ONVIF™
PACS Architecture and Design Considerations

Version 18.06
June 2018

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT. THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

CONTENTS

Contributors

ASSA ABLOY	Patrik Björling Rygert
ASSA ABLOY	Mattias Rengstedt
Axis Communications AB	Johan Adolfsson
Axis Communications AB	Marcus Johansson
Axis Communications AB	Robert Rosengren
Axis Communications AB	Derek Wang
Axis Communications AB	Emil Selinder
AxxonSoft	Yuri Timenkov
Bosch	Mohane Caliaperoumal
Bosch	Dirk Schreiber
Hirsch Electronics/ Identive Group	Rob Zivney
Honeywell	Marine Drive
Honeywell	Neelendra Bhandari
Honeywell	Uvaraj Thangarajan
Honeywell	Vinay Ghule
PACOM	Eugene Scully
PACOM	Steve Barton
Schneider Electric	Mike Berube
Siemens AG	Klaus Baumgartner
Siemens AG	Suresh Raman
Siemens AG	Suresh Krishnamurthy

1 Scope

1.1 General

This document defines the core concepts of physical access control and sets some basic rules for all ONVIF PACS web service specifications.

Web service usage and common ONVIF functionality are outside of the scope of this document. Please refer to [ONVIF Core Specification] for more information.

1.2 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 Annex H of [ISO/IEC Directives].

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

ISO/IEC Directives*, Part 2, Principles and rules for the structure and drafting of ISO and IEC documents, Edition 7.0, May 2016*
<http://www.iec.ch/members_experts/refdocs/iec/isoiecdir-2%7Bed7.0%7Den.pdf>

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

Access Control Unit
Part of an access control system that interfaces with readers, locking devices and sensing devices, making a decision to grant or deny access through a portal.

From an ONVIF perspective, it is a device or system implementing at least the access control service. Often, it is a microprocessor-based circuit board that manages access to a secure area. The access control unit receives information that it uses to determine through which doors and at what times credential holders are granted access to secure areas. Based on that information, the access control unit can lock/unlock doors, sound alarms, and communicate status to a host computer.

Access Point
A logical composition of a physical door, reader(s) and/or a request-to-exit device controlling access in one direction.

Area
A protected or controlled area defined by a physical boundary, through which passage is controlled by means of one or more doors.

Client
An ONVIF service requester. A typical ONVIF network system may have multiple clients that handle device configuration and device management operations for numerous devices. A device providing services may also act as a client to other devices.

Device
An ONVIF service provider implementing one or more ONVIF services. E.g. an access control unit or a door control unit.

Door
A physical door, barrier, turnstile, etc. which can be controlled remotely and restricts access between two areas. A door is usually equipped with an electronic lock and a door monitor.

Door Control Unit From an ONVIF perspective, it is a device or system implementing at least the door control service, but not the access control service. Often, it is a microprocessor-based circuit board that manages door locks and/or door monitors for one or more doors.

Door Lock A device that secures a door to prevent access, except when explicitly allowed by the access control system. Lock types include electromagnetic, electric strike, etc.

Door Monitor Electrical component used to monitor the open or closed status of a door, or locked/unlocked status of a locking device, or the secure/unsecure status of an electromagnetic lock or armature plate.

Also known as door contact sensor.

Reader Device for the input of credentials. Examples include card readers, biometric readers, etc.

3.2 Abbreviations

ACMS	Access Control Management System
BMS	Building Management System
HTTP	Hypertext Transfer Protocol
PACS	Physical Access Control System
PSIM	Physical Security Information Management
TLS	Transport Layer Security

4 Overview

4.1 Introduction

The ONVIF PACS specifications provide interfaces to enable integration of physical security equipment with other devices (e.g., video cameras) and systems (e.g., video monitoring system, PSIM, BMS etc.).

The standard specifies only the data and control flow between a client and the ONVIF services without reference to how the services are deployed in physical devices.

The ONVIF PACS specifications do not define internal communication between an access control unit and its components if they are implemented on a single device. However, future versions may provide interfaces to integrate these parts from different vendors.

4.2 Interoperability

The ONVIF specification provides new interoperability opportunities by separating configuration from control and monitoring. In traditional systems, the central management system pushes all configurations data to devices on startup and expects that this configuration data is not changed by other clients. Instead, each ONVIF client shall expect that all information is stored on end-devices and can be changed by others.

ONVIF PACS relies on Service-Oriented Architecture principles. This allows installations where different components can be replaced or updated independently.

4.3 Event handling

Event handling is a crucial part of access control operations. In addition to real-time event delivery ONVIF provides the means for accessing stored events on the edge to deliver them if connection is lost.

Events are divided into 3 groups depending on their origin and purpose:

1. Configuration change events. These events are provided to achieve interoperability between several clients that control a single device simultaneously.

2. Transaction events. The core functionality of PACS that provides daily monitoring of all access events, including access granted events designed to notify clients about all detailed information (who, when and probably where have passed) on every particular access granted event, access denial events (that may or may not contain reason information), etc.

