Open Connectivity Foundation

admin@openconnectivity.org

OCF UPnP Bridge and Resources Specification

Version 0.2

Legal Disclaimer

THIS IS A DRAFT SPECIFICATION ONLY AND HAS NOT BEEN ADOPTED BY THE OPEN CONNECTIVITY FOUNDATION. THIS DRAFT SPECIFICATION MAY NOT BE RELIED UPON FOR ANY PURPOSE OTHER THAN REVIEW OF THE CURRENT STATE OF THE DEVELOPMENT OF THIS DRAFT SPECIFICATION. THE OPEN CONNECTIVITY FOUNDATION AND ITS MEMBERS RESERVE THE RIGHT WITHOUT NOTICE TO YOU TO CHANGE ANY OR ALL PORTIONS HEREOF, DELETE PORTIONS HEREOF, MAKE ADDITIONS HERETO, DISCARD THIS DRAFT SPECIFICATION IN ITS ENTIRETY OR OTHERWISE MODIFY THIS DRAFT SPECIFICATION AT ANY TIME. YOU SHOULD NOT AND MAY NOT RELY UPON THIS DRAFT SPECIFICATION IN ANY WAY, INCLUDING BUT NOT LIMITED TO THE DEVELOPMENT OF ANY PRODUCTS OR SERVICES. IMPLEMENTATION OF THIS DRAFT SPECIFICATION IS DONE AT YOUR OWN RISK AMEND AND IT IS NOT SUBJECT TO ANY LICENSING GRANTS OR COMMITMENTS UNDER THE OPEN CONNECTIVITY FOUNDATION INTELLECTUAL PROPERTY RIGHTS POLICY OR OTHERWISE. IN CONSIDERATION OF THE OPEN CONNECTIVITY FOUNDATION GRANTING YOU ACCESS TO THIS DRAFT SPECIFICATION, YOU DO HEREBY WAIVE ANY AND ALL CLAIMS ASSOCIATED HEREWITH INCLUDING BUT NOT LIMITED TO THOSE CLAIMS DISCUSSED BELOW, AS WELL AS CLAIMS OF DETRIMENTAL RELIANCE.

The OIC logo is a trademark of Open Interconnect Consortium, Inc. in the United States or other countries. \*Other names and brands may be claimed as the property of others.

Copyright © 2016 Open Interconnect Consortium, Inc. All rights reserved.

Copying or other form of reproduction and/or distribution of these works are strictly prohibited

CONTENTS

[1 Scope 5](#_Toc471806280)

[2 Normative references 6](#_Toc471806281)

[3 Terms, definitions symbols and abbreviations 8](#_Toc471806282)

[3.1 Terms and definitions 8](#_Toc471806283)

[3.2 Symbols and abbreviations 9](#_Toc471806284)

[3.3 Conventions 10](#_Toc471806285)

[4 Document conventions and organization 11](#_Toc471806286)

[4.1 Notation 11](#_Toc471806287)

[4.2 Data types 11](#_Toc471806288)

[5 Operational Scenarios 12](#_Toc471806289)

[5.1 Creation 12](#_Toc471806290)

[5.2 Discovery 12](#_Toc471806291)

[5.2.1 Endpoint Discovery 12](#_Toc471806292)

[5.2.2 Resource Discovery 12](#_Toc471806293)

[6 OCF-UPnP Bridge Device 13](#_Toc471806294)

[7 Introspection 17](#_Toc471806295)

[8 Security 18](#_Toc471806296)

[9 Device Type Mapping 19](#_Toc471806297)

[9.1 Introduction 19](#_Toc471806298)

[9.2 UPnP Device Types to OCF Device Types 19](#_Toc471806299)

[10 Resource Definitions 20](#_Toc471806300)

[10.1 Introduction 20](#_Toc471806301)

[10.2 UPnP Service Types to OCF Resource Types 20](#_Toc471806302)

Figures

[Figure 1: OCF-UPnP Bridge Device Component 8](#_Toc471806303)

[Figure 3: Example of an OCF-UPnP Resource 12](#_Toc471806304)

[Figure 4 Overview of an OCF-UPnP Bridge Device bridging UPnP Devices 13](#_Toc471806305)

[Figure 5 - Requesting Bridge and Light devices within an OCF-UPnP Bridge 15](#_Toc471806306)

