OCF Core Specification

VERSION 2.0.2 | April 2019

CONTACT **admin@openconnectivity.org**

Legal Disclaimer

CONTENTS

Figures

Tables

1 Scope

The OCF specifications are divided into two sets of documents:

– Core Specification documents: The Core Specification documents specify the Framework, i.e., the OCF core architecture, interfaces, protocols and services to enable OCF profiles implementation for Internet of Things (IoT) usages and ecosystems.

– Vertical Domain Specification documents: The Vertical Domain Specification documents specify OCF Device profiles to enable IoT usages for different vertical market segments such as smart home, industrial, healthcare, and automotive. They also specify Resource definitions to enable vertical services and use case. Such specifications include ISO/IEC 30118-5:2018 which is built upon the interfaces and network security of the OCF core architecture defined in the Core Specification.

This document is the OCF Core specification which specifies the Framework and core architecture.

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.

ISO 8601, *Data elements and interchange formats – Information interchange –Representation of dates and times*, International Standards Organization, December 3, 2004

ISO/IEC DIS 20924, *Information Technology – Internet of Things – Vocabulary*, June 2018
https://www.iso.org/standard/69470.html

ISO/IEC 30118-2:2018, Information technology – Open Connectivity Foundation (OCF) Specification – Part 2: Security specification
https://www.iso.org/standard/74239.html
Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

ISO/IEC 30118-5:2018, Information technology – Open Connectivity Foundation (OCF) Specification – Part 5: Smart home device specification
https://www.iso.org/standard/74242.html
Latest version available at: https://openconnectivity.org/specs/OCF_Device_Specification.pdf

OCF Easy Wi-Fi Setup, Information technology – Open Connectivity Foundation (OCF) Specification – Part 7: Wi-Fi Easy Setup specification
Latest version available at: https://openconnectivity.org/specs/OCF_Wi-Fi_Easy_Setup_Specification_v2.0.1.pdf

IETF RFC 768, *User Datagram Protocol*, August 1980
https://www.rfc-editor.org/info/rfc768

IETF RFC 3339, *Date and Time on the Internet: Timestamps*, July 2002
https://www.rfc-editor.org/info/rfc3339

IETF RFC 3986, *Uniform Resource Identifier (URI): General Syntax, January 2005*.
https://www.rfc-editor.org/info/rfc3986

IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*, July 2005
https://www.rfc-editor.org/info/rfcfse4122

IETF RFC 4287, *The Atom Syndication Format*, December 2005,
https://www.rfc-editor.org/info/rfc4287

IETF RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, September 2007
https://www.rfc-editor.org/info/rfc4941

IETF RFC 5424, *The Syslog Protocol*, March 2009
https://tools.ietf.org/html/rfc5424IETF RFC 5646, *Tags for Identifying Languages*, September 2009
https://www.rfc-editor.org/info/rfc5646

IETF RFC 5988, *Web Linking: General Syntax, October 2010*
https://www.rfc-editor.org/info/rfc5988

IETF RFC 6347, *Datagram Transport Layer Security Version 1.2*, January 2012
https://www.rfc-editor.org/info/rfc6347

IETF RFC 6434, *IPv6 Node Requirements*, December 2011
https://www.rfc-editor.org/info/rfc6434

IETF RFC 6573, *The Item and Collection Link Relations*, April 2012
https://www.rfc-editor.org/info/rfc6573

IETF RFC 6690, *Constrained RESTful Environments (CoRE) Link Format*, August 2012
https://www.rfc-editor.org/info/rfc6690

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

IETF RFC 7084, *Basic Requirements for IPv6 Customer Edge Routers*, November 2013
https://www.rfc-editor.org/info/rfc7084

IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014
https://www.rfc-editor.org/info/rfc7159

IETF RFC 7252, *The Constrained Application Protocol (CoAP)*, June 2014
https://www.rfc-editor.org/info/rfc7252

IETF RFC 7301, *Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension*, July 2014
https://www.rfc-editor.org/info/rfc7301

IETF RFC 7595, *Guidelines and Registration Procedures for URI Schemes*, June 2015
https://www.rfc-editor.org/info/rfc7595

IETF RFC 7641, *Observing Resources in the Constrained Application Protocol (CoAP)*, September 2015
https://www.rfc-editor.org/info/rfc7641

IETF RFC 7721, *Security and Privacy Considerations for IPv6 Address Generation Mechanisms*, March 20016
https://www.rfc-editor.org/info/rfc7721

IETF RFC 7959, *Block-Wise Transfers in the Constrained Application Protocol (CoAP)*, August 2016
https://www.rfc-editor.org/info/rfc7959

IETF RFC 8075, *Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)*, February 2017
https://www.rfc-editor.org/info/rfc8075

IETF RFC 8323, *CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets*, February 2018
https://www.rfc-editor.org/info/rfc8323

IANA IPv6 Multicast Address Space Registry
http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml

IANA Media Types Assignment, March 2017
http://www.iana.org/assignments/media-types/media-types.xhtml

IANA Link Relations, October 2017
http://www.iana.org/assignments/link-relations/link-relations.xhtml

JSON Schema Validation, *JSON Schema: interactive and non-interactive validation*, January 2013
http://json-schema.org/draft-04/json-schema-validation.html

OpenAPI specification, *fka Swagger RESTful API Documentation Specification*, Version 2.0
https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md

3 Terms, definitions, and abbreviated terms

3.1 Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

– ISO Online browsing platform: available at https://www.iso.org/obp.

– IEC Electropedia: available at http://www.electropedia.org/.

3.1.1
Alert
information provided by the Device (3.1.10) by means of a specialised Resource Type (3.1.32) that provides details with regard to potential problems, issues, or Device (3.1.10) status of interest on which action can be taken

3.1.2
Atomic Measurement
a design pattern that ensures that the Client (3.1.3) can only access the Properties (3.1.31) of linked Resources (3.1.28) atomically, that is as a single group

3.1.3
Client
a logical entity that accesses a Resource (3.1.28) on a Server (3.1.38)

3.1.4
Collection
a Resource (3.1.28) that contains zero or more Links (3.1.18)

3.1.5
Common Properties
Properties (3.1.31) specified for all Resources (3.1.28)

3.1.6
Composite Device
a Device (3.1.10) that is modelled as multiple Device Types (3.1.11); with each component Device Type (3.1.11) being exposed as a Collection (3.1.4)

3.1.7
Configuration Source
a cloud or service network or a local read-only file which contains and provides configuration related information to the Devices (3.1.10)

3.1.8
Core Resources
those Resources (3.1.28) that are defined in this document

3.1.9
Default OCF Interface
an OCF Interface (3.1.15) used to generate the response when an OCF Interface (3.1.15) is omitted in a request

3.1.10
Device
a logical entity that assumes one or more roles, e.g., Client (3.1.3), Server (3.1.38)

Note 1 to entry: More than one Device (3.1.10) can exist on a Platform (3.1.27).

3.1.11
Device Type
a uniquely named definition indicating a minimum set of Resource Types (3.1.32) that a Device (3.1.10) supports

Note 1 to entry: A Device Type (3.1.11) provides a hint about what the Device (3.1.10) is, such as a light or a fan, for use during Resource (3.1.28) discovery.

3.1.12
Discoverable Resource
a Resource (3.1.28) that is listed in "/oic/res"

3.1.13
OCF Endpoint
entity participating in the OCF protocol, further identified as the source or destination of a request and response messages for a given Transport Protocol Suite

Note 1 to entry: Example of a Transport Protocol Suite would be CoAP over UDP over IPv6.

3.1.14
Framework
a set of related functionalities and interactions defined in this document, which enable interoperability across a wide range of networked devices, including IoT

3.1.15
OCF Interface
interface description in accordance with IETF RFC 6690 and as defined by OCF that provides a view to and permissible responses from a Resource (3.1.28)

3.1.16
Introspection
mechanism to determine the capabilities of the hosted Resources (3.1.28) of a Device (3.1.10)

3.1.17
Introspection Device Data (IDD)
data that describes the payloads per implemented method of the Resources (3.1.28) that make up the Device (3.1.10)

Note 1 to entry: See 11.8 for all requirements and exceptions.

659 **3.1.18**
660 **Links**
661 extends typed web links accorfiding to IETF RFC 5988

662 **3.1.19**
663 **Non-Discoverable Resource**
664 a Resource (3.1.28) that is not listed in "/oic/res"

665 Note 1 to entry: The Resource (3.1.28) can be reached by a Link (3.1.18) which is conveyed by another Resource
666 (3.1.28). For example a Resource (3.1.28) linked in a Collection (3.1.4) does not have to be listed in "/oic/res", since
667 traversing the Collection (3.1.4) would discover the Resource (3.1.28) implemented on the Device (3.1.10).

668 **3.1.20**
669 **Non-OCF Device**
670 a Device (3.1.10) which does not comply with the OCF Device (3.1.10) requirements

671 **3.1.21**
672 **Notification**
673 the mechanism to make a Client (3.1.3) aware of state changes in a Resource (3.1.28)

674 **3.1.22**
675 **Observe**
676 the act of monitoring a Resource (3.1.28) by sending a RETRIEVE operation which is cached by
677 the Server (3.1.38) hosting the Resource (3.1.28) and reprocessed on every change to that
678 Resource (3.1.28)

679 **3.1.23**
680 **OpenAPI 2.0**
681 Resource (3.1.28) and Intropection Device Data (3.1.17) definitions used in this document as
682 defined in the OpenAPI specification

683 **3.1.24**
684 **Parameter**
685 an element that provides metadata about a Resource (3.1.28) referenced by the target URI of a
686 Link (3.1.18)

687 **3.1.25**
688 **Partial UPDATE**
689 an UPDATE operation to a Resource (3.1.28) that includes a subset of the Properties (3.1.31) that
690 are visible via the OCF Interface (3.1.15) being applied for the Resource Type (3.1.32)

691 **3.1.26**
692 **Physical Device**
693 the physical thing on which a Device(s) (3.1.10) is exposed

694 **3.1.27**
695 **Platform**
696 a Physical Device (3.1.26) containing one or more Devices (3.1.10)

697 **3.1.28**
698 **Resource**
699 represents an entity modelled and exposed by the Framework (3.1.14)

700 **3.1.29**
701 **Resource Directory**
702 a set of descriptions of Resources (3.1.28) where the actual Resources (3.1.28) are held on Servers
703 (3.1.38) external to the Device (3.1.10) hosting the Resource Directory (3.1.29), allowing lookups
704 to be performed for those Resources (3.1.28)

Note 1 to entry: This functionality can be used by sleeping Servers (3.1.38) or Servers (3.1.38) that choose not to listen/respond to multicast requests directly.

3.1.30
Resource Interface
a qualification of the permitted requests on a Resource (3.1.28)

3.1.31
Property
a significant aspect or Parameter (3.1.24) of a Resource (3.1.28), including metadata, that is exposed through the Resource (3.1.28)

3.1.32
Resource Type
a uniquely named definition of a class of Properties (3.1.31) and the interactions that are supported by that class

Note 1 to entry: Each Resource (3.1.28) has a Property (3.1.31) "rt" whose value is the unique name of the Resource Type (3.1.32).

3.1.33
Scene
a static entity that stores a set of defined Property (3.1.31) values for a Collection (3.1.4) of Resources (3.1.28)

Note 1 to entry: A Scene (3.1.33) is a prescribed setting of a set of Resources (3.1.28) with each having a predetermined value for the Property (3.1.31) that has to change.

3.1.34
Scene Collection
a Collection (3.1.4) that contains an enumeration of possible Scene Values (3.1.36) and the current Scene Value (3.1.36)

Note 1 to entry: The member values of the Scene Collection (3.1.34) are Scene Members (3.1.35).

3.1.35
Scene Member
a Resource (3.1.28) that contains mappings of Scene Values (3.1.36) to values of a Property (3.1.31) in the Resource (3.1.28)

3.1.36
Scene Value
a Scene (3.1.33) enumerator representing the state in which a Resource (3.1.28) can be

3.1.37
Secure OCF Endpoint
an OCF Endpoint (3.1.13) with a secure connection (e.g., CoAPS)

3.1.38
Server
a Device (3.1.10) with the role of providing Resource (3.1.28) state information and facilitating remote interaction with its Resources (3.1.28)

Note 1 to entry: A Server (3.1.38) can be implemented to expose Non-OCF Device (3.1.20) resources to Clients (3.1.3) (see 5.6).

3.1.39
Unsecure OCF Endpoint
an OCF Endpoint () with an unsecure connection (e.g., CoAP)

3.1.40
Vertical Resource Type
a Resource Type (3.1.32) in a vertical domain specification

Note 1 to entry: An example of a Vertical Resource Type (3.1.40) would be "oic.r.switch.binary".

3.2 Abbreviated terms

3.2.1
ACL
Access Control List

Note 1 to entry: The details are defined in ISO/IEC 30118-2:2018.

3.2.2
BLE
Bluetooth Low Energy

3.2.3
CBOR
Concise Binary Object Representation

3.2.4
CoAP
Constrained Application Protocol

3.2.5
CoAPS
Secure Constrained Application Protocol

3.2.6
DTLS
Datagram Transport Layer Security

Note 1 to entry: The details are defined in IETF RFC 6347.

3.2.7
EXI
Efficient XML Interchange

3.2.8
IP
Internet Protocol

3.2.9
IRI
Internationalized Resource Identifiers

3.2.10
ISP
Internet Service Provider

3.2.11
JSON
JavaScript Object Notation

3.2.12
mDNS
Multicast Domain Name Service

3.2.13
MTU
Maximum Transmission Unit

3.2.14
NAT
Network Address Translation

3.2.15
OCF
Open Connectivity Foundation

the organization that created this document

3.2.16
REST
Representational State Transfer

3.2.17
RESTful
REST-compliant Web services

3.2.18
UDP
User Datagram Protocol

Note 1 to entry: The details are defined in IETF RFC 768.

3.2.19
URI
Uniform Resource Identifier

3.2.20
URN
Uniform Resource Name

3.2.21
UTC
Coordinated Universal Time

3.2.22
UUID
Universal Unique Identifier

3.2.23
XML
Extensible Markup Language

4 Document conventions and organization

4.1 Conventions

In this documenta 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.

4.2 Notation

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

Required (or shall or mandatory)(M).

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

Recommended (or should)(S).

– These features add functionality supported by Core Architecture and should be implemented. Recommended features take advantage of the capabilities 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 behaviour that is permitted but not recommended.

Allowed (may or allowed)(O).

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

DEPRECATED.

– Although these features are still described in this document, they should not be implemented except for backward compatibility. The occurrence of a deprecated feature during operation of an implementation compliant with the current documenthas 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 document.

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.

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

Words that are emphasized are printed in italic.

In all of the Property and Resource definition tables that are included throughout this document the "Mandatory" column indicates that the item detailed is mandatory to implement; the mandating of inclusion of the item in a Resource Payload associated with a CRUDN action is dependent on the applicable schema for that action.

4.3 Data types

Resources are defined using data types derived from JSON values as defined in IETF RFC 7159. However, a Resource can overload a JSON defined value to specify a particular subset of the JSON value, using validation keywords defined in JSON Schema Validation.

Among other validation keywords, clause 7 in JSON Schema Validation defines a "format" keyword with a number of format attributes such as "uri" and "date-time", and a "pattern" keyword with a regular expression that can be used to validate a string. This clause defines patterns that are available for use in describing OCF Resources. The pattern names can be used in documenttext where JSON format names can occur. The actual JSON schemas shall use the JSON type and pattern instead.

For all rows defined in Table 1, the JSON type is string.

Table 1 – Additional OCF Types

Pattern Name	Pattern	Description
"csv"	<none>	A comma separated list of values encoded within a string. The value type in the csv is described by the Property where the csv is used. For example a csv of integers. NOTE csv is considered deprecated and an array of strings should be used instead for new Resources.
"date"	^([0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|2[0-9]|1[0-9]|0[1-9])$	The full-date format pattern according to IETF RFC 3339
"duration"	^(P(?!$)([0-9]+Y)?([0-9]+M)?([0-9]+W)?([0-9]+D)?((T(?=[0-9]+[HMS])([0-9]+H)?([0-9]+M)?([0-9]+S)?)?))$|^(P[0-9]+W)$\|^(P[0-9]{4})-(1[0-2]\|0[1-9])-(3[0-1]\|2[0-9]\|1[0-9]\|0[1-9])T(2[0-3]\|1[0-9]\|0[1-9]):([0-5][0-9]):([0-5][0-9])$|^(P[0-9]{4})(1[0-2]|0[1-9])(3[0-1]|2[0-9]|1[0-9]|0[1-9])T(2[0-3]|1[0-9]|0[1-9])([0-5][0-9])([0-5][0-9])$	A string representing duration formatted as defined in ISO 8601. Allowable formats are: P[n]Y[n]M[n]DT[n]H[n]M[n]S, P[n]W, P[n]Y[n]-M[n]-DT[0-23]H[0-59]:M[0-59]:S, and P[n]W, P[n]Y[n]M[n]DT[0-23]H[0-59]M[0-59]S. P is mandatory, all other elements are optional, time elements must follow a T.
"int64"	^0|(-?[1-9][0-9]{0,18})$	A string instance is valid against this attribute if it contains an integer in the range [-(2**63), (2**63)-1] NOTE IETF RFC 7159 clause 6 explains that JSON integers outside the range [-(2**53)+1, (2**53)-1] are not interoperable and so JSON numbers cannot be used for 64-bit numbers.
"language-tag"	^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$	An IETF language tag formatted according to IETF RFC 5646 clause 2.1.
"uint64"	^0|([1-9][0-9]{0,19})$	A string instance is valid against this attribute if it contains an integer in the range [0, (2**64)-1] Also see note for "int64"
"uuid"	^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$	A UUID string representation formatted according to IETF RFC 4122 clause 3.

Strings shall be encoded as UTF-8 unless otherwise specified.

In a JSON schema, "maxLength" for a string indicates the maximum number of characters not octets. However, "maxLength" shall also indicate the maximum number of octets. If no "maxLength" is defined for a string, then the maximum length shall be 64 octets.

5 Architecture

5.1 Overview

The architecture enables resource based interactions among IoT artefacts, i.e. physical devices or applications. The architecture leverages existing industry standards and technologies and provides solutions for establishing connections (either wireless or wired) and managing the flow of information among Devices, regardless of their form factors, operating systems or service providers.

Specifically, the architecture provides:

– A communication and interoperability framework for multiple market segments (Consumer, Enterprise, Industrial, Automotive, Health, etc.), OSs, platforms, modes of communication, transports and use cases.

