded — l'accès est refusé, car l'identifiant exige un renouvellement (enregistrement d'un nouveau code PIN ou de nouvelles empreintes digitales, par exemple).
- **AntipassbackState**
  structure qui indique l'état du dispositif antiretour. Ce champ doit être pris en charge si la fonctionnalité ResetAntipassbackSupported est définie sur true.

5.3.2.8 Etat du dispositif antiretour

Structure qui contient les informations relatives à l'état du dispositif antiretour.

- **AntipassbackViolated**
  Indique si le dispositif antiretour de l'identifiant est violé.

5.3.2.9 Commande GetCredentialInfo

Cette opération demande une liste des éléments CredentialInfo qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 4.

Tableau 4 – Commande GetCredentialInfo

GetCredentialInfo	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetCredentialInfoRequest	*Ce message contient:* • *"Token": Jetons des éléments CredentialInfo à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetCredentialInfoResponse	*Ce message contient:* • *"CredentialInfo": Liste des éléments CredentialInfo.* tcr:CredentialInfo **CredentialInfo [0][unbounded]**
Codes de défaut	***Description***
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

5.3.3 Commande GetCredentialInfoList

Cette opération demande une liste de l'ensemble des éléments CredentialInfo fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 5.

Tableau 5 – Commande GetCredentialInfoList

GetCredentialInfoList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetCredentialInfoListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetCredentialInfoListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"CredentialInfo": Liste des éléments CredentialInfo.* xs:string **NextStartReference [0][1]** tcr:CredentialInfo **CredentialInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

5.3.4 Commande GetCredentials

Cette opération demande une liste des éléments Credential qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 6.

Tableau 6 – Commande GetCredentials

GetCredentials	Classe d'accès: READ_SYSTEM_SECRET
Nom de message	**Description**
GetCredentialsRequest	*Ce message contient:* • *"Token": Jetons des identifiants à obtenir* pt:ReferenceToken **Token [1][unbounded]**
GetCredentialsResponse	*Ce message contient:* • *"Credential": Liste des éléments Credential* tcr:Credential **Credential [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

5.3.5 Commande GetCredentialList

Cette opération demande une liste de l'ensemble des éléments Credential fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 7.

Tableau 7 – Commande GetCredentialList

GetCredentialList	Classe d'accès: READ_SYSTEM_SECRET
Nom de message	**Description**
GetCredentialListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetCredentialListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"Credential": Liste des éléments Credential* xs:string **NextStartReference [0][1]** tcr:Credential **Credential [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

5.3.6 Commande CreateCredential

Cette opération crée l'identifiant spécifié dans le dispositif.

Un appel à cette méthode prend une structure Credential et une structure CredentialState comme paramètres d'entrée. L'état de l'identifiant peut être créé à l'état désactivé ou activé.

Le champ Token de la structure Credential doit être vide, et le dispositif doit attribuer un jeton à l'identifiant. Le jeton attribué doit être renvoyé dans la réponse.

Si le client envoie une valeur dans le champ Token, le dispositif doit retourner le code de défaut générique InvalidArgVal.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 8.

Tableau 8 – Commande CreateCredential

CreateCredential	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
CreateCredentialRequest	*Ce message contient:* • *"Credential": Identifiant à crée;* • *"State": Etat de l'identifiant.* tcr:Credential **Credential [1][1]** tcr:CredentialState **State [1][1]**
CreateCredentialResponse	*Ce message contient:* • *"Token": Jeton de l'identifiant créé.* pt:ReferenceToken **Token [1][1]**
Codes de défaut	**Description**
env:Sender ter:CapabilityViolated ter:MaxCredentials	*L'espace n'est pas suffisant pour créer un nouvel identifiant (voir fonctionnalité MaxCredentials).*
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	*Le nombre de profils d'accès par identifiant est trop élevé (voir fonctionnalité MaxAccessProfilesPerCredential).*
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	*La validité de l'identifiant n'est pas prise en charge par le dispositif (voir fonctionnalité CredentialValiditySupported).*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*La validité du profil d'accès de l'identifiant n'est pas prise en charge par le dispositif (voir fonctionnalité CredentialAccessProfileValiditySupported).*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Le type d'identificateur spécifié n'est pas pris en charge par le dispositif (voir fonctionnalité SupportedIdentifierType).*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	*Le même type d'identificateur a été utilisé plusieurs fois.*
env:Sender ter:InvalidArgVal ter:InvalidFormatType	*Le type de format de l'identificateur spécifié n'est pas pris en charge par le dispositif.*
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	*L'information d'identification spécifiée n'est pas conforme à la définition FormatType.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	*La même combinaison de type d'identificateur, de format et de valeur a été utilisée plusieurs fois (certains dispositifs peuvent ne pas prendre en charge l'emploi de valeurs d'identificateur en double).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	*Le type d'exemption spécifié n'est pas pris en charge par le dispositif (voir fonctionnalité SupportedExemptionType).*

5.3.7 Commande SetCredential

Cette méthode permet de synchroniser l'identifiant d'un client avec le dispositif.

Un appel à cette méthode prend une structure Credential et une structure CredentialState comme paramètres d'entrée. Le champ Token de la structure Credential ne doit pas être vide.

Si aucun identifiant avec le jeton spécifié n'existe dans le dispositif, l'identifiant est créé. L'état de l'identifiant peut être créé à l'état désactivé ou activé.

Si un identifiant avec le jeton spécifié existe dans le dispositif, l'identifiant est modifié. L'état de l'identifiant est défini en fonction de l'état spécifié (ce comportement n'est pas le même que la commande ModifyCredential). Toutes les informations d'identification et tous les profils d'accès de l'identifiant existants sont supprimés et remplacés par les entités spécifiées.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité ClientSuppliedTokenSupported, il doit mettre en œuvre cette commande.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 9.

Tableau 9 – Commande SetCredential

SetCredential	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
SetCredentialRequest	*Ce message contient:* • *"Credential": Identifiant à créer;* • *"State": Etat de l'identifiant.* tcr:Credential **Credential [1][1]** tcr:CredentialState **State [1][1]**
SetCredentialResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*Le dispositif ne prend pas en charge la fourniture du jeton par le client.*
env:Sender ter:CapabilityViolated ter:MaxCredentials	*L'espace n'est pas suffisant pour créer un nouvel identifiant (voir fonctionnalité MaxCredentials).*
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	*Le nombre de profils d'accès par identifiant est trop élevé (voir fonctionnalité MaxAccessProfilesPerCredential).*
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	*La validité de l'identifiant n'est pas prise en charge par le dispositif (voir fonctionnalité CredentialValiditySupported).*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*La validité du profil d'accès de l'identifiant n'est pas prise en charge par le dispositif (voir fonctionnalité CredentialAccessProfileValiditySupported).*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Le type d'identificateur spécifié n'est pas pris en charge par le dispositif (voir fonctionnalité SupportedIdentifierType).*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	*Le même type d'identificateur a été utilisé plusieurs fois.*
env:Sender ter:InvalidArgVal ter:InvalidFormatType	*Le type de format de l'identificateur spécifié n'est pas pris en charge par le dispositif.*
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	*L'information d'identification spécifiée n'est pas conforme à la définition FormatType.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	*La même combinaison de type d'identificateur, de format et de valeur a été utilisée plusieurs fois (certains dispositifs peuvent ne pas prendre en charge l'emploi de valeurs d'identificateur en double).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	*Le type d'exemption spécifié n'est pas pris en charge par le dispositif (voir fonctionnalité SupportedExemptionType).*

5.3.8 Commande ModifyCredential

Cette opération modifie l'identifiant spécifié.

Le jeton de l'identifiant à modifier est spécifié dans le champ Token de la structure Credential et ne doit pas être vide. Tous les autres champs de la structure doivent écraser les champs de l'identifiant spécifié.

Lorsqu'un identifiant existant est modifié, l'état n'est pas modifié de manière explicite. La seule façon pour un client de changer l'état d'un identifiant consiste à appeler la commande EnableCredential, DisableCredential ou ResetAntipassback de manière explicite.

Toutes les informations d'identification et tous les profils d'accès de l'identifiant existants sont supprimés et remplacés par les entités spécifiées.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 10.

Tableau 10 – Commande ModifyCredential

ModifyCredential	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
ModifyCredentialRequest	*Ce message contient:* • *"Credential": Détails de l'identifiant.* tcr:Credential **Credential [1][1]**
ModifyCredentialResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	*Le nombre de profils d'accès par identifiant est trop élevé (voir fonctionnalité MaxAccessProfilesPerCredential).*
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	*La validité de l'identifiant n'est pas prise en charge par le dispositif (voir fonctionnalité CredentialValiditySupported).*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*La validité du profil d'accès de l'identifiant n'est pas prise en charge par le dispositif (voir fonctionnalité CredentialAccessProfileValiditySupported).*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Le type d'identificateur spécifié n'est pas pris en charge par le dispositif (voir fonctionnalité SupportedIdentifierType).*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	*Le même type d'identificateur a été utilisé plusieurs fois.*
env:Sender ter:InvalidArgVal ter:InvalidFormatType	*Le type de format de l'identificateur spécifié n'est pas pris en charge par le dispositif.*
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	*L'information d'identification spécifiée n'est pas conforme à la définition FormatType.*
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	*La même combinaison de type d'identificateur, de format et de valeur a été utilisée plusieurs fois (certains dispositifs peuvent ne pas prendre en charge l'emploi de valeurs d'identificateur en double).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	*Le type d'exemption spécifié n'est pas pris en charge par le dispositif (voir fonctionnalité SupportedExemptionType).*

5.3.9 Commande DeleteCredential

Cette méthode permet de supprimer l'identifiant spécifié.

S'il est associé à une ou plusieurs entités, certains dispositifs peuvent ne pas être en mesure de supprimer l'identifiant et un défaut ReferenceInUse doit donc être généré.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 11.

Tableau 11 – Commande DeleteCredential

DeleteCredential	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
DeleteCredentialRequest	*Ce message contient:* • *"Token": Jeton de l'identifiant à supprimer.* pt:ReferenceToken **Token [1][1]**
DeleteCredentialResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*

5.3.10 Commande GetCredentialState

Cette méthode permet de renvoyer l'état des identifiants spécifiés.

Si la fonctionnalité ResetAntipassbackSupported est définie sur true, le dispositif doit fournir l'état du dispositif antiretour dans la structure CredentialState renvoyée.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 12.

Tableau 12 – Commande GetCredentialState

GetCredentialState	Classe d'accès: READ_SYSTEM_SENSITIVE
Nom de message	**Description**
GetCredentialStateRequest	*Ce message contient:* • *"Token": Jeton de l'identifiant.* pt:ReferenceToken **Token [1][1]**
GetCredentialStateResponse	*Ce message contient:* • *"State": Etat de l'identifiant.* tcr:CredentialState **State [1][1]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*

5.3.11 Commande EnableCredential

Cette méthode permet d'activer un identifiant.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 13.

Tableau 13 – Commande EnableCredential

EnableCredential	Classe d'accès: ACTUATE
Nom de message	**Description**
EnableCredentialRequest	*Ce message contient:* • *"Token": Jeton de l'identifiant;* • *"Reason": Raison pour laquelle l'identifiant a été activé.* pt:ReferenceToken **Token [1][1]** pt:Name **Reason [0][1]**
EnableCredentialResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*

5.3.12 Commande DisableCredential

Cette méthode permet de désactiver un identifiant.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 14.

Tableau 14 – Commande DisableCredential

DisableCredential	Classe d'accès: ACTUATE
Nom de message	**Description**
DisableCredentialRequest	*Ce message contient:* • *"Token": Jeton de l'identifiant;* • *"Reason": Raison pour laquelle l'identifiant a été désactivé.* pt:ReferenceToken **Token [1][1]** pt:Name **Reason [0][1]**
DisableCredentialResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*

5.3.13 Commande ResetAntipassbackViolation

Cette méthode permet de réinitialiser les violations du dispositif antiretour concernant l'identifiant spécifié.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité ResetAntipassbackSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 15.

Tableau 15 – Commande ResetAntipassbackViolation

ResetAntipassbackViolation	Classe d'accès: ACTUATE
Nom de message	**Description**
ResetAntipassbackViolationRequest	*Ce message contient:* • *"CredentialToken": Jeton de l'identifiant.* pt:ReferenceToken **CredentialToken [1][1]**
ResetAntipassbackViolationResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*

5.3.14 Commande GetSupportedFormatTypes

Cette méthode permet de renvoyer l'ensemble des types de formats pris en charge d'un type d'identificateur spécifié pris en charge par le dispositif.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 16.

Tableau 16 – Commande GetSupportedFormatTypes

GetSupportedFormatTypes	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSupportedFormatTypesRequest	*Ce message contient:* • *"CredentialIdentifierTypeName": Nom du type d'information d'identification.* xs:string **CredentialIdentifierTypeName**
GetSupportedFormatTypesResponse	*Ce message contient:* • *"FormatTypeInfo": Types de formats de l'identificateur.* tcr:CredentialIdentifierFormatTypeInfo **FormatTypeInfo [1][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Type d'identificateur introuvable.*

5.3.15 Commande GetCredentialIdentifiers

Cette méthode permet de renvoyer l'ensemble des informations d'identification d'un identifiant.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 17.

Tableau 17 – Commande GetCredentialIdentifiers

GetCredentialIdentifiers	Classe d'accès: READ_SYSTEM_SECRET
Nom de message	**Description**
GetCredentialIdentifersRequest	*Ce message contient:* • *"CredentialToken": Jeton de l'identifiant.* pt:ReferenceToken **CredentialToken [1][1]**
GetCredentialIdentifersResponse	*Ce message contient:* • *"CredentialIdentifier": Identificateurs de l'identifiant* tcr:CredentialIdentifier **CredentialIdentifier [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*

5.3.16 Commande SetCredentialIdentifier

Cette opération crée ou met à jour une information d'identification d'un identifiant.

Si le type de l'information d'identification spécifiée existe déjà, l'information d'identification actuelle de ce type est remplacée. Sinon, l'information d'identification est ajoutée.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 18.

Tableau 18 – Commande SetCredentialIdentifier

SetCredentialIdentifier	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
SetCredentialIdentiferRequest	*Ce message contient:* • *"CredentialToken": Jeton de l'identifiant;* • *"CredentialIdentifier": Identificateur de l'identifiant.* pt:ReferenceToken **CredentialToken [1][1]** tcr:CredentialIdentifier **CredentialIdentifier [1][1]**
SetCredentialIdentiferResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Le type d'identificateur spécifié n'est pas pris en charge par le dispositif (voir fonctionnalité SupportedIdentifierType).*
env:Sender ter:InvalidArgVal ter:InvalidFormatType	*Le type de format de l'identificateur spécifié n'est pas pris en charge par le dispositif.*
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	*La valeur de l'information d'identification n'est pas conforme à la définition FormatType.*

5.3.17 Commande DeleteCredentialIdentifier

Cette méthode permet de supprimer l'ensemble des valeurs de l'identificateur du type spécifié. Cependant, si le nom du type d'identificateur n'existe pas dans le dispositif, ce dernier n'en tient pas compte sans générer aucune réponse.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 19.

Tableau 19 – Commande DeleteCredentialIdentifier

DeleteCredentialIdentifier	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
DeleteCredentialIdentiferRequest	*Ce message contient:* • *"CredentialToken": Jeton de l'identifiant;* • *"CredentialIdentifierTypeName": Nom du type d'identificateur d'un identifiant.* pt:ReferenceToken **CredentialToken [1][1]** pt:Name **CredentialIdentifierTypeName [1][1]**
DeleteCredentialIdentiferResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*
env:Receiver ter:ConstraintViolated ter:MinIdentifiersPerCredential	*Au moins une information d'identification est exigée.*

5.3.18 Commande GetCredentialAccessProfiles

Cette méthode permet de renvoyer l'ensemble des profils d'accès associés à un identifiant.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 20.

Tableau 20 – Commande GetCredentialAccessProfiles

GetCredentialAccessProfiles	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetCredentialAccessProfilesRequest	*Ce message contient:* • *"CredentialToken": Jeton de l'identifiant.* pt:ReferenceToken **CredentialToken [1][1]**
GetCredentialAccessProfilesResponse	*Ce message contient:* • *"CredentialAccessProfile": Profils d'accès de l'identifiant.* tcr:CredentialAccessProfile **CredentialAccessProfile [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*

5.3.19 Commande SetCredentialAccessProfiles

Cette opération ajoute ou met à jour les profils d'accès d'un identifiant.

Le dispositif doit mettre à jour le profil d'accès de l'identifiant si le jeton du profil d'accès dans le profil d'accès de l'identifiant spécifié correspond. Sinon, le profil d'accès de l'identifiant est ajouté.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 21.

Tableau 21 – Commande SetCredentialAccessProfiles

SetCredentialAccessProfiles	Classe d'accès: ACTUATE
Nom de message	**Description**
SetCredentialAccessProfilesRequest	*Ce message contient:* • *"CredentialToken": Jeton de l'identifiant;* • *"CredentialAccessProfile": Profils d'accès de l'identifiant.* pt:ReferenceToken **CredentialToken [1][1]** tcr:CredentialAccessProfile **CredentialAccessProfile [1][unbounded]**
SetCredentialAccessProfilesResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	*Le nombre de profils d'accès par identifiant est trop élevé (voir fonctionnalité MaxAccessProfilesPerCredential).*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*La validité du profil d'accès de l'identifiant n'est pas prise en charge par le dispositif (voir fonctionnalité CredentialAccessProfileValiditySupported).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*

5.3.20 Commande DeleteCredentialAccessProfiles

Cette méthode permet de supprimer les profils d'accès de l'identifiant pour le jeton d'identifiant spécifié.

Toutefois, si aucun profil d'accès de l'identifiant correspondant n'est trouvé, ce dernier ne tient pas compte des jetons du profil d'accès correspondants sans générer aucune réponse.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 22.