[Figure 6 - Requesting all device types within an OCF-UPnP Bridge 16](#_Toc471806307)

Tables

[Table 9‑1 UPnP Device Types to OCF Device Types Mapping. 19](#_Toc471806308)

[Table 10‑1 UPnP Service Types to OCF Resource Types Mapping 20](#_Toc471806309)

[Table 10‑2 OCF-UPnP Resource Summary 20](#_Toc471806310)

# Scope

This document specifies and describes OCF-UPnP Data Models, the framework for translation between OCF devices and UPnP Devices, and specifies the behavior of a translator that exposes UPnP devices to OCF clients, and exposes OCF devices to UPnP applications. Translation of protocols other than UPnP is left to a future version of this specification. This document provides generic requirements that apply unless overridden by a more specific document

# 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.

OIC Specifications, *Open Interconnect Consortium Specifications*, Version 1.1.

<https://openconnectivity.org/resources/specifications>

* OIC Core Specification, *Open Interconnect Consortium Core Specification*, Version 1.1

<https://openconnectivity.org/specs/OIC_Core_Specification_v1.1.0.pdf>

* OIC Resource Type Specification, *Open Interconnect Consortium Resource Type Specification*, Version 1.1

<https://openconnectivity.org/specs/OIC_Resource_Type_Specification_v1.1.0.pdf>

* OIC Smart Home Device Specification, *Open Interconnect Consortium Smart Home Device Specification*, Version 1.1

<https://openconnectivity.org/specs/OIC_SmartHome_Device_Specification_v1.1.0.pdf>

* OIC Security Specification, *Open Interconnect Consortium Security Specification*, Version 1.1

<https://openconnectivity.org/specs/OIC_Security_Specification_v1.1.0.pdf>

UPnP Specifications-Standards: Device Control Protocols, *Universal Plug and Play - Standards: Device Control Protocols* *Specifications*  
<https://openconnectivity.org/resources/specifications/upnp/specifications>

OCF UPnP Add-on Services Specification

* DeviceProtection:1 **Service** Specifications, *DeviceProtection:1 Service* *Specifications*  
  <https://openconnectivity.org/resources/specifications/upnp/specifications/device-protection-v-1-0>

OCF UPnP Certification Testing - *OCF* *UPnP Certification Testing*

<https://openconnectivity.org/certification/upnp-certification>

OCF-UPnP Data Models, *OCF-UPnP Data Models*  
<https://github.com/openconnectivityfoundation/UPnP-models>

IEEE 754, *IEEE Standard for Floating-Point Arithmetic*, August 2008

IETF RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace, July 2005  
<https://www.rfc-editor.org/info/rfc4122>

IETF RFC 4648, The Base16, Base32, and Base64 Data Encodings, October 2006  
<https://www.rfc-editor.org/info/rfc4648>

IETF RFC 6973, *Privacy Considerations for Internet Protocols,* July 2013  
<https://www.rfc-editor.org/info/rfc6973>

IETF RFC 7049, *Concise Binary Object Representation (CBOR)*, October 2013  
<https://www.rfc-editor.org/info/rfc7049>

IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014 [http://www/ietf.org/rfc/rfc7159.txt](http://www/ietf.org/rfc/rfc7159.txt%20)

JSON Schema Core, *JSON Schema: core definitions and terminology*, January 2013  
<http://json-schema.org/latest/json-schema-core.html>

JSON Schema Validation, *JSON Schema: interactive and non interactive validation*, January 2013  
<http://json-schema.org/latest/json-schema-validation.html>

RAML, *Restful API modelling language,* Version 0.8.

<https://github.com/raml-org/raml-spec/blob/master/raml-0.8.md>

# Terms, definitions symbols and abbreviations

## Terms and definitions

OCF-UPnP Bridge Device

An OCF-UPnP Bridge Device is conformant to the normative requirements contained in this specification. The OCF-UPnP Bridge Device is represents UPnP devices that exist on the network and communicates using UPnP Protocol rather than OCF protocols



Figure 1: OCF-UPnP Bridge Device Component

UPnP Device(s)

A Device(s) that is(are) conformant to the normative requirements contained in *Universal Plug and Play - Standards: Device Control Protocols* *Specifications*.

