INTERNATIONAL STANDARD

ISO/IEC 23093-2

Second edition
2022-03

Information technology — Internet of media things —

Part 2: Discovery and communication API

Technologies de l'information — Internet des objets media — Partie 2: API pour la découverte et la communication

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

Reference number
ISO/IEC 23093-2:2022(E)

© ISO/IEC 2022

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Published in Switzerland

© ISO/IEC 2022 – All rights reserved

Contents

Page

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

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 document 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 or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC 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) or the IEC list of patent declarations received (see patents.iec.ch).

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. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 23093-2:2019), which has been technically revised.

The main changes are as follows:

— modification of the introduction;

— addition of new APIs for discovery and communication;

— addition of a transaction model using state channels.

A list of all parts in the ISO/IEC 23093 series can be found on the ISO and IEC websites.

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 and www.iec.ch/national-committees.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

Introduction

The ISO/IEC 23093 series provides an architecture and specifies APIs and compressed representation of data flowing between media things.

The APIs for the media things facilitate discovering other media things in the network, connecting and efficiently exchanging data between media things. The APIs also support transaction tokens to access valuable functionalities, resources, and data from media things.

Media things related information consists of characteristics and discovery data, setup information from a system designer, raw and processed sensed data, and actuation information. The ISO/IEC 23093 series specifies input and output data formats for media sensors, media actuators, media storages, media analysers, etc. Media analysers can process sensed data from media sensors to produce analysed data, and the media analysers can be cascaded in order to extract semantic information.

This document contains the APIs to discover media things in the network and communication between media things and the APIs to facilitate transactions between media things.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

Information technology — Internet of media things —

Part 2:
Discovery and communication API

1 Scope

This document specifies the abstract class of a media thing (MThing), which is a basic component to construct the Internet of media things. The MThing class contains the basic APIs to:

— discover other MThing(s) in the network;

— connect/disconnect MThing(s);

— support transactions (e.g. payments) using media tokens between MThings.

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/IEC 23093-1, *Information technology — Internet of media things — Part 1: Architecture*

ISO/IEC 23093-3:2019, *Information technology — Internet of media things — Part 3: Media data formats and API*

ISO/IEC 21000-7:2007, *Information technology — Multimedia framework (MPEG-21) — Part 7: Digital Item Adaptation*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23093-1 apply.

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

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

4 APIs

4.1 General

This clause specifies APIs to discover MThings, and connect/disconnect communication between MThings. Besides, APIs and return class types are specified to provide MThing information and hardware descriptions.

An MThing can be discovered by its capabilities or supported media token types. The discovered MThing(s) can then relay its (their) information to the requester (i.e. another MThing).

Figure 1 shows the process to discover MThings in the network by a required capability. Each MThing, which supports the required capability, can send back its information. In the figure, an MThing

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

broadcasts a message using `discoverMThingByCapability()` to look for MSensors capable of capturing audio (Figure 1, item 1). Each MSensor (e.g. MMicrophone1, MMicrophone2, MCamera1) sends back its information using `sendBackMThingInfo()`, which returns the data format `MThingInfo` specified in ISO/IEC 23093-3 (Figure 1, item 2).

Figure 2 shows the process of connecting MThings with a capability. First, an MThing (i.e. ReqMThing) can ask the availability of the specific capability (e.g. SENSOR_CAPTURE_AUDIO) with the function `isCapabilityAvailable()` to another MThing (Figure 2, item 1). If the capability "SENSOR_CAPTURE_AUDIO" is currently unavailable, the MThing (i.e. MMicrophone1) notifies "unavailable" (Figure 2, item 2). Then, the ReqMThing can ask the availability of the specific capability again to other MThings (Figure 2, item 3) with the binary representation of "MSensor" and "SENSOR_CAPTURE_AUDIO". Because the MCamera1 notifies the ReqMThing that its "SENSOR_CAPTURE_AUDIO" capability is available (Figure 2, item 4), the ReqMThing can connect to the camera and reserve its "SENSOR_CAPTURE_AUDIO" capability (Figure 2, item 5). As long as the capability "SENSOR_CAPTURE_AUDIO" of the MCamera1 is used by the ReqMThing, other MThings cannot access the corresponding capability.

