ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Recommendation

ITU-T H.627.3 (12/2022)

SERIES H: Audiovisual and multimedia systems

Broadband, triple-play and advanced multimedia services – Advanced multimedia services and applications

---

Protocols for intelligent video surveillance systems

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.349
Directory services architecture for audiovisual and multimedia services	H.350–H.359
Quality of service architecture for audiovisual and multimedia services	H.360–H.369
Telepresence, immersive environments, virtual and extended reality	H.420–H.439
Supplementary services for multimedia	H.450–H.499
MOBILITY AND COLLABORATION PROCEDURES	
Overview of Mobility and Collaboration, definitions, protocols and procedures	H.500–H.509
Mobility for H-Series multimedia systems and services	H.510–H.519
Mobile multimedia collaboration applications and services	H.520–H.529
Security for mobile multimedia systems and services	H.530–H.539
Security for mobile multimedia collaboration applications and services	H.540–H.549
VEHICULAR GATEWAYS AND INTELLIGENT TRANSPORTATION SYSTEMS (ITS)	
Architecture for vehicular gateways	H.550–H.559
Vehicular gateway interfaces	H.560–H.569
BROADBAND, TRIPLE-PLAY AND ADVANCED MULTIMEDIA SERVICES	
Broadband multimedia services over VDSL	H.610–H.619
Advanced multimedia services and applications	**H.620–H.629**
Content delivery and ubiquitous sensor network applications	H.640–H.649
IPTV MULTIMEDIA SERVICES AND APPLICATIONS FOR IPTV	
General aspects	H.700–H.719
IPTV terminal devices	H.720–H.729
IPTV middleware	H.730–H.739
IPTV application event handling	H.740–H.749
IPTV metadata	H.750–H.759
IPTV multimedia application frameworks	H.760–H.769
IPTV service discovery up to consumption	H.770–H.779
Digital Signage	H.780–H.789
E-HEALTH MULTIMEDIA SYSTEMS, SERVICES AND APPLICATIONS	
Personal health systems	H.810–H.819
Interoperability compliance testing of personal health systems (HRN, PAN, LAN, TAN and WAN)	H.820–H.859
Multimedia e-health data exchange services	H.860–H.869
Safe listening	H.870–H.879

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T H.627.3

Protocols for intelligent video surveillance systems

Summary

Recommendation ITU-T H.627.3 defines protocols for intelligent video surveillance systems, including the functional architecture, functional interface, overall requirements of the protocol, message flows and relevant protocols. This Recommendation is based on Recommendation ITU-T H.626.5, *Architecture for intelligent video surveillance systems*.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T H.627.3	2022-12-14	16	11.1002/1000/15204

Keywords

Intelligent video surveillance, protocol.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

		Page
1	Scope	1
2	References	1
3	Definitions	1
	3.1 Terms defined elsewhere	1
	3.2 Terms defined in this Recommendation	2
4	Abbreviations and acronyms	3
5	Conventions	3
6	Introduction to functional architecture of an IVS system	4
7	Overall requirements	4
	7.1 Introduction of relevant protocol	4
	7.2 Interface description	7
8	Message flows and relevant protocols	16
	8.1 Message flow	16
	8.2 Interface definition	19
Annex A – Data type and object JSON-Schema description		36
	A.1 Data type	36
	A.2 JSON-Schema description	38
Bibliography		74

Recommendation ITU-T H.627.3

Protocols for intelligent video surveillance systems

1 Scope

This Recommendation defines protocols for intelligent video surveillance (IVS) systems, including the functional architecture, functional interface, overall requirements of the protocol, message flows and relevant protocols. The protocols of the video surveillance (VS) system refer to [ITU-T H.627].

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T H.626.5] Recommendation ITU-T H.626.5 (V2) (2022), *Architecture for intelligent video surveillance system*.

[ITU-T H.627] Recommendation ITU-T H.627 (V2) (2020), *Signalling and protocols for a video surveillance system.*

[IETF RFC 2045] IETF RFC 2045 (1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies.*

[IETF RFC 2046] IETF RFC 2046 (1996), *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types.*

[IETF RFC 3339] IETF RFC 3339 (2002), *Date and Time on the Internet: Timestamps.*

[IETF RFC 3986] IETF RFC 3986 (2005), *Uniform Resource Identifier (URI): Generic Syntax.*

[IETF RFC 7616] IETF RFC 7616 (2015), *HTTP Digest Access Authentication.*

[IETF RFC 9110] IETF RFC 9110 (2022), *HTTP Semantics.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 centre management unit (CMU) [ITU-T H.626.5]: The centre management unit (CMU) is located at the centre of the video surveillance (VS) system. Its main functions include centralized system management, service operation management, etc.

3.1.2 client intelligent video (CIV) [ITU-T H.626.5]: An intelligent identification module in the customer unit (CU). It identifies required information from input video and outputs the result and retrieves the recorded video data with specified information.

3.1.3 data storage and service (DSS) [ITU-T H.626.5]: The storage and sharing centre for video and image information within the intelligent video surveillance system. The DSS accepts registration, keepalive reports from the intelligent premises unit (IPU), intelligent video management (IVM) or intelligent application service (IAS) and obtains video and image information from them. It can also

provide information network sharing services such as information querying, subscription, guard and alarm, etc.

3.1.4 intelligent application service (IAS) [ITU-T H.626.5]: The intelligent application service (IAS) is located at the application service core of the intelligent video surveillance system. Its main functions include providing application services about the video and image information, such as information query and retrieval, subscription and notification, manual annotation, etc.; providing external operations such as the creation and start-up of intelligent analysis tasks based on intelligent video management (IVM); obtaining the list of devices from the central control server (CCS) or data storage and service (DSS) and sharing them with the IVM so that it can schedule intelligent video units (IVUs) for video stream analysis, etc.

3.1.5 intelligent customer unit (ICU) [ITU-T H.626.5]: The client subsystem within the intelligent video surveillance system. The client intelligent video (CIV) is added to the customer unit (CU) in order to achieve comprehensive intelligent video analysis.

3.1.6 intelligent premises unit (IPU) [ITU-T H.626.5]: The premises subsystem within the intelligent video surveillance system. The premises intelligent video (PIV) module is added to the premises unit (PU) for intelligent video analysis.

3.1.7 intelligent video management (IVM) [ITU-T H.626.5]: The intelligent video management (IVM) unit supports strategies configuration of intelligent applications by users and video sources schedules dynamically. The IVM accepts registration, deletion, capability reports from intelligent video units (IVUs) and schedules the IVUs dynamically. It can also store, manage, and schedule those capabilities dynamically.

3.1.8 intelligent video unit (IVU) [ITU-T H.626.5]: The intelligent video unit (IVU) identifies specific objects automatically, and outputs recognition results to an intelligent video management (IVM) system. The recognition information includes triggered events and acquired data. One or more intelligent analysis algorithms can be loaded or unloaded on an intelligent video unit (IVU) according to different requirements.

3.1.9 media distribution unit (MDU) [ITU-T H.626.5]: The media distribution unit (MDU) is used to transport media from the premises unit (PU) to the customer unit (CU). Its main functions include media receiving, media processing, media routing, media transmission, media forwarding and media replication.

3.1.10 media storage unit (MSU) [ITU-T H.626.5]: The media storage unit (MSU) is used to retrieve and store media and provide media serving capability. Its main functions include media storage, media serving, media indexing and media downloading.

3.1.11 premises intelligent video (PIV) [ITU-T H.626.5]: An intelligent identification module in the premises unit (PU). It identifies required information from input video and outputs the result.

3.1.12 service-control unit (SCU) [ITU-T H.626.5]: The service-control unit (SCU) is located at the centre of the video surveillance (VS) system. It is a distributed network equipment, fulfilling the access functions of the premises unit (PU) and the customer unit (CU). Its main functions include access registration, access authentication, identification, authorization, call control, location, presence and target media serving function selection.

3.1.13 video and image information [ITU-T H.626.5]: Video clips, images, related files and content description information.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

APE Acquisition and Processing Equipment

API Application Programming Interface

CCS Central Control Server

CIV Client Intelligent Video

CMU Centre Management Unit

CU Customer Unit

DSS Data Storage and Service

HTTP Hyper Text Transfer Protocol

HTTPS Hyper Text Transfer Protocol over Secure Socket Layer

IAS Intelligent Application Service

ICU Intelligent Customer Unit

IPU Intelligent Premises Unit

IVM Intelligent Video Management

IVS Intelligent Video Surveillance

IVU Intelligent Video Unit

JSON JavaScript Object Notation

MDU Media Distribution Unit

MS Media Server

MSU Media Storage Unit

PIV Premises Intelligent Video

PU Premises Unit

REST Representation State Transfer

SBI Searched By Image

SCU Service-Control Unit

URI Universal Resource Identifier

URL Uniform Resource Locator

5 Conventions

In this Recommendation, the following conventions apply:

– The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

– The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

6 Introduction to functional architecture of an IVS system

The functional architecture of an IVS system is illustrated in Figure 6-1. For the detailed entities and functions, see clause 6.2 of [ITU-T H.626.5].

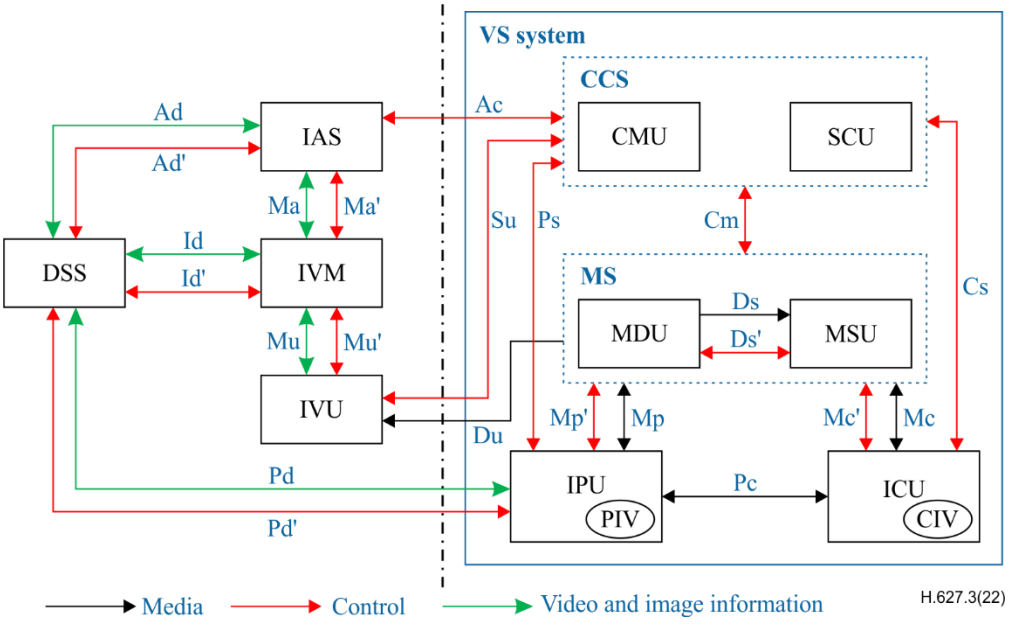

Figure 6-1 – Functional architecture of IVS system

7 Overall requirements

7.1 Introduction of relevant protocol

This Recommendation uses the representation state transfer (REST) style [b-REST] to design the application programming interface (API). The implementation of the protocol based on Hyper Text Transfer Protocol (HTTP) or Hyper Text Transfer Protocol over Secure Socket Layer (HTTPS).

7.1.1 Protocol requirements

7.1.1.1 URI

Universal resource identifier (URI) uses nouns to describe the provided resources. The definition is required to meet the requirements of section 4.2 of [IETF RFC 9110] and support Path Templating. Path Templating refers to the usage of curly brace({}) to mark a section of a uniform resource locator (URL) path as replaceable using path parameters.

7.1.1.2 Request header

The request header field is required to be extended to add <User-Identify>. The field name User-Identify is used to carry the system ID of the requester to identify the requester's identity.

7.1.1.3 Request method

Request method (section 9 of [IETF RFC 9110]) is used to describe operations performed on resources, the specific correspondence is as follows:

a) GET: Read resources, call cannot change the resource state, only return data to the requester without any side effects.

b) POST: Create a resource, it can only be used to add a resource that does not exist before.

c) PUT: Update resources, generally used to update existing resources, if the specified resource does not exist, it can also be created.

d) DELETE: Delete resources.

7.1.1.4 Message body

The encoding method of the message body adopts UTF-8. Under normal circumstances, the interface message body is required to be encapsulated in JavaScript object notation (JSON) format, and the Content-Type header field is required to be set to application/json. When carrying video clips or images, Base64 encodes the data into a string and stores it in JSON. The encoding method is required to meet the requirements of [IETF RFC 2045]; when transmitting large video clip data or image data is recommended to be encapsulated by multipart, and the encapsulation method is required to meet the requirements of section 5.1 of [IETF RFC 2046].

7.1.1.5 Status code

Status code is used to describe the result of operations performed on resources, and its definition is required to meet the requirements of section 15 of [IETF RFC 9110].

7.1.2 Query requirements

The query part of the URI is required to meet the requirements of section 3.4 of [IETF RFC 3986], and is required to support regular expressions.

7.1.2.1 Expansion requirements

This Recommendation extends the query content:

```
query=(o.p1  OP  v1)*&( (o.p2  OP   v2)   OP2   (o.p3   OP   v3) )*
```

In the expression, o indicates the object to be queried, p indicates the property of the object, and * indicates that the preceding () and the content inside appear 0 or more times. Operator OP is required to comply with Table 1, Table 2 and Table 3, OP2 takes the value of AND or OR. The parameters of the object attribute limit, sort field, number of records and other parameters of the result returned by the query is required to comply with the regulations in Table 4.

Table 1 – Arithmetic operators

Operator	Meaning
+	addition
-	subtraction
*	multiplication
/	division
%	Returns the integer remainder of a division operation

Table 2 – Logical operators

Operator	Meaning
AND	If both Boolean expressions are TRUE, then TRUE
OR	If one of the two Boolean expressions is TRUE, then TRUE

Table 3 – Comparison operators

Operator	Meaning
=	equal
>	more than
<	Less than
>=	greater or equal to
<=	Less than or equal to
<>	not equal to
!=	not equal to
!<	not less than
!>	not greater than

Table 4 – Conventions of variable names and meanings

Variable name	Meaning	Description
Fields	Used to specify the returned query property collection	Multiple properties are separated by ","
Sort	Specify sort field	"-" means descending order, otherwise ascending order, placed before the sort field
PageRecordNum	Specify the number of records per page	
RecordStartNo	Specify the start record number of the first page	
MaxNumRecordReturn	Maximum number of records returned	

7.1.2.2 Query example

a) Search for person records with red or pink packages, and the returned results are sorted by the upper age limit, and only two attributes of PersonID and SourceID are returned.

```
GET /Persons?((Person.BagColor = Red) OR (Person.BagColor =
Pink))&(Sort = Person.AgeUpLimit)&(Fields =
(Person.PersonID, Person.SourceID))
```

b) Find person information, only query the first 200 records, 25 records per page, starting from the first record, sort by age upper limit, and return all attributes.

```
GET /Persons?(MaxNumRecordReturn = 200)&(PageRecordNum =
25)&(RecordStartNo = 1) &(Sort = Person.AgeUpLimit)
```

c) Query ApeID contains 110109 ape information

```
GET /APEs?(APE.ApeID like 110109)
```

7.1.3 Safety requirements

The two parties involved in the information exchange is required to conduct identity authentication. The authentication process is required to follow the digital digest authentication specified in [IETF RFC 7616] and is required to follow [IETF RFC 9110]. During digest authentication, the server random number nonce is required to be reset after it exceeds a limited period of time, and the limited period of time is recommended as 3600 s.

7.2 Interface description

7.2.1 Public interface

– It is required to support registration, de-registration, keep-alive, and time calibration.

– It is required to re-register after a random time within 300 s, when the registration fails.

– It is required to support both registration parties configure the "heartbeat interval" parameter (the default heartbeat interval is 90 s).

– It is required to support the registrant periodically send heartbeat messages to the registered party to keep the heartbeat alive according to the "heartbeat interval", after the successful registration.

– It is required to support both registration parties configure the "Keep-alive timeout times" parameter uniformly (the default number of heartbeat timeouts is 3 times).

Comparative table of registration, de-registration, keep-alive, and time calibration with interface message is shown in Table 5.