– A common and consistent model for describing the environment and enabling information and semantic interoperability.

– Common communication protocols for discovery and connectivity.

– Common security and identification mechanisms.

– Opportunity for innovation and product differentiation.

– A scalable solution addressing different Device capabilities, applicable to smart devices as well as the smallest connected things and wearable devices.

The architecture is based on the Resource Oriented Architecture design principles and described in the 5.2 through 5.6 respectively. 5.2 presents the guiding principles for OCF operations. 5.3 defines the functional block diagram and Framework. 5.5 provides an example scenario with roles. 5.6 provides an example scenario of bridging to non- OCF ecosystem.

5.2 Principle

In the architecture, Entities in the physical world (e.g., temperature sensor, an electric light or a home appliance) are represented as Resources. Interactions with an entity are achieved through its Resource representations (see 7.7) using operations that adhere to Representational State Transfer (REST) architectural style, i.e., RESTful interactions.

The architecture defines the overall structure of the Framework as an information system and the interrelationships of the Entities that make up OCF. Entities are exposed as Resources, with their unique identifiers (URIs) and support interfaces that enable RESTful operations on the Resources. Every RESTful operation has an initiator of the operation (the Client) and a responder to the operation (the Server). In the Framework, the notion of the Client and Server is realized through roles (see 5.5). Any Device can act as a Client and initiate a RESTful operation on any Device acting as a Server. Likewise, any Device that exposes Entities as Resources acts as a Server. Conformant to the REST architectural style, each RESTful operation contains all the information necessary to understand the context of the interaction and is driven using a small set of generic operations, i.e., CREATE, RETRIEVE, UPDATE, DELETE and NOTIFY (CRUDN) defined in clause 8, which include representations of Resources.

Figure 1 depicts the architecture.

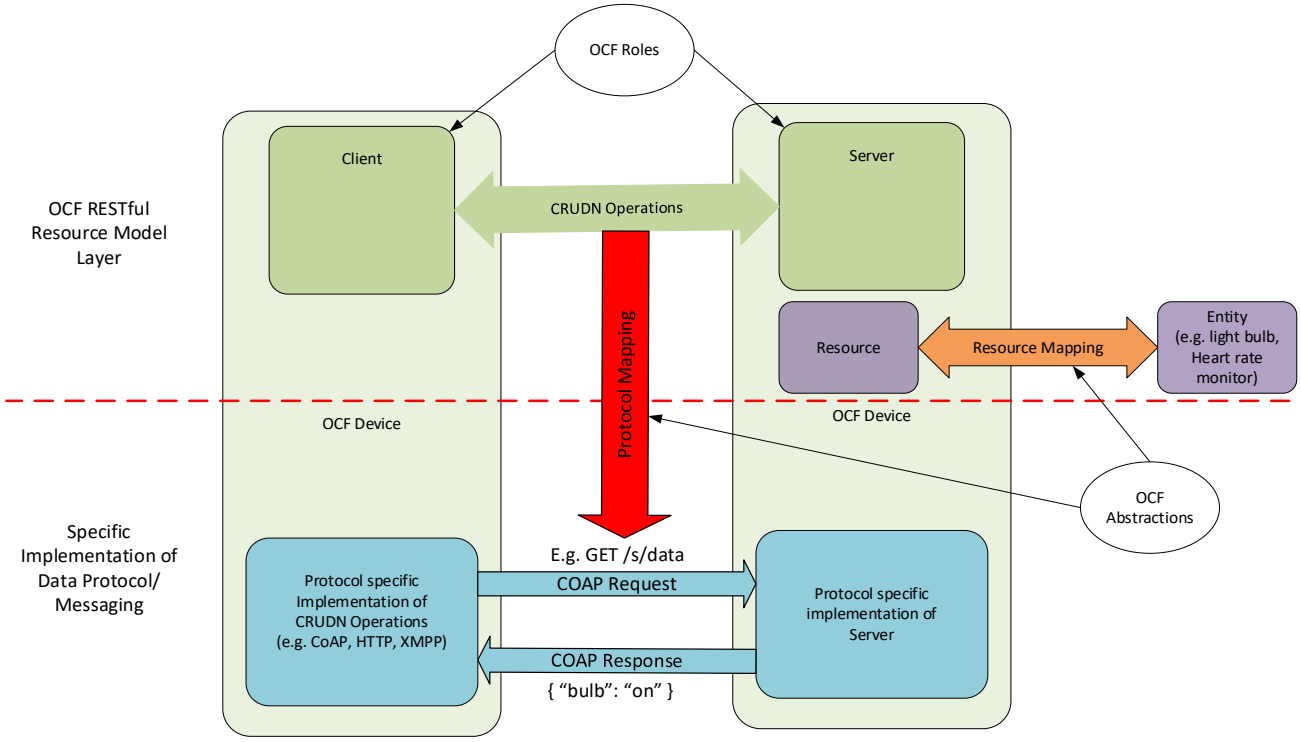


Figure 1 – Architecture - concepts

The architecture is organized conceptually into three major aspects that provide overall separation of concern: Resource model, RESTful operations and abstractions.

– Resource model: The Resource model provides the abstractions and concepts required to logically model, and logically operate on the application and its environment. The Core Resource model is common and agnostic to any specific application domain such as smart home, industrial or automotive. For example, the Resource model defines a Resource which abstracts an entity and the representation of a Resource maps the entity's state. Other Resource model concepts can be used to model other aspects, for example behaviour.

– RESTful operations: The generic CRUDN operations are defined using the RESTful paradigm to model the interactions with a Resource in a protocol and technology agnostic way. The specific communication or messaging protocols are part of the protocol abstraction and mapping of Resources to specific protocols is provided in 11.8.

– Abstraction: The abstractions in the Resource model and the RESTful operations are mapped to concrete elements using abstraction primitives. An entity handler is used to map an entity to a Resource and connectivity abstraction primitives are used to map logical RESTful operations to data connectivity protocols or technologies. entity handlers may also be used to map Resources to Entities that are reached over protocols that are not natively supported by OCF.

5.3 Functional block diagram

The functional block diagram encompasses all the functionalities required for operation. These functionalities are categorized as L2 connectivity, networking, transport, Framework, and application profiles. The functional blocks are depicted in Figure 2.

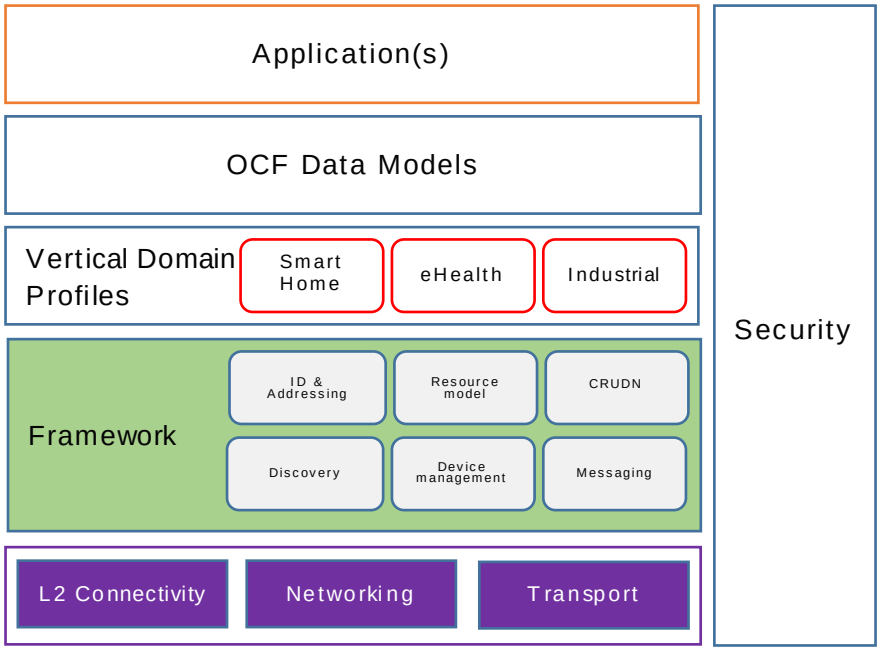


Figure 2 – Functional block diagram

– *L2 connectivity*: Provides the functionalities required for establishing physical and data link layer connections (e.g., Wi-Fi™ or Bluetooth® connection) to the network.

– *Networking*: Provides functionalities required for Devices to exchange data among themselves over the network (e.g., Internet).

– *Transport*: Provides end-to-end flow transport with specific QoS constraints. Examples of a transport protocol include TCP and UDP or new Transport protocols under development in the IETF, e.g., Delay Tolerant Networking (DTN).

– *Framework*: Provides the core functionalities as defined in this document. The functional block is the source of requests and responses that are the content of the communication between two Devices.

– *Vertical Domain profile*: Provides market segment specific functionalities, e.g., functions for the smart home market segment.

When two Devices communicate with each other, each functional block in a Device interacts with its counterpart in the peer Device as shown in Figure 3.

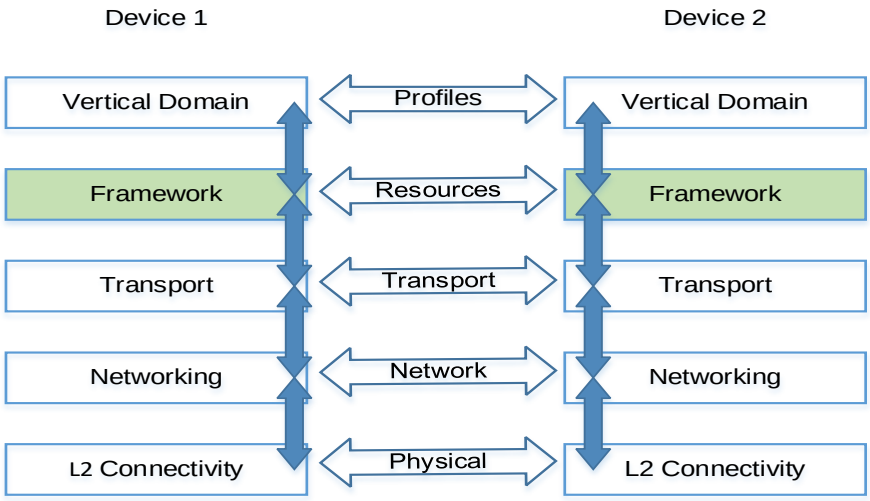


Figure 3 – Communication layering model

5.4 Framework

Framework consists of functions which provide core functionalities for operation.

- *Identification and addressing.* Defines the identifier and addressing capability. The Identification and addressing function is defined in clause 6.
- *Discovery*. Defines the process for discovering available.
  - Devices (OCF Endpoint Discovery in clause 10) and
  - Resources (Resource discovery in 11.3).
- *Resource model*. Specifies the capability for representation of entities in terms of Resources and defines mechanisms for manipulating the Resources. The Resource model function is defined in clause 7.
- *CRUDN*. Provides a generic scheme for the interactions between a Client and Server as defined in clause 8.
- *Messaging*. Provides specific message protocols for RESTful operation, i.e. CRUDN. For example, CoAP is a primary messaging protocol. The messaging function is defined in 12.
- *Device management***.** Specifies the discipline of managing the capabilities of a Device, and includes Device provisioning and initial setup as well as Device monitoring and diagnostics. The Device management function is defined in 11.5.
- *Security***.** Includes authentication, authorization, and access control mechanisms required for secure access to Entities. The security function is defined in clause 13.

5.5 Example Scenario with roles

Interactions are defined between logical entities known as roles. Three roles are defined: Client, Server and Intermediary.

Figure 4 illustrates an example of the roles in a scenario where a smart phone sends a request message to a thermostat; the original request is sent over HTTP, but is translated into a CoAP request message by a gateway in between, and then delivered to the thermostat. In this example, the smart phone takes the role of a Client, the gateway takes the role of an Intermediary and the thermostat takes the role of a Server.

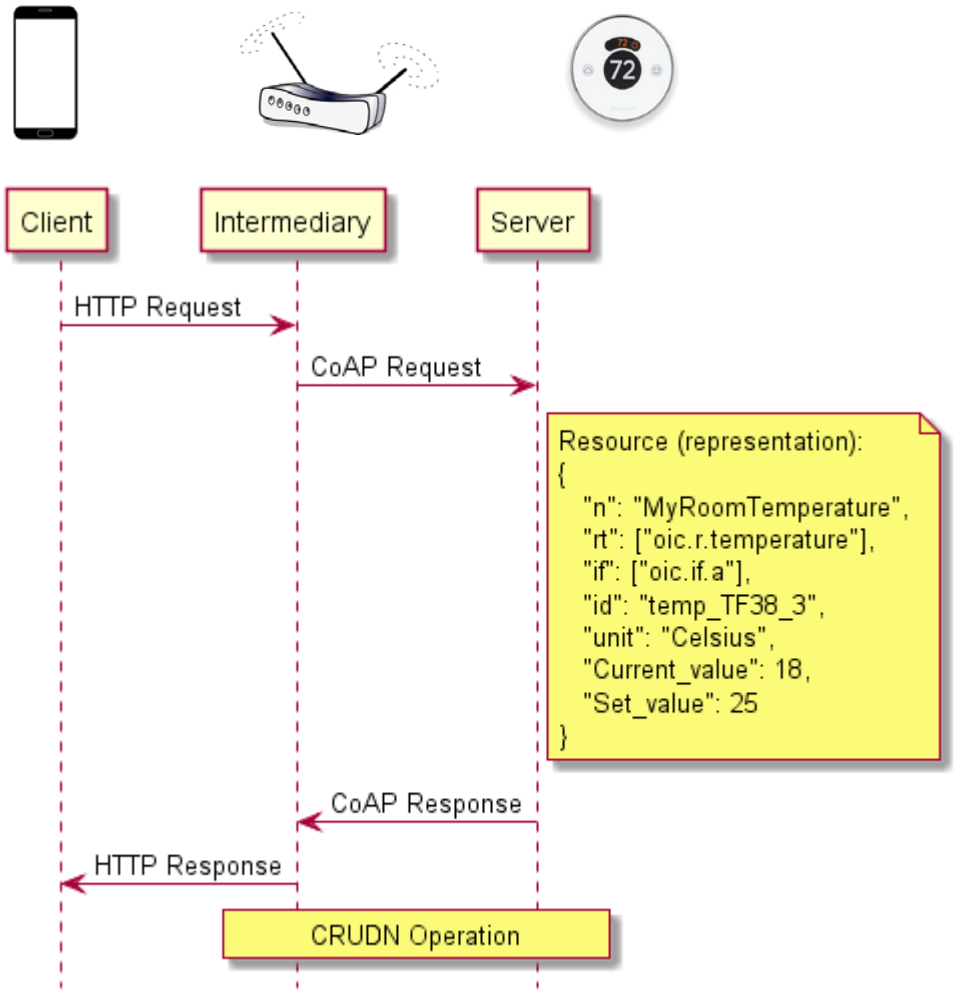


Figure 4 – Example illustrating the roles

5.6 Example Scenario: Bridging to Non- OCF ecosystem

The use case for this scenario is a display (like a wrist watch) that is used to monitor a heart rate sensor that implements a protocol that is not OCF supported.

Figure 5 provides a detailed logical view of the concepts described in Figure 1.

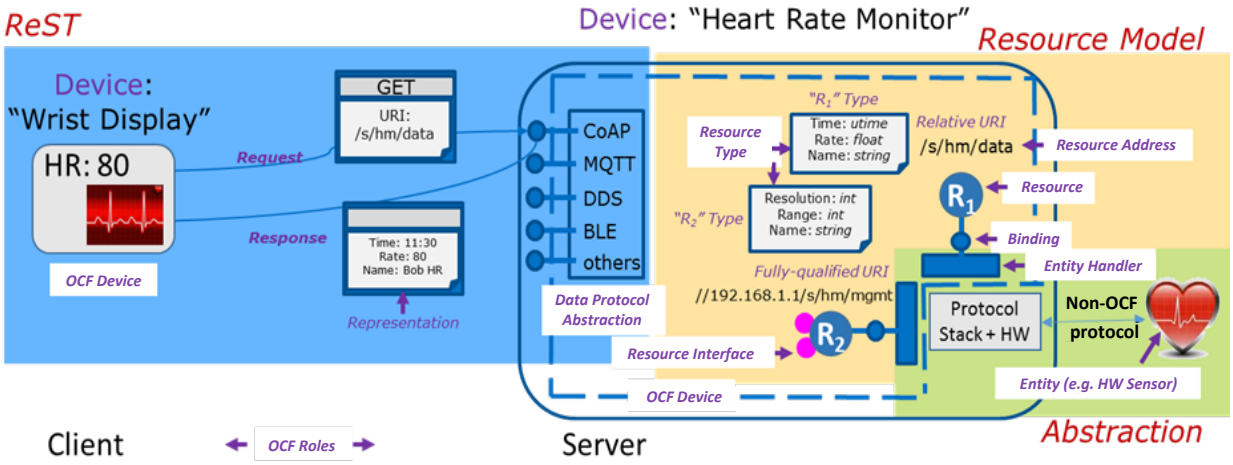


Figure 5 – Framework - Architecture Detail

The details may be implemented in many ways, for example, by using a Server with an entity handler to interface directly to a Non- OCF device as shown in Figure 6.

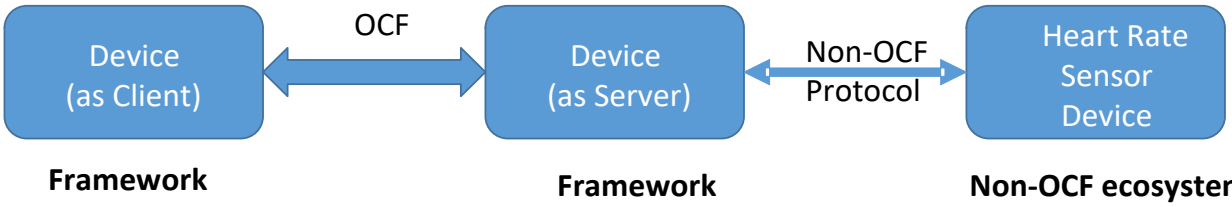


Figure 6 – Server bridging to Non- OCF device

On start-up the Server runs the entity handlers which discover the non- OCF systems (e.g., Heart Rate Sensor Device) and create Resources for each Device or functionality discovered. The entity handler creates a Resource for each discovered Device or functionality and binds itself to that Resource. These Resources are made discoverable by the Server.