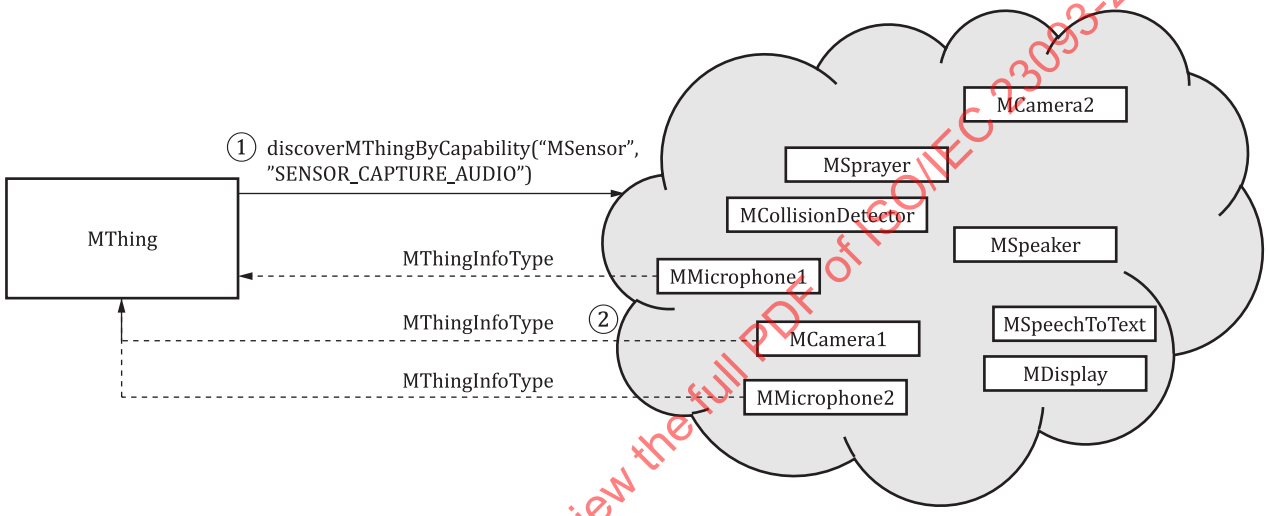


Figure 1 — Discovering MThings by capability

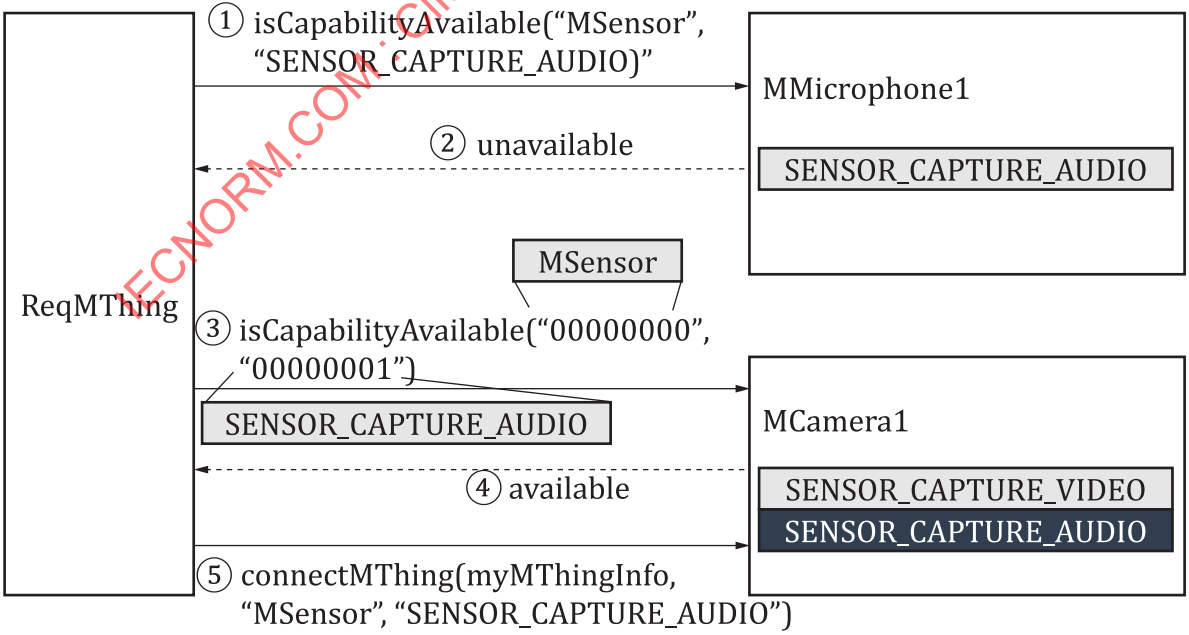


Figure 2 — Connecting MThings based on a capability

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

Figure 3 shows the process of disconnecting (i.e. releasing capabilities of) an MThing. The ReqMThing can release either all of its reserved capabilities of the MCamera (Figure 3, items 1 and 2) or a designated capability (e.g. SENSOR_CAPTURE_AUDIO) (Figure 3, items 3 and 4) using `disconnectMThing()`.