Table 5 – Comparative table of registration, de-registration, keep alive, time calibration with interface message

No.	Interface name	Description	Interface message clause	Method
1	registration	Registration	8.2.1	POST
2	de-registration	De-registration	8.2.2	POST
3	keep_alive	Keep alive	8.2.3	POST
4	time_calibration	Time calibration	8.2.4	POST

7.2.2 Interfaces of reference point Pd/Pd'

– It is required to support registration, de-registration, keep_alive, and time_calibration functions from IPU to DSS. Interfaces as shown in Table 5.

– It is required to support upload automatically acquisition video and image information from IPU to DSS. Interfaces as shown in Table 6.

– It is required to support retransmission, when upload failed, after uploading the information.

NOTE 1 – Upload failed means that the response is not received within a certain time (e.g., 5 s).

– It is required to support cache information, de-registration, and re-registration, when link problem happened. It is required to support transmission cache information after re-registration successful.

NOTE 2 – Link problem means that unsuccessful retransmission multiple times (e.g., 3 times).

Table 6 – Comparative table of automatic video and image information uploading with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_video_clips_create_and_query	Upload video clips automatically in batch.	DSS	IPU	8.2.6.1	POST
2	batch_image_create_and_query	Upload image in batch automatically.	DSS	IPU	8.2.7.1	POST
3	batch_motor_vehicles_Create_query_update_delete	Upload motor vehicle in batch automatically.	DSS	IPU	8.2.8.1	POST
4	batch_non-motor_vehicles_create_query_update_delete	Upload non-motor vehicle in batch automatically.	DSS	IPU	8.2.9.1	POST

7.2.3 Interfaces of reference point Ma/Ma'

– It is required to support registration, de-registration, keep_alive, and time_calibration functions from IVM to IAS. Interfaces as shown in Table 5.

– It is required to support retrieving target by image and attribution between IAS and IVM. Interfaces as shown in Table 7.

Table 7 – Comparative table of target retrieving by image and attribution with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	synchronously_target_query_by_image	Create a task to target query by image (synchronization).	IVM	IAS	8.2.10.1	POST
2	asynchronously_create_and_delete_target_query_by_image	Create and delete a task to target query by image (asynchronous).	IVM	IAS	8.2.10.2	POST, DELETE
3	create_and_delete_target_retrieval_by_image	Upload a result of object to target query by image.	IAS	IVM	8.2.10.3	POST
4	batch_motor_vehicles_create_query_update_and_delete	Query motor vehicle by attribution.	IVM	IAS	8.2.8.1	GET
5	batch_non-motor_vehicles_create_query_update_and_delete	Query non-motor vehicle by attribution.	IVM	IAS	8.2.9.1	GET

– It is recommended to support analysis task controlling and analysis task status query between IAS and IVM. Interfaces as shown in Table 8.

Table 8 – Comparative table of analysis task control and status query with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	analysis_task_control	Pause, resume, and stop analysis task.	IVM	IAS	8.2.12	POST
2	analysis_task_status_query	Query analysis task status.	IVM	IAS	8.2.11	GET

7.2.4 Interfaces of reference point Ad/Ad'

– It is required to support registration, de-registration, keep_alive, and time_calibration functions from IAS to DSS. Interfaces as shown in Table 5.

– It is required to support queries based on feature attributes such as video and image information objects and their combinations, business operations and structural objects from IAS to DSS. Interfaces as shown in Table 9.

Table 9 – Comparative table of feature attributes and their combination query with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_video_clips_create_and_query	Query video clips in batch.	DSS	IAS	8.2.6.1	GET
2	a_single_video_clip_query_update_and_delete	Query a single video clip.	DSS	IAS	8.2.6.2	GET
3	a_single_video_clip query_update_and_delete_object_information	Query object information of video clips.	DSS	IAS	8.2.6.3	GET
4	batch_image_create_and_query	Query images in batch.	DSS	IAS	8.2.7.1	GET
5	a_single_image_query_update_and_delete	Query a single image.	DSS	IAS	8.2.7.2	GET
6	a_single_image_query_update_and_delete_object_information	Query image object information.	DSS	IAS	8.2.7.3	GET
7	batch_motor_vehicles_create_query_update_and_delete	Query motor vehicles in batch.	DSS	IAS	8.2.8.1	GET
8	batch_motor_vehicle_create_query_update_delete	Query a single motor vehicle.	DSS	IAS	8.2.8.2	GET
9	batch_non-motor_vehicles_create_query_update_and_delete	Query non-motor vehicles in batch.	DSS	IAS	8.2.9.1	GET
10	a_single_non-motor_vehicle_query_update_and_delete	Query a single non-motor vehicle.	DSS	IAS	8.2.9.2	GET

Table 9 – Comparative table of feature attributes and their combination query with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
11	batch_video_image_event_create_and_query	Query video and image events in batch.	DSS	IAS	8.2.14.1	GET
12	a_single_video_image_event_query_update_and_delete	Query a single video and image event.	DSS	IAS	8.2.14.2	GET
13	a_single_video_image_event_query_update_and_delete_object_information	Query object information of video and image event.	DSS	IAS	8.2.14.3	GET
14	batch_disposition_create_query_update_and_delete	Query disposition task.	DSS	IAS	8.2.15.1	GET
15	disposition_notification	Query disposition notification record.	DSS	IAS	8.2.16	GET
16	batch_subscription_create_query_update_and_delete	Query subscription task.	DSS	IAS	8.2.17.1	GET
17	subscription_notification	Query notification record.	DSS	IAS	8.2.18	GET
18	create_and delete_target_retrival_by_image	Query to retrieve the target result by image.	DSS	IAS	8.2.10.3	GET

– It is required to support the manual acquisition of video and image events and the creation, update and deletion of video and image information objects from IAS to DSS. Interfaces as shown in Table 10.

Table 10 – Comparative table of manual acquisition of video and image information object with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_video_image_event_create_and_query	Create video and image events in batch.	DSS	IAS	8.2.14.1	POST
2	a_single_video_image_event_query_update_and_delete	Update and delete a single video and image event.	DSS	IAS	8.2.14.2	PUT, DELETE
3	a_single_video_image_event_query_update_and_delete_object_information	Update and delete video and image event object information.	DSS	IAS	8.2.14.3	PUT, DELETE
4	batch_video_clips_create_and_query	Create manually acquisition video clips in batch.	DSS	IAS	8.2.6.1	POST

Table 10 – Comparative table of manual acquisition of video and image information object with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
5	a_single_video_clip_query_update_and_delete	Update and delete a single video clip acquisition manually.	DSS	IAS	8.2.6.2	PUT, DELETE
6	A_single_video_clip query_update_and_delete_object_information	Update and delete the object information of manually acquisition video clips.	DSS	IAS	8.2.6.3	PUT, DELETE
7	batch_image_create_and_query	Create manually acquisition images in batch.	DSS	IAS	8.2.7.1	POST
8	a_single_image_query_update_and_delete	Update and delete a single acquisition image manually.	DSS	IAS	8.2.7.2	PUT, DELETE
9	a_single_image_query_update_and_delete_object_information	Update and delete manually acquisition image object information.	DSS	IAS	8.2.7.3	PUT, DELETE
10	batch_motor_vehicles_create_query_update_and_delete	Create, update and delete motor vehicles in batch.	DSS	IAS	8.2.8.1	POST, PUT, DELETE
11	batch_motor_vehicle_create_query_update_delete	Update and delete a single motor vehicle.	DSS	IAS	8.2.8.2	PUT, DELETE
12	batch_non-motor_vehicles_create_query_update_and_delete	Create, update and delete non-motor vehicles in batch.	DSS	IAS	8.2.9.1	POST, PUT, DELETE
13	a_single_non-motor_vehicle_query_update_and_delete	Update and delete a single non-motor vehicle	DSS	IAS	8.2.9.2	PUT, DELETE

– It is required to support query and update IPU from IAS to DSS, and to query video tollgates, lanes and other information. Interfaces as shown in Table 11.

Table 11 – Comparative table of query and update the acquisition device/system with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	acquisition_device_query_and_update	Query the acquisition device.	DSS	IAS	8.2.5	GET
2	acquisition_device_query_and_update	Change the password of the acquisition device.	DSS	IAS	8.2.5	PUT
3	video_tollgate_query	Query the video tollgate.	DSS	IAS	8.2.13.1	GET
4	lane_query	Query the lanes.	DSS	IAS	8.2.13.2	GET

– It is required to support the disposition and notification of video and image information objects and the retrieval of targets by image between IAS and DSS. Interfaces as shown in Table 12.

Table 12 – Comparative table of disposition and notification with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_disposition_create_query_update_and_delete	Disposition in batch.	DSS	IAS	8.2.15.1	POST
2	batch_disposition_create_query_update_and_delete	Update and delete disposition tasks in batch.	DSS	IAS	8.2.15.1	PUT, DELETE
3	withdraw_disposition	Withdraw disposition.	DSS	IAS	8.2.15.2	PUT
4	disposition_notification	Notification.	IAS	DSS	8.2.16	POST
5	disposition_notification	Delete notification record.	IAS	DSS	8.2.16	DELETE
6	synchronously_target_query_by_image	Create a task to target query by image (synchronization).	DSS	IAS	8.2.10.1	POST
7	asynchronously_create_and_delete_target_query_by_image	Create and delete a task to target query by image (asynchronous).	DSS	IAS	8.2.10.2	POST, DELETE
8	create_and_delete_target_retrieval_by_image	Upload a result of object to target query by image.	IAS	DSS	8.2.10.3	POST

– It is required to support subscriptions and notifications of video and image information objects, IPU directories, etc. between IAS and DSS. Interfaces as shown in Table 13.

Table 13 – Comparative table of subscription and notification with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_subscription_create_query_update_and_delete	Subscription in batch.	DSS	IAS	8.2.17.1	POST
2	batch_subscription_create_query_update_and_delete	Update and delete subscription tasks in batch.	DSS	IAS	8.2.17.1	PUT, DELETE
3	unsubscription	Unsubscription.	DSS	IAS	8.2.17.2	POST
4	subscription_notification	Notification.	IAS	DSS	8.2.18	POST
5	subscription_notification	Delete notification record.	IAS	DSS	8.2.18	DELETE

7.2.5 Interfaces of reference point Id/Id'

– It is required to support registration, de-registration, keep alive, and time calibration functions from IVM to DSS. Interfaces as shown in Table 5.

– It is required to support upload automatically acquisition video and image information from IVM to DSS. Functions as shown in Table 14.

– It is required to support retransmission, when upload failed, after uploading the information.

NOTE 1 – Upload failed means that the response is not received within a certain time (e.g., 5 s).

– It is required to support cache information, de-registration, and re-registration, when link problem happened. It is required to support transmission cache information after re-registration successful.

NOTE 2 – Link problem means that unsuccessful retransmission multiple times (e.g., 3 times).

Table 14 – Comparative table of uploading the automatically acquisition video and image information with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_video_clips_create_and_query	Upload automatically acquisition video clips.	DSS	IVM	8.2.6.1	POST
2	batch_image_create_and_query	Upload automatically acquisition images.	DSS	IVM	8.2.7.1	POST
3	batch_motor_vehicles_create_query_update_delete	Upload automatically acquisition motor vehicles.	DSS	IVM	8.2.8.1	POST
4	batch_non-motor_vehicles_create_query_update_delete	Upload automatically acquisition non-motorized vehicles.	DSS	IVM	8.2.9.1	POST

– It is required to support queries based on feature attributes such as video and image information objects and their combinations from IVM to DSS. Interfaces as shown in Table 15.

Table 15 – Comparative table of feature attributes and their combination query with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_video_clips_create_and_query	Query video clips in batch.	DSS	IVM	8.2.6.1	GET
2	a_single_video_clip_query_update_and_delete	Query a single video clip.	DSS	IVM	8.2.6.2	GET
3	a_single_video_clip query_update_and_delete_object_information	Query object information of video clips.	DSS	IVM	8.2.6.3	GET
4	batch_image_create_and_query	Query images in batch.	DSS	IVM	8.2.7.1	GET
5	a_single_image_query_update_and_delete	Query a single image.	DSS	IVM	8.2.7.2	GET
6	a_single_image_query_update_and_delete_object_information	Query image object information.	DSS	IVM	8.2.7.3	GET
7	batch_motor_vehicles_create_query_update_and_delete	Query motor vehicles in batch.	DSS	IVM	8.2.8.1	GET
8	batch_motor_vehicle_create_query_update_delete	Query a single motor vehicle.	DSS	IVM	8.2.8.2	GET
9	batch_non-motor_vehicles_create_query_update_and_delete	Query non-motor vehicles in batch.	DSS	IVM	8.2.9.1	GET
10	a_single_non-motor_vehicle_query_update_and_delete	Query a single non-motor vehicle.	DSS	IVM	8.2.9.2	GET
11	batch_video_image_event_create_and_query	Query video and image events in batch.	DSS	IVM	8.2.14.1	GET
12	a_single_video_image_event_query_update_and_delete	Query a single video and image event.	DSS	IVM	8.2.14.2	GET
13	a_single_video_image_event_query_update_and_delete_object_information	Query object information of video and image event.	DSS	IVM	8.2.14.3	GET
14	batch_disposition_create_query_update_and_delete	Query disposition task.	DSS	IVM	8.2.15.1	GET
15	disposition_notification	Query notification record.	DSS	IVM	8.2.16	GET

Table 15 – Comparative table of feature attributes and their combination query with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
16	batch_subscription_create_query_update_and_delete	Query subscription task.	DSS	IVM	8.2.17.1	GET
17	subscription_notification	Query notification record.	DSS	IVM	8.2.18	GET
18	create_and_delete_target_retrieval_by_image	Query to retrieve the target result by image.	DSS	IVM	8.2.10.3	GET

– It is required to support the disposition and notification of video and image information objects and the retrieval of targets by image between IVM and DSS. Interfaces as shown in Table 16.

Table 16 – Comparative table of disposition and notification with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_disposition_create_query_update_and_delete	Disposition in batch.	IVM	DSS	8.2.15.1	POST
2	batch_disposition_create_query_update_and_delete	Update and delete disposition tasks in batch.	IVM	DSS	8.2.15.1	PUT, DELETE
3	withdraw_disposition	Withdraw disposition.	IVM	DSS	8.2.15.2	PUT
4	disposition_notification	Notification.	DSS	IVM	8.2.16	POST
5	disposition_notification	Delete notification record.	DSS	IVM	8.2.16	DELETE
6	synchronously_target_query_by_image	Create synchronization for image retrieval target task.	IVM	DSS	8.2.10.1	POST
7	synchronously_create_and_delete_target_query_by_image	Create and delete asynchronous target tasks for image retrieval.	IVM	DSS	8.2.10.2	POST, DELETE
8	create_and_delete_target_retrieval_by_image	Upload the target results for image retrieval.	DSS	IVM	8.2.10.3	POST

– It is required to support subscriptions and notifications of video and image information objects, acquisition device / system directories, etc. between IVM and DSS. Interfaces as shown in Table 17.

Table 17 – Comparative table of subscription and notification with interface message

No.	Interface name	Description	Interface provider	Interface caller	Interface message clause	Method
1	batch_subscription_create_query_update_and_delete	Subscription in batch.	DSS	IVM	8.2.17.1	POST
2	batch_subscription_create_query_update_and_delete	Update and delete subscription tasks in batch.	DSS	IVM	8.2.17.1	PUT, DELETE
3	unsubscription	Unsubscription.	DSS	IVM	8.2.17.2	POST
4	subscription_notification	Notification.	IVM	DSS	8.2.18	POST
5	subscription_notification	Delete notification record.	IVM	DSS	8.2.18	DELETE

8 Message flows and relevant protocols

8.1 Message flow

8.1.1 Create resource message interaction process

The interface caller creates the video and image information object for the interface provider. The interface caller sends an HTTP POST request to the interface provider according to the interface access form, and the interface provider returns a response of whether the creation is successful or not to the interface caller. The interactive process is shown in Figure 2.

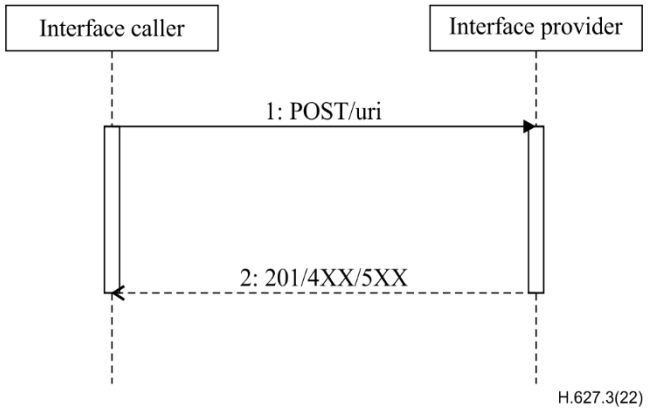

Figure 2 – Creation flow

The message process is described as follows:

1) The interface caller sends an HTTP POST request to the interface provider for /uri corresponding resource creation.