UPnP Protocol

Protocol is conformant to the normative requirements contained in *Universal Plug and Play - Standards: Device Control Protocols* *Specifications*

Virtual UPnP Client

A logical representation of an OCF Client, which an OCF-UPnP Bridge Device exposes to UPnP Devices.

Translator

An OCF-UPnP Bridge Device component that is responsible for translating to or from a specific UPnP Protocol/device(s).

Virtual OCF Device

A logical representation of an UPnP Device, which an OCF-UPnP Bridge Device exposes to OCF Clients.

OCF Client

a logical entity that accesses an OCF Resource on an OCF Server, which might be a Virtual OCF Device exposed by the OCF-UPnP Bridge Device.

OCF Device

A logical entity that assumes one or more OCF roles (OCF Client, OCF Server). More than one OCF Device can exist on the same physical platform.

OCF Server

A logical entity with the role of providing resource state information and allowing remote control of its resources.



OCF Resource

Represents an artifact modelled and exposed by the OCF Framework

OCF Resource Property

A significant aspect or notion including metadata that is exposed through the OCF Resource

OCF Resource Type

An OCF Resource Property that represents the data type definition for the OCF Resource

OCF-UPnP Resource

Represents an artifact modelled and exposed by an UPnP Protocol.

OCF-UPnP Resource Type

A schema used with an UPnP Protocol and is conformant to the normative requirements contained in *Resource Definitions*.

## Symbols and abbreviations

**CRUDN**

Create Read Update Delete Notify

Indicating which operations are possible on the resource

**CSV**

Comma Separated Value

Construction to have more fields in 1 string separated by commas. If a value contains a comma, then the comma can be escaped by adding “\” in front of the comma.

**OIC**

Open Interconnect Consortium

The organization that created these specifications

**OCF**

Open Connectivity Foundation

Organization that created these specifications

**RAML**

RESTful API Modeling Language

Simple and succinct way of describing practically RESTful APIs (see RAML)

**UPnP**

Universal Plug and Play

A set of [networking protocols](https://en.wikipedia.org/wiki/Networking_protocol) that permits networked [devices](https://en.wikipedia.org/wiki/Peripheral_device), such as personal computers, printers, Internet gateways, [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) access points and mobile devices to seamlessly discover each other's presence on the network and establish functional [network services](https://en.wikipedia.org/wiki/Network_service) for data sharing, communications, and entertainment. It is fully defined in UPnP Specifications-Standards: Device Control Protocols. Since 2016, all UPnP standards are now managed by the OCF -<https://en.wikipedia.org/wiki/Open_Connectivity_Foundation>.

## Conventions

In this specification, a number of terms, conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Network Architecture). Any lowercase uses of these words have the normal technical English meaning.

# Document conventions and organization

For the purposes of this document, the terms and definitions given in *OIC Core Specification* and *OIC Resource Type Specification* apply.

## Notation

In this document, features are described as required, recommended, allowed or DEPRECATED as follows:

Required (or shall or mandatory).

These basic features shall be implemented to comply with OIC Core Architecture. The phrases “shall not”, and “PROHIBITED” indicate behavior that is prohibited, i.e. that if performed means the implementation is not in compliance.

Recommended (or should).

These features and functionality supported by OIC Core Architecture and should be implemented. Recommended features take advantage of the capabilities OIC Core Architecture, usually without imposing major increase of complexity. Notice that for compliance testing, if a recommended feature is implemented, it shall meet the specified requirements to be in compliance with these guidelines. Some recommended features could become requirements in the future. The phrase “should not” indicates behavior that is permitted but not recommended.

Allowed (or allowed).

These features are neither required nor recommended by OIC Core Architecture, but if the feature is implemented, it shall meet the specified requirements to be in compliance with these guidelines.

Conditionally allowed (CA)

The definition or behaviour depends on a condition. If the specified condition is met, then the definition or behaviour is allowed, otherwise it is not allowed.

Conditionally required (CR)

The definition or behaviour depends on a condition. If the specified condition is met, then the definition or behaviour is required. Otherwise the definition or behaviour is allowed as default unless specifically defined as not allowed.

DEPRECATED

Although these features are still described in this specification, they should not be implemented except for backward compatibility. The occurrence of a deprecated feature during operation of an implementation compliant with the current specification has no effect on the implementation’s operation and does not produce any error conditions. Backward compatibility may require that a feature is implemented and functions as specified but it shall never be used by implementations compliant with this specification.

