INTERNATIONAL STANDARD

ISO 20078-3

First edition
2019-05

Road vehicles — Extended vehicle (ExVe) web services —

Part 3: Security

Véhicule routiers — Web services du véhicule étendu (ExVe) —

Partie 3: Sécurité

Reference number
ISO 20078-3:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication*.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Road vehicles — Extended vehicle (ExVe) web services —

Part 3: **Security**

1 Scope

This document defines how to authenticate users and Accessing Parties on a web services interface. It also defines how a Resource Owner can delegate Access to its Resources to an Accessing Party. Within this context, this document also defines the necessary roles and required separation of duties between these in order to fulfil requirements stated on security, data privacy and data protection.

All conditions and dependencies of the roles are defined towards a reference implementation using *OAuth 2.0 compatible framework* and *OpenID Connect 1.0 compatible framework.*

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.

ISO 20078-1, *Road vehicles — Extended vehicle (ExVe) 'web services' — Content*

3 Terms, definitions and abbreviations

For the purposes of this document, the terms, definitions and abbreviations given in ISO 20078-1 and 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 https://www.iso.org/obp

3.1
Identity Token
ID Token
digitally signed JWT and contains claims about the authenticated Resource Owner

3.2
Access Token
AT
digitally signed JWT issued by the Identity Provider or Authorization Provider and consumed by the Resource Provider

Note 1 to entry: An Access Token represents an authorization that is issued to the client and limited by scope and has a defined expiration time.

3.3
Refresh Token
RT
credential (string) issued to the Accessing Party by the Identity Provider or the Authorization Provider and used to obtain a new Access Token when the currently used AT expires, or to obtain additional ATs depending on the intended scope of use

3.4
Authorization Code
intermediate result of a successful Resource Owner authorization process and is used by authorized clients to obtain Access Tokens and optionally Refresh Tokens

3.5
Claim
asserted information about a certain entity

EXAMPLE ROID, Resource Owner’s first name, last name, address, Connected Vehicle’s capability and/or other attributes.

3.6
Token Issuer
entity that generates and provides Identity Tokens, Access Tokens, and Refresh Tokens

4 General

4.1 Processes

The following processes are specific to each Offering Party. The definition of these processes is not part of this document but shall be in place in order to apply this specification.

REQ_04_01_01	The process to register a Resource Owner at the Identity Provider shall be the responsibility of the Offering Party.

REQ_04_01_02	The process to register an Accessing Party at the Authorization Provider shall be the responsibility of the Offering Party.

REQ_04_01_03	The process to confirm the technical eligibility of Connected Vehicles and provision of their associated ExVe Resources shall be the responsibility of the Offering Party.

REQ_04_01_04	The process to verify a Resource Owner’s current and valid ownership of the concerned resource shall be the responsibility of the Offering Party.

4.2 Conditions

REQ_04_02_05	The Offering Party shall be able to restrict or deny the Accessing Party and/or the Resource Owner Access to the Offering Party’s web services and portals.

NOTE 1 This could be done to, for example, fulfil security and legislation requirements.

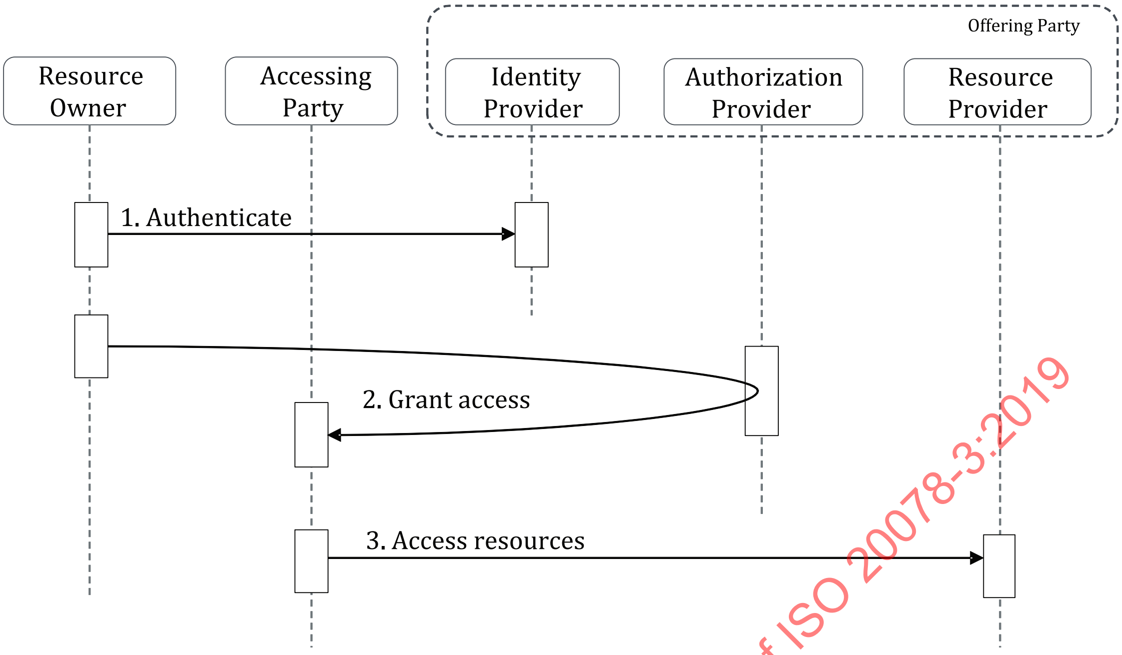
REQ_04_02_06	If the Offering Party revokes a granted registration of an Accessing Party, the Offering Party may delete all Containers created by the Accessing Party, if Containers are used.

NOTE 2 Revocation of the registration can be due to access violation or other misuse of the web services.