Tableau 22 – Commande DeleteCredentialAccessProfiles

DeleteCredentialAccessProfiles	Classe d'accès: ACTUATE
Nom de message	**Description**
DeleteCredentialAccessProfilesRequest	*Ce message contient:* • *"CredentialToken": Jeton de l'identifiant;* • *"AccessProfileToken": Jetons des profils d'accès.* pt:ReferenceToken **CredentialToken [1][1]** pt:ReferenceToken **AccessProfileToken [1][unbounded]**
DeleteCredentialAccessProfilesResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton d'identifiant est introuvable.*

5.4 Rubriques de notification

5.4.1 Généralités

Le 5.4 définit les rubriques de notification spécifiques au service de l'identifiant.

5.4.2 Présentation de l'événement (informative)

Le service de l'identifiant spécifie les événements lorsque l'état de l'identifiant et les identifiants changent.

Les rubriques principales pour les changements d'état sont les suivantes:

- tns1:Credential/State/Enabled;
- tns1:Credential/State/AppViolation.

Les rubriques principales pour les notifications de changement de configuration sont les suivantes:

- tns1:Configuration/Credential/Changed;
- tns1:Configuration/Credential/Removed.

5.4.3 Changements d'état

5.4.3.1 Généralités

Le dispositif doit fournir les événements de changement d'état afin d'informer les clients abonnés lorsque l'état de l'entité identifiant change. Le dispositif doit utiliser les rubriques définies en 5.4.3, ainsi que la description de message associée.

5.4.3.2 Identifiant

Chaque fois que l'état de l'identifiant (activé ou désactivé) est modifié, le dispositif doit fournir l'événement suivant (le champ Reason est l'un des champs répertoriés en 5.3.2.7, par exemple "pt:CredentialLost"):

```
Topic: tns1:Credential/State/Enabled
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="State"
                              Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="Reason"
                              Type="xs:string"/>
    <tt:SimpleItemDescription Name="ClientUpdated"
                              Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

ClientUpdated est défini sur true si le changement d'état a été engagé par le client. ClientUpdated est défini sur false si le dispositif a engagé le changement d'état (après trois saisies consécutives d'un mauvais code PIN, par exemple).

Le dispositif doit fournir l'événement suivant chaque fois que le dispositif antiretour est violé:

```
Topic: tns1:Credential/State/ApbViolation
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="ApbViolation"
                              Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="ClientUpdated"
                              Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

5.4.4 Modifications de configuration

5.4.4.1 Généralités

Chaque fois que les données de configuration sont modifiées, ajoutées ou supprimées, le dispositif doit fournir ces événements afin d'avertir les clients abonnés.

5.4.4.2 Identifiant

Chaque fois que les données de configuration d'un identifiant (y compris les informations d'identification et les profils d'accès de l'identifiant) sont modifiées ou qu'un identifiant est ajouté, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/Credential/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Chaque fois qu'un identifiant est supprimé, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/Credential/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

6 Service des règles d'accès

6.1 Généralités

Le service des règles d'accès définit le profil d'accès et ses politiques d'accès. Les identifiants sont associés à un profil d'accès pour l'autorisation des accès à une installation.

Le service des règles d'accès définit le moment (WHEN) et l'emplacement (WHERE) auxquels les identifiants ont accès. Chaque identifiant est associé à des profils d'accès, qui définissent chacun l'accès d'un groupe de personnes. Par exemple, les employés auront accès aux portes du bureau pendant les heures de bureau. Un autre exemple concerne l'accès à un appartement à toute heure par une famille. Chaque profil d'accès est composé d'un certain nombre de politiques d'accès, qui définissent chacun à quel moment un point d'accès peut être accessible.

Le service est flexible, ce qui permet d'octroyer un accès à un autre élément qu'un point d'accès (en définissant EntityType sur un QName autre qu'AccessPointInfo).

La Figure 3 représente les structures de données principales impliquées dans le service des règles d'accès.

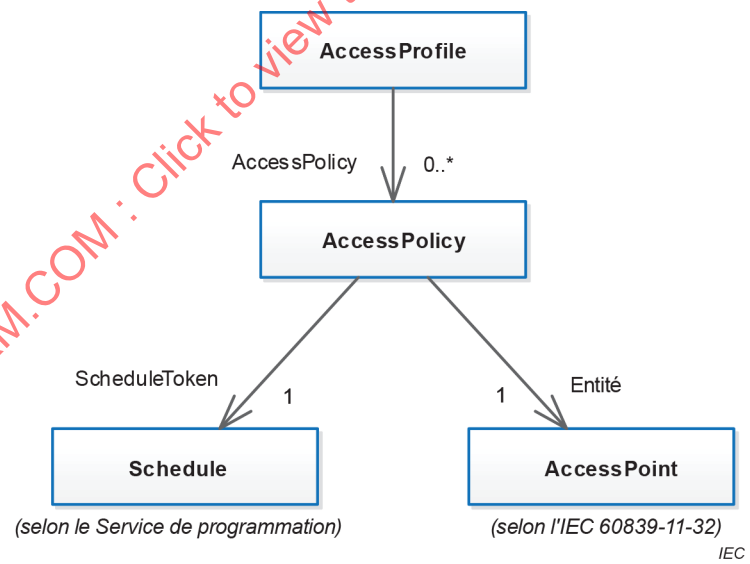


Figure 3 – Structures de données principales dans le service des règles d'accès

6.2 Fonctionnalités de service

6.2.1 Généralités

Un dispositif conforme ONVIF doit fournir les fonctionnalités de service de deux manières:

1) avec la méthode GetServices du service Dispositif (Device) lorsque IncludeCapability est défini sur true. Pour plus d'informations, se référer à la spécification principale de l'OVNI;
2) avec la méthode GetServiceCapabilities.

6.2.2 Structure de données ServiceCapabilities

Les fonctionnalités de service reflètent les fonctionnalités facultatives d'un service. Les informations sont statiques et ne changent pas pendant le fonctionnement du dispositif. Les fonctionnalités suivantes sont disponibles:

- **MaxLimit**
  indique le nombre maximal d'entités renvoyées par une demande Get<Entity>List ou Get<Entity>request unique. Le dispositif ne doit jamais renvoyer dans une même réponse un nombre d'entités supérieur à ce nombre;
- **MaxAccessProfiles**
  indique le nombre maximal de profils d'accès pris en charge par le dispositif;
- **MaxAccessPoliciesPerAccessProfile**
  indique le nombre maximal de politiques d'accès par profil d'accès pris en charge par le dispositif;
- **MultipleSchedulesPerAccessPointSupported**
  indique si plusieurs politiques d'accès peuvent ou non se référer au même point d'accès dans un profil d'accès;
- **ClientSuppliedTokenSupported**
  indique que le client est autorisé à fournir le jeton lors de la création des profils d'accès. Pour permettre l'utilisation de la commande SetAccessProfile, la valeur doit être définie sur true.

6.2.3 Commande GetServiceCapabilities

Cette opération renvoie les fonctionnalités du service des règles d'accès.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 23.

Tableau 23 – Commande GetServiceCapabilities

GetServiceCapabilities	Classe d'accès: PRE_AUTH
Nom de message	**Description**
GetServiceCapabilitiesRequest	*Ce message doit être vide.*
GetServiceCapabilitiesResponse	*Ce message contient:* • *"Capabilities": le message de réponse de la fonctionnalité répertorie les fonctionnalités du service des règles d'accès demandées, organisées selon une structure de fonctionnalités XML hiérarchique.* tar:ServiceCapabilities **Capabilities [1][1]**
Codes de défaut	**Description**
	Aucun défaut spécifique à cette commande.

6.3 Informations sur les profils d'accès

6.3.1 Généralités

Les profils d'accès définissent qui peut accéder à quel élément et à quel moment.

6.3.2 Structures de données

6.3.2.1 AccessProfileInfo

La structure AccessProfileInfo contient les informations de base concernant un profil d'accès. Le dispositif doit fournir les champs suivants pour chaque instance AccessProfile:

- **token**
  identificateur unique au service du profil d'accès;
- **Name**
  nom lisible par l'utilisateur. Il doit comporter 64 caractères au maximum.

Afin de fournir davantage d'informations, le dispositif peut inclure le champ facultatif suivant:

- **Description**
  description du profil d'accès lisible par l'utilisateur. Elle doit comporter 1 024 caractères au maximum.

6.3.2.2 AccessProfile

La structure AccessProfile contient des informations sur la collecte des politiques d'accès. Le dispositif doit inclure toutes les propriétés de la structure AccessProfileInfo, ainsi qu'une liste des politiques d'accès.

- **AccessPolicy**
  Liste des structures de politiques d'accès, qui définissent chacune le programme au cours duquel un point d'accès peut être accessible.

6.3.2.3 AccessPolicy

La politique d'accès est l'association d'un point d'accès et d'un programme. Il définit quand un point d'accès peut être accessible au moyen d'un profil d'accès qui contient cette politique d'accès. Si un profil d'accès contient plusieurs politiques d'accès qui spécifient différents programmes pour le même point d'accès, cela entraîne une union des programmes.

Un dispositif doit fournir les champs suivants pour chaque instance AccessPolicy:

- **ScheduleToken**
  référence au programme utilisé par la politique d'accès;
- **Entity**
  référence à l'entité utilisée par le moteur de règles. Le type d'entité peut être spécifié par le champ facultatif EntityType expliqué ci-dessous, mais est généralement un point d'accès.

Afin de fournir davantage d'informations, le dispositif peut inclure le champ facultatif suivant:

- **EntityType**
  type d'entité facultatif; si ce champ n'est pas défini, il convient de prendre pour hypothèse un type de point d'accès défini par le service de contrôle d'accès ONVIF. Cette information peut également être représentée par la valeur QName "tac:AccessPoint", où tac correspond à l'espace de nommage de [Spécification du service de contrôle d'accès]. Ce champ est fourni pour des extensions futures; il permet d'étendre une politique d'accès afin de couvrir d'autres types d'entités que les points d'accès.

6.3.3 Commande GetAccessProfileInfo

Cette opération demande une liste des éléments AccessProfileInfo qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 24.

Tableau 24 – Commande GetAccessProfileInfo

GetAccessProfileInfo	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetAccessProfileInfoRequest	*Ce message contient:* • *"Token": Jetons des éléments AccessProfileInfo à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetAccessProfileInfoResponse	*Ce message contient:* • *"AccessProfileInfo": Liste des éléments AccessProfileInfo.* tar:AccessProfileInfo **AccessProfileInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

6.3.4 Commande GetAccessProfileInfoList

Cette opération demande une liste de l'ensemble des éléments AccessProfileInfo fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se référer au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 25.

Tableau 25 – Commande GetAccessProfileInfoList

GetAccessProfileInfoList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetAccessProfileInfoListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAccessProfileInfoListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"AccessProfileInfo": Liste des éléments AccessProfileInfo.* xs:string **NextStartReference [0][1]** tar:AccessProfileInfo **AccessProfileInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

6.3.5 Commande GetAccessProfiles

Cette opération demande une liste des éléments AccessProfile qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 26.

Tableau 26 – Commande GetAccessProfiles

GetAccessProfiles	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetAccessProfileRequest	*Ce message contient:* • *"Token": Jetons des éléments AccessProfile à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetAccessProfileResponse	*Ce message contient:* • *"AccessProfile": Liste des éléments Access Profile.* tar:AccessProfile **AccessProfile [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

6.3.6 Commande GetAccessProfileList

Cette opération demande une liste de l'ensemble des éléments AccessProfile fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se référer au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 27.

Tableau 27 – Commande GetAccessProfileList

GetAccessProfileList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetAccessProfileListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAccessProfileListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"AccessProfile": Liste des éléments Access Profile.* xs:string **NextStartReference [0][1]** tar:AccessProfile **AccessProfile [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

6.3.7 Commande CreateAccessProfile

Cette opération crée le profil d'accès spécifié dans le dispositif.

Le champ Token de la structure AccessProfile doit être vide et le dispositif doit attribuer un jeton au profil d'accès. Le jeton attribué doit être renvoyé dans la réponse.

Si le client envoie une valeur dans le champ Token, le dispositif doit retourner le code de défaut générique InvalidArgVal.

Si plusieurs politiques d'accès au sein d'un profil d'accès spécifient différents programmes pour le même point d'accès, cela entraîne une union des programmes.

La Figure 4 représente un exemple de plusieurs programmes utilisés au même point d'accès. Chaque ligne de la figure ci-dessous correspond à une politique d'accès. Les première et deuxième lignes correspondent à deux politiques d'accès, qui utilisent différents programmes pour le même point d'accès.

Périodes du programme individuel					Politiques d'accès	
08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	**Programme**	**Point d'accès**
					Programme 1	Entrée Pièce A
					Programme 2	Entrée Pièce A
					Programme 1	Entrée Pièce B

Figure 4 – Plusieurs programmes par point d'accès

Etant donné que le Programme 1 et le Programme 2 définissent le moment auquel Entrée Pièce A peut être accessible, le détenteur d'identifiant constate l'union du Programme 1 et du Programme 2, comme le montre la première ligne de la Figure 5 ci-dessous.

Périodes de l'union des programmes					Politiques d'accès	
08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	**Programme**	**Point d'accès**
					Union du Programme 1 et du Programme 2	Entrée Pièce A
					Programme 1	Entrée Pièce B

Figure 5 – Résultat de l'union des programmes

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 28.

Tableau 28 – Commande CreateAccessProfile

CreateAccessProfile	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
CreateAccessProfileRequest	*Ce message contient:* • *"AccessProfile": Profil d'accès à créer.* tar:AccessProfile **AccessProfile [1][1]**
CreateAccessProfileResponse	*Ce message contient:* • *"Token": Jeton du profil d'accès créé.* pt:ReferenceToken **Token [1][1]**
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:MaxAccessProfiles	*L'espace n'est pas suffisant pour ajouter un nouvel AccessProfile (voir fonctionnalité MaxAccessProfiles).*
env:Sender ter:CapabilityViolated ter:MaxAccessPoliciesPerAccessProfile	*Le nombre d'AccessPolicies dans un AccessProfile est trop élevé (voir fonctionnalité MaxAccessPoliciesPerAccessProfile).*
env:Sender ter:CapabilityViolated ter:MultipleSchedulesPerAccessPointSupported	*Plusieurs AccessPoints ne sont pas pris en charge pour le même programme (voir fonctionnalité MultipleSchedulesPerAccessPointSupported).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*

6.3.8 Commande SetAccessProfile

Cette méthode permet de synchroniser le profil d'accès d'un client avec le dispositif.

Si aucun profil d'accès avec le jeton spécifié n'existe dans le dispositif, le profil d'accès est créé. Si un profil d'accès avec le jeton spécifié existe, le profil d'accès est modifié.

Un appel à cette méthode prend une structure AccessProfile comme paramètre d'entrée. Le champ Token du profil d'accès ne doit pas être vide.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité ClientSuppliedTokenSupported, il doit mettre en œuvre cette commande.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 29.

Tableau 29 – Commande SetAccessProfile

SetAccessProfile	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
SetAccessProfileRequest	*Ce message contient:* • *"AccessProfile": Elément AccessProfile à créer ou modifier.* tar:AccessProfile **AccessProfile [1][1]**
SetAccessProfileResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*Le dispositif ne prend pas en charge la fourniture du jeton par le client.*
env:Receiver ter:CapabilityViolated ter:MaxAccessProfiles	*L'espace n'est pas suffisant pour ajouter un nouvel AccessProfile (voir fonctionnalité MaxAccessProfiles).*
env:Sender ter:CapabilityViolated ter:MaxAccessPoliciesPerAccessProfile	*Le nombre d'AccessPolicies dans un AccessProfile est trop élevé (voir fonctionnalité MaxAccessPoliciesPerAccessProfile).*
env:Sender ter:CapabilityViolated ter:MultipleSchedulesPerAccessPointSupported	*Plusieurs AccessPoints ne sont pas pris en charge pour le même programme (voir fonctionnalité MultipleSchedulesPerAccessPointSupported).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*

6.3.9 Commande ModifyAccessProfile

Cette opération modifie le profil d'accès spécifié.

Le jeton du profil d'accès à modifier est spécifié dans le champ Token de la structure AccessProfile et ne doit pas être vide. Tous les autres champs de la structure doivent écraser les champs du profil d'accès spécifié.

Si plusieurs politiques d'accès spécifient différents programmes pour le même point d'accès, cela entraîne une union des programmes. Voir Figure 4 et Figure 5 ci-dessus.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 30.

Tableau 30 – Commande ModifyAccessProfile

ModifyAccessProfile	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
ModifyAccessProfileRequest	*Ce message contient:* • *"AccessProfile": Informations relatives au profil d'accès.* tar:AccessProfile **AccessProfile[1][1]**
ModifyAccessProfileResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Jeton du profil d'accès introuvable.*
env:Sender ter:CapabilityViolated ter:MaxAccessPoliciesPerAccessProfile	*Le nombre d'AccessPolicies dans un AccessProfile est trop élevé (voir fonctionnalité MaxAccessPoliciesPerAccessProfile).*
env:Sender ter:CapabilityViolated ter:MultipleSchedulesPerAccessPointSupported	*Plusieurs AccessPoints ne sont pas pris en charge pour le même programme (voir fonctionnalité MultipleSchedulesPerAccessPointSupported).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*

6.3.10 Commande DeleteAccessProfile

Cette opération supprime le profil d'accès spécifié.

Si le profil d'accès est supprimé, toutes les politiques d'accès associées au profil d'accès sont également supprimées.

S'il est associé à une ou plusieurs entités, certains dispositifs peuvent ne pas être en mesure de supprimer le profil d'accès et un défaut ReferenceInUse doit donc être généré.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 31.

Tableau 31 – Commande DeleteAccessProfile

DeleteAccessProfile	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
DeleteAccessProfileRequest	*Ce message contient:* • *"Token": Jeton du profil d'accès à supprimer.* pt:ReferenceToken **Token [1][1]**
DeleteAccessProfileResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Jeton du profil d'accès introuvable.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Suppression impossible. Le jeton du profil d'accès est en cours d'utilisation.*

6.4 Rubriques de notification

6.4.1 Généralités

Le 6.4 définit les rubriques de notification spécifiques au service des règles d'accès.

6.4.2 Présentation de l'événement (informative)

Le service de règles d'accès spécifie les événements lorsque les profils d'accès changent.

Les rubriques principales pour les notifications de changement de configuration sont les suivantes:

- tns1:Configuration/AccessProfile/Changed;
- tns1:Configuration/AccessProfile/Removed.