Strings that are to be taken literally are enclosed in “double quotes”.

Words that are emphasized are printed in italic.

## Data types

See **Error! Reference source not found.**.

# Operational Scenarios

The overall goals are to:

1. Make UPnP Devices appear to OCF clients as if they were native OCF devices

## Creation

All resources defined in the *Resource Definitions* that are created and associated OCF-UPnP Bridge are required to have as a minimum the following link schema properties:

/upnpDeviceResUR

{

"rt": ["oic.r.upnp.device", "oic.wk.col"],

"if": ["oic.if.baseline"],

"ins": "uuid:UUID"

}

*Properties*

*URI*

Figure 3: Example of an OCF-UPnP Resource

“rt” and “if” are currently the only required properties defined by the *OIC Core Specification* Section 10. This needs to be expand in the *OCF 1.0 Core Specification* to support other core properties (i.e. “ins”). All OCF-UPnP devices and resources created are required to include “ins” property and populate with "uuid:UUID" with unique id of the UPnP Device. The use case for this with allow reduce the amount request and filtering by the OCF Client. This will allow for OCF Clients to communicate direct with specific UPnP device. This will be used in the discovery.

## Discovery

## Endpoint Discovery

Clients may discover Servers by using the mechanisms defined by the *OIC Core Specification* Section 10. A Client may populate an “rt” query parameter with the Device Types that the Client wants to discover, or if no “rt” query parameter is provided then the search is for all available Device Types irrespective.

UPnP Bridge Devices may be discovered by Device Type or implemented Resource Type. This difference is conveyed by the wanted “rt” argument of the *OIC Core Specification* discovery method (see section 11.3 of the *OIC Core Specification*).

The values that may be used for discovering a specific Device Type are listed in Table 12‑2 . The values that may be used to discover a specific Resource Type are listed in the *OIC Smart Home Device Specification* in section 6.

The discovery process provides the base URI of the Device that is acting as a Server to the Client. The structure of the detected Device can then be retrieved by Resource Discovery.

## Resource Discovery

Section intentionally left blank - Todo

# OCF-UPnP Bridge Device

This section describes the functionality of an OCF-UPnP Bridge Device; such a device is illustrated in Figure 4.

An OCF-UPnP Bridge Device is a device that represents one or more UPnP devices as OCF Devices on the network and/or represents one or more OCF Devices using another protocol on the network. The Bridged Devices themselves are out of the scope of this document. The only difference between a native OCF Device and a Virtual Bridged Device is how the device is encapsulated in an OCF-UPnP Bridge Device.

An OCF-UPnP Bridge Device shall be indicated on the network with a Device Type of “oic.d.bridge”. This provides to an OCF Client an explicit indication that the discovered Device is performing a bridging function. This is useful for a number of reasons; 1) when establishing a home network the Client can determine that the bridge is reachable and functional when no bridged devices are present, 2) allows for specific actions to be performed on the bridge taking into account the known functionality a bridge supports, 3) should the bridged devices be subject to a progressive reveal it enables user indications to be provided showing that sequence of discovery, 4) allows for explicit discovery of all devices that are serving a bridging function which benefits trouble shooting and maintenance actions on behalf of a user. When such a device is discovered the exposed Resources on the OCF-UPnP Bridge Device describe UPnP devices. For example, as shown in Figure 4 Overview of an OCF-UPnP Bridge Device bridging UPnP Devices.



Figure 4 Overview of an OCF-UPnP Bridge Device bridging UPnP Devices

It is expected that the OCF-UPnP Bridge Device discoveries the all UPnP Devices and create and/or translated them OCF devices during the start-up of the OCF-UPnP Bridge. The exposed set of devices can change as bridged devices are added or removed from the bridge. The adding and removing of bridged devices is implementation dependent. When an OCF-UPnP Bridge Device changes its set of exposed devices it shall notify any subscribed Clients

OCF-UPnP Bridge shall respond to network discovery commands on behalf of the exposed bridged devices. All bridged devices with all their Resources shall be discoverable via “/oic/res” of the OCF-UPnP Bridge Device. The Resources of bridged devices shall either be directly included as Links in “/oic/res” of the OCF-UPnP Bridge or as Links within a Collection (see Section 6.2 for application of a Device Type to a Collection Resource) that is itself directly included in “/oic/res”.