5 Basic Communication Flow

5.1 General

This document separates the activities necessary for authentication, authorization and Resource Access into three distinct communication flows with separate duties (see Figure 1).



1 The Resource Owner is authenticated by the Identity Provider.

2 The Resource Owner is granting access to the Accessing Party. The granting is handled by the Authorization Provider.

3 The Accessing Party is accessing resources from the Resource Provider.

Figure 1 — The roles and the three distinct communication flows

5.2 Authentication

The Identity Provider is responsible for authenticating the Resource Owner and managing the Resource Owner profile, based on the Resource Owner registration. The Resource Owner credentials are revealed only to the Identity Provider, and the Identity Provider confirms a successful authentication to concerned parties. If the Resource Owner has given consent, the Accessing Party will be authorized to access the Resource Owner's profile (Figure 2).

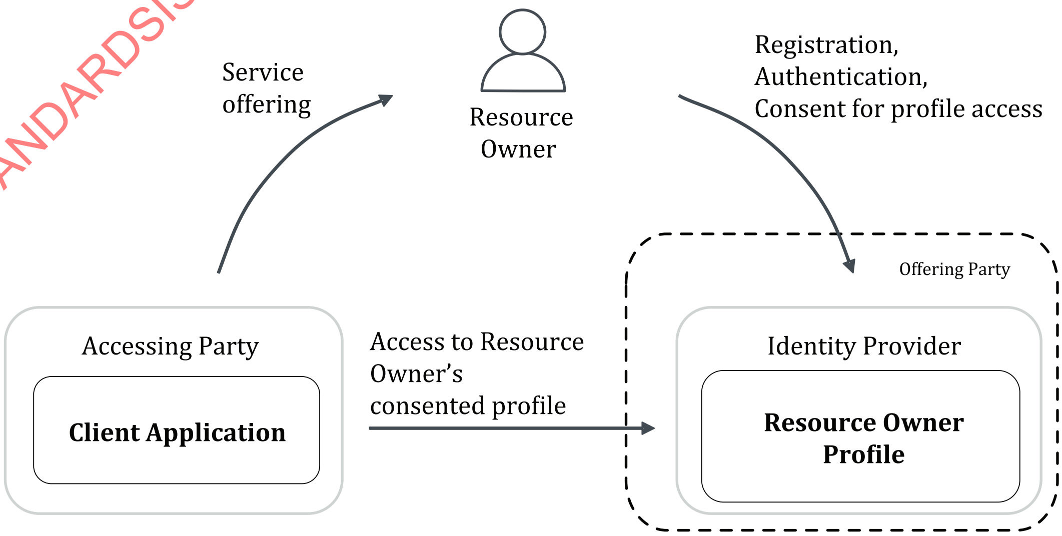


Figure 2 — Resource Owner Authentication and Access to Resource Owner's Profile

REQ_05_02_01	The Identity Provider shall offer a suitable authentication method and shall perform the authentication process. After a successful authentication, the Identity Provider shall confirm the identity of the authenticated Resource Owner.

REQ_05_02_02	The Resource Owner's credentials shall only need to be known by the Resource Owner and be possible to be verified by the Identity Provider.

REQ_05_02_03	The Resource Owner's registration and authentication (at the Identity Provider), shall be separated from the authorization process to grant access to Resources (via the Authorization Provider).

REQ_05_02_04	If the Identity Provider is able to expose the Resource Owner's profile to the Accessing Party, it is only the Resource Owner that shall be able to grant or deny access.

5.3 Authorization

The Client Application as a component of the Accessing Party requires Access to Resources on behalf of the Resource Owner. At the authorization step, the Accessing Party requests authorization to access the Resources provided by the Resource Provider (Offering Party). The required authorization is requested at the Authorization Provider, providing the intended scope. By the consent of the Resource Owner, the Authorization Provider returns a limited authorization to the client application of the Accessing Party. Using the obtained authorization, the Client Application can access Resources.

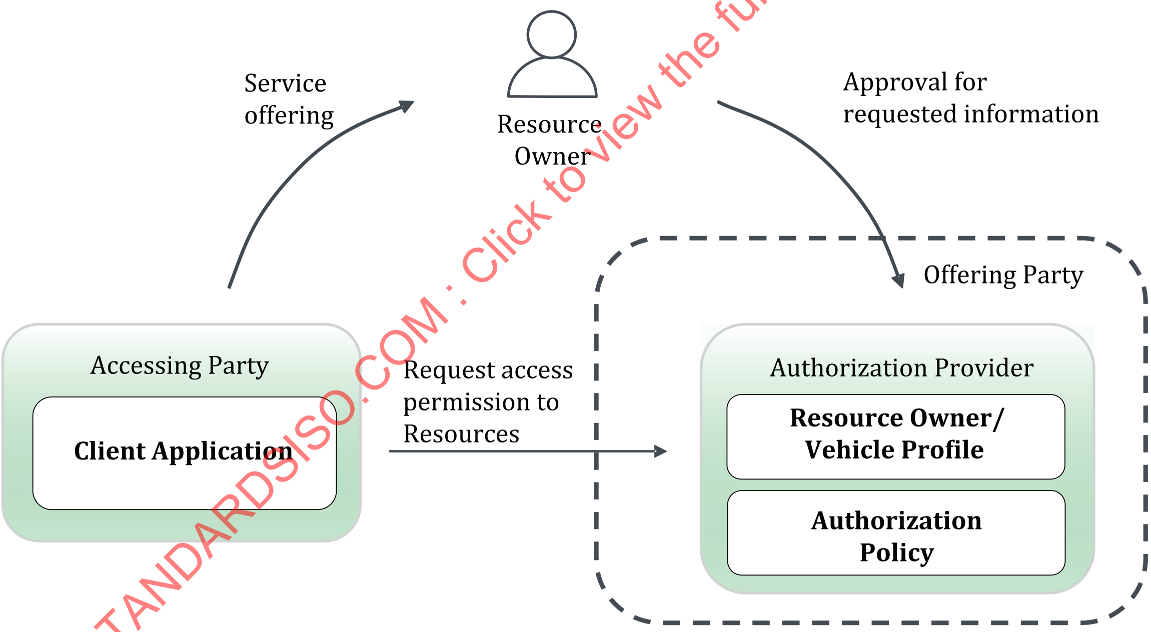


Figure 3 — Requesting Access to Resources

REQ_05_03_01	Before accessing the Resource, the Accessing Party shall request Access at the Authorization Provider providing the intended scope.

