ONVIF®

Profile M Client Test Specification

Version 23.06

June 2023

REVISION HISTORY

Vers.	Date	Description
23.06	Mar 13, 2023	The following were changes according to #25: Test Cases for Profile Optional Features section was added Metadata Configuration Using Media2 (Profile M) feature was updated to be Optional
22.06	Dec 27, 2022	Profile Normative Reference were removed from test cases according to #364
21.12	Oct 12, 2021	The following was done according to #425: Check Condition based on Device Features of Discovery feature was changed from 'All' to 'Discovery'
21.06	Jun 03, 2021	The following was updated according to #325: SETSYNCHRONIZATIONPOINT-1 test name was changed from SET SYNCHRONIZATION POINT to SET SYNCHRONIZATION POINT (EVENT SERVICE). Set Synchronization Point feature was renamed to Set Synchronization Point (Event Service)
21.06	Jun 01, 2021	'Check Condition based on Device Features' sections were updated for the following features according to #383 Image Data Object Classification Human Face Metadata Vehicle Metadata License Plate Metadata GeoLocation Metadata Face Recognition Event Line Crossing Counter Event License Plate Recognition Event Event Broker Configuration
20.12	Dec 8, 2020	DEVICEDISCOVERYTYPEFILTER-1 DEVICE DISCOVERY TYPE FILTER was updated according to #406: Types value check was updated to accept QName list instead of one QName value.
20.12	Oct 27, 2020	Network Configuration feature was removed according to #395
20.12	Aug 31, 2020	Event Handling Using Pull Points Feature: Check Condition based on Device Features was changed according to #325.
20.06	Mar 30, 2020	The following features were changed according to #382

		Human Face Metadata Human Body Metadata Vehicle Metadata License Plate Metadata
20.06	Mar 13, 2020	License Plate Recognition topic was changed according to #379
20.06	Mar 13, 2020	Face Match Event feature was replaced with Face Recognition Event feature according to #379
20.06	Mar 12, 2020	Event Broker Configuration Test Cases section was added according to #372
20.06	Feb 05, 2020	Get Supported Metadata feature was added according to #368
20.06	Jan 23, 2020	Conditional feature added according to #326: System (reused feature) Metadata Profile Configuration Using Media2 (Profile M) (new feature) Analytics Profile Configuration Using Media2 (reused feature) Analytics Module Configuration (new feature) Create Profile Using Media2 (new feature) Video Streaming Using Media2 (Profile M) (new feature) Image Data (new feature) HTTPS Streaming Using Media2 (Profile M) (new feature) Event Handling Using Pull Points (new feature) Rule Configuration (new feature) Object Classification (new feature) Human Face Metadata (new feature) Human Body Metadata (new feature) Vehicle Metadata (new feature) License Plate Metadata (new feature) GeoLocation Metadata (new feature) Face Match Event (new feature) License Plate Recognition Event (new feature) Line Crossing Counter Event (new feature)
19.12	Oct 22, 2019	The following was done according to #325: Check Condition based on Device Features of HTTP Digest Authentication for RTSP feauture was changed from 'Profile T' to 'Profile T or Profile M'.
19.12	Aug 23, 2019	Initial version.

Table of Contents

1 Introduction

The goal of the ONVIF Test Specification set is to make it possible to realize fully interoperable IP physical security implementations from different vendors. This specification also acts as an input document to the development of a test tool which will be used to test the ONVIF Client implementation conformance towards ONVIF standard. This Client Test Tool analyzes network communications between ONVIF Devices and Clients being tested and determines whether a specific Client is ONVIF conformant (see ONVIF Conformance Process Specification).

This particular document defines feaures and related test cases required for testing Profile M features of a Client application e.g. metadata streaming, analytics configuration. It also describes the test framework, test setup, prerequisites, test policies needed for the execution of the described test cases.

1.1 Scope

This ONVIF Profile M Client Test Specification defines and regulates the conformance testing procedure for the ONVIF conformant Clients in the scope of Profile M features. Conformance testing is meant to be black-box network traces analysis and verification. The objective of this specification is to provide the test cases to test individual requirements of ONVIF Clients in the scope of Profile M features according to ONVIF Profile Specifications.

The principal intended purposes are:

- Provide self-assessment tool for implementations.
- Provide comprehensive test suite coverage for Profile M features.

This specification **does not** address the following:

- 3rd parties Client use cases
- Non-functional (performance and regression) testing and analysis.
- SOAP Implementation Interoperability test i.e. Web Services Interoperability Basic Profile version 2.0 (WS-I BP2.0).
- Network protocol implementation Conformance test for HTTPS and HTTP protocols.

The following sections cover test cases needed for the verification of relevant features as mentioned in the ONVIF Profile Specifications.

1.2 Test Cases for Profile Mandatory Features

This section defines test cases which are mandatory for Profile M Client conformance.

1.2.1 HTTP Digest

HTTP Digest section defines security mechanism for HTTP Digest Authentication.

1.2.2 HTTP Digest Authentication for RTSP

HTTP Digest Authentication for RTSP section defines security mechanism for Digest Authentication for RTSP.

1.2.3 Get Services

Get Services section specifies Client ability to retrieve list of services with using GetServices operation.

1.2.4 Discovery

Discovery section defines Client ability to locate services on a local network using Web Services Dynamic Discovery (WS-Discovery) protocol. It uses IP multicast address 239.255.255.250 and TCP and UDP port 3702 and SOAP-over-UDP standard for communication between nodes.

1.2.5 Device Discovery Type Filter

Device Discovery Type Filter Test Cases section defines Client ability to locate services, which are support Device Discovery Type on a local network using Web Services Dynamic Discovery (WS-Discovery) protocol. It uses IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with Types filter that contains tds:Device or with skipped Types filter.

1.2.6 Metadata Streaming Using Media2 (Profile M)

Metadata Streaming Using Media2 section specifies receiving of metadata stream from Device using Madia2 Service.

1.2.7 Metadata Configuration Using Media2 (Profile M)

Metadata Configuration Using Media2 section specifies listing and modification of metadata configurations on Device.

1.2.8 Get Supported Metadata

Get Supported Metadata section specifies Client ability to retrieve metadata SampleFrame from a device.

1.3 Test Cases for Profile Conditional Features

This section defines test cases which are mandatory for Profile M Client conformance.

1.3.1 System

System section defines Client ability to obtain Device information and configure of system settings on Device.

1.3.2 Metadata Profile Configuration Using Media2 (Profile M)

Metadata Profile Configuration Using Media2 section specifies Client ability to cofigure a media profile with video source coniguration and with metadata configuration.

1.3.3 Analytics Profile Configuration Using Media2

Analytics Profile Configuration Using Media2 section specifies Client ability to add analytics configuration to a media profile.

1.3.4 Analytics Module Configuration

Analytics Module Configuration section specifies Client ability to retrieve supported Analytics Module description, assigned Analytics Module Configurations, adding and removing of Analytics Modules to/from an existing VideoAnalyticsConfiguration.

1.3.5 Create Profile Using Media2

Create Profile Using Media2 section specifies Client ability to create a Media2 profile on Device.

1.3.6 Video Streaming Using Media2 (Profile M)

Video Streaming Using Media2 section specifies receiving of video stream from Device using Madia2 Service.

1.3.7 Image Data

Image Data section specifies Client ability to get event or metadata image via the image URI and ability to receive base64 encoding image data is supported by Device.

1.3.8 HTTPS Streaming Using Media2 (Profile M)

HTTPS Streaming Using Media2 (Profile M) section specifies Client ability to establish specific media streams over RTP/RTSP/HTTPS/TCP.

1.3.9 Event Handling Using Pull Points

Event Handling using Pull Points section defines Client ability to initiate and receive notifications (events) from a Device using realtime pullpoints.

1.3.10 Rule Configuration

Rule Configuration section specifies Client ability to configure rules.

1.3.11 Object Classification

Object Classification section specifies Client ability to retrieve metadata streaming with included Class element.

1.3.12 Human Face Metadata

Human Face Metadata section specifies Client ability to retrieve metadata streaming with included Human Face Metadata.

1.3.13 Human Body Metadata

Human Body Metadata section specifies Client ability to retrieve metadata streaming with included Human Body Metadata.

1.3.14 Vehicle Metadata

Vehicle Metadata section specifies Client ability to retrieve metadata streaming with included Vehicle Metadata.

1.3.15 License Plate Metadata

License Plate Metadata section specifies Client ability to retrieve metadata streaming with included License Plate Metadata.

1.3.16 GeoLocation Metadata

GeoLocation Metadata section specifies Client ability to retrieve metadata streaming with included GeoLocation Metadata.

1.3.17 Face Recognition Event

Face Recognition Event section specifies Client ability to receive tns1:RuleEngine/Recognition/Face events.

1.3.18 License Plate Recognition Event

License Plate Recognition Event section specifies Client ability to receive tns1:RuleEngine/Recognition/LicensePlate events.

1.3.19 Line Crossing Counter Event

Line Crossing Counter Event section specifies Client ability to receive tns1:RuleEngine/CountAggregation/Counter events.

1.3.20 Event Broker Configuration

Event Broker Configuration section defines Client ability to configure Event Broker on a Device.

1.4 Supplementary Features and Test Cases

This section defines supplementary features and test cases which are not the part of profile, but Profile M Features results depends on them.

2 Normative References

- ONVIF Conformance Process Specification:

  https://www.onvif.org/profiles/conformance/

- ONVIF Profile Policy:

  https://www.onvif.org/profiles/

- ONVIF Network Interface Specifications:

  https://www.onvif.org/profiles/specifications/

- ISO/IEC Directives, Part 2, Annex H:

  www.iso.org/directives

- ISO 16484-5:2014-09 Annex P:

  https://www.iso.org/obp/ui/#!iso:std:63753:en

- WS-BaseNotification:

  http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf

- W3C SOAP 1.2, Part 1, Messaging Framework:

  http://www.w3.org/TR/soap12-part1/

- W3C XML Schema Part 1: Structures Second Edition:

  http://www.w3.org/TR/xmlschema-1/

- W3C XML Schema Part 2: Datatypes Second Edition:

  "http://www.w3.org/TR/xmlschema-2/ [http://www.w3.org/TR/xmlschema-2/]

- W3C Web Services Addressing 1.0 – Core:

  http://www.w3.org/TR/ws-addr-core/

- ONVIF Profile M Specification:

  [TODO: put link to profile page] [http://TODO]

- IETF RFC 2326, Real Time Streaming Protocol (RTSP):

  http://www.ietf.org/rfc/rfc2326.txt

3 Terms and Definitions

3.1 Conventions

The key words "shall", "shall not", "should", "should not", "may", "need not", "can", "cannot" in this specification are to be interpreted as described in [ISO/IEC Directives Part 2].

3.2 Definitions

This section describes terms and definitions used in this document.

Address	An address refers to a URI.
Profile	See ONVIF Profile Policy.
ONVIF Device	Computer appliance or software program that exposes one or multiple ONVIF Web Services.
ONVIF Client	Computer appliance or software program that uses ONVIF Web Services.
Conversation	A Conversation is all exchanges between two MAC addresses that contains SOAP request and response.
Network	A network is an interconnected group of devices communicating using the Internet protocol.
Network Trace Capture file	Data file created by a network protocol analyzer software (such as Wireshark). Contains network packets data recorded during a live network communications.
SOAP	SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols.
Client Test Tool	ONVIF Client Test Tool that tests ONVIF Client implementation towards the ONVIF Test Specification set.
Valid Device Response	Device has responded to specific request with code HTTP or RTSP 200 OK and SOAP fault message has not appeared.
Profile M	The Profile M Specification.
Configuration Entity	A network video device media abstract component that produces or consumes a media stream on the network, i.e. video and/or audio stream.
Media Profile	Maps a video and audio sources and outputs encoders as well as PTZ and analytics configurations.
Metadata	All streaming data except video and audio, including video analytics results, PTZ position data and other metadata (such as textual data from POS applications).

Reference Token	Token provided by the device to uniquely reference an instance of a physicalIO, configuration or profile.
Video Analytics	Algorithms or programs used to analyze video data and to generate data describing object location and behaviour.

3.3 Abbreviations

This section describes abbreviations used in this document.

HTTP Hyper Text Transport Protocol.

HTTPS Hyper Text Transport Protocol over Secure Socket Layer.

IP Internet Protocol.

IPv4 Internet Protocol version 4.

TCP Transport Control Protocol.

UDP User Datagram Protocol.

URI Uniform Resource Identifier.

WSDL Web Services Description Language.

XML eXtensible Markup Language.

RTCP RTP Control Protocol.

RTP Realtime Transport Protocol.

RTSP Real Time Streaming Protocol.

SDP Session Description Protocol.

3.4 Namespaces

Prefix and namespaces used in this test specification are listed in Table 1. These prefixes are not part of the standard and an implementation can use any prefix.