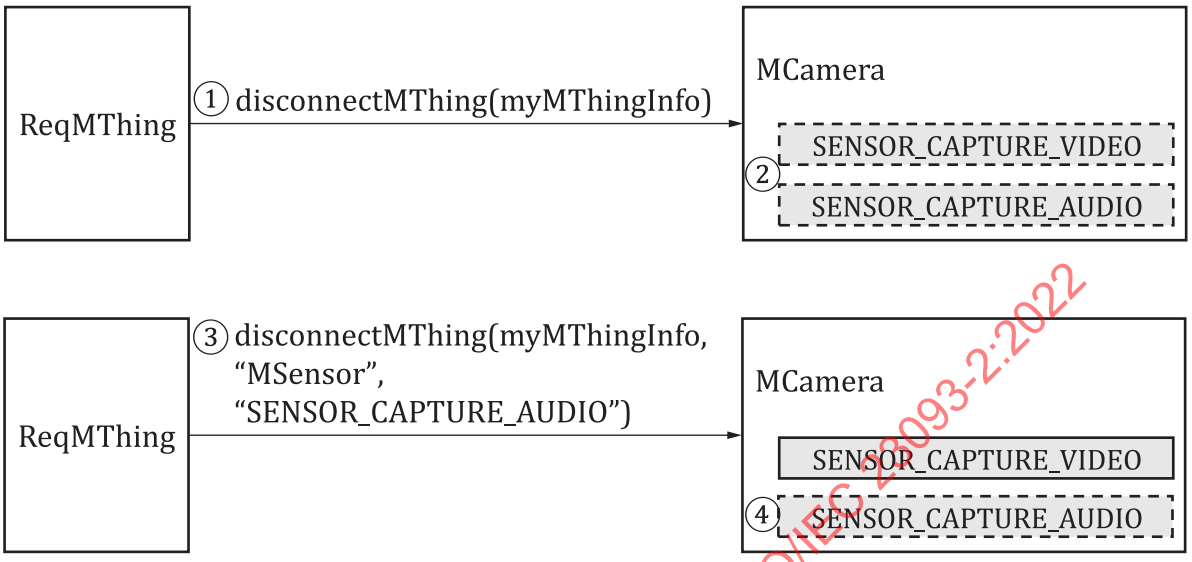


Figure 3 — Disconnecting an MThing

Figure 4 shows the process where an MThing alerts other MThings that all or some of its capabilities will be unavailable. The MCamera can warn the MThing_01 that either all of the capabilities reserved by MThing_01 are unavailable (Figure 4, items 1 and 2) or a designated capability (e.g. SENSOR_CAPTURE_AUDIO) is no longer available to MThing_01 (Figure 4, items 3 and 4) using `alertDisconnection()`.