REQ_05_03_02	The Authorization Provider shall be responsible for the management of the Authorization Policy and shall manage all granted Accesses.

REQ_05_03_03	The Authorization Provider shall trust the confirmation of successful authentication as provided by the Identity Provider.

REQ_05_03_04	The Authorization Policy shall be defined by the Offering Party concerning the authorization process.

REQ_05_03_05	The Authorization provider shall be able to verify the relationship between Resource Owners and their Resources.

REQ_05_03_06	Only the Resource Owner shall be able to grant Access to a Resource.

NOTE The Access is granted to an Accessing Party at the Offering Party.

REQ_05_03_07	Granting Access to resources shall be done either directly or via Containers. The Offering Party decides if one or both of the granting methods shall be provided to the Accessing Parties.

REQ_05_03_08	The Resource Owner shall be able to revoke a granted Access to a Resource at any time.

REQ_05_03_09	If Containers are used, the Resource Owner shall be able to revoke a granted Access to a Containers at any time.

REQ_05_03_10	The Authorization Provider shall ask the Resource Owner for the approval before providing the authorization to the Accessing Party resulting in a granted Access.

REQ_05_03_11	Upon request the Offering Party shall present a Resource Owner's granted Accesses to the Resource Owner.

REQ_05_03_12	The Resource Owner shall be able to deny an Access request to a Resource, or if Containers are used, to a Container at any time.

REQ_05_03_13	If the Ownership of a Resource or the relationship between the Resource Owner and the Resource ends, Access to the corresponding Resources, and if Containers are used, also to Containers, shall be revoked.

REQ_05_03_14	If Containers are used and if a Container is deleted, all Access granted to that Container shall be revoked.

5.4 Resource Access

Using the Access, the Accessing Party can access the Resources, hosted by the Resource Provider.

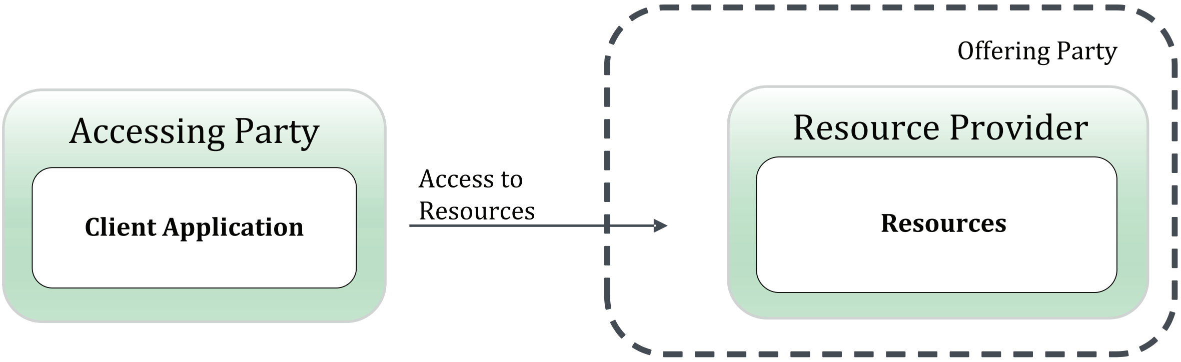


Figure 4 — Access to Resources via the Resource Provider

REQ_05_04_01	The Resource Provider shall perform Access control to the Resources according to the Authorization Policy.

5.5 Separation of duties

Separation of duties concerns the separation of tasks and responsibilities between entities involved in the authentication, authorization and access to Resources.