Table 3.1. Defined namespaces in this specification

Prefix	Namespace URI	Description
soapenv	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [SOAP 1.2, Part 1]
xs	http://www.w3.org/2001/XMLSchema	Instance namespace as defined by XS [XML-Schema, Part1] and [XMLSchema,Part 2]
xsi	http://www.w3.org/2001/XMLSchema-instance	XML schema instance namespace
tns1	http://www.onvif.org/ver10/topics	The namespace for the ONVIF topic namespace

Prefix	Namespace URI	Description
tt	http://www.onvif.org/ver10/schema	ONVIF XML schema descriptions
tds	http://www.onvif.org/ver10/device/wsdl	The namespace for the WSDL device service
tev	http://www.onvif.org/ver10/events/wsdl	The namespace for the WSDL event service
ter	http://www.onvif.org/ver10/error	The namespace for ONVIF defined faults
wsnt	http://docs.oasis-open.org/wsn/b-2	Schema namespace of the [WS-BaseNotification] specification.
wsa	http://www.w3.org/2005/08/addressing	Device addressing namespace as defined by [WS-Addressing].
tan	http://www.onvif.org/ver20/analytics/wsdl	The namespace for the WSDL Analytics service
tr2	http://www.onvif.org/ver20/media/wsdl	The namespace for the WSDL Media 2 service

4 Test Overview

This section provides information for the test setup procedure and required prerequisites that should be followed during test case execution.

An ONVIF Client conformant to Profile M is an ONVIF Client that at least supports the following features: [TODO: add breaf description of profile features as itemizedlist]

An ONVIF Profile is described by a fixed set of functionalities through a number of services that are provided by the ONVIF standard. A number of services and functionalities are mandatory for each type of ONVIF Profile. An ONVIF Device and ONVIF Client may support any combination of Profiles and other optional services and functionalities.

4.1 General

Test Cases are grouped depending on features. Each Test Cases group provides description of feature requirement level for Profiles, expected scenario under test and related test cases:

- Feature Level Requirement
- Expected Scenarios Under Test
- List of Test Cases

4.1.1 Feature Level Requirement

Feature Level Requirement item contains a feature ID, check condition based on Device features, required number of Devices and feature requirement level for the Profiles, which will be used for Profiles conformance.

To claim this Feature as supported Client shall pass Expected Scenario Under Test:

- for each Device, which supports Device Features defined in Check Condition Based on Device Features
- for at least with number of Devices specified in Required Number of Devices

If Feature Level Requirement is defined as Mandatory for some Profile, Client shall support this Feature to claim this Profile Conformance.

4.1.2 Expected Scenarios Under Test

Expected Scenarios Under Test item contains expected scenario under test, conditions when the feature will be defined as supported and as not supported.

4.1.3 Test Cases

Test Case items contain list of test cases which are related to feature. Test cases provide exact procedure of testing feature support conditions.

Each Test Case contains the following parts:

- Test Label - Unique label for each test
- Test Case ID - Unique ID for each test
- Profile Normative References - Requirement level for the feature under test is defined in Profile Specification. This reference is informative and will not be used in conformance procedure.
- Feature Under Test - Feature which is under current test. Typically a particular command or an event.
- Test Purpose - The purpose of current test case.
- Pre-Requisite - The pre-requisite defines when the test should be performed. In case if pre-reqiusite does not match, the test result will be NOT DETECTED.
- Test Procedure - scenario expected to be reflected in network trace file.
- Test Result - Passed and failed criteria of the test case. Depending on these criteria test result will be defined as PASSED or FAILED.

4.2 Test Setup

Collect Network traces files required by the test cases.

Collect Feature List XML files for Devices detected in the Network Trace files.

Client shall support all mandatory and conditional features listed in the Device Feature List XML file supplied for the Profiles supported by the Client.

For compatibility with the Profile M, the ONVIF Client shall follow the requirements of the conformance process. For details please see the latest ONVIF Conformance Process Specification.

4.3 Prerequisites

The pre-requisites for executing the test cases described in this Test Specification include:

The Device shall be configured with an IPv4 address.

The Device shall be able to be discovered by the Client.

5 Test Cases for Profile Mandatory Features

5.1 HTTP Digest Test Cases

5.1.1 Feature Level Requirement:

Validated Feature: HTTP Digest authentication (HTTPDigest)

Check Condition based on Device Features: Digest

Required Number of Devices: 3

Profile A Requirement: Mandatory

Profile C Requirement: Mandatory

Profile D Requirement: Mandatory

Profile G Requirement: Mandatory

Profile S Requirement: Mandatory

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.1.2 Expected Scenarios Under Test:

1. Client invokes a specific command which is under testing without any user credentials (no UsernameToken, no HTTP Digest authentication header).
2. Device returns HTTP 401 Unauthorized error along with WWW-Authentication: Digest header.
3. Client re-sends request with HTTP Digest Authentication header corresponding to header provided in device response.
4. Device sends a valid response to this request.
5. Client is considered as supporting HTTP Digest if the following conditions are met:
   - Device returns a valid response to specific request with HTTP Digest authentication header.
6. Client is considered as NOT supporting HTTP Digest if the following is TRUE:

- All HTTP Digest attempts detected are failed.

5.1.3 HTTP DIGEST

Test Label: Security - HTTP Digest Authentication.

Test Case ID: HTTPDIGEST-1

Feature Under Test: HTTP Digest (HTTPDigest_HTTPDigestAuthentication)

Test Purpose: To verify that the Client supports the HTTP Digest Authentication for HTTP level security.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with HTTP Digest Authentication present.

Test Procedure (expected to be reflected in network trace file):

1. Client sends a request that requires authentication (e.g. GetUsers) to the Device without any authentication.
2. Device rejects the request with HTTP error code 401 AND an HTTP Digest challenge.
3. Client sends a valid request with HTTP Digest Authentication.
4. Device accepts the correct request with response code HTTP 200 OK.

Test Result:

PASS -

- [S1] Client request contains (HTTP GET method OR HTTP POST method) without any authentication AND
- Client HTTP GET request has a proper hierarchy (refer to [RFC 1945]) AND
  - [S2] Device response contains "HTTP/* 401 Unauthorized" AND
  - [S3] Device response contains "realm=*" element AND
  - [S4] Device response contains "nonce=*" element AND
  - [S5] Client request contains (HTTP GET method OR HTTP POST method) with "Authorization: Digest username=*" element AND

- Client HTTP GET request with HTTP Authentication has a proper hierarchy (refer to [RFC 1945]) AND
  - [S6] Client request contains "realm=*" element with value from Device response AND
  - [S7] Client request contains "nonce=*" element with value from Device response AND
  - [S8] Client request contains "uri=*" element AND
  - [S9] Device response contains "HTTP/* 200 OK".

FAIL -

- The Client failed PASS criteria.

5.2 HTTP Digest Authentication for RTSP Test Cases

5.2.1 Feature Level Requirement:

Validated Feature: HTTP Digest Authentication for RTSP (HTTPDigestForRTSP)

Check Condition based on Device Features: Profile T or Profile M

Required Number of Devices: 3

Profile S Requirement: None

Profile G Requirement: None

Profile A Requirement: None

Profile C Requirement: None

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.2.2 Expected Scenarios Under Test:

1. Client invokes a specific RTSP command which is under testing without any user credentials (no UsernameToken, no HTTP Digest authentication header).
2. IF Device returns HTTP 401 Unauthorized error along with WWW-Authentication: Digest header, then Client resends RTSP command with WWW-Authenticate header.

3. Client is considered as supporting HTTP Digest Authentication for RTSP if the following conditions are met:
   - Device returns a valid response to specific RTSP request with HTTP Digest authentication header.
4. Client is considered as NOT supporting HTTP Digest Authentication for RTSP if the following is TRUE:
   - All HTTP Digest attempts detected for RTSP are failed.

5.2.3 HTTP DIGEST AUTHENTICATION FOR RTSP

Test Label: HTTP Digest For RTSP

Test Case ID: HTTPDIGESTFORRTSP-1

Feature Under Test: HTTP Digest For RTSP (HTTPDigestForRTSP_HTTPDigestForRTSPTest)

Test Purpose: To verify that the Client supports the HTTP Digest Authentication for RTSP level security.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with HTTP Digest Authentication for RTSP commands present

Test Procedure (expected to be reflected in network trace file):

1. Client sends a RTSP request that requires authentication (e.g. DESCRIBE) to the Device without any authentication.
2. Device rejects the request with a RTSP 401 status code, AND a WWW-Authenticate Response Header.
3. Client re-sends the RTSP request with a Authorization Request Header.
4. Device accepts the correct request with RTSP 200 OK status code.

Test Result:

PASS -

- There is Client RTSP request in Test Procedure that does not contain any authentication AND
- Device response on the Client RTSP request fulfills the following requirements:

- It has RTSP 401 status code AND
- WWW-Authenticate Response Header contains challenge = "Digest" element AND
- WW-Authenticate Response Header contains "realm=*" element AND
- WW-Authenticate Response Header contains "nonce=*" element AND

- There is Client RTSP request in Test Procedure that fulfills the following requirements
  - WW-Authenticate Request Header credentials = "Digest" element AND
  - WW-Authenticate Request Header contains "realm=*" element with value from Device response AND
  - WW-Authenticate Request Header contains "nonce=*" element with value from Device response AND
  - WW-Authenticate Request Header contains "uri=*" element AND
- Device responds with code RTSP 200 OK.

FAIL -

- The Client failed PASS criteria.

5.3 Get Services Test Cases

5.3.1 Feature Level Requirement:

Validated Feature: Get Services (GetServices)

Check Condition based on Device Features: GetServices is supported by Device.

Required Number of Devices: 3

Profile A Requirement: Mandatory

Profile D Requirement: Mandatory

Profile C Requirement: Mandatory

Profile G Requirement: Mandatory

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.3.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve a services using **GetServices** commad.
2. Client is considered as supporting Get Services if the following conditions are met:
   - Client supports Capabilities_GetServicesRequest feature (please see CAPABILITIES-1 GET SERVICES section).
3. Client is considered as NOT supporting Get Services if ANY of the following is TRUE:
   - Client does not support Capabilities_GetServicesRequest feature (please see CAPABILITIES-1 GET SERVICES section).

5.3.3 GET SERVICES

Test Label: Capabilities - Determine the available Services

Test Case ID: CAPABILITIES-1

Feature Under Test: Get Services (Capabilities_GetServicesRequest)

Test Purpose: To verify that Device Capabilities is received using GetServices request.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetServices command present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetServices request message to retrieve all services of the Device.
2. Verify that GetServicesResponse message from the Device contains code HTTP 200 OK without SOAP Fault.

Test Result:

PASS -

- Client **GetServices** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetServices** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetServices>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND

- [S3] Device response contains "<GetServicesResponse>" tag.

FAIL -

- The Client failed PASS criteria.

5.4 Discovery Test Cases

5.4.1 Feature Level Requirement:

Validated Feature: Discovery (Discovery)

Check Condition based on Device Features: Discovery

Required Number of Devices: 3

Profile S Requirement: Conditional

Profile C Requirement: Conditional

Profile G Requirement: Conditional

Profile A Requirement: Mandatory

Profile T Requirement: Mandatory

Profile D Requirement: Mandatory

Profile M Requirement: Mandatory

5.4.2 Expected Scenarios Under Test:

1. Client sends Probe message to multicast IP address 239.255.255.250 and port 3702 to locate services on a local network.
2. Client is considered as supporting Discovery if the following conditions are met:
   - Probe request detected AND at least one ProbeMatch response detected
3. Client is considered as NOT supporting Discovery if the following is TRUE:
   - No Valid Device Response to Probe request.

5.4.3 WS-DISCOVERY

Test Label: Discovery - WS-Discovery

Test Case ID: DISCOVERY-1

Feature Under Test: WS-Discovery (Discovery_WSDiscovery)

Test Purpose: To verify that Client is able to send Probe request and receive ProbeMatch response from Device.

Pre-Requisite:

- The Network Trace Capture files contain at least one Client Probe request to multicast IP address and one ProbeMatch response from Device directly to the Client.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes Probe request message to multicast IP address 239.255.255.250 and port 3702.
2. Device sends ProbeMatch message directly to the Client.

Test Result:

PASS -

- Client **Probe** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **Probe** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<Action>" tag after the "<Header>" tag AND
  - [S2] "<Action>" includes URL address which ends with "Probe" value AND
  - [S3] Client request contains "<MessageID>" with non-empty string value AND
  - [S4] Client request contains "<Probe>" tag after the "<Body>" tag AND
  - [S5] Device response message contains "<ProbeMatches>" tag after the "<Body>" tag.

FAIL -

- The Client failed PASS criteria.

5.5 Device Discovery Type Filter Test Cases

5.5.1 Feature Level Requirement:

Validated Feature: Device Discovery Type Filter (DeviceDiscoveryTypeFilter)

Check Condition based on Device Features: Device Discovery Type is supported by Device.