Once the Resources are created and made discoverable, then the Display Device can discover these Resources and operate on them using the mechanisms described in this document. The requests to a Resource on the Server are then interpreted by the entity handler and forwarded to the Non- OCF device using the protocol supported by the Non-OCF device. The returned information from the Non- OCF device is then mapped to the appropriate response for that Resource.

5.7 OCF Cloud architecture

This clause describes the architecture of OCF Cloud in Figure 7: and Figure 8

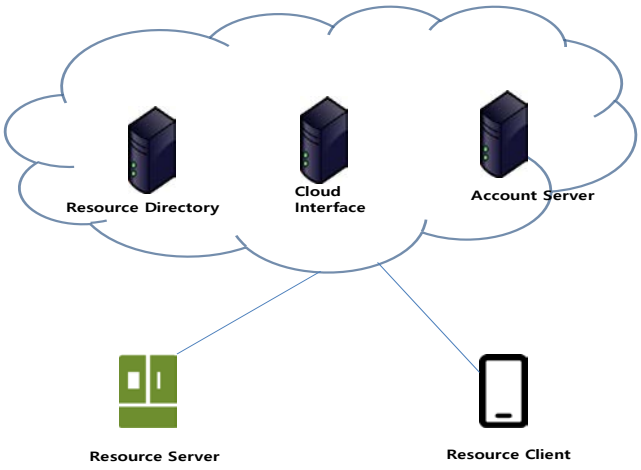


Figure 7 – OCF Cloud deployment architecture

The Cloud architecture comprises of following three network entities:

– *Cloud Interface Server* – A logical entity to which an OCF Device primarily. It encapsulates Account Server and Resource Directory features. The Cloud Interface routes the packet between OCF Devices based on the request URI in the packet header. The Client needs to keep the persistent connection alive to the Server.

– *Account Server* – A logical entity that handles Device registration, Auth Token validation and handles sign-in and token-refresh requests from the Device.

– *Resource Directory* – A logical entity holding Resource information published by Servers. A Client when looking for a Resource receives a response from the Resource Directory on behalf of the Server. Then with information included in the response form the Resource Directory, the Client directly connects to the Server.

When a Client try to access a Server, the Client connects to Cloud Interface Server then Cloud Interface routes the received message to the indicated Server after checking the privilege.

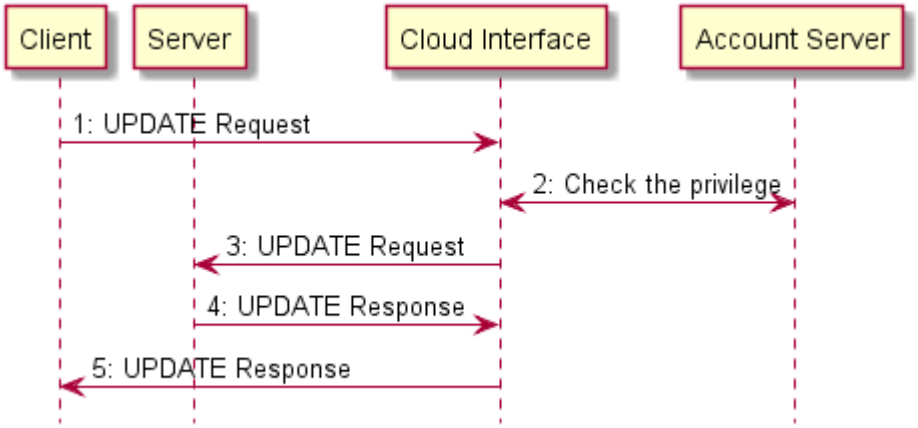


Figure 8 – OCF Endpoint routing

6 Identification and addressing

6.1 Introduction

Facilitating proper and efficient interactions between elements in the Framework, requires a means to identify, name and address these elements.

The *identifier* unambiguously identifies an element in a context or domain. The context or domain may be determined by the use or the application. The identifier is expected to be immutable over the lifecycle of that element and is unambiguous within a context or domain.

The *address* is used to define a place, way or means of reaching or accessing the element in order to interact with it. An address may be mutable based on the context.

The *name* is a handle that distinguishes the element from other elements in the Framework. The name may be changed over the lifecycle of that element.

There may be methods or resolution schemes that allow determining any of these based on the knowledge of one or more of others (e.g., determine name from address or address from name).

Each of these aspects may be defined separately for multiple contexts (e.g., a context could be a layer in a stack). So an address may be a URL for addressing Resource and an IP address for addressing at the connectivity layer. In some situations, both these addresses would be required. For example, to do RETRIEVE (see 8.3) operation on a particular Resource representation, the Client needs to know the address of the target Resource and the address of the Server through which the Resource is exposed.

In a context or domain of use, a name or address could be used as identifier or vice versa. For example, a URL could be used as an identifier for a Resource and designated as a URI.

The remainder of this clause discusses the identifier, address and naming from the point of view of the Resource model and the interactions to be supported by the Resource model. Examples of interactions are the RESTful interactions, i.e. CRUDN operation (clause 8) on a Resource. Also the mapping of these to transport protocols, e.g., CoAP is described.

6.2 Identification

6.2.1 Overview

An identifier is unambiguous within the context or domain of use. There are many schemes that may be used to generate an identifier that has the required properties. The identifier may be context-specific in that the identifier is expected to be and guaranteed to be unambiguous only within that context or domain. Identifier may also be context- independent where these identifiers are guaranteed to be unambiguous across all contexts and domains both spatially and temporally. The context-specific identifiers could be defined by simple schemes like monotonic enumeration or may be defined by overloading an address or name, for example an IP address may be an identifier within the private domain behind a gateway in a smart home. On the other hand, context-independent identifiers require a stronger scheme that derives universally unique identities, for example any one of the versions of Universally Unique Identifiers (UUIDs). Context independent identifier may also be generated using hierarchy of domains where the root of the hierarchy is identified with a UUID and sub-domains may generate context independent identifier by concatenating context-specific identifiers for that domain to the context-independent identifier of their parent.

6.2.2 Resource identification and addressing

A Resource may be identified using a URI and addressed by the same URI if the URI is a URL. In some cases a Resource may need an identifier that is different from a URI; in this case, the

Resource may have a Property whose value is the identifier. When the URI is in the form of a URL, then the URI may be used to address the Resource.

An OCF URI is based on the general form of a URI as defined in IETF RFC 3986 as follows:

```
<scheme>://<authority>/<path>?<query>
```

Specifically the OCF URI is specified in the following form:

```
ocf://<authority>/<path>?<query>
```

The following is a description of values that each component takes.

The *scheme* for the URI is "ocf". The "ocf" scheme represents the semantics, definitions and use as defined in this document. If a URI has the portion preceding the "//" (double slash) omitted, then the "ocf" scheme shall be assumed.

Each transport binding is responsible for specifying how an OCF URI is converted to a transport protocol URI before sending over the network by the requestor. Similarly on the receiver side, each transport binding is responsible for specifying how an OCF URI is converted from a transport protocol URI before handing over to the Resource model layer on the receiver.

The authority of an OCF URI shall be the Device ID ("di") value, as defined in [OCF Security], of the Server.

The *path* is a string that unambiguously identifies or references a Resource within the context of the Server. In this version of the document, a path shall not include pct-encoded non-ASCII characters or NUL characters. A *path* shall be preceded by a "/" (slash). The *path* may have "/" (slash) separated segments for human readability reasons. In the OCF context, the "/" (slash) separated segments are treated as a single string that directly references the Resources (i.e. a flat structure) and not parsed as a hierarchy. On the Server, the path or some substring in the path may be shortened by using hashing or some other scheme provided the resulting reference is unique within the context of the host.

Once a path is generated, a Client accessing the Resource or recipient of the URI should use that path as an opaque string and should not parse to infer a structure, organization or semantic.

A query string shall contain a list of "<name>=<value>" segments (aka name-value pair) each separated by a "&" (ampersand). The query string will be mapped to the appropriate syntax of the protocol used for messaging. (e.g., CoAP).

A URI may be either fully qualified or relative generation of URI.

A URI may be defined by the Client which is the creator of that Resource. Such a URI may be relative or absolute (fully qualified). A relative URI shall be relative to the Device on which it is hosted. Alternatively, a URI may be generated by the Server of that Resource automatically based on a pre-defined convention or organization of the Resources, based on an OCF Interface, based on some rules or with respect to different roots or bases.

The absolute path reference of a URI is to be treated as an opaque string and a Client should not infer any explicit or implied structure in the URI – the URI is simply an address. It is also recommended that Devices hosting a Resource treat the URI of each Resource as an opaque string that addresses only that Resource. (e.g., URI's "/a" and "/a/b" are considered as distinct addresses and Resource b cannot be construed as a child of Resource a).

6.3 Namespace:

The relative URI prefix "/oic/" is reserved as a namespace for URIs defined in OCF specifications and shall not be used for URIs that are not defined in OCF specifications.

6.4 Network addressing

The following are the addresses used in this document:

IP address

– An IP address is used when the Device is using an IP configured interface.
– When a Device only has the identity information of its peer, a resolution mechanism is needed to map the identifier to the corresponding address.

7 Resource model

7.1 Introduction

The Resource model defines concepts and mechanisms that provide consistency and core interoperability between Devices in the OCF ecosystems. The Resource model concepts and mechanisms are then mapped to the transport protocols to enable communication between the Devices – each transport provides the communication protocol interoperability. The Resource model, therefore, allows for interoperability to be defined independent of the transports.

In addition, the concepts in the Resource model support modelling of the primary artefacts and their relationships to one and another and capture the semantic information required for interoperability in a context. In this way, OCF goes beyond simple protocol interoperability to capture the rich semantics required for true interoperability in Wearable and Internet of Things ecosystems.

The primary concepts in the Resource model are: entity, Resources, Uniform Resource Identifiers (URI), Resource Types, Properties, Representations, OCF Interfaces, Collections and Links. In addition, the general mechanisms are CREATE, RETRIEVE, UPDATE, DELETE and NOTIFY. These concepts and mechanisms may be composed in various ways to define the rich semantics and interoperability needed for a diverse set of use cases that the Framework is applied to.

In the OCF Resource model Framework, an entity needs to be visible, interacted with or manipulated, it is represented by an abstraction called a Resource. A Resource encapsulates and represents the state of an entity. A Resource is identified, addressed and named using URIs.

Properties are "key=value" pairs and represent state of the Resource. A snapshot of these Properties is the Representation of the Resource. A specific view of the Representation and the mechanisms applicable in that view are specified as OCF Interfaces. Interactions with a Resource are done as Requests and Responses containing Representations.

A Resource instance is derived from a Resource Type. The uni-directional relationship between one Resource and another Resource is defined as a Link. A Resource that has Properties and Links is a Collection.

A set of Properties can be used to define a state of a Resource. This state may be retrieved or updated using appropriate Representations respectively in the response from and request to that Resource.

A Resource (and Resource Type) could represent and be used to expose a capability. Interactions with that Resource can be used to exercise or use that capability. Such capabilities can be used to define processes like discovery, management, advertisement etc. For example: *discovery of Resources on a Device* can be defined as the retrieval of a representation of a specific Resource where a Property or Properties have values that describe or reference the Resources on the Device.

The information for Request or Response with the Representation may be communicated on the wire by serializing using a transfer protocol or encapsulated in the payload of the transport protocol

– the specific method is determined by the normative mapping of the Request or Response to the transport protocol. See 11.8 for transport protocols supported.

The OpenAPI 2.0 definitions (Annex D) used in this document are normative. This includes that all defined JSON payloads shall comply with the indicated OpeAPI 2.0 definitions. Annex D contains all of the OpenAPI 2.0 definitions for Resource Types defined in this document.

7.2 Resource

A Resource shall be defined by one or more Resource Type(s) – see Annex D for Resource Type. A request to CREATE a Resource shall specify one or more Resource Types that define that Resource.

A Resource is hosted in a Device. A Resource shall have a URI as defined in clause 6. The URI may be assigned by the Authority at the creation of the Resource or may be pre-defined by the specification of the Resource Type. An example Resource representation is depicted in Figure 9.

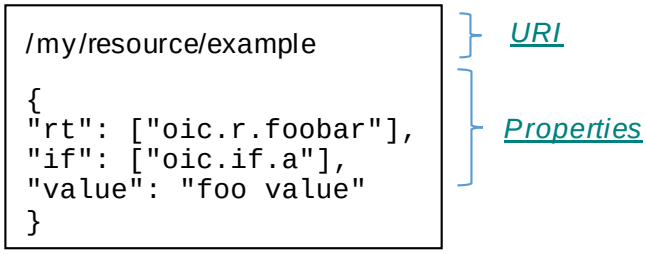


Figure 9 – Example Resource

Core Resources are the Resources defined in this document to enable functional interactions as defined in clause 10 (e.g., Discovery, Device management, etc). Among the Core Resources, "/oic/res", "/oic/p", and "/oic/d" shall be supported on all Devices. Devices may support other Core Resources depending on the functional interactions they support.

7.3 Property

7.3.1 Introduction

A Property describes an aspect that is exposed through a Resource including meta-information related to that Resource.

A Property shall have a name i.e. Property Name and a value i.e. Property Value. The Property is expressed as a key-value pair where key is the Property Name and value the Property Value like <Property Name> = <Property Value>. For example if the "temperature" Property has a Property Name "temp" and a Property Value "30F", then the Property is expressed as "temp=30F". The specific format of the Property depends on the encoding scheme. For example, in JSON, Property is represented as "key": value (e.g., "temp": 30).

In addition, the Property definition shall have a

– *Value Type* – the Value Type defines the values that a Property Value may take. The Value Type may be a simple data type (e.g. string, Boolean) as defined in 4.3 or may be a complex data type defined with a schema. The Value Type may define
  – Value Rules define the rules for the set of values that the Property Value may take. Such rules may define the range of values, the min-max, formulas, the set of enumerated values, patterns, conditional values, and even dependencies on values of other Properties. The rules may be used to validate the specific values in a Property Value and flag errors.
– *Mandatory* – specifies if the Property is mandatory or not for a given Resource Type.

– *Access modes* – specifies whether the Property may be read, written or both. Updates are equivalent to a write. "r" is used for read and "w" is used for write – both may be specified. Write does not automatically imply read.

The definition of a Property may include the following additional information – these items are informative:

– *Property Title* - a human-friendly name to designate the Property; usually not sent over the wire.

– *Description* – descriptive text defining the purpose and expected use of this Property.

In general, a Property is meaningful only within the Resource to which it is associated. However a base set of Properties that may be supported by all Resources, known as Common Properties, keep their semantics intact across Resources i.e. their "key=value" pair means the same in any Resource. Detailed tables for all Common Properties are defined in 7.3.2.

7.3.2 Common Properties

7.3.2.1 Introduction

The Common Properties defined in this clause may be specified for all Resources. The following Properties are defined as Common Properties: Resource Type, Resource Interface, Name, and Resource Identity.

The name of a Common Property shall be unique and shall not be used by other Properties. When defining a new Resource Type, its non-common Properties shall not use the name of existing Common Properties (e.g., "rt", "if", "n", "id"). When defining a new "Common Property", it should be ensured that its name has not been used by any other Properties. The uniqueness of a new Common Property name can be verified by checking all the Properties of all the existing OCF defined Resource Types. However, this may become cumbersome as the number of Resource Types grow. To prevent such name conflicts in the future, OCF may reserve a certain name space for Common Property. Potential approaches are (1) a specific prefix (e.g. "oic") may be designated and the name preceded by the prefix (e.g. "oic.psize") is only for Common Property; (2) the names consisting of one or two letters are reserved for Common Property and all other Properties shall have the name with the length larger than the 2 letters; (3) Common Properties may be nested under specific object to distinguish themselves.

The ability to UPDATE a Common Property (that supports write as an access mode) is restricted to the "oic.if.rw" (read-write) OCF Interface; thus a Common Property shall be updatable using the read-write OCF Interface if and only if the Property supports write access as defined by the Property definition and the associated schema for the read-write OCF Interface.

The following Common Properties for all Resources are specified in 7.3.2.2 through 7.3.2.6 and summarized as follows:

– *Resource Type* ("rt") – this Property is used to declare the Resource Type of that Resource. Since a Resource could be define by more than one Resource Type the Property Value of the Resource Type Property can be used to declare more than one Resource type. For example: "rt": ["oic.wk.d", "oic.d.airconditioner"] declares that the Resource containing this Property is defined by either the "oic.wk.d" Resource Type or the "oic.d.airconditioner" Resource Type. See 7.3.2.3 for details.

– *OCF Interface* ("if") – this Property declares the OCF Interfaces supported by the Resource. The Property Value of the OCF Interface Property can be multi-valued and lists all the OCF Interfaces supported. See 7.3.2.4 for details.

– *Name* ("n") – the Property declares human-readable name assigned to the Resource. See 7.3.2.5.

– *Resource Identity* ("id"): its Property Value shall be a unique (across the scope of the host Server) instance identifier for a specific instance of the Resource. The encoding of this identifier is Device and implementation dependent. See 7.3.2.6 for details.

7.3.2.2 Property Name and Property Value definitions

The Property Name and Property Value as used in this document:

– *Property Name*– the key in "key=value" pair. Property Name is case sensitive and its data type is "string". Property names shall contain only letters A to Z, a to z, digits 0 to 9, hyphen, and dot, and shall not begin with a digit.

– *Property Value* – the value in "key=value" pair. Property Value is case sensitive when its data type is "string".

7.3.2.3 Resource Type

Resource Type Property is specified in 7.4.

7.3.2.4 OCF Interface

OCF Interface Property is specified in 7.6.

7.3.2.5 Name

A human friendly name for the Resource, i.e. a specific resource instance name (e.g., MyLivingRoomLight), The Name Property is as defined in Table 2

Table 2 – Name Property Definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	"n"	"string"	N/A	N/A	R, W	No	Human understandable name for the Resource.

The Name Property is read-write unless otherwise restricted by the Resource Type (i.e. the Resource Type does not support UPDATE or does not support UPDATE using read-write).

7.3.2.6 Resource Identity

The Resource Identity Property shall be a unique (across the scope of the host Server) instance identifier for a specific instance of the Resource. The encoding of this identifier is Device and implementation dependent as long as the uniqueness constraint is met, noting that an implementation may use a uuid as defined in 4.3. The Resource Identity Property is as defined in Table 3.

Table 3 – Resource Identity Property Definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Identity	"id"	"string" or uuid	Implementation Dependent	N/A	R	No	Unique identifier of the Resource (over all Resources in the Device)

7.4 Resource Type

7.4.1 Introduction

Resource Type is a class or category of Resources and a Resource is an instance of one or more Resource Types.