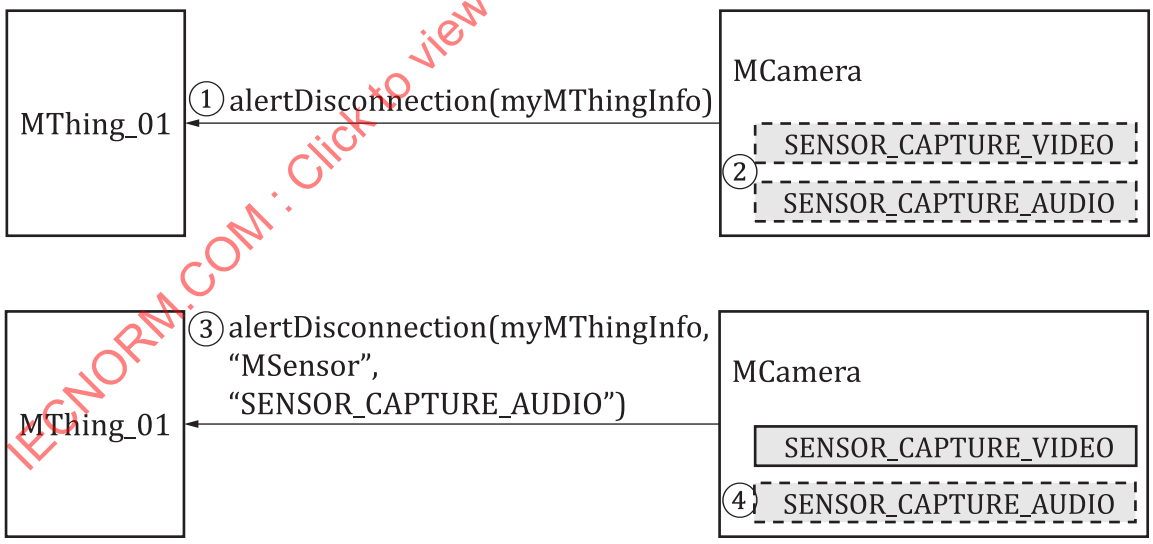


Figure 4 — Alerting disconnection to an MThing

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

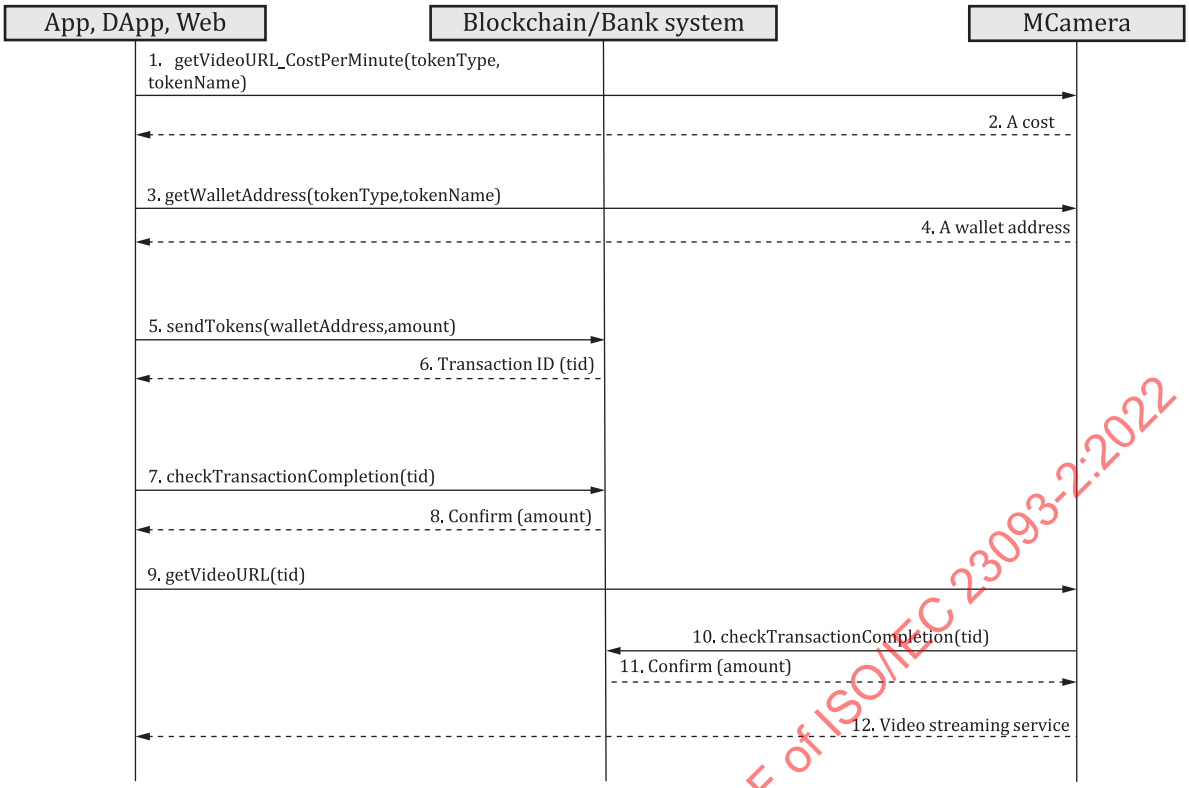


Figure 5 — Transaction process between a user and an MThing

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

Figure 5 shows a sequence diagram of a transaction process between a user and an MThing. Suppose that a user wants to watch a video captured from a nearby camera (i.e. MCamera). The user asks a cost per minute to watch a video taken by the MCamera using `getVideoURL_CostPerMinute()`. Using the user interface of Apps, DApps, or Web (Figure 5, item 1), the user can send desired media token defined by `tokenType` (e.g. cryptocurrency or legal tender) and `tokenName` (e.g. Bitcoin or US dollar). The MCamera returns the cost per minute to let the user use the `getVideoURL()` function (Figure 5, item 2). If the user wants to watch the video with the responded price, they ask for a wallet address (Figure 5, item 3) of the desired MToken. Again, the MCamera responds with the proper wallet address (Figure 5, item 4). The user sends some MTokens to the wallet address through a payment system like a blockchain or a banking system (Figure 5, item 5) which returns a transaction ID (Figure 5, item 6). With this transaction ID (i.e. `tid`), the user can confirm (Figure 5, item 8) whether the token transaction was completed to the designated wallet address or not (Figure 5, item 7) using `checkTransactionCompletion(tid)`. Once the transaction is confirmed, the user can ask for the video stream service to the MCamera using `getVideoURL(tid)` (Figure 5, item 9). The MCamera checks for the completion of the token transaction using `checkTransactionCompletion(tid)` (Figure 5, item 10) and the amount of MTokens received (Figure 5, item 11). The MCamera returns a video URL and streams the video as much as the user paid (Figure 5, item 12). The details of APIs, `getVideoURL_CostPerMinute()` and `getVideoURL()` are given in ISO/IEC 23093-3.