Required Number of Devices: 3

Profile S Requirement: None

Profile A Requirement: Mandatory

Profile C Requirement: Conditional

Profile D Requirement: Mandatory

Profile G Requirement: Conditional

Profile T Requirement: Mandatory

Profile M Requirement: Mandatory

5.5.2 Expected Scenarios Under Test:

1. Client sends Probe message to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with Types filter that contains **tds:Device** or with skipped Types filter.
2. Client is considered as supporting Device Discovery Type if the following conditions are met:
   - **Probe** Client message that fulfills the following requirement is detected:
     - Types filter contains tds:Device or empty or skipped AND
     - Probe is sent to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] AND
     - Probe is sent to UDP port 3702 AND
   - There is **ProbeMatch** Device message that correspond to Client **Probe**.
3. Client is considered as NOT supporting Device Discovery Type if the following is TRUE:
   - No valid Device **ProbeMatch** message that is correspond to Client **Probe** message.

5.5.3 DEVICE DISCOVERY TYPE FILTER

Test Label: Discovery - Device Discovery Type Filter

Test Case ID: DEVICEDISCOVERYTYPEFILTER-1

Feature Under Test: Device Discovery Type Filter (DeviceDiscoveryTypeFilter_DeviceDiscoveryFilter)

Test Purpose: To verify that Client is able to discover devices with Device Discovery Type.

Pre-Requisite:

- The Network Trace Capture files contains at least one Client Probe message that does not filter out devices with Device Discovery Type that is sent to multicast WS-Discovery address.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes Probe request message to multicast IPv4 address 239.255.255.250 or multicast IPv6 address [FF02::C] and port 3702 with **Types** that contains tds:Device.
2. Device sends ProbeMatch message to the Client.

Test Result:

PASS -

- Client **Probe** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **Probe** request in Test Procedure fulfills the following requirements:
  - [S1] It is sent to 239.255.255.250 IPv4 address OR [FF02::C] IPv6 address AND
  - [S2] It is sent to 3702 UDP port AND
  - [S3] **soapenv:Envelope/soapenv:Header** element has child element **wsadis:Action** AND
  - [S4] **wsadis:Action** includes URL address which ends with "Probe" value AND
  - [S5] **soapenv:Envelope/soapenv:Header** element has child element **wsadis:MessageID** with non-empty string value AND
  - [S6] **soapenv:Body** element has child element **d:Probe** AND
  - [S7] IF **d:Probe** element has child element **d:Types** THEN it contains value is equal to **tds:Device** OR empty string value AND
  - [S8] There is Device **ProbeMatches** message in test procedure that fulfills the following requirements:
    - [S9] **soapenv:Body** element has child element **d:ProbeMatches** AND

- [S10] **soapenv:Envelope/soapenv:Header/wsadis:RelatesTo** element value is equeal to **soapenv:Envelope/soapenv:Header/wsadis:MessageID** value in **Probe** message AND

PASS WITH WARNING -

- **d:Probe/d:Types** element is skipped OR
- **d:Probe/d:Types** element has empty string value.

FAIL -

- The Client failed PASS criteria.

5.6 Metadata Streaming Using Media2 Test Cases

5.6.1 Feature Level Normative Reference:

Validated Feature: Metadata Streaming Using Media2 for Profile M (Media2_MetadataStreaming_ProfileM)

Check Condition based on Device Features: Media2 Service is supported by Device.

Required Number of Devices: 3

Profile M Requirement: Mandatory

5.6.2 Expected Scenarios Under Test:

1. Client connects to Device to receive metadata stream using Media2 Service.
2. Client is considered as supporting Metadata Streaming Using Media2 if the following conditions are met:
   - Client supports Media2_GetProfiles_Media2_GetProfilesRequest feature (please see MEDIA2_GETPROFILES-1 GET PROFILES USING MEDIA2 section) AND
   - Client supports Media2_GetStreamURI_Media2_GetStreamURIRequest feature (please see MEDIA2_GETSTREAMURI-1 GET STREAM URI USING MEDIA2 section) AND
   - Client supports Media2_MediaStreaming_Media2_UDP feature (please see MEDIA2_MEDIASTREAMING-2 STREAMING OVER UDP USING MEDIA2 section) OR Media2_MediaStreaming_Media2_HTTP feature (please see

MEDIA2_MEDIASTREAMING-3 STREAMING OVER HTTP USING MEDIA2 section) AND

- Client supports Media2_MetadataStreaming_MetadataStreamingUsingMedia2 feature (please see MEDIA2_METADATASTREAMING-1 METADATA STREAMING USING MEDIA2 section).

3. Client is considered as NOT supporting Metadata Streaming Using Media2 if the following is TRUE:

   - Client does not support Media2_GetProfiles_Media2_GetProfilesRequest feature (please see MEDIA2_GETPROFILES-1 GET PROFILES USING MEDIA2 section) OR
   - Client does not support both Media2_MediaStreaming_Media2_UDP feature (please see MEDIA2_MEDIASTREAMING-2 STREAMING OVER UDP USING MEDIA2 section) AND Media2_MediaStreaming_Media2_HTTP feature (please see MEDIA2_MEDIASTREAMING-3 STREAMING OVER HTTP USING MEDIA2 section) OR
   - Client does not support Media2_GetStreamURI_Media2_GetStreamURIRequest feature (please see MEDIA2_GETSTREAMURI-1 GET STREAM URI USING MEDIA2 section) OR
   - Client does not support Media2_MetadataStreaming_MetadataStreamingUsingMedia2 feature (please see MEDIA2_METADATASTREAMING-1 METADATA STREAMING USING MEDIA2 section) OR

5.7 Get Supported Metadata Test Cases

5.7.1 Feature Level Requirement:

Validated Feature: Get Supported Metadata (GetSupportedMetadata)

Check Condition based on Device Features: Supported Metadata is supported by Device.

Required Number of Devices: 3

Profile M Requirement: Mandatory

5.7.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve metadata SampleFrame using the **GetSupportedMetadata** operation.

2. Client is considered as supporting Get Supported Metadata if the following conditions are met:
   - Client is able to retrieve metadata SampleFrame using the **GetSupportedMetadata** operation.
3. Client is considered as NOT supporting Get Supported Metadata if ANY of the following is TRUE:
   - No valid device response to **GetSupportedMetadata** request.

5.7.3 GET SUPPORTED METADATA

Test Label: Get Supported Metadata

Test Case ID: GETSUPPORTEDMETADATA-1

Feature Under Test: Get Supported Metadata (GetSupportedMetadata_GetSupportedMetadataRequest)

Test Purpose: To verify that Client is able to retrieve metadata SampleFrame using the **GetSupportedMetadata** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetSupportedMetadata operation present.
- Device supports Supported Metadata.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetSupportedMetadata** request message to retrieve metadata SampleFrame from the Device.
2. Device responds with code HTTP 200 OK and **GetSupportedMetadataResponse** message.

Test Result:

PASS -

- Client **GetSupportedMetadata** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetSupportedMetadata** request in Test Procedure fulfills the following requirements:

- [S1] **soapenv:Body** element has child element **axt:GetSupportedMetadata** AND

- Device response on the **GetSupportedMetadata** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **axt:GetSupportedMetadataResponse**.

FAIL -

- The Client failed PASS criteria.

6 Test Cases for Profile Conditional Features

6.1 System Test Cases

6.1.1 Feature Level Requirement:

Validated Feature: System (System)

Check Condition based on Device Features: None

Required Number of Devices: 3

Profile A Requirement: Conditional

Profile C Requirement: Conditional

Profile G Requirement: Conditional

Profile S Requirement: Conditional

Profile T Requirement: Conditional

Profile D Requirement: Conditional

Profile M Requirement: Conditional

6.1.2 Expected Scenarios Under Test:

1. Client connects to Device to get information, such as manufacturer, model, firmware version and etc.
2. Client is considered as supporting System if the following conditions are met:
   - Client is able to list Device information using the GetDeviceInformation operation.
3. Client is considered as NOT supporting System if ANY of the following is TRUE:
   - No Valid Device Response to GetDeviceInformation request.

6.1.3 GET DEVICE INFORMATION

Test Label: System - Get Device Information

Test Case ID: SYSTEM-1

Feature Under Test: Get Device Information (System_GetDeviceInformation)

Test Purpose: To verify that Client is able to list Device information using the GetDeviceInformation operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetDeviceInformation operation present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetDeviceInformation request message to list Device information.
2. Device responds with code HTTP 200 OK and GetDeviceInformationResponse message.

Test Result:

PASS -

- Client **GetDeviceInformation** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetDeviceInformation** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetDeviceInformation>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetDeviceInformationResponse>" tag.

FAIL -

- The Client failed PASS criteria.

6.2 Metadata Profile Configuration Using Media2 (Profile M) Test Cases

6.2.1 Feature Level Normative Reference:

Validated Feature: Metadata Profile Configuration Using Media2 for Profile M (Media2_MetadataProfileConfiguration_ProfileM)

Check Condition based on Device Features: Metadata feature and either Video Source or Video under Media2 Service is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.2.2 Expected Scenarios Under Test:

1. Client connects to Device to configure a media profile with Video Source Configuration and with Metadata Configuration.
2. Client is considered as supporting Metadata Profile Configuration Using Media2 for Profile M if the following conditions are met:
   - Client supports **Media2_VideoSourceConfiguration_Media2_GetVideoSourceConfigurations** feature (please see MEDIA2_VIDEOSOURCECONFIGURATION-1 GET VIDEO SOURCE CONFIGURATIONS USING MEDIA2 section) AND
   - Client is able to add a Video Source Configuration using EITHER **AddConfiguration** operation with Type = VideoSource OR **CreateProfile** operation with Type = VideoSource
   - Client supports **Media2_MetadataProfileConfiguration** feature (please see Metadata Profile Configuration Using Media2 section).
3. Client is considered as NOT supporting Metadata Profile Configuration Using Media2 for Profile M if ANY of the following is TRUE:
   - Client does not support **Media2_VideoSourceConfiguration_Media2_GetVideoSourceConfigurations** feature (please see MEDIA2_VIDEOSOURCECONFIGURATION-1 GET VIDEO SOURCE CONFIGURATIONS USING MEDIA2 section) AND
   - Client is not able to add a Video Source Configuration neither using **AddConfiguration** operation nor using **CreateProfile** operation OR
   - No valid responses for **AddConfiguration** request with Type = VideoSource if deteced OR
   - No valid responses for **CreateProfile** request with Type = VideoSource if deteced OR
   - Client does not support **Media2_MetadataProfileConfiguration** feature (please see Metadata Profile Configuration Using Media2 section).

6.2.3 ADD VIDEO SOURCE CONFIGURATION USING MEDIA2

Test Label: Add Video Source Configuration

Test Case ID: MEDIA2_METADATAPROFILECONFIGURATION_PROFILEM-1

Feature Under Test: Add Video Source Configuration Using Media2 (Media2_MetadataProfileConfiguration_ProfileM_Media2_AddVideoSourceConfiguration)

Test Purpose: To verify that Client is able to add an video source configuration to a media profile using the **AddConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddConfiguration** operation with Type = **VideoSource** present.
- Device supports Media2 Video Source feature (Media2_VideoSource) OR device supports Media2 Video feature (Media2_Video).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **AddConfiguration** request message with Type = **VideoSource** and with Token element to add an video source configuration to specified media profile on the Device.
2. Device responds with code HTTP 200 OK and **AddConfigurationResponse** message.

Test Result:

PASS -

- Client **AddConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:AddConfiguration** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to "VideoSource" AND
  - [S3] It has **tr2:Configuration/tr2:Token** element AND
- Device response to the **AddConfiguration** request fulfills the following requirements:
  - [S4] It has HTTP 200 response code AND
  - [S5] **soapenv:Body** element has child element **tr2:AddConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

6.2.4 CREATE MEDIA PROFILE WITH VIDEO SOURCE CONFIGURATION USING MEDIA2

Test Label: Create Media2 Profile with Video Source Configuration

Test Case ID: MEDIA2_METADATAPROFILECONFIGURATION_PROFILEM-2

Feature Under Test: Create Media2 Profile With Video Source Configuration (Media2_MetadataProfileConfiguration_ProfileM_Media2_CreateMediaProfile)

Test Purpose: To verify that Client is able to create media profile with video source configuration using the **CreateProfile** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **CreateProfile** operation with Type = **VideoSource** present.
- Device supports Media2 Video Source feature (Media2_VideoSource) OR device supports Media2 Video feature (Media2_Video).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **CreateProfile** request message with **tr2:Configuration\tr2:Type** = **VideoSource** and with specified **tr2:Token** element for this Configuration to create profile with video source configuration on the Device.
2. Device responds with code HTTP 200 OK and **CreateProfileResponse** message.

Test Result:

PASS -

- Client **CreateProfile** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateProfile** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:CreateProfile** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to **"VideoSource"** AND
  - [S3] **tr2:Configuration** element with **tr2:Type** value is equal to **"VideoSource"** has **tr2:Token** element AND
- Device response to the **CreateProfile** request fulfills the following requirements:

- [S4] It has HTTP 200 response code AND
- [S5] **soapenv:Body** element has child element **tr2:CreateProfileResponse**.

FAIL -

- The Client failed PASS criteria.

6.3 Analytics Profile Configuration Using Media2 Test Cases

6.3.1 Feature Level Requirement:

Validated Feature: Analytics Media2 Profile Configuration (Media2_AnalyticsProfileConfiguration)

Check Condition based on Device Features: Analytics (Media2 Service) is supported by Device.