The Resource Types of a Resource is declared using the Resource Type Common Property as described in 7.3.2.3 or in a Link using the Resource Type Parameter.

A Resource Type may either be pre-defined by OCF or in custom definitions by manufacturers, end users, or developers of Devices (vendor-defined Resource Types). Resource Types and their definition details may be communicated out of band (i.e. in documentation) or be defined explicitly using a meta-language which may be downloaded and used by APIs or applications. OCF has adopted OpenAPI 2.0 as the specification method for OCF's RESTful interfaces and Resource definitions.

Every Resource Type shall be identified with a Resource Type ID which shall be represented using the requirements and ABNF governing the Resource Type attribute in IETF RFC 6690 (clause 2 for ABNF and clause 3.1 for requirements) with the caveat that segments are separated by a "." (period). The entire string represents the Resource Type ID. When defining the ID each segment may represent any semantics that are appropriate to the Resource Type. For example, each segment could represent a namespace. Once the ID has been defined, the ID should be used opaquely and implementations should not infer any information from the individual segments. The string "oic", when used as the first segment in the definition of the Resource Type ID, is reserved for OCF-defined Resource Types. All OCF defined Resource Types are to be registered with the IANA Core Parameters registry as described also in IETF RFC 6690.

7.4.2 Resource Type Property

A Resource when instantiated or created shall have one or more Resource Types that are the template for that Resource. The Resource Types that the Resource conforms to shall be declared using the "rt" Common Property for the Resource as defined in Table 4. The Property Value for the "rt" Common Property shall be the list of Resource Type IDs for the Resource Types used as templates (i.e., "rt"=<list of Resource Type IDs>).

Table 4 – Resource Type Common Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Type	"rt"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	Yes	The Property name rt is as described in IETF RFC 6690

Resource Types may be explicitly discovered or implicitly shared between the user (i.e. Client) and the host (i.e. Server) of the Resource.

7.4.3 Resource Type definition

Resource Type is specified as follows:

– *Pre-defined URI* (optional) – a pre-defined URI may be specified for a specific Resource Type in an OCF specification. When a Resource Type has a pre-defined URI, all instances of that Resource Type shall use only the pre-defined URI. An instance of a different Resource Type shall not use the pre-defined URI.

– *Resource Type Title* (optional) – a human friendly name to designate the Resource Type.

- *Resource Type ID* – the value of "rt" Property which identifies the Resource Type, (e.g., "oic.wk.p").
- *Resource Interfaces* – list of the OCF Interfaces that may be supported by the Resource Type.
- *Properties* – definition of all the Properties that apply to the Resource Type. The Resource Type definition shall define whether a property is mandatory, conditional mandatory, or optional.
- *Related Resource Types* (optional) – the specification of other Resource Types that may be referenced as part of the Resource Type, applicable to Collections.
- *Mime Types* (optional) – mime types supported by the Resource including serializations (e.g., application/cbor, application/json, application/xml).

Table 5 and Table 6 provides an example description of an illustrative foobar Resource Type and its associated Properties.

Table 5 – Example foobar Resource Type

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction	M/CR/O
none	"foobar"	"oic.r.foobar"	"oic.if.a"	Example "foobar" Resource	Actuation	O

Table 6 – Example foobar Properties

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Type	"rt"	"array"	N/A	N/A	R	Yes	Resource Type
OCF Interface	"if"	"array"	N/A	N/A	R	Yes	OCF Interface
Foo value	value	"string"	N/A	N/A	R	Yes	Foo value

For example, an instance of the foobar Resource Type.

```
{
"rt": ["oic.r.foobar"],
"if": ["oic.if.a"],
"value": "foo value"
}
```

For example, a schema representation for the foobar Resource Type.

```
{
  "$schema": "http://json-schema.org/draft-04/schema",
  "type": "object",
  "properties": {
    "rt": {
      "type": "array",
      "items" : {
        "type" : "string",
        "maxLength": 64
      },
```



```
        "minItems" : 1,
        "readOnly": true,
        "description": "Resource Type of the Resource"
      },
      "if": {
        "type": "array",
        "items": {
          "type" : "string",
          "enum" : ["oic.if.baseline", "oic.if.ll", "oic.if.b", "oic.if.lb", "oic.if.rw",
"oic.if.r", "oic.if.a", "oic.if.s"]
      },
      "value": {"type": "string"}
    },
    "required": ["rt", "if", "value"]
}
```

7.4.4 Multi-value "rt" Resource

Multi-value "rt" Resource means a Resource with multiple Resource Types where none of the included Resource Types denote a well-known Resource Type (i.e. "oic.wk.<thing>"). Such a Resource is associated with multiple Resource Types and so has an "rt" Property Value of multiple Resource Type IDs (e.g. "rt": ["oic.r.switch.binary", "oic.r.light.brightness"]). The order of the Resource Type IDs in the "rt" Property Value is meaningless. For example, "rt": ["oic.r.switch.binary", "oic.r.light.brightness"] and "rt": ["oic.r.light.brightness", "oic.r.switch.binary"] have the same meaning.

Resource Types for multi-value "rt" Resources shall satisfy the following conditions:

– Property Name – Property Names for each Resource Type shall be unique (within the scope of the multi-value "rt" Resource) with the exception of Common Properties, otherwise there will be conflicting Property semantics. If two Resource Types have a Property with the same Property "Name, a multi-value "rt" Resource shall not be composed of these Resource Types.

A multi-value "rt" Resource satisfies all the requirements for each Resource Type and conforms to the OpenAPI 2.0 definitions for each component Resource Type. Thus the mandatory Properties of a multi-value "rt" Resource shall be the union of all the mandatory Properties of each Resource Type. For example, mandatory Properties of a Resource with "rt": ["oic.r.switch.binary", "oic.r.light.brightness"] are "value" and "brightness", where the former is mandatory for "oic.r.switch.binary" and the latter for "oic.r.light.brightness".

The multi-value "rt" Resource Interface set shall be the union of the sets of OCF Interfaces from the component Resource Types. The Resource Representation in response to a CRUDN action on an OCF Interface shall be the union of the schemas that are defined for that OCF Interface. The Default OCF Interface for a multi-value "rt" Resource shall be the baseline OCF Interface ("oic.if.baseline") as that is the only guaranteed common OCF Interface between the Resource Types.

For clarity if each Resource Type supports the same set of OCF Interfaces, then the resultant multi-value "rt" Resource has that same set of OCF Interfaces with a Default OCF Interface of baseline ("oic.if.baseline").

See 7.10.3 for the handling of query parameters as applied to a multi-value "rt" Resource.

7.5 Device Type

A Device Type is a class of Device. Each Device Type defined will include a list of minimum Resource Types that a Device shall implement for that Device Type. A Device may expose additional standard and vendor defined Resource Types beyond the minimum list. The Device Type is used in Resource discovery as specified in 11.3.4.

Like a Resource Type, a Device Type can be used in the Resource Type Common Property or in a Link using the Resource Type Parameter.

A Device Type may either be pre-defined (ISO/IEC 30118-5:2018) or in custom definitions by manufacturers, end users, or developers of Devices (vendor-defined Device Types). Device Types and their definition details may be communicated out of band (like in documentation).

Every Device Type shall be identified with a Resource Type ID using the same syntax constraints as a Resource Type.

7.6 OCF Interface

7.6.1 Introduction

An OCF Interface provides first a view into the Resource and then defines the requests and responses permissible on that view of the Resource. So this view provided by an OCF Interface defines the context for requests and responses on a Resource. Therefore, the same request to a Resource when targeted to different OCF Interfaces may result in different responses.

An OCF Interface may be defined by either this document (a Core OCF Interface), ISO/IEC 30118-5:2018 (a vertical OCF Interface) or manufacturers, end users or developers of Devices (a vendor-defined OCF Interface).

The OCF Interface Property lists all the OCF Interfaces the Resource support. All Resources shall have at least one OCF Interface. The Default OCF Interface shall be defined by an OCF specification and inherited from the Resource Type definition. The Default OCF Interface associated with all Resource Types defined in this documentshall be the supported OCF Interface listed first within the applicable enumeration in the definition of the Resource Type (see Annex D). All Default OCF Interfaces specified in an OCF specification shall be mandatory.

In addition to any OCF specification defined OCF Interface, all Resources shall support the baseline OCF Interface ("oic.if.baseline") as defined in 7.6.3.2.

See 7.10.4 for the use of queries to enable selection of a specific OCF Interface in a request.

An OCF Interface may accept more than one media type. An OCF Interface may respond with more than one media type. The accepted media types may be different from the response media types. The media types are specified with the appropriate header parameters in the transfer protocol. (NOTE: This feature has to be used judiciously and is allowed to optimize representations on the wire) Each OCF Interface shall have at least one media type.

7.6.2 OCF Interface Property

Table 7 – Resource Interface Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
OCF **Interface**	"if"	"array"	Array of strings, conveying OCF Interfaces	N/A	R	Yes	Property to declare the OCF Interfaces supported by a Resource.

The OCF Interfaces supported by a Resource shall be declared using the OCF Interface Common Property (Table 7), e.g., ""if": ["oic.if.ll", "oic.if.baseline"]". The Property Value of an OCF Interface Property shall be a lower case string with segments separated by a "." (dot). The string "oic", when used as the first segment in the OCF Interface Property Value, is reserved for OCF-defined OCF

Interfaces. The OCF Interface Property Value may also be a reference to an authority similar to IANA that may be used to find the definition of an OCF Interface. A Resource Type shall support one or more of the OCF Interfaces defined in 7.6.3.

7.6.3 OCF Interface methods

7.6.3.1 Overview

OCF Interface methods shall not violate the defined OpenAPI 2.0 definitions for the Resources as defined in Annex D.

The defined OCF Interfaces are listed in Table 8:

Table 8 – OCF standard OCF Interfaces

OCF Interface	Name	Applicable Operations	Description
baseline	"oic.if.baseline"	RETRIEVE, NOTIFY, UPDATE	The baseline OCF Interface defines a view into all Properties of a Resource including the Meta Properties. This OCF Interface is used to operate on the full Representation of a Resource.
links list	"oic.if.ll"	RETRIEVE, NOTIFY	The links list OCF Interface provides a view into Links in a Collection (Resource). Since Links represent relationships to other Resources, the links list OCF Interfaces may be used to discover Resources with respect to a context. The discovery is done by retrieving Links to these Resources. For example: the Core Resource "/oic/res" uses this OCF Interface to allow discovery of Resource hosted on a Device.
batch	"oic.if.b"	RETRIEVE, NOTIFY, UPDATE	The batch OCF Interface is used to interact with a Collection of Resources at the same time. This also removes the need for the Client to first discover the Resources it is manipulating – the Server forwards the requests and aggregates the responses
read-only	"oic.if.r"	RETRIEVE NOTIFY	The read-only OCF Interface exposes the Properties of a Resource that may be read. This OCF Interface does not provide methods to update Properties, so can only be used to read Property Values.
read-write	"oic.if.rw"	RETRIEVE, NOTIFY, UPDATE	The read-write OCF Interface exposes only those Properties that may be read from a Resource during a RETRIEVE operation and only those Properties that may be written to a Resource during and UPDATE operation.
actuator	"oic.if.a"	RETRIEVE, NOTIFY, UPDATE	The actuator OCF Interface is used to read or write the Properties of an actuator Resource.
sensor	"oic.if.s"	RETRIEVE, NOTIFY	The sensor OCF Interface is used to read the Properties of a sensor Resource.

7.6.3.2 Baseline OCF Interface

7.6.3.2.1 Overview

The Representation that is visible using the baseline OCF Interface includes all the Properties of the Resource including the Common Properties. The baseline OCF Interface shall be defined for all Resource Types. All Resources shall support the baseline OCF Interface.

7.6.3.2.2 Use of RETRIEVE

The baseline OCF Interface is used when a Client wants to retrieve all Properties of a Resource; that is the Server shall respond with a Resource representation that includes all of the implemented

Properties of the Resource. When the Server is unable to send back the whole Resource representation, it shall reply with an error message. The Server shall not return a partial Resource representation.

An example response to a RETRIEVE request using the baseline OCF Interface:

```
{
"rt": ["oic.r.temperature"],
"if": ["oic.if.a","oic.if.baseline"],
"temperature": 20,
"units": "C",
"range": [0,100]
}
```

7.6.3.2.3 Use of UPDATE

Using the baseline OCF Interface, all Properties of a Resource with the exception of Common Properties may be modified using an UPDATE request with a list of Properties and their desired values if a Resource Type has an associated schema for UPDATE using baseline. If the OCF Interfaces exposed by a Resource in addition to the baseline OCF Interface do not support the UPDATE semantic then UPDATE using the baseline OCF Interface is also not supported.

7.6.3.3 Links List OCF Interface

7.6.3.3.1 Overview

The links list OCF Interface provides a view into the list of Links in a Resource. The Representation visible through this OCF Interface has only the Links exposed as Property(-ies) that is(are) an array (or arrays) of Links by the Resource – so this OCF Interface is used to manipulate or interact with the list of Links. The Links list may be RETRIEVEd using this OCF Interface.

The lists list OCF Interface is defined as follows:

- The links list OCF Interface name is "oic.if.ll".
- If there are no Links present in a Resource, then an empty list shall be returned in response to a RETRIEVE request using the links list OCF Interface.
- The Representation determined by this OCF Interface depends on the requesting Client. For a Client that includes an OCF-Accept-Content-Format-Version option as defined in 12.2.5 in the request the response only includes the Property value(s) of the Property(-ies) that are arrays of Links, hence a Collection or "/oic/res" response with oic.if.ll is an array of Links. For a Client that does not include an OCF-Accept-Content-Format-Version option the response is as defined in Annex E.
- The array of Links may be observed by a Client using the links list OCF Interface (i.e. by following the procedures in clause 11.4.2 with the addition of a query parameter of "?if=oic.if.ll").
- Any CREATE, UPDATE, or DELETE operation on any Link in the array of Links shall result in the complete Resource representation for the links list OCF Interface as defined for the target Resource (i.e. the full array of Links) subject to any applied filtering being provided in the notification that is sent to the Client that initiated the Observe request.
- If the act of deleting a Link results in no Links being present then an empty list shall be sent in a notification.

7.6.3.3.2 Example: links list OCF Interface

A request to a Collection, where the request is to RETRIEVE the Links in room (the Links could be referencing lights, fans, electric sockets etc).

```
GET ocf://<devID>/a/room/1?if=oic.if.ll
The response would be the array of OCF Links
```



```
[
  {
    "href": "/the/light/1",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "eps":[
{"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
  },
  {
    "href": "/the/light/2",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
  },
  {
    "href": "/my/fan/1",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
"eps":[
{"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
  {
    "href": "/his/fan/2",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "eps":[
{"ep": "coaps://[2001:db8:a::b1d4]:55555"}]
  }
]
```

7.6.3.4 Batch OCF Interface

7.6.3.4.1 Overview

The batch OCF Interface is used to interact with a Collection of Resources using a single/same Request. The batch OCF Interface can be used to RETRIEVE or UPDATE the Properties of the linked Resources with a single request.

The batch OCF Interface is defined as follows:

– The batch OCF Interface name is "oic.if.b"

– A Collection Resource has linked Resources that are represented as URIs. In the "href" Property of the batch payload the URI shall be fully qualified for remote Resources and a relative reference for local Resources.

– The original request is modified to create new requests targeting each of the linked Resources in the Collection by substituting the URI in the original request with the URI of the linked Resource. The payload in the original request is replicated in the payload of the new requests.

– The requests shall be forwarded assuming use of the Default OCF Interface of the linked Resources.

– Requests shall only be forwarded to linked Resources that are identified by relation types "item" or "hosts" ("hosts" is the default relation type value should the "rel" Link Parameter not be present). Requests shall not be forwarded to linked Resources that do not contain the "item" or "hosts" relation type values.

– Properties of the Collection Resource itself may be included in payloads using "oic.if.b" OCF Interface by exposing a single Link with the link relation "self" along with "item" within the Collection, and ensuring that Link resolution cannot become an infinite loop due to recursive references. For example, if the Default OCF Interface of the Collection is "oic.if.b", then the Server might recursively include its batch representation within its batch representation, in an endless loop. See 7.6.3.4.2 for an example of use of a Link containing "rel": ["self","item"] to

include Properties of the Collection Resource, along with linked Resources, in "oic.if.b" payloads.

– If the Default OCF Interface of a Collection Resource is exposed using the Link relation "self", and the Default OCF Interface contains Properties that expose any Links, those Properties shall not be included in a batch representation which includes the "self" Link.

– Any request forwarded to a linked Resource that is a Collection (including a "self" Link reference) shall have the Default OCF Interface of the linked Collection Resource applied.

– All the responses from the linked Resources shall be aggregated into a single Response to the Client. The Server may timeout the response to a time window, the Server may choose any appropriate window based on conditions.

– If a linked Resource cannot process the request, an empty response, i.e. a JSON object with no content ("{}") as the representation for the "rep" Property, or error response should the linked Resource Type provide an error schema or diagnostic payload, shall be returned by the linked Resource. These empty or error responses for all linked Resources that exhibit an error shall be included in the aggregated response to the original Client request. See the example in 7.6.3.4.2.

– If any of the linked Resources returns an error response, the aggregated response sent to the Client shall also indicate an error (e.g. 4.xx in CoAP). If all of the linked Resources return successful responses, the aggregated response shall include the success response code.

– The aggregated response shall be an array of objects representing the responses from each linked Resource. Each object in the response shall include at least two items: (1) the URI of the linked Resource (fully qualified for remote Resources, or a relative reference for local Resources) as "href": <URI> and (2) the individual response object or array of objects if the linked Resource is itself a Collection using "rep" as the key, e.g. "rep": { <representation of individual response> }.

– If the Collection Resource is marked as Observable, linked Resources referenced in the Collection may be Observed using the batch OCF Interface. If the Collection Resource is not marked as Observable then the Collection cannot be Observed and Observe requests to the Collection shall be handled as defined for the case where request validation fails in clause 11.4.2.4. The Observe mechanism shall work as defined in 11.4.2 with the Observe request forwarded to each of the linked Resources. All responses to the request shall be aggregated into a single response to the Client using the same representations and status codes as for RETRIEVE operations using the batch OCF Interface.

– Should any one of the Observable linked Resources fail to honour the Observe request the response to the batch Observe request shall also indicate that the entire request was not honoured using the mechanism described in 11.4.2.4.