Figure 6 shows a sequence diagram of a transaction process between MThings. Suppose that a camera (i.e. MCamera) wants to stream a captured video to a nearby display (i.e. MDisplay) and play it. The MCamera asks a cost per minute to play a video on the MDisplay using `setVideoURL_CostPerMinute()` (Figure 6, item 1) with desired media token defined by `tokenType` (e.g. cryptocurrency or legal tender) and `tokenName` (e.g. Bitcoin or US dollar). The MDisplay returns the cost per minute to use the `setVideoURL()` function (Figure 6, item 2). If the MCamera wants to play the video with the responded price, it asks for a wallet address (Figure 6, item 3) of the desired MToken. Again, the MDisplay responses with the proper wallet address (Figure 6, item 4). The MCamera sends some MTokens to the wallet address through a payment system like a blockchain or a banking system (Figure 6, item 5) which returns a transaction ID (Figure 6, item 6). With this transaction ID (i.e. `tid`), the MCamera can confirm (Figure 6, item 8) whether the token transaction was completed to the designated wallet address or not (Figure 6, item 7)

© ISO/IEC 2022 – All rights reserved

using `checkTransactionCompletion(tid)`. Once the transaction is confirmed, the MCamera can ask for the video play service to the MDisplay using `setVideoURL(tid, url)` (Figure 6, item 9). The MDisplay checks for the completion of the token transaction using `checkTransactionCompletion(tid)` (Figure 6, item 10) and the amount of MTokens received (Figure 6, item 11). The MDisplay plays the video using the received video URL as much as the MCamera paid and notifies the completion of the video playing service (Figure 6, item 12). The details of APIs, `setVideoURL_CostPerMinute()` and `setVideoURL()` are given in ISO/IEC 23093-3.