Required Number of Devices: 1

Profile T Requirement: Conditional

Profile M Requirement: Conditional

6.3.2 Expected Scenarios Under Test:

1. Client connects to Device to add compatible analytics configuration to a Media Profile.
2. Client is considered as supporting Analytics Profile Configuration if the following conditions are met:
   - Client is able to retrieve analytics configurations compatible with media profile using **GetAnalyticsConfigurations** operation with specified ProfileToken element.
   - Client is able to add an analytics configuration using **AddConfiguration** operation with Type element value is equal to **Analytics**.
3. Client is considered as NOT supporting Analytics Profile Configuration if ANY of the following is TRUE:
   - No valid responses for **GetAnalyticsConfigurations** request with ProfileToken element OR
   - No valid responses for **AddConfiguration** request with Type element value is equal to **Analytics**.

6.3.3 GET ANALYTICS CONFIGURATIONS COMPATIBLE WITH PROFILE USING MEDIA2

Test Label: Get Analytics Configurations

Test Case ID: MEDIA2_ANALYTICSPROFILECONFIGURATION-1

Feature Under Test: Get Analytics Configurations Compatible With Media2 Profile (Media2_AnalyticsProfileConfiguration_Media2_GetCompatibleAnalyticsConfigurations)

Test Purpose: To verify that list of analytics configurations compatible with a media profile is received by Client using the **GetAnalyticsConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetAnalyticsConfigurations** operation with specified **ProfileToken** element present.
- Device supports Media2 Analytics feature (Media2_Analytics).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAnalyticsConfigurations** request message with **ProfileToken** element to retrieve a list of analytics configurations compatible with requested media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAnalyticsConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetAnalyticsConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAnalyticsConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetAnalyticsConfigurations** AND
  - [S2] It has **tr2:ProfileToken** element AND
- Device response to the **GetAnalyticsConfigurations** request fulfills the following requirements:

- [S3] It has HTTP 200 response code AND
- [S4] **soapenv:Body** element has child element **tr2:GetAnalyticsConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

6.3.4 ADD ANALYTICS CONFIGURATION USING MEDIA2

Test Label: Add Analytics Configuration

Test Case ID: MEDIA2_ANALYTICSPROFILECONFIGURATION-2

Feature Under Test: Add Analytics Configuration To Media2 Profile (Media2_AnalyticsProfileConfiguration_Media2_AddAnalyticsConfiguration)

Test Purpose: To verify that Client is able to add an analytics configuration to a media profile using the **GetAnalyticsConfigurations** and **AddConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddConfiguration** operation with Type = **Analytics** present.
- Device supports Media2 Analytics feature (Media2_Analytics).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAnalyticsConfigurations** request message with specified **ProfileToken** to retrieve compatible analytics configurations for specified media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetAnalyticsConfigurationsResponse** message.
3. Client invokes **AddConfiguration** request message with Type element value is equal to **Analytics** and with Configuration token that was recieved in **GetAnalyticsConfigurationsResponse** message for the same media profile to add an analytics configuration to specified media profile on the Device.
4. Device responds with code HTTP 200 OK and **AddConfigurationResponse** message.

Test Result:

PASS -

- Client **AddConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:AddConfiguration** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to **Analytics** AND
- Device response to the **AddConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:AddConfigurationResponse**.
- There is Client **GetAnalyticsConfigurations** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked before the Client **AddConfiguration** request AND
  - [S6] It has **tr2:ProfileToken** element with value is equal to **tr2:ProfileToken** element value from the **AddConfiguration** request AND
  - [S7] It is the last **GetAnalyticsConfigurations** request which corresponds to [S5], AND [S6] AND
- Device response to the **GetAnalyticsConfigurations** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **tr2:GetAnalyticsConfigurationsResponse** AND
  - [S10] It contains **tr2:Configurations** element with **@token** attribute value equal to **tr2:Configuration/tr2:Token** value for Configuration with **tr2:Configuration/tr2:Type** value is equal to **Analytics** from the **AddConfiguration** request message.

FAIL -

- The Client failed PASS criteria.

6.4 Analytics Module Configuration Test Cases

6.4.1 Feature Level Requirement:

Validated Feature: Analytics Module Configuration (AnalyticsModuleConfiguration)

Check Condition based on Device Features: Analytics Modules under Analytics Service is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.4.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve supported Analytics Module description using the **GetSupportedAnalyticsModules** operation.
2. Client connects to Device to retrieve assigned Analytics Module Configurations using the **GetAnalyticsModules** operation.
3. Client connects to Device to add one or more Analytics Modules to an existing VideoAnalyticsConfiguration using **CreateAnalyticsModules** operation.
4. Client connects to Device to remove one or more Analytics Modules from an existing VideoAnalyticsConfiguration using **DeleteAnalyticsModules** operation.
5. Client is considered as supporting Analytics Module Configuration if the following conditions are met:
   - Client is able to retrieve supported Analytics Module description using the **GetSupportedAnalyticsModules** operation AND
   - Client is able to retrieve assigned Analytics Module Configurations using the **GetAnalyticsModules** operation AND
   - Client is able to add Analytics Modules to an existing VideoAnalyticsConfiguration using **CreateAnalyticsModules** operation AND
   - Client is able to remove Analytics Modules from an existing VideoAnalyticsConfiguration using **DeleteAnalyticsModules** operation AND
6. Client is considered as NOT supporting Analytics Module Configuration if ANY of the following is TRUE:
   - No valid device response to **GetSupportedAnalyticsModules** request OR
   - No valid device response to **GetAnalyticsModules** request OR
   - No valid device response to **CreateAnalyticsModules** request OR
   - No valid device response to **DeleteAnalyticsModules** request.

6.4.3 GET SUPPORTED ANALYTICS MODULES

Test Label: GetSupportedAnalyticsModules

Test Case ID: ANALYTICSMODULECONFIGURATION-1

Feature Under Test: GetSupportedAnalyticsModules (AnalyticsModuleConfiguration_GetSupportedAnalyticsModules)

Test Purpose: To verify that Client is able to retrieve supported Analytics Module description using the **GetSupportedAnalyticsModules** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetSupportedAnalyticsModules operation present.
- Device supports AnalyticsModules feature.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetSupportedAnalyticsModules** request message to retrieve supported Analytics Module description from the Device.
2. Device responds with code HTTP 200 OK and **GetSupportedAnalyticsModulesResponse** message.

Test Result:

PASS -

- Client **GetSupportedAnalyticsModules** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetSupportedAnalyticsModules** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tan:GetSupportedAnalyticsModules** AND
- Device response on the **GetSupportedAnalyticsModules** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tan:GetSupportedAnalyticsModulesResponse**.

FAIL -

- The Client failed PASS criteria.

6.4.4 GET ANALYTICS MODULES

Test Label: GetAnalyticsModules

Test Case ID: ANALYTICSMODULECONFIGURATION-2

Feature Under Test: GetAnalyticsModules (AnalyticsModuleConfiguration_GetAnalyticsModules)

Test Purpose: To verify that Client is able to retrieve assigned Analytics Modules using the **GetAnalyticsModules** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetAnalyticsModules operation present.
- Device supports AnalyticsModules feature.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetAnalyticsModules** request message to retrieve assigned Analytics Modules from the Device.
2. Device responds with code HTTP 200 OK and **GetAnalyticsModulesResponse** message.

Test Result:

PASS -

- Client **GetAnalyticsModules** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetAnalyticsModules** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tan:GetAnalyticsModules** AND
- Device response on the **GetAnalyticsModules** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tan:GetAnalyticsModulesResponse**.

FAIL -

- The Client failed PASS criteria.

6.4.5 CREATE ANALYTICS MODULES

Test Label: CreateAnalyticsModules

Test Case ID: ANALYTICSMODULECONFIGURATION-3

Feature Under Test: CreateAnalyticsModules (AnalyticsModuleConfiguration_CreateAnalyticsModules)

Test Purpose: To verify that Client is able to add analytics modules to an existing VideoAnalyticsConfiguration using the **CreateAnalyticsModules** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with CreateAnalyticsModules operation present.
- Device supports AnalyticsModules feature.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **CreateAnalyticsModules** request message to add analytics modules to an existing VideoAnalyticsConfiguration.
2. Device responds with code HTTP 200 OK and **CreateAnalyticsModulesResponse** message.

Test Result:

PASS -

- Client **CreateAnalyticsModules** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateAnalyticsModules** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tan:CreateAnalyticsModules** AND
- Device response on the **CreateAnalyticsModules** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tan:CreateAnalyticsModulesResponse**.

FAIL -

- The Client failed PASS criteria.

6.4.6 DELETE ANALYTICS MODULES

Test Label: DeleteAnalyticsModules

Test Case ID: ANALYTICSMODULECONFIGURATION-4

Feature Under Test: DeleteAnalyticsModules (AnalyticsModuleConfiguration_DeleteAnalyticsModules)

Test Purpose: To verify that Client is able to remove analytics modules from an existing VideoAnalyticsConfiguration using the **DeleteAnalyticsModules** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with DeleteAnalyticsModules operation present.
- Device supports AnalyticsModules feature.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **DeleteAnalyticsModules** request message to remove analytics modules from an existing VideoAnalyticsConfiguration.
2. Device responds with code HTTP 200 OK and **DeleteAnalyticsModulesResponse** message.

Test Result:

PASS -

- Client **DeleteAnalyticsModules** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DeleteAnalyticsModules** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tan:DeleteAnalyticsModules** AND
- Device response on the **DeleteAnalyticsModules** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tan:DeleteAnalyticsModulesResponse**.

FAIL -

- The Client failed PASS criteria.

6.5 Create Profile Using Media2 Test Cases

6.5.1 Feature Level Normative Reference:

Validated Feature: Create Profile Using Media2 (Media2_CreateProfile)

Check Condition based on Device Features: Media2 Service is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.5.2 Expected Scenarios Under Test:

1. Client connects to Device to retrive media profiles from the DUT.
2. Client is considered as supporting Create Profile if the following conditions are met:
   - Client is able to retrieve media profiles using **CreateProfile** operation (Media2 Service).
3. Client is considered as NOT supporting Create Profile if ANY of the following is TRUE:
   - No valid response to **CreateProfile** request (Media2 Service).

6.5.3 CREATE PROFILE USING MEDIA2

Test Label: CreateProfile

Test Case ID: MEDIA2_CREATEPROFILE-1

Feature Under Test: Create Profile Using Media2 (Media2_CreateProfile_Media2_CreateProfileRequest)

Test Purpose: To verify that media profile is created by Client using the **CreateProfile** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **CreateProfile** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **CreateProfile** request message to create a media profile on the Device.
2. Device responds with code HTTP 200 OK and **CreateProfileResponse** message.

Test Result:

PASS -

- Client **CreateProfile** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateProfile** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:CreateProfile** AND
- Device response on the **CreateProfile** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:CreateProfileResponse**.

FAIL -

- The Client failed PASS criteria.

6.6 Video Streaming Using Media2 Test Cases

6.6.1 Feature Level Normative Reference:

Validated Feature: Video Streaming Using Media2 for Profile M (Media2_VideoStreaming_ProfileM)

Check Condition based on Device Features: H264 or H265 under Media2 Service is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.6.2 Expected Scenarios Under Test:

1. Client connects to Device to receive video stream using Media2 Service.
2. Client is considered as supporting Video Streaming Using Media2 (Profile M) if the following conditions are met:
   - Client supports Media2_VideoStreaming_H264_Media2_H264VideoStreaming feature (please see MEDIA2_VIDEOSTREAMING_H264-1 H264 VIDEO STREAMING USING MEDIA2 section) AND

- Client supports Media2_VideoStreaming_H265_Media2_H265VideoStreaming feature (please see MEDIA2_VIDEOSTREAMING_H265-1 H265 VIDEO STREAMING USING MEDIA2 section).

3. Client is considered as NOT supporting Video Streaming Using Media2 (Profile M) if the following is TRUE:

   - Client does NOT support Media2_VideoStreaming_H264_Media2_H264VideoStreaming feature (please see MEDIA2_VIDEOSTREAMING_H264-1 H264 VIDEO STREAMING USING MEDIA2 section) AND

   - Client does NOT support Media2_VideoStreaming_H265_Media2_H265VideoStreaming feature (please see MEDIA2_VIDEOSTREAMING_H265-1 H265 VIDEO STREAMING USING MEDIA2 section).

6.7 Image Data Test Cases

6.7.1 Feature Level Requirement:

Validated Feature: Image Data (ImageData)

Check Condition based on Device Features: Embedded Image Sending Type.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.7.2 Expected Scenarios Under Test:

1. Client connects to Profile M simulator to retrieve metadata with image URI sending in ImageRef element. Client retrieves image using HTTP GET sent to the image URI.

2. Client connects to Device to receive base64 encoding image data included in the Appearance Node on metadata streaming.

