INTERNATIONAL STANDARD

ISO/IEC 23093-3

First edition
2019-11

Information technology — Internet of media things —

Part 3: Media data formats and APIs

Technologies de l'information — Internet des objets media —

Partie 3: API et formats des données

Reference number
ISO/IEC 23093-3:2019(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

Page

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).

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 http://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.

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*.

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

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.

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 provide means for supporting transaction tokens in order 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 data formats of input and output for media sensors, media actuators, media storages, media analyzers, etc. Sensed data from media sensors can be processed by media analyzers to produce analysed data, and the media analyzers can be cascaded in order to extract semantic information.

This document contains the tools to describe data exchanged between media things (e.g. media sensors, media actuators, media analyzers, media storages) and their APIs. It addresses the normative aspects of the data and APIs for media things and also illustrates non-normative examples.

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

ISO and the IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured the 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 statements of the holders of these patents right are 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 identified in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Internet of media things —

Part 3:
Media data formats and APIs

1 Scope

This document specifies syntax and semantics of description schemes to represent data exchanged by media things (e.g. media sensors, media actuators, media analyzers, media storages). Moreover, it specifies the APIs to exchange these data between media things.

This document does not specify how the process of sensing and analyzing is carried out but specifies the interfaces between the media things.

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 15938-5:2003, *Information technology — Multimedia content description interface — Part 5: Multimedia description schemes*

ISO/IEC 23005-2, *Information technology — Media context and control — Part 2: Control information*

ISO/IEC 23005-5, *Information technology — Media context and control — Part 5: Data formats for interaction devices*

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

ISO/IEC 23093-2, *Information technology — Internet of media things — Part 2: Discovery and communication API*

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 23093-1 and 23093-2 and the following apply.

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

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

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

3.1.1
media actuator
MActuator
MThing that can actuate

3.1.2
media aggregator
MAggregator
MThing that contains multiple MThings

3.1.3
media analyzer
MAnalyzer
MThing that can analyze media or metadata, and produce interpreted media, metadata, or commands

3.1.4
media manager
MManager
MThing that can register a list of MThings or be facilitated to search other MThings

3.1.5
media sensor
MSensor
MThing that can sense and produce media data

3.1.6
media storage
MStorage
MThing that can save media or metadata

3.2 Abbreviated terms

API application programming interface

MACV media actuator command vocabulary

MAOV media analyzer output vocabulary

MSOV media sensor output vocabulary

MTDL media thing description language

SCDV sensor capability description vocabulary

XML extensible mark-up language

XSI XML streaming instructions

3.3 Schema documents

In the main text of this document, the syntax of description schemes and descriptors is provided whenever possible as a single schema document.

In some cases though, and in particular for Clauses 6, 7 and 8, the syntax of description schemes and descriptors is provided as a collection of schema snippets imbricated with other text. In order to form a valid schema document, these schema components are intended to be gathered in the same document

with the schema wrapper provided at the head of the clause. For better readability, the relevant schema documents are provided in Annex B.

In all cases, each schema document has a `version` attribute, the value of which is "ISO/IEC 23093-3". Furthermore, an informative identifier is given as the value of the `id` attribute of the `schema` component. This identifier is non-normative and used as a convention in this document to reference another schema document. In particular, it is used for the `schemaLocation` attribute of the `include` and `import` schema components.

3.4 Use of prefixes

For clarity, throughout this document, consistent namespace prefixes are used.

"`xsi:`" prefix is not normative. It is a naming convention in this document to refer to an element of the `http://www.w3.org/2001/XMLSchema-instance` namespace.

"`xml:`" and "`xmlns:`" are normative prefixes defined in Reference [1]. The prefix "`xml:`" is by definition bound to "`http://www.w3.org/XML/1998/namespace`". The prefix "`xmlns:`" is used only for namespace bindings and is not itself bound to any namespace name.