2) The interface provider returns a response to the interface caller regarding whether the creation is successful or not. Response code 201 is returned for success; Response code 4XX is returned for failure caused by the interface caller, and the message body carries the corresponding failure reason; Response code 5XX is returned for failure caused by the interface provider, and the message body carries the corresponding failure reason.

8.1.2 Query resource message interaction process

The interface caller can query the video and image information object, acquisition device and other information from the interface provider. The interface caller sends an HTTP GET request to the interface provider according to the interface access form, and the interface provider returns the reading result to the interface caller. The interactive process is shown in Figure 3.

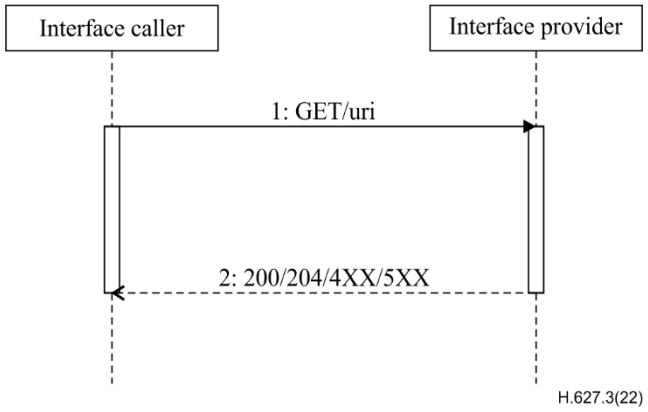

Figure 3 – Query flow

The message process is described as follows:

1) The interface caller sends an HTTP GET request to the interface provider for /uri corresponding resources. If pagination is required, specify it in the query condition.

2) The interface provider processes the combined query condition and returns the results that meet the conditions. Successfully processed and found a qualified result, response code 200 is returned, and the message body carries the result; the successful processing and no qualified result is found, response code 204 is returned, and the message body is empty; the failure caused by the interface caller returns response code 4XX, and the message body carries the corresponding failure reason; the failure caused by the interface provider returns response code 5XX, and the message body carries the corresponding failure reason.

8.1.3 Update resource message interaction process

The interface caller updates the video and image information object, disposition and announcement object, subscription and notification object, etc. to the interface provider; the interface caller sends an HTTP PUT request to the interface provider according to the interface access form, and the interface provider returns the update result to the interface caller. The interactive process is shown in Figure 4.

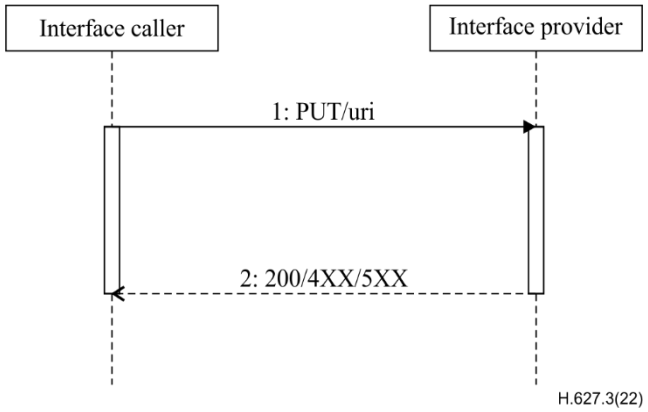

Figure 4 – Update flow

The message process is described as follows:

1) The interface caller sends an HTTP PUT update request to the interface provider for /uri corresponding resources.

2) The interface provider returns a response of whether the update is successful or not to the interface caller. Response code 200 is returned for success; Response code 4XX is returned for failures caused by the interface caller, and the message body carries the corresponding failure reason; Response code 5XX is returned for failures caused by the interface provider, and the message body carries the corresponding failure reason.

8.1.4 Delete resource message interaction process

The interface caller deletes the video and image information object, disposition and announcement object, the subscription and notification object, etc. to the interface provider. The interface caller sends an HTTP DELETE request to the interface provider according to the interface access form, and the interface provider returns the delete result to the interface caller. The interactive process is shown in Figure 5.

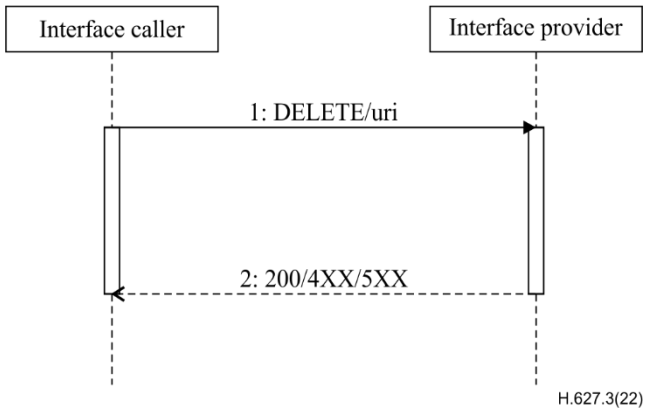

Figure 5 – Delete flow

The message process is described as follows:

1) The interface caller sends an HTTP DELETE to delete request to the interface provider for /uri corresponding resources, and the resource identifier to be deleted is specified in the query string.

2) The interface provider returns a response of whether the deletion is successful or not to the interface caller. Response code 200 is returned for success; response code 4XX is returned for failures caused by the interface caller, and the message body carries the corresponding failure reason; response code 5XX is returned for failures caused by the interface provider, and the message body carries the corresponding failure reason.

8.1.5 Asynchronous operation message interaction process

The interface caller performs asynchronous operations such as disposition and announcement, subscription and notification, and image retrieval of the target to the interface provider. The interface caller sends an HTTP POST request to the interface provider according to the interface access form, and the interface provider asynchronously returns the result to the interface caller after a period of processing. The interactive process is shown in Figure 6.

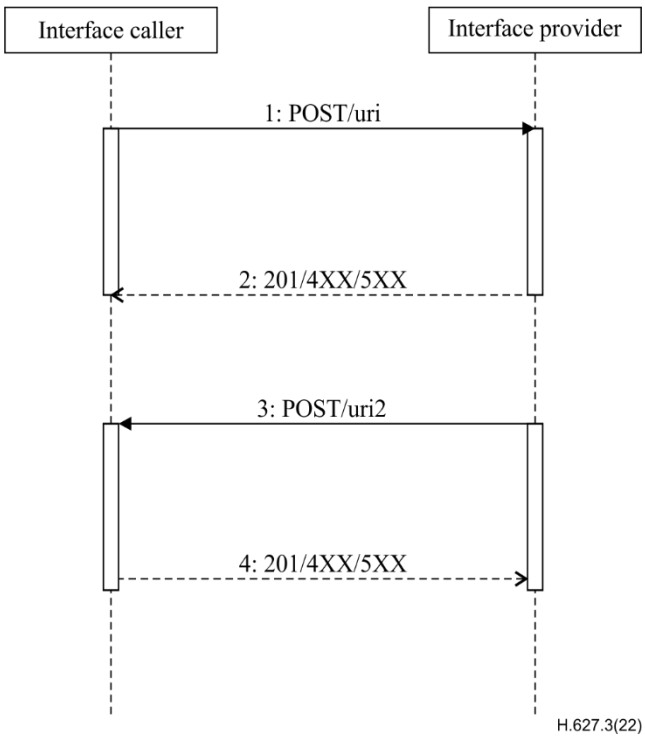

Figure 6 – Asynchronous operation flow

The message process is described as follows:

1) The interface caller sends an HTTP POST request to the interface provider to perform asynchronous operations for /uri corresponding resources.

2) The interface provider returns a response to the interface caller regarding whether the reception is successful or not. Response code 201 is returned for success; response code 4XX is returned for failure caused by the interface caller, and the message body carries the corresponding failure reason; Response code 5XX is returned for failure caused by the interface provider, and the message body carries the corresponding failure reason.

3) The interface provider processes the asynchronous request and sends the processing result to /uri2 specified by the interface caller via HTTP POST.

4) The interface caller returns a response to the interface provider regarding whether the reception is successful or not. Response code 201 is returned for success; response code 4XX is returned for failure caused by the interface caller, and the message body carries the corresponding failure reason; response code 5XX is returned for failure caused by the interface provider, and the message body carries the corresponding failure reason.

8.2 Interface definition

The interface definition description is shown in Table 18. Among them, the URI item describes the interface resource request URI; the function item comprehensively describes the functional behaviour; the method item lists the HTTP methods involved in the resource, including POST, GET, PUT, and DELETE; the query string item describes the key-value pair of the resource; the message body item describes the message body of the HTTP request; the returned result item describes the response message body of the HTTP request; use "<object tag>" to point the object that constitutes the message body and the returned result; the comment item gives a special description of the interface message. The interface message body and response results are described by the JSON Schema shown in Annex A.

Table 18 – Interface definition description

URI			
Function			
Method	**Query string**	**Message body**	**Return result**
POST			
GET			
PUT			
DELETE			
Annotation			

8.2.1 Interface of registration

Table 19 – Interface of registration

URI	/Register		
Function	registration		
Method	**Query string**	**Message body**	**Return result**
POST	none	<Register>	<ResponseStatus>
Annotation	1) Json Schema description of Register is shown in clause A.2.1 /schema/componets#/$defs/register 2) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus, and the Id of ResponseStatus is the DeviceID requested to register 3) The registration process is required to comply with the process specified in [IETF RFC 9110], in which Authentication Schema uses Digest (shown in [IETF RFC 7616] for details) 4) POST process is shown in clause 8.1.1		

8.2.2 Interface of de-registration

Table 20 – Interface of de-registration

URI	/UnRegister		
Function	de-registration		
Method	**Query string**	**Message body**	**Return result**
POST	none	<UnRegister>	<ResponseStatus>
Annotation	1) Json Schema description of UnRegister is shown in clause A.2.1 /schema/componets#/$defs/unRegister 2) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 3) POST process is shown in clause 8.1.1		

8.2.3 Interface of keep_alive

Table 21 – Interface of keep_alive

URI	/Keepalive		
Function	keep alive		
Method	**Query string**	**Message body**	**Return result**
POST	none	<Keepalive>	<ResponseStatus>
Annotation	1) Json Schema description of Keep alive is shown in clause A.2.1 /schema/componets#/$defs/keepalive 2) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 3) Message flow is shown in clause 8.1.1		

8.2.4 Interface of time_calibration

Table 22 – Interface of time_calibration

URI	/Time		
Function	time calibration		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<SystemTime>
Annotation	1) Json Schema description of SystemTime is shown in clause A.2.1 /schema/componets#/$defs/systemTime 2) Message flow is shown in clause 8.1.2		

8.2.5 Interface of acquisition_device_query_and_update

Table 23 – Interface of acquisition_device_query_and_update

URI	/APEs		
Function	IPU query and update		
Method	**Query string**	**Message body**	**Return result**
GET	APE attribute key-value pairs	none	<APEList> or<ResponseStatus>
PUT	none	<APEList>	<ResponseStatusList>
Annotation	1) Definition of APE attributes is shown in clause A.2.2/schema/ape-list#/$defs/ape/properties 2) Json Schema description of APEList is shown in clause A.2.2/schema/ape-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method is shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) The PUT method is used to modify the password of the IPU to realize dynamic password management 7) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 8) GET process is shown in clause 8.1.2 and PUT process is shown in clause 8.1.3		

8.2.6 Interface of video clip

8.2.6.1 Interface of batch_video_clips_create_and_query

Table 24 – Interface of batch_video_clips_create_and_query

URI	/VideoSlices		
Function	create and query the combined information of video clip objects and related semantic attribute objects.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<VideoSliceList>	<ResponseStatusList>
GET	VideoSliceInfo attribute key-value pairs	none	<VideoSliceList>or<ResponseStatus>
Annotation	1) Definition of VideoSliceInfo attributes is shown in clause A.2.5/schema/video-slice-info-list#/$defs/videoSliceInfo/properties 2) Json Schema description of VideoSliceList is shown in clause A.2.19/schema/video-slice-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) POST process is shown in clause 8.1.1 and GET process is shown in clause 8.1.2		

8.2.6.2 Interface of a_single_video_clip_query_update_and_delete

Table 25 – Interface of a_single_video_clip_query_update_and_delete

URI	/VideoSlices/{VideoID}		
Function	query, update and delete the combination information of a single video clip object and related semantic attribute objects.		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<VideoSlice>or<ResponseStatus>
PUT	none	<VideoSlice>	<ResponseStatus>
DELETE	none	none	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of VideoSlice is shown in clause A.2.19/schema/video-slice-list#/$defs/videoSlice 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) GET process is shown in clause 8.1.2 and DELETE process is shown in clause 8.1.4		

8.2.6.3 Interface of a_single_video_clip_query_update_and_delete_object_information

Table 26 – Interface of a_single_video_clip_query_update_and_delete_object_information

URI	/VideoSlices/{VideoID}/Info		
Function	query, update and delete object information of a single video clip.		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<VideoSliceInfo>or<ResponseStatus>
PUT	none	<VideoSliceInfo>	<ResponseStatus>
DELETE	none	none	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of VideoSliceInfo is shown in clause A.2.5/schema/video-slice-info-list#$defs/videoSliceInfo/properties 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) GET process is shown in clause 8.1.2, PUT process is shown in 8.1.3, DELETE process is shown in clause 8.1.4		

8.2.7 Interface of image

8.2.7.1 Interface of batch_image_create_and_query

Table 27 – Interface of batch_image_create_and_query

URI	/Images		
Function	create and query the combined information of image objects and related semantic attribute objects.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<ImageList>	<ResponseStatusList>
GET	ImageInfo attribute key-value pairs	none	<ImageList>or<ResponseStatus>
Annotation	1) Definition of ImageInfo attributes is shown in clause A.2.6/schema/image-info-list#/$defs/imageInfo/properties 2) Json Schema description of ImageList is shown in clause A.2.18/schema/image-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) POST process is shown in clause 8.1.1 and GET process is shown in clause 8.1.2		

8.2.7.2 Interface of a_single_image_query_update_and_delete

Table 28 – Interface of a_single_image_query_update_and_delete

URI	/Images/{ImageID}		
Function	query, update and delete the combination information of a single image object and related semantic attribute objects.		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<Image>or<ResponseStatus>
PUT	none	<Image>	<ResponseStatus>
DELETE	none	none	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of Image is shown in clause A.2.18/schema/image-list#/$defs/image 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, DELETE process is shown in clause 8.1.4		

8.2.7.3 Interface of a_single_image_query_update_and_delete_object_information

Table 29 – Interface of a_single_image_query_update_and_delete_object_information

URI	/Images/{ImageID}/Info		
Function	query, update and delete object information of a single image.		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<ImageInfo>or<ResponseStatus>
PUT	none	<ImageInfo>	<ResponseStatus>
DELETE	none	none	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of ImageInfo is shown in clause A.2.6/schema/image-info-list#/$defs/imageInfo/properties 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, DELETE process is shown in clause 8.1.4		

8.2.8 Interface of motor vehicle

8.2.8.1 Interface of batch_motor_vehicles_create_query_update_delete

Table 30 – Interface of batch_motor_vehicles_create_query_update_and_delete

URI	/MotorVehicles		
Function	create, query, update, and delete motor vehicle object information, and support batch operations.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<MotorVehicleList>	<ResponseStatusList>
GET	MotorVehicle attribute key-value pairs	none	<MotorVehicleList>or<ResponseStatus>
PUT	none	<MotorVehicleList>	<ResponseStatusList>
DELETE	The key is IDList, and the value is the motor vehicle object identifier separated by ";"	none	<ResponseStatusList>
Annotation	1) Definition of MotorVehicle attributes is shown in clause A.2.7/schema/motor-vehicle-list#/$defs/motorVehicle/properties 2) Json Schema description of MotorVehicleList is shown in clause A.2.7/schema/motor-vehicle-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) POST process is shown in clause 8.1.1, GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, and DELETE process in clause 8.1.4		

8.2.8.2 Interface of a_single_motor_vehicle_query_update_and_delete

Table 31 – Interface of a_single_motor_vehicle_query_update_and_delete

URI	/MotorVehicles/{MotorVehicleID}		
Function	query, update and delete object information of a single motor vehicle.		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<MotorVehicle>or<ResponseStatus>
PUT	none	<MotorVehicle>	<ResponseStatus>
DELETE	none	none	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of MotorVehicle is shown in clause A.2.7/schema/motor-vehicle-list#/$defs/motorVehicle 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, DELETE process is shown in clause 8.1.4		