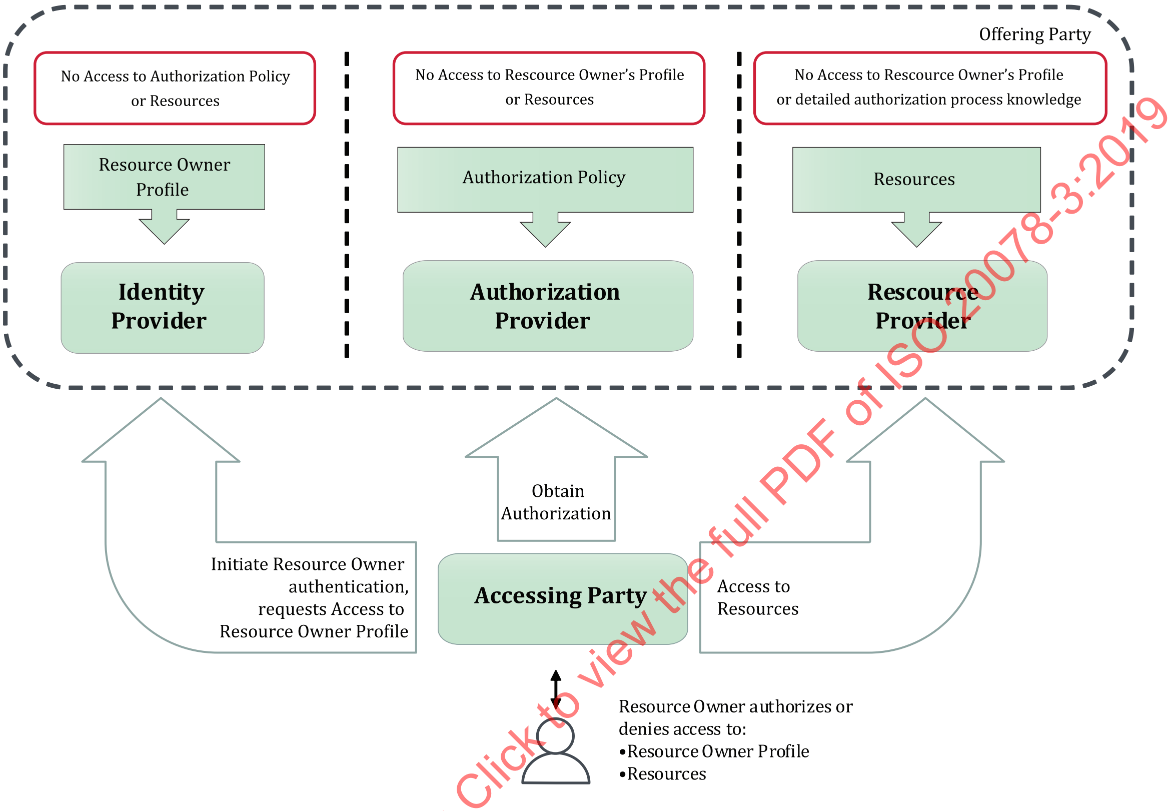

Offering Party
No Access to Authorization Policy or Resources
No Access to Rescource Owner's Profile or Resources
No Access to Rescource Owner's Profile or detailed authorization process knowledge
Resource Owner Profile
Authorization Policy
Resources
Identity Provider
Authorization Provider
Rescource Provider
Initiate Resource Owner authentication, requests Access to Resource Owner Profile
Obtain Authorization
Access to Resources
Accessing Party
Resource Owner authorizes or denies access to:
•Resource Owner Profile
•Resources

Figure 5 — Separation of duties between involved roles

Figure 5 describe the separation of duties between involved roles, where the Offering Party has the three roles: Identity Provider, Authorization Provider, and Resource Provider.

REQ_05_05_01	The Identity Provider shall not be dependent on the Authorization Policy.

REQ_05_05_02	The Identity Provider shall not influence the Authorization Policy.

REQ_05_05_03	The Identity Provider shall not access the Resources.

REQ_05_05_04	The Authorization Provider shall not access the Resource Owner Profile.

REQ_05_05_05	The Authorization Provider shall only use the unique Resource Owner ID to identify the resource owner.

NOTE 1 The Resource Owner ID is generated and communicated by the trusted Identity Provider.

REQ_05_05_06	The Authorization Provider shall not have Access to Resources provided by the Resource Provider.

REQ_05_05_07	The Resource Provider shall not access the Resource Owner Profile.

REQ_05_05_08	The Resource Provider shall not know details about the authorization process.

REQ_05_05_09	The Resource Provider trusts the Authorization Provider and shall verify whether the provided authorization matches the Access control rules defined for the requested Resources.

REQ_05_05_10	The Resource Owner shall not need to share credentials with the Accessing Party to enable the Accessing Party to access the Resources.

REQ_05_05_11	The Accessing Party shall only access the Resources with the consent of the Resource Owner.

NOTE 2 The requirements stated above do not impose requirements on specific architecture, design or organizational structure.

Figure 6 shows the major logical components of the involved roles and the associated entities:

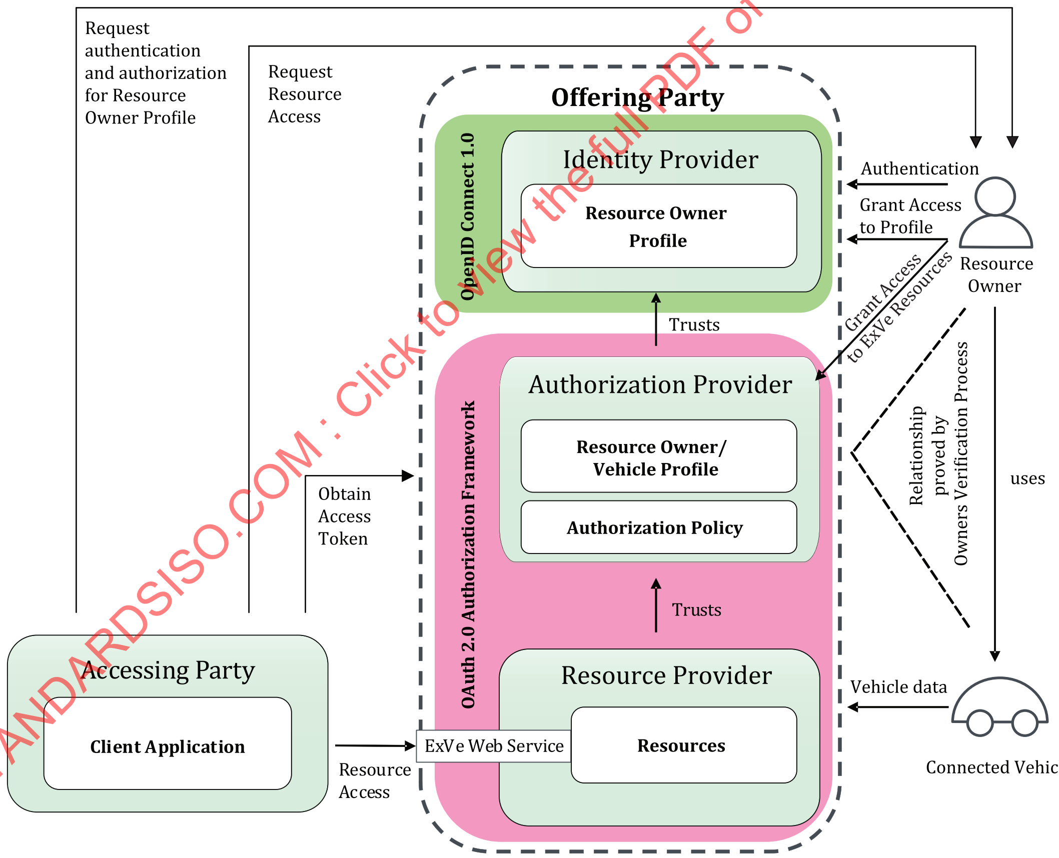


Figure 6 — Involved roles and associated entities

5.6 Implementation Related Considerations

The physical implementation and assignment of roles to real parties differs from the logical representation as shown in Figure 6, and follows the defined requirements as referenced by the following figure.