– If any of the Observable Resources in a request to a Collection using the batch OCF Interface replies with an error or Observe Cancel, the Observations of all other linked Resources shall be cancelled and the error or Observe Cancel status shall be returned to the Observing Client.

NOTE Behavior may be different for Links that do network requests vs. local Resources.

– All notifications to the Client that initiated an Observe request using the batch OCF Interface shall use the batch representation for the Collection. This is the aggregation of any individual Observe notifications received by the Device hosting the Collection from the individual Observe requests that were forwarded to the linked Resources.

– Linked Resources which are not marked Observable in the Links of a Collection shall not trigger Notifications, but may be included in the response to, and subsequent Notifications resulting from, an Observe request to the batch OCF Interface of a Collection.

– Each notification shall contain the most current values for all of the Linked Resources that would be included if the original Observe request were processed again. The Server hosting the

Collection may choose to RETRIEVE all of the linked Resources each time, or may choose to employ caching to avoid retrieving linked Resources on each Notification.

– If a Linked Resource is Observable and has responded with a successful Observe response, the most recently reported value of that Resource is considered to be the most current value and may be reported in all subsequent Notifications.

– Links in the Collection should be Observed by using the "oic.if.ll" OCF Interface. A notification shall be sent any time the contents of the "oic.if.ll" OCF Interface representation are changed; that is, if a Link is added, if a Link is removed, or if a Link is updated. Notifications on the "oic.if.ll" OCF Interface shall contain all of the Links in the "oic.if.ll" OCF Interface representation.

– Other Properties of the Collection Resource, if present, may be Observed by using the OCF Interfaces defined in the definition for the Resource Type, including using the "oic.if.baseline" OCF Interface.

– The Client may choose to restrict the linked Resources to which the request is forwarded by including additional query parameters in the request. The Server should process any additional query parameters in a request that includes "oic.if.b" as selectors for linked Resources that are to be processed by the request.

– A Client shall perform UPDATE operations using the batch OCF Interface by creating a payload that is similar to a RETRIEVE response payload from a batch OCF Interface request. The Server shall send a separate UPDATE request to each of the linked Resources according to each "href" Property and the corresponding value of the "rep" Property.

– If the "href" value is empty, denoted by a zero length string or "" in JSON, the "rep" Property shall be applied to linked Resources in the Collection.

– Items with the empty "href" and link-specific "href" shall not be mixed in the same UPDATE request.

– All of the Properties in the UPDATE request may not be supported by the linked Resource. In such cases, writable Properties in the UPDATE request that are supported by the linked Resource shall be modified and Properties that are not supported shall be silently ignored.

– The UPDATE response shall contain the updated values using the same payload schema as RETRIEVE operations if provided by the linked Resource, along with the appropriate status code. The aggregated response payload shall reflect the known state of the updated Properties after the batch update was completed. If no payload is provided by the updated Resource then an empty response (i.e. "rep": {}) shall be provided for that Resource.

– A Collection shall not support the use of the UPDATE operation to add, modify or remove Links in an existing Collection using the "oic.if.baseline", "ic.if.rw" or "oic.if.a" OCF Interfaces.

7.6.3.4.2 Examples: Batch OCF Interface

Note that the examples provided in Table 9 are illustrative and do not include all mandatory schema elements in all cases. It is assumed that the Default OCF Interface for the Resource Type "x.org.example.rt.room" is specified in its Resource Type definition file as "oic.if.rw", which exposes the Properties "x.org.example.colour" and "x.org.example.size".

1646 **Table 9 – Batch OCF Interface Example**

Resources

```
/a/room/1
{
  "rt": "x.org.example.rt.room"],
  "if": ["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],
  "x.org.example.colour": "blue",
  "x.org.example.dimension": "15bx15wx10h",
  "links": [
     {"href": "/a/room/1", "rel": ["self", "item"], "rt":
["x.org.example.rt.room"], "if":
["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],"p": {"bm": 2} },
     {"href": "/the/light/1", "rel": ["item"], "rt": ["oic.r.switch.binary"],
"if": ["oic.if.a","oic.if.baseline"], "ins": "11111", "p": {"bm": 2} },
     {"href": "/the/light/2", "rel": ["item"], "rt": ["oic.r.switch.binary"],
"if": ["oic.if.a" ,"oic.if.baseline"], "ins": "22222", "p": {"bm": 2} },
     {"href": "/my/fan/1", "rel": ["item"], "rt": ["oic.r.switch.binary"],
"if": ["oic.if.a", "oic.if.baseline"], "ins": "33333", "p": {"bm": 2} },
     {"href": "/his/fan/2", "rel": ["item"], "rt": ["oic.r.switch.binary"],
"if": ["oic.if.a", "oic.if.baseline"], "ins": "44444", "p": {"bm": 2} },
     {"href": "/the/switches/1", "rel": ["item"], "rt": ["oic.wk.col"],
"if":["oic.if.ll", "oic.if.b", "oic.if.baseline"], "ins": "55555", "p": {"bm":
2} }
  ]
}

/the/light/1
{
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "value": false
}

/the/light/2
{
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "value": true
}

/my/fan/1
{
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "value": true
}

/his/fan/2
{
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
  "value": false
}
/the/switches/1
{
  "rt": ["oic.wk.col"],
  "if":["oic.if.ll", "oic.if.b", "oic.if.baseline"],
"links": [
  {
    "href": "/switch-1a",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a","oic.if.baseline"],
    "p": {"bm": 2}
  }
```


	<pre>{
	"href": "/switch-1b",
	"rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a","oic.if.baseline"],
	"p": {"bm": 2 }
	}
	]
	}</pre>

Use of batch, successful response	Request: GET /a/room/1?if=oic.if.b
	Becomes the following individual request messages issued by the Device in the Client role
	GET /a/room/1 (NOTE: uses the Default OCF Interface as specified for the Collection Resource, in this example oic.if.rw)
	GET /the/light/1 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /the/light/2 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /my/fan/1 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /his/fan/2 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /the/switches/1 (NOTE: Uses the Default OCF Interface for the Collection that is within the Collection)
	Response:
	[
	{
	"href": "/a/room/1",
	"rep": {"x.org.example.colour": "blue","x.org.example.dimension": "15bx15wx10h"}
	},
	{
	"href": "/the/light/1",
	"rep": {"value": false}
	},
	{
	"href": "/the/light/2",
	"rep": {"value": true}
	},
	{
	"href": "/my/fan/1",
	"rep": {"value": true}
	},
	{
	"href": "/his/fan/2",
	"rep": {"value": false}
	},
	{
	"href": "/the/switches/1",
	"rep": [
	{
	"href": "/switch-1a",
	"rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a","oic.if.baseline"],
	"p": {"bm": 2},
	"eps":[
	{"ep": "coaps://[2001:db8:a::b1d4]:55555"}
	]
	},
	{
	"href": "/switch-1b",
	"rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a","oic.if.baseline"],
	"p": {"bm": 2 },
	"eps":[
	{"ep": "coaps://[2001:db8:a::b1d4]:55555"}
	]
	}]
	}
	]

Use of batch, error response	Should any of the RETRIEVE requests in the previous example fail then the response includes an empty payload for that Resource instance and an error code is sent. The following example assumes errors from "/my/fan/1" and "/the/switches/1"
	
	Error Response:
	
	[
	{
	"href": "/a/room/1",
	"rep": {"x.org.example.colour": "blue","x.org.example.dimension": "15bx15wx10h"}
	},
	{
	"href": "/the/light/1",
	"rep": {"value": false}
	},
	{
	"href": "/the/light/2",
	"rep": {"value": true}
	},
	{
	"href": "/my/fan/1",
	"rep": {}
	},
	{
	"href": "/his/fan/2",
	"rep": {"value": false}
	},
	{
	"href": "/the/switches/1",
	"rep": {}
	}
	]

Use of batch

(UPDATE has POST semantics)

```
UPDATE /a/room/1?if=oic.if.b
[
  {
    "href": "",
    "rep": {
      "value": false
    }
  }
]
```

Since the "href" value in the UPDATE request is empty, the request is forwarded to all Resources in the Collection and becomes:

```
UPDATE /a/room/1 { "value": false }
UPDATE /the/light/1 { "value": false }
UPDATE /the/light/2 { "value": false }
UPDATE /my/fan/1 { "value": false }
UPDATE /his/fan/2 { "value": false }
UPDATE /the/switches/1 { "value": false }
```

Response:

```
[
  {
    "href": "/the/light/1",
    "rep": {"value": false}
  },
  {
    "href": "/the/light/2",
    "rep": {"value": false}
  },
  {
    "href": "/my/fan/1",
    "rep": {"value": false}
  },
  {
    "href": "/his/fan/2",
    "rep": {"value": false}
  },
  {
    "href": "/the/switches/1",
    "rep":
      {
      }
  }
]
```

Since /a/room/1 does not have a "value" Property exposed by its Default OCF Interface, the UPDATE request will be silently ignored and it will not be included in the UPDATE response.

Since the UPDATE request with the links list OCF Interface is not allowed, an empty payload for the "/the/switches/1" is included in the UPDATE response and an error code is sent.

Use of batch (UPDATE has POST semantics)	UPDATE /a/room/1?if=oic.if.b
	[
	  {
	  "href": "/the/light/1",
	  "rep": {
	  "value": false
	  }
	  },
	  {
	  "href": "/the/light/2",
	  "rep": {
	  "value": true
	  }
	  },
	  {
	  "href": "/a/room/1",
	  "rep": {
	  "x.org.example.colour": "red"
	  }
	  }
	]
	
	This turns /the/light/1 off, turns /the/light/2 on, and sets the colour of /a/room/1 to "red".
	
	The response will be same as response for GET /a/room/1?if=oic.if.b with the updated Property values as shown.
	[
	 {
	  "href": "/a/room/1",
	  "rep":{"x.org.example.colour": "red",
	  "x.org.example.dimension": "15bx15wx10h"}
	 },
	 {
	  "href": "/the/light/1",
	  "rep": {"value": false}
	 },
	 {
	  "href": "/the/light/2",
	  "rep": {"value": true}
	 }
	]
	Example use of additional query parameters to select items by matching Link Parameters.
	
	Turn on light 1 based on the "ins" Link Parameters value of "11111"
	
	UPDATE /a/room/1?if=oic.if.b&ins=11111
	[
	  {
	  "href": "",
	  "rep": {
	  "value": false
	  }
	  }
	]
	
	Similar to the earlier example, "href": "" applies the UPDATE request to all of the Resources in the Collection. Since the additional query parameter ins=11111 selects only links that have a matching "ins" value, only one link is selected. The payload is applied to the target Resource of that link, /the/light/1.

	Retrieving the item using the same query parameter:
	`RETRIEVE /a/room/1?if=oic.if.b&ins=11111`
	
	Response payload:
	`[`
	`  {`
	`  "href": "/the/light/1",`
	`  "rep": {`
	`  "value": false`
	`  }`
	`  }`
	`]`

7.6.3.5 Actuator OCF Interface

The actuator OCF Interface is the OCF Interface for viewing Resources that may be actuated i.e. changes some value within or the state of the entity abstracted by the Resource:

– The actuator OCF Interface name shall be "oic.if.a"

– The actuator OCF Interface shall expose in the Resource Representation all mandatory Properties as defined by the applicable OpenAPI 2.0 schema; the actuator OCF Interface may also expose in the Resource Representation optional Properties as defined by the applicable OpenAPI 2.0 schema that are implemented by the target Device.

For example, a "Heater" Resource (for illustration only):

```
/a/act/heater
{
   "rt": ["acme.gas"],
   "if": ["oic.if.baseline", "oic.if.r", "oic.if.a", "oic.if.s"],
   "settemp": 10,
   "currenttemp" : 7
}
```

The actuator OCF Interface with respect to "Heater" Resource (for illustration only):

a) Retrieving values of an actuator.

```
Request: GET /a/act/heater?if="oic.if.a"

Response:
{
   "settemp": 10,
   "currenttemp" : 7
}
```

b) Correct use of actuator OCF Interface.

```
Request: POST /a/act/heater?if="oic.if.a"
{
  "settemp": 20
}
Response:
            {
              Ok
            }
```

c) Incorrect use of actuator OCF Interfance.


```
Request: POST /a/act/heater?if="oic.if.a"
  {
   "if": ["oic.if.s"]   ← this is visible through baseline OCF Interface
}
Response:
{
   Error
}
```

– A RETRIEVE request using this OCF Interface shall return the Representation for this Resource subject to any query and filter parameters that may also exist.

– An UPDATE request using this OCF Interface shall provide a payload or body that contains the Properties that will be updated on the target Resource.

7.6.3.6 Sensor OCF Interface

The sensor OCF Interface is the OCF Interface for retrieving measured, sensed or capability specific information from a Resource that senses:

– The sensor OCF Interface name shall be "oic.if.s".

– The sensor OCF Interface shall expose in the Resource Representation all mandatory Properties as defined by the applicable OpenAPI 2.0 schema; the sensor OCF Interface may also expose in the Resource Representation optional Properties as defined by the applicable OpenAPI 2.0 schema that are implemented by the target Device.

– A RETRIEVE request using this OCF Interface shall return this representation for the Resource subject to any query and filter parameters that may also exist.

NOTE: The example here is with respect to retrieving values of a sensor

```
Request: GET /a/act/heater?if="oic.if.s"

Response:
{
  "currenttemp": 7
}
```

Incorrect use of the sensor.

```
Request: PUT /a/act/heater?if="oic.if.s" ← PUT is not allowed
{
  "settemp": 20   ← this is possible through actuator OCF Interface
}
Response:
{
  Error
}
```

Another incorrect use of the sensor.

```
Request: POST /a/act/heater?if="oic.if.s" ← POST is not allowed
 {
  "currenttemp": 15   ← this is possible through actuator OCF Interface
}
Response:
{
  Error
}
```


7.6.3.7 Read-only OCF Interface

The read-only OCF Interface exposes only the Properties that may be read. This includes Properties that may be read-only, read-write but not Properties that are write-only or set-only. The applicable operations that can be applied to a Resource are only RETRIEVE and NOTIFY. An attempt by a Client to apply a method other than RETRIEVE or NOTIFY to a Resource shall be rejected with an error response code.

7.6.3.8 Read-write OCF Interface

The read-write OCF Interface is a generic OCF Interface to support reading and setting Properties in a Resource. The applicable methods that can be applied to a Resource are only RETRIEVE, NOTIFY, and UPDATE. For the RETRIEVE and NOTIFY operations, the behaviour is the same as for the "oic.if.r" OCF Interface defined in 7.6.3.7. For the UPDATE operation, read-only Properties (i.e. Properties tagged with "readOnly=True" in the OpenAPI 2.0 definition) shall not be in the UPDATE payload. An attempt by a Client to apply a method other than RETRIEVE, NOTIFY, or UPDATE to a Resource shall be rejected with an error response code.

7.7 Resource representation

Resource representation captures the state of a Resource at a particular time. The Resource representation is exchanged in the request and response interactions with a Resource. A Resource representation may be used to retrieve or update the state of a Resource.

The Resource representation shall not be manipulated by the data connectivity protocols and technologies (e.g., CoAP, UDP/IP or BLE).

7.8 Structure

7.8.1 Introduction

In many scenarios and contexts, the Resources may have either an implicit or explicit structure between them. A structure can, for example, be a tree, a mesh, a fan-out or a fan-in. The Framework provides the means to model and map these structures and the relationships among Resources. The primary building block for Resource structures in Framework is the Collection. A Collection represents a container, which is extensible to model complex structures.

7.8.2 Resource Relationships

7.8.2.1 Introduction

Resource relationships are expressed as Links. A Link embraces and extends typed web links concept as a means of expressing relationships between Resources. A Link consists of a set of Parameters that define:

– a context URI,

– a target URI,

– a relation from the context URI to the target URI, and

– elements that provide metadata about the target URI, the relationship or the context of the Link.

The target URI is mandatory and the other items in a Link are optional. Additional items in the Link may be made mandatory based on the use of the links in different contexts (e.g. in Collections, in discovery, in bridging etc.). OpenAPI 2.0 schema for the Link payload is provided in Annex D.

An example of a Link is:

```
{"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",
"oic.if.baseline"], "p": {"bm": 3}, "rel": "item"}
```

Two Links are distinct from each other when at least one Parameter is different. For example the two Links show here are distinct and can appear in the same list of Links.


```
{"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",
"oic.if.baseline"], "p": {"bm": 2}, "rel": "item"}
{"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",
"oic.if.baseline"], "p": {"bm": 2}}
```

The documentmay mandate Parameters and Parameter values as required for certain capabilities. For all Links returned in a response to a RETRIEVE on "/oic/res", if a Link does not explicitly include the "rel" Parameter, a value of "rel"="hosts" shall be assumed. The relation value of "hosts" is defined by IETF RFC 6690, the value of "item" by IETF RFC 6573, and the value of "self" by IETF RFC 4287 and all are registered in the IANA Registry for Link Relations defined in IANA Link Relations.

As shown in Annex D the relation between the context URI and target URI in a Link is specified using the "rel" JSON element and the value of this element specifies the particular relation.

The context URI of the Link shall implicitly be the URI of the Resource (or specifically a Collection) that contains the Link unless the Link specifies the "anchor" Parameter. The "anchor" Parameter is used to change the context URI of a Link – the relationship with the target URI is based off the anchor URI when the "anchor" is specified. "Anchor" Parameter uses transfer protocol URI for OIC 1.1 Link (e.g. "anchor": "coaps://[fe80::b1d6]:44444") and OCF URI defined in Sec 6 for OCF 1.0 Links (e.g. "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989"). For optional backward compatibility with OIC 1.1, "anchor" Parameter uses transfer protocol URI for OIC 1.1 Link (e.g. "anchor": "coaps://[fe80::b1d6]:44444").

An example of using "anchors" in the context of Collections – a floor has rooms and rooms have lights – the lights may be defined in floor as Links but the Links will have the "anchor" set to the URI of the rooms that contain the lights (the relation is contains). This allows all lights in a floor to be turned on or off together while still having the lights defined with respect to the rooms that contain them (lights may also be turned on by using the room URI too). For example, here is the use of "anchor" in Link:

```
/a/floor {
  "links": [
    {
      "href": "/x/light1",
      "anchor": "/a/room1",   ** Note: /a/room1 has the item relationship with /x/light1; not /a/floor **
      "rel": "item"
    }
  ]
}

/a/room1 {
  "links": [
    {
      ** Note: /a/room1 "contains" the /x/light since /a/room1 is the implicit context URI **
      "href": "/x/light1",
      "rel": "item"
    }
  ]
}
```

7.8.2.2 Parameters

7.8.2.2.1 "ins" or Link Instance Parameter

The "ins" Parameter identifies a particular Link instance in a list of Links. The "ins" Parameter may be used to modify or delete a specific Link in a list of Links. The value of the "ins" Parameter is set at instantiation of the Link by the OCF Device (Server) that is hosting the list of Links – once it has been set, the "ins" Parameter shall not be modified for as long as the Link is a member of that list.