3. Alarms and faults events. These events provide health status monitoring allowing operators take action in case of hardware failure, intrusion or other suspicious activity.

Please refer to [ONVIF Core Specification] for details on event delivery mechanism.

4.4 Architecture

The ONVIF PACS specifications do not mandate any specific physical device layout. The scheme provided in Figure 1 is not intended to be taken as a pattern but to serve as a reference for better understanding of the given specification. Based on the definitions below, different physical configurations of an access controlled door are possible.

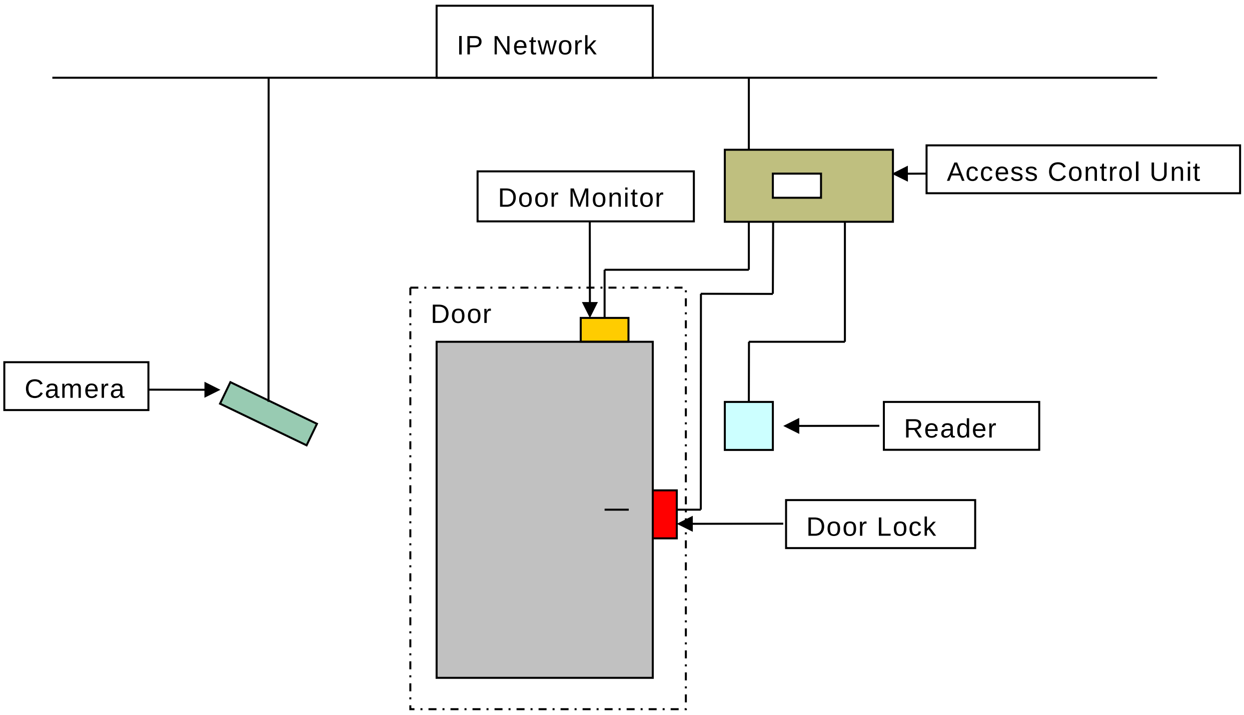

Figure 1 – Schematic overview of an access controlled door

A door that is controlled by a physical access control system is equipped with the following devices:

- An access control unit that provides connections for reader, door sensor, door lock and additional digital inputs and outputs. This panel enables the software to interact with the physical devices. Sometimes these panels also contain storage and local intelligence to provide an offline functionality, so that the door will work as expected, even if there is no management system above available.
- A reader that is able to read a credential. In most cases a reader is only mounted at the outer (unsecure) side of the door. If the system monitors when somebody is leaving an area, a reader will be mounted on both sides of the door.
- A door monitor that signals the control panel that the door is open or closed.
- A door lock that can be engaged by the access control unit to release the door, e.g. in case of an authorized credential is recognized.
- (Optional) a camera that shows the person waiting for the door to be opened.

The access control unit will through the IP network be connected to a system, typically a monitoring console, for monitoring and configuration.

4.5 External authorization (Overriding)

External authorization is a feature used to take access decisions for an access point outside the access control unit. External authorization entails but is not limited to a policy within the access control unit where the access control unit delegates the access decisions to an outside entity such as a guard or ACMS.

4.6 Security considerations

The specification assumes possibility of building PACS systems interacting on device level. This implies more security consideration than regular client-server interaction. The [ONVIF Core Specification] defines several mechanisms to achieve this. They include, but are not limited to

- TLS for transport encryption
- HTTP and WS-Security for client authentication
- User management and Access Policies for client authorization
- IEEE 802.1X certificate management for server authentication and spoofing protection.

Please refer to the respective whitepapers and specifications for more information.