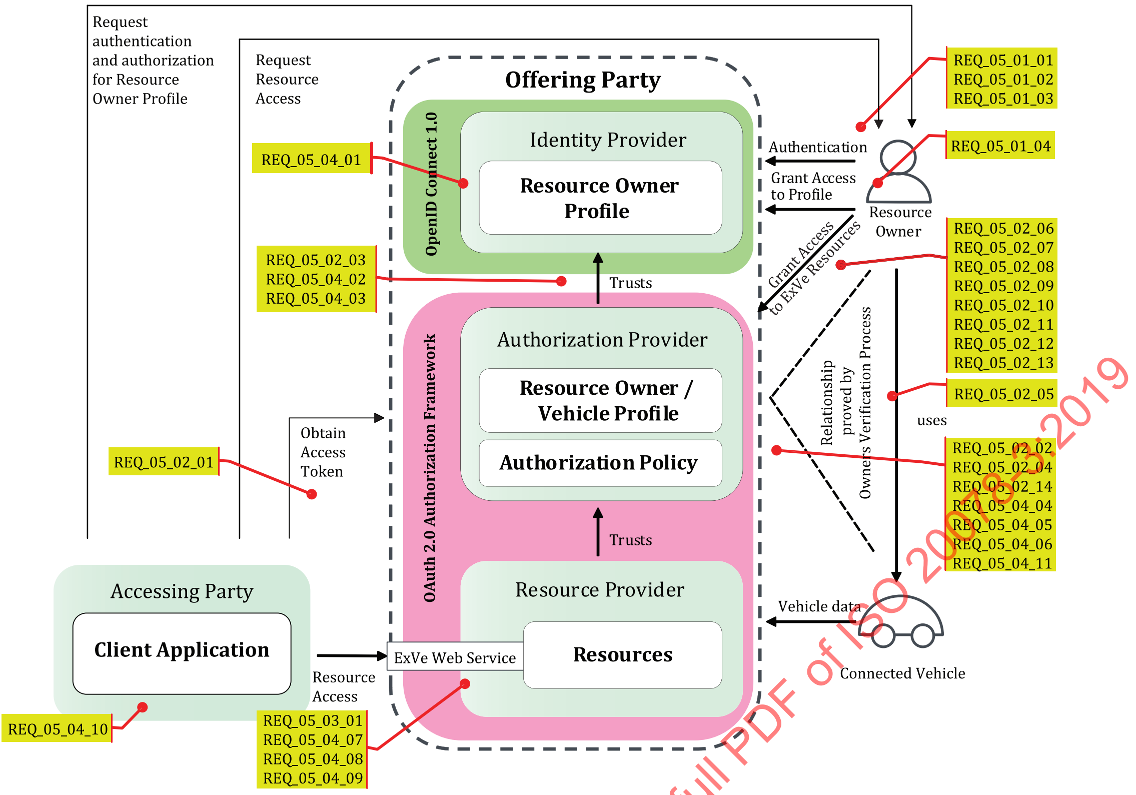


Figure 7 — Logical representation of involved roles and their entities in reference to the defined requirements

Additionally, actual implementation often depends on national legal requirements (e.g. handling of Resource Owner Profile, implemented Resource Owner's Verification Process etc.) and the required trusted relationship between involved components especially Identity Provider, Authorization Provider, and Resource Provider.

REQ_05_06_01	All communication paths between involved entities shall use secured connections.

REQ_05_06_02	The Identity Provider, Authorization Provider, and Resource Provider are responsible for ensuring that only recent cipher suites are used.

NOTE Changes in the interface are communicated to Accessing Parties within a reasonable notice period.

If the Offering Party encounters an unreliable Accessing Party, the Offering Party can temporarily or permanently revoke the Accessing Party's access. This is done in order to protect the Resource Owners. Examples of circumstances that could trigger this are: insecure smartphone applications, disabled host verification, data breach of database, forbidden caching or storage of resource data, usage of discouraged security algorithms.

REQ_05_06_03	It shall be possible to validate the authenticity and integrity of information provided by the Identity Provider, Authorization Provider and Resource Provider.

REQ_05_06_04	To ensure the interoperability between involved entities in different physical environments, an implementation shall follow a framework compatible with OAuth 2.0 and OpenID Connect 1.0.

Annex A provides one example of how to implement OAuth 2.0 and OpenID Connect 1.0.

Annex A
(informative)

Reference Implementation using OAuth 2.0 and OpenID Connect 1.0

A.1 Introduction

This reference implementation is designed in accordance with the general approach (see Clause 4) using OAuth 2.0 framework[1] and OpenID Connect 1.0[2] specifications. OAuth 2.0 is used to implement an authorization mechanism for requesting of authorization and accessing Resources. OpenID Connect 1.0 is used as an authentication layer on top of the OAuth 2.0 framework for Resource Owner related scenarios, where the proof of the Resource Owner identity using appropriate authentication method through an Identity Provider is required.