8.2.9 Interface of non-motor vehicle

8.2.9.1 Interface of batch_non-motor_vehicles_create_query_update_delete

Table 32 – Interface of batch_non-motor_vehicles_create_query_update_and_delete

URI	/NonMotorVehicles		
Function	create, query, update, and delete non-motor vehicle object information, and support batch operations.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<NonMotorVehicle List>	<ResponseStatusList>
GET	NonMotorVehicle attribute key-value pairs	none	<NonMotorVehicleList>or<ResponseStatus>
PUT	none	<NonMotorVehicle List>	<ResponseStatusList>
DELETE	the key is IDList, and the value is the non-motor vehicle object identifier separated by ";"	none	<ResponseStatusList>
Annotation	1) Definition of NonMotorVehicle attributes is shown in clause A.2.8 /schema/nonmotor-vehicle-list#/$defs/nonMotorVehicle/properties 2) Json Schema description of NonMotorVehicleList is shown in clause A.2.8 /schema/nonmotor-vehicle-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) POST process is shown in clause 8.1.1, GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, and DELETE process in clause 8.1.4		

8.2.9.2 Interface of a_single_non-motor_vehicle_query_update_and_delete

Table 33 – Interface of a_single_non-motor_vehicle_query_update_and_delete

URI	/NonMotorVehicles/{NonMotorVehicleID}		
Function	query, update and delete object information of a single non-motor vehicle.		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<NonMotorVehicle>or<ResponseStatus>
PUT	none	<NonMotorVehicle>	<ResponseStatus>
DELETE	none	none	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of Non-Motor Vehicle is shown in clause A.2.8 /schema/nonmotor-vehicle-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus		

Table 33 – Interface of a_single_non-motor_vehicle_query_update_and_delete

	4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, DELETE process is shown in clause 8.1.4

8.2.10 Interface of target query by image

8.2.10.1 Interface of synchronously_target_query_by_image

Table 34 – Interface of synchronously_target_query_by_image

URI	/ImageSearchedByImagesSync		
Function	query the target object by image synchronously. Realize the image query target for the video and image information objects such as a motor vehicle, etc., and synchronously return the image query target result.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<ImageSearchedByImage>	<ImageResultSBI>or<ResponseStatus>
Annotation	1) Definition of ImageSearchedByImage attributes is shown in clause A.2.15 /schema/image-searched-by-image 2) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 3) Json Schema description of ImageResultSBI is shown in clause A.2.16 /schema/image-result-SBI-list 4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) POST process is shown in clause 8.1.1		

8.2.10.2 Interface of asynchronously_create_and_delete_target_query_by_image

Table 35 – Interface of asynchronously_create_and_delete_target_query_by_image

URI	/ImageSearchedByImagesAsync		
Function	create and delete asynchronously query the target object by image. Realize the image retrieval target for the video and image information objects such as motor vehicle, etc., and asynchronously return the image retrieval target result.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<ImageSearchedByImage>	<ResponseStatus>
DELETE	The key is IDList, and the value is the search ID separated by ";"	none	<ResponseStatusList>
Annotation	1) Definition of ImageSearchedByImage attributes is shown in clause A.2.15 /schema/image-searched-by-image 2) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 3) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList		

Table 35 – Interface of asynchronously_create_and_delete_target_query_by_image

	4) When the operation fails, a ResponseStatus is returned, explaining the reason for the failure. 5) DELETE process is shown in clause 8.1.4 6) The addition of the image retrieval target and the addition of the image retrieval target result are a pair of asynchronous operation interfaces. After the image retrieval target is issued through the POST method, when the target that meets the conditions appears, the retrieval result is synchronously uploaded through the POST method. The process is shown in clause 8.1.5.

8.2.10.3 Interface of create_and_delete_target_retrieval_by_image

Table 36 – Interface of create_and_delete_target_retrieval_by_image

URI	/ImageResultSBIs		
Function	create and query to retrieve the target results object by image, and support batch operation.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<ImageResultSBIList>	<ResponseStatusList>
GET	ImageResultSBI attribute key-value pairs	none	<ImageResultSBIList>or<ResponseStatus>
Annotation	1) Definition of ImageResultSBI attributes is shown in clause A.2.16/schema/image-result-SBI-list#/$defs/imageResultSBIList/properties 2) Json Schema description of ImageResultSBIList is shown in clause A.2.16 /schema/image-result-SBI-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) When the operation fails, a ResponseStatus is returned, explaining the reason for the failure. 6) GET process is shown in clause 8.1.2 7) The addition of the image retrieval target and the addition of the image retrieval target result are a pair of asynchronous operation interfaces. After the image retrieval target is issued through the POST method, when the target that meets the conditions appears, the retrieval result is synchronously uploaded through the POST method. The process is shown in clause 8.1.5.		

8.2.11 Interface of analysis_task_status_query

Table 37 – Interface of analysis_task_status_query

URI	/TaskStatus		
Function	query the status of the video image analysis task.		
Method	**Query string**	**Message body**	**Return result**
GET	TaskStatus attribute key-value pairs	none	<TaskStatusList>or<ResponseStatus>
Annotation	1) Definition of TaskStatus attributes is shown in clause A.2.20/schema/task-status-list#/$defs/taskStatus/properties		

Table 37 – Interface of analysis_task_status_query

	2) Json Schema description of TaskStatusList is shown in clause A.2.20 /schema/task-status-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 5) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 6) GET process is shown in clause 8.1.2

8.2.12 Interface of analysis_task_control

Table 38 – Interface of analysis_task_control

URI	/AnalyzeTaskControl		
Function	create analysis task control objects to realize the start and stop control of each analysis task of the video image.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<AnalyzeTaskControl>	<ResponseStatus>
Annotation	1) Json Schema description of AnalyzeTaskControl is shown in clause A.2.21 /schema/task-control 2) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 3) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 4) POST process is shown in clause 8.1.1		

8.2.13 Interface of video tollgate and lane

8.2.13.1 Interface of video_tollgate_query

Table 39 – Interface of video_tollgate_query

URI	/Tollgates		
Function	query video tollgate		
Method	**Query string**	**Message body**	**Return result**
GET	Tollgate attribute key-value pairs	none	<TollgateList>or<ResponseStatus>
Annotation	1) Definition of Tollgate attributes is shown in clause A.2.3/schema/tollgate-list#/$defs/tollgate/properties 2) Json Schema description of TollgateList is shown in clause A.2.3/schema/tollgate-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 5) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 6) GET process is shown in clause 8.1.2		

8.2.13.2 Interface of lane_query

Table 40 – Interface of lane_query

URI	/Tollgates		
Function	query lane		
Method	**Query string**	**Message body**	**Return result**
GET	Lane attribute key-value pairs	none	<LaneList>or<ResponseStatus>
Annotation	1) Definition of Lane attributes is shown in clause A.2.4/schema/lane-list#/$defs/lane/properties 2) Json Schema description of LaneList is shown in clause A.2.4/schema/lane-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 5) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 6) GET process is shown in clause 8.1.2		

8.2.14 Interface of video image event

8.2.14.1 Interface of batch_video_image_event_create_and_query

Table 41 – Interface of batch_video_image_event_create_and_query

URI	/Cases		
Function	create and query the combined information of video image event, support batch operation.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<CaseList>	<ResponseStatusList>
GET	CaseInfo attribute key-value pairs	none	<CaseList>or<ResponseStatus>
Annotation	1) Definition of CaseInfo attributes is shown in clause A.2.9/schema/case-info-list#/$defs/caseInfo/properties 2) Json Schema description of CaseList is shown in clause A.2.17/schema/case-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) POST process is shown in clause 8.1.1 and GET process is shown in clause 8.1.2		

8.2.14.2 Interface of a_single_video_image_event_query_update_and_delete

Table 42 – Interface of a_single_video_image_event_query_update_and_delete

URI	/Cases/{CaseID}		
Function	query, update and delete combined information of a single video image event.		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<Case>or<ResponseStatus>
PUT	none	<Case>	<ResponseStatus>
DELETE	none	none	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of Case is shown in clause A.2.17/schema/case-list#/$defs/case 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, DELETE process is shown in clause 8.1.4		

8.2.14.3 Interface of a_single_video_image_event_query_update_and_delete_object_information

Table 43 – Interface of a_single_video_image_event_query_update_and_delete_object_information

URI	/Cases/{CaseID}/Info		
Function	query, update and delete description information of a single video image event.		
Method	**Query string**	**Message body**	**Return result**
GET	none	none	<CaseInfo>or<ResponseStatus>
PUT	none	<CaseInfo>	<ResponseStatus>
DELETE	none	none	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of CaseInfo is shown in clause A.2.9/schema/case-info-list#/$defs/caseInfo/properties 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, DELETE process is shown in clause 8.1.4		

8.2.15 Interface of disposition

8.2.15.1 Interface of batch_disposition_create_query_update_and_delete

Table 44 – Interface of batch_disposition_create_query_update_and_delete

URI	/Dispositions		
Function	create, query, update and delete disposition object information, support batch operations.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<DispositionList>	<ResponseStatusList>
GET	Disposition attribute key-value pairs	none	<DispositionList>or<ResponseStatus>
PUT	none	<DispositionList>	<ResponseStatusList>
DELETE	the key is IDList, and the value is the non-motor vehicle object identifier separated by ";"	none	<ResponseStatusList>
Annotation	1) Definition of Disposition attributes is shown in clause A.2.10/schema/disposition-list#/$defs/disposition/properties 2) Json Schema description of DispositionList is shown in clause A.2.10 /schema/disposition-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, and DELETE process in clause 8.1.4 8) Disposition and notification is a pair of asynchronous operation interface, after the disposition is released through the POST method, when the qualified target appears, the notification is uploaded asynchronously through the POST method, and its process is shown in clause 8.1.5		

8.2.15.2 Interface of withdraw_disposition

Table 45 – Interface of withdraw_disposition

URI	/Dispositions/{DispositionID}		
Function	update the disposition task to achieve withdraw		
Method	**Query string**	**Message body**	**Return result**
PUT	none	<Disposition>	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of Disposition is shown in clause A.2.10/schema/disposition-list#/$defs/disposition 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus		

Table 45 – Interface of withdraw_disposition

	4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) Update the Disposition status to "withdraw" through PUT method to achieve the withdrawal function, and its process is shown in clause 8.1.3.

8.2.16 Interface of disposition_notification

Table 46 – Interface of disposition_notification

URI	/DispositionNotifications		
Function	create, query and delete disposition notification object, support batch operations.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<DispositionNotificationList>	<ResponseStatusList>
GET	DispositionNotificationList attribute key-value pairs	none	<DispositionNotificationList>or<ResponseStatus>
DELETE	the key is IDList, and the value is the non-motor vehicle object identifier separated by ";"	none	<ResponseStatusList>
Annotation	1) Definition of DispositionNotificationList attributes is shown in clause A.2.11 /schema/disposition-notification-list#/$defs/dispositionNotification/properties 2) Json Schema description of DispositionNotificationList is shown in clause A.2.11 /schema/disposition-notification-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) GET process is shown in clause 8.1.2, and DELETE process in clause 8.1.4 8) Disposition and notification is a pair of asynchronous operation interface, after the disposition is released through the POST method, when the qualified target appears, the notification is uploaded asynchronously through the POST method, and its process is shown in clause 8.1.5		

8.2.17 Interface of subscription

8.2.17.1 Interface of batch_subscription_create_query_update_and_delete

Table 47 – Interface of batch_subscription_create_query_update_and_delete

URI	/Subscribes		
Function	implement subscriptions to video image information objects, etc, support batch operations.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<SubscribeList>	<ResponseStatusList>
GET	Subscribe attribute key-value pairs	none	<SubscribeList>or<ResponseStatus>
PUT	none	<SubscribeList>	<ResponseStatusList>
DELETE	the key is IDList, and the value is the non-motor vehicle object identifier separated by ";"	none	<ResponseStatusList>
Annotation	1) Definition of Subscribe attributes is shown in clause A.2.12/schema/subscribe-list#/$defs/subscribe/properties 2) Json Schema description of SubscribeList is shown in clause A.2.12 /schema/subscribe-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) GET process is shown in clause 8.1.2, PUT process is shown in clause 8.1.3, and DELETE process in clause 8.1.4 8) Subscription and notification is a pair of asynchronous operation interface, after the subscription is released through the POST method, when the qualified target appears, the notification is uploaded asynchronously through the POST method, and its process is shown in clause 8.1.5		

8.2.17.2 Interface of unsubscription

Table 48 – Interface of unsubscription

URI	/Subscribes/{SubscribeID}		
Function	update the subscription task to achieve withdrawl		
Method	**Query string**	**Message body**	**Return result**
PUT	none	<Subscribe>	<ResponseStatus>
Annotation	1) The {} uses in URI is shown in clause 7.1.1.1 2) Json Schema description of Subscribe is shown in clause A.2.12/schema/subscribe-list#/$defs/subscribe 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus		

Table 48 – Interface of unsubscription

	4) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 5) Update the Subscribe status to "unsubscription" through PUT method to achieve the unsubscription function, and its process is shown in clause 8.1.3.

8.2.18 Interface of subscription notification

Table 49 – Interface of subscription notification

URI	/SubscribeNotifications		
Function	create, query and delete subscription notification object, support batch operations.		
Method	**Query string**	**Message body**	**Return result**
POST	none	<SubscribeNotificationList>	<ResponseStatusList>
GET	SubscribeNotification List attribute key-value pairs	none	<SubscribeNotificationList>or<ResponseStatus>
DELETE	the key is IDList, and the value is the non-motor vehicle object identifier separated by ";"	none	<ResponseStatusList>
Annotation	1) Definition of SubscribeNotification attributes is shown in clause A.2.13 /schema/subscribe-notification-list#/$defs/subscribeNotification/properties 2) Json Schema description of SubscribeNotificationList is shown in clause A.2.13 /schema/subscribe-notification-list 3) Json Schema description of ResponseStatus is shown in clause A.2.1 /schema/componets#/$defs/responseStatus 4) Json Schema description of ResponseStatusList is shown in clause A.2.1 /schema/componets#/$defs/responseStatusList 5) Query string requirements of the GET method are shown in clause 7.1.2. When the query string is empty, it means that there is no restriction 6) When the operation fails, a ResponseStatus is returned to explain the reason for the failure 7) GET process is shown in clause 8.1.2, and DELETE process in clause 8.1.4 8) Subscription and notification is a pair of asynchronous operation interface, after the subscription is released through the POST method, when the qualified target appears, the notification is uploaded asynchronously through the POST method, and its process is shown in clause 8.1.5		

Annex A

Data type and object JSON-Schema description

(This annex forms an integral part of this Recommendation.)

This annex defines the data type and JSON-Schema description[1] of the interface object in clause 8. Clause A.1 defines the data type of object attribute. Clause A.2 defines JSON-Schema description comparative with tables in clause 8.2.

A.1 Data type

This Recommendation uses JSON-Schema to describe the data objects passed between each unit. For more information about JSON-Schema, please refer to JSON Schema Core [b-JSON-1] and JSON Schema Validation [b-JSON-2] and Understanding JSON [b-JSON-3].

Data types are based on the types supported by the draft-bhutton-schema-00 (https://datatracker.ietf.org/doc/html/draft-bhutton-json-schema-00#section-4.2.1).null is not supported as a type.

In order to describe the data object more conveniently, this Recommendation expands the attribute of the format keyword. The expanded attribute is defined in Table A.1.

Table A.1 – Format attribute extension

Format attribute	Type	Comments
integer	number	signed 32 bit
long	number	signed 64 bit
double	number	
dateTime	string	as defined in section 5.6 of [IETF RFC 3339] dateTime = date-fullyear date-month date-mday time-hour time-minute time-second
dateTimeMS	string	as defined in section 5.6 of [IETF RFC 3339] time-millsecord = 3 digits, range from 000 to 999 dateTime = date-fullyear date-month date-mday time-hour time-minute time-second time-millsecord
deviceIDType	string	deviceIDType in accordance with [ITU-T H.626] clause 7.1.4

[1] The Software Copyright Holder represents and warrants that to the best of its knowledge, it has the necessary copyright rights to waive all of the copyright rights as permissible under national law in the Software such that the Software can be used by implementers without further licensing concerns.

No patent licence is granted, nor is a patent licensing commitment made, by implication, estoppel or otherwise.

Disclaimer: Other than as expressly provided herein, (1) the Software is provided "AS IS" WITH NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGMENT OF INTELLECTUAL PROPERTY RIGHTS and (2) neither the Software Copyright Holder (or its affiliates) nor the ITU shall be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the Software."