7.8.2.2.2 "p" or Policy Parameter

The Policy Parameter defines various rules for correctly accessing a Resource referenced by a target URI. The Policy rules are configured by a set of key-value pairs.

The policy Parameter "p" is defined by:

– "bm" key: The "bm" key corresponds to an integer value that is interpreted as an 8-bit bitmask. Each bit in the bitmask corresponds to a specific Policy rule. The rules are specified for "bm" in Table 10:

Table 10 – "bm" Property definition

Bit Position	Policy rule	Comment
Bit 0 (the LSB)	discoverable	The discoverable rule defines whether the Link is to be included in the Resource discovery message via "/oic/res". If the Link is to be included in the Resource discovery message, then "p" shall include the "bm" key and set the discoverable bit to value 1. If the Link is NOT to be included in the Resource discovery message, then "p" shall either include the "bm" key and set the discoverable bit to value 0 or omit the "bm" key entirely.
Bit 1 (2^{nd} LSB)	observable	The Observable rule defines whether the Resource referenced by the target URI supports the NOTIFY operation. With the self-link, i.e. the Link with "rel" value of "self", "/oic/res" can have a Link with the target URI of "/oic/res" and indicate itself Observable. The "self" is defined by IETF RFC 4287 and registered in the IANA Registry for "rel" value defined at IANA Link Relations. If the Resource supports the NOTIFY operation, then "p" shall include the "bm" key and set the Observable bit to value 1. If the Resource does NOT support the NOTIFY operation, then "p" shall either include the "bm" key and set the Observable bit to value 0 or omit the "bm" key entirely.
Bits 2-7	--	Reserved for future use. All reserved bits in "bm" shall be set to value 0.

NOTE If all the bits in "bm" are defined to value 0, then the "bm" key may be omitted entirely from "p" as an efficiency measure. However, if any bit is set to value 1, then "bm" shall be included in "p" and all the bits shall be defined appropriately.

– "sec" and "port" in the remaining bullets shall be used only in a response payload when the request does not include an OCF-Accept-Content-Format-Version option as defined in 12.2.5. In a payload sent in response to a request that includes an OCF-Accept-Content-Format-Version option "sec" and "port" shall not be used and instead the "eps" Parameter shall provide the information for an encrypted connection. See **Annex E** for the schema for the "p" Parameter that includes "sec" and "port".

– "sec" key: The "sec" key corresponds to a Boolean value that indicates whether the Resource referenced by the target URI is accessed via an encrypted connection. If "sec" is true, the Resource is accessed via an encrypted connection, using the "port" specified. If "sec" is false, the Resource is accessed via an unencrypted connection, or via an encrypted connection (if such a connection is made using the "port" settings for another Resource, for which "sec" is true).

– "port" key: The "port" key corresponds to an integer value that is used to indicate the port number where the Resource referenced by the target URI may be accessed via an encrypted connection.

– If the Resource is only available via an encrypted connection (i.e. DTLS over IP), then

- "p" shall include the "sec" key and its value shall be true.
- "p" shall include the "port" key and its value shall be the port number where the encrypted connection may be established.

– If the Resource is only available via an unencrypted connection, then

- "p" shall include the "sec" key and its value shall be false or "p" shall omit the "sec" key; the default value of "sec" is false.
- "p" shall omit the "port" key.

– If the Resource is available via both an encrypted and unencrypted connection, then

- "p" shall include the "sec" key and its value shall be false or "p" shall omit the "sec" key; the default value of "sec" is false.
- "p" may omit the "port" key. If the "port" key is omitted, the Resource shall be available using the same "port" information as another Resource on the Device for which "sec" is true.

– Access to the Resource on the port specified by the "port" key shall be made by an encrypted connection (e.g. "coaps://"). (Note that unencrypted connection to the Resource may be possible on a separate port discovered thru multicast discovery).

– Note that access to the Resource is controlled by the ACL for the Resource. A successful encrypted connection does not ensure that the requested action will succeed. See ISO/IEC 30118-2:2018 clause 12 for more information.

Example 1: This shows the Policy Parameter for a Resource that is discoverable but not Observable, and for which authenticated accesses shall be done via CoAPS port 33275.

```
"p": {"bm": 1}
```

Example 2: This shows a self-link, i.e. the "/oic/res" Link in itself that is discoverable and Observable.

```
{
  "href": "/oic/res",
  "rel": "self",
  "rt": ["oic.wk.res"],
  "if": ["oic.if.ll", "oic.if.baseline"],
  "p": {"bm": 3}
}
```

7.8.2.2.3 "type" or Media Type Parameter

The "type" Parameter may be used to specify the various media types that are supported by a specific target Resource. The default type of "application/vnd.ocf+cbor" shall be used when the "type" element is omitted. Once a Client discovers this information for each Resource, it may use one of the available representations in the appropriate header field of the Request or Response.

7.8.2.2.4 "di" or Device ID Parameter

The "di" Parameter specifies the Device ID of the Device that hosts the target Resource defined in the in the "href" Parameter.

The Device ID may be used to qualify a relative reference used in the "href" or to lookup OCF Endpoint information for the relative reference.

7.8.2.2.5 "eps" Parameter

The "eps" Parameter indicates the OCF Endpoint information of the target Resource.

"eps" shall have as its value an array of items and each item represents OCF Endpoint information with "ep" and "pri" as specified in 10.2. "ep" is mandatory but "pri" is optional.

This is an example of "eps" with multiple OCF Endpoints.

```
"eps": [
  {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
  {"ep": "coaps://[fe80::b1d6]:1122"},
  {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
]
```

When "eps" is present in a link, the OCF Endpoint information in "eps" can be used to access the target Resource referred by the "href" Parameter.

Note that the type of OCF Endpoint – Secure or Unsecure – that a Resource exposes merely determines the connection type(s) guaranteed to be available for sending requests to the Resource. For example, if a Resource only exposes a single CoAP "ep", it does not guarantee that the Resource cannot also be accessed via a Secure OCF Endpoint (e.g. via a CoAPS "ep" from another Resource's "eps information). Nor does exposing a given type of OCF Endpoint ensure that access to the Resource will be granted using the "ep" information. Whether requests to the Resource are granted or denied by the Access Control layer is separate from the "eps" information, and is determined by the configuration of the /acl2 Resource (see ISO/IEC 30118-2:2018 clause 13.5.3 for details).

When present, max-age information (e.g. Max-Age option for CoAP defined in IETF RFC 7252) determines the maximum time "eps" values may be cached before they are considered stale.

7.8.2.3 Formatting

When formatting in JSON, the list of Links shall be an array.

7.8.2.4 List of Links in a Collection

A Resource that exposes one or more Properties that are defined to be an array of Links where each Link can be discretely accessed is a Collection. The Property Name "links" is recommended for such an array of Links.

This is an example of a Resource with a list of Links.

```
/Room1
{
  "rt": ["my.room"],
  "if": ["oic.if.ll", "oic.if.baseline" ],
  "color": "blue",
  "links":
  [
    {
      "href": "/oic/d",
      "rt": ["oic.d.light", "oic.wk.d"],
      "if": [ "oic.if.r", "oic.if.baseline" ],
      "p": {"bm": 1}
    },
    {
      "href": "/oic/p",
      "rt": ["oic.wk.p"],
      "if": [ "oic.if.r", "oic.if.baseline" ],
      "p": {"bm": 1}
    },
    {
      "href": "/switch",
      "rt": ["oic.r.switch.binary"],
      "if": [ "oic.if.a", "oic.if.baseline" ],
      "p": {"bm": 3},
      "mt": [ "application/vnd.ocf+cbor", "application/exi+xml" ]
    },
```



```
    {
      "href": "/brightness",
      "rt": ["oic.r.light.brightness"],
      "if": [ "oic.if.a", "oic.if.baseline" ],
      "p": {"bm": 3}
    }
  ]
}
```

7.8.2.5 Properties describing an array of Links

If a Resource Type that defines an array of Links (e.g. Collections, Atomic Measurements) has restrictions on the "rt" values that can be within the array of Links, the Resource Type will define the "rts" Property. The "rts" Property as defined in Table 11 will include all "rt" values allowed for all Links in the array. If the Resource Type does not define the "rts" Property or the "rts" Property is an empty array, then any "rt" value is permitted in the array of Links.

For all instances of a Resource Type that defines the "rts" Property, the "rt" Link Parameter in every Link in the array of Links shall be one of the "rt" values that is included in the "rts" Property.

Table 11 – Resource Types Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Types	"rts"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	No	An array of Resource Types that are supported within an array of Links exposed by a Resource.

If a Resource Type that defines an array of Links has "rt" values which are required to be in the array, the Resource Type will define the "rts-m" Property, as defined in Table 12, which will contain all of the "rt" vaues that are required to be in the array of Links. If "rts-m" is defined, and "rts" is defined and is not an empty array, then the "rt" values present in "rts-m" will be part of the values present in "rts". Moreover, if the "rts-m" Property is defined, it shall be mandated (i.e. included in the "required" field of a JSON definition) in the Resource definition and Introspection Device Data (see 11.8).

For all instances of a Resource Type that defines the "rts-m" Property, there shall be at least one Link in the array of Links corresponding to each one of the "rt" values in the "rts-m" Property; for all such Links the "rt" Link Parameter shall contain at least one of the "rt" values in the "rts-m" Property.

Table 12 – Mandatory Resource Types Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Mandatory Resource Types	"rts-m"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	No	An array of Resource Types that are mandatory to be exposed within an array of Links exposed by a Resource.

7.8.3 Collections

7.8.3.1 Overview

A Resource that contains one or more references (specified as Links) to other Resources is a Collection. These references may be related to each other or just be a list; the Collection provides a means to refer to this set of references with a single handle (i.e. the URI). A simple Resource is kept distinct from a Collection. Any Resource may be turned into a Collection by binding Resource references as Links. Collections may be used for creating, defining or specifying hierarchies, indexes, groups, and so on.

A Collection shall have at least one Resource Type and at least one OCF Interface bound at all times during its lifetime. During creation time of a Collection the Resource Type and OCF Interfaces are specified. The initial defined Resource Types and OCF Interfaces may be updated during its life time. These initial values may be overridden using mechanism used for overriding in the case of a Resource. Additional Resource Types and OCF Interfaces may be bound to the Collection at creation or later during the lifecycle of the Collection.

A Collection shall define a Property that is an array with zero or more Links. The target URIs in the Links may reference another Collection or another Resource. The referenced Collection or Resource may reside on the same Device as the Collection that includes that Link (called a local reference) or may reside on another Device (called a remote reference). The context URI of the Links in the array shall (implicitly) be the Collection that contains that Property. The (implicit) context URI may be overridden with explicit specification of the "anchor" Parameter in the Link where the value of "anchor" is the new base of the Link.

A Resource may be referenced in more than one Collection, therefore, a unique parent-child relationship is not guaranteed. There is no pre-defined relationship between a Collection and the Resource referenced in the Collection, i.e., the application may use Collections to represent a relationship but none is automatically implied or defined. The lifecycles of the Collection and the referenced Resource are also independent of one another.

If the "drel" Property is defined for the Collection then all Links that don't explicitly specify a relationship shall inherit this default relationship in the context of that Collection. The default relationship defines the implicit relationship between the Collection and the target URI in the Link.

In the following example a Property "links" represents the list of Links in a Collection. The "links" Property has, as its value, an array of items and each item is a Link.

```
/my/house   ← This is IRI/URI of the Resosurce
{
  "rt": ["my.r.house"],   ← This and the next 3 lines are the Properies of the Resource
  "color": "blue",
  "n": "myhouse",
  "links": [
    {   ← This and the next 4 lines are the Parameters of a Liink
      "href": "/door",
      "rt": ["oic.r.door"],
      "if": ["oic.if.b", "oic.if.ll", "oic.if.baseline"]
    },

    {
      "href": "/door/lock",
      "rt": ["oic.r.lock"],
      "if": ["oic.if.b", "oic.if.baseline"],
      "type": ["application/cbor", "application/exi+xml"]
    },

    {
      "href": "/light",
```



```
        "rt": ["oic.r.light"],
        "if": ["oic.if.s", "oic.if.baseline"]
      },

      {
        "href": "/binarySwitch",
        "rt": ["oic.r.switch.binary"],
        "if": ["oic.if.a", "oic.if.baseline"],
        "type": ["application/cbor"]
      }

    ]
}
```

A Collection may be:

– A pre-defined Collection where the Collection has been defined a priori and the Collection is static over its lifetime. Such Collections may be used to model, for example, an appliance that is composed of other Devices or fixed set of Resources representing fixed functions.

– A Device local Collection where the Collection is used only on the Device that hosts the Collection. Such Collections may be used as a short-hand on a Client for referring to many Servers as one.

– A centralized Collection where the Collection is hosted on a Device but other Devices may access or update the Collection.

– A hosted Collection where the Collection is centralized but is managed by an authorized agent or party.

7.8.3.2 Collection Properties

A Collection shall define a Property that is an array of Links (the Property Name "links" is recommended). In addition, other Properties may be defined for the Collection by the Resource Type. The mandatory and recommended Common Properties for a Collection are shown in Table 13. This list of Common Properties is in addition to those defined for Resources in 7.3.2.

Table 13 – Common Properties for Collections (in addition to Common Properties defined in 7.3.2)

Property	Description	Property Name	Value Type	Mandatory
Links	The array of Links in the Collection	Per Resource Type definition	json Array of Links	Yes
Resource Types	The list of allowed Resource Types for Links in the Collection. If this Property is not defined or is null string then any Resource Type is permitted	As defined in Table 11	As defined in Table 11	No
Mandatory Resource Types	The list of Resource Types for Links that are mandatory in the Collection.	As defined in Table 12	As defined in Table 12	No

7.8.3.3 Default Resource Type

A default Resource Type, "oic.wk.col", is available for Collections. This Resource Type shall be used only when another type has not been defined on the Collection or when no Resource Type has been specified at the creation of the Collection.

The default Resource Type provides support for the Common Properties including an array of Links with the Property Name "links".

7.8.3.4 Default OCF Interface

All instances of a Collection shall support the links list ("oic.if.ll") OCF Interface in addition to the baseline ("oic.if.baseline") OCF Interface. An instance of a Collection may optionally support additional OCF Interfaces that are defined within this document. The Default OCF Interface for a Collection shall be links list ("oic.if.ll") unless otherwise specified by the Resource Type definition.

7.8.4 Atomic Measurement

7.8.4.1 Overview

Certain use cases require that the Properties of multiple Resources are only accessible as a group and individual access to those Properties of each Resource by a Client is prohibited. The Atomic Measurement Resource Type is defined to meet this requirement. This is accomplished through the use of the Batch OCF Interface.

7.8.4.2 Atomic Measurement Properties

An Atomic Measurement shall define a Property that is an array of Links (the Property Name "links" is recommended). In addition, other Properties may be defined for the Atomic Measurement by the Resource Type. The mandatory and recommended Common Properties for an Atomic Measurement are shown in Table 14. This list of Common Properties is in addition to those defined for Resources in 7.3.2.

Table 14 – Common Properties for Atomic Measurement (in addition to Common Properties defined in 7.3.2)

Property	Description	Property Name	Value Type	Mandatory
Links	The array of Links in the Atomic Measurement	Per Resource Type definition	json Array of Links	Yes
Resource Types	The list of allowed Resource Types for Links in the Atomic Measurement. If this Property is not defined or is null string then any Resource Type is permitted	As defined in Table 11	As defined in Table 11	No
Mandatory Resource Types	The list of Resource Types for Links that are mandatory in the Atomic Measurement.	As defined in Table 12	As defined in Table 12	No

7.8.4.3 Normative behaviour

The normative behaviour of an Atomic Measurement is as follows:

- The behaviour of the Batch OCF Interface ("oic.if.b") on the Atomic Measurement is defined as follows:
  - Only RETRIEVE and NOTIFY operations are supported, for Batch OCF Interface, on Atomic Measurement; the behavior of the RETRIEVE and NOTIFY operations shall be the same as specified in 7.6.3.4, with exceptions as provided for in 7.8.4.3.
  - The UPDATE operation is not allowed, for Batch OCF Interface, on Atomic Measurement; if an UPDATE operation is received, it shall result in a method not allowed error code.
  - An error response shall not include any representation of a linked Resource (i.e. empty response for all linked Resources).
- Any linked Resource within an Atomic Measurement (i.e. the target Resource of a Link in an Atomic Measurement) is subject to the following conditions:
  - Linked Resources within an Atomic Measurement and the Atomic Measurement itself shall exist on a single Server.
  - CRUDN operations shall not be allowed on linked Resources and shall result in a forbidden error code.
  - Linked Resources shall not expose the "oic.if.ll" OCF Interface. Since CRUDN operations are not allowed on linked Resources, the "oic.if.ll" OCF Interface would never be accessible.
- Links to linked Resources in an Atomic Measurement shall only be accessible through the "oic.if.ll" or the "oic.if.baseline" OCF Interfaces of an Atomic Measurement.
  - The linked Resources shall not be listed in "/oic/res".
- A linked Resource in an Atomic Measurement shall have defined one of "oic.if.a", "oic.if.s", "oic.if.r", or "oic.if.rw" as its Default OCF Interface.
- Not all linked Resources in an Atomic Measurement are required to be Observable. If an Atomic Measurement is being Observed using the "oic.if.b" OCF Interface, notification responses shall not be generated when the linked Resources which are not marked Observable are updated or change state.
- All linked Resources in an Atomic Measurement shall be included in every RETRIEVE and Observe response when using the "oic.if.b" OCF Interface.
- An Atomic Measurement shall support the "oic.if.b" and the "oic.if.ll" OCF Interfaces.
- Filtering of linked Resources in an Atomic Measurement is not allowed. Query parameters that select one or more individual linked Resources in a request to an Atomic Measurement shall result in a "forbidden" error code.
- If the "rel" Link Parameter is included in a Link contained in an Atomic Measurement, it shall have either the "hosts" or the "item" value.
- The Default OCF Interface of an Atomic Measurement is "oic.if.b".

7.8.4.4 Security considerations

Access rights to an Atomic Measurement Resource Type is as specified in clause 12.2.7.2 (ACL considerations for batch request to the Atomic Measurement Resource Type) of ISO/IEC 30118-2:2018).

