ONVIF™
Cloud Integration Specification

Version 25.06

June, 2025

CONTENTS

1 Scope

This document defines the connection protocol for sharing cloud native devices between cloud-based clients.

2 Normative references

IETF RFC 2617 - HTTP Authentication: Basic and Digest Access Authentication
<https://tools.ietf.org/html/rfc2617>

IETF RFC 6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage
<https://tools.ietf.org/html/rfc6750>

IETF RFC 7616 - HTTP Digest Access Authentication
<https://tools.ietf.org/html/rfc7616>

ONVIF Core Specification
<http://www.onvif.org/onvif/specs/core/ONVIF-Core-Specification.pdf>

ONVIF Security Service Specification
<https://www.onvif.org/specs/srv/media/ONVIF-Security-Service-Spec.pdf>

OpenAPI 3.1.0
<https://spec.openapis.org/oas/v3.1.0>

3 Terms and Definitions

3.1 Definitions

Device Sharing	The procedure, started by the owner of the device, to allow another Operational Cloud Service to connect to the device.
Device Transfer	The procedure, started by the owner of the device, to allow another Operational Cloud Service to connect to the device. With device transfer, the original cloud service loses access.
Operational Cloud Service	The cloud platform receiving access to the devices from a provisioning cloud service.
Manufacturer Cloud Service	The initial cloud platform, provided by the device vendor, where the user can take ownership of the devices.
Provisioning Cloud Service	The cloud platform currently managing the device and starts either device transfer or sharing.
Share Token	A Token passed between a "Provisioning Cloud Service" and an "Operational Cloud Service" to prove that the "Provisioning Cloud Service" has access rights to the device.
Uplink	The connection established by the local service to the remote client.
Web Services	A web service is a software component or system that allows devices to communicate over the internet by exchanging data.

3.2 Abbreviations

JSON	JavaScript Object Notation
JWT	JSON Web Token
MCS	Manufacturer Cloud Service
OCS	Operational Cloud Service
PCS	Provisioning Cloud Service

VMS	Video Management System
YAML	Yet Another Markup Language.

4 Overview

On-boarding devices in cloud native Video Management Systems involves two major steps:

- A device taken out of the box connects with its pre-programmed parameters to a Manufacturer Cloud Service (MCS), so that the user can take ownership of the device. Once the device is successfully claimed by the user, the MCS behaves as a Provisioning Cloud Service (PCS).
- The user can transfer the device from a PCS to an Operational Cloud Service (OCS), i.e. the cloud native VMS the user expects to use for his surveillance system.

N.B. that at any time, a OCS may act as a PCS.

The scope of this cloud service covers only the second step, i.e. device sharing or transferring between different cloud services. The procedure outlined in these specifications can be used to share or transfer a device from a MCS to an OCS during the initial setup, but also from an old PCS to a new OCS.

Figure Figure 1 demonstrates the two phases. **N.B.** the grey rectangle visually identifies the scope of these specifications.