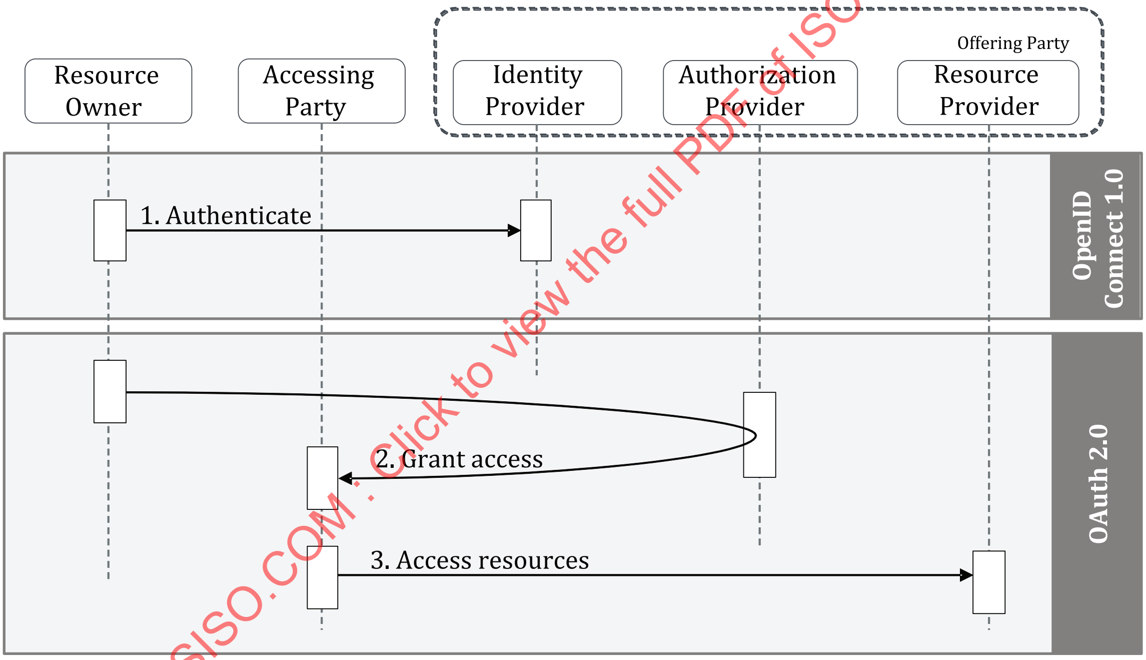


Figure A.1 — Extended Vehicle split to the usage of OAuth 2.0 and OpenID Connect 1.0

The Client Application of the Accessing Party should support an implementation of the standard OAuth 2.0[1] for authorization requests and access to protected Resources, and may support OpenID Connect 1.0[2] for Resource Owner authentication and access to the profile of the Resource Owner.

Both standards are using the term *Authorization Server*. However, this document differentiates between **logical** components, the *Identity Server* maintained by the Identity Provider and the *Authorization Server* maintained by the Authorization Provider. In this reference implementation, the ExVe Identity Server refers to OpenID Connect 1.0 Authorization Server and the ExVe Authorization Server refers to OAuth 2.0 Authorization Server.

The Reference Implementation does not cover all of the technical details. The terms and definitions to facilitate the understanding of the referenced implementation are provided in Clause 3.

The implementation of the components should comply with the following guidelines.

— For Resource Owner authentication, OpenID Connect 1.0 Authorization Code Flow, OIDC Core[2] should be used by the Accessing Party.

- The Identity Provider should provide a “UserInfo” endpoint as defined in OpenID Connect 1.0[2] to make the Resource Owner Profile available.
- OAuth 2.0 grant type *Authorization Code* is recommended when requesting authorization for protected Resources owned by a Resource Owner, RFC 6749[1]. Offering Party and Accessing Party can agree on other grant types.
- In the authorization code flow, the Client Application will first get an *Authorization Code* which then needs to be exchanged for the identity token (Identity Provider) or the access token (Authorization Provider).
- The Identity Provider and/or the Authorization Provider may request a registration of the Client Application before the Client Application can consume services provided by the Identity Server and/or the Authorization Server. With successful registration the Client Application will receive client credentials. The design of the client registration process, the credential type and the client authentication method are under the responsibility of the Identity Provider and the Authorization Provider.
- OAuth 2.0 grant type *Client Credentials* can be used for Resources, where runtime interaction with the Resource Owner is not required, RFC 6749[1].
- The Authorization Server and the Identity Server should provide a service for revocation of granted permissions in accordance with the OAuth 2.0 Token Revocation, RFC 7009[7].
- The Issuer of tokens (Identity Server, Authorization Server) may expose OAuth 2.0 Token Introspection Endpoints according to RFC 7662[4].
- All tokens (identity token, refresh token, access token) should be digitally signed using asymmetric keys as defined in JSON Web Signature (JWS), RFC 7515.[8] Allowed algorithms are defined in JSON Web Algorithms, RFC 7518[5].
- The Token Issuer should provide all valid public keys for signature validation as defined in JSON Web Key (JWK), RFC 7517[3].
- The Access token type should be *bearer* as defined in RFC 6750[9].
- The Access tokens may be self-contained or may reference the authorization information stored at the token issuer. Self-contained access tokens allow the Resource Server to perform an authorization decision without further interaction with the Authorization Server. To allow the reliable revocation of self-contained tokens the lifetime should be limited to maximum one hour.
- If issued, the Client Application should store refresh tokens in a long-term secure storage and continue to use them as long as they remain valid. Refresh tokens should be treated by the clients as a secret and need only be sent exclusively to the issuer of the refresh token.
- Implementers should pay attention to the section *Security Considerations* in RFC 6749[1], RFC 7517[3], RFC 7662[4], RFC 7518[5], RFC 6819[6], RFC 7009[7], RFC 7515[8], RFC 6750[9], RFC 7636[10].

A.2 Claims

A.2.1 General

For ExVe specific claims the prefix *exve.* can be used.