The response to a RETRIEVE on “/oic/res” shall only include the devices that match the RETRIEVE request.

The resource reference determined from each Link exposed by "/oic/res" on the OCF-UPnP Bridge shall be unique. The OCF-UPnP Bridge shall meet the requirements defined in the *OIC Core Specification* for population of the Properties and Link parameters in “/oic/res”.

The relationship between the OCF-UPnP Bridge and the devices exposed therein is indicated via the population of the “rel” element within the Link in “/oic/res”. The value “contains bridged” should be used when the bridged device is not part of the OCF-UPnP Bridge device. Conversely, when the bridge and embedded bridged devices share a common physical platform, the value “contains” should be used.

Figure 6 - Requesting all device types within an OCF-UPnP Bridge provides one possible instantiation of an OCF-UPnP Bridge. Note the use of the Device Type for an OCF-UPnP Bridge and the Device Type of the exposed devices that are modelled leveraging the Collection semantics defined by the *OIC Core Specification*. This is for illustrative purposes only.

@startuml

title Detecting devices and resources on an OCF-UPnP Bridge

hide footbox

note over oicclient

request OCF UPnP Bridge Device

end note

oicclient -> oicserver: get [/oic/res?rt=oic.d.bridge"]

note right oicserver

[

{"di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",

"links": [

{ "href": "/oic/d",

"rt": ["oic.d.bridge","oic.wk.d"],

"if": "oic.if.baseline",

"rel": "hosts"

}

]

}

]

end note

oicserver -> oicclient: “string”

note over oicclient

request light devices

end note

oicclient -> oicserver: get [/oic/res?rt=oic.d.light"]

note right oicserver

[

{"di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",

"links": [

{ "href": "/myLightDevice",

"rt": ["oic.wk.col","oic.d.light"],

"if": ["oic.if.ll","oic.if.baseline"],

"rel": "contains bridged"

}

]

}

]

end note

oicserver -> oicclient: “string"

@enduml

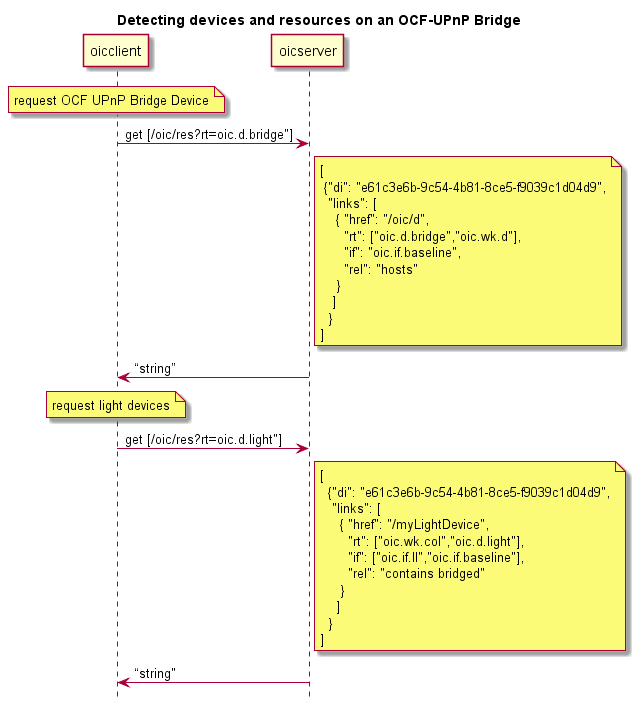


Figure 5 - Requesting Bridge and Light devices within an OCF-UPnP Bridge

Figure 5 - Requesting Bridge and Light devices within an OCF-UPnP Bridge provides one possible instantiation of an OCF-UPnP Bridge and light. Note the use of the Device Type for an OCF-UPnP Bridge and the Device Type of the exposed devices that are modelled leveraging the Collection semantics defined by the *OIC Core Specification*. This is for illustrative purposes only.