3. Client is considered as supporting Image Data if the following conditions are met:

   - ONVIF Profile M Simulator detects **ImageData_ImageURI** feature as supported AND

   - Client is able to retrieve base64 encoding image data included in the Appearance Node on metadata streaming.

4. Client is considered as NOT supporting Image Data if ANY of the following is TRUE:

- ONVIF Profile M Simulator does NOT detect **ImageData_ImageURI** feature as supported OR
- Client is NOT able to retrieve base64 encoding image data included in the Appearance Node on metadata streaming.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports receiving of base64 encoding image data.

6.7.3 METADATA IMAGE URI

Test Label: Get image via image URI

Test Case ID: IMAGEDATA-1

Feature Under Test: Get image via image URI (ImageData_ImageURI)

Test Purpose: To verify that a Client is able to get metadata image via the image URI.

Pre-Requisite:

- The .osc file of conversation with ONVIF Profile M Simulator is present (please, refer to 'Profile M Simulator' section in Help) .

Test Procedure (expected to be reflected in network trace file):

1. Client configures an media2 profile on the ONVIF Profile M simulator for metadata streaming with alanytics. Client starts metadata streaming.
2. ONVIF Profile M simulator sends metadata streaming with Image URI.
3. Client invokes **HTTP GET** request to Image Uri.
4. ONVIF Profile M simulator responds with code HTTP 200 OK and sends image.

Test Result:

PASS -

- Client **HTTP GET** request in Test Procedure that fulfills the following requirements:
  - [S1] It invoked to address which equal to **tt:MetadataStream/tt:VideoAnalytics/tt:Frame/tt:Object/tt:Appearance/tt:ImageRef** value from the ONVIF Profile M metadata streaming. AND
- ONVIF Profile M Simulator response on the **HTTP GET** request fulfills the following requirements:

- [S2] It has **HTTP 200** response code.

FAIL -

- The Client failed PASS criteria.

6.8 HTTPS Streaming Using Media2 (Profile M) Test Cases

6.8.1 Feature Level Requirement:

Validated Feature: HTTPS Streaming Using Media2 (Profile M) (Media2_HTTPSStreaming_ProfileM)

Check Condition based on Device Features: No (ONVIF Profile M Simulator is used as device).

Required Number of Devices: 1

Profile M Requirement: Conditional

6.8.2 Expected Scenarios Under Test:

1. Client connects to ONVIF Profile M Simulator to initiate HTTPS metadata Streaming.
2. Client is considered as supporting HTTPS Streaming if the following conditions are met:
   - ONVIF Profile M Simulator detects **Streaming over RTP/RTSP/HTTPS/TCP** feature as supported.
3. Client is considered as NOT supporting HTTPS Streaming if the following is TRUE:
   - ONVIF Profile M Simulator detects **Streaming over RTP/RTSP/HTTPS/TCP** feature as not supported.

6.9 Event Handling Using Pull Points Test Cases

6.9.1 Feature Level Requirement:

Validated Feature: Event Handling Using Pull Points (EventHandlingUsingPullPoints)

Check Condition based on Device Features: Pull Point Notification is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.9.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieving of events from a device using ONVIF realtime pullpoints.
2. Client is considered as supporting Event Handling Using Pull Points if the following conditions are met:
   - Client supports EventHandling_PullPoint feature (please see EVENTHANDLING-1 PULLPOINT section) AND
   - Client supports SetSynchronizationPoint_SetSynchronizationPointAction feature (please see SETSYNCHRONIZATIONPOINT-1 SET SYNCHRONIZATION POINT (EVENT SERVICE) section).
3. Client is considered as NOT supporting Event Handling Using Pull Points if the following conditions are met:
   - Client does not support EventHandling_PullPoint feature (please see EVENTHANDLING-1 PULLPOINT section) OR
   - Client does not support SetSynchronizationPoint_SetSynchronizationPointAction feature (please see SETSYNCHRONIZATIONPOINT-1 SET SYNCHRONIZATION POINT (EVENT SERVICE) section).

6.10 Rule Configuration Test Cases

6.10.1 Feature Level Requirement:

Validated Feature: Rule Configuration (RuleConfiguration)

Check Condition based on Device Features: Media 2 Service and Rule Engine is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.10.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve available Rules using the **GetRules** operation.

2. Client connects to Device to retrieve supported rule description using the **GetSupportedRules** operation.
3. Client connects to Device to add one or more rules to an existing VideoAnalyticsConfiguration using the **CreateRules** operation.
4. Client connects to Device to remove one or more rules from a VideoAnalyticsConfiguration using the **DeleteRules** operation.
5. Client is considered as supporting Rule Configuration if the following conditions are met:
   - Client supports GetRules_GetRulesRequest feature (please see GETRULES-1 GET RULES section) AND
   - Client supports GetSupportedRules_GetSupportedRulesRequest feature (please see GETSUPPORTEDRULES-1 GET SUPPORTED RULES section) AND
   - Client is able to add one or more rules to an existing VideoAnalyticsConfiguration using **CreateRules** operation AND
   - Client is able to remove one or more rules from an existing VideoAnalyticsConfiguration using **DeleteRules** operation.
6. Client is considered as NOT supporting Rule Configuration if ANY of the following is TRUE:
   - Client does not support GetRules_GetRulesRequest feature (please see GETRULES-1 GET RULES section) OR
   - Client does not support GetSupportedRules_GetSupportedRulesRequest feature (please see GETSUPPORTEDRULES-1 GET SUPPORTED RULES section) OR
   - No valid device response to **CreateRules** request OR
   - No valid device response to **DeleteRules** request.

6.10.3 CREATE RULES

Test Label: Create Rules

Test Case ID: RULECONFIGURATION-1

Feature Under Test: Create Rules (RuleConfiguration_CreateRulesRequest)

Test Purpose: To verify that Client is able to create rules using the **CreateRules** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with CreateRules operation present.
- Device supports Rule Engine (RuleEngine).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **CreateRules** request message to add Rules to an existing VideoAnalyticsConfiguration on the Device.
2. Device responds with code HTTP 200 OK and **CreateRulesResponse** message.

Test Result:

PASS -

- Client **CreateRules** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreateRules** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **axt:CreateRules** AND
- Device response on the **CreateRules** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **axt:CreateRulesResponse**.

FAIL -

- The Client failed PASS criteria.

6.10.4 DELETE RULES

Test Label: Delete Rules

Test Case ID: RULECONFIGURATION-2

Feature Under Test: Delete Rules (RuleConfiguration_DeleteRulesRequest)

Test Purpose: To verify that Client is able to delete rules using the **DeleteRules** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with DeleteRules operation present.

- Device supports Rule Engine (RuleEngine).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **DeleteRules** request message to remove Rules from an existing VideoAnalyticsConfiguration on the Device.
2. Device responds with code HTTP 200 OK and **DeleteRulesResponse** message.

Test Result:

PASS -

- Client **DeleteRules** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DeleteRules** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **axt:DeleteRules** AND
- Device response on the **DeleteRules** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **axt:DeleteRulesResponse**.

FAIL -

- The Client failed PASS criteria.

6.11 Object Classification Test Cases

6.11.1 Feature Level Requirement:

Validated Feature: Object Classification (ObjectClassification)

Check Condition based on Device Features: Object Classification is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.11.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve metadata streaming with included Class element **MetadataStream/VideoAnalyticsStream/ Frame/Object/Appearance /Class/Type**.

2. Client is considered as supporting Object Classification if the following conditions are met:
   - Client is able to retrieve metadata streaming with included Class element **MetadataStream/VideoAnalyticsStream/ Frame/Object/Appearance /Class/Type**.
3. Client is considered as NOT supporting Object Classification if ANY of the following is TRUE:
   - Client is not able to retrieve metadata streaming with included Class element **MetadataStream/VideoAnalyticsStream/ Frame/Object/Appearance /Class/Type**.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports this feature.

6.12 Human Face Metadata Test Cases

6.12.1 Feature Level Requirement:

Validated Feature: Human Face Metadata (HumanFaceMetadata)

Check Condition based on Device Features: Human Face is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.12.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/HumanFace** element.
2. Client is considered as supporting Human Face Metadata if the following conditions are met:
   - Client is able to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/HumanFace** element.
3. Client is considered as NOT supporting Human Face Metadata if ANY of the following is TRUE:
   - Client is NOT able to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/HumanFace** element.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports this feature.

6.13 Human Body Metadata Test Cases

6.13.1 Feature Level Requirement:

Validated Feature: Human Body Metadata (HumanBodyMetadata)

Check Condition based on Device Features: Human Body is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.13.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/HumanBody** element.
2. Client is considered as supporting Human Body Metadata if the following conditions are met:
   - Client is able to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/HumanBody** element.
3. Client is considered as NOT supporting Human Body Metadata if ANY of the following is TRUE:
   - Client is NOT able to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/HumanBody** element.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports this feature.

6.14 Vehicle Metadata Test Cases

6.14.1 Feature Level Requirement:

Validated Feature: Vehicle Metadata (VehicleMetadata)

Check Condition based on Device Features: Vehicle Info is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.14.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/VehicleInfo** element.
2. Client is considered as supporting Vehicle Metadata if the following conditions are met:
   - Client is able to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/VehicleInfo** element.
3. Client is considered as NOT supporting Vehicle Metadata if ANY of the following is TRUE:
   - Client is NOT able to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/VehicleInfo** element.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports this feature.

6.15 License Plate Metadata Test Cases

6.15.1 Feature Level Requirement:

Validated Feature: License Plate Metadata (LicensePlateMetadata)

Check Condition based on Device Features: License Plate Info is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.15.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/LicensePlateInfo** element.
2. Client is considered as supporting License Plate Metadata if the following conditions are met:
   - Client is able to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/LicensePlateInfo** element.
3. Client is considered as NOT supporting License Plate Metadata if ANY of the following is TRUE:

- Client is NOT able to retrieve metadata streaming with included **MetadataStream / VideoAnalyticsStream/Frame/Object/ Appearance/LicensePlateInfo** element.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports this feature.

6.16 GeoLocation Metadata Test Cases

6.16.1 Feature Level Requirement:

Validated Feature: GeoLocation Metadata (GeoLocationMetadata)

Check Condition based on Device Features: Geo Location is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.16.2 Expected Scenarios Under Test:

1. Client connects to Device to retrieve metadata streaming with included **MetadataStream/ VideoAnalyticsStream/Frame/Object/ Appearance/GeoLocation** element.
2. Client is considered as supporting GeoLocation Metadata if the following conditions are met:
   - Client is able to retrieve metadata streaming with included **MetadataStream/ VideoAnalyticsStream/Frame/Object/ Appearance/GeoLocation** element.
3. Client is considered as NOT supporting GeoLocation Metadata if ANY of the following is TRUE:
   - Client is NOT able to retrieve metadata streaming with included **MetadataStream/ VideoAnalyticsStream/Frame/Object/ Appearance/GeoLocation** element.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports this feature.

6.17 Face Recognition Event Test Cases

6.17.1 Feature Level Requirement:

Validated Feature: Face Recognition Event (FaceRecognitionEvent)

Check Condition based on Device Features: Face Recognition Rule is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.17.2 Expected Scenarios Under Test:

1. Client gets notifications about the Face Recognition EITHER subscribing to device messages using **CreatePullPointSubscription** operation OR subscribing to **MQTT Broker** server OR initiating of **metadata streaming**.

2. Client is considered as supporting Face Recognition Event if the following conditions are met:

   - Client supports Media2_MetadataStreaming_ProfileM feature (please see Metadata Streaming Using Media2 Test Cases section) OR EventHandling_Pullpoint feature (please see EVENTHANDLING-1 PULLPOINT section) OR EventBrokerConfiguration feature (please see Event Broker Configuration Test Cases section)
     AND

   - Client is able to receive **tns1:RuleEngine/Recognition/Face** notification.

3. Client is considered as NOT supporting Face Recognition Event if the following conditions are met:

   - Client does not support Media2_MetadataStreaming_ProfileM feature (please see Metadata Streaming Using Media2 Test Cases section) AND EventHandling_Pullpoint feature (please see EVENTHANDLING-1 PULLPOINT section) AND EventBrokerConfiguration feature (please see Event Broker Configuration Test Cases section) OR

   - Client is NOT able to receive **tns1:RuleEngine/Recognition/Face** notification.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports getting of this notification.

6.18 License Plate Recognition Event Test Cases

6.18.1 Feature Level Requirement:

Validated Feature: License Plate Recognition Event (LicensePlateRecognitionEvent)

Check Condition based on Device Features: License Plate Recognition Rule is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.18.2 Expected Scenarios Under Test:

1. Client gets notifications about the License Plate Recognition EITHER subscribing to device messages using **CreatePullPointSubscription** operation OR subscribing to **MQTT Broker** server OR initiating of **metadata streaming**.
2. Client is considered as supporting License Plate Recognition Event if the following conditions are met:
   - Client supports Media2_MetadataStreaming_ProfileM feature (please see Metadata Streaming Using Media2 Test Cases section) OR EventHandling_Pullpoint feature (please see EVENTHANDLING-1 PULLPOINT section) OR EventBrokerConfiguration feature (please see Event Broker Configuration Test Cases section) AND
   - Client is able to receive **tns1:RuleEngine/Recognition/LicensePlate** notification.
3. Client is considered as NOT supporting License Plate Recognition Event if the following conditions are met:
   - Client does not support Media2_MetadataStreaming_ProfileM feature (please see Metadata Streaming Using Media2 Test Cases section) AND EventHandling_Pullpoint feature (please see EVENTHANDLING-1 PULLPOINT section) AND EventBrokerConfiguration feature (please see Event Broker Configuration Test Cases section) AND
   - Client is NOT able to receive **tns1:RuleEngine/Recognition/LicensePlate** notification.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports getting of this notification.