6.4.3 Modifications de configuration

6.4.3.1 Généralités

Chaque fois que les données de configuration sont modifiées, ajoutées ou supprimées, le dispositif doit fournir ces événements afin d'avertir les clients abonnés.

6.4.3.2 Profil d'accès

Chaque fois que les données de configuration d'un profil d'accès sont modifiées ou qu'un profil d'accès est ajouté, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/AccessProfile/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AccessProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Chaque fois qu'un profil d'accès est supprimé, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/AccessProfile/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AccessProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

7 Service de comportement d'authentification

7.1 Généralités

Ce service propose des commandes pour la gestion du comportement d'authentification et des niveaux de sécurité.

Les profils d'authentification permettent de définir la manière dont les détenteurs d'identifiant peuvent bénéficier d'un accès à un point d'accès en définissant quand sont exigés les différents niveaux de sécurité.

7.2 Exemple

L'exemple suivant utilise un programme (voir 8.4.2.2) qui définit les informations représentées à la Figure 6:

- plage horaire de 09:00 à 17:00 du lundi au vendredi (jours normaux);
- aucune plage horaire n'est définie pour les samedis et dimanches normaux;
- un groupe de jours particuliers pour les demi-journées de travail, avec une plage horaire de 09:00 à 13:00;
- un groupe de jours particuliers pour les jours fériés, avec une plage horaire de 09:00 à 17:00.

En outre, ce service définit quatre niveaux de sécurité: "Card" (Carte), "Card+PIN" (Carte+PIN), "Dual Card" (Double carte) et "No access" (Aucun accès) (voir 7.5.2.2).

En utilisant des contraintes de niveau de sécurité (voir 7.4.2.4), les quatre états de programmes peuvent être mappés à un niveau de sécurité et à un mode d'authentification comme indiqué dans le Tableau 32:

Tableau 32 – Exemple de mapping de l'état du programme et du niveau de sécurité

Jour particulier?	Plage horaire active?	Niveau de sécurité obtenu	Mode d'authentification obtenu
Non/Jour normal	Non	"Card+PIN"	pt:SingleCredential
Non/Jour normal	Oui	"Card"	pt:SingleCredential
"Demi-journées de travail"	Non	"Card+PIN"	pt:SingleCredential
"Demi-journées de travail"	Oui	"Card"	pt:SingleCredential
"Jours fériés"	Non	"No access"	(non pertinent)
"Jours fériés"	Oui	"Card"	pt:DualCredential

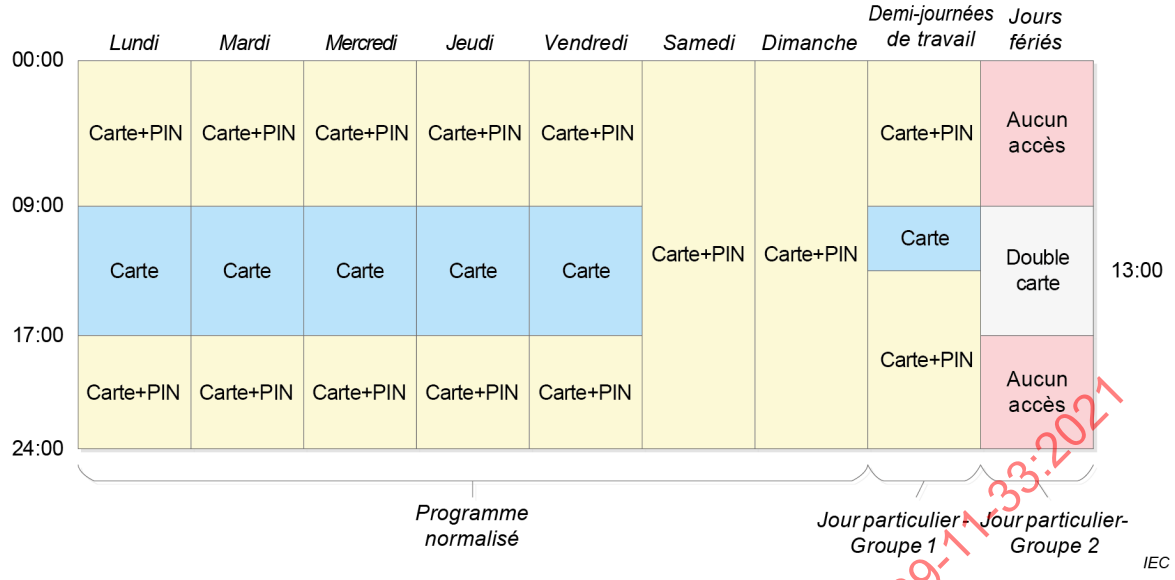


Figure 6 – Exemple de comportement d'authentification

7.3 Fonctionnalités de service

7.3.1 Généralités

Le dispositif doit fournir les fonctionnalités de service de deux manières:

1) avec la méthode GetServices du service Dispositif (Device) lorsque IncludeCapability est défini sur true. Pour plus d'informations, se référer à [Spécification principale];
2) avec la méthode GetServiceCapabilities.

7.3.2 Structure de données ServiceCapabilities

Les fonctionnalités de service reflètent les fonctionnalités facultatives d'un service. Les informations sont statiques et ne changent pas pendant le fonctionnement du dispositif. Les fonctionnalités suivantes sont disponibles:

- **MaxLimit**
  indique le nombre maximal d'entités renvoyées par une demande Get<Entity>List ou Get<Entity>request unique. Le dispositif ne doit jamais renvoyer dans une même réponse un nombre d'entités supérieur à ce nombre;
- **MaxAuthenticationProfiles**
  indique le nombre maximal de profils d'authentification pris en charge par le dispositif. Le dispositif doit prendre en charge au moins un profil d'authentification;
- **MaxPoliciesPerAuthenticationProfile**
  indique le nombre maximal de politiques d'authentification par profil d'authentification pris en charge par le dispositif;
- **MaxSecurityLevels**
  indique le nombre maximal de niveaux de sécurité pris en charge par le dispositif. Le dispositif doit prendre en charge au moins un niveau de sécurité;
- **MaxRecognitionGroupsPerSecurityLevel**
  indique le nombre maximal de groupes de reconnaissance par niveau de sécurité pris en charge par le dispositif;
- **MaxRecognitionMethodsPerRecognitionGroup**
  indique le nombre maximal de méthodes de reconnaissance par groupe de reconnaissance pris en charge par le dispositif;

- **ClientSuppliedTokenSupported**
  indique que le client est autorisé à fournir le jeton lors de la création des profils d'authentification et des niveaux de sécurité. Pour permettre l'utilisation des commandes SetAuthenticationProfile et SetSecurityLevel, la valeur doit être définie sur true.

7.3.3 Commande GetServiceCapabilities

Cette opération renvoie les fonctionnalités du service de comportement d'authentification.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 33.

Tableau 33 – Commande GetServiceCapabilities

GetServiceCapabilities	Classe d'accès: PRE_AUTH
Nom de message	**Description**
GetServiceCapabilitiesRequest	*Ce message doit être vide.*
GetServiceCapabilitiesResponse	*Ce message contient:* • *"Capabilities": le message de réponse de la fonctionnalité répertorie les fonctionnalités du service de comportement d'authentification demandées, organisées selon une structure de fonctionnalités XML hiérarchique.* tab:ServiceCapabilities **Capabilities [1][1]**
Codes de défaut	**Description**
	Aucun défaut spécifique à cette commande.

7.4 Informations sur le profil d'authentification

7.4.1 Généralités

Les profils d'authentification sont utilisés pour définir le comportement d'authentification d'un type de point d'accès. Par exemple, tous les points d'accès sont configurés pour exiger un accès par carte pendant les heures de bureau, un accès par carte+PIN la nuit et aucun accès pendant les vacances.

Le comportement d'authentification d'un type de point d'accès est défini en associant des niveaux de sécurité aux programmes. Lorsque le programme est actif, le niveau de sécurité spécifié est exigé.

Si un certain point dans le temps n'est couvert par aucun programme, le point d'accès est défini sur le niveau de sécurité par défaut.

La Figure 7 représente les objets connexes d'un profil d'authentification.

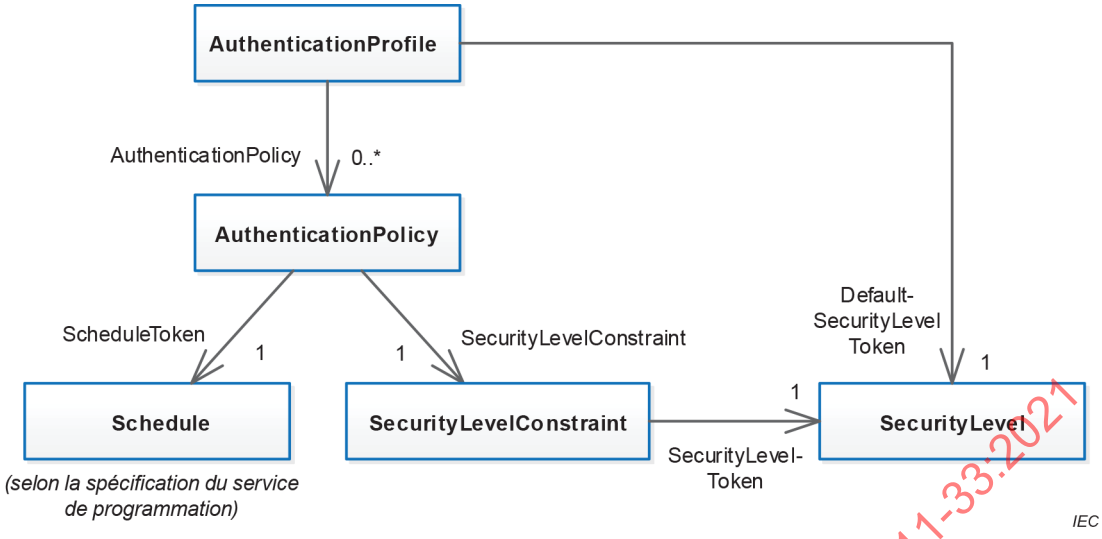


Figure 7 – Objets connexes d'un profil d'authentification

7.4.2 Structures de données

7.4.2.1 AuthenticationProfileInfo

La structure AuthenticationProfileInfo contient des informations sur une instance de profil d'authentification spécifique.

Le dispositif doit fournir les champs suivants pour chaque instance AuthenticationProfileInfo:

- **token**
  identificateur unique au service du profil d'authentification;
- **Name**
  nom descriptif, tel que "Portes d'entrée – entrée". Il doit comporter 64 caractères au maximum.

Afin de fournir davantage d'informations, le dispositif peut inclure le champ facultatif suivant:

- **Description**
  description du profil d'authentification lisible par l'utilisateur. Elle doit comporter 1 024 caractères au maximum.

7.4.2.2 AuthenticationProfile

La structure AuthenticationProfile doit inclure l'ensemble des propriétés de la structure AuthenticationProfileInfo, un niveau de sécurité par défaut, un mode d'authentification, ainsi qu'une liste des instances AuthenticationPolicy.

Le dispositif doit fournir les champs suivants pour chaque instance AuthenticationProfile:

- **DefaultSecurityLevelToken**
  niveau de sécurité par défaut utilisé si aucune des politiques d'authentification ne comporte de programme qui couvre la durée d'accès (ou si aucune politique d'authentification n'est définie);
- **AuthenticationPolicy**
  chaque politique d'authentification associe un niveau de sécurité à un programme (au cours duquel le niveau de sécurité spécifié est exigé au point d'accès).

7.4.2.3 AuthenticationPolicy

La politique d'authentification est l'association d'un niveau de sécurité et d'un programme. Elle définit à quel moment un certain niveau de sécurité doit attribuer un accès au détenteur d'un identifiant. Chaque niveau de sécurité se voit attribuer une priorité unique. Si les politiques d'authentification comportent des programmes qui se chevauchent, le niveau de sécurité avec la priorité la plus élevée est utilisé.

Le dispositif doit fournir les champs suivants pour chaque instance AuthenticationPolicy:

- **ScheduleToken**
  référence au programme utilisé par la politique d'authentification;
- **SecurityLevelConstraint**
  liste des structures de contraintes de niveau de sécurité qui définissent les conditions du niveau de sécurité à utiliser.

  Au moins une contrainte de niveau de sécurité doit être spécifiée.

7.4.2.4 SecurityLevelConstraint

Cette structure définit quel niveau de sécurité il convient d'activer selon l'état du programme. L'état d'un programme comporte deux valeurs booléennes qui correspondent à quatre états:

- le programme normalisé est actif et il ne s'agit pas d'un jour particulier;
- le programme normalisé n'est pas actif et il ne s'agit pas d'un jour particulier;
- il s'agit d'un jour particulier et une période définie dans le programme pour les jours particuliers est active;
- il s'agit d'un jour particulier, mais aucune période définie dans le programme pour les jours particuliers n'est active.

Si l'état du programme correspond aux paramètres ActiveRegularSchedule et ActiveSpecialDaySchedule de cette structure, le niveau de sécurité spécifié est utilisé.

Noter que si le dispositif ne prend pas en charge les jours particuliers, la valeur du champ ActiveSpecialDaySchedule n'est pas prise en compte.

Le dispositif doit fournir les champs suivants pour chaque instance SecurityLevelConstraint:

- **ActiveRegularSchedule**
  correspond au champ Active de la structure ScheduleState dans [Spécification du service de programmation];
- **ActiveSpecialDaySchedule**
  correspond au champ SpecialDay de la structure ScheduleState dans [Spécification du service de programmation]. Ce champ n'est pas pris en compte si le dispositif ne prend pas en charge les jours particuliers;
- **AuthenticationMode**
  définit le mode d'authentification. Les modes d'authentification qui commencent par le préfixe pt: sont réservés à la définition des modes d'authentification spécifiques à l'ONVIF. Pour les modes d'authentification personnalisés, il est permis d'utiliser du texte libre. Les modes d'authentification suivants sont définis par l'ONVIF:
  - pt:SingleCredential — mode normal, où un seul détenteur d'identifiant doit se voir attribuer un accès;
  - pt:DualCredential — deux détenteurs d'identifiants sont exigés pour obtenir l'accès.
- **SecurityLevelToken**
  référence au niveau de sécurité utilisé par la politique d'authentification.

7.4.3 Commande GetAuthenticationProfileInfo

Cette opération demande une liste des éléments AuthenticationProfileInfo qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 34.

Tableau 34 – Commande GetAuthenticationProfileInfo

GetAuthenticationProfileInfo	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetAuthenticationProfileInfoRequest	*Ce message contient:* • *"Token": Jetons des éléments AuthenticationProfileInfo à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetAuthenticationProfileInfoResponse	*Ce message contient:* • *"AuthenticationProfileInfo": Liste des éléments AuthenticationProfileInfo.* tab:AuthenticationProfileInfo **AuthenticationProfileInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

7.4.4 Commande GetAuthenticationProfileInfoList

Cette opération demande une liste de l'ensemble des éléments AuthenticationProfileInfo fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se référer au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 35.

Tableau 35 – Commande GetAuthenticationProfileInfoList

GetAuthenticationProfileInfoList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetAuthenticationProfileInfoListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAuthenticationProfileInfoListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"AuthenticationProfileInfo": Liste des éléments AuthenticationProfileInfo.* xs:string **NextStartReference [0][1]** tab:AuthenticationProfileInfo **AuthenticationProfileInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

7.4.5 Commande GetAuthenticationProfiles

Cette opération demande une liste des éléments AuthenticationProfile qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 36.

Tableau 36 – Commande GetAuthenticationProfiles

GetAuthenticationProfiles	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetAuthenticationProfileRequest	*Ce message contient:* • *"Token": Jetons des éléments AuthenticationProfile à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetAuthenticationProfileResponse	*Ce message contient:* • *"AuthenticationProfile": Liste des éléments AuthenticationProfile.* tab:AuthenticationProfile **AuthenticationProfile [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

7.4.6 Commande GetAuthenticationProfileList

Cette opération demande une liste de l'ensemble des éléments AuthenticationProfile fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 37.

Tableau 37 – Commande GetAuthenticationProfileList

GetAuthenticationProfileList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetAuthenticationProfileListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetAuthenticationProfileListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"AuthenticationProfile": Liste des éléments AuthenticationProfile.* xs:string **NextStartReference [0][1]** tab:AuthenticationProfile **AuthenticationProfile [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

7.4.7 Commande CreateAuthenticationProfile

Cette opération crée le profil d'authentification spécifié dans le dispositif.

Le champ Token de la structure AuthenticationProfile doit être vide et le dispositif doit attribuer un jeton au profil d'authentification. Le jeton attribué doit être renvoyé dans la réponse.

Si le client envoie une valeur dans le champ Token, le dispositif doit retourner le code de défaut générique InvalidArgVal.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 38.

Tableau 38 – Commande CreateAuthenticationProfile

CreateAuthenticationProfile	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
CreateAuthenticationProfileRequest	*Ce message contient:* • *"AuthenticationProfile": Profil d'authentification à créer.* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
CreateAuthenticationProfileResponse	*Ce message contient:* • *"Token": Jeton du profil d'authentification créé.* pt:ReferenceToken **Token [1][1]**
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:MaxAuthenticationProfiles	*L'espace n'est pas suffisant pour ajouter un nouveau profil d'authentification (voir fonctionnalité MaxAuthenticationProfiles).*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*Le nombre d'entités AuthenticationPolicy référencées dans AuthenticationProfile est trop élevé (voir fonctionnalité MaxPoliciesPerAuthenticationProfile).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*

7.4.8 Commande SetAuthenticationProfile

Cette méthode permet de synchroniser le profil d'authentification d'un client avec le dispositif.

Si aucun profil d'authentification avec le jeton spécifié n'existe dans le dispositif, le profil d'authentification est créé. Si un profil d'authentification avec le jeton spécifié existe, le profil d'authentification est modifié.

Un appel à cette méthode prend une structure AuthenticationProfile comme paramètre d'entrée. Le champ Token de la structure AuthenticationProfile ne doit pas être vide.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité ClientSuppliedTokenSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 39.