@startuml

title Detecting devices and resources on an OCF-UPnP Bridge

hide footbox

note over oicclient

request OCF UPnP Bridge Device

end note

oicclient -> oicserver: get [/oic/res?rt=oic.d.bridge"]

note right oicserver

[

{"di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",

"links": [

{ "href": "/oic/d",

"rt": ["oic.d.bridge","oic.wk.d"],

"if": "oic.if.baseline",

"rel": "hosts"

}

]

}

]

end note

oicserver -> oicclient: “string”

note over oicclient

request all devices

end note

oicclient -> oicserver: get [/oic/res"]

note right oicserver

[

{"di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",

"links": [

{ "href": "/oic/d",

"rt": ["oic.d.bridge","oic.wk.d"],

"if": "oic.if.baseline",

"rel": "hosts"

},

{ "href": "/myLightDevice",

"rt": ["oic.wk.col","oic.d.light"],

"if": ["oic.if.ll","oic.if.baseline"],

"rel": "contains bridged"

},

{ "href": "/myLightSwitch",

"rt": [“oic.wk.col”,"oic.d.switch"],

"if": ["oic.if.ll","oic.if.baseline"],

"rel": "contains bridged"

}

]

}

]

end note

oicserver -> oicclient: "string"

@enduml

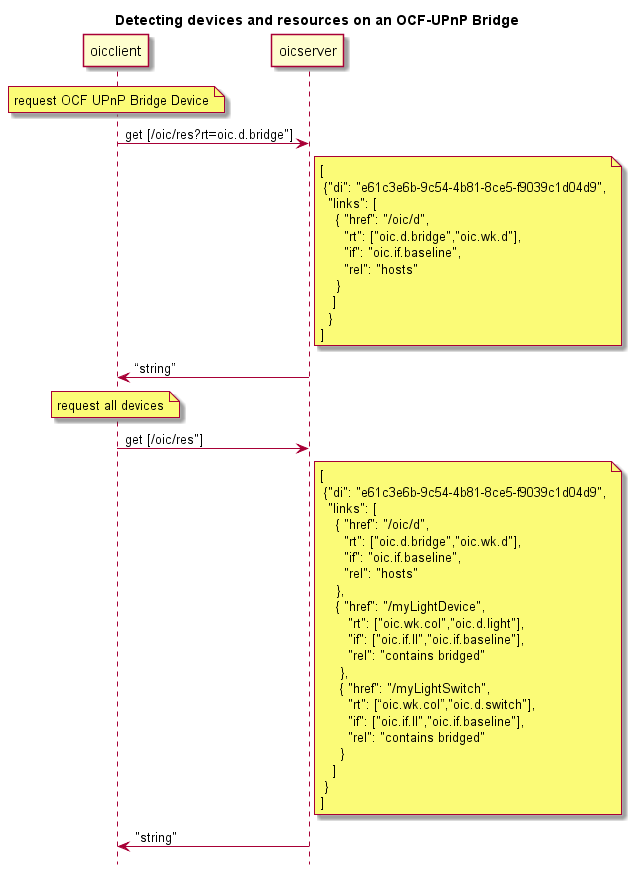


Figure 6 - Requesting all device types within an OCF-UPnP Bridge

# Introspection

Whenever possible, the translation code should make use of metadata available that indicates what the sender and recipient of the message in question are expecting. For example, devices certified by the *OCF* *UPnP Certification Testing* are required to carry the introspection data for each service, action and stateVariable they expose (see *UPnP Specifications-Standards: Device Control Protocols*). The *OIC Core Specification* makes no such requirement, but the *OCF 1.0 Core Specification* will. When the metadata is available, translators should convert the incoming payload to exactly the format expected by the recipient and should use information when translating replies to form a more useful message.

# Security

UPnP Bridge Device shall implement the mandated security Resources specified in OIC Security Specification. Additionally UPnP Bridge Device shall secure all links used to access Resources using DTLS

The general problem is how to ensure that devices and controllers in OCF and UPnP contexts to not open significant security risks when connected via a bridge. The process of looking at this problem in the context of OCF and UPnP will be instructive in the more general problem of connecting any ecosystem to OCF through a bridge.

Here are some attack vectors to consider:

* UPnP control points might get unauthorized access to OCF servers and do bad stuff.
* OCF clients might get unauthorized access to UPnP devices and do similarly bad stuff.
* Bad actors may get access to a compromised bridge and wreak havoc all around.
* Communication channels might be compromised and rogue messages inserted.
* Communications might be vulnerable to DOS attacks.

This is just some security risks this will need to review by the Security Work Group.