All other prefixes used in either the text or examples of this document are not normative, e.g. "`mtdl:`", "`msov:`", "`macv:`", "`maov:`", "`mpeg7:`", "`scdv:`".

In particular, most of the informative examples in this document are provided as XML fragments without the normally required XML document declaration and, thus, miss a correct namespace binding context declaration. In these descriptions fragments, the different prefixes are bound to the `namespaces` as given in Table 1.

Table 1 — Mapping of prefixes to namespaces in examples and text

Prefix	Corresponding namespace
scdv	urn:mpeg:mpeg-v:2018:01-SCDV-NS
mpeg7	urn:mpeg:mpeg7:schema:2004
mtdl	urn:mpeg:mpeg-IoMT:2018:01-MTDL-NS
msov	urn:mpeg:mpeg-IoMT:2018:01-MSOV-NS
macv	urn:mpeg:mpeg-IoMT:2018:01-MACV-NS
maov	urn:mpeg:mpeg-IoMT:2018:01-MAOV-NS
xsi	http://www.w3.org/2001/XMLSchema-instance
xsd	http://www.w3.org/2001/XMLSchema

Unlike the informative descriptions examples, the normative specification of the syntax of tools in XML schema follows the namespace binding context defined in the relevant schema declaration such as the one defined in 6.2.

4 APIs

4.1 General

This subclause specifies APIs and their descriptions to operate MThings and/or exchange structured data between MThings. Figure 1 shows an example of "GET" and "SET" functions invoked between

MThings. For example, an MSensor should have “GET” functions to evoke and provide its sensed data. An MStorage should have “SET” functions to save sensed data obtained by an MSensor or to save analyzed data provided by an MAnalyzer. An MAnalyzer should provide “GET” functions to produce metadata by analyzing sensed data from MSensors or to generate metadata by analyzing data fed by other MAnalyzers. Finally, an MActuator should provide “SET” functions to control its functionalities. If there is no structured data exchanged between MThings, each MThing can have simple “SET” functions to be controlled by other MThings.

Figure 2 demonstrates an example of a function call sequence diagram between MThings. A speech recognizer (AZ1) requests an audio data to a microphone (S1) by invoking the function `getAudioURL()`. The microphone (S1) sends back the corresponding URL to the speech recognizer (AZ1). In this case, the return type of the URL is a simple string. If, however, an MSensor returns data with standard structures, the data can be delivered by the return type class either “MPEGVSensedDataType” or “IoMT SensedDataType”, which can be described by XML, JSON, or Binary representation.

A text-to-speech converter (AZ2) requests speech texts extracted from the audio data (i.e. sensed data from S1) to the speech recognizer (AZ1) by invoking the function `getSpeechText()`. The speech recognizer (AZ1) sends back the recognized speech texts with the standard structure to the text-to-speech converter (AZ2). The data provided by an MAnalyzer can be delivered by either a simple string like a URL or the return type class called “IoMT AnalyzedDataType”, which can be described by XML, JSON, or Binary representation.

Finally, the text-to-speech converter (AZ2) invokes the function `setAudioURL()` and the function `setPlay()` to actuate (i.e. generate the audible speech sound) the speaker (AC1). Again, the actuation data feeding to an MActuator can be delivered by either a simple string like a URL or the return type class of “MPEGVCommandType” or “IoMT ActuationDataType”, which can be described by XML, JSON, or Binary representation.

The function calls trigger MThings either to generate and exchange data or to control MThings.

The function definitions (APIs) are defined for MSensor, MActuator, MAnalyzer, MStorage, MManager, MAggregator, and their return type classes in the following subclauses.