Tableau 39 – Commande SetAuthenticationProfile

SetAuthenticationProfile	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
SetAuthenticationProfileRequest	*Ce message contient:* • *"AuthenticationProfile": Elément AuthenticationProfile à créer ou modifier.* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
SetAuthenticationProfileResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*Le dispositif ne prend pas en charge la fourniture du jeton par le client.*
env:Receiver ter:CapabilityViolated ter:MaxAuthenticationProfiles	*L'espace n'est pas suffisant pour ajouter un nouvel AuthenticationProfile (voir fonctionnalité MaxAuthenticationProfiles).*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*Le nombre d'entités AuthenticationPolicy référencées dans AuthenticationProfile est trop élevé (voir fonctionnalité MaxPoliciesPerAuthenticationProfile).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*
env:Sender ter:CapabilityViolated ter:SupportedSecurityLevels	*Le profil d'authentification mis à jour contient des niveaux de sécurité qui ne sont pas pris en charge par un point d'accès qui fait référence à ce profil d'authentification (certains dispositifs peuvent ne pas valider ce profil).*

7.4.9 Commande ModifyAuthenticationProfile

Cette opération modifie le profil d'authentification spécifié.

Le jeton du profil d'authentification à modifier est spécifié dans le champ Token de la structure AuthenticationProfile et ne doit pas être vide. Tous les autres champs de la structure doivent écraser les champs du profil d'authentification spécifié.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 40.

Tableau 40 – Commande ModifyAuthenticationProfile

ModifyAuthenticationProfile	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
ModifyAuthenticationProfileRequest	*Ce message contient:* • *"AuthenticationProfile": Elément AuthenticationProfile à modifier.* tab:AuthenticationProfile **AuthenticationProfile [1][1]**
ModifyAuthenticationProfileResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton spécifié est introuvable.*
env:Sender ter:CapabilityViolated ter:MaxPoliciesPerAuthenticationProfile	*Le nombre d'entités AuthenticationPolicy référencées dans AuthenticationProfile est trop élevé (voir fonctionnalité MaxPoliciesPerAuthenticationProfile).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*
env:Sender ter:CapabilityViolated ter:SupportedSecurityLevels	*Le profil d'authentification mis à jour contient des niveaux de sécurité qui ne sont pas pris en charge par un point d'accès qui fait référence à ce profil d'authentification (certains dispositifs peuvent ne pas valider ce profil).*

7.4.10 Commande DeleteAuthenticationProfile

Cette opération supprime le profil d'authentification spécifié.

Si le profil d'authentification est supprimé, toutes les politiques d'authentification associées au profil d'authentification sont également supprimées.

S'il est associé à une ou plusieurs entités, certains dispositifs peuvent ne pas être en mesure de supprimer le profil d'authentification et un défaut ReferenceInUse doit donc être généré.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 41.

Tableau 41 – Commande DeleteAuthenticationProfile

DeleteAuthenticationProfile	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
DeleteAuthenticationProfileRequest	*Ce message contient:* • *"Token": Jeton de l'élément AuthenticationProfile à supprimer.* pt:ReferenceToken **Token [1][1]**
DeleteAuthenticationProfileResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Jeton du profil d'authentification introuvable.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Suppression impossible. Le jeton du profil d'authentification est en cours d'utilisation.*

7.5 Informations sur le niveau de sécurité

7.5.1 Généralités

Les niveaux de sécurité sont définis comme des méthodes de reconnaissance individuelles, des combinaisons de méthodes de reconnaissance (à l'aide de l'opérateur logique AND ou OR) voire aucune méthode de reconnaissance (ouvert). Des noms explicatifs sont attribués aux niveaux de sécurité, tels que "Card", "Card+PIN", "Fingerprint or Iris", "Open".

Les facteurs d'authentification possibles sont Card, PIN, Fingerprint, Face, Iris, Vein, Palm et REX (qui n'est pas câblé afin de permettre un déverrouillage direct, mais où l'appareil peut accomplir certaines actions, comme prendre une décision).

La Figure 8 représente les objets connexes d'un niveau de sécurité.

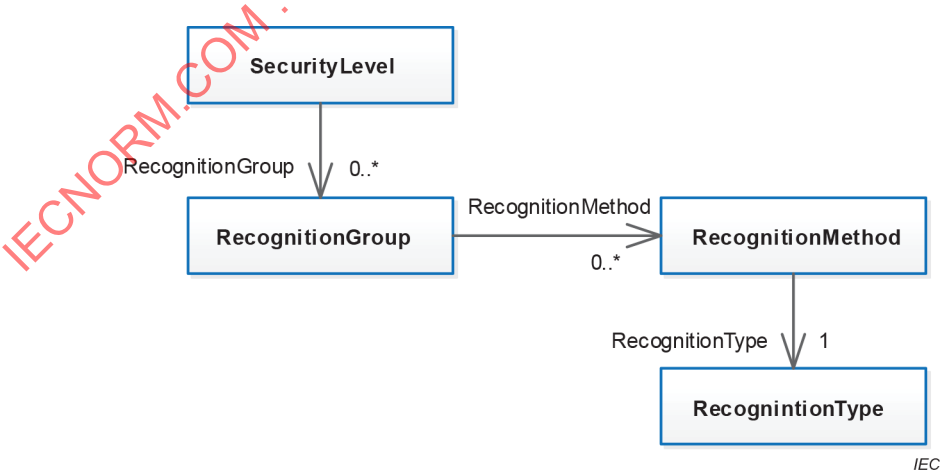


Figure 8 – Objets connexes d'un niveau de sécurité

7.5.2 Structures de données

7.5.2.1 SecurityLevelInfo

La structure SecurityLevelInfo contient des informations sur une instance de niveau de sécurité spécifique.

Le dispositif doit fournir les champs suivants pour chaque instance SecurityLevelInfo:

- **token**
  identificateur unique au service du niveau de sécurité;
- **Name**
  nom lisible par l'utilisateur. Il doit comporter 64 caractères au maximum;
- **Priority**
  un nombre élevé indique que le niveau de sécurité est jugé comme étant plus sûr que les niveaux de sécurité avec des priorités moindres. La priorité est utilisée lorsqu'un profil d'authentification comporte des programmes qui se chevauchent avec différents niveaux de sécurité. Lors de l'accès à un point d'accès, les politiques d'authentification sont parcourues par ordre de priorité (en commençant par la priorité la plus élevée). Lorsqu'un programme qui couvre la durée d'accès est trouvé, le niveau de sécurité associé est utilisé et le traitement s'arrête. Deux niveaux de sécurité ne peuvent pas avoir la même priorité.

Afin de fournir davantage d'informations, le dispositif peut inclure le champ facultatif suivant:

- **Description**
  description du niveau de sécurité lisible par l'utilisateur. Elle doit comporter 1 024 caractères au maximum.

7.5.2.2 SecurityLevel

La structure SecurityLevel doit inclure l'ensemble des propriétés de la structure SecurityLevelInfo, ainsi qu'un ensemble de groupes de reconnaissance.

Les groupes de reconnaissance sont utilisés pour définir un opérateur logique OR entre les groupes. Chaque groupe de reconnaissance se compose d'une ou de plusieurs méthodes de reconnaissance.

Exemple: Un groupe de reconnaissance contient les méthodes de reconnaissance pt:Card et pt:Fingerprint. Un autre groupe contient les méthodes de reconnaissance pt:Card et pt:Face. Le point d'accès nécessite alors Card+Fingerprint (Carte+Empreinte digitale) ou Card+Face (Carte+Visage).

L'ordre des méthodes de reconnaissance demandées peut être défini au niveau du point d'accès à l'aide du champ Order de la structure de la méthode de reconnaissance.

L'exemple de la Figure 9 représente trois niveaux de sécurité. Le premier niveau ne comporte pas de groupe de reconnaissance, et le point d'accès n'exige aucune authentification lorsqu'il est actif. Le second niveau est décrit ci-dessus. Le troisième niveau définit un groupe de reconnaissance, mais sans méthodes de reconnaissance, ce qui a le même effet que lorsque le point d'accès est désactivé.

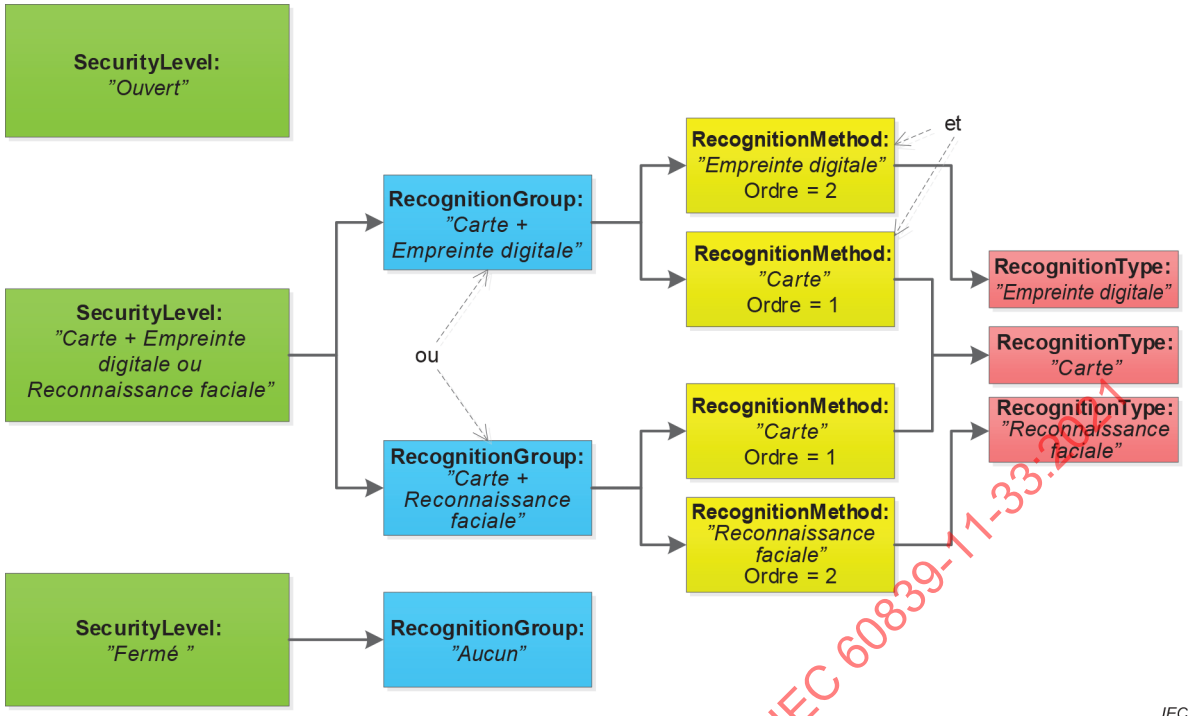


Figure 9 – Exemples de niveaux de sécurité

Un dispositif doit fournir le champ suivant pour chaque instance SecurityLevel:

- **RecognitionGroup**
  les groupes de reconnaissance sont utilisés pour définir un opérateur logique OR entre les groupes. Chaque groupe de reconnaissance se compose d'une ou de plusieurs méthodes de reconnaissance.

  L'absence de groupes de reconnaissance signifie que le point d'accès est ouvert.

7.5.2.3 RecognitionGroup

Un dispositif doit fournir le champ suivant pour chaque instance RecognitionGroup:

- **RecognitionMethod**
  Liste des méthodes de reconnaissance à demander au niveau du point d'accès.

7.5.2.4 RecognitionMethod

La reconnaissance est l'action qui identifie les utilisateurs autorisés qui demandent un accès, en comparant les données d'identifiant présentées et les données d'identifiant enregistrées. Une méthode de reconnaissance est mémorisée, biométrique ou contenue au sein d'un identifiant physique. Un type de reconnaissance est soit une méthode de reconnaissance, soit une entrée physique comme un bouton de demande de sortie.

Un dispositif doit fournir les champs suivants pour chaque instance RecognitionMethod:

- **RecognitionType**
  type de reconnaissance demandé. Le type est texte. Peut être l'un des types ONVIF réservés suivants ou un type personnalisé:
  - pt:Card la carte est utilisée comme méthode de reconnaissance;
  - pt:PIN le code PIN est utilisé comme méthode de reconnaissance;
  - pt:Fingerprint l'analyse des empreintes digitales est utilisée comme méthode de reconnaissance;

- pt:Face la reconnaissance faciale est utilisée comme méthode de reconnaissance;
- pt:Iris la reconnaissance de l'iris est utilisée comme méthode de reconnaissance;
- pt:Vein la reconnaissance veineuse est utilisée comme méthode de reconnaissance;
- pt:Palm la reconnaissance de la paume est utilisée comme méthode de reconnaissance;
- pt:REX un bouton de demande de sortie est utilisé pour autoriser l'accès.

- **Order**
  la valeur de l'ordre définit quand cette méthode de reconnaissance est demandée par rapport aux autres méthodes de reconnaissance du même niveau de sécurité. Un nombre inférieur indique que la méthode de reconnaissance est demandée avant les méthodes de reconnaissance qui comportent un nombre supérieur.

7.5.3 Commande GetSecurityLevelInfo

Cette opération demande une liste des éléments SecurityLevelInfo qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 42.

Tableau 42 – Commande GetSecurityLevelInfo

GetSecurityLevelInfo	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSecurityLevelInfoRequest	*Ce message contient:* • *"Token": Jetons des éléments SecurityLevelInfo à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetSecurityLevelInfoResponse	*Ce message contient:* • *"SecurityLevelInfo": Liste des éléments SecurityLevelInfo.* tab:SecurityLevelInfo **SecurityLevelInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

7.5.4 Commande GetSecurityLevelInfoList

Cette opération demande une liste de l'ensemble des éléments SecurityLevelInfo fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 43.

Tableau 43 – Commande GetSecurityLevelInfoList

GetSecurityLevelInfoList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSecurityLevelInfoListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSecurityLevelInfoListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"SecurityLevelInfo": Liste des éléments SecurityLevelInfo.* xs:string **NextStartReference [0][1]** tab:SecurityLevelInfo **SecurityLevelInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

7.5.5 Commande GetSecurityLevels

Cette opération demande une liste des éléments SecurityLevel qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 44.

Tableau 44 – Commande GetSecurityLevels

GetSecurityLevels	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSecurityLevelsRequest	*Ce message contient:* • *"Token": Jetons des éléments SecurityLevel à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetSecurityLevelsResponse	*Ce message contient:* • *"SecurityLevel": Liste des éléments SecurityLevel.* tab:SecurityLevel **SecurityLevel [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

7.5.6 Commande GetSecurityLevelList

Cette opération demande une liste de l'ensemble des éléments SecurityLevel fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 45.

Tableau 45 – Commande GetSecurityLevelList

GetSecurityLevelList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSecurityLevelListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSecurityLevelListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"SecurityLevel": Liste des éléments SecurityLevel.* xs:string **NextStartReference [0][1]** tab:SecurityLevel **SecurityLevel [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

7.5.7 Commande CreateSecurityLevel

Cette opération crée le niveau de sécurité spécifié dans le dispositif.

Le champ Token de la structure SecurityLevel doit être vide et le dispositif doit attribuer un jeton au niveau de sécurité. Le jeton attribué doit être renvoyé dans la réponse.

Si le client envoie une valeur dans le champ Token, le dispositif doit retourner le code de défaut générique InvalidArgVal.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 46.

Tableau 46 – Commande CreateSecurityLevel

CreateSecurityLevel	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
CreateSecurityLevelRequest	*Ce message contient:* • *"SecurityLevel": Niveau de sécurité à créer.* tab:SecurityLevel **SecurityLevel [1][1]**
CreateSecurityLevelResponse	*Ce message contient:* • *"Token": Jeton du niveau de sécurité créé.* pt:ReferenceToken **Token [1][1]**
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:MaxSecurityLevels	*L'espace n'est pas suffisant pour ajouter de nouveaux éléments SecurityLevel (voir fonctionnalité MaxSecurityLevels).*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*Le nombre de groupes de méthodes de reconnaissance dans un SecurityLevel est trop élevé (voir fonctionnalité MaxRecognitionGroupsPerSecurityLevel).*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognitionGroup	*Le nombre de méthodes de reconnaissance dans un groupe de reconnaissance est trop élevé (voir fonctionnalité MaxRecognitionMethodsPerRecognitionGroup).*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*La priorité du niveau de sécurité est déjà utilisée par un autre niveau de sécurité.*

7.5.8 Commande SetSecurityLevel

Cette méthode permet de synchroniser un niveau de sécurité d'un client avec le dispositif.

Si aucun niveau de sécurité avec le jeton spécifié n'existe dans le dispositif, le niveau de sécurité est créé. Si un niveau de sécurité avec le jeton spécifié existe, le niveau de sécurité est modifié.

Un appel à cette méthode prend une structure SecurityLevel comme paramètre d'entrée. Le champ Token de la structure SecurityLevel ne doit pas être vide.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité ClientSuppliedTokenSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 47.

Tableau 47 – Commande SetSecurityLevel

SetSecurityLevel	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
SetSecurityLevelRequest	*Ce message contient:* • *"SecurityLevel": Niveau de sécurité à créer ou à modifier.* tab:SecurityLevel **SecurityLevel [1][1]**
SetSecurityLevelResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*Le dispositif ne prend pas en charge la fourniture du jeton par le client.*
env:Receiver ter:CapabilityViolated ter:MaxSecurityLevels	*L'espace n'est pas suffisant pour ajouter de nouveaux éléments SecurityLevel (voir fonctionnalité MaxSecurityLevels).*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*Le nombre de groupes de méthodes de reconnaissance dans un SecurityLevel est trop élevé (voir fonctionnalité MaxRecognitionGroupsPerSecurityLevel).*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognitionGroup	*Le nombre de méthodes de reconnaissance dans un groupe de méthodes de reconnaissance est trop élevé (voir fonctionnalité MaxRecognitionMethodsPerRecognitionGroup).*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*La priorité du niveau de sécurité est déjà utilisée par un autre niveau de sécurité.*
env:Sender ter:InvalidArgVal ter:MissingToken	*Le jeton de l'élément niveau de sécurité doit être spécifié.*

7.5.9 Commande ModifySecurityLevel

Cette opération modifie le niveau de sécurité spécifié.