Table A.1 – Format attribute extension

Format attribute	Type	Comments
basicObjectIdType	string	The basic information object type of the video image is composed of 45 digits, and the formula is as follows: basicObjectIdType = deviceIDType subType dateTime serialNumber The deviceIDType and dateTime in the formula are defined in front of the table. subType= 01-videoslice 02-image 03-file 04-reserved 05-dataClassTab:motorVehicle 06-dataClassTab:nonMotorVehicle 99-other serialNumber= 5-digit serial number: It generally follows numerical order. Display sequentially from 00001
imageCntObjectIdType	string	The semantic attribute object type of video image information is composed of 52 digits, and the formula is as follows: imageCntObjectIdType = basicObjectIdType subType serialNumber The basicObjectIdType in the formula is defined in front of the table. subType= 01-motorVehicle 02-nonMotorVehicle 99-other serialNumber= 5-digit serial number: It generally follows numerical order. Sequential representation starting from 00001.
caseObjectIdType	string	The time object identification coding type of video image information is composed of 34 digits caseObjectIdType=countryCode zoneCode reservedCode dateTime serialNumber The dateTime in the formula is defined in front of the table. countryCode=4-digit country code zoneCode=8-digit zone code, it is recommended to use postal code or customized zone code. reservedCode=4-digit code, it is reserved for industry cloassification or other usage. serialNumber=4 digit serial number
businessObjectIdType	string	The service operation identification type of video image information is composed of 37 digits businessObjectIdType= countryCode zoneCode reservedCode subType dateType serialNumber The dateTime in the formula is defined in front of the table.

Table A.1 – Format attribute extension

Format attribute	Type	Comments
		countryCode=4-digit country code zoneCode=8-digit zone code, it is recommended to use postal code or customized zone code. reservedCode=4-digit code,it is reserved for industry cloassification or other usage. subType=01-disposition 02-disposition notification 03-subscribe 04-subscribe notification 05-analysis task 99-other serialNumber=4 digit serial number

A.2 JSON-Schema description

A.2.1 /schema/components

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/components",
     "description":" Common item definition ",
     "$defs":{
          "deviceID":{
               "description":"Device ID, identifying the uniqueness of the
device",
               "type":"string",
               "format":"deviceIDType"
          },
          "basicObjectId":{
               "description":"Video image basic information object ID and data
organization library, used to identify the uniqueness of videoSliceInfo,
imageInfo, motorVehicle library, nonMotorVehicle library, etc.",
               "type":"string",
               "format":"basicObjectIdType"
          },
          "imageCntObjectId":{
               "description":"Video image information semantic attribute object
ID, used to identify the uniqueness of motorVehicle, nonMotorVehicle, etc.",
               "type":"string",
               "format":"imageCntObjectIdType"
          },
          "caseObjectId":{
               "description":"Case object ID",
               "type":"string",
               "format":"caseObjectIdType"
          },
          "businessObjectId":{
               "description":"Unified identification of business operation tasks,
disposition/cancel disposition, announcements, subscriptions, notifications,
analysis tasks, etc.",
               "type":"string",
               "format":"businessObjectIdType"
          },
          "hDirection":{
               "description":"Horizontal direction. 1: west to east; 2: east to
west; 3: north to south; 4: south to north; 5: southwest to northeast; 6:
```

```
northeast to southwest; 7: northwest to southeast; 8: southeast to northwest; 9:
Other directions",
                "enum":[1,2,3,4,5,6,7,8,9]
            },
            "vDirection":{
                "description":"Vertical direction. 1: From top to bottom; 2: From
bottom to top; 3: Horizontal; 9: Other",
                "enum":[1,2,3,9]
            },
            "statusType":{
                "description":"Device status, 1: online; 2: offline; 9: other",
                "enum":["1","2","9"]
            },
            "storagePath":{
                "description":"Storage path",
                "type":"string",
                "format":"uri",
                "maxLength":256
            },
            "urlPath":{
                "description":"url address",
                "type":"string",
                "format":"uri",
                "maxLength":256
            },
            "flag":{
                "description":"Judgmental flag. 0:false;1:true",
                "enum":[0,1]
            },
            "wearedFlag":{
                "description":"Weared flag. 0:false;1:true;2:not sure",
                "enum":[0,1,2]
            },
            "securityLevel":{
                "description":"Sensitivity level. 1: Sensitive; 2: Normal; 9:
Other",
                "enum":[1,2,9]
            },
            "binary":{
                "description":"File Binary Content ",
                "type":"string",
                "contentMediaType":"application/octet-stream",
                "contentEncoding":"binary"
            },
            "base64":{
                "description":"File base64 encoding, the base64 encoding method
adopts the method specified by rfc2046",
                "type":"string",
                "contentEncoding":"base64"
            },
            "infoKind":{
                "description":"Information acquisition classification, 0: other;
1: automatic acquired by acquisition device or analysis system; 2: manual
acquisition and input through the application platform",
                "enum":[0,1,2]
            },
            "imageCoordinates":{
                "description":"The position of the smaller image in the larger
image, starting from the upper left pixel of the larger image, moving
horizontally to the right is the X-axis forward direction, and moving
horizontally downward is the Y-axis forward direction, in pixels.",
                "type":"integer"
            },
            "color":{
```

```
                "description":"The video focuses on the target color. A: white; B:
gray; C: yellow; D: pink; E: red; F: purple; G: green; H: blue; I: brown; J:
black; K: orange; L: green; Y: colorless; Z: other",
                "enum":["A","B","C","D","E","F","G","H","I","J","K","L","Y","Z"]
           },
           "dataClassTab":{
                "description":"Structure definition of data organization library",
                "type":"object",
                "properties" : {
                     "TabID":{
                          "description":"Library ID",
                          "$ref":"#/$defs/basicObjectId"
                     },
                     "TabName":{
                          "description":"Library name",
                          "type":"string"
                     },
                     "Description":{
                          "description":"Library Description",
                          "type":"string"
                     },
                     "IsAffirmed":{
                          "description":"Known identity. 0: Known identity; 1:
Unknown identity",
                          "enum":[0,1]
                     },
                },
                "required":["TabID","TabName"]
           },
           "dataClassTabList":{
                "description":"Structure definition of data organization library
list",
                "type":"array",
                "items":{
                     "$ref":"#/$defs/dataClassTab"
                }
           },
           "responseStatus":{
                "description":"Structure definition of response object",
                "type":"object",
                "properties" : {
                     "RequestURL":{
                          "description":"URL for corresponding operation",
                          "$ref":"#/$defs/urlPath"
                     },
                     "StatusCode":{
                          "description":"status code. 0:OK, normal. 1. Other
errors. 4: invalid operation; 7: The JSON format is invalid. 8: Invalid JSON
content",
                          "enum":[0,1,4,7,8]
                     },
                     "StatusString":{
                          "description":"Response description information",
                          "type":"string"
                     },
                     "Id":{
                          "description":"The POST method will return a new ID if
it succeeds in creating a resource, but it will not return if it fails.",
                          "type":"string"
                     },
                     "LocalTime":{
                          "description":"Current time, used for occasions when
time adjustment is needed",
                          "type":"string",
```

```
                    "format":"dateTime"
                }
            },
            "required":["RequestURL","StatusCode","StatusString"]
        },
        "responseStatusList":{
            "description":"Structure definition of response object list",
            "type":"array",
            "items":{
                "$ref":"#/$defs/responseStatus"
            }
        },
        "register":{
            "description":"Structure definition of registered object",
            "type":"object",
            "properties" : {
                "DeviceID":{
                    "description":"Registration ID refers to apeID,
application platform ID, analysis system ID, etc.",
                    "$ref":"#/$defs/deviceID"
                }
            },
            "required":["DeviceID"]
        },
        "unRegister":{
            "description":"Structure definition of de-registration object",
            "type":"object",
            "properties" : {
                "DeviceID":{
                    "description":"Logout ID refers to apeID, application
platform ID, analysis system ID, etc.",
"$ref":"#/$defs/deviceID"
                }
            },
            "required":["DeviceID"]
        },
        "keepalive":{
            "description":"Structure Definition of keep alive objects",
            "type":"object",
            "properties" : {
                "DeviceID":{
                    "description":"Logout ID refers to apeID, application
platform ID, analysis system ID, etc.",
                    "$ref":"#/$defs/deviceID"
                }
            },
            "required":["DeviceID"]
        },
        "systemTime":{
            "description":"System time object",
            "type":"object",
            "properties" : {
                "VIIDServerID":{
                    "description":"Server ID",
                    "$ref":"#/$defs/deviceID"
                },
                "TimeMode":{
                    "description":"Time calibration mode. 1: network; 2:
manual",
                    "enum":["1","2"]
                },
                "LocalTime":{
                    "description":"Date time",
                    "type":"string",
```

```
                    "format":"dateTime"
                },
                "TimeZone":{
                    "description":"Time zone",
                    "type":"string"
                }
            },
            "required":["VIIDServerID","TimeMode","LocalTime","TimeZone"]
        },
        "featureExtTask":{
            "description":"Feature vector extraction task object",
            "type":"object",
            "properties" : {
                "ExtractID":{
                    "description":"Extract task ID",
                    "$ref":"#/$defs/businessObjectId"
                },
                "IsSaved":{
                    "description":"Save settings. 0: do not save; 1: save",
                    "enum":[0,1]
                },
                "MotorVehicleListObject":{
                    "description":"List of motor vehicle objects",
                    "$ref":"/schema/motor-vehicle-list"
                },
                "NonMotorVehicleListObject":{
                    "description":"List of Non-motor vehicle objects",
                    "$ref":"/schema/nonmotor-vehicle-list"
                }
            },
            "required":["ExtractID","IsSaved"]
        },
        "featureExtTaskList":{
            "description":"Feature vector extraction task list",
            "type":"array",
            "items":{
                "$ref":"#/$defs/featureExtTask"
            }
        }
    }
}
```

A.2.2 /schema/ape-list

```
{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
    "$id":"http://www.itu.int/schema/ape-list",
    "description":"Structure definition of acquisition device list",

    "type":"array",
    "items":{
        "$ref":"#/$defs/ape"
    },
    "$defs":{
        "ape":{
            "type":"object",
            "description":"Acquisition device",
            "properties" : {
                "ApeID":{
                    "description":"Camera ID",
                    "$ref":"/schema/components#/$defs/deviceID"
                },
                "Name":{
                    "description":"Camera name",
                    "maxLength":100,
```

```
                    "type":"string"
                },
                "Model":{
                    "description":"Camera model",
                    "type":"string",
                    "maxLength":100
                },
                "IPAddr":{
                    "type":"string",
                    "format":"ipv4"
                },
                "IPV6Addr":{
                    "type":"string",
                    "format":"ipv6"
                },
                "Port":{
                    "type":"integer"
                },
                "Longtitude":{
                    "description":"Earth latitude and longitude coordinates,
accurate to 6 decimal places",
                    "type":"number",
                    "maximum":180,
                    "minimum":-180
                },
                "Latitude":{
                    "description":"Earth latitude and longitude coordinates,
accurate to 6 decimal places",
                    "type":"number",
                    "maximum":90,
                    "minimum":-90
                },
                "PlaceCode":{
                    "description":"Region Code",
                    "type":"string"
                },
                "Place":{
                    "description":"Description of the specific installation
location of the camera",
                    "type":"string"
                },
                "DepartCode":{
                    "description":"Department or industry code of the
camera",
                    "type":"string"
                },
                "FuncType":{
                    "description":"Camera function type",
                    "type":"string"
                },
                "PositionType":{
                    "description":"Camera acquisition position type",
                    "type":"int"
                },
                "CapDirection":{
                    "description":"Vehicle capture direction, 0: rear of the
car; 1: back rear of the car, compatible with non-video tollgate information
equipment",
                    "enum":[0,1]
                },
                "MonitorDirection":{
                    "description":"Monitoring direction",
                    "$ref":"/schema/components#/$defs/hDirection"
                },
```

```
                    "MonitorAreaDesc":{
                         "description":"Monitoring area description",
                         "type":"string"
                    },
                    "IsOnline":{
                         "$ref":"/schema/components#/$defs/statusType"
                    },
                    "OwnerApsID":{
                         "description":"ID of the owning acquisition system",
                         "deprecated":true,
                         "$ref":"/schema/components#/$defs/deviceID"
                    },
                    "UserId":{
                         "description":"User account, used to modify the login
account of the acquisition device",
                         "type":"string"
                    },
                    "Password":{
                         "description":"Password, used to modify the login
password of the acquisition device",
                         "type":"string",
                         "maxLength":32
                    }
               },

     "required":["ApeID","Name","Model","IPAddr","Port","Longtitude","Latitude",
"PlaceCode","IsOnline"]
          },
          "apeStatus":{
               "type":"object",
               "description":"Acquisition device status object",
               "properties" : {
                    "ApeID":{
                         "description":"Camera ID",
                         "$ref":"/schema/components#/$defs/deviceID"
                    },
                    "IsOnline":{
                         "$ref":"/schema/components#/$defs/statusType"
                    },
                    "CurrentTime":{
                         "description":"Current time",
                         "type":"string",
                         "format":"dateTime"
                    }
               },
               "required":["ApeID","IsOnline","CurrentTime"]
          },
          "apeStatusList":{
               "description":"Acquisition device status list",
               "type":"array",
               "items":{
                    "$ref":"#/$defs/apeStatus"
               }
          }
     }
}
```

A.2.3 /schema/tollgate-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/tollgate-list",
     "description":"The structure definition of the tollgate list",

     "type":"array",
```

```
    "items":{
        "$ref":"#/$defs/tollgate"
    },
    "$defs":{
        "tollgate":{
            "type":"object",
            "description":"tollgate, equipped with points for video image
information acquisition and identification device for moving vehicles,
pedestrians, etc.",
            "properties" : {
                "TollgateID":{
                    "description":"Tollgate ID",
                    "$ref":"/schema/components#/$defs/deviceID"
                },
                "Name":{
                    "description":"Tollgate name",
                    "maxLength":100,
                    "type":"string"
                },
                "Longtitude":{
                    "$ref":"/schema/ape-
list#/$defs/ape/properties/Longtitude"
                },
                "Latitude":{
                    "$ref":"/schema/ape-list#/$defs/ape/properties/Latitude"
                },
                "PlaceCode":{
                    "description":"Installation location area code",
                    "type":"string"
                },
                "Place":{
                    "description":"Description of the specific installation
location of the tollgate",
                    "type":"string"
                },
                "Status":{
                    "description":"Tollgate status",
                    "$ref":"/schema/components#/$defs/statusType"
                },
                "TollgateUsage":{
                    "description":"Use of tollgate, 80: security tollgate;
81: traffic tollgate; 82: other",
                    "enum":["80","81","82"]
                },
                "LaneNum":{
                    "description":"Number of tollgate lanes",
                    "type":"integer"
                },
                "ActiveTime":{
                    "description":"Tollgate activation time",
                    "type":"string",
                    "format":"dateTime"
                }
            },

    "required":["TollgateID","Name","Longtitude","Latitude","PlaceCode","Status
","TollgateCat","TollgateUsage"]
        }
    }
}
```

A.2.4 /schema/lane-list

```
{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
```

```
    "$id":"http://www.itu.int/schema/lane-list",
    "description":"Structure definition of lane list",

    "type":"array",
    "items":{
        "$ref":"#/$defs/lane"
    },
    "$defs":{
        "lane":{
            "type":"object",
            "description":"Tollgate, equipped with points for video image
information acquisition and identification device for moving vehicles,
pedestrians, etc.",
            "properties" : {
                "TollgateID":{
                    "$ref":"/schema/tollgate-
list#/$defs/tollgate/properties/TollgateID"
                },
                "LaneNo":{
                    "description":"Lane number",
                    "type":"integer"
                },
                "Name":{
                    "description":"Lane name",
                    "maxLength":100,
                    "type":"string"
                },
                "Direction":{
                    "description":"Lane direction",
                    "$ref":"/schema/components#/$defs/hDirection"
                },
                "Desc":{
                    "description":"Lane description",
                    "type":"string"
                },
                "MaxSpeed":{
                    "description":"Speed limit",
                    "type":"integer"
                },
                "CityPass":{
                    "description":"Code for the direction of entering and
leaving the city, 1: enter the city; 2: out of the city; 3: mixed in and out of
the city",
                    "enum":[1,2,3]
                },
                "ApeID":{
                    "$ref":"/schema/ape-list#/$defs/ape/properties/ApeID"
                }
            },
            "required":["TollgateID","LaneNo","Name","Direction"]
        }
    }
}
```