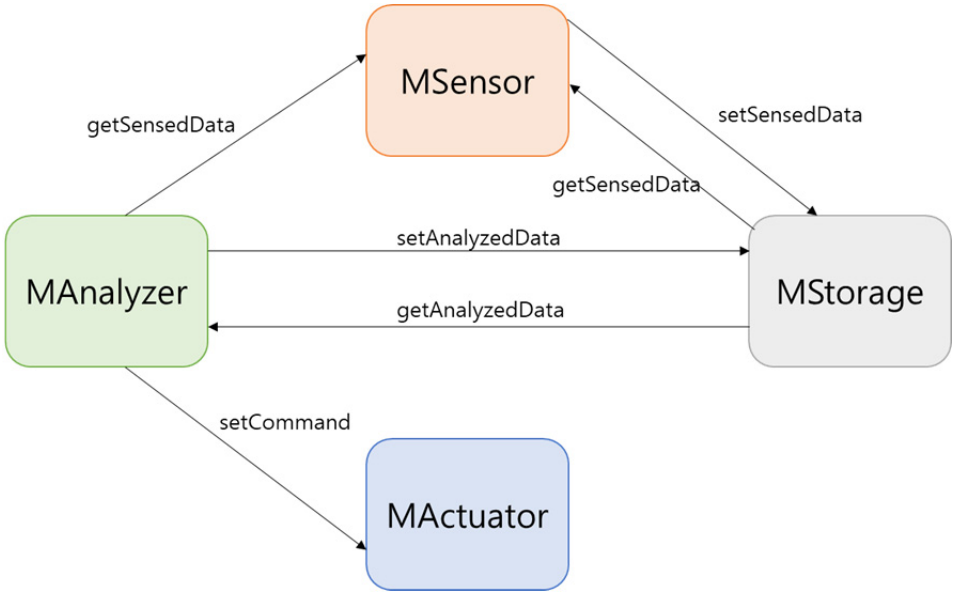

Figure 1 — Function invocation between MThings

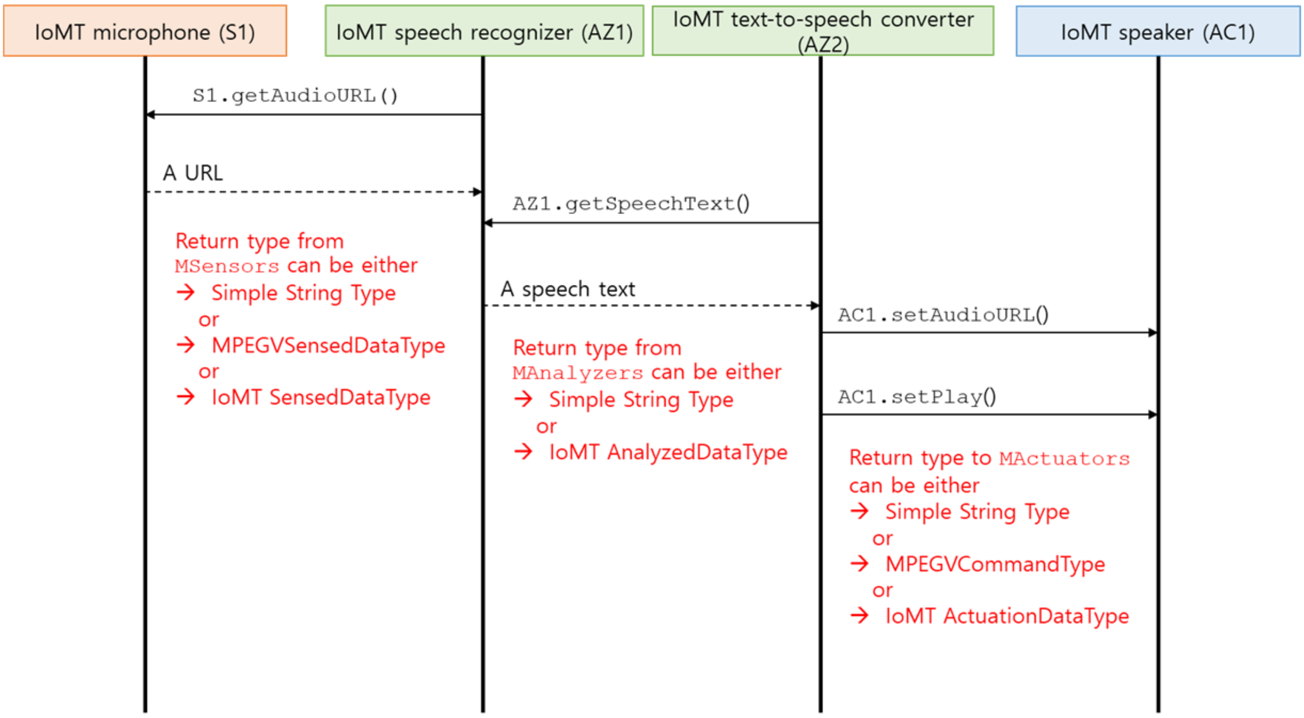

Figure 2 — Sequence diagram of function calls between MThings

4.2 APIs for IoMT sensors

4.2.1 General

This subclause defines API classes of IoMT sensors.

4.2.2 MSensor class

General

This subclause defines an MSensor class, which shall inherit the features of MThing class defined in ISO/IEC 23093-2.

APIs

Table 2 presents basic APIs of MSensor.

Table 2 — MSensor APIs

Nested Classes	
Modifier and Type	Method and Description
Constructor	
Constructor and Description	
MSensor()	
Default constructor.	
MSensor(String id)	
MSensor(String id, String serverIPAddress, integer serverPort)	
Fields	
Modifier and Type	Field and Description
Methods	
Modifier and Type	Method and Description
MPEGVCapabilityType	getMPEGVCapability()
	This function returns a class (i.e. Java or C++) or a structure (i.e. C) that shall include a returning type (e.g. XML, Binary) and sensor capabilities from ISO/IEC 23005-2 (MPEG-V Part 2).
MPEGVSensedDataType	getMPEGVSendsedData()
	This function returns a class (i.e. Java or C++) or a structure (i.e. C) that shall include a returning type (e.g. XML, Binary) and sensed data from ISO/IEC 23005-5 (MPEG-V Part 5).
CapabilityListType	getSensorCapabilityList();
	This function returns a class (i.e. Java or C++) or a structure (i.e. C) that shall include a returning type (e.g. XML, Binary) and a capability list specified in this document.