Le jeton du niveau de sécurité à modifier est spécifié dans le champ Token de la structure SecurityLevel et ne doit pas être vide. Tous les autres champs de la structure doivent écraser les champs du niveau de sécurité spécifié.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 48.

Tableau 48 – Commande ModifySecurityLevel

ModifySecurityLevel	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
ModifySecurityLevelRequest	*Ce message contient:* • *"SecurityLevel": Niveau de sécurité à modifier.* tab:SecurityLevel **SecurityLevel [1][1]**
ModifySecurityLevelResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton spécifié est introuvable.*
env:Sender ter:CapabilityViolated ter:MaxRecognitionGroupsPerSecurityLevel	*Le nombre de groupes de méthodes de reconnaissance dans un SecurityLevel est trop élevé (voir fonctionnalité MaxRecognitionGroupsPerSecurityLevel).*
env:Sender ter:CapabilityViolated ter:MaxRecognitionMethodsPerRecognitionGroup	*Le nombre de méthodes de reconnaissance dans un groupe de méthodes de reconnaissance est trop élevé (voir fonctionnalité MaxRecognitionMethodsPerRecognitionGroup).*
env:Sender ter:InvalidArgVal ter:DuplicatePriority	*La priorité du niveau de sécurité est déjà utilisée par un autre niveau de sécurité.*

7.5.10 Commande DeleteSecurityLevel

Cette méthode permet de supprimer le niveau de sécurité spécifié.

S'il est associé à une ou plusieurs entités, certains dispositifs peuvent ne pas être en mesure de supprimer le niveau de sécurité et un défaut ReferenceInUse doit donc être généré.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 49.

Tableau 49 – Commande DeleteSecurityLevel

DeleteSecurityLevel	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
DeleteSecurityLevelRequest	*Ce message contient:* • *"Token": Jeton de l'élément SecurityLevel à supprimer.* pt:ReferenceToken **Token [1][1]**
DeleteSecurityLevelResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Jeton du niveau de sécurité introuvable.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Suppression impossible. Le jeton de niveau de sécurité est en cours d'utilisation.*

7.6 Rubriques de notification

7.6.1 Généralités

Le 7.6 définit les rubriques de notification spécifiques au service de comportement d'authentification.

7.6.2 Présentation de l'événement (informative)

Le service de comportement d'authentification spécifie les événements lorsque les profils d'authentification ou les niveaux de sécurité changent.

Les rubriques principales pour les notifications de changement de configuration sont les suivantes:

- tns1:Configuration/AuthenticationProfile/Changed;
- tns1:Configuration/AuthenticationProfile/Removed;
- tns1:Configuration/SecurityLevel/Changed;
- tns1:Configuration/SecurityLevel/Removed.

7.6.3 Modifications de configuration

7.6.3.1 Généralités

Chaque fois que les données de configuration sont modifiées, ajoutées ou supprimées, le dispositif doit fournir ces événements afin d'avertir les clients abonnés.

7.6.3.2 Profil d'authentification

Chaque fois que les données de configuration d'un profil d'authentification sont modifiées ou qu'un profil d'authentification est ajouté, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/AuthenticationProfile/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AuthenticationProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Chaque fois qu'un profil d'authentification est supprimé, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/AuthenticationProfile/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="AuthenticationProfileToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

7.6.3.3 Niveau de sécurité

Chaque fois que les données de configuration d'un niveau de sécurité sont modifiées ou qu'un niveau de sécurité est ajouté, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/SecurityLevel/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SecurityLevelToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Chaque fois qu'un niveau de sécurité est supprimé, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/SecurityLevel/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SecurityLevelToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```


8 Service de programmation

8.1 Généralités

Le service de programmation fournit des fonctions pour la gestion des programmes et des jours particuliers.

Par exemple, les programmes peuvent être utilisés pour définir quand est octroyé l'accès ou quand il convient qu'une caméra effectue l'enregistrement.

Le service de programmation prend en charge les programmes à la fois en heure locale et en heure UTC. En général, un programme est exprimé en heure locale.

La Figure 10 représente les structures de données principales impliquées dans le service de programmation.

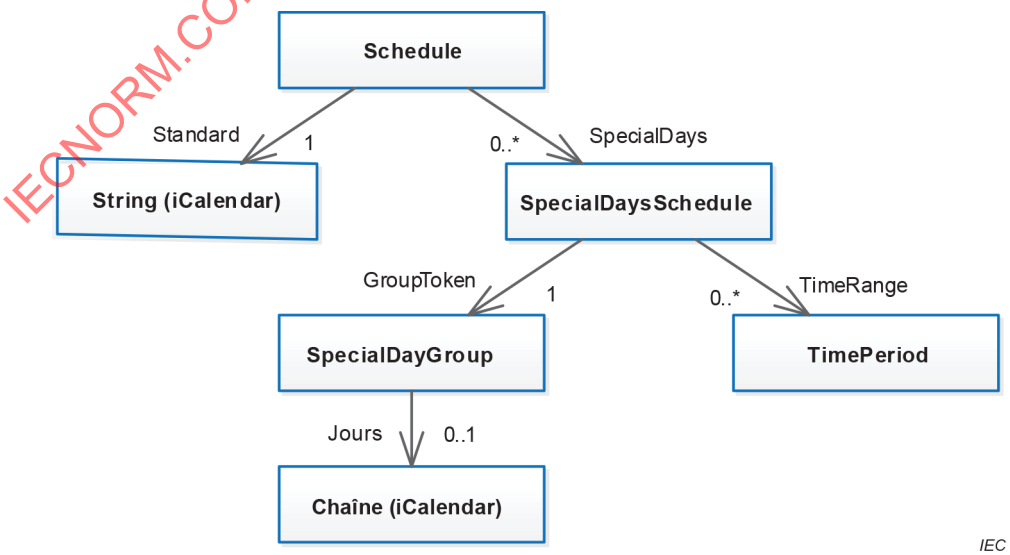


Figure 10 – Structures de données principales dans le service de programmation

Un groupe de jours particuliers peut être utilisé pour définir les heures d'accès à l'installation pendant les vacances. Un autre groupe de jours particuliers peut être utilisé pour définir les heures d'accès à l'installation pendant les demi-journées de travail.

Les groupes de jours particuliers sont généralement réutilisés par différents programmes. Par exemple:

un système EACS est partagé par deux sites, l'un se situant en Suède et l'autre en Pologne. Toutes les vacances polonaises sont regroupées au sein d'un groupe de jours particuliers appelé "Vacances polonaises". Toutes les vacances suédoises sont regroupées au sein d'un groupe de jours particuliers appelé "Vacances suédoises". De plus, la Suède a des demi-journées de travail avant certains jours fériés, de sorte qu'un autre groupe est créé sous le nom "Demi-journées suédoises;

un programme appelé "Horaire des employés polonais" est créé; il définit l'accès au bureau pour les employés polonais pendant tous les jours de la semaine, y compris le week-end (l'accès du lundi au vendredi de 07:00 à 20:00 et l'accès du samedi au dimanche de 10:00 à 14:00, par exemple). Le programme est lié au groupe "Vacances polonaises" qui définit l'accès au travail pendant ces jours (pas d'accès pendant les vacances, par exemple;

de même, un programme appelé "Horaire des employés suédois" est créé; il définit l'accès au bureau pour les employés suédois pendant tous les jours de la semaine, y compris le week-end. Le programme est lié à la fois au groupe "Vacances suédoises" (pas d'accès, par exemple) et au groupe "Demi-journées suédoises" (accès de 07:00 à 13:00, par exemple).

Les programmes et les jours particuliers sont définis par iCalendar vEvents. iCalendar est un format très flexible qui permet d'utiliser des types de périodicité (annuel, mensuel, hebdomadaire, quotidien). Pour plus d'informations, voir RFC 5545.

Pour expliquer brièvement le format iCalendar, prendre les exemples suivants (seules les parties des structures iCalendar sont présentées ci-dessous):

événement iCalendar qui commence à 20:00 le 25 novembre 2014 et se termine à 02:00 le 26 novembre 2014:

```
BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART:20141125T200000
DTEND:20141126T020000
END:VEVENT
END:VCALENDAR
```

événement iCalendar hebdomadaire qui se répète chaque mardi et jeudi de 14:00 à 16:00, dont la première occurrence est le 25 novembre 2014 et la dernière occurrence est le 9 décembre 2014 (5 occurrences):

```
BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART:20141125T140000
DTEND:20141125T160000
RRULE:FREQ=WEEKLY;COUNT=5;BYDAY=TU,TH
END:VEVENT
END:VCALENDAR
```

l'exemple suivant a le même effet que l'exemple précédent, mais la date de fin de la série est spécifiée en lieu et place du nombre d'occurrences:

```
BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART:20141125T140000
DTEND:20141125T160000
RRULE:FREQ=WEEKLY;UNTIL:20141209T140000;BYDAY=TU,TH
END:VEVENT
END:VCALENDAR
```

Seul un certain sous-ensemble de la spécification iCalendar doit être pris en charge afin d'obtenir un service utilisable. ServiceCapabilities reflète la prise en charge des fonctionnalités supplémentaires dans la spécification iCalendar.

8.2 Périodicité

8.2.1 Généralités

Lorsque la fonctionnalité ExtendedRecurrenceSupported est définie sur true, le format de périodicité iCalendar étendu est pris en charge. Les jours particuliers peuvent continuer à être utilisés, mais peuvent ne pas être nécessaires.

Il convient de définir la fonctionnalité ExtendedRecurrenceSupported sur false pour les dispositifs qui ne prennent pas en charge la périodicité iCalendar étendue. Les mots clés iCalendar "FREQ" et "BYDAY" sont utilisés pour définir des événements récurrents hebdomadaires qui se répètent plusieurs jours dans la semaine (voir exemples iCalendar ci-dessus). Lorsque la fonctionnalité ExtendedRecurrenceSupported est définie sur false, l'année de la date de début des événements récurrents doit être 1970 et le nombre d'occurrences (éventuellement la date de fin) ne doit pas être indiqué (c'est-à-dire les événements récurrents qui se répètent indéfiniment).

La Figure 11 représente l'effet des paramètres de la fonctionnalité ExtendedRecurrenceSupported pour le programme et les jours particuliers respectivement.

	ExtendedRecurrenceSupported	
	FALSE	**TRUE**
Programmes	Périodicité hebdomadaire	Périodicité étendue
Jours particuliers	Aucune périodicité autorisée	Périodicité étendue

Figure 11 – Matrice de prise en charge de la périodicité

En 8.2.2 et 8.2.3, la syntaxe ABNF doit être utilisée (voir RFC 5234). Les noms de règles situés à gauche du signe égal (dtstart, dtend et recur-rule-part) redéfinissent les règles décrites dans la RFC 5545. Les noms de règles situés à droite du signe égal (date-month, date-mday, time-hour, time-minute, time-second, bymolist, bywdaylist, bymodaylist, weekday et enddate) doivent être définis conformément à la RFC 5545.

8.2.2 Périodicité hebdomadaire

Seul le type de périodicité iCalendar hebdomadaire doit être pris en charge. Les événements non récurrents et les autres types récurrents ne sont pas pris en charge.

Le dispositif ne doit accepter que les événements récurrents qui commencent par l'année "1970" (le mois et le jour sont nécessaires pour correspondre au jour de la semaine de la périodicité).

```
dtstart = "DTSTART" ":" "1970" date-month date-mday
          "T" time-hour time-minute time-second
```

Le dispositif ne doit autoriser de dates de fin qu'avant minuit de la date de début spécifiée.

```
dtend = "DTEND" ":" "1970" date-month date-mday
        "T" time-hour time-minute time-second
```

où date-month et date-mday doivent être identiques à dtstart ou un jour plus tard, mais avec time-hour = "00", time-minute = "00" et time-second = "00".

Seules les heures locales sont prises en charge.

Le dispositif ne doit pas accepter un nombre d'occurrences ou une date de fin pour les événements récurrents. Le modèle de périodicité suivant est autorisé:

```
recur-rule-part = "WEEKLY" [ ";" "BYDAY" "=" (weekday *("," weekday)) ]
```

8.2.3 Périodicité étendue

Le dispositif doit prendre en charge le modèle de périodicité suivant:

```
recur-rule-part =
       (  ( "YEARLY" )
        / ( "YEARLY" ";" "BYMONTH" "=" bymolist )
        / ( "MONTHLY" )
        / ( "MONTHLY" ";" "BYDAY" "=" bywdaylist )
        / ( "MONTHLY" ";" "BYMONTHDAY" "=" bymodaylist )
        / ( "WEEKLY" )
        / ( "WEEKLY" ";" "BYDAY" "=" (weekday *("," weekday)) )
        / ( "DAILY" )
        / ( "HOURLY" )
        / ( "MINUTELY" )
        / ( "SECONDLY" ) )
       [ ";" "INTERVAL" "=" 1*DIGIT ]
       [ ";" "COUNT" "=" 1*DIGIT / ";" "UNTIL" "=" enddate ]
```

8.2.4 Périodicité de programme normalisé

Les occurrences d'une série d'événements récurrents peuvent être supprimées pour créer des exceptions aux événements récurrents.

La Figure 12 représente une série d'événements récurrents. Chaque barre verticale représente une journée de 24 h, où les zones vertes représentent des événements (périodes). Les événements se produisent chaque lundi, mais il y a une exception le quatrième lundi, où aucun événement ne se produit.

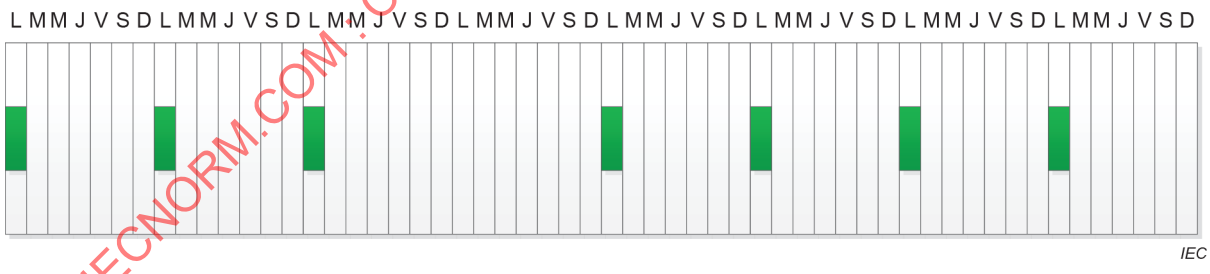


Figure 12 – Evénements récurrents avec une exception

8.2.5 Périodicité de jours particuliers

Contrairement aux exceptions, le service de programmation prend également en charge les jours particuliers. Des jours particuliers sont utilisés pour définir des comportements différents pendant les jours fériés ou les demi-journées de travail, par exemple. Des exemples de comportements différents incluent l'enregistrement vidéo à un débit de trames différent ou des temps d'accès différents à une installation. Les jours particuliers sont regroupés et portent des noms tels que "Vacances américaines", "Vacances britanniques", "Demi-journées suédoises".

La Figure 13 représente la même série d'événements récurrents, mais à défaut d'une exception, le quatrième lundi est traité comme un jour particulier avec un comportement différent.

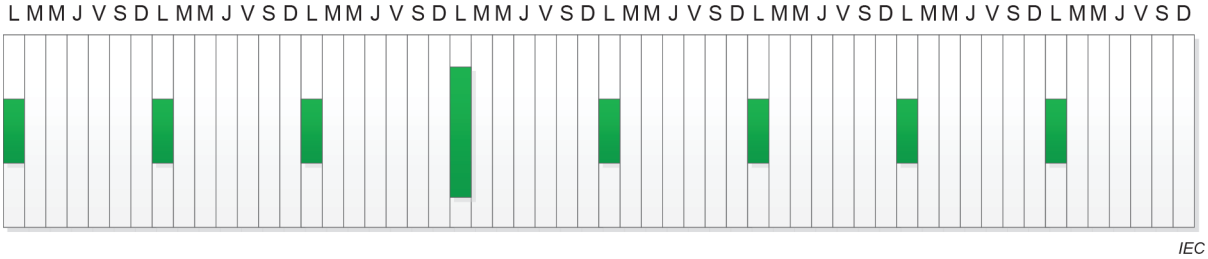


Figure 13 – Evénements récurrents avec un jour particulier

Dans le domaine EACS, il est essentiel de prendre en charge des comportements différents pendant les jours particuliers. La plupart des unités de contrôle d'accès conventionnelles actuellement disponibles sur le marché prennent uniquement en charge la périodicité hebdomadaire. En général, ils ne prennent pas en charge les exceptions, mais ils prennent en charge les jours particuliers.

8.3 Fonctionnalités de service

8.3.1 Généralités

Le dispositif doit fournir les fonctionnalités de service de deux manières:

1) avec la méthode GetServices du service Dispositif (Device) lorsque IncludeCapability est défini sur true. Pour plus d'informations, se référer à [Spécification principale];
2) avec la méthode GetServiceCapabilities.