7.8.4.5 Default Resource Type

The Resource Type is defined as "oic.wk.atomicmeasurement" as defined in Table 15.

Table 15 – Atomic Measurement Resource Type

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction	M/CR/O
none	Atomic Measurement	"oic.wk.atomicmeasurement"	"oic.if.ll" "oic.if.baseline" "oic.if.b"	A specialisation of the Collection pattern to ensure atomic RETRIEVAL of its referred Resources	RETRIEVE, NOTIFY	O

The Properties for Atomic Measurement are as defined in Table 16.

Table 16 – Properties for Atomic Measurement (in addition to Common Properties defined in 7.3.2)

Property	Description	Property name	Value Type	Mandatory
Links	The set of links that point to the linked Resources	Per Resource Type definition	json Array of Links	Yes

7.9 Third (3rd) party specified extensions

This clause describes how a 3rd party may add Device Types, Resource Types, 3rd party defined Properties to an existing or 3rd party defined Resource Type, 3rd party defined enumeration values to an existing enumeration and 3rd party defined Parameters to an existing defined Property.

A 3rd party may specify additional (non-OCF) Resources within an OCF Device. A 3rd party may also specify additional Properties within an existing OCF defined Resource Type. Further a 3rd party may extend an OCF defined enumeration with 3rd party defined values.

A 3rd party defined Device Type may expose both 3rd party and OCF defined Resource Types. A 3rd party defined Device Type must expose the mandatory Resources for all OCF Devices defined within this document.

A 3rd party defined Resource Type shall include any mandatory Properties defined in this documentand also any vertical specified mandatory Properties. All Properties defined within a 3rd party defined Resource Type that are part of the OCF namespace that are not Common Properties as defined in this documentshall follow the 3rd party defined Property rules in Table 17.

The following table defines the syntax rules for 3rd party defined Resource Type elements. Within the table the term "Domain_Name" refers to a domain name that is owned by the 3rd party that is defining the new element.

Table 17 – 3rd party defined Resource elements

	Resource Element	Vendor Definition Rules
New 3rd party defined Device Type	"rt" Property Value of "/oic/d"	"x.<Domain_Name>.<Resource identification>"
New 3rd party defined Resource Type	"rt" Property Value	"x.<Domain_Name>.<Resource identification>"
New 3rd party defined Property within the OCF namespace	Property Name	"x.<Domain_Name>.<Property>"

Additional 3rd party defined values in an OCF specified enumeration	Enumeration Property Value	"x.<Domain_Name>.<enum value>"
Additional 3rd party defined Parameter in an OCF specified Property	Parameter key word	x.<Domain_Name>.<parameter keyword>

With respect to the use of the Domain_Name in this scheme the labels are reversed from how they appear in DNS or other resolution mechanisms. The 3rd party defined Device Type and Resource Type otherwise follow the rules defined in clause 7.4.2. 3rd party defined Resource Types should be registered in the IANA Constrained RESTful Environments (CoRE) Parameters registry.

For example:

```
x.com.samsung.galaxyphone.accelerator
x.com.cisco.ciscorouterport
x.com.hp.printerhead
x.org.allseen.newinterface.newproperty
```

7.10 Query Parameters

7.10.1 Introduction

Properties and Parameters (including those that are part of a Link) may be used in the query part of a URI (see 6.2.2) as one criterion for selection of a particular Resource. This is done by declaring the Property (i.e. <Property Name> = <desired Property Value>) as one of the segments of the query. Only ASCII strings are permitted in query filters, and NULL characters are disallowed in query filters. This means that only Property Values with ASCII characters may be matched in a query filter.

The Resource is selected when all the declared Properties or Link Parameters in the query match the corresponding Properties or Link Parameters in the target.

7.10.2 Use of multiple parameters within a query

When a query contains multiple separate query parameters these are delimited by an "&" as described in 6.2.2.

A Client may apply multiple separate query parameters, for example "?ins=11111&rt=oic.r.switch.binary". If such queries are supported by the Server this shall be accomplished by matching "all of" the different query parameter types ("rt", "ins", "if", etc) against the target of the query. In the example, this resolves to an instance of oic.r.switch.binary that also has an "ins" populated as "11111". There is no significance applied to the order of the query parameters.

A Client may select more than one Resource Type using repeated query parameters, for example "?rt=oic.r.switch.binary&rt=oic.r.ramptime". If such queries are supported by the Server this shall be accomplished by matching "any of" the repeated query parameters against the target of the query. In the example, any instances of "oic.r.switch.binary" and/or "oic.r.ramptime" that may exist are selected.

A Client may combine both multiple repeated parameters and multiple separate parameters in a single query, for example "?if=oic.if.b&ins=11111&rt=oic.r.switch.binary&rt=oic.r.ramptime". If such queries are supported by the Server this shall be accomplished by matching "any of" the repeated query parameters and then matching "all of" the different query parameter types. In the example any instances of "oic.r.switch.binary" and/or "oic.r.ramptime" that also have an "ins" of "11111" that may exist are selected in a batch response.

NOTE The parameters within a query string are represented within the actual messaging protocol as defined in clause 11.9.

7.10.3 Application to multi-value "rt" Resources

An "rt" query for a multi-value "rt" Resource with the Default OCF Interface of "oic.if.a", "oic.if.s", "oic.if.r", "oic.if.rw" or "oic.if.baseline" is an extension of a generic "rt" query. When a Server receives a RETRIEVE request for a multi-value "rt" Resource with an "rt" query, (i.e. GET /ResExample?rt=oic.r.foo), the Server should respond only when the query value is an item of the "rt" Property Value of the target Resource and should send back only the Properties associated with the query value(s). For example, upon receiving GET /ResExample?rt=oic.r.switch.binary targeting a Resource with "rt": ["oic.r.switch.binary", "oic.r.light.brightness"], the Server responds with only the Properties of oic.r.switch.binary.

7.10.4 OCF Interface specific considerations for queries

7.10.4.1 OCF Interface selection

When an OCF Interface is to be selected for a request, it shall be specified as a query parameter in the URI of the Resource in the request message. If no query parameter is specified, then the Default OCF Interface shall be used. If the selected OCF Interface is not one of the permitted OCF Interfaces on the Resource then selecting that OCF Interface is an error and the Server shall respond with an error response code.

For example, the baseline OCF Interface may be selected by adding "if=oic.if.baseline" to the list of query parameters in the URI of the target Resource. For example: "GET /oic/res?if=oic.if.baseline".

7.10.4.2 Batch OCF Interface

See 7.6.3.4 for details on the batch OCF Interface itself. Query parameters may be used with the batch OCF Interface in order to select particular Resources in a Collection for retrieval or update; these parameters are used to select items in the Collection by matching Link Parameter Values.

When Link selection query parameters are used with RETRIEVE operations applied using the batch OCF Interface, only the Resources in the Collection with matching Link Parameters should be returned.

When Link selection query parameters are used with UPDATE operations applied using the batch OCF Interface, only the Resources having matching Link Parameters should be updated.

See 7.6.3.4.2 for examples of RETRIEVE and UPDATE operations that use Link selection query parameters.

8 CRUDN

8.1 Overview

CREATE, RETRIEVE, UPDATE, DELETE, and NOTIFY (CRUDN) are operations defined for manipulating Resources. These operations are performed by a Client on the Resources contained in n Server.

On reception of a valid CRUDN operation a Server hosting the Resource that is the target of the request shall generate a response depending on the OCF Interface included in the request; or based on the Default OCF Interface for the Resource Type if no OCF Interface is included.

CRUDN operations utilize a set of parameters that are carried in the messages and are defined in Table 18. A Device shall use CBOR as the default payload (content) encoding scheme for Resource representations included in CRUDN operations and operation responses; a Device may negotiate a different payload encoding scheme (e.g, see in 12.2.4 for CoAP messaging). Clauses 8.2 through 8.6 respectively specify the CRUDN operations and use of the parameters. The type definitions for these terms will be mapped in the clause 12 for each protocol.

Table 18 – Parameters of CRUDN messages

Applicability	Name	Denotation	Definition
All messages	*fr*	From	The URI of the message originator.
	to	To	The URI of the recipient of the message.
	ri	Request Identifier	The identifier that uniquely identifies the message in the originator and the recipient.
	cn	Content	Information specific to the operation.
Requests	*op*	Operation	Specific operation requested to be performed by the Server.
	obs	Observe	Indicator for an Observe request.
Responses	*rs*	Response Code	Indicator of the result of the request; whether it was accepted and what the conclusion of the operation was. The values of the response code for CRUDN operations shall conform to those as defined in clause 5.9 and 12.1.2 in IETF RFC 7252.
	obs	Observe	Indicator for an Observe response.

8.2 CREATE

8.2.1 Overview

The CREATE operation is used to request the creation of new Resources on the Server. The CREATE operation is initiated by the Client and consists of three steps, as depicted in Figure 10.

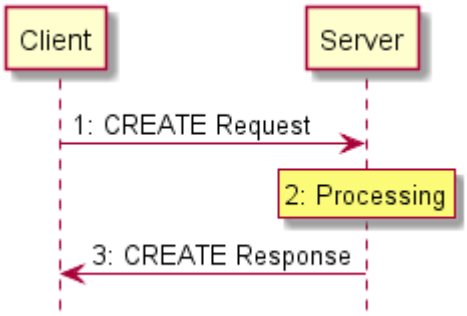


Figure 10 – CREATE operation

8.2.2 CREATE request

The CREATE request message is transmitted by the Client to the Server to create a new Resource by the Server. The CREATE request message will carry the following parameters:

- *fr*: Unique identifier of the Client
- *to*: URI of the target Resource responsible for creation of the new Resource.
- *ri*: Identifier of the CREATE request.
- *cn*: Information of the Resource to be created by the Server.
  - *cn* will include the URI and Resource Type Property of the Resource to be created.
  - *cn* may include additional Properties of the Resource to be created.
- *op*: CREATE

8.2.3 Processing by the Server

Following the receipt of a CREATE request, the Server may validate if the Client has the appropriate rights for creating the requested Resource. If the validation is successful, the Server creates the requested Resource. The Server caches the value of *ri* parameter in the CREATE request for inclusion in the CREATE response message.

8.2.4 CREATE response

The Server shall transmit a CREATE response message in response to a CREATE request message from a Client. The CREATE response message will include the following parameters.

- *fr*: Unique identifier of the Server
- *to*: Unique identifier of the Client
- *ri*: Identifier included in the CREATE request
- *cn*: Information of the Resource as created by the Server.
  - *cn* will include the URI of the created Resource.
  - *cn* will include the Resource representation of the created Resource.
- *rs*: The result of the CREATE operation.

8.3 RETRIEVE

8.3.1 Overview

The RETRIEVE operation is used to request the current state or representation of a Resource. The RETRIEVE operation is initiated by the Client and consists of three steps, as depicted in Figure 11.

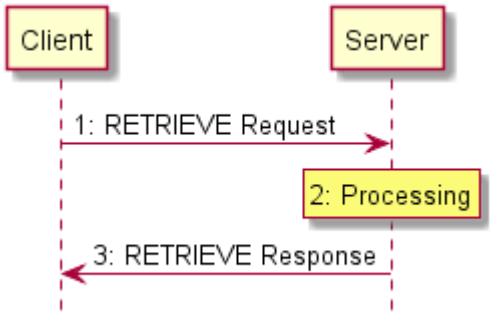


Figure 11 – RETRIEVE operation

8.3.2 RETRIEVE request

RETRIEVE request message is transmitted by the Client to the Server to request the representation of a Resource from a Server. The RETRIEVE request message will carry the following parameters.

- *fr*: Unique identifier of the Client.
- *to*: URI of the Resource the Client is targeting.
- *ri*: Identifier of the RETRIEVE request.
- *op*: RETRIEVE.

8.3.3 Processing by the Server

Following the receipt of a RETRIEVE request, the Server may validate if the Client has the appropriate rights for retrieving the requested data and the Properties are readable. The Server caches the value of *ri* parameter in the RETRIEVE request for use in the response

8.3.4 RETRIEVE response

The Server shall transmit a RETRIEVE response message in response to a RETRIEVE request message from a Client. The RETRIEVE response message will include the following parameters.

- *fr*: Unique identifier of the Server.
- *to*: Unique identifier of the Client.
- *ri*: Identifier included in the RETRIEVE request.
- *cn*: Information of the Resource as requested by the Client.
  - *cn* should include the URI of the Resource targeted in the RETRIEVE request.
- *rs*: The result of the RETRIEVE operation.

8.4 UPDATE

8.4.1 Overview

The UPDATE operation is either a Partial UPDATE or a complete replacement of the information in a Resource in conjunction with the OCF Interface that is also applied to the operation. The UPDATE operation is initiated by the Client and consists of three steps, as depicted in Figure 12.

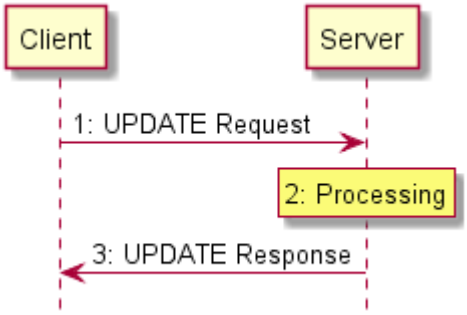


Figure 12 – UPDATE operation

8.4.2 UPDATE request

The UPDATE request message is transmitted by the Client to the Server to request the update of information of a Resource on the Server. The UPDATE request message will carry the following parameters.

- *fr*: Unique identifier of the Client.
- *to*: URI of the Resource targeted for the information update.
- *ri*: Identifier of the UPDATE request.
- *op*: UPDATE.
- *cn*: Information, including Properties, of the Resource to be updated at the target Resource.

8.4.3 Processing by the Server

8.4.3.1 Overview

Following the receipt of an UPDATE request, the Server may validate if the Client has the appropriate rights for updating the requested data. If the validation is successful the Server updates the target Resource information according to the information carried in *cn* parameter of the UPDATE request message. The Server caches the value of *ri* parameter in the UPDATE request for use in the response.

An UPDATE request that includes Properties that are read-only shall be rejected by the Server with an *rs* indicating a bad request.

An UPDATE request shall be applied only to the Properties in the target Resource visible via the applied OCF Interface that support the operation. An UPDATE of non-existent Properties is ignored.

An UPDATE request shall be applied to the Properties in the target Resource even if those Property Values are the same as the values currently exposed by the target Resource.

8.4.3.2 Resource monitoring by the Server

The Server shall monitor the state the Resource identified in the Observe request from the Client. Anytime there is a change in the state of the Observed Resource or an UPDATE operation applied to the Resource, the Server sends another RETRIEVE response with the Observe indication. The mechanism does not allow the Client to specify any bounds or limits which trigger a notification, the decision is left entirely to the Server.

8.4.3.3 Additional RETRIEVE responses with Observe indication

The Server shall transmit updated RETRIEVE response messages following Observed changes in the state of the Resources requested by the Client. The RETRIEVE response message shall include the parameters listed in 11.4.2.4.

8.4.4 UPDATE response

The UPDATE response message will include the following parameters:

- *fr*: Unique identifier of the Server.
- *to*: Unique identifier of the Client.
- *ri*: Identifier included in the UPDATE request.
- *rs*: The result of the UPDATE request.

The UPDATE response message may also include the following parameters:

- cn: The Resource representation following processing of the UPDATE request.

8.5 DELETE

8.5.1 Overview

The DELETE operation is used to request the removal of a Resource. The DELETE operation is initiated by the Client and consists of three steps, as depicted in Figure 13.

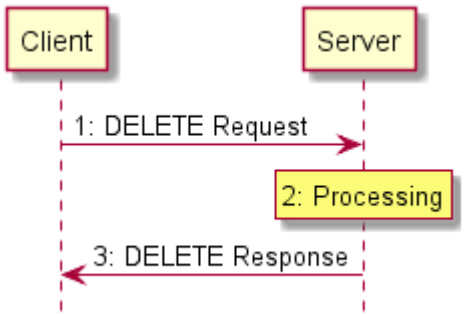


Figure 13 – DELETE operation

8.5.2 DELETE request

DELETE request message is transmitted by the Client to the Server to delete a Resource on the Server. The DELETE request message will carry the following parameters:

– *fr*: Unique identifier of the Client.

– *to*: URI of the target Resource which is the target of deletion.

– *ri*: Identifier of the DELETE request.

– *op*: DELETE.

8.5.3 Processing by the Server

Following the receipt of a DELETE request, the Server may validate if the Client has the appropriate rights for deleting the identified Resource, and whether the identified Resource exists. If the validation is successful, the Server removes the requested Resource and deletes all the associated information. The Server caches the value of *ri* parameter in the DELETE request for use in the response.

8.5.4 DELETE response

The Server shall transmit a DELETE response message in response to a DELETE request message from a Client. The DELETE response message will include the following parameters.

– *fr*: Unique identifier of the Server.

– *to*: Unique identifier of the Client.

– *ri*: Identifier included in the DELETE request.

– *rs*: The result of the DELETE operation.

8.6 NOTIFY

8.6.1 Overview

The NOTIFY operation is used to request asynchronous notification of state changes. Complete description of the NOTIFY operation is provided in 11.4. The NOTIFY operation uses the NOTIFICATION response message which is defined here.

8.6.2 NOTIFICATION response

The NOTIFICATION response message is sent by a Server to notify the URLs identified by the Client of a state change. The NOTIFICATION response message carries the following parameters.

– *fr*: Unique identifier of the Server.

– *to*: URI of the Resource target of the NOTIFICATION message.

– *ri*: Identifier included in the CREATE request.

– *op*: NOTIFY.

– *cn*: The updated state of the Resource.

9 Network and connectivity

9.1 Introduction

The Internet of Things is comprised of a wide range of applications which sense and actuate the physical world with a broad spectrum of device and network capabilities: from battery powered nodes transmitting 100 bytes per day and able to last 10 years on a coin cell battery, to mains powered nodes able to maintain Megabit video streams. It is estimated that many 10s of billions of IoT devices will be deployed over the coming years.

It is desirable that the connectivity options be adapted to the IP layer. To that end, IETF has completed considerable work to adapt Bluetooth®, Wi-Fi, 802.15.4, LPWAN, etc. to IPv6. These adaptations, plus the larger address space and improved address management capabilities, make IPv6 the clear choice for the OCF network layer technology.