A.2.5 /schema/video-slice-info-list

```
{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
    "$id":"http://www.itu.int/schema/video-slice-info-list",
    "description":"Structure definition of video slice list",

    "type":"array",
    "items":{
        "$ref":"#/$defs/videoSliceInfo"
    },
```

```
    "$defs":{
        "videoSliceInfo":{
            "type":"object",
            "description":"Video slice object content definition",
            "properties" : {
                "VideoID":{
                    "description":"Tollgate ID",
                    "$ref":"/schema/components#/$defs/basicObjectId"
                },
                "InfoKind":{
                    "$ref":"/schema/components#/$defs/infoKind"
                },
                "VideoSourcePos":{
                    "description":"Video capture part, key public area part,
A\\d{6}",
                    "type":"string",
                    "deprecated":true
                },
                "IsAbstractVideo":{
                    "description":"Abstract video flag, true: abstract
video; false: original video",
                    "type":"boolean"
                },
                "EventSort":{
                    "description":"Video image content analysis event type.
01: car passing; 02: cars tripping wires; 03: cars intruding; 04: cars
wandering; 05: cars and objects disappear; 06: vehicles in violation of traffic
regulations; 07: objects left behind; 08: cars are stranded; 99: Others",
                    "enum":["01","02","03","04","05","06","07","08","99"]
                },
                "DeviceID":{
                    "description":" Acquisition device code",
                    "$ref":"/schema/components#/$defs/deviceID"
                },
                "StoragePath":{
                    "description":"Storage path",
                    "$ref":"/schema/components#/$defs/storagePath"
                },
                "ThumbnailStoragePath":{
                    "description":"Thumbnail storage path",
                    "$ref":"/schema/components#/$defs/storagePath"
                },
                "FileHash":{
                    "description":"Video slice hash",
                    "type:":"string",
                    "oneOf":[
                        {    "maxLength":128,    "minLength":128       },
                        {    "maxLength":256,    "minLength":256       }]
                },
                "FileFormat":{
                    "description":"Video format. Mpg:MPEG-1 MPG;M2ts:MPEG-2
TS;Ps:MPEG-2 PS",

"enum":["Mpg","Mov","Avi","Rm","Rmvb","Flv","Vob","M2ts","Mp4","Es","Ps","Ts","W
mv","Other"]
                },
                "VideoEncodeFormat":{
                    "description":"coding format. 01:AVC/H.264;02:MPEG-
4;03:MPEG-2;04:MJPEG;05:H.263;06:HEVC/H.265; 99:Other",
                    "enum":["01","02","03","04","05","06","99"]
                },
                "AudioFlag":{
                    "description":"Audio flag",
                    "$ref":"/schema/components#/$defs/flag"
```

```
                    },
                    "AudioEncodeFormat":{
                         "description":"Audio coding format.
01:G.711a;02:G.711u;03:G.723;04:G.729;05:G.726;06:AAC ;99:other",
                         "enum":["01","02","03","04","05","06","99"]
                    },
                    "Title":{
                         "description":"Video title description",
                         "type":"string"
                    },
                    "TitleNote":{
                         "description":"Title supplements and remarks",
                         "type":"string"
                    },
                    "SpecialName":{
                         "description":"Name of the subject to which the video
belongs",
                         "type":"string"
                    },
                    "Keyword":{
                         "description":"Words or phrases with retrieval
significance that express the main content of video material",
                         "type":"string"
                    },
                    "ContentDescription":{
                         "description":"Brief description of video content.",
                         "type":"string"
                    },
                    "ShotPlaceCode":{
                         "description":"Shooting location area code",
                         "$ref":"/schema/ape-
list#/$defs/ape/properties/PlaceCode"
                    },
                    "ShotPlaceFullAdress":{
                         "$ref":"/schema/ape-list#/$defs/ape/properties/Place"
                    },
                    "ShotPlaceLongitude":{
                         "$ref":"/schema/ape-
list#/$defs/ape/properties/Longitude"
                    },
                    "ShoPlacetLatitude":{
                         "$ref":"/schema/ape-list#/$defs/ape/properties/Latitude"
                    },
                    "HorizontalShotDirection":{
                         "description":"Horizontal shooting direction",
                         "$ref":"/schema/components#/$defs/hDirection"
                    },
                    "VerticalShotDirection":{
                         "description":"Vertical shooting direction",
                         "$ref":"/schema/components#/$defs/vDirection"
                    },
                    "SecurityLevel":{
                         "$ref":"/schema/components#/$defs/securityLevel"
                    },
                    "VideoLen":{
                         "description":"Duration of video slice, in seconds.",
                         "type":"number",
                         "format":"long",
                         "minimum":1,
                         "maximum":99999999
                    },
                    "BeginTime":{
                         "description":"Video start time",
                         "type":"string",
```

```
                    "format":"dateTime"
                },
                "EndTime":{
                    "description":"Video end time",
                    "type":"string",
                    "format":"dateTime"
                },
                "TimeErr":{
                    "description":"Time error, in seconds",
                    "type":"integer"
                },
                "Width":{
                    "description":"Width in pixels",
                    "type":"integer"
                },
                "Height":{
                    "description":"Height in pixels",
                    "type":"integer"
                },
                "QualityGrade":{
                    "description":"Video quality level. 1: worst; 2: bad; 3:
normal; 4: fine; 5: best",
                    "enum":[1,2,3,4,5]
                },
                "CollectorName":{
                    "description":"The name of the person who collected the
video data or the name of the acquisition  system/platform",
                    "type":"string"
                },
                "CollectorOrg":{
                    "description":"The name of the video acquisition unit",
                    "type":"string"
                },
                "CollectorPhone":{
                    "description":"Contact number of the person who
collected the video.",
                    "type":"string"
                },
                "EntryClrk":{
                    "description":"The name of the entry person of the video
or the name of the entry system",
                    "type":"string"
                },
                "EntryClrkOrg":{
                    "description":"The name of storage unit of video
materials",
                    "type":"string"
                },
                "EntryTime":{
                    "description":"Entry time, this attribute is not needed
in the POST method, it is automatically generated when creating",
                    "type":"string",
                    "format":"dateTime"
                },
                "VideoProcFlag":{
                    "description":"Video processing flag",
                    "$ref":"/schema/components#/$defs/flag"
                },
                "FileSize":{
                    "description":"Video slice size, in bytes",
                    "type":"number",
                    "format":"long"
                },
                "FileData":{
```

```
                    "$ref":"/schema/components#/$defs/binary"
               }
          },


     "required":["InfoKind","VideoSourcePos","FileFormat","VideoEncodeFormat","A
udioFlag","Title","ContentDescription","ShotPlaceCode","ShotPlaceFullAdress","Vi
deoLen","BeginTime","EndTime","TimeErr","Width","Height"],

          "allOf":[
          {
               "if":{
                    "properties":{"IsAbstractVideo":true  }
               },
               "then":{
                    "properties":{
                         "anyOf":[{
                              "OriginVideoID":{
                                   "description":"Original video ID",

     "$ref":"/schema/components#/$defs/basicObjectId"
                              }
                         },{
                              "OriginVideoURL":{
                                   "description":"Original video URL",

     "$ref":"/schema/components#/$defs/storagePath"
                              }
                         }]
                    }
               }
          },
          {
               "if":{
                    "properties":{"InfoKind":1   }
               },
               "then":{
                    "required":["EventSort"]
               }
          },
          {
               "if":{
                    "properties":{"InfoKind":2   }
               },
               "then":{

     "required":["VideoID","CollectorName","CollectorOrg","EntryClrk","EntryClrk
Org"]
               }
          }
          ]

     }
}
```

A.2.6 /schema/image-info-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/image-info-list",
     "description":"Structure definition of image list",

     "type":"array",
     "items":{
```

```
          "$ref":"#/$defs/imageInfo"
     },
     "$defs":{
          "imageInfo":{
               "type":"object",
               "description":"Image content definition",
               "properties" : {
                    "ImageID":{
                         "description":"Image ID",
                         "$ref":"/schema/components#/$defs/basicObjectId"
                    },
                    "InfoKind":{
                         "$ref":"/schema/components#/$defs/infoKind"
                    },
                    "ImageSourcePos":{
                         "description":"Image acquisition part, key public area
part, A\\d{6}",
                         "type":"string",
                         "deprecated":true
                    },
                    "SourceVideoID":{
                              "description":"Source video ID",
                              "$ref":"/schema/components#/$defs/basicObjectId"
                    },
                    "OriginImageID":{
                         "description":"Original image ID",
                         "$ref":"/schema/components#/$defs/basicObjectId"
                    },
                    "EventSort":{
                         "$ref":"/schema/video-slice-info-
list#/$defs/videoSliceInfo/properties/EventSort"
                    },
                    "DeviceID":{
                         "description":" Acquisition device code",
                         "$ref":"/schema/components#/$defs/deviceID"
                    },
                    "StoragePath":{
                         "description":"Storage path",
                         "$ref":"/schema/components#/$defs/storagePath"
                    },
                    "FileHash":{
                         "$ref":"/schema/video-slice-info-
list#/$defs/videoSliceInfo/properties/FileHash"
                    },
                    "FileFormat":{
                         "description":"Image format",

     "enum":["Bmp","Gif","Jpeg","Jfif","Kdc","Pcd","Pcx","Pic","Pix","Png","Psd"
,"Tapga","Tiff","Wmf","Jp2","Other"]
                    },
                    "ShotTime":{
                         "description":"Shooting Time",
                         "type":"string",
                         "format":"dateTime"
                    },
                    "Title":{
                         "description":"Description of image name",
                         "type":"string"
                    },
                    "TitleNote":{
                         "description":"Title supplements and remarks",
                         "type":"string"
                    },
                    "SpecialName":{
```

```
                    "description":"The name of the subject to which the
video belongs",
                    "type":"string"
                },
                "Keyword":{
                    "description":"Words or phrases with retrieval
significance that express the main content of the video",
                    "type":"string"
                },
                "ContentDescription":{
                    "description":"Brief description of video content",
                    "type":"string"
                },
                "ShotPlaceCode":{
                    "description":"Shooting location area code",
                    "$ref":"/schema/ape-
list#/$defs/ape/properties/PlaceCode"
                },
                "ShotPlaceFullAdress":{
                    "$ref":"/schema/ape-list#/$defs/ape/properties/Place"
                },
                "ShotPlaceLongitude":{
                    "$ref":"/schema/ape-
list#/$defs/ape/properties/Longitude"
                },
                "ShoPlacetLatitude":{
                    "$ref":"/schema/ape-list#/$defs/ape/properties/Latitude"
                },
                "HorizontalShotDirection":{
                    "description":"Horizontal shooting direction",
                    "$ref":"/schema/components#/$defs/hDirection"
                },
                "VerticalShotDirection":{
                    "description":"Vertical shooting direction",
                    "$ref":"/schema/components#/$defs/vDirection"
                },
                "SecurityLevel":{
                    "$ref":"/schema/components#/$defs/securityLevel"
                },
                "Width":{
                    "description":"Width in pixels",
                    "type":"integer"
                },
                "Height":{
                    "description":"Height in pixels",
                    "type":"integer"
                },
                "CameraManufacturer":{
                    "description":"manufacturer name, if the photo is from a
camera, fill in the camera manufacturer",
                    "type":"string"
                },
                "CameraVersion":{
                    "description":"Camera brand and model. If the photo is
from a camera, fill in the model of the camera.",
                    "type":"string"
                },
                "ApertureValue":{
                    "description":"Aperture value",
                    "type":"integer"
                },
                "ISOSensitivity":{
                    "description":"ISO sensitivity value",
                    "type":"integer"
```

```
                    },
                    "FocalLength":{
                         "description":"Focal length",
                         "type":"integer"
                    },
                    "QualityGrade":{
                         "description":"Video quality level. 1: worst; 2: bad; 3:
normal; 4: fine; 5: best",
                         "enum":[1,2,3,4,5]
                    },
                    "CollectorName":{
                         "description":"The name of the person who collected the
video data or the name of the acquisition system/platform",
                         "type":"string"
                    },
                    "CollectorOrg":{
                         "description":"The name of the video data acquisition
unit",
                         "type":"string"
                    },
                    "CollectorPhone":{
                         "description":"Contact number of the person who
collected the video.",
                         "type":"string"
                    },
                    "EntryClrk":{
                         "description":"The name of the entry person of the video
or the name of the entry system",
                         "type":"string"
                    },
                    "EntryClrkOrg":{
                         "description":"The name of storage unit of video",
                         "type":"string"
                    },
                    "EntryTime":{
                         "description":"Entry time, this attribute is not needed
in the POST method, it is automatically generated when creating",
                         "type":"string",
                         "format":"dateTime"
                    },
                    "ImgProcFlag":{
                         "description":"Image processing flag",
                         "$ref":"/schema/components#/$defs/flag"
                    },
                    "FileSize":{
                         "description":"Video slice size, in bytes",
                         "type":"number",
                         "format":"long"
                    },
                    "FileData":{
                         "$ref":"/schema/components#/$defs/binary"
                    }
               },


     "required":["InfoKind","ImageSourcePos","FileFormat","ShotTime","Title","Co
ntentDescription","ShotPlaceFullAdress","SecurityLevel","Width","Height"],

               "allOf":[
               {
                    "if":{
                         "properties":{"InfoKind":1   }
                    },
                    "then":{
```

```
                    "required":["EventSort"]
               }
          },
          {
               "if":{
                    "properties":{"InfoKind":2   }
               },
               "then":{

     "required":["ImageID","ShotPlaceCode","CollectorName","CollectorOrg","Entry
Clrk","EntryClrkOrg"]
               }
          }
          ]

     }
}
```

A.2.7 /schema/motor-vehicle-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/motor-vehicle-list",
     "description":"Structure definition of motor vehicle list",

     "type":"array",
     "items":{
          "$ref":"#/$defs/motorVehicle"
     },
     "$defs":{
          "motorVehicle":{
               "type":"object",
               "description":"Motor vehicle description information definition",
               "properties" : {
                    "MotorVehicleID":{
                         "description":"Motor vehicle ID",
                         "$ref":"/schema/components#/$defs/imageCntObjectId"
                    },
                    "InfoKind":{
                         "$ref":"/schema/components#/$defs/infoKind"
                    },
                    "SourceID":{
                         "description":"The VideoID or imageID of the source of
the collector object",
                         "$ref":"/schema/components#/$defs/basicObjectId"
                    },
                    "TollgateID":{
                         "description":"Associated tollgate number",
                         "$ref":"/schema/components#/$defs/deviceID"
                    },
                    "DeviceID":{
                         "description":"Device ID",
                         "$ref":"/schema/components#/$defs/deviceID"
                    },
                    "StorageUrl1":{
                         "description":"Close-up photo",
                         "$ref":"/schema/components#/$defs/storagePath"
                    },
                    "StorageUrl2":{
                         "description":"License plate photo.",
                         "$ref":"/schema/components#/$defs/storagePath"
                    },
                    "LeftTopX":{
```

```
                    "description":"X-axis coordinate of the upper left
corner of the small picture of the motor vehicle",
                    "$ref":"/schema/components#/$defs/imageCoordinates"
               },
               "LeftTopY":{
                    "description":"Y-axis coordinate of the upper left
corner of the small picture of the motor vehicle",
                    "$ref":"/schema/components#/$defs/imageCoordinates"
               },
               "RightBtmX":{
                    "description":"X-axis coordinate of the lower right
corner of the small picture of the motor vehicle",
                    "$ref":"/schema/components#/$defs/imageCoordinates"
               },
               "RightBtmY":{
                    "description":"Y-axis coordinate of the lower right
corner of the small picture of the motor vehicle",
                    "$ref":"/schema/components#/$defs/imageCoordinates"
               },
               "LaneNo":{
                    "$ref":"/schema/lane-list#/$defs/lane/properties/LaneNo"
               },
               "HasPlate":{
                    "description":"There is a license plate or not",
                    "$ref":"/schema/components#/$defs/flag"
               },
               "PlateClass":{
                    "description":"Types of license plates. 01: large car
license plates; 02: small car license plates; 99: other license plates",
                    "enum":["01","02","99"]
               },
               "PlateColor":{
                    "description":"License plate color",
                    "$ref":"/schema/components#/$defs/color"
               },
               "PlateNo":{
                    "description":"License plate number",
                    "type":"string"
               },
               "PlateNoAttach":{
                    "description":"Trailer license plate number",
                    "type":"string"
               },
               "PlateDescribe":{
                    "description":"License plate description, license plate
frame advertising information, including dealership, contact number, etc.",
                    "type":"string"
               },
               "IsAltered":{
                    "description":"License plate altered or not",
                    "$ref":"/schema/components#/$defs/flag"
               },
               "IsCovered":{
                    "description":"License plate covered or not",
                    "$ref":"/schema/components#/$defs/flag"
               },
               "Speed":{
                    "description":"Speed, in kilometers per hour.",
                    "type":"double"
               },
               "Direction":{
                    "description":"Driving Direction",
                    "$ref":"/schema/components#/$defs/hDirection"
               },
```

```
                    "VehicleModel":{
                         "description":"Vehicle model",
                         "type":"string",
                         "examples":["Passat,Volvo,BMW"]
                    },
                    "VehicleStyles":{
                         "description":"Vehicle year",
                         "examples":["2022,2021"],
                         "type":"string"
                    },
                    "VehicleColor":{
                         "description":"Vehicle color",
                         "$ref":"/schema/components#/$defs/color"
                    },
                    "VehicleColorDepth":{
                         "description":"Vehicle color depth. 0: dark; 1: light",
                         "enum":[0,1]
                    },
                    "VehicleFrontItem":{
                         "description":"Items on the front of the vehicle. 1:
logo, detection mark; 2: perfume; 3: ornaments; 4: main sun visor; 99: Other",
                         "enum":[1,2,3,4,99]
                    },
                    "DescOfFrontItem":{
                         "description":"Description of items on the front of the
vehicle",
                         "type":"string"
                    },
                    "VehicleRearItem":{
                         "description":"Items on the rear of the vehicle. 1:
Pillow; 2: Umbrella; 3: Paper towel (box); 4: Car sticker; 5: Logo; 99: other",
                         "enum":[1,2,3,4,5,99]
                    },
                    "DescOfRearItem":{
                         "description":"Description of items on the rear of the
vehicle",
                         "type":"string"
                    },
                    "PassTime":{
                         "description":"Passing time",
                         "type":"string",
                         "format":"dateTimeMS"
                    },
                    "Sunvisor":{
                         "description":"The sun visor is put down or not",
                         "$ref":"/schema/components#/$defs/flag"
                    },
                    "SafetyBelt":{
                         "description":"The driver is wearing a seat belt or
not",
                         "$ref":"/schema/components#/$defs/flag"
                    },
                    "Calling":{
                         "description":"The driver is calling or not",
                         "$ref":"/schema/components#/$defs/flag"
                    },
                    "PlateReliability":{
                         "description":"The recognition reliability of the entire
license plate number is expressed as a percentage from 0 to 100. The larger the
value, the higher the reliability.",
                         "type":"string"
                    },
                    "PlateCharReliability":{
```

```
                    "description":"The recognition credibility of the number
plate number is expressed as a percentage from 0 to 100. The larger the value,
the higher the credibility. Arranged in the manner of <character 1-credibility
1, character 2-credibility 2>, the middle is English half-width connecting line,
comma",
                    "type":"string",
                    "examples":["B-90","1-90"]
                },
                "SubImageInfoListObject":{
                    "$ref":"/schema/subimage-info-list"
                },
                "FeatureInfoListObject":{
                    "$ref":"/schema/feature-ext-result-
list#/$defs/featureInfoList"
                },
                "TabID":{
                    "description":"Attribution category label",
                    "$ref":"/schema/components#/$defs/basicObjectId"
                }
            },

            "required":["MotorVehicleID","InfoKind","SourceID","StorageUrl1"],


            "if":{
                "properties":{"InfoKind":1   }
            },
            "then":{

    "required":["DeviceID","LeftTopX","LeftTopY","RightBtmX","RightBtmY"]
            }

        }
    }
}
```