8.3.2 Structure de données ServiceCapabilities

Les fonctionnalités de service reflètent les fonctionnalités facultatives d'un service. Les informations sont statiques et ne changent pas pendant le fonctionnement du dispositif. Les fonctionnalités suivantes sont disponibles:

- **MaxLimit**
  indique le nombre maximal d'entités renvoyées par une demande Get<Entity>List ou Get<Entity>request unique. Le dispositif ne doit jamais renvoyer dans une même réponse un nombre d'entités supérieur à ce nombre;
- **MaxSchedules**
  indique le nombre maximal de programmes pris en charge par le dispositif. Le dispositif doit prendre en charge au moins un programme;
- **MaxTimePeriodsPerDay**
  indique le nombre maximal de périodes par jour que le dispositif prend en charge dans un programme qui comprend des jours particuliers. Le dispositif doit prendre en charge au moins une période par jour;
- **SpecialDaysSupported**
  si la fonctionnalité est prise en charge, le dispositif doit prendre en charge les jours particuliers;
- **MaxSpecialDayGroups**
  indique le nombre maximal d'entités du groupe de jours particuliers pris en charge par le dispositif. Le dispositif doit prendre en charge au moins une entité SpecialDayGroup si la valeur SpecialDaysSupported est définie sur true. Cette valeur est ignorée si la valeur SpecialDaysSupported est définie sur false;
- **MaxDaysInSpecialDayGroup**
  indique le nombre maximal de jours par entité SpecialDayGroup pris en charge par le dispositif. Le dispositif doit prendre en charge au moins un jour par entité SpecialDayGroup si la valeur SpecialDaysSupported est définie sur true. Cette valeur est ignorée si la valeur SpecialDaysSupported est définie sur false;

- **MaxSpecialDaysSchedules**
  indique le nombre maximal d'entités SpecialDaysSchedule référencées par un programme pris en charge par le dispositif. Cette valeur est ignorée si la valeur SpecialDaysSupported est définie sur false;
- **ExtendedRecurrenceSupported**
  si la valeur ExtendedRecurrenceSupported est définie sur true, les programmes doivent prendre en charge la périodicité étendue conformément au 8.2.3 ;

  si les valeurs ExtendedRecurrenceSupported et SpecialDaysSupported sont définies sur true, les jours particuliers prennent en charge la périodicité étendue conformément au 8.2.3;

  si la valeur ExtendedRecurrenceSupported est définie sur false, seule la périodicité hebdomadaire doit être prise en charge pour les programmes (conformément au 8.2.2) et aucune périodicité ne doit être prise en charge pour les jours particuliers;
- **StateReportingSupported**
  sI cette fonctionnalité est définie sur true, le dispositif doit mettre en œuvre la commande GetScheduleState et doit notifier les clients abonnés chaque fois que les programmes deviennent actifs ou inactifs;
- **ClientSuppliedTokenSupported**
  indique que le client est autorisé à fournir le jeton lors de la création des programmes et des groupes de jours particuliers. Pour permettre l'utilisation des commandes SetSchedule et SetSpecialDayGroup, la valeur doit être définie sur true.

8.3.3 Commande GetServiceCapabilities

Cette opération renvoie les fonctionnalités du service de programmation.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 50.

Tableau 50 – Commande GetServiceCapabilities

GetServiceCapabilities	Classe d'accès: PRE_AUTH
Nom de message	**Description**
GetServiceCapabilitiesRequest	*Ce message doit être vide.*
GetServiceCapabilitiesResponse	*Ce message contient:* • *"Capabilities": le message de réponse de la fonctionnalité répertorie les fonctionnalités du service de programmation demandées, organisées selon une structure de fonctionnalités XML hiérarchique.* tsc:ServiceCapabilities **Capabilities [1][1]**
Codes de défaut	**Description**
	Aucun défaut spécifique à cette commande.

8.4 Informations sur le programme

8.4.1 Généralités

Un programme est un ensemble de périodes, qui peuvent inclure un ou plusieurs programmes de jours particuliers.

8.4.2 Structures de données

8.4.2.1 ScheduleInfo

Le type ScheduleInfo représente le programme comme un objet physique. La structure contient des informations sur une instance de programme spécifique.

Un dispositif doit fournir les champs suivants pour chaque instance ScheduleInfo:

- **token**
  identificateur unique au service du programme;
- **Name**
  nom lisible par l'utilisateur. Il doit comporter 64 caractères au maximum.

Afin de fournir davantage d'informations, le dispositif peut inclure le champ facultatif suivant:

- **Description**
  description du programme lisible par l'utilisateur. Elle doit comporter 1 024 caractères au maximum.

8.4.2.2 Programme

La structure Schedule doit inclure l'ensemble des propriétés de la structure ScheduleInfo, les événements normalisés (format iCalendar), ainsi qu'une liste des instances SpecialDaysSchedule.

Un dispositif doit fournir les champs suivants pour chaque instance Schedule:

- **Standard**
  structure iCalendar qui définit un certain nombre d'événements. Les événements peuvent être récurrents ou non récurrents. Par exemple, les événements peuvent être utilisés pour contrôler quand il convient qu'une caméra effectue l'enregistrement ou quand une installation est accessible;

  certains dispositifs peuvent ne pas être en mesure de prendre en charge l'ensemble des fonctionnalités d'iCalendar. Le réglage de la fonctionnalité de service ExtendedRecurrenceSupported sur false permet à un plus grand nombre de dispositifs d'être conformes à l'ONVIF. Le type est chaîne (mais contient une structure iCalendar);
- **SpecialDays**
  pour les dispositifs qui ne peuvent pas prendre en charge l'ensemble des fonctionnalités d'iCalendar, la prise en charge des jours particuliers est primordiale. Chaque instance SpecialDaysSchedule définit un autre ensemble de périodes qui neutralisent le programme normal pour une liste spécifiée de jours particuliers. Le type est SpecialDaysSchedule.

8.4.2.3 SpecialDaysSchedule

Programme de neutralisation qui définit d'autres périodes pour un groupe de jours particuliers.

Si plusieurs structures SpecialDaysSchedule contiennent des périodes contradictoires pour le même jour particulier, cela entraîne une union des périodes.

La Figure 14 représente l'utilisation d'instances SpecialDaysSchedule. Une instance Schedule appelée "Programme des employés" contient une structure iCalendar qui définit les accès normaux et contient un certain nombre d'instances SpecialDaysSchedule. Chaque SpecialDaysSchedule contient un jeton pour un SpecialDayGroup, ainsi que zéro ou plusieurs instances TimePeriod.

Dans cet exemple, la première instance SpecialDaysSchedule contient le jeton du groupe de jours particuliers "Vacances", mais ne contient aucune période. Par conséquent, aucun accès n'est accordé aux jours particuliers répertoriés dans le groupe de jours particuliers "Vacances".

La deuxième instance SpecialDaysSchedule contient le jeton du groupe de jours particuliers "Demi-journées" et couvre la période "07:00 à 13:00", ce qui octroie un accès pendant ces heures pour les jours particuliers répertoriés dans le groupe de jours particuliers "Demi-journées".

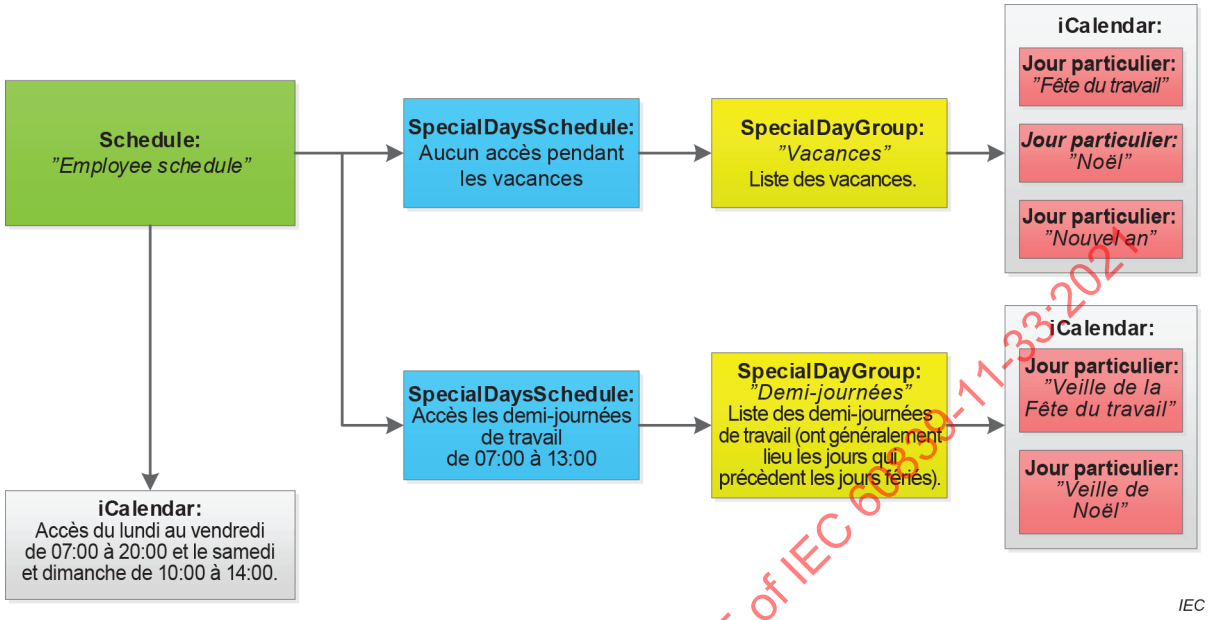


Figure 14 – Exemple de SpecialDaysSchedule

Si un jour particulier est défini comme faisant partie d'un jour (nécessite que ExtendedRecurrenceSupported soit défini sur true), l'accès ne doit être accordé que lorsque la période et la partie du jour particulier se chevauchent.

Prendre l'exemple représenté sur la Figure 15. Il est pris pour hypothèse un programme normalisé qui autorise l'accès en semaine de 06:00 à 18:00 (repéré en vert sur la Figure 15), mais pas le week-end. Il est ensuite pris pour hypothèse un jour particulier qui commence le vendredi 10 juin à 14:00 et se termine le samedi 11 juin à 10:00 (repéré en jaune sur la Figure 15). Si le programme des jours particuliers définit une période de 08:00 à 16:00 (délimitée par des pointillés rouges sur la Figure 15), l'accès est autorisé de 06:00 à 16:00 le vendredi 10 juin et de 08:00 à 10:00 le samedi 11 juin. Voir l'exemple décrit en 8.8.9.

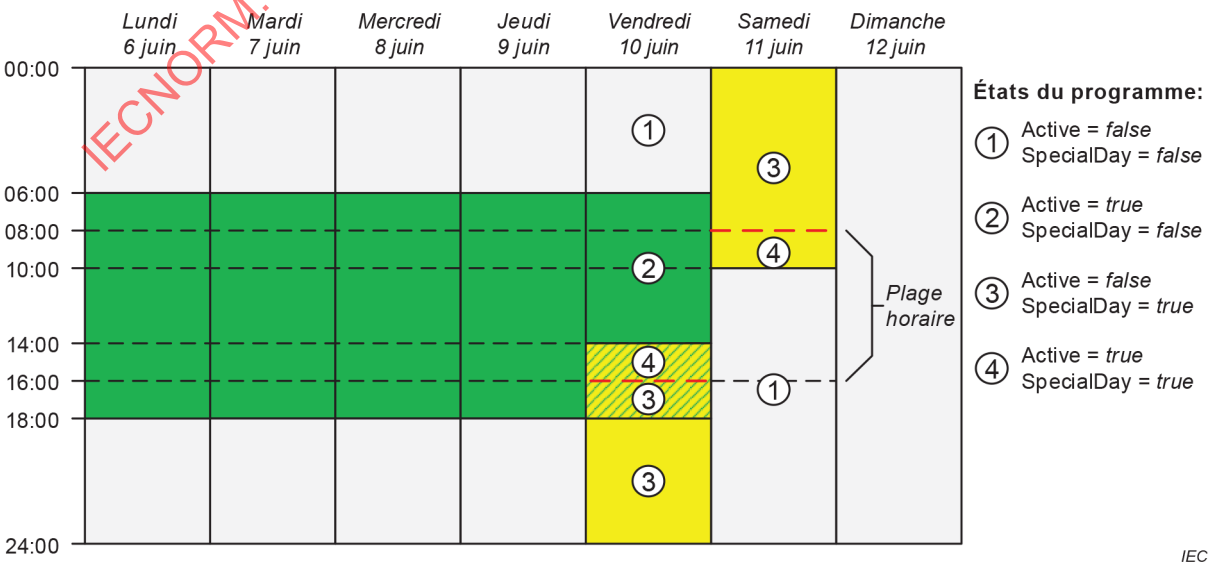


Figure 15 – Exemple de jour particulier avec plage horaire

Un dispositif doit fournir les champs suivants pour chaque instance SpecialDaysSchedule:

- **GroupToken**
  indique la liste des jours particuliers dans un programme;
- **TimeRange**
  indique les autres périodes pour la liste de jours particuliers (neutralisent le programme normal). Par exemple, le programme normal indique qu'il est actif les lundis de 08:00 à 17:00. Cependant, ce lundi est un jour particulier et les autres périodes indiquent que le programme est actif de 09:00 à 11:00 et de 13:00 à 16:00;

  si aucune période n'est définie, aucun accès n'est autorisé. Le type est TimePeriod.

8.4.2.4 TimePeriod

Une période définit une heure de début et de fin. Pour un accès sur l'ensemble de la journée, l'heure de début est "00:00:00" sans heure de fin définie. Pour une période sans heure de fin, le programme dure jusqu'à minuit. Si une heure de fin est spécifiée, elle doit toujours être postérieure à l'heure de début, sinon un défaut InvalidArgVal doit être généré par le dispositif.

Un dispositif doit fournir les champs suivants pour chaque instance TimePeriod:

- **From**
  indique l'heure de début.

Afin de fournir davantage d'informations, le dispositif peut inclure le champ facultatif suivant:

- **Until**
  indique l'heure de fin. Cet élément est facultatif. S'il n'est pas indiqué, la période se termine à minuit. L'heure de fin est exclusive, ce qui signifie que cet instant précis ne fait pas partie de la période. Pour déterminer si un instant (t) fait partie d'une période, la formule StartTime ≤ t < EndTime est utilisée.

8.4.3 Commande GetScheduleInfo

Cette opération demande une liste des éléments ScheduleInfo qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 51.

Tableau 51 – Commande GetScheduleInfo

GetScheduleInfo	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetScheduleInfoRequest	*Ce message contient:* • *"Token": Jetons des éléments ScheduleInfo à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetScheduleInfoResponse	*Ce message contient:* • *"ScheduleInfo": Liste des éléments ScheduleInfo.* tsc:ScheduleInfo **ScheduleInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

8.4.4 Commande GetScheduleInfoList

Cette opération demande une liste de l'ensemble des éléments ScheduleInfo fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 52.

Tableau 52 – Commande GetScheduleInfoList

GetScheduleInfoList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetScheduleInfoListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetScheduleInfoListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"ScheduleInfo": Liste des éléments ScheduleInfo.* xs:string **NextStartReference [0][1]** tsc:ScheduleInfo **ScheduleInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

8.4.5 Commande GetSchedules

Cette opération demande une liste des éléments Schedule qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 53.

Tableau 53 – Commande GetSchedules

GetSchedules	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetScheduleRequest	*Ce message contient:* • *"Token": Jetons des éléments Schedule à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetScheduleResponse	*Ce message contient:* • *"Schedule": Liste des éléments Schedule.* tsc:Schedule **Schedule [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

8.4.6 Commande GetScheduleList

Cette opération demande une liste de l'ensemble des éléments Schedule fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 54.

Tableau 54 – Commande GetScheduleList

GetScheduleList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetScheduleListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetScheduleListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"Schedule": Liste des éléments Schedule.* xs:string **NextStartReference [0][1]** tsc:Schedule **Schedule [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

8.4.7 Commande CreateSchedule

Cette opération crée le programme spécifié dans le dispositif.

Le champ Token de la structure Schedule doit être vide et le dispositif doit attribuer un jeton au programme. Le jeton attribué doit être renvoyé dans la réponse.

Si le client envoie une valeur dans le champ Token, le dispositif doit retourner le code de défaut générique InvalidArgVal.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 55.

Tableau 55 – Commande CreateSchedule

CreateSchedule	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
CreateScheduleRequest	*Ce message contient:* • *"Schedule": Programme à créer.* tsc:Schedule **Schedule [1][1]**
CreateScheduleResponse	*Ce message contient:* • *"Token": Jeton du programme créé.* pt:ReferenceToken **Token [1][1]**
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:MaxSchedules	*L'espace n'est pas suffisant pour ajouter un nouveau programme (voir fonctionnalité MaxSchedules).*
env:Sender ter: CapabilityViolated ter:MaxSpecialDaysSchedules	*Le nombre d'entités SpecialDaysSchedule référencées dans ce programme est trop élevé (voir fonctionnalité MaxSpecialDaysSchedules).*
env:Sender ter: CapabilityViolated ter:MaxTimePeriodsPerDay	*Le nombre de périodes dans un programme journalier est trop élevé (voir fonctionnalité MaxTimePeriodsPerDay).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*

8.4.8 Commande SetSchedule

Cette méthode permet de synchroniser le programme d'un client avec le dispositif.

Si aucun programme avec le jeton spécifié n'existe dans le dispositif, le programme est créé. Si un programme avec le jeton spécifié existe, alors le programme est modifié.

Un appel à cette méthode prend une structure Schedule comme paramètre d'entrée. Le champ Token de la structure Schedule ne doit pas être vide.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité ClientSuppliedTokenSupported, il doit mettre en œuvre cette commande.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 56.

Tableau 56 – Commande SetSchedule

SetSchedule	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
SetScheduleRequest	*Ce message contient:* • *"Schedule": Elément Schedule à créer ou modifier* tsc:Schedule **Schedule [1][1]**
SetScheduleResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*Le dispositif ne prend pas en charge la fourniture du jeton par le client.*
env:Receiver ter:CapabilityViolated ter:MaxSchedules	*L'espace n'est pas suffisant pour ajouter un nouveau programme (voir fonctionnalité MaxSchedules).*
env:Sender ter:CapabilityViolated ter:MaxSpecialDaysSchedules	*Le nombre d'entités SpecialDaysSchedule référencées dans ce programme est trop élevé (voir fonctionnalité MaxSpecialDaysSchedules).*
env:Sender ter:CapabilityViolated ter:MaxTimePeriodsPerDay	*Le nombre de périodes dans un programme journalier est trop élevé (voir fonctionnalité MaxTimePeriodsPerDay).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référencé est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*

8.4.9 Commande ModifySchedule

Cette opération modifie le programme spécifié.

Le jeton du programme à modifier est spécifié dans le champ Token de la structure Schedule et ne doit pas être vide. Tous les autres champs de la structure doivent écraser les champs du programme spécifié.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 57.

Tableau 57 – Commande ModifySchedule

ModifySchedule	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
ModifyScheduleRequest	*Ce message contient:* • *"Schedule": Programme à modifier/mettre à jour.* tsc:Schedule **Schedule [1][1]**
ModifyScheduleResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton spécifié est introuvable.*
env:Sender ter:CapabilityViolated ter:MaxSpecialDaysSchedules	*Le nombre d'entités SpecialDaysSchedule référencées dans ce programme est trop élevé (voir fonctionnalité MaxSpecialDaysSchedules).*
env:Sender ter: CapabilityViolated ter:MaxTimePeriodsPerDay	*Le nombre de périodes dans un programme journalier est trop élevé (voir fonctionnalité MaxTimePeriodsPerDay).*
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	*Un jeton d'entité référence est introuvable (certains dispositifs peuvent ne pas valider les entités référencées).*

8.4.10 Commande DeleteSchedule

Cette opération supprime le programme spécifié.