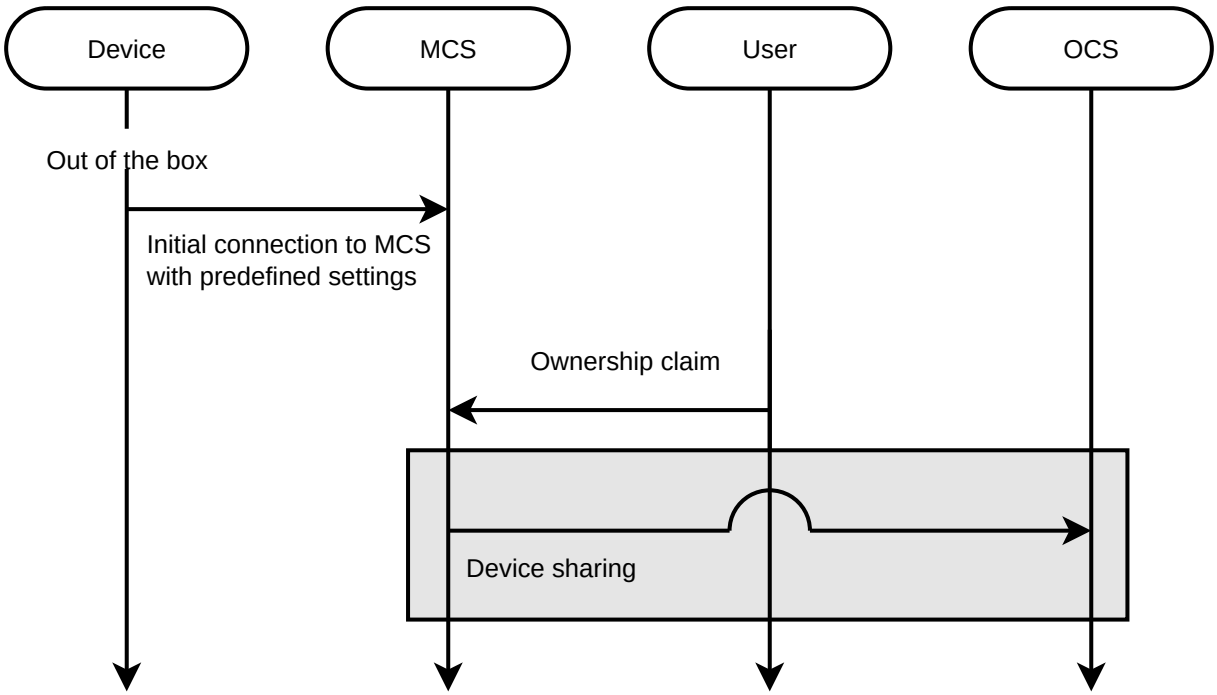

Figure 1: Onboarding phases and ONVIF scope

4.1 Web Services

Unlike for other services defined by ONVIF, in this document the term Web Services is the name of a standardized method of integrating applications using open, platform independent Web Services standards such as JSON, OpenAPI 3.1.0 and YAML over an IP network. JSON is used as the data description syntax, OpenAPI is used for message transfer and YAML is used for describing the services.

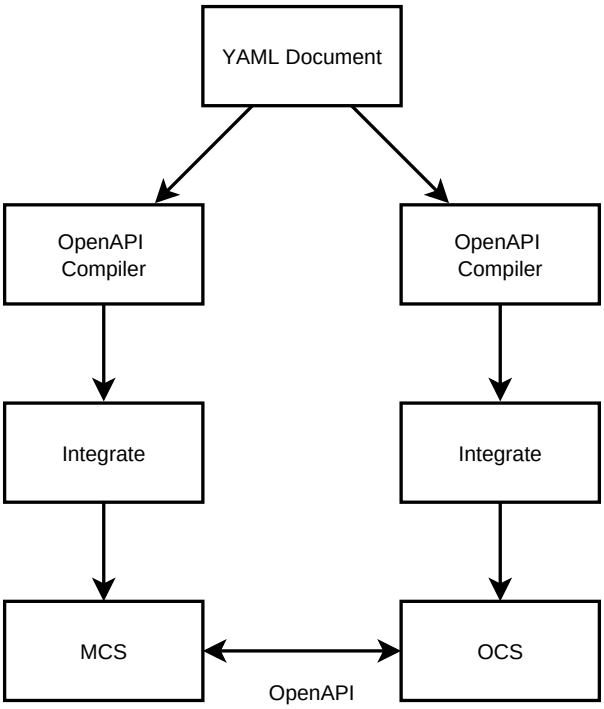

Figure 2: OpenAPI based development principles

Figure 2 gives an overview of the basic principles for development based on Web Services and OpenAPI. The service provider (MCS) implements the ONVIF service. The service is described using the YAML-based file. Then, the YAML file is used as the basis for the service requester (OCS) implementation/integration. Integration is simplified using code generating tools that generate platform specific code that can be used by the developers to integrate the Web Service.

The Web Service provider and requester communicate using the OpenAPI message exchange protocol. OpenAPI is a lightweight messaging protocol used to encode the information in a Web Service request and in a response message before sending them over a network. OpenAPI messages are independent of any operating system or protocol and may be transported using a variety of Internet protocols. This ONVIF standard defines conformant transport protocols for the SOAP messages for the described Web Services.

The Web Service overview section introduces into the general ONVIF service structure, the command definition syntax in the specification, error handling principles and the adopted Web Service security mechanisms.

4.2 Security

4.2.1 Authentication

The functions defined within this document do not require authehtication, since presenting a valid share token is a valid proof that the user was successfully authenticated by the MCS.

4.2.2 Information protection

The services defined in this standard, whenever consumed, shall be protected by using only HTTPS as transport, in order to protect the share token from interception and unauthorized use.

4.3 Format of the share token

To start the device sharing operation, the OCS must provide a share token to the PCS, to prove it has the rights to access the camera. The way the OCS retrieves the share token is beyond the scope of this specification.

The share token must be a JWT and must include the following claims:

- *onvif:xaddr*: The full URI where the PCS will be accepting the incoming requests from the OCS

- *onvif:sn*: The serial number of the device.
- *onvif:model*: The device model.
- *onvif:manufacturer*: The manufacturer of the device.

The claims *onvif:sn*, *onvif:model* and *onvif:manufacturer* shall match the values returned by GetDeviceInformation of the Device service.

5 Device sharing

5.1 startDeviceSharing

This operation triggers sharing a device with a new Operational Cloud Service.

Once the information is sent to the camera, the camera will generate a public key associated to the assigned ClientID. The generated public key will be passed to the PCS, so that it will transfer it to the OCS with the deviceSharingCompleted function, so that it will be able to authenticate the device and let it retrieve the JWT meant to be used to authenticate the Uplink Service.

REQUEST:

startDeviceSharing [startDeviceSharing]

RESPONSE:

- **This is an empty message**

FAULTS:

400 - Invalid Argument Value

5.2 deviceSharingCompleted

This operation notifies the new Operational Cloud Service that the camera was successfully configured.

The public key shall be a base64 encoded DER-encoded PKCS#10 certification request.

REQUEST:

deviceSharingCompleted [deviceSharingCompleted]

RESPONSE:

This is an empty message.

FAULTS:

No command specific faults defined.

Annex A.
Revision History

Rev.	Date	Editor	Changes
25.06	June 2025	Ottavio Campana	First release