A.2.8 /schema/nonmotor-vehicle-list

```
{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
    "$id":"http://www.itu.int/schema/nonmotor-vehicle-list",
    "description":"Structure definition of non-motor vehicle list",

    "type":"array",
    "items":{
        "$ref":"#/$defs/nonMotorVehicle"
    },
    "$defs":{
        "nonMotorVehicle":{
            "type":"object",
            "description":"Non-motor vehicle description information content
definition",
            "properties" : {
                "MotorVehicleID":{
                    "description":"Non-motor vehicle ID",
                    "$ref":"/schema/components#/$defs/imageCntObjectId"
                },
                "InfoKind":{
                    "$ref":"/schema/components#/$defs/infoKind"
                },
                "SourceID":{
                    "description":"VideoID or imageID of the source of the
non-motor vehicle object",
                    "$ref":"/schema/components#/$defs/basicObjectId"
                },
```

```
                    "DeviceID":{
                         "description":"Device ID",

                         "$ref":"/schema/components#/$defs/deviceID"
                    },
                    "LeftTopX":{
                         "description":"X-axis coordinate of the upper left
corner of the small picture of non-motor vehicle",
                         "$ref":"/schema/components#/$defs/imageCoordinates"
                    },
                    "LeftTopY":{
                         "description":"Y-axis coordinate of the upper left
corner of the small picture of the non-motor vehicle",
                         "$ref":"/schema/components#/$defs/imageCoordinates"
                    },
                    "RightBtmX":{
                         "description":"X-axis coordinate of the lower right
corner of the small picture of non-motor vehicle",
                         "$ref":"/schema/components#/$defs/imageCoordinates"
                    },
                    "RightBtmY":{
                         "description":"Y-axis coordinate of the lower right
corner of the small picture of non-motor vehicle",
                         "$ref":"/schema/components#/$defs/imageCoordinates"
                    },
                    "HasPlate":{
                         "description":"There is a license plate or not",
                         "$ref":"/schema/components#/$defs/flag"
                    },
                    "PlateClass":{
                         "$ref":"/schema/motor-vehicle-
list#/$defs/motorVehicle/properties/PlateClass"
                    },
                    "PlateColor":{
                         "description":"License plate color",
                         "$ref":"/schema/components#/$defs/color"
                    },
                    "PlateNo":{
                         "description":"License plate number",
                         "type":"string"
                    },
                    "PlateDescribe":{
                         "description":"License plate description, license plate
frame advertising information, including dealership, contact number, etc.",

                         "type":"string"
                    },
                    "NonMotorVehicleStructClass":{
                         "description":"Non-motor vehicle structure category",
                         "type":"string",
                         "pattern":"^\\d{3}$"
                    },
                    "SubImageInfoListObject":{
                         "$ref":"/schema/subimage-info-list"
                    },
                    "FeatureInfoListObject":{
                         "$ref":"/schema/feature-ext-result-
list#/$defs/featureInfoList"
                    },
                    "TabID":{
                         "description":"Attribution category label",
                         "$ref":"/schema/components#/$defs/basicObjectId"
                    }
               },
```

```
                "required":["NonMotorVehicleID","InfoKind","SourceID"],


                "if":{
                     "properties":{"InfoKind":1   }
                },
                "then":{

     "required":["DeviceID","LeftTopX","LeftTopY","RightBtmX","RightBtmY"]
                }

          }
     }
}
```

A.2.9 /schema/case-info-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/case-info-list",
     "description":"Structure definition of case list",

     "type":"array",
     "items":{
          "$ref":"#/$defs/caseInfo"
     },
     "$defs":{
          "caseInfo":{
               "type":"object",
               "description":"Case description information content definition",
               "properties" : {
                    "CaseID":{
                         "description":"Case ID",
                         "$ref":"/schema/components#/$defs/caseObjectId"
                    },
                    "CaseLinkMark":{
                         "description":"The video slice or image of the
associated case, can be multiple, if multiple, they are separated by an English
semicolon \";\"",
                         "$ref":"/schema/components#/$defs/basicObjectId"
                    },
                    "CaseName":{
                         "description":"Case name",
                         "type":"string"
                    },
                    "CaseAbstract":{
                         "description":"Abstract description of case",
                         "type":"string"
                    },
                    "CreateTime":{
                         "description":"Creation time",
                         "type":"string",
                         "format":"dateTime"
                    },
                    "PlaceCode":{
                         "description":"Case occurrence area code",
                         "type":"string"
                    },
                    "PlaceFullAddress":{
                         "description":"Detailed location description of the
case.",
                         "type":"string"
                    },
                    "Longtitude":{
```

```
                    "description":"Longitude of the case",
                    "$ref":"/schema/ape-
list#/$defs/ape/properties/Longtitude"
               },
               "Latitude":{
                    "description":"Latitude of the case",
                    "$ref":"/schema/ape-list#/$defs/ape/properties/Latitude"
               },
               "MotorVehicleIDs":{
                    "description":"Case-related motor vehicle
identification, can be multiple, if multiple, they are separated by an English
semicolon \";\"",
                    "$ref":"/schema/components#/$defs/imageCntObjectId"
               },
               "NonMotorVehicleIDs":{
                    "description":"Case-related non-motor vehicle
identification, can be multiple, if multiple, they are separated by an English
semicolon \";\"",

                    "$ref":"/schema/components#/$defs/imageCntObjectId"
               },
               "RelateCaseIdList":{
                    "description":"Case associated case identification, can
be multiple, if multiple, they are separated by an English semicolon \";\"",
                    "$ref":"/schema/components#/$defs/caseObjectId"
               }
          },

     "required":["CaseID","CaseName","CaseAbstract","CreateTime","PlaceCode","Pl
aceFullAddress"]
          }
     }
}
```

A.2.10 /schema/disposition-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/disposition-list",
     "description":"Structure definition of the disposition list",

     "type":"array",
     "items":{
          "$ref":"#/$defs/disposition"
     },
     "$defs":{
          "disposition":{
               "type":"object",
               "description":"Structure description of disposition object",
               "properties" : {
                    "DispositionID":{
                         "description":"The disposition ID is generated by DSS
when disposing, and cannot be empty when cascading calls",
                         "$ref":"/schema/components#/$defs/businessObjectId"
                    },
                    "Title":{
                         "description":"The title of disposition, describing the
theme and purpose of disposition",
                         "type":"string"
                    },
                    "DispositionCategory":{
                         "description":"Disposition category. 1: Motor vehicle;
2: Non-motor vehicle; 3: Keyword",
                         "enum":[1,2,3,4,5]
                    },
```

```
                "TargetFeature":{
                    "description":"Features of the disposed object. Value is
a combination of multiple kvs:'((k1=v1) and (k2=v2)) or (k3=v3)' Supports
regular expressions for fuzzy matching. The value and meaning of kn are, 1:
license plate; 2: license plate color; 3: vehicle brand; 4: vehicle model; 5:
vehicle model year; 6: keyword.",
                    "type":"string",
                    "examples":"2=blue"
                },
                "TargetImageURI":{
                    "description":"Disposed object image path",
                    "$ref":"/schema/components#/$defs/storagePath"
                },
                "PriorityLevel":{
                    "description":"Disposition priority, 1 means the highest
level",
                    "enum":[1,2,3]
                },
                "ApplicantName":{
                    "description":"Applicant",
                    "type":"string"
                },
                "ApplicantInfo":{
                    "description":"Applicant's contact information, mobile
phone number, email address, etc.",
                    "type":"string"
                },
                "ApplicantOrg":{
                    "description":"Applicant unit",
                    "type":"string"
                },
                "BeginTime":{
                    "description":"Disposition start time",
                    "type":"string",
                    "format":"dateTime"
                },
                "EndTime":{
                    "description":"Disposition end time",
                    "type":"string",
                    "format":"dateTime"
                },
                "CreatTime":{
                    "description":"The creation time of disposing is
automatically generated by the system",
                    "type":"string",
                    "format":"dateTime"
                },
                "OperateType":{
                    "description":"Disposition type.0: Disposition; 1:
Remove disposition",
                    "enum":[0,1]
                },
                "DispositionStatus":{
                    "description":"Disposition status. 0: Disposed; 1:
Removed disposition; 2: Disposition expired; 9: Not disposed",
                    "enum":[0,1,2,9]
                },
                "DispositionRange":{
                    "description":"Disposition range. 1: tollgate; 2: area
range",
                    "enum":[1,2]
                },
                "TollgateList":{
```

```
                        "description":"The ID of the disposed tollgate, can be
multiple, if multiple, they are separated by an English semicolon \";\"",
                        "$ref":"/schema/tollgate-
list#/$defs/tollgate/properties/TollgateID"
                    },
                    "DispositionArea":{
                        "description":"Disposition area code, can be multiple,
if multiple, they are separated by an English semicolon \";\"",
                        "type":"string"
                    },
                    "DeviceList":{
                        "description":"Disposed device ID, can be multiple, if
multiple, they are separated by an English semicolon \";\"",
                        "$ref":"/schema/ape-list#/$defs/ape/properties/ApeID"
                    },
                    "ReceiveAddr":{
                        "description":"The receiving address of the notification
information, according to the address to decide whether to receive or forward","
                        "$ref":"/schema/components#/$defs/urlPath"
                    },
                    "ReceiveMobile":{
                        "description":"Notification receiving mobile phone
number, can be multiple, if multiple, they are separated by an English semicolon
\";\"",
                        "type":"string"
                    },
                    "Reason":{
                        "description":"Reason for disposition",
                        "type":"string"
                    },
                    "DispositionRemoveOrg":{
                        "description":"Name of removing disposition unit",
                        "type":"string"
                    },
                    "DispositionRemovePerson":{
                        "description":"Removing disposition people",
                        "type":"string"
                    },
                    "DispositionRemoveTime":{
                        "description":"Removing disposition time",
                        "type":"string",
                        "format":"dateTime"
                    },
                    "DispositionRemoveReason":{
                        "description":"Reason for removing disposition",
                        "type":"string"
                    },
                    "SubImageInfoListObject":{
                        "description":"Image list",
                        "$ref":"/schema/subimage-info-list"
                    },
                    "FeatureObject":{
                        "description":"Feature list",
                        "$ref":"/schema/feature-ext-result-
list#/$defs/featureInfo"
                    },
                    "ResultImageDeclare":{
                        "description":"Return picture convention. -1: No
picture; 01: Large image of the vehicle; 02: Small color image of the license
plate; 03: Binary image of the license plate; 04: Vehicle logo; 05: Violation
composite image; 06: Passing vehicle composite image; 07: Vehicle close-up
image; 08: Non-motorized vehicle image; 100: General picture",
                        "enum":["-
1","01","02","03","04","05","06","07","08",100]
```

```
                    },
                    "ResultFeatureDeclare":{
                         "description":"Return feature vector flag. -1: No need
to return feature vector; 1: Need to return feature vector",
                         "enum":[-1,1]
                    },
                    "TabID":{
                         "description":"Disposition label flag",
                         "$ref":"/schema/components#/$defs/basicObjectIdType"
                    },
                    "AlarmSensitivity":{
                         "description":"Warning sensitivity, the higher the
sensitivity, the lower the warning threshold, the higher the false alarm rate.
1: Level 1; 2: Level 2; 3: Level 3",
                         "enum":[1,2,3]
                    }
               },
               "required":["OperateType"],

               "allOf":[
               {
                    "if":{
                         "properties":{"OperateType":0    }
                    },
                    "then":{

     "required":["Title","DispositionCategory","TargetFeature","ApplicantName","
ApplicantOrg","DispositionRange"]
                    }
               },
               {
                    "if":{
                         "properties":{"OperateType":1    }
                    },
                    "then":{
                         "required":["DispositionID"]
                    }
               }
               ]
          }
     }
}
```

A.2.11 /schema/disposition-notification-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/disposition-notification-list",
     "description":"Structure definition of notification list",

     "type":"array",
     "items":{
          "$ref":"#/$defs/dispositionNotification"
     },
     "$defs":{
          "dispositionNotification":{
               "type":"object",
               "description":"Structure description of notification list",
               "properties" : {
                    "NotificationID":{
                         "description":"Notification ID corresponding to
disposition",
                         "$ref":"/schema/components#/$defs/businessObjectId"
                    },
                    "DispositionID":{
```

```
                    "description":"Disposition ID",
                    "$ref":"/schema/components#/$defs/businessObjectId"
               },
               "Title":{
                    "description":"The title of disposition, describing the
theme and purpose of disposition",
                    "type":"string"
               },
               "TriggerTime":{
                    "description":"Trigger time",
                    "type":"string",
                    "format":"dateTime"
               },
               "MotorVehicleObject":{
                    "description":"Automatically collected motor vehicle
data",
                    "$ref":"/schema/motor-vehicle-list#/$defs/motorVehicle"
               },
               "NonMotorVehicleObject":{
                    "description":"Automatically collected non-motor vehicle
data",
                    "$ref":"/schema/nonmotor-vehicle-
list#/$defs/nonMotorVehicle"
               },
               "CaseInfoObject":{
                    "description":"Automatically collected case data",
                    "$ref":"/schema/case-info-list#/$defs/caseInfo"
               },
               "DispositionTargetID":{
                    "description":"The disposition target ID can be the
identification of the disposed vehicle, etc., or the image identification of
disposed multiple image lists.",
                    "type":"string"
               }