CapabilityListType	getAvailableSensorCapabilityList()
	This function returns a class (i.e. Java or C++) or a structure (i.e. C) that shall include a returning type (e.g. XML, Binary) and an available capability list specified in this document.
CapabilityListType	getAppliedSensorCapabilityList()
	This function returns a class (i.e. Java or C++) or a structure (i.e. C) that shall include a returning type (e.g. XML, Binary) and an applied capability list specified in this document.
SensedDataType	getCapturedTime()
	This function returns a captured time of sensed data.
SensedDataType	getCapturedTime(String tid)
	This function returns a captured time of sensed data. The `tid` *is the transaction ID of a payment for using this function.*
float	getCapturedTime_Cost(int tokenType, String tokenName)
	This function returns the amount of tokens to use `getCapturedTime()`*. If* `tokenType` *is 0, it denotes "cryptocurrency", if* tokenType *is 1, it denotes "legal tender". The* `tokenName` *is described in string (e.g. term ID or binary representation) from* `TokenTypeCS` *specified in A.5. If the requested token is not supported, returns –1.* Ex) getCapturedTime_Cost(0, "BTC") or getCapturedTime_Cost(0, "00000001") Ex) getCapturedTime_Cost(1, "USD") or getCapturedTime_Cost(1, "10010100")

4.2.3 API for IoMT microphone

General

This subclause defines a class of an IoMT microphone which shall inherit the features of `MSensor` class.

APIs

Table 3 presents APIs of an IoMT microphone.