6.19 Line Crossing Counter Event Test Cases

6.19.1 Feature Level Requirement:

Validated Feature: Line Crossing Counter Event (LineCrossingCounterEvent)

Check Condition based on Device Features: Line Crossing Counting Rule is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.19.2 Expected Scenarios Under Test:

1. Client gets notifications about the Line Crossing Counter EITHER subscribing to device messages using **CreatePullPointSubscription** operation OR subscribing to **MQTT Broker** server OR initiating of **metadata streaming**.

2. Client is considered as supporting Line Crossing Counter Event if the following conditions are met:

   - Client supports Media2_MetadataStreaming_ProfileM feature (please see Metadata Streaming Using Media2 Test Cases section) OR EventHandling_Pullpoint feature (please see EVENTHANDLING-1 PULLPOINT section) OR EventBrokerConfiguration feature (please see Event Broker Configuration Test Cases section) AND

   - Client is able to receive **tns1:RuleEngine/CountAggregation/Counter** notification.

3. Client is considered as NOT supporting Line Crossing Counter Event if the following conditions are met:

   - Client does not support Media2_MetadataStreaming_ProfileM feature (please see Metadata Streaming Using Media2 Test Cases section) AND EventHandling_Pullpoint feature (please see EVENTHANDLING-1 PULLPOINT section) AND EventBrokerConfiguration feature (please see Event Broker Configuration Test Cases section) OR

   - Client is NOT able to receive **tns1:RuleEngine/CountAggregation/Counter** notification.

NOTE: Test Operator shall mark corresponding checkbox on the Onvif Client Test Tool UI in case a client supports getting of this notification.

6.20 Event Broker Configuration Test Cases

6.20.1 Feature Level Requirement:

Validated Feature: Event Broker Configuration (EventBrokerConfiguration)

Check Condition based on Device Features: Event Broker is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Conditional

6.20.2 Expected Scenarios Under Test:

1. Client connects to Device to get event broker configurations, add new event broker configuration, delete event broker configuration.
2. Client is considered as supporting Event Broker Configuration if the following conditions are met:
   - Client is able to retrieve event broker configurations using **GetEventBrokers** operation AND
   - Client is able to add event broker configuration using **AddEventBroker** operation AND
   - Client is able to remove event broker configuration using **DeleteEventBroker** operation.
3. Client is considered as NOT supporting Event Broker Configuration if the following conditions are met:
   - No valid response to **GetEventBrokers** request OR
   - No valid response to **AddEventBroker** request OR
   - No valid response to **DeleteEventBroker** request.

6.20.3 GET EVENT BROKERS

Test Label: GetEventBrokers

Test Case ID: EVENTBROKERCONFIGURATION-1

Feature Under Test: GetEventBrokers (EventBrokerConfiguration_GetEventBrokers)

Test Purpose: To verify that event broker configuration provided by Device are received by Client using the **GetEventBrokers** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetEventBrokers** operation for Event Service present.
- Device supports MQTT feature.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetEventBrokers** request message to retrieve a event broker configuration or a list of event broker configurations from the Device.

2. Device responds with code HTTP 200 OK and **GetEventBrokersResponse** message.

Test Result:

PASS -

- Client **GetEventBrokers** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetEventBrokers** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tev:GetEventBrokers** AND
- Device response on the **GetEventBrokers** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tev:GetEventBrokersResponse**.

FAIL -

- The Client failed PASS criteria.

6.20.4 ADD EVENT BROKER

Test Label: AddEventBroker

Test Case ID: EVENTBROKERCONFIGURATION-2

Feature Under Test: AddEventBroker (EventBrokerConfiguration_AddEventBroker)

Test Purpose: To verify that Client adds event broker configuration on Device using the **AddEventBroker** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddEventBroker** operation for Event Service present.
- Device supports MQTT feature.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **AddEventBroker** request message to add an event broker configuration on the Device.
2. Device responds with code HTTP 200 OK and **AddEventBrokerResponse** message.

Test Result:

PASS -

- Client **AddEventBroker** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddEventBroker** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tev:AddEventBroker** AND
- Device response on the **AddEventBroker** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tev:AddEventBrokerResponse**.

FAIL -

- The Client failed PASS criteria.

6.20.5 DELETE EVENT BROKER

Test Label: DeleteEventBroker

Test Case ID: EVENTBROKERCONFIGURATION-3

Feature Under Test: DeleteEventBroker (EventBrokerConfiguration_DeleteEventBroker)

Test Purpose: To verify that Client removes event broker configuration from Device using the **DeleteEventBroker** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **DeleteEventBroker** operation for Event Service present.
- Device supports MQTT feature.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **DeleteEventBroker** request message to remove an event broker configuration from the Device.
2. Device responds with code HTTP 200 OK and **DeleteEventBrokerResponse** message.

Test Result:

PASS -

- Client **DeleteEventBroker** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **DeleteEventBroker** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tev:DeleteEventBroker** AND
- Device response on the **DeleteEventBroker** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tev:DeleteEventBrokerResponse**.

FAIL -

- The Client failed PASS criteria.

7 Test Cases for Profile Optional Features

7.1 Metadata Configuration Using Media2 Test Cases

7.1.1 Feature Level Normative Reference:

Validated Feature: Metadata Configuration Using Media2 (Media2_MetadataConfiguration_ProfileM)

Check Condition based on Device Features: Media2 Service is supported by Device.

Required Number of Devices: 1

Profile M Requirement: Optional

7.1.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Metadata Configuration using Media 2 Service.
2. Client is considered as supporting Metadata Configuration Using Media2 if the following conditions are met:
   - Client supports Media2_MetadataConfiguration feature (please see Metadata Configuration Using Media2 Test Cases section).
3. Client is considered as NOT supporting Metadata Configuration Using Media2 if ANY of the following is TRUE:
   - Client does not support Media2_MetadataConfiguration feature (please see Metadata Configuration Using Media2 Test Cases section).

8 Supplementary Features and Test Cases

8.1 Metadata Configuration Using Media2 Test Cases

8.1.1 Feature Level Normative Reference:

Validated Feature: Metadata Configuration Using Media2 (Media2_MetadataConfiguration)

Check Condition based on Device Features: Media2 Service is supported by Device.

Required Number of Devices: 1

Profile T Requirement: Conditional

Profile M Requirement: Mandatory

8.1.2 Expected Scenarios Under Test:

1. Client connects to Device to modify Metadata Configuration.
2. Client is considered as supporting Metadata Configuration if the following conditions are met:
   - Client is able to retrieve metadata configuratoins using **GetMetadataConfigurations** operation (Media2 Service) AND
   - Client is able to retrieve metadata configuratoin options using **GetMetadataConfigurationOptions** operation (Media2 Service) AND
   - Client is able to modify metadata configuratoin using **SetMetadataConfiguration** operation (Media2 Service) AND
3. Client is considered as NOT supporting Metadata Configuration if ANY of the following is TRUE:
   - No valid response to **GetMetadataConfigurations** request (Media2 Service) OR
   - No valid response to **GetMetadataConfigurationOptions** request (Media2 Service) OR
   - No valid response to **SetMetadataConfiguration** request (Media2 Service) OR

8.1.3 GET METADATA CONFIGURATIONS USING MEDIA2

Test Label: Metadata Configuration - Get Metadata Configurations

Test Case ID: MEDIA2_METADATACONFIGURATION-1

Feature Under Test: Get Metadata Configurations Using Media2 (Media2_MetadataConfiguration_Media2_GetMetadataConfigurations)

Test Purpose: To verify that metadata configuration provided by Device is received by Client using the **GetMetadataConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetMetadataConfigurations** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetMetadataConfigurations** request message to retrieve an metadata configuration or a list of metadata configurations from the Device.
2. Device responds with code HTTP 200 OK and **GetMetadataConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetMetadataConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetMetadataConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetMetadataConfigurations** AND
- Device response on the **GetMetadataConfigurations** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetMetadataConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

8.1.4 GET METADATA CONFIGURATION OPTIONS USING MEDIA2

Test Label: Metadata Configuration - Get Metadata Configuration Options

Test Case ID: MEDIA2_METADATACONFIGURATION-2

Feature Under Test: Get Metadata Configuration Options Using Media2 (Media2_MetadataConfiguration_Media2_GetMetadataConfigurationOptions)

Test Purpose: To verify that metadata configuration options provided by Device is received by Client using the **GetMetadataConfigurationOptions** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetMetadataConfigurationOptions** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetMetadataConfigurationOptions** request message to retrieve an metadata configuration options from the Device.
2. Device responds with code HTTP 200 OK and **GetMetadataConfigurationOptionsResponse** message.

Test Result:

PASS -

- Client **GetMetadataConfigurationOptions** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetMetadataConfigurationOptions** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetMetadataConfigurationOptions** AND
- Device response on the **GetMetadataConfigurationOptions** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND

- [S3] **soapenv:Body** element has child element **tr2:GetMetadataConfigurationOptionsResponse**.

FAIL -

- The Client failed PASS criteria.

8.1.5 SET METADATA CONFIGURATION USING MEDIA2

Test Label: Metadata Configuration - Set Metadata Configuration

Test Case ID: MEDIA2_METADATACONFIGURATION-3

Feature Under Test: Set Metadata Configuration Using Media2 (Media2_MetadataConfiguration_Media2_SetMetadataConfiguration)

Test Purpose: To verify that Client is able to change metadata configuration provided by Device using the **SetMetadataConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetMetadataConfiguration** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **SetMetadataConfiguration** request message to change an metadata configuration on the Device.
2. Device responds with code HTTP 200 OK and **SetMetadataConfigurationResponse** message.

Test Result:

PASS -

- Client **SetMetadataConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetMetadataConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:SetMetadataConfiguration** AND

- Device response on the **SetMetadataConfiguration** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:SetMetadataConfigurationResponse**.

FAIL -

- The Client failed PASS criteria.

8.2 GET SERVICES

Test Label: Capabilities - Determine the available Services

Test Case ID: CAPABILITIES-1

Feature Under Test: Get Services (Capabilities_GetServicesRequest)

Test Purpose: To verify that Device Capabilities is received using GetServices request.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with GetServices command present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes GetServices request message to retrieve all services of the Device.
2. Verify that GetServicesResponse message from the Device contains code HTTP 200 OK without SOAP Fault.

Test Result:

PASS -

- Client **GetServices** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetServices** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<GetServices>" tag after the "<Body>" tag AND
  - [S2] Device response contains "HTTP/* 200 OK" AND
  - [S3] Device response contains "<GetServicesResponse>" tag.

FAIL -

- The Client failed PASS criteria.

8.3 METADATA STREAMING USING MEDIA2

Test Label: Metadata Streaming Using Media2

Test Case ID: MEDIA2_METADATASTREAMING-1

Feature Under Test: Metadata Streaming (Media2_MetadataStreaming_MetadataStreamingUsingMedia2)

Test Purpose: To verify that the Client is able to retrieve the Metadata Streaming.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Metadata Streaming using Media2 Service.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for Media2 service for media profile that contains Metadata Configuration. GetStreamUri request is set for RtspUnicast OR RtspMulticast OR RTSP OR RtspOverHttp transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "application" and with encoding name "vnd.onvif.metadata" or "vnd.onvif.metadata.gzip" or "vnd.onvif.metadata.exi.onvif" or "vnd.onvif.metadata.exi.ext".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for metadata streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RtspOverHttp transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "application" (m=application) with sessions attribute "rtpmap" with encoding name "vnd.onvif.metadata" OR "vnd.onvif.metadata.gzip" OR "vnd.onvif.metadata.exi.onvif" OR "vnd.onvif.metadata.exi.ext" (see ONVIF Streaming Spec) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request invoked for Media2 Service in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **tr2:GetStreamUriResponse\tr2:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S12] It invoked for the same Device as for the Client **RTSP SETUP** request AND

- [S13] It invoked after the Client **RTSP SETUP** request AND
- [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

8.4 STREAMING OVER UDP USING MEDIA2

Test Label: Media Streaming - UDP

Test Case ID: MEDIA2_MEDIASTREAMING-2

Feature Under Test: Streaming Over UDP Using Media2 (Media2_MediaStreaming_Media2_UDP)

Test Purpose: To verify that stream over UDP protocol was successfully established between Client and Device using RTSP commands and then successfully stopped.