          },

     "required":["NotificationID","DispositionID","Title","TriggerTime"]
          }
     }
}
```

A.2.12 /schema/subscribe-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/subscribe-list",
     "description":"Structure definition of subscription list",
     "type":"array",
     "items":{
          "$ref":"#/$defs/subscribe"
     },
     "$defs":{
          "subscribe":{
               "type":"object",
               "description":"Subscription object structure description",
               "properties" : {
                    "SubscribeID":{
                         "description":"Subscription ID, generated by DSS when
desposing, cannot be empty when cascading call",
                         "$ref":"/schema/components#/$defs/businessObjectId"
                    },
                    "Title":{
                         "description":"The title of disposition, describing the
theme and purpose of disposition",
```

```
                        "type":"string"
                    },
                    "SubscribeDetail":{
                        "description":"Subscription category, can be multiple,
if multiple, usually they are separated by an English half-width','. 1:
Acquisition device catalog; 2: Acquisition device status; 3: Acquisition system
catalog ; 4: Acquisition system status; 5: Video tollgate catalog; 6: Single
bayonet record; 7: Lane catalog; 8: Single lane record; 9: Motor vehicle
information; 10: Self-non-motor vehicle information; 11: File information; 12:
Video slice; 13: Image; 14: Data classification label list",
     "enum":["1","2","3","4","5","6","7","8","9","10","11","12","13","14"],
                        "examples":"1,2"
                    },
                    "ResourceClass":{
                        "description":"Subscription resource category. 0:
tollgate; 1: device; 2: acquisition  content; 3: case; 4: image library; 5:
administrative division",
                        "enum":[0,1,2,3,4,5]
                    },
                    "ResourceURI":{
                        "description":"Subscription resource path, tollgate ID,
device ID, acquisition content ID, case ID, target image library ID, area code
2/4/6 bits, etc., can be multiple, only one resource category at a time.",
                        "type":"string"
                    },
                    "ApplicantName":{
                        "description":"Applicant",
                        "type":"string"
                    },
                    "ApplicantOrg":{
                        "description":"Applicant unit",
                        "type":"string"
                    },
                    "BeginTime":{
                        "description":"Subscription start time",
                        "type":"string",
                        "format":"dateTime"
                    },
                    "EndTime":{
                        "description":"Subscription end time",
                        "type":"string",
                        "format":"dateTime"
                    },
                    "ReceiveAddr":{
                        "description":"Notification information receiving
address, according to the address to decide whether to receive or forward",
                        "$ref":"/schema/components#/$defs/urlPath"
                    },
                    "ReportInterval":{
                        "description":"Report interval, in seconds, <=0 means no
limit",
                        "type":"integer"
                    },
                    "Reason":{
                        "description":"Reason for subscription",
                        "type":"string"
                    },
                    "OperateType":{
                        "description":"Subscription type. 1: Subscription; 2:
Cancel subscription",
                        "enum":[1,2]
                    },
                    "SubscribeStatus":{
```

```
                    "description":"Subscription status. 1: subscribing; 2:
canceled subscription; 3: subscription expired; 4: not subscribed; 9: other",
                    "enum":[1,2,3,4,9]
               },
               "SubscribeCancelOrg":{
                    "description":"Unsubscribe unit name",
                    "type":"string"
               },
               "SubscribeCancelPerson":{
                    "description":"Subscription canceller",
                    "type":"string"
               },
               "CancelTime":{
                    "description":"Cancel subscription time",
                    "type":"string",
                    "format":"dateTime"
               },
               "CancelReason":{
                    "description":"Reason for canceling subscription",
                    "type":"string"
               },
               "ResultImageDeclare":{
                    "description":"Return picture convention. 00: No
picture; 01: Large image of the vehicle; 02: Small color image of the license
plate; 03: Binary image of the license plate; 04: Car logo; 05: Violation
composite image; 06: Passing vehicle composite image; 07: Vehicle close-up
image;08: Non-motorized vehicle image; 100 :General picture",
     "enum":["00","01","02","03","04","05","06","07","08","09","10","100"]
               },
               "ResultFeatureDeclare":{
                    "description":"feature vector declare. 0: No need to
return feature vector; 1: Need to return feature vector ",
                    "enum":[0,1]
               },
               "TabID":{
                    "description":"Subscription category label
identification",
                    "$ref":"/schema/components#/$defs/basicObjectIdType"
               }
          },

     "required":["OperateType","ResultImageDeclare","ResultFeatureDeclare"],

          "allOf":[
          {
               "if":{
                    "properties":{"OperateType":0    }
               },
               "then":{

     "required":["Title","SubscribeDetail","ResourceClass","ResourceURI","Applic
antName","ApplicantOrg","BeginTime","EndTime"]
               }
          },
          {
               "if":{
                    "properties":{"OperateType":1    }
               },
               "then":{
                    "required":["SubscribeID"]
               }
          }
          ]
     }
```

```
    }
}
```

A.2.13 /schema/subscribe-notification-list

```
{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
    "$id":"http://www.itu.int/schema/subscribe-notification-list",
    "description":"Structure definition of notification list",

    "type":"array",
    "items":{
        "$ref":"#/$defs/subscribeNotification"
    },
    "$defs":{
        "subscribeNotification":{
            "type":"object",
            "description":"Notification object structure description",
            "properties" : {
                "NotificationID":{
                    "description":"Subscribe to the corresponding
notification ID",
                    "$ref":"/schema/components#/$defs/businessObjectId"
                },
                "SubscribeID":{
                    "description":"Subscription ID",
                    "$ref":"/schema/components#/$defs/businessObjectId"
                },
                "Title":{
                    "description":"The title of disposition, describing the
theme and purpose of arming",
                    "type":"string"
                },
                "TriggerTime":{
                    "description":"Trigger time",
                    "type":"string",
                    "format":"dateTime"
                },
                "InfoIDs":{
                    "description":"Identification acquisition of detailed
information (motor vehicle, non-motor vehicle) of subscription notification",
                    "type":"string"
                },
                "CaseListobject":{
                    "description":"Case data acquisition",
                    "$ref":"/schema/case-info-list"
                },
                "Tollgate":{
                    "description":"Tollgate data acquisition",
                    "$ref":"/schema/tollgate-list"
                },
                "Lane":{
                    "description":"Lane data acquisition",
                    "$ref":"/schema/lane-list"
                },
                "DeviceList":{
                    "description":" Acquisition device data acquisition",
                    "$ref":"/schema/ape-list"
                },
                "DeviceStatusList":{
                    "description":" Acquisition device status data
acquisition",
                    "$ref":"/schema/ape-list#/$defs/apeStatusList"
                },
                "MotorVehicleListobject":{
```

```
                    "description":"Motor vehicle data acquisition",
                    "$ref":"/schema/motor-vehicle-list"
               },
               "NonMotorVehicleListobject":{
                    "description":"Non-motor vehicle data acquisition",
                    "$ref":"/schema/nonmotor-vehicle-list"
               },
               "DataClassTabListobject":{
                    "description":"Automatically collected case data",
                    "$ref":"/schema/componets#/$defs/dataClassTabList"
               },
               "ExecuteOperation":{
                    "description":"Update operation. 1: add; 2: modify; 3:
delete",
                    "enum":[1,2,3]
               }

          },

     "required":["NotificationID","DispositionID","Title","TriggerTime"]
          }
     }
}
```

A.2.14 /schema/subimage-info-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/subimage-info-list",
     "description":"Structure definition of sub-image list",

     "type":"array",
     "items":{
          "$ref":"#/$defs/subImageInfo"
     },
     "$defs":{
          "subImageInfo":{
               "type":"object",
               "description":"Image content definition",
               "properties" : {
                    "ImageID":{
                         "description":"Image ID",
                         "$ref":"/schema/components#/$defs/basicObjectId"
                    },
                    "EventSort":{
                         "$ref":"/schema/video-slice-info-
list#/$defs/videoSliceInfo/properties/EventSort"
                    },
                    "DeviceID":{
                         "description":"Acquisition device code",
                         "$ref":"/schema/components#/$defs/deviceID"
                    },
                    "StoragePath":{
                         "description":"Storage path",
                         "$ref":"/schema/components#/$defs/storagePath"
                    },
                    "Type":{
                         "description":"Image file hash value. 01: Motor vehicle
image; 02: License plate color small image; 03: License plate binarization
image; 04: Vehicle logo image; 05: Violation composite image; 06: Passing
vehicle composite image; 07: Vehicle close-up image;08: Non-motorized vehicle
image; 09: Object image; 10: Vehicle Feature map, binary feature map; 11: non-
motor vehicle feature map; 12: general feature map; 100: general picture",
     "enum:":["01","02","03","04","05","06","07","08","09","10","11","12","13","
100"]
```

```
                    },
                    "FileFormat":{
                         "$ref":"/schema/image-info-
list#/$defs/imageInfo/properties/FileFormat"
                    },
                    "ShotTime":{
                         "description":"Shooting time",
                         "type":"string",
                         "format":"dateTime"
                    },
                    "Width":{
                         "description":"Width in pixels, in pixels",
                         "type":"integer"
                    },
                    "Height":{
                         "description":"Height in pixels, in pixels",
                         "type":"integer"
                    },
                    "Data":{
                         "$ref":"/schema/components#/$defs/base64"
                    },
                    "FeatureInfoObject":{
                         "$ref":"/schema/feature-ext-result-
list#/$defs/featureInfo"
                    }
               }

          }
     }
}
```

A.2.15 /schema/image-searched-by-image

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/image-searched-by-image",
     "description":"Structure definition of object searched by image",

     "type":"object",

     "properties" : {
          "SearchID":{
               "description":"Search ID, the 13th to 14th digits take 11 to
indicate that the target is searched by image",
               "$ref":"/schema/components#/$defs/basicObjectId"
          },
          "SearchType":{
               "description":"Search type, specify the search target type",
               "enum":["MotorVehicle","NonMotorVehicle"]
          },
          "SubImageObject":{
               "description":"Picture information searched by pictures in the
image searching",
               "$ref":"/schema/subimage-info-list#/$defs/subImageInfo"
          },
          "FeatureObject":{
               "description":"Feature vector information searched by Feature
vector in the image searching",
               "$ref":"/schema/feature-ext-result-list#/$defs/featureInfo"
          },
          "Threshold":{
               "description":"Similarity threshold",
               "type":"number",
               "format":"double",
               "minimum":0,
```

```
                "maximum":1
            }
        },
        "required":["SearchID"]
}
```

A.2.16 /schema/image-result-SBI-list

```
{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
    "$id":"http://www.itu.int/schema/image-result-SBI-list",
    "description":"Structure definition of target result list object searched
by image",
        "type":"array",
    "items":{
        "$ref":"#/$defs/imageResultSBIList"
    },
    "$defs":{
        "imageResultSBIList":{
            "type":"object",
            "description":"Query target result object by image",
            "properties" : {
                "SearchID":{
                    "description":"Search ID corresponding to the result",
                    "$ref":"/schema/components#/$defs/basicObjectId"
                },
                "MotorVehicleListObject":{
                    "description":"Returned motor vehicle search results",
                    "$ref":"/schema/motor-vehicle-list"
                },
                "NonMotorVehicleListObject":{
                    "description":"Returned non-motor vehicle search
results",
                    "$ref":"/schema/nonmotor-vehicle-list"
                }
            },
            "required":["SearchID"]
        }
    }
}
```

A.2.17 /schema/case-list

```
{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
    "$id":"http://www.itu.int/schema/case-list",
    "description":"The definition of the structure of the video image case
acquisition list. The video image case acquisition refers to objects that
combine video image cases and associated video slices, images, motor vehicles,
non-motor vehicles, etc.",
    "type":"array",
    "items":{
        "$ref":"#/$defs/case"
    },
    "$defs":{
        "case":{
            "type":"object",
            "description":"Video image case content definition",
            "properties" : {
                "CaseInfo":{
                    "$ref":"/schema/case-info-list#/$defs/caseInfo"
                },
                "VideoSliceListObject":{
                    "$ref":"/schema/video-slice-list"
```

```
                    },
                    "ImageListObject":{
                         "$ref":"/schema/image-list"
                    },
                    "MotorVehicleListObject":{
                              "$ref":"/schema/motor-vehicle-list"
                    },
                    "NonMotorVehicleListObject":{
                         "$ref":"/schema/nonmotor-vehicle-list"
                    }
               },

               "required":["CaseInfo"]
          }
     }
}
```

A.2.18 /schema/image-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/image-list",
     "description":"Structure definition of the image acquisition list, which
are objects that combine images with associated, motor vehicles, non-motor
vehicles, etc.",
     "type":"array",
     "items":{
          "$ref":"#/$defs/image"
     },
     "$defs":{
          "image":{
               "type":"object",
               "description":"Image content definition",
               "properties" : {
                    "ImageInfo":{
                         "$ref":"/schema/image-info-list#/$defs/imageInfo"
                    },
                    "MotorVehicleListObject":{
                              "$ref":"/schema/motor-vehicle-list"
                    },
                    "NonMotorVehicleListObject":{
                         "$ref":"/schema/nonmotor-vehicle-list"
                    }
               },
               "required":["ImageInfo"]
          }
     }
}
```

A.2.19 /schema/video-slice-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/video-slice-list",
     "description":"Structure definition of video slice acquisition list, video
slice acquisition refers to the object that combines video slices and associated
motor vehicles, non-motor vehicles, etc.",
     "type":"array",
     "items":{
          "$ref":"#/$defs/videoSlice"
     },
     "$defs":{
          "videoSlice":{
               "type":"object",
               "description":"Video slice content definition",
               "properties" : {
```

```
                    "VideoSliceInfo":{
                         "$ref":"/schema/video-slice-info-list#
/$defs/videoSliceInfo"
                    },
                    "MotorVehicleListObject":{
                              "$ref":"/schema/motor-vehicle-list"
                    },
                    "NonMotorVehicleListObject":{
                         "$ref":"/schema/nonmotor-vehicle-list"
                    }
               },
               "required":["VideoSliceInfo"]
          }
     }
}
```

A.2.20 /schema/task-status-list

```
{
     "$schema":"https://json-schema.org/draft/2020-12/schema",
     "$id":"http://www.itu.int/schema/task-status-list",
     "description":"Structure definition of analysis task status",

     "type":"array",

     "items":{
          "$ref":"#/$defs/taskStatus"
     },
     "$defs":{
          "taskStatus":{
               "description":"Structure definition of analysis task status",
               "type":"object",
               "properties" : {
                    "TaskID":{
                         "description":"Task ID",
                         "$ref":"/schema/components#/$defs/basicObjectId"
                    },
                    "CommitTime":{
                         "description":"Task submission time",
                         "type":"string",
                         "format":"dateTime"
                    },
                    "BeginTime":{
                         "description":"Task start time",
                         "type":"string",
                         "format":"dateTime"
                    },
                    "ExpectEndTime":{
                         "description":"Estimated task completion time",
                         "type":"string",
                         "format":"dateTime"
                    },
                    "Status":{
                         "description":"status. 1: queuing; 2: executing; 3:
suspended; 4: completed; 9: other",
                         "enum":["1","2","3","4","9"]
                    },
                    "ConsumeTime":{
                         "description":"The effective time the task has been
executed, in seconds",
                         "type":"number",
                         "format":"integer"
                    },
                    "Progress":{
                         "description":"Task progress percentage",
```

```
                    "type":"number",
                    "format":"integer",
                    "minimum":0,
                    "maximum":100
                }
            },

            "required":["TaskID","CommitTime","BeginTime","Status"]
        }
    }
}
```

A.2.21 /schema/task-control

```
{
    "$schema":"https://json-schema.org/draft/2020-12/schema",
    "$id":"http://www.itu.int/schema/task-control",
    "description":"Structure definition of analysis task control",
    "type":"object",
    "properties" : {
        "TaskID":{
            "description":"Task ID",
            "$ref":"/schema/components#/$defs/basicObjectId"
        },
        "Control":{
            "description":"Task control code. 1: create task; 2: Start task;
3: pause task; 4: continue task; 5: modify task parameters; 6: abort task; 9:
other",
            "enum":["1","2","3","4","5","6","9"]
        }
    },

    "required":["TaskID","Control"]
}
```

Bibliography

[b-REST] *Architectural Styles and the Design of Network-based Software Architecture CHAPTER 5: Representation State Transfer*.

[b-JSON-1] *JSON Schema: A Media Type for Describing JSON Documents*. http://json-schema.org/draft/2020-12/json-schema-core.html

[b-JSON-2] *JSON Schema Validation: A Vocabulary for Structural Validation of JSON*. http://json-schema.org/draft/2020-12/json-schema-validation.html

[b-JSON-3] *Understanding JSON Schema*. http://json-schema.org/understanding-json-schema

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Published in Switzerland
Geneva, 2023