A.2.2 ID Token Claims

In addition to required claims defined in OpenID Connect 1.0[2], ID token can contain following custom claim:

exve.roid (Unique Resource Owner ID)

A.2.3 Access Token Claims

In addition to required claims defined in JSON Web Token (JWT) RFC 7519[11], access token may contain following custom claims:

exve.roid (Resource Owner ID)

exve.cid (Container ID)

exve.rid (Resource ID)

One or more Access IDs are linked to the Resource Owner ID, Resource IDs and/or Container IDs.

A.2.4 Refresh Token Claims

In addition to required claims defined in JSON Web Token (JWT) RFC 7519[11], refresh token should at a minimum contain the following custom claim:

exve.roid (Unique Resource Owner ID)

NOTE The refresh token is used with the scope to request a new access token, as the refresh token only contains the Resource Owner ID.

A.3 Use Cases

A.3.1 Access to Protected Resources with Resource Owner's Approval at Runtime

The Accessing Party wants to access Resource Owner-related Resources and an authorization at runtime is required (approval).

This initial access needs four steps.

1) Resource Owner authentication (and optionally granting access to the Resource Owner's profile) by the Identity Provider.

2) Obtain basic profile information about the Resource Owner from the Identity Provider.

3) Requesting authorization for required Resources at the Authorization Provider.

4) Access to Resources at the Resource Provider.

In the following example the Accessing Party has implemented an application running on a webserver (client application) and interacting with the Resource Owner via a User Agent (Browser).