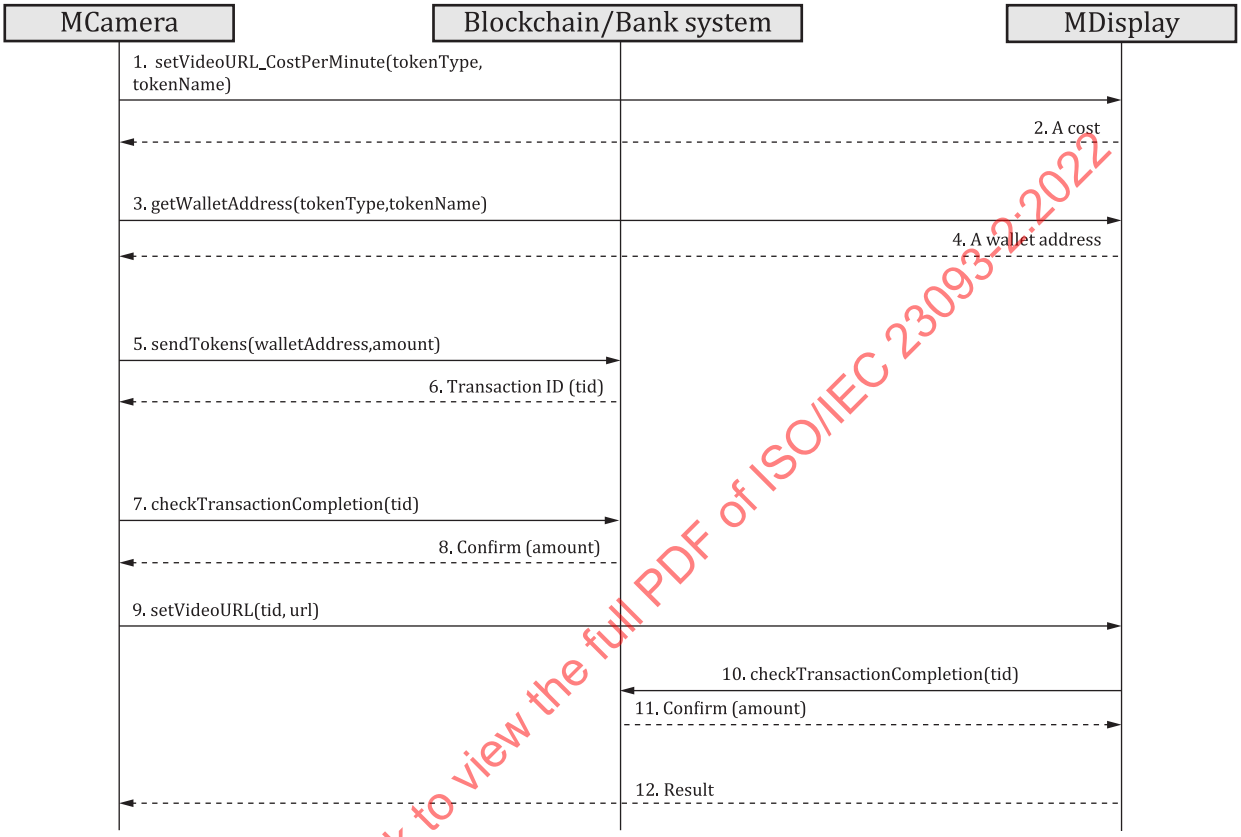
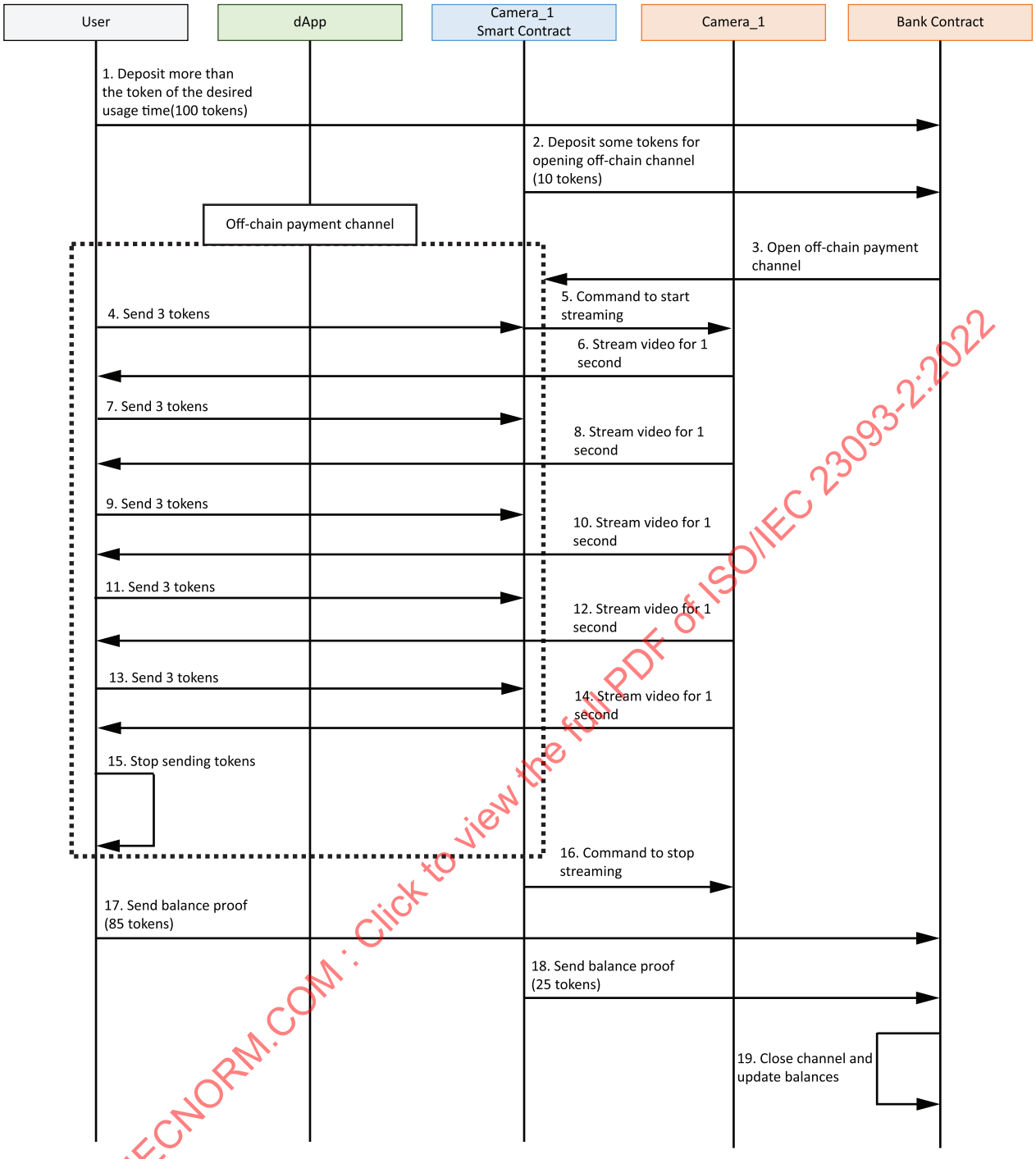


Figure 6 — Transaction process between MThings

The transaction processes presented in Figures 5 and 6 lack counting on the error-resilient design against any unexpected service interruption. Figure 7 shows the scenario of applying a state channel method to the IoT camera video streaming system to overcome any unexpected service interruption.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved



IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

Figure 7 — Transaction scenario using off-chain state channels

The smart contracts of the user and the camera each send a deposit to the bank contract on the main blockchain to open the state channel. To receive video streaming for 30 seconds (e.g. three tokens/s), the user deposits 100 tokens, which are more than 90 tokens (Figure 7, item 1). The smart contract of MCamera deposits ten tokens to establish a channel (Figure 7, item 2). After confirming the deposit, the bank contract opens the state channel (Figure 7, item 3). The user starts transmitting three tokens per second over the state channel (Figure 7, item 4). After confirming the token deposit, the camera's smart contract instructs the camera to stream (Figure 7, item 5), and the camera starts streaming one second of video (Figure 7, item 6). The user continues to transmit three tokens per second (Figure 7, items 7, 9, 11, 13), and the camera continues to stream one second of video (Figure 7, items 8, 10, 12, 14). If the token transfer is interrupted for some unexpected reason (Figure 7, item 15), the smart contract will immediately stop streaming (Figure 7, item 16). The user and the camera's smart contract send their

© ISO/IEC 2022 – All rights reserved

proof of equity to the bank contract on the main blockchain (Figure 7, items 17, 18). Bank contracts verify their shares and close the state channel (Figure 7, item 19).