Pre-Requisite:

- Device supports Media2 Real Time Streaming (Media2_RealTimeStreaming).
- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/UDP" or "RTP/AVP" and which does not contain Require header with "onvif-replay" value present.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service with **rt2:Protocol** element value equals to "RtspUnicast" or "RtspMulticast".

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile with Transport element with "RtspUnicast" value or "RtspMulticast" value.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request with **Transport** tag in RTSP header that contains "RTP/AVP/UDP" or "RTP/AVP" and without "onvif-replay" Require header to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/UDP" OR "RTP/AVP" (transport=RTP, profile=AVP, lower-transport=TCP or skipped) (see [RFC 2326]) AND
  - [S2] It does not contain **Require** request header field with value is equal to "onvif-replay" AND

- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S3] It has RTSP 200 response code AND
- There is Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:
  - [S4] It invoked for the same Device as for the Client **RTSP SETUP** request AND
  - [S5] It invoked before the Client **RTSP SETUP** request AND
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S6] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND
  - [S7] It has RTSP 200 response code AND
- There is a Device **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S8] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S9] It invoked before the Client **RTSP DESCRIBE** request AND
  - [S10] **tr2:GetStreamUri/tr2:Protocol** element value is equal to "RtspUnicast" or "RtspMulticast"
- Device response on the **GetStreamUri** request to Media2 Service fulfills the following requirements:
  - [S11] It contains **tr2:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
  - [S12] It has HTTP 200 response code AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S13] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S14] It invoked after the Client **RTSP SETUP** request AND
  - [S15] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S16] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:

- [S17] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S18] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S19] It invoked after the Client **RTSP PLAY** request AND
  - [S20] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S21] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

8.5 STREAMING OVER HTTP USING MEDIA2

Test Label: Media Streaming - HTTP

Test Case ID: MEDIA2_MEDIASTREAMING-3

Feature Under Test: Streaming Over HTTP Using Media2 (Media2_MediaStreaming_Media2_HTTP)

Test Purpose: To verify that stream over HTTP protocol was successfully established between Client and Device using RTSP commands and then successfully stopped.

Pre-Requisite:

- Device supports HTTP streaming for Media2 Service (Media2_RTPRTSPHTTP).
- The Network Trace Capture files contains at least one Conversation between Client and Device with RTSP SETUP request with transport parameter as "RTP/AVP/TCP" and which does not contain Require header with "onvif-replay" value and which is tunneled in HTTP present.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service with **rt2:Protocol** element value equals to RtspOverHttp.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile with Protocol element with "RtspOverHttp" value.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request in HTTP tunnel to retrieve media stream description.
4. Device responds with code RTSP 200 OK.
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header in HTTP tunnel with **Transport** tag in RTSP header that contains "RTP/AVP/TCP" to set media session parameters.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header in HTTP tunnel to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request in HTTP tunnel to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

PASS -

- Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S1] It contains **Transport** request header field with value is equal to "RTP/AVP/TCP" (transport=RTP, profile=AVP, lower-transport=TCP) (see [RFC 2326]) AND
  - [S2] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
  - [S3] It is tunneled in HTTP AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S4] It has RTSP 200 response code AND
- There is Client **RTSP DESCRIBE** request in Test Procedure fulfills the following requirements:

- [S5] It invoked for the same Device as for the Client **RTSP SETUP** request AND
- [S6] It invoked before the Client **RTSP SETUP** request AND
- [S7] It is tunneled in HTTP AND

- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S8] SDP packet contains media type with Control URL that was used to send **RTSP SETUP** (see [RFC 2326, C.1.1 Control URL]) AND
  - [S9] It has RTSP 200 response code AND
- There is a Device **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S10] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S11] It invoked before the Client **RTSP DESCRIBE** request AND
  - [S12] **tr2:GetStreamUri/tr2:Protocol** element value is equal to "RtspOverHttp"
- Device response on the **GetStreamUri** request to Media2 Service fulfills the following requirements:
  - [S13] It has HTTP 200 response code AND
  - [S14] It contains **tr2:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S15] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S16] It invoked after the Client **RTSP SETUP** request AND
  - [S17] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S18] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
  - [S19] It is tunneled in HTTP AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S20] It has RTSP 200 response code AND

- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S21] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S22] It invoked after the Client **RTSP PLAY** request AND
  - [S23] RTSP address that was used to send it is correspond to any media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S24] It is tunneled in HTTP AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S25] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

8.6 GET PROFILES USING MEDIA2

Test Label: GetProfiles

Test Case ID: MEDIA2_GETPROFILES-1

Feature Under Test: Get Profiles Using Media2 (Media2_GetProfiles_Media2_GetProfilesRequest)

Test Purpose: To verify that media profiles provided by Device are received by Client using the **GetProfiles** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetProfiles** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetProfiles** request message to retrieve a media profile or a list of media profiles from the Device.
2. Device responds with code HTTP 200 OK and **GetProfilesResponse** message.

Test Result:

PASS -

- Client **GetProfiles** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetProfiles** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetProfiles** AND
- Device response on the **GetProfiles** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetProfilesResponse**.

FAIL -

- The Client failed PASS criteria.

8.7 GET STREAM URI USING MEDIA2

Test Label: GetStreamUri

Test Case ID: MEDIA2_GETSTREAMURI-1

Feature Under Test: Get Stream URI Using Media2 (Media2_GetStreamURI_Media2_GetStreamURIRequest)

Test Purpose: To verify that stream URI provided by Device is received by Client using the **GetStreamUri** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message to retrieve a stream URI from the Device.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.

Test Result:

PASS -

- Client **GetStreamUri** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetStreamUri** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetStreamUri** AND
- Device response on the **GetStreamUri** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **tr2:GetStreamUriResponse**.

FAIL -

- The Client failed PASS criteria.

8.8 Metadata Profile Configuration Using Media2 Test Cases

8.8.1 Feature Level Requirement:

Validated Feature: Metadata Media2 Profile Configuration (Media2_MetadataProfileConfiguration)

Check Condition based on Device Features: Media2 Service is supported by Device.

Required Number of Devices: 1

Profile T Requirement: Conditional

Profile M Requirement: Conditional

8.8.2 Expected Scenarios Under Test:

1. Client connects to Device to add compatible Metadata configuration to a Media Profile.
2. Client is considered as supporting Metadata Profile Configuration if the following conditions are met:
   - Client is able to retrieve Metadata configurations compatible with media profile using **GetMetadataConfigurations** operation with specified ProfileToken element.
   - Client is able to add an Metadata configuration using **AddConfiguration** operation with Type element value is equal to Metadata.
3. Client is considered as NOT supporting Metadata Profile Configuration if ANY of the following is TRUE:

- No valid responses for **GetMetadataConfigurations** request with ProfileToken element OR
- No valid responses for **AddConfiguration** request with Type element value is equal to Metadata is detected.

8.8.3 GET METADATA CONFIGURATIONS COMPATIBLE WITH PROFILE USING MEDIA2

Test Label: Get Metadata Configurations

Test Case ID: MEDIA2_METADATAPROFILECONFIGURATION-1

Feature Under Test: Get Metadata Configurations Compatible With Media2 Profile (Media2_MetadataProfileConfiguration_Media2_GetCompatibleMetadataProfileConfigurations)

Test Purpose: To verify that list of metadata configurations compatible with a media profile is received by Client using the **GetMetadataConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetMetadataConfigurations** operation with specified **ProfileToken** element present.
- Device supports Media2 Service (Media2Service).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetMetadataConfigurations** request message with **ProfileToken** element to retrieve a list of metadata configurations compatible with requested media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetMetadataConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetMetadataConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetMetadataConfigurations** request in Test Procedure fulfills the following requirements:

- [S1] **soapenv:Body** element has child element **tr2:GetMetadataConfigurations** AND
- [S2] It has **tr2:ProfileToken** element AND

- Device response to the **GetMetadataConfigurations** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:GetMetadataConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

8.8.4 ADD METADATA CONFIGURATION USING MEDIA2

Test Label: Add Metadata Configuration

Test Case ID: MEDIA2_METADATAPROFILECONFIGURATION-2

Feature Under Test: Add Metadata Configuration To Media2 Profile (Media2_MetadataProfileConfiguration_Media2_AddMetadataProfileConfiguration)

Test Purpose: To verify that Client is able to add an metadata configuration to a media profile using the **AddConfiguration** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **AddConfiguration** operation with Type value is equal to "Metadata" present.
- Device supports Media2 Service (Media2Service).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetMetadataConfigurations** request message with specified **ProfileToken** to retrieve compatible metadata configurations for specified media profile from the Device.
2. Device responds with code HTTP 200 OK and **GetMetadataConfigurationsResponse** message.
3. Client invokes **AddConfiguration** request message with Type element value is equal to Metadata and with Configuration token that was recieved in

GetMetadataConfigurationsResponse message for the same media profile to add an metadata configuration to specified media profile on the Device.

4. Device responds with code HTTP 200 OK and **AddConfigurationResponse** message.

Test Result:

PASS -

- Client **AddConfiguration** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **AddConfiguration** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:AddConfiguration** AND
  - [S2] It has **tr2:Configuration/tr2:Type** element with value is equal to "Metadata" AND
- Device response to the **AddConfiguration** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tr2:AddConfigurationResponse**.
- There is Client **GetMetadataConfigurations** request in Test Procedure that fulfills the following requirements:
  - [S5] It is invoked before the Client **AddConfiguration** request AND
  - [S6] It has **tr2:ProfileToken** element with value is equal to **tr2:ProfileToken** element value from the **AddConfiguration** request AND
  - [S7] It is the last **GetMetadataConfigurations** request which corresponds to [S5], AND [S6] AND
- Device response to the **GetMetadataConfigurations** request fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] **soapenv:Body** element has child element **tr2:GetMetadataConfigurationsResponse** AND
  - [S10] It contains **tr2:Configurations** element with **@token** attribute value equal to **tr2:Configuration/tr2:Token** value for Configuration with **tr2:Configuration/tr2:Type** value is equal to Metadata from the **AddConfiguration** request message.

FAIL -

- The Client failed PASS criteria.

8.9 GET VIDEO SOURCE CONFIGURATIONS USING MEDIA2

Test Label: Video Source Configuration - Get Video Source Configurations

Test Case ID: MEDIA2_VIDEOSOURCECONFIGURATION-1

Feature Under Test: Get Video Source Configurations Using Media2 (Media2_VideoSourceConfiguration_Media2_GetVideoSourceConfigurations)

Test Purpose: To verify that video source configuration provided by Device is received by Client using the **GetVideoSourceConfigurations** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetVideoSourceConfigurations** operation for Media2 Service present.
- Device supports Media2 Service (Media2Service).
- Device supports Video feature for Media2 Service (Media2_Video) OR Device supports Video Source feature for Media2 Service (Media2_VideoSource).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetVideoSourceConfigurations** request message to retrieve an video source configuration or a list of video source configurations from the Device.
2. Device responds with code HTTP 200 OK and **GetVideoSourceConfigurationsResponse** message.

Test Result:

PASS -

- Client **GetVideoSourceConfigurations** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetVideoSourceConfigurations** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tr2:GetVideoSourceConfigurations** AND
- Device response on the **GetVideoSourceConfigurations** request fulfills the following requirements:

- [S2] It has HTTP 200 response code AND
- [S3] **soapenv:Body** element has child element **tr2:GetVideoSourceConfigurationsResponse**.

FAIL -

- The Client failed PASS criteria.

8.10 PULLPOINT

Test Label: Event Handling - Pull Point

Test Case ID: EVENTHANDLING-1

Feature Under Test: Pull Point (EventHandling_PullPoint)

Test Purpose: To verify that the Client is able to retrieve events using Pull Point.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with Pull Point event type.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes CreatePullPointSubscription message.
2. Device responds with code HTTP 200 OK and CreatePullPointSubscriptionResponse message.
3. Client invokes PullMessages command with Timeout and MessageLimit elements.
4. Device responds with code HTTP 200 OK and PullMessagesResponse message.

Test Result:

PASS -

- Client **CreatePullPointSubscription** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **CreatePullPointSubscription** request in Test Procedure fulfills the following requirements:
  - [S1] Client request contains "<CreatePullPointSubscription>" tag after the "<Body>" tag AND

- [S2] Device response contains "HTTP/* 200 OK" AND
- [S3] Device response contains "<CreatePullPointSubscriptionResponse>" tag AND

- Client **PullMessages** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **PullMessages** request in Test Procedure fulfills the following requirements:
  - [S4] Client request contains "<PullMessages>" tag after the "<Body>" tag AND
  - [S7] Device response contains "HTTP/* 200 OK" AND
  - [S8] Device response contains "<PullMessagesResponse>" tag.

FAIL -

- The Client failed PASS criteria.

8.11 SET SYNCHRONIZATION POINT (EVENT SERVICE)

Test Label: Set Synchronization Point - Set Synchronization Point

Test Case ID: SETSYNCHRONIZATIONPOINT-1

Feature Under Test: Set Synchronization Point (SetSynchronizationPoint_SetSynchronizationPointAction)

Test Purpose: To verify that the Client is able to use **SetSynchronizationPoint** operation for subscribtion.