S'il est associé à une ou plusieurs entités, certains dispositifs peuvent ne pas être en mesure de supprimer le programme et un défaut ReferenceInUse doit donc être généré.

Si aucun jeton n'a été spécifié dans la demande, le dispositif doit renvoyer le code de défaut générique InvalidArgs.

Un dispositif doit prendre en charge cette commande comme décrit dans le Tableau 58.

Tableau 58 – Commande DeleteSchedule

DeleteSchedule	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
DeleteScheduleRequest	*Ce message contient:* • *"Token": Jeton du programme à supprimer.* pt:ReferenceToken **Token [1][1]**
DeleteScheduleResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton du programme est introuvable.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Suppression impossible. Le jeton du programme est en cours d'utilisation.*

8.5 Informations sur les groupes de jours particuliers

8.5.1 Généralités

Un groupe de jours particuliers est un ensemble de jours (ou parties de la journée) qui exige une neutralisation du programme normal par un autre programme, par exemple, les vacances, les demi-journées, les dimanches travaillés.

8.5.2 Structures de données

8.5.2.1 SpecialDayGroupInfo

La structure SpecialDayGroupInfo contient les informations de base concernant la liste des jours particuliers.

Un dispositif doit fournir les champs suivants pour chaque instance SpecialDayGroupInfo:

- **token**
  identificateur unique au service du groupe de jours particuliers;
- **Name**
  nom lisible par l'utilisateur. Il doit comporter 64 caractères au maximum.

Afin de fournir davantage d'informations, le dispositif peut inclure le champ facultatif suivant:

- **Description**
  description des jours particuliers lisibles par l'utilisateur. Elle doit comporter 1 024 caractères au maximum.

8.5.2.2 SpecialDayGroup

La structure SpecialDayGroup doit inclure l'ensemble des propriétés de la structure SpecialDayGroupInfo, ainsi qu'un ensemble de jours particuliers. Un groupe de jours particuliers est un ensemble de jours (ou parties de la journée) qui exige une neutralisation du programme normal par un autre programme, par exemple, les vacances, les demi-journées, les dimanches travaillés.

Un dispositif doit fournir le champ suivant pour chaque instance SpecialDayGroup:

- **Days**
  structure iCalendar qui contient un groupe de jours particuliers. Le type est chaîne (mais contient une structure iCalendar).

8.5.3 Commande GetSpecialDayGroupInfo

Cette opération demande une liste des éléments SpecialDayGroupInfo qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité SpecialDaysSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 59.

Tableau 59 – Commande GetSpecialDayGroupInfo

GetSpecialDayGroupInfo	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSpecialDayGroupInfoRequest	*Ce message contient:* • *"Token": Jetons des éléments SpecialDayGroupInfo à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetSpecialDayGroupInfoResponse	*Ce message contient:* • *"SpecialDayGroupInfo": Liste des éléments SpecialDayGroupInfo.* tsc:SpecialDayGroupInfo **SpecialDayGroupInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

8.5.4 Commande GetSpecialDayGroupInfoList

Cette opération demande une liste de l'ensemble des éléments SpecialDayGroupInfo fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité SpecialDaysSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 60.

Tableau 60 – Commande GetSpecialDayGroupInfoList

GetSpecialDayGroupInfoList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSpecialDayGroupInfoListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSpecialDayGroupInfoListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"SpecialDayGroupInfo": Liste des éléments SpecialDayGroupInfo.* xs:string **NextStartReference [0][1]** tsc:SpecialDayGroupInfo **SpecialDayGroupInfo [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

8.5.5 Commande GetSpecialDayGroups

Cette opération demande une liste des éléments SpecialDayGroup qui correspondent aux jetons indiqués.

Le dispositif doit ignorer les jetons qu'il ne peut pas résoudre et doit renvoyer une liste vide si aucun élément ne correspond aux jetons spécifiés. Le dispositif ne doit pas renvoyer de défaut dans ce cas.

Si le nombre d'éléments demandés est supérieur à la valeur MaxLimit, un défaut TooManyItems doit être renvoyé.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité SpecialDaysSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 61.

Tableau 61 – Commande GetSpecialDayGroups

GetSpecialDayGroups	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSpecialDayGroupsRequest	*Ce message contient:* • *"Token": Jetons des éléments SpecialDayGroup à obtenir.* pt:ReferenceToken **Token [1][unbounded]**
GetSpecialDayGroupsResponse	*Ce message contient:* • *"SpecialDayGroup": Liste des éléments SpecialDayGroup.* tsc:SpecialDayGroup **SpecialDayGroup [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgs ter:TooManyItems	*Le nombre d'éléments demandés est trop élevé (voir fonctionnalité MaxLimit).*

8.5.6 Commande GetSpecialDayGroupList

Cette opération demande une liste de l'ensemble des éléments SpecialDayGroupList fournis par le dispositif.

Un appel à cette méthode doit renvoyer un paramètre StartReference lorsque les données n'ont pas toutes été renvoyées alors qu'un nombre supérieur de données est disponible. La référence doit être valide pour récupérer l'ensemble de données suivant. Pour plus d'informations, se reporter au 4.7.3 de l'IEC 60839-11-32:2016.

Le nombre d'éléments renvoyés ne doit pas être supérieur à la valeur du paramètre Limit.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité SpecialDaysSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 62.

Tableau 62 – Commande GetSpecialDayGroupList

GetSpecialDayGroupList	Classe d'accès: READ_SYSTEM
Nom de message	**Description**
GetSpecialDayGroupListRequest	*Ce message contient:* • *"Limit": Nombre maximal d'entrées à renvoyer. Si aucune valeur n'est spécifiée, que la valeur soit inférieure à 1 ou qu'elle soit supérieure au nombre d'éléments pris en charge par l'appareil, le nombre d'éléments est déterminé par l'appareil;* • *"StartReference": Démarrer le renvoi des entrées à partir de cette référence de début. Si aucune valeur n'est spécifiée, les entrées doivent commencer depuis le début de l'ensemble de données.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetSpecialDayGroupListResponse	*Ce message contient:* • *"NextStartReference": StartReference à utiliser lors du prochain appel pour obtenir les éléments suivants. Si cet élément n'est pas spécifié, cela signifie qu'il n'y a aucun autre élément à obtenir;* • *"SpecialDayGroup": Liste des éléments SpecialDayGroup.* xs:string **NextStartReference [0][1]** tsc:**SpecialDayGroup SpecialDayGroup [0][unbounded]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*La StartReference n'est pas valide ou a expiré. Il est nécessaire que le client commence la récupération depuis le début.*

8.5.7 Commande CreateSpecialDayGroup

Cette opération crée le groupe de jours particuliers spécifié dans le dispositif.

Le champ Token de la structure SpecialDayGroup doit être vide et le dispositif doit attribuer un jeton au groupe de jours particuliers. Le jeton attribué doit être renvoyé dans la réponse.

Si le client envoie une valeur dans le champ Token, le dispositif doit retourner le code de défaut générique InvalidArgVal.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité SpecialDaysSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 63.

Tableau 63 – Commande CreateSpecialDayGroup

CreateSpecialDayGroup	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
CreateSpecialDayGroupRequest	*Ce message contient:* • *"SpecialDayGroup": Groupe de jours particuliers à créer.* tsc:SpecialDayGroup **SpecialDayGroup [1][1]**
CreateSpecialDayGroupResponse	*Ce message contient:* • *"Token": Jeton du groupe de jours particuliers créé.* pt:ReferenceToken **Token [1][1]**
Codes de défaut	**Description**
env:Receiver ter: CapabilityViolated ter:MaxSpecialDayGroups	*L'espace n'est pas suffisant pour ajouter de nouveaux éléments SpecialDayGroup (voir fonctionnalité MaxSpecialDayGroups).*
env:Sender ter: CapabilityViolated ter:MaxDaysInSpecialDayGroup	*Le nombre de jours particuliers dans un SpecialDayGroup est trop élevé (voir fonctionnalité MaxDaysInSpecialDayGroup).*

8.5.8 Commande SetSpecialDayGroup

Cette méthode permet de synchroniser un groupe de jours particuliers avec le dispositif dans un client.

Si aucun groupe de jours particuliers avec le jeton spécifié n'existe dans le dispositif, le groupe de jours particuliers est créé. Si un groupe de jours particuliers avec le jeton spécifié existe, le groupe de jours particuliers est modifié.

Un appel à cette méthode prend une structure SpecialDayGroup comme paramètre d'entrée. Le champ Token de la structure SpecialDayGroup doit être vide.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité ClientSuppliedTokenSupported, il doit mettre en œuvre cette commande comme décrit dans le Tableau 64.

Tableau 64 – Commande SetSpecialDayGroup

SetSpecialDayGroup	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
SetSpecialDayGroupRequest	*Ce message contient:* • *"SpecialDayGroup": Elément SpecialDayGroup à créer ou modifier.* tsc:SpecialDayGroup **SpecialDayGroup [1][1]**
SetSpecialDayGroupResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	*Le dispositif ne prend pas en charge la fourniture du jeton par le client.*
env:Receiver ter:CapabilityViolated ter:MaxSpecialDayGroups	*L'espace n'est pas suffisant pour ajouter de nouveaux éléments SpecialDayGroup (voir fonctionnalité MaxSpecialDayGroups).*
env:Sender ter:CapabilityViolated ter:MaxDaysInSpecialDayGroup	*Le nombre de jours particuliers dans un SpecialDayGroup est trop élevé (voir fonctionnalité MaxDaysInSpecialDayGroup).*
env:Sender ter:InvalidArgVal ter:MissingToken	*Le jeton de l'élément groupe de jours particuliers doit être spécifié.*

8.5.9 Commande ModifySpecialDayGroup

Cette opération modifie le groupe de jours particuliers spécifié.

Le jeton du groupe de jours particuliers à modifier est spécifié dans le champ Token de la structure SpecialDayGroup et ne doit pas être vide. Tous les autres champs de la structure doivent écraser les champs du groupe de jours particuliers spécifié.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité SpecialDaysSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 65.

Tableau 65 – Commande ModifySpecialDayGroup

ModifySpecialDayGroup	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
ModifSpecialDayGroupRequest	*Ce message contient:* • *"SpecialDayGroup": Elément SpecialDayGroup à modifier/mettre à jour.* tsc:SpecialDayGroup **SpecialDayGroup [1][1]**
ModifySpecialDayGroupResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton spécifié est introuvable.*
env:Sender ter: CapabilityViolated ter:MaxDaysInSpecialDayGroup	*Le nombre de jours particuliers dans un SpecialDayGroup est trop élevé (voir fonctionnalité MaxDaysInSpecialDayGroup).*

8.5.10 Commande DeleteSpecialDayGroup

Cette méthode permet de supprimer le groupe de jours particuliers spécifié.

S'il est associé à un ou plusieurs programmes, certains dispositifs peuvent ne pas être en mesure de supprimer le groupe de jours particuliers et un défaut ReferenceInUse doit être généré.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité SpecialDaysSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 66.

Tableau 66 – Commande DeleteSpecialDayGroup

DeleteSpecialDayGroup	Classe d'accès: WRITE_SYSTEM
Nom de message	**Description**
DeleteSpecialDayGroupRequest	*Ce message contient:* • *"Token": Jeton de l'élément SpecialDayGroup à supprimer.* pt:ReferenceToken **Token [1][1]**
DeleteSpecialDayGroupResponse	*Ce message doit être vide.*
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton du groupe de jours particuliers est introuvable.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Suppression impossible. Le jeton du groupe de jours particuliers est en cours d'utilisation.*

8.6 Etat du programme

8.6.1 Structure de données ScheduleState

L'instance ScheduleState contient les informations d'état d'un programme. Les deux champs ci-dessous correspondent aux quatre états qu'un programme peut avoir.

Un dispositif doit fournir les champs suivants pour chaque instance ScheduleState:

- **Active**
  indique que l'heure actuelle se situe dans les limites du programme ou des périodes de ses programmes de jours particuliers. Par exemple, si ce programme est utilisé pour déclencher un enregistrement automatique sur une source vidéo, le drapeau Actif est défini sur true lorsque l'enregistrement basé sur le programme enregistre par hypothèse.

Afin de fournir davantage d'informations, le dispositif peut inclure le champ facultatif suivant:

- **SpecialDay**
  indique que l'heure actuelle se situe dans les limites des périodes de ses programmes de jours particuliers. Par exemple, si ce programme est utilisé pour effectuer un enregistrement vidéo à un débit de trames inférieur pendant des jours particuliers, le drapeau SpecialDay est défini sur true. Si les jours particuliers ne sont pas pris en charge par le dispositif, ce champ peut ne pas être indiqué et peut être interprété comme false par le client.

Exemple

Prendre pour hypothèse un programme qui définit un programme normalisé actif de 09:00 à 17:00 en semaine. Il existe deux groupes de jours particuliers: un groupe définit les demi-journées de travail, l'autre groupe définit les jours fériés. Le programme des jours particuliers définit une période qui va de 09:00 à 13:00 pour les demi-journées de travail.

La Figure 16 représente le moment, où les quatre états se produisent:

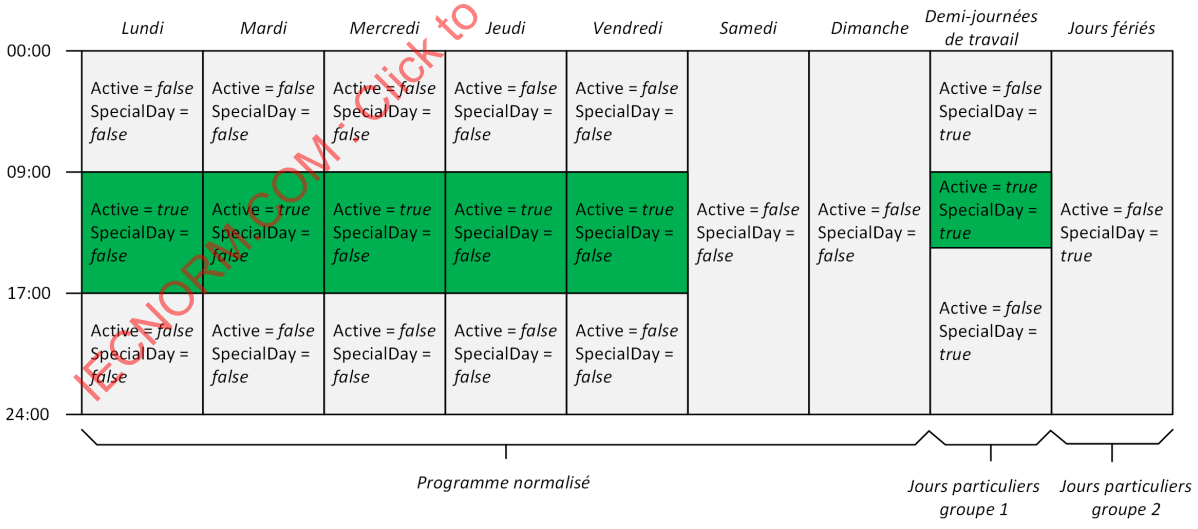


Figure 16 – Etats du programme

8.6.2 Commande GetScheduleState

Cette opération demande l'élément ScheduleState pour l'instance de programme spécifiée par le jeton indiqué.

Lorsqu'un dispositif indique qu'il prend en charge la fonctionnalité StateReportingSupported, il doit prendre en charge cette méthode comme décrit dans le Tableau 67.

Tableau 67 – Commande GetScheduleState

GetScheduleState	Classe d'accès: READ_SYSTEM_SENSITIVE
Nom de message	**Description**
GetScheduleStateRequest	*Ce message contient:* • *"Token": Jeton de l'instance Schedule pour obtenir ScheduleState.* pt:ReferenceToken **Token [1][1]**
GetScheduleStateResponse	*Ce message contient:* • *"ScheduleState": Element ScheduleState.* tsc:ScheduleState **ScheduleState [1][1]**
Codes de défaut	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Le jeton du programme est introuvable.*

8.7 Rubriques de notification

8.7.1 Généralités

Le 8.7 définit les rubriques de notification spécifiques au service de programme.

8.7.2 Présentation de l'événement (informative)

Le service de programmation spécifie les événements lorsque les programmes ou les jours particuliers sont actifs et lorsque les programmes ou les jours particuliers sont modifiés.

La rubrique principale pour les changements d'état est la suivante:

- tns1:Schedule/State/Active.

Les rubriques principales pour les notifications de changement de configuration sont les suivantes:

- tns1:Configuration/Schedule/Changed;
- tns1:Configuration/Schedule/Removed;
- tns1:Configuration/SpecialDays/Changed;
- tns1:Configuration/SpecialDays/Removed.

8.7.3 Changements d'état

8.7.3.1 Généralités

Le dispositif doit fournir les événements de changement d'état afin d'informer les clients abonnés lorsque l'état de l'entité programme change. Le dispositif doit utiliser les rubriques définies en 8.7.3, ainsi que la description de message associée.