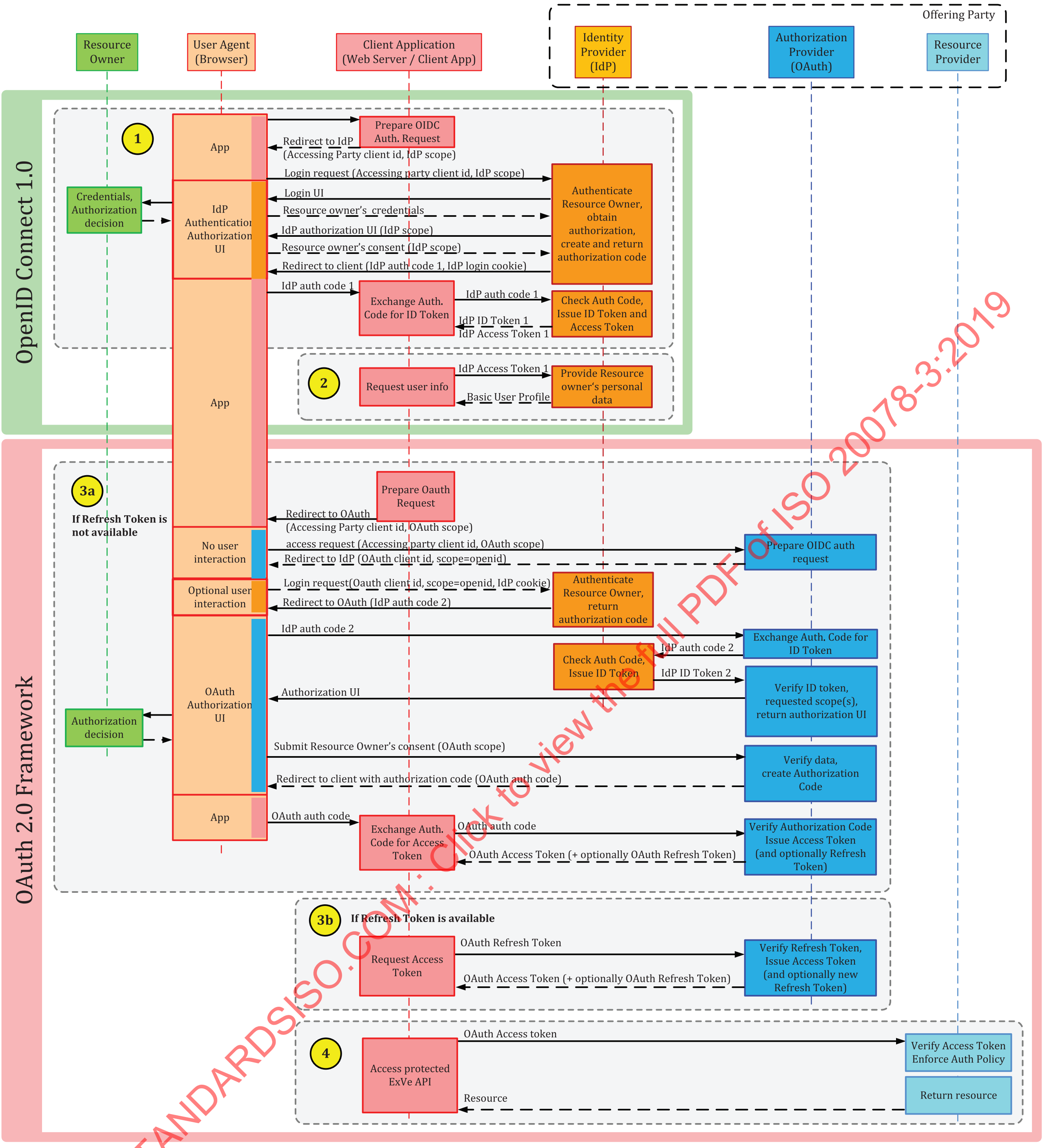

Offering Party
Resource Owner
User Agent (Browser)
Client Application (Web Server / Client App)
Identity Provider (IdP)
Authorization Provider (OAuth)
Resource Provider
OpenID Connect 1.0
1
App
Prepare OIDC Auth. Request
Redirect to IdP
(Accessing Party client id, IdP scope)
Login request (Accessing party client id, IdP scope)
Credentials, Authorization decision
IdP Authentication Authorization UI
Login UI
Resource owner's credentials
IdP authorization UI (IdP scope)
Resource owner's consent (IdP scope)
Redirect to client (IdP auth code 1, IdP login cookie)
Authenticate Resource Owner, obtain authorization, create and return authorization code
IdP auth code 1
Exchange Auth. Code for ID Token
IdP auth code 1
Check Auth Code, Issue ID Token and Access Token
IdP ID Token 1
IdP Access Token 1
App
2
Request user info
IdP Access Token 1
Basic User Profile
Provide Resource owner's personal data
OAuth 2.0 Framework
3a
If Refresh Token is not available
Prepare Oauth Request
Redirect to OAuth
(Accessing Party client id, OAuth scope)
No user interaction
access request (Accessing party client id, OAuth scope)
Redirect to IdP (OAuth client id, scope=openid)
Prepare OIDC auth request
Optional user interaction
Login request(Oauth client id, scope=openid, IdP cookie)
Redirect to OAuth (IdP auth code 2)
Authenticate Resource Owner, return authorization code
IdP auth code 2
Exchange Auth. Code for ID Token
IdP auth code 2
Check Auth Code, Issue ID Token
IdP ID Token 2
Authorization decision
OAuth Authorization UI
Authorization UI
Verify ID token, requested scope(s), return authorization UI
Submit Resource Owner's consent (OAuth scope)
Verify data, create Authorization Code
Redirect to client with authorization code (OAuth auth code)
App
OAuth auth code
Exchange Auth. Code for Access Token
OAuth auth code
Verify Authorization Code Issue Access Token (and optionally Refresh Token)
OAuth Access Token (+ optionally OAuth Refresh Token)
3b
If Refresh Token is available
Request Access Token
OAuth Refresh Token
Verify Refresh Token, Issue Access Token (and optionally new Refresh Token)
OAuth Access Token (+ optionally OAuth Refresh Token)
4
Access protected ExVe API
OAuth Access token
Verify Access Token Enforce Auth Policy
Resource
Return resource

Figure A.2 — Obtaining Authorization for Protected Resources Owned by a Resource Owner

Detailed description of the diagram as shown in Figure A.2.

1) Resource Owner identity should be verified by the associated Identity Provider as defined in OpenID Connect 1.0[2] section *Authentication using the Authorization Code Flow*. For this, the Accessing Party's Client Application redirects the browser to the Identity Provider and initiates the authentication process where the Resource Owner authenticates directly with the Identity Provider. The Identity Provider checks the credentials and shows the Resource Owner the scope of personal data the client application server wants to access. Optionally, the Client Application can also request access to the Resource Owner Profile of the Resource Owner ("UserInfo" Endpoint). As a result of the successful Resource Owner authentication (credentials correct and user grants

permission for the requested identity scope), the Identity Provider returns an authorization code and redirects the browser back to the client application server. This quite complex process ensures that the Resource Owner does not have to provide their credentials to the client application. The Resource Owner can also check during the redirection to be connected to their well-known identity provider (HTTPS) and is in control of the provided personal data (scope).

The browser hands over the authorization code to the client application server. With the authorization code the client application server requests a digitally signed ID Token from the Identity Provider. If optionally requested and granted, the Client Application can get additionally the Access Token for the "UserInfo" Endpoint, issued by the Identity Server for the scope granted by the Resource Owner. The issued tokens are typically only valid for the requesting client, i.e. in this example the client application.

2) The Client application can use the Access Token for the "UserInfo" Endpoint to obtain (as an example) the Basic Profile of the Resource Owner. The Basic Profile contains, for example, a subset of the Resource Owner's profile data. This step is optional; the ID Token itself can hold enough personal information for some use cases. The needed identity scope (subset) can vary depending on the use case and is granted by the Resource Owner (cf. step 1).

3) A Request for authorization is based on the Authorization Code Grant as defined in RFC 6749[1]. Depending on the availability and validity of the refresh token, the Client Application can request the authorization following either step 3 a) (first-time access) or step 3 b) (subsequent authorization requests if refresh token is available and valid).

   a) If the refresh token is not available on the client side, the Resource Owner approval is required. The Client Application requests authorization at the Authorization Provider, providing the intended authorization scope and the client application id. The Authorization Provider will redirect the browser to the Identity Provider to be able to authenticate the Resource Owner, in the same way as in step 1. As the Resource Owner has already been authenticated by the same Identity Provider in the previous step, the Resource Owner will in most cases just confirm its identity. The Authorization Provider validates the requested authorization scope, the Resource Owner ID, the Resource Owner's relationship with the connected vehicle and other subjects according to the defined authorization policy. If successfully validated the Authorization Provider requests Resource Owner's approval providing the authorization UI. This process uses the similar technical browser redirections as step 1. The step might look complicated, but enables that the Resource Owner can check to be connected to their well-known Authorization Provider (HTTPS) and is in control of the granted authorization scope. With Resource Owner's consent, the Authorization Provider issues the digitally signed Access Token for the requested resources and returns the Access Token to the Client Application. Optionally, the Authorization Provider may issue a Refresh Token, limited to the scope granted by the Resource Owner.

   b) For the subsequent access to Resources, the Client Application should use the Refresh Token, issued by the Authorization Server to retrieve new Access Tokens, as long as the new authorization request is within the scope of the Refresh Token. Steps 1, 2, and 3 a) can be omitted.

4) The Client Application accesses Resources by providing the Access Token. The Resource Server validates the Access Token claims and Access Token signature, checks whether the Access Token matches the defined access control rules and, if successful, processes the request.

A.3.2 Access to Protected Resources with Resource Owner's Approval at Runtime (simplified)

If the Accessing Party does not need to access basic profile information, the flow can be somewhat simplified and the order of the steps changes compared to A.3.1.

This initial access needs two steps.

1) Requesting authorization for required Resources at the Authorization Provider including Resource Owner authentication via the Identity Provider.