UPnP has a security service called DeviceProtection that allows for x.509 authentication and secure communications over TLS. DeviceProtection is optional for UPnP V1.0 and V1.1 and required UPnP V2.0. While many existing UPnP implementations have chosen **not** to implement the DeviceProtection service and of course it is available for use when implemented. Based on that, it could argue the following points:

* UPnP devices and control points that implement DeviceProtection have adequate security to be used in the IoT contexts addressed by OCF. In other words, they can be authenticated, have levels of authorization associated with different roles, and communicate with each other over trusted secure channels.
* UPnP devices that do not implement DeviceProtection should not be trusted. They should only be used in contexts where that trust is not a requirement or confirmed by the Application/End-user.

Therefore, it suggested the following general policy for bridging to UPnP (and similar no-secure ecosystems):

* OCF is responsible for all aspects of security on the OCF side of the bridge.
* OCF is responsible for the security of the bridge itself.
* If an UPnP control point does not implement UPnP DeviceProtection then the bridge shall limit its access and authorization in the OCF domain to contexts that don’t require security (if there are any). An UPnP “control point” is any device that can send UPnP control point messages regardless of whether it has UPnP device features.
* If an UPnP device does not implement UPnP DeviceProtection, OCF does not necessarily need to do anything. The OCF side of the bridge must not trust the device, but the insecurity of the UPnP side of the bridge is not expected to be repaired by OCF.
* If a UPnP device or control point implements DeviceProtection (and is certified), and if it has OCF-trusted credentials, it may be trusted to act as a full participant in the OCF ecosystem. (NOTE: UPnP DeviceProtection allows for self-signed certificates. OCF policy must decide whether to trust these certificates

# Device Type Mapping

## Introduction

This Section contains the mappings of to/from Device Types.

## UPnP Device Types to OCF Device Types

The following table captures the equivalency mapping between UPnP Device Types defined OCF Device Types (see OCF UPnP Specifications-Standards: Device Control Protocols) and defined OCF Device Types (see Table 10-1 in OIC Smart Home Device Specification). The minimum Resource sets for each OCF Device is provided in *OIC Resource Type Specification*.

Table 9‑1 UPnP Device Types to OCF Device Types Mapping.

|  |  |  |
| --- | --- | --- |
| Classification | UPnP Device Types | OCF Device Types |
| Lighting | BinaryLight:1 | oic.d.light |
| DimmableLight:1 |

# Resource Definitions

This section contains definitions for all OCF-UPnP Resource Types; the complete set is listed in ***Table 10‑1*** *UPnP Service Types to OCF Resource Types Mapping* and **Table 10‑2** *OCF-UPnP Resource Summary*. All sections provide example representations of the Resource Type following the application of the default interface that is applied for that specific Resource Type.

All Resource Types shall be created in accordance with the *OIC Core Specification* Section 7.2. All comparisons against a Resource Type shall be case insensitive.

All Resource Types in this document are prefixed with “oic.r” denoting that it is an OIC defined Resource Type.

## Introduction

This Section contains the mappings of to/from Resource Types.

## UPnP Service Types to OCF Resource Types

The following table captures the equivalency mapping between UPnP defined Service Types (see *UPnP Specifications-Standards: Device Control Protocols*) and OCF defined Resource Types (see Table 10-1 in *OIC Smart Home Device Specification*). The minimum Resource sets for each OCF Device is provided in *OIC Resource Type Specification*.

Table 10‑1 UPnP Service Types to OCF Resource Types Mapping

|  |  |  |
| --- | --- | --- |
| Classification | UPnP Service Types | OCF Resource Types |
| Lighting | SwitchPower:1 | oic.r.switch.binary |
| Dimming:1 | oic.r.switch.binary |
| oic.r.light.brightness |

Table 10‑2 OCF-UPnP Resource Summary

|  |  |  |
| --- | --- | --- |
| Friendly Name | Resource Type | Section |
| UPnP Device | oic.r.upnp.device | 12.3 |
| UPnP Service | oic.r.upnp.service | 12.5 |
| UPnP Action | oic.r.upnp.action | 12.7 |
| UPnP State Variable | oic.r.upnp.stateVariable | 12.6 |
| Generic | oic.r.generic | 12.8 |
| Icon | oic.r.icon | 12.4 |
|  |  | <link> |