8.7.3.2 Programme

Si la fonctionnalité StateReportingSupported est définie sur true, le service/dispositif doit être capable de générer l'événement suivant lorsqu'un programme ou ses jours particuliers deviennent actifs ou inactifs en fonction de l'heure du dispositif. Il s'agit d'un événement de propriété qui indique si le programme est actif ou non.

```
Topic: tns1:Schedule/State/Active
<tt:MessageDescription IsProperty="true">
  <tt:Source>
    <tt:SimpleItemDescription Name="ScheduleToken"
                              Type="pt:ReferenceToken"/>
    <tt:SimpleItemDescription Name="Name"
                              Type="xs:string"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="Active"
                              Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="SpecialDay"
                              Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

8.7.4 Modifications de configuration

8.7.4.1 Généralités

Chaque fois que les données de configuration sont modifiées, ajoutées ou supprimées, le dispositif compatible doit fournir ces événements afin d'avertir les clients abonnés.

8.7.4.2 Programme

Chaque fois que les données de configuration d'un programme (y compris SpecialDaysSchedule) sont modifiées ou qu'un programme est ajouté, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/Schedule/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="ScheduleToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt: MessageDescription>
```

Chaque fois qu'un programme est supprimé, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/Schedule/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="ScheduleToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

8.7.4.3 SpecialDayGroup

Chaque fois que les données de configuration d'un élément SpecialDayGroup sont modifiées ou qu'un élément SpecialDayGroup est ajouté, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/SpecialDays/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SpecialDaysToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Chaque fois qu'un élément SpecialDayGroup est supprimé, le dispositif doit fournir l'événement suivant:

```
Topic: tns1:Configuration/SpecialDays/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="SpecialDaysToken"
                              Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

8.8 Exemples

8.8.1 Généralités

Toutes les structures iCalendar utilisées dans les exemples ci-dessous ne sont pas complètes. Seules les parties explicatives sont incluses.

8.8.2 Accès 24h/24, 7j/7 pour le personnel administratif

Dans l'exemple suivant, les valeurs de SpecialDaysSupported et de ExtendedRecurrenceSupported ne sont pas pertinentes.

```
<Schedule token="402">
      <Name>Access 24*7</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access 24*7
            DTSTART:19700101T000000
            DTEND:19700102T000000
            RRULE:FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

8.8.3 Accès le lundi et le mercredi de 06:00 à 20:00 pour l'équipe de nettoyage

Dans l'exemple suivant, les valeurs de SpecialDaysSupported et de ExtendedRecurrenceSupported ne sont pas pertinentes.

```
<Schedule token="402">
      <Name>Access on Monday and Wednesday from 06:00 to 20:00</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access on Monday and Wednesday from 06:00 to 20:00
            DTSTART:19700105T060000
            DTEND:19700105T200000
            RRULE:FREQ=WEEKLY;BYDAY=MO,WE
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

8.8.4 Accès le vendredi de 18:00 à 07:00 pour le personnel d'entretien

Cet exemple peut être réalisé de deux manières selon les fonctionnalités du service de programmation exposé par le dispositif.

Cas A

L'exemple suivant est valide pour ExtendedRecurrenceSupported=true. La valeur de SpecialDaysSupported n'est pas pertinente.

Si ExtendedRecurrenceSupported=false pour l'exemple ci-dessous, le dispositif génère un défaut; car l'année dans DTSTART n'est pas "1970".

```
<Schedule token="402">
      <Name>Access from Friday 18:00 to 07:00 for maintenance staff</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access from Friday 18:00 to 07:00 for maintenance staff
            DTSTART:20140523T180000
            DTEND:20140526T070000
            RRULE:FREQ=WEEKLY;BYDAY=FR
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

Cas B

L'exemple suivant est valide pour SpecialDaysSupported=false. La valeur de ExtendedRecurrenceSupported n'est pas pertinente.

```
<Schedule token="402">
      <Name>Access from Friday 18:00 to 07:00 for maintenance staff</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access from Friday 18:00 to Friday midnight
            DTSTART:19700102T180000
            DTEND:19700103T000000
            RRULE:FREQ=WEEKLY;BYDAY=FR
            END:VEVENT
            BEGIN:VEVENT
            SUMMARY:Access for all Saturdays and all Sundays
            DTSTART:19700103T000000
            DTEND:19700104T000000
            RRULE:FREQ=WEEKLY;BYDAY=SA,SU
            END:VEVENT
            BEGIN:VEVENT
            SUMMARY:Access from Sunday midnight to Monday 07:00
            DTSTART:19700105T000000
            DTEND:19700105T070000
            RRULE:FREQ=WEEKLY;BYDAY=MO
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

8.8.5 Accès en semaine de 08:00 à 17:00 pour les employés

Dans l'exemple suivant, les valeurs de SpecialDaysSupported et de ExtendedRecurrenceSupported ne sont pas pertinentes.

```
<Schedule token="402">
      <Name>Access on weekdays from 08:00 to 17:00 for employees</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access on weekdays from 08:00 to 17:00 for employees
            DTSTART:19700101T080000
            DTEND:19700101T170000
            RRULE:FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

8.8.6 Accès du 15 janvier 2014 au 14 janvier 2015 de 09:00 à 18:00

Cet exemple peut être réalisé de deux manières selon les fonctionnalités du service de programmation exposé par le dispositif.

Cas A

L'exemple suivant est valide pour ExtendedRecurrenceSupported=true. La valeur de SpecialDaysSupported n'est pas pertinente.

```
<Schedule token="402">
      <Name>Access from Jan 15, 2014, to Jan 14, 2015, from 09:00 to 18:00</Name>
      <Standard>
            BEGIN:VCALENDAR
            BEGIN:VEVENT
            SUMMARY:Access from Jan 15, 2014, to Jan 14, 2015, from 09:00 to 18:00
            DTSTART:20140115T090000
            DTEND:20150114T180000
            RRULE:FREQ=DAILY
            END:VEVENT
            END:VCALENDAR
      </Standard>
</Schedule>
```

Cas B

L'exemple suivant est valide pour SpecialDaysSupported=false et ExtendedRecurrenceSupported=false.

Ce scénario ne peut pas être directement mis en œuvre dans le dispositif en utilisant uniquement le programme si la fonctionnalité ExtendedRecurrenceSupported n'est pas prise en charge. Le système EACS utilise les dates de début et de fin dans les politiques d'accès qui renvoient à des programmes. En outre, il est nécessaire que le client ou l'hôte divise le programme en un programme hebdomadaire (quelle que soit la périodicité prise en charge par le dispositif) et qu'il télécharge ces programmes sur le dispositif chaque fois que le programme est modifié (15 janvier 2014 et 14 janvier 2015).

8.8.7 Exemple de jours particuliers 1

Un employé a un accès défini conformément au programme donné en exemple en 8.8.5. Cependant, l'administrateur du site souhaite restreindre l'accès aux employés le jour de Noël (jeudi) étant donné qu'il s'agit d'un jour férié national. L'administrateur du site dispose d'une entité de jours particuliers appelée "Jours fériés nationaux", à laquelle s'ajoute le 25 décembre. L'administrateur du site modifie le programme afin d'ajouter un programme de jours particuliers pour l'entité "Jours fériés nationaux" afin de neutraliser les accès et d'empêcher les employés d'accéder à l'installation pendant ces jours.

L'exemple suivant est valide pour SpecialDaysSupported=true. La valeur de ExtendedRecurrenceSupported n'est pas pertinente.

```
<SpecialDayGroup token="8765">
      <Name>National Holidays</Name>
      <Days>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Christmas day
            DTSTART:20141225T000000
            DTEND:20141226T000000
            UID:Holiday@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Days>
<SpecialDayGroup>

<Schedule token="402">
      <Name>Access on Monday and Wednesday from 08:00 to 17:00. No access on Christmas
day</Name>
      <Standard>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Access on Monday and Wednesday from 08:00 to 17:00
            DTSTART:19700105T080000
            DTEND:19700105T170000
            RRULE:FREQ=WEEKLY;BYDAY=MO,WE
            UID:Event@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Standard>
      <SpecialDaysSchedule>
            <GroupToken>8765</GroupToken>
            <TimeRange></TimeRange>
            <!-- No time period indicates no access on referred special days -->
      </SpecialDaysSchedule>
</Schedule>
```

8.8.8 Exemple de jours particuliers 2

Un concierge a un accès défini conformément au calendrier donné en exemple en 8.8.5. Cependant, l'administrateur du site souhaite autoriser l'accès à l'équipe de nettoyage le jour de Noël (jeudi), car il s'agit d'un jour férié national. L'administrateur du site dispose d'une entité de jours particuliers appelée "Jours fériés nationaux", à laquelle s'ajoute le 25 décembre. L'administrateur du site modifie le programme afin d'ajouter un programme de jour particulier pour l'entité "Jours fériés nationaux" afin de neutraliser les accès et de permettre au concierge d'accéder à l'installation.

L'exemple suivant est valide pour SpecialDaysSupported=true. La valeur de ExtendedRecurrenceSupported n'est pas pertinente.

```
<SpecialDayGroup token="8765">
      <Name>National Holidays</Name>
      <Days>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Christmas day
            DTSTART:20141225T000000
            DTEND:20141226T000000
            UID:Holiday@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Days>
<SpecialDayGroup>

<Schedule token="402">
      <Name>Access on Monday and Wednesday from 08:00 to 17:00. No access on Christmas
day</Name>
      <Standard>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Access on Monday and Wednesday from 08:00 to 17:00
            DTSTART:19700105T080000
            DTEND:19700105T170000
            RRULE:FREQ=WEEKLY;BYDAY=MO,WE
            UID:Event@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Standard>
      <SpecialDaysSchedule>
            <GroupToken>8765</GroupToken>
            <TimeRange>
                  <From>00:00:00</From>
            </TimeRange>
      </SpecialDaysSchedule>
</Schedule>
```

8.8.9 Exemple de jours particuliers 3

Un programme normalisé octroie un accès en semaine de 07:00 à 18:00, mais pas le week-end. Un jour particulier commence le vendredi 10 juin à 14:00 et se termine le samedi 11 juin à 10:00. Le programme des jours particuliers définit une plage horaire qui va de 08:00 à 16:00.

```
<SpecialDayGroup token="7312">
      <Name>Team building days</Name>
      <Days>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Boot camp
            DTSTART:20160610T140000
            DTEND:20160611T100000
            UID:TeamBuilding@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Days>
<SpecialDayGroup>

<Schedule token="245">
       <Name>Employee schedule 2016</Name>
      <Description>Access on weekdays from 07:00 to 18:00.
             Access from 08:00 to 16:00 on Team building days</Description>
      <Standard>
            BEGIN:VCALENDAR
            PRODID:VERSION:2.0
            BEGIN:VEVENT
            SUMMARY:Access on weekdays from 07:00 to 18:00
            DTSTART:20160101T070000
            DTEND:20161231T180000
            RRULE:FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR
            UID:Workhours@ONVIF.com
            END:VEVENT
            END:VCALENDAR
      </Standard>
      <SpecialDaysSchedule>
            <GroupToken>7312</GroupToken>
            <StartTime>08:00:00</StartTime>
            <Until>16:00:00</Until>
      </SpecialDaysSchedule>
</Schedule>
```

Annexe A
(normative)

Schéma XML d'interface de contrôle d'accès

A.1 Service de l'identifiant WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:tcr="http://www.onvif.org/ver10/credential/wsdl"
    name="CredentialService"
    targetNamespace="http://www.onvif.org/ver10/credential/wsdl">
  <!-- The data types definition for the 'Credential Service' -->
  <wsdl:types>
    <xs:schema targetNamespace="http://www.onvif.org/ver10/credential/wsdl"
               xmlns:xs="http://www.w3.org/2001/XMLSchema"
               xmlns:pt="http://www.onvif.org/ver10/pacs"
               xmlns:tcr="http://www.onvif.org/ver10/credential/wsdl"
               elementFormDefault="qualified"
               version="18.06">
      <xs:import namespace="http://www.onvif.org/ver10/pacs"
                 schemaLocation="../../pacs/types.xsd"/>
      <!--ServiceCapabilities definitions-->
      <xs:complexType name="ServiceCapabilities">
        <xs:annotation>
          <xs:documentation>
            The service capabilities reflect optional functionality of a service. The
            information is static and does not change during device operation.
          </xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:element name="SupportedIdentifierType" type="pt:Name" minOccurs="1"
                      maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>
                A list of identifier types that the device supports. Identifier types starting
                with the prefix pt: are reserved to define ONVIF specific types. For custom
                defined identifier types shall all share the "pt:&lt;Name&gt;" syntax.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="Extension" type="tcr:ServiceCapabilitiesExtension" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="MaxLimit" type="pt:PositiveInteger" use="required">
          <xs:annotation>
            <xs:documentation>
              The maximum number of entries returned by a single Get&lt;Entity&gt;List or
              Get&lt;Entity&gt; request. The device shall never return more than this number of
              entities in a single response.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="CredentialValiditySupported" type="xs:boolean" use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates that the device supports credential validity.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="CredentialAccessProfileValiditySupported" type="xs:boolean"
                      use="required">
          <xs:annotation>
            <xs:documentation>
              Indicates that the device supports validity on the association between a
              credential and an access profile.
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute name="ValiditySupportsTimeValue" type="xs:boolean" use="required">
          <xs:annotation>
            <xs:documentation>
```

```
        Indicates that the device supports both date and time value for validity. If set
        to false, then the time value is ignored.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="MaxCredentials" type="pt:PositiveInteger" use="required">
    <xs:annotation>
      <xs:documentation>
        The maximum number of credential supported by the device.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="MaxAccessProfilesPerCredential" type="pt:PositiveInteger"
                use="required">
    <xs:annotation>
      <xs:documentation>
        The maximum number of access profiles for a credential.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ResetAntipassbackSupported" type="xs:boolean" use="required">
    <xs:annotation>
      <xs:documentation>
        Indicates the device supports resetting of anti-passback violations and notifying
        on anti-passback violations.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ClientSuppliedTokenSupported" type="xs:boolean" default="false">
    <xs:annotation>
      <xs:documentation>
        Indicates that the client is allowed to supply the token when creating
        credentials. To enable the use of the command SetCredential, the value must be set
        to true.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="DefaultCredentialSuspensionDuration" type="xs:string"
                default="PT5M">
    <xs:annotation>
      <xs:documentation>
        The default time period that the credential will temporary be suspended (e.g. by
        using the wrong PIN a predetermined number of times). The time period is defined
        as an [ISO 8601] duration string (e.g. "PT5M").
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="ServiceCapabilitiesExtension">
  <xs:sequence>
    <xs:element name="SupportedExemptionType" type="pt:Name" minOccurs="0"
                maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>
          A list of exemptions that the device supports. Supported exemptions starting
          with the prefix pt: are reserved to define ONVIF specific exemption types and
          these reserved exemption types shall all share "pt:&lt;Name&gt;" syntax.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of definition -->
<!--CredentialInfo definitions-->
<xs:complexType name="CredentialInfo">
  <xs:annotation>
    <xs:documentation>
      The CredentialInfo type represents the credential as a logical object. The structure
      contains the basic information of a specific credential instance. The device shall
      provide the following fields for each credential.
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="pt:DataEntity">
      <xs:sequence>
```

```
          <xs:element name="Description" type="pt:Description" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                User readable description for the credential. It shall be up to 1024
                characters.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="CredentialHolderReference">
            <xs:annotation>
              <xs:documentation>
                An external reference to a person holding this credential. The reference is
                a username or used ID in an external system, such as a directory service.
              </xs:documentation>
            </xs:annotation>
            <xs:simpleType base="xs:string">
              <xs:annotation>
                <xs:documentation></xs:documentation>
              </xs:annotation>
              <xs:restriction base="xs:string">
                <xs:maxLength value="64"/>
                <xs:minLength value="0"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="ValidFrom" type="xs:dateTime" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                The start date/time validity of the credential. If the
                ValiditySupportsTimeValue capability is set to false, then only date is
                supported (time is ignored).
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="ValidTo" type="xs:dateTime" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                The expiration date/time validity of the credential. If the
                ValiditySupportsTimeValue capability is set to false, then only date is
                supported (time is ignored).
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
        <xs:anyAttribute processContents="lax"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!-- End of definition -->
  <!--Credential definitions-->
  <xs:complexType name="Credential">
    <xs:annotation>
      <xs:documentation>
        A Credential is a physical/tangible object, a piece of knowledge, or a facet of a
        person's physical being, that enables an individual access to a given physical
        facility or computer-based information system. A credential holds one or more
        credential identifiers. To gain access one or more identifiers may be required.
      </xs:documentation>
    </xs:annotation>
    <xs:complexContent>
      <xs:extension base="tcr:CredentialInfo">
        <xs:sequence>
          <xs:element name="CredentialIdentifier" type="tcr:CredentialIdentifier"
                      minOccurs="1" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>
                A list of credential identifier structures. At least one credential
                identifier is required. Maximum one credential identifier structure per type
                is allowed.
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="CredentialAccessProfile" type="tcr:CredentialAccessProfile"
                      minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>
                A list of credential access profile structures.
              </xs:documentation>
```

```
        </xs:annotation>
      </xs:element>
      <xs:element name="ExtendedGrantTime" type="xs:boolean" default="false">
        <xs:annotation>
          <xs:documentation>
            A boolean indicating that the credential holder needs extra time to get
            through the door. ExtendedReleaseTime will be added to ReleaseTime, and
            ExtendedOpenTime will be added to OpenTime
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Attribute" type="pt:Attribute" minOccurs="0"
                  maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            A list of credential attributes as name value pairs. Key names starting with
            the prefix pt: are reserved to define PACS specific attributes following the
            "pt:&lt;Name&gt;" syntax.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="Extension" type="tcr:CredentialExtension" minOccurs="0"/>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
<xs:complexType name="CredentialExtension">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
</xs:complexType>
<!-- End of data types definition -->
<!--CredentialIdentifier definitions-->
<xs:complexType name="CredentialIdentifier">
  <xs:annotation>
    <xs:documentation>
      A credential identifier is a card number, unique card information, PIN or biometric
      information such as fingerprint, iris, vein, face recognition, that can be validated
      in an access point.
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="Type" type="tcr:CredentialIdentifierType">
      <xs:annotation>
        <xs:documentation>Contains the details of the credential identifier type. Is of
          type CredentialIdentifierType.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ExemptedFromAuthentication" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          If set to true, this credential identifier is not considered for authentication.
          For example if the access point requests Card plus PIN, and the credential
          identifier of type PIN is exempted from authentication, then the access point
          will not prompt for the PIN.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Value" type="xs:hexBinary">
      <xs:annotation>
        <xs:documentation>
          The value of the identifier in hexadecimal representation.
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    <!-- first ONVIF then Vendor -->
  </xs:sequence>
  <xs:anyAttribute processContents="lax"/>
</xs:complexType>
<xs:complexType name="CredentialIdentifierType">
  <xs:annotation>
    <xs:documentation>
      Specifies the name of credential identifier type and its format for the credential
      value.
```