Pre-Requisite:

- The Network Trace Capture files contains at least one Conversation between Client and Device with **SetSynchronizationPoint** operations present.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **SetSynchronizationPoint** message with valid **wsa:Action** header to synchronize its properties with the properties of the device.
2. Device responses with code HTTP 200 OK and **SetSynchronizationPointResponse** message.

Test Result:

PASS -

- Client **SetSynchronizationPoint** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **SetSynchronizationPoint** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **tev:SetSynchronizationPoint** AND
  - [S2] It contains **wsa:Action** element in header equal to "http://www.onvif.org/ver10/events/wsdl/PullPointSubscription/SetSynchronizationPointRequest" AND
- Device response on the **SetSynchronizationPoint** request fulfills the following requirements:
  - [S3] It has HTTP 200 response code AND
  - [S4] **soapenv:Body** element has child element **tev:SetSynchronizationPointResponse**

FAIL -

- The Client failed PASS criteria.

8.12 GET RULES

Test Label: Get Rules

Test Case ID: GETRULES-1

Feature Under Test: Get Rules (GetRules_GetRulesRequest)

Test Purpose: To verify that Client is able to retrieve available rules using the **GetRules** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetRules operation present.
- Device supports Rule Engine (RuleEngine).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetRules** request message to retrieve available Rules from the Device.
2. Device responds with code HTTP 200 OK and **GetRulesResponse** message.

Test Result:

PASS -

- Client **GetRules** request messages are valid according to XML Schemas listed in Namespaces AND

- Client **GetRules** request in Test Procedure fulfills the following requirements:
  - [S1] **soapenv:Body** element has child element **axt:GetRules** AND
- Device response on the **GetRules** request fulfills the following requirements:
  - [S2] It has HTTP 200 response code AND
  - [S3] **soapenv:Body** element has child element **axt:GetRulesResponse**.

FAIL -

- The Client failed PASS criteria.

8.13 GET SUPPORTED RULES

Test Label: Get Supported Rules

Test Case ID: GETSUPPORTEDRULES-1

Feature Under Test: Get Supported Rules (GetSupportedRules_GetSupportedRulesRequest)

Test Purpose: To verify that Client is able to retrieve supported rules using the **GetSupportedRules** operation.

Pre-Requisite:

- The Network Trace Capture files contains at least one conversation between Client and Device with GetSupportedRules operation present.
- Device supports Rule Engine (RuleEngine).

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetSupportedRules** request message to retrieve supported Rules from the Device.
2. Device responds with code HTTP 200 OK and **GetSupportedRulesResponse** message.

Test Result:

PASS -

- Client **GetSupportedRules** request messages are valid according to XML Schemas listed in Namespaces AND
- Client **GetSupportedRules** request in Test Procedure fulfills the following requirements:

- [S1] **soapenv:Body** element has child element **axt:GetSupportedRules** AND

- Device response on the **GetSupportedRules** request fulfills the following requirements:

  - [S2] It has HTTP 200 response code AND

  - [S3] **soapenv:Body** element has child element **axt:GetSupportedRulesResponse**.

FAIL -

- The Client failed PASS criteria.

8.14 H264 VIDEO STREAMING USING MEDIA2

Test Label: Video Streaming - H264

Test Case ID: MEDIA2_VIDEOSTREAMING_H264-1

Feature Under Test: H264 Video Streaming Using Media2 (Media2_VideoStreaming_H264_Media2_H264VideoStreaming)

Test Purpose: To verify that the Client is able to initiate and retrieve a video stream with H264 encoding type.

Pre-Requisite:

- Device supports H264 encoding for Video Streaming for Media2 Service (Media2_H264).

- The Network Trace Capture files contains at least one Conversation between Client and Device with Video Streaming of H264 encoding type.

- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service.

- Device supports H264 encoding for Video Streaming using Media2.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Video Source Configuration and Video Encoder Configuration with H264 Encoding value. GetStreamUri request is set for RtspUnicast OR RtspMulticast OR RTSP OR RtspOverHttp transport.

2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.

3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.

4. Device responds with code RTSP 200 OK and SDP information with Media Type: "video" and with encoding name "H264".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for H264 video streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.
9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "video" (m=video) with sessions attribute "rtpmap" with encoding name "H264" (see [RFC 3984], item 8.2.1. Mapping of MIME Parameters to SDP) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND

- There is a Device response on the **GetStreamUri** request for Media2 Service in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **tr2:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND
- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S12] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

8.15 H265 VIDEO STREAMING USING MEDIA2

Test Label: Video Streaming - H265

Test Case ID: MEDIA2_VIDEOSTREAMING_H265-1

Feature Under Test: H265 Video Streaming Using Media2 (Media2_VideoStreaming_H265_Media2_H265VideoStreaming)

Test Purpose: To verify that the Client is able to initiate and retrieve a video stream with H265 encoding type.

Pre-Requisite:

- Device supports H265 encoding for Video Streaming for Media2 Service (Media2_H265).
- The Network Trace Capture files contains at least one Conversation between Client and Device with Video Streaming of H265 encoding type.
- The Network Trace Capture files contains at least one Conversation between Client and Device with **GetStreamUri** for Media2 Service.
- Device supports H265 encoding for Video Streaming using Media2.

Test Procedure (expected to be reflected in network trace file):

1. Client invokes **GetStreamUri** request message for media profile that contains Video Source Configuration and Video Encoder Configuration with H265 Encoding value. GetStreamUri request is set for RtspUnicast OR RtspMulticast OR RTSP OR RtspOverHttp transport.
2. Device responds with code HTTP 200 OK and **GetStreamUriResponse** message.
3. Client invokes **RTSP DESCRIBE** request to retrieve media stream description.
4. Device responds with code RTSP 200 OK and SDP information with Media Type: "video" and with encoding name "H265".
5. Client invokes **RTSP SETUP** request without "onvif-replay" Require header and with transport parameter element to to set media session parameters for H265 video streaming.
6. Device responds with code RTSP 200 OK.
7. Client invokes **RTSP PLAY** request without "onvif-replay" Require header to start media stream.
8. Device responds with code RTSP 200 OK.

9. Client invokes **RTSP TEARDOWN** request to terminate the RTSP session.
10. If Device sends response to RTSP TEARDOWN, it has code RTSP 200 OK or RTSP 454.

Test Result:

Note: RTSP requests and RTSP response could be tunneled in HTTP if RTP-Unicast/RTSP/HTTP/TCP transport is used.

PASS -

- There is Client **RTSP DESCRIBE** request in Test Procedure
- Device response on the **RTSP DESCRIBE** request fulfills the following requirements:
  - [S1] It has RTSP 200 response code AND
  - [S2] SDP packet contains media type "video" (m=video) with sessions attribute "rtpmap" with encoding name "H265" (see [RFC 3984], item 8.2.1. Mapping of MIME Parameters to SDP) AND
- There is Client **RTSP SETUP** request in Test Procedure fulfills the following requirements:
  - [S3] It invoked for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S4] It invoked after the Client **RTSP DESCRIBE** request AND
  - [S5] RTSP address that was used to send **RTSP SETUP** is correspond to corresponding media Control URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S6] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP SETUP** request fulfills the following requirements:
  - [S7] It has RTSP 200 response code AND
- There is a Device response on the **GetStreamUri** request for Media2 Service in Test Procedure fulfills the following requirements:
  - [S8] It has HTTP 200 response code AND
  - [S9] It received for the same Device as for the Client **RTSP DESCRIBE** request AND
  - [S10] It received before the Client **RTSP DESCRIBE** request AND
  - [S11] It contains **tr2:Uri** element which value is equal to RTSP address that was used to send the **RTSP DESCRIBE** request AND

- There is Client **RTSP PLAY** request in Test Procedure fulfills the following requirements:
  - [S12] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S13] It invoked after the Client **RTSP SETUP** request AND
  - [S14] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
  - [S15] It does not contain **Require** request header field with value is equal to "onvif-replay" AND
- Device response on the **RTSP PLAY** request fulfills the following requirements:
  - [S16] It has RTSP 200 response code AND
- There is Client **RTSP TEARDOWN** request in Test Procedure fulfills the following requirements:
  - [S17] It invoked for the same RTSP session as the Client **RTSP SETUP** request AND
  - [S18] It invoked after the Client **RTSP PLAY** request AND
  - [S19] RTSP address that was used to send it is correspond to corresponding media Control URL or session Control URL or Content-Base URL from SDP packet (see [RFC 2326, C.1.1 Control URL]) AND
- If there is Device response on the **RTSP TEARDOWN** request then it fulfills the following requirements:
  - [S20] It has RTSP 200 response code.

FAIL -

- The Client failed PASS criteria.

Annex A Test for Appendix A

A.1 Required Number of Devices Summary

Required number of devices and Device feature dependency used in this test specification are listed in the Table.

Table A.1. Required Number of Devices Summary

Feature ID	Feature Name	Required Number of Devices	Check Condition based on Device Features	Check Condition based on Device Features ID
tc.HTTPDigest	HTTP Digest	3	Digest	Digest
tc.HTTPDigestForRTSP	HTTP Digest Authentication for RTSP	3	Profile T or Profile M	ProfileTSupported or ProfileMSupported
tc.GetServices	Get Services	3	GetServices is supported by Device.	GetServices
tc.Discovery	Discovery	3	Discovery	All
tc.DeviceDiscoveryTypeFilter	Device Discovery Type Filter	3	Device Discovery Type is supported by Device.	DiscoveryTypesTdsDevice
tc.Media2_MetadataStreaming_ProfileM	Metadata Streaming Using Media2	3	Media2 Service is supported by Device.	Media2Service
tc.GetSupportedMetadata	Get Supported Metadata	3	Supported Metadata is supported by Device.	SupportedMetadata
tc.System	System	3	None	All
tc.Media2_MetadataProfileConfiguration_ProfileM	Metadata Profile Configuration Using Media2 (Profile M)	1	Metadata feature and either Video Source or Video under Media2 Service is supported by Device.	Media2_Metadata AND (Media2_VideoSource OR Media2_Video)

Feature ID	Feature Name	Required Number of Devices	Check Condition based on Device Features	Check Condition based on Device Features ID
tc.Media2_AnalyticsProfileConfiguration	Analytics Profile Configuration Using Media2	1	Analytics (Media2 Service) is supported by Device.	Media2_Analytics
tc.AnalyticsModuleConfiguration	Analytics Module Configuration	1	Analytics Modules under Analytics Service is supported by Device.	AnalyticsModules
tc.Media2_CreateProfile	Create Profile Using Media2	1	Media2 Service is supported by Device.	Media2Service
tc.Media2_VideoStreaming_ProfileM	Video Streaming Using Media2	1	H264 or H265 under Media2 Service is supported by Device.	Media2_H264 OR Media2_H265
tc.ImageData	Image Data	1	Embedded Image Sending Type.	EmbeddedImageSendingType
tc.Media2_HTTPSStreaming_ProfileM	HTTPS Streaming Using Media2 (Profile M)	1	No (ONVIF Profile M Simulator is used as device).	
tc.EventHandlingUsingPullPoints	Event Handling Using Pull Points	1	Pull Point Notification is supported by Device.	no UnsupportedPullPointNotification
tc.RuleConfiguration	Rule Configuration	1	Media 2 Service and Rule Engine is supported by Device.	Media2Service AND RuleEngine
tc.ObjectClassification	Object Classification	1	Object Classification is supported by Device.	ObjectClassification
tc.HumanFaceMetadata	Human Face Metadata	1	Human Face is supported by Device.	HumanFace

Feature ID	Feature Name	Required Number of Devices	Check Condition based on Device Features	Check Condition based on Device Features ID
tc.HumanBody Metadata	Human Body Metadata	1	Human Body is supported by Device.	HumanBody
tc.VehicleMetadata	Vehicle Metadata	1	Vehicle Info is supported by Device.	VehicleInfo
tc.LicensePlateMetadata	License Plate Metadata	1	License Plate Info is supported by Device.	LicensePlateInfo
tc.GeoLocationMetadata	GeoLocation Metadata	1	Geo Location is supported by Device.	GeoLocation
tc.FaceRecognitionEvent	Face Recognition Event	1	Face Recognition Rule is supported by Device.	FaceRecognitionRule
tc.LicensePlateRecognitionEvent	License Plate Recognition Event	1	License Plate Recognition Rule is supported by Device.	LicensePlateRecognitionRule
tc.LineCrossingCounterEvent	Line Crossing Counter Event	1	Line Crossing Counting Rule is supported by Device.	LineCrossingCountingRule
tc.EventBrokerConfiguration	Event Broker Configuration	1	Event Broker is supported by Device.	EventBroker
tc.Media2_MetadataConfiguration_ProfileM	Metadata Configuration Using Media2	1	Media2 Service is supported by Device.	Media2Service
tc.Media2_MetadataConfiguration	Metadata Configuration Using Media2	1	Media2 Service is supported by Device.	Media2Service
tc.Media2_MetadataProfileConfiguration	Metadata Profile Configuration Using Media2	1	Media2 Service is supported by Device.	Media2Service