4.7 Physical and logical security

This specification distinguishes two types of security:

- Physical security prevents unauthorized personnel, attackers or accidental intruders from physically accessing a building, room or etc.
- Logical security protects information and restricts access to managing equipment.

4.8 Design considerations

4.8.1 Instance-level capabilities

A single PACS device may have diverse components of the same type. For example, an access control unit may operate two doors: one at the entrance to the building which has secure locking, monitoring and alarm abilities, and the other one is internal which can be only locked and unlocked.

Therefore, capabilities can be divided into two groups:

- Overall service capabilities;
- Capabilities for a particular entity in the service. It can also work in conjunction with the GetEventProperties function to provide a finer control over the system.

4.8.2 Retrieving status

The PACS family of ONVIF services defines two parallel mechanisms for retrieving status information for most entities:

- Get<*Entity*>State functions return a cumulative snapshot of the current state, operating mode and other run-time information.
- The Event Service returns up-to-date and consistent states of entities. Each entity provides a set of events (usually one per each field in the State type) to notify a client about status changes. As far as these events are property events, a client receives the current state whenever a new subscription is initialized.

4.8.3 Retrieving system configuration

The PACS family of ONVIF services defines several Get-functions that can return data incrementally. These functions allow the processing of a large number of entities even though resources are highly constrained.

To return data incrementally, these functions make use of a parameter called StartReference. StartReference is a device internal identifier used to continue fetching data from the last position, and allows a client to iterate over a large dataset in smaller chunks. The device handles a reasonable number of different StartReferences at the same time and they live for a reasonable time so that clients are able to fetch complete datasets.

An ONVIF compliant client always passes the value returned from a previous request to continue fetching data. Client do not use the same reference more than once.

For example, the StartReference can be incrementing start position number or underlying database transaction identifier.

The returned NextStartReference is used as the StartReference parameter in successive calls, and may be changed by device in each call.

The following pseudo-code demonstrates how information about all access points can be obtained from a device:

```
StartRef = null
do {
  Response = GetAccessPointInfoList(StartReference = StartRef)
  if (Response.AccessPointInfo != null) {
    AllAccessPoints.Append(Response.AccessPointInfo)
  }
  StartRef = Response.NextStartReference
} while (StartRef != null)
```

4.9 Naming considerations

4.9.1 Data structures

All PACS specifications have entities in two variants; a full version of the object (e.g. Credential) and a limited version of the object (e.g. CredentialInfo).

The limited version, suffixed with Info, normally contains basic information such as token, name, and description. The full version contains more detailed configuration (referenced entities belongs here).

The full version of the object may also contain more sensitive information, and will therefore require different user levels to be accessed. User levels are described in [ONVIF Core Specification].

4.9.2 Operations

4.9.2.1 Get<Entity>Info

The Get<Entity>Info set of commands retrieves lists of the limited version of the entity. This is done by specifying a list of tokens in the request.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than the MaxLimit capability, a TooManyItems fault shall be returned.

4.9.2.2 Get<Entity>InfoList

The Get<Entity>InfoList set of commands retrieves paged lists of the limited version of the entity. This is done by specifying a StartReference and a Limit parameter in the request.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Please refer to section 4.8.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

4.9.2.3 Get<Entities>

The Get<Entities> set of commands retrieves lists of the full version of the entity. This is done by specifying a list of tokens in the request.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than the MaxLimit capability, a TooManyItems fault shall be returned.

4.9.2.4 Get<Entity>List

The Get<Entity>List set of commands retrieves paged lists of the full version of the entity. This is done by specifying a StartReference and a Limit parameter in the request.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Please refer to section 4.8.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

4.9.2.5 Create<Entity>

The Create<Entity> set of commands create a full version of the entity.

The token field in the request shall be empty and the device shall allocate a token for the created entity. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

4.9.2.6 Set<Entity>

The Set<Entity> set of commands synchronizes the device with the information in the request.

If an entity with the specified token does not exist in the device, the entity is created. If an entity with the specified token exists, then the entity is modified.

The client shall always specify a token in the request. A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

4.9.2.7 Modify<Entity>

The Modify<Entity> set of commands modifies an existing entity.

The token of the entity to modify shall be specified in the token field of the request and shall not be empty. All other fields in the structure shall overwrite the fields of the specified entity in the device.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

4.9.2.8 Delete<Entity>

The Delete<Entity> set of commands deletes an existing entity.

If the entity have dependencies to other entities some devices may not be able to delete the entity, and consequently a ReferenceInUse fault shall be generated.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

4.10 PACS types

The WSDL files for all PACS services include the types.xsd schema definition.

It contains the following types:

- **DataEntity**

  All PACS entities extend the DataEntity type. It contains the token field.

- **Name**

  The name field of all PACS entities are of type Name.

- **Description**

  The description field of all PACS entities are of type Description.

- **PositiveInteger**

  A type for positive integers, with the minimum value of 1.

- **Attribute**

  A key/value pair with a field for the name and a field for the value.

Annex A. Revision History

Rev.	Date	Editor	Changes
18.06	Jun-2018	Patrik Björling Rygert	Extracted from Access Control Specification