Existing blockchains take a considerable time for transactions to be registered and agreed upon in a block. Therefore, the current payment model is designed to pay the amount of service time in advance. The refund process is very complicated and unsafe when an unexpected situation arises and the service is not adequately received. However, the payment model using the state channel can execute the transaction in real-time, so it does not pay for the service at once. As a result, errors can be detected and reacted to in real-time. If the token is continuously transmitted through the state channel and the token transmission continues, the MCamera continues the service. If an unexpected situation occurs and the token transmission is stopped, the MCamera may stop streaming immediately. Since the service is provided as long as the payment is made in real-time, there is no need to refund the service through the refund process. Also, if the MCamera does not provide the service properly, the service user can stop the token transfer immediately.

4.2 Abstract Class of MThing

4.2.1 General

The class `MThing` is a base class of all kinds of MThings. This class includes the essential fields and functions of an MThing.

MThing APIs

Table 1 presents a class for MThing.

Table 1 — MThing class

Nested Classes	
Modifier and Type	Method and Description
Constructor	
Constructor and Description	
MThing()	
Default constructor.	
MThing(string id)	
MThing(string id, string ipAddress, int port)	
Fields	
Modifier and Type	Field and Description
string	id
	Unique identifier.
string	ipAddress
int	port
Methods	
Modifier and Type	Method and Description
Type	get()

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

Table 1 *(continued)*

void	set(Type)
List< MThingInfoType >	discoverMThingByCapability(string MThingType, string capability)
	This function returns a list of MThings, which support the capability required. The `MThingType` and `capability` (i.e. formal parameters) shall be described in a string (e.g. term ID or binary representation) from `MThingTypeCS` and `capabilityCS` following ISO/IEC 23093-3:2019, A.1 and A.2, respectively.
List< MThingInfoType >	discoverMThingByToken(int tokenType, string tokenName)
	This function returns a list of MThings, which support the token type and name. If `tokenType` is 0, it denotes "cryptocurrency", and if `tokenType` is 1, it denotes "legal tender". The `tokenName` shall be described in a string (e.g. term ID or binary representation) from `TokenTypeCS` following ISO/IEC 23093-3:2019, A.5. Ex) dicoverMThingByToken(0, "BTC") or dicoverMThingByToken(0, "00000001") Ex) dicoverMThingByToken(1, "USD") or dicoverMThingByToken(1, "10010100")
int	isCapabilityAvailable(string MThingType, string capability)
	This function asks if the designated `capability` with `MThingType` is available. The `MThingType` and `capability` (i.e. formal parameters) shall be described in a string (e.g. term ID or binary representation) from `MThingTypeCS` and `capabilityCS` following ISO/IEC 23093-3:2019, A.1 and A.2, respectively. If the `capability` is available, it returns 1; otherwise, it returns 0. Ex) isCapabilityAvailable("MSensor", "SENSOR_CAPTURE_AUDIO") or isCapabilityAvailable("00000000", "00000001")
string	getWalletAddress(int tokenType, string tokenName)
	This function returns a wallet address according to the token type and the token name requested. If `tokenType` is 0, it denotes "cryptocurrency", and if `tokenType` is 1, it denotes "legal tender". The `tokenName` shall be described in a string (e.g. term ID or binary representation) from `TokenTypeCS` following ISO/IEC 23093-3:2019, A.5. The function returns a wallet address if the task succeeds or null if there is no wallet address for the designated `tokenType` and `tokenName`. Ex) getWalletAddress(0, "BTC") or getWalletAddress(0, "00000001") Ex) getWalletAddress(1, "USD") or getWalletAddress(1, "10010100")
string	sendTokens(string walletAddress, float amount)
	This function transfers tokens to a receiver MThing using its wallet address. The `amount` of tokens is transferred to the `walletAddress`. The function returns a transaction ID if the task succeeds or returns null if the task fails.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

Table 1 *(continued)*

float	checkTransactionCompletion(string tid)
	This function checks the completion of a transaction with the transaction ID, `tid`. An MThing (e.g. a token sender) and its corresponding MThing (e.g. a token receiver) can check whether the (token) transaction with the transaction ID is completed or not. It returns the number of tokens if the transaction has been completed, returns -1, otherwise.
int	connectMThing(MThingInfoType myMThingInfo)
	This function reserves all available capabilities of the target MThing by sending the (requester) MThing information, `myMThingInfo`. If the task succeeds, returns 1, if fails, returns 0, and returns -1 if the targetMThing does not respond.
int	connectMThing(MThingInfoType myMThingInfo, string targetMThingType, string targetCapability)
	This function reserves the designated capability of the target MThing by sending the (requester) MThing information, `myMThingInfo`. The `targetMThingType` and `targetCapability` shall be described in a string (e.g. term ID or binary representation) from `MThingTypeCS` and `capabilityCS` following ISO/IEC 23093-3:2019, A.1 and A.2, respectively. If the task succeeds, returns 1, if fails, returns 0, and returns -1 if the targetMThing does not respond. Ex) connectMThing(myMThingInfo, “MSensor”, “SENSOR_CAPTURE_AUDIO”); or connectMThing(myMThingInfo, “00000000”, “00000001”);
int	disconnectMThing(MThingInfoType myMThingInfo)
	This function releases all reserved capabilities of the target MThing by sending the (requester) MThing information, `myMThingInfo`. If the task succeeds, returns 1, if fails, returns 0, and returns -1 if the targetMThing does not respond.
int	disconnectMThing(MThingInfoType myMThingInfo, string targetMThingType, string targetCapability)
	This function releases the designated capability of the target MThing by sending the (requester) MThing information, `myMThingInfo`. The `targetMThingType` and `targetCapability` shall be described in a string (e.g. term ID or binary representation) from `MThingTypeCS` and `capabilityCS` following ISO/IEC 23093-3:2019, A.1 and A.2, respectively. If the task succeeds, returns 1, if fails, returns 0, and returns -1 if the targetMThing does not respond. Ex) disconnectMThing(myMThingInfo, “MSensor”, “SENSOR_CAPTURE_AUDIO”); or disconnectMThing(myMThingInfo, “00000000”, “00000001”);
int	alertDisconnection(MThingInfoType myMThingInfo)
	This function alerts a disconnection of all reserved capabilities of the requester MThing to the target MThing by sending the (requester) MThing information, `myMThingInfo`. If the task succeeds, returns 1, if fails, returns 0, and returns -1 if the targetMThing does not respond.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

Table 1 *(continued)*

int	alertDisconnection(MThingInfoType myMThingInfo, string myMThingType, string myCapability)
	This function alerts a disconnection of the designated capability of a requester MThing to a target MThing by sending the (requester) MThing information, `myMThingInfo`. The `myMThingType` and `myCapability` shall be described in a string (e.g. term ID or binary representation) from `MThingTypeCS` and `capabilityCS` following ISO/IEC 23093-3:2019, A.1 and A.2, respectively. If the task succeeds, returns 1, if fails, returns 0, and returns -1 if the targetMThing does not respond. Ex) alertDisconnectCapability(myMThingInfo, “MSensor”, “SENSOR_CAPTURE_AUDIO”); or alertDisconnectCapability(myMThingInfo, “00000000”, “00000001”);
MPEG21TerminalCapabilityType	getMPEG21TerminalCapabilityList()
	This function produces hardware descriptions that shall be described following ISO/IEC 21000-7 `TerminalCapabilityType`. The returned type, `MPEG21TerminalCapabilityType`, is defined in the following section.

4.3 Return type class

4.3.1 MThingInfoType

4.3.1.1 General

This subclause specifies the return class types of MThing information. For easy industrial adoption, the return type classes are provided in JAVA, C++, and C.

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved

4.3.1.2 JAVA class

Table 2 presents return class types of MThing information in JAVA.

Table 2 — Return class types of MThing information in JAVA

Constructor	
Constructor and Description	
MThingInfoType()	
Default constructor.	
MThingInfoType(int type, string MThingInfo)	
Fields	
Modifier and Type	Field and Description
int	type
	A returning type of MThing Information. 0 = None, 1 = XML, 2 = Binary
string	MThingInfo
	MThing information shall be described by the data format of `MThingInfo` following ISO/IEC 23093-3.
Methods	
Modifier and Type	Method and Description
int	getType()
void	setType(int type)
string	getMThingInfo()
void	setMThingInfo(string MThingInfo)

4.3.1.3 C++ class

Table 3 presents return class types of MThing information in C++.

Table 3 — Return class types of MThing information in C++

Constructor and Destructor	
Constructor and Destructor Description	
MThingInfoType()	
Default constructor.	
MThingInfoType(int type, string MThingInfo)	
~ MThingInfoType()	
Default destructor.	
Member Data	
Modifier and Type	Member Data and Description
int	type

IECNORM.COM : Click to view the full PDF of ISO/IEC 23093-2:2022

© ISO/IEC 2022 – All rights reserved