9.2 Architecture

While the aging IPv4 centric network has evolved to support complex topologies, its deployment was primarily provisioned by a single Internet Service Provider (ISP) as a single network. More complex network topologies, often seen in residential home, are mostly introduced through the acquisition of additional home network devices, which rely on technologies like private Network Address Translation (NAT). These technologies require expert assistance to set up correctly and should be avoided in a home network as they most often result in breakage of constructs like routing, naming and discovery services.

The multi-segment ecosystem OCF addresses will not only cause a proliferation of new devices and associated routers, but also new services introducing additional edge routers. All these new requirements require advance architectural constructs to address complex network topologies like the one shown in Figure 14.

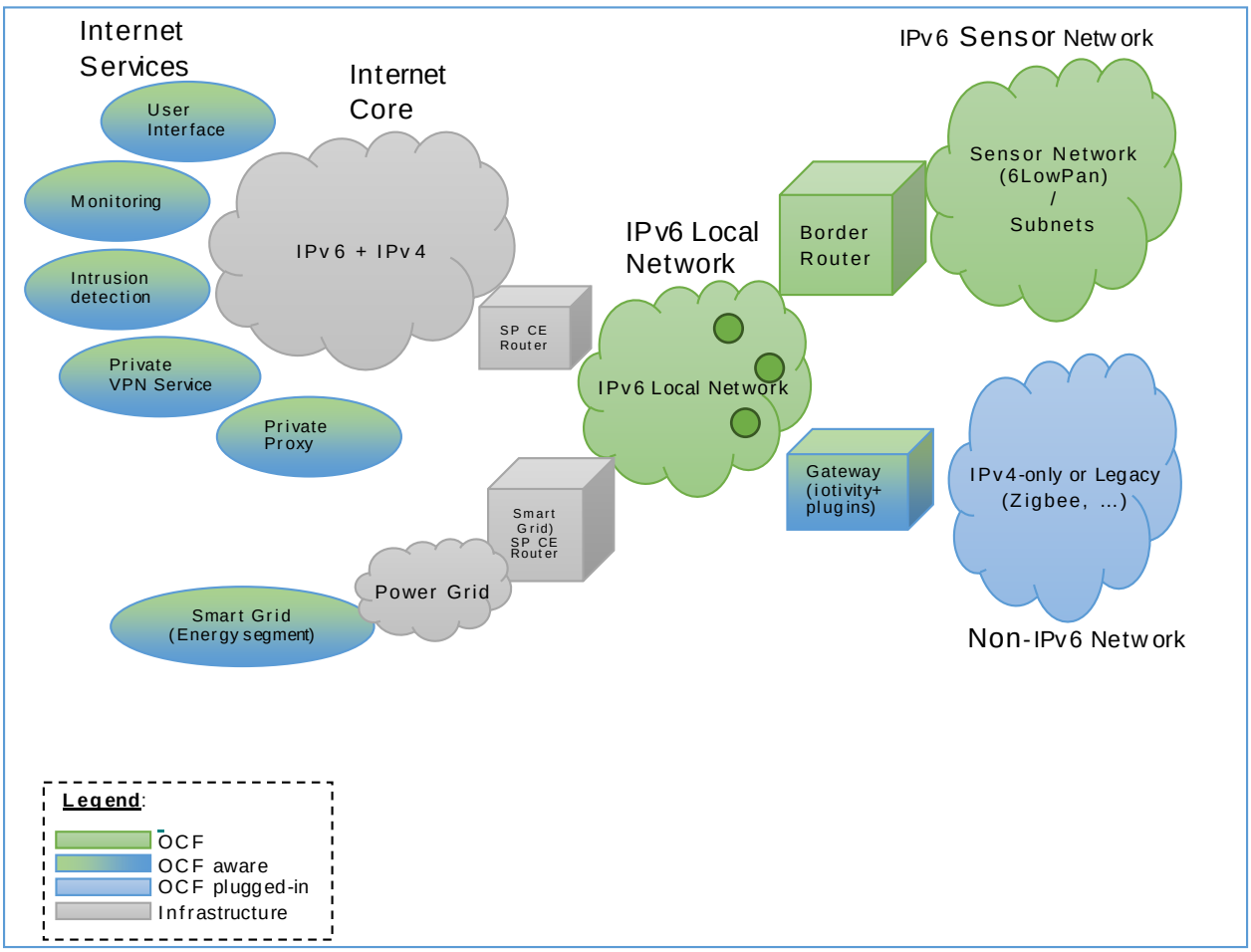


Figure 14 – High Level Network & Connectivity Architecture

In terms of IETF RFC 6434, IPv6 nodes assume either a router or host role. Nodes may further implement various specializations of those roles:

– A Router may implement Customer Edge Router capabilities as defined in IETF RFC 7084.

– Nodes limited in processing power, memory, non-volatile storage or transmission capacity requires special IP adaptation layers (6LoWPAN) and/or dedicated routing protocols (RPL). Examples include devices transmitting over low power physical layer like IEEE 802.14.5, ITU G9959, Bluetooth Low Energy, DECT Ultra Low Energy, and Near Field Communication (NFC).

– A node may translate and route messaging between IPv6 and non-IPv6 networks.

9.3 IPv6 network layer requirements

9.3.1 Introduction

Projections indicate that many 10s of billions of new IoT endpoints and related services will be brought online in the next few years. These endpoint's capabilities will span from battery powered nodes with limited compute, storage, and bandwidth to more richly resourced devices operating over Ethernet and WiFi links.

Internet Protocol version 4 (IPv4), deployed some 30 years ago, has matured to support a wide variety of applications such as Web browsing, email, voice, video, and critical system monitoring and control. However, the capabilities of IPv4 are at the point of exhaustion, not the least of which is that available address space has been consumed.

The IETF long ago saw the need for a successor to IPv4, thus the development of IPv6. OCF recommends IPv6 at the network layer. Amongst the reasons for IPv6 recommendations are:

– Larger address space. Side-effect: greatly reduce the need for NATs.

– More flexible addressing architecture. Multiple addresses and types per interface: Link-local, ULA, GUA, variously scoped Multicast addresses, etc. Better ability to support multi-homed networks, better re-numbering capability, etc.

– More capable auto configuration capabilities: DHCPv6, SLAAC, Router Discovery, etc.

– Technologies enabling IP connectivity on constrained nodes are based upon IPv6.

– All major consumer operating systems (IoS, Android, Windows, Linux) are already IPv6 enabled.

– Major Service Providers around the globe are deploying IPv6.

9.3.2 IPv6 node requirements

9.3.2.1 Introduction

In order to ensure network layer services interoperability from node to node, mandating a common network layer across all nodes is vital. The protocol should enable the network to be: secure, manageable, and scalable and to include constrained and self-organizing meshed nodes. OCF mandates IPv6 as the common network layer protocol to ensure interoperability across all Devices. More capable Devices may also include additional protocols creating multiple-stack Devices. The remainder of this clause will focus on interoperability requirements for IPv6 hosts, IPv6 constrained hosts and IPv6 routers. The various protocol translation permutations included in multi-stack gateway devices may be addresses in subsequent addendums of this document.

9.3.2.2 IP Layer

An IPv6 node shall support IPv6 and it shall conform to the requirements as specified in IETF RFC 6434.

10 OCF Endpoint

10.1 OCF Endpoint definition

The specific definition of an OCF Endpoint depends on the Transport Protocol Suite being used. For the example of CoAP over UDP over IPv6, the OCF Endpoint is identified by an IPv6 address and UDP port number.

Each Device shall associate with at least one OCF Endpoint with which it can exchange request and response messages. When a message is sent to an OCF Endpoint, it shall be delivered to the Device which is associated with the OCF Endpoint. When a request message is delivered to an OCF Endpoint, path component is enough to locate the target Resource.

A Device can be associated with multiple OCF Endpoints. For example, an Device can have several IP addresses or port numbers or support both CoAP and HTTP transfer protocol. Different Resources in an Device may be accessed with the same OCF Endpoint or need different ones. Some Resources may use one OCF Endpoint and others a different one. It depends on an implementation.

On the other hand, an OCF Endpoint can be shared among multiple Devices, only when there is a way to clearly designate the target Resource with request URI. For example, when multiple CoAP servers use uniquely different URI paths for all their hosted Resources, and the CoAP implementation demultiplexes by path, they can share the same CoAP OCF Endpoint. However, this is not possible in this version of the document, because a pre-determined URI (e.g. "/oic/d") is mandatory for some mandatory Resources (e.g. "oic.wk.d").

10.2 OCF Endpoint information

10.2.1 Introduction

OCF Endpoint is represented by OCF Endpoint information which consists of two items of key-value pair, "ep" and "pri".

10.2.2 "ep"

"ep" represents Transport Protocol Suite and OCF Endpoint Locator specified as follows:

– *Transport Protocol Suite* - a combination of protocols (e.g. CoAP + UDP + IPv6) with which request and response messages can be exchanged for RESTful transaction (i.e. CRUDN). A Transport Protocol Suite shall be indicated by a URI scheme name. All scheme names supported by this documentare IANA registered, these are listed in Table 19. A vendor may also make use of a non-IANA registered scheme name for their own use (e.g. "com.example.foo"), this shall follow the syntax for such scheme names defined by IETF RFC 7595. The behaviour of a vendor-defined scheme name is undefined by this document. All OCF defined Resource Types when exposing OCF Endpoint Information in an "eps" (see 10.2.4) shall include at least one "ep" with a Transport Protocol Suite as defined in Table 19.

– *OCF Endpoint Locator* – an address (e.g. IPv6 address + Port number) or an indirect identifier (e.g., DNS name) resolvable to an IP address, through which a message can be sent to the OCF Endpoint and in turn associated Device. The OCF Endpoint Locator for "coap" and "coaps" shall be specified as "IP address: port number". The OCF Endpoint Locator for "coap+tcp" or "coaps+tcp" shall be specified as "IP address: port number" or "DNS name: port number" or "DNS name" such that the DNS name shall be resolved to a valid IP address for the target Resource with a name resolution service (i.e., DNS). For the 3rd case, when the port number is omitted, the default port "5683" (and "5684") shall be assumed for "coap+tcp" (and for "coaps+tcp") scheme respectively as defined in IETF RFC 8323.Temporary addresses should not be used because OCF Endpoint Locators are for the purpose of accepting incoming sessions, whereas temporary addresses are for initiating outgoing sessions (IETF RFC 4941). Moreover, its inclusion in "/oic/res" can cause a privacy concern (IETF RFC 7721).

"ep" shall have as its value a URI (as specified in IETF RFC 3986) with the scheme component indicating Transport Protocol Suite and the authority component indicating the OCF Endpoint Locator.

An "ep" example for "coap" and "coaps" is as illustrated:

```
"ep": "coap://[fe80::b1d6]:1111"
```

An "ep" example for "coap+tcp" and "coaps+tcp" is as illustrated:


```
"ep": "coap+tcp://[2001:db8:a::123]:2222"
"ep": "coap+tcp://foo.bar.com:2222"
"ep": "coap+tcp://foo.bar.com"
```

The current list of "ep" with corresponding Transport Protocol Suite is shown in Table 19:

Table 19 – "ep" value for Transport Protocol Suite

Transport Protocol Suite	scheme	OCF Endpoint Locator	"ep" Value example
coap+udp+ip	"coap"	IP address + port number	"coap://[fe80::b1d6]:1111"
coaps + udp + ip	"coaps"	IP address + port number	"coaps://[fe80::b1d6]:1122"
coap + tcp + ip	"coap+tcp"	IP address + port number DNS name: port number DNS name	"coap+tcp://[2001:db8:a::123]:2222" "coap+tcp://foo.bar.com:2222" "coap+tcp://foo.bar.com"
coaps + tcp + ip	"coaps+tcp"	IP address + port number DNS name: port number DNS name	"coaps+tcp://[2001:db8:a::123]:2233" "coaps+tcp://[2001:db8:a::123]:2233" "coaps+tcp://foo.bar.com:2233"

10.2.3 "pri"

When there are multiple OCF Endpoints, "pri" indicates the priority among them.

"pri" shall be represented as a positive integer (e.g. "pri": 1) and the lower the value, the higher the priority.

The default "pri" value is 1, i.e. when "pri" is not present, it shall be equivalent to "pri": 1.

10.2.4 OCF Endpoint information in "eps" Parameter

To carry OCF Endpoint information, a new Link Parameter "eps" is defined in 7.8.2.2.5. "eps" has an array of items as its value and each item represents OCF Endpoint information with two key-value pairs, "ep" and "pri", of which "ep" is mandatory and "pri" is optional.

OCF Endpoint Information in an "eps" Parameter is valid for the target Resource of the Link, i.e., the Resource referred by "href" Parameter. OCF Endpoint information in an "eps" Parameter may be used to access other Resources on the Device, but such access is not guaranteed.

A Client may resolve the "ep" value to an IP address for the target Resource, i.e., the address to access the Device which hosts the target Resource. A valid (transfer protocol) URI for the target Resource can be constructed with the scheme, host and port components from the "ep" value and the "path" component from the "href" value.

Links with an "eps":

```
{
  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9 ",
  "href": "/myLightSwitch",
  "rt": ["oic.r.switch.binary"],
  "if": ["oic.if.a", "oic.if.baseline"],
```



```
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
      {"ep": "coaps://[fe80::b1d6]:1122"}
    ]
  }

  {
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
    "href": "/myTemperature",
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap+tcp://foo.bar.com", "pri": 2},
      {"ep": "coaps+tcp://foo.bar.com:1122"}
    ]
  }
```

In the previous example, "anchor" represents the hosting Device, "href", target Resource and "eps" the two OCF Endpoints for the target Resource. The (fully-qualified) URIs for the target Resource are as illustrated:

```
coap://[fe80::b1d6]:1111/myLightSwitch
coaps://[fe80::b1d6]:1122/myLightSwitch
coap+tcp://foo.bar.com:5683/myTemperature
```

coaps+tcp://foo.bar.com:1122/myTemperatureIf the target Resource of a Link requires a secure connection (e.g. CoAPS), "eps" Parameter shall be used to indicate the necessary information (e.g. port number) in OCF 1.0 payload. For optional backward compatibility with OIC 1.1, the "sec" and "port" shall only be used in OIC 1.1 payload.

10.3 OCF Endpoint discovery

10.3.1 Introduction

OCF Endpoint discovery is defined as the process for a Client to acquire the OCF Endpoint information for Device or Resource.

10.3.2 Implicit discovery

If a Device is the source of a CoAP message (e.g. "/oic/res" response), the source IP address and port number may be combined to form the OCF Endpoint Locator for the Device. Along with a "coap" scheme and default "pri" value, OCF Endpoint information for the Device may be constructed.

In other words, a "/oic/res" response message with CoAP may implicitly carry the OCF Endpoint information of the responding Device and in turn all the hosted Resources, which may be accessed with the same transfer protocol of CoAP. In the absence of an "eps" Parameter, a Client shall be able to utilize implicit discovery to access the target Resource.

10.3.3 Explicit discovery with "/oic/res" response

OCF Endpoint information may be explicitly indicated with the "eps" Parameter of the Links in "/oic/res".

As in 10.3.2, an "/oic/res" response may implicitly indicate the OCF Endpoint information for some Resources hosted by the responding Device. However implicit discovery, i.e., inference of OCF Endpoint information from CoAP response message, may not work for some Resources on the same Device. For example, some Resources may allow only secure access via CoAPS which requires the "eps" Parameter to indicate the port number. Moreover "/oic/res" may expose a target Resource which belongs to another Device.

When the OCF Endpoint for a target Resource of a Link cannot be implicitly inferred, the "eps" Parameter shall be included to provide explicit OCF Endpoint information with which a Client can access the target Resource. In the presence of the "eps" Parameter, a Client shall be able to utilize it to access the target Resource. For "coap" and "coaps", a Client may use the IP address in the "ep" value in the "eps" Parameter to access the target Resource. For "coap+tcp" and "coaps+tcp", a Client may use the IP address in the "eps" Parameter or resolve the DNS name in the "eps" Parameter to acquire a valid IP address for the target Resource. If "eps" Parameter omits the port number, then the default port "5683" (and "5684") shall be assumed for "coap+tcp" (and "coaps+tcp") scheme as defined in IETF RFC 8323.To access the target Resource of a Link, a Client may use the "eps" Parameter in the Link, if it is present and fall back on implicit discovery if not.

This applies to the case of "/oic/res" for a Resource Directory or Bridge Device which usually carries the Links for Resources which another Device hosts.

This is an example of an "/oic/res" response from a Bridge Device with two Bridged Devices, having the "eps" Parameter in Links.

```
[
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/res",
    "rel": "self",
    "rt": ["oic.wk.res"],
    "if": ["oic.if.ll", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:55555"},
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/d",
    "rt": ["oic.wk.d", "oic.d.bridge"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:55555"},
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/oic/p",
    "rt": ["oic.wk.p"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:55555"},
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
    ]
  },
  {
    "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "href": "/mySecureMode",
    "rt": ["oic.r.securemode"],
    "if": ["oic.if.rw", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
```



```
        {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
    },
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/sec/doxm",
      "rt": ["oic.r.doxm"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:55555"},
        {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
    },
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/sec/pstat",
      "rt": ["oic.r.pstat"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
    },
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/sec/cred",
      "rt": ["oic.r.cred"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
    },
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/sec/acl2",
      "rt": ["oic.r.acl2"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
    },
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/myIntrospection",
      "rt": ["oic.wk.introspection"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/res",
      "rt": ["oic.wk.res"],
      "if": ["oic.if.ll", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:66666"},
        {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
```



```
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/d",
      "rt": ["oic.wk.d", "oic.d.light", "oic.d.virtual"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:66666"},
        {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/p",
      "rt": ["oic.wk.p"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:66666"},
        {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/myLight",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/sec/doxm",
      "rt": ["oic.r.doxm"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:66666"},
        {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/sec/pstat",
      "rt": ["oic.r.pstat"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/sec/cred",
      "rt": ["oic.r.cred"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
    },
```



```
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/sec/acl2",
      "rt": ["oic.r.acl2"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/myLightIntrospection",
      "rt": ["oic.wk.introspection"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [{"ep": "coaps://[2001:db8:a::b1d4]:22222"}]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/res",
      "rt": ["oic.wk.res"],
      "if": ["oic.if.ll", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:77777"},
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
       "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/d",
      "rt": ["oic.wk.d", "oic.d.fan", "oic.d.virtual"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:77777"},
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/p",
      "rt": ["oic.wk.p"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:77777"},
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/myFan",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
```



```
      "href": "/oic/sec/doxm",
      "rt": ["oic.r.doxm"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:77777"},
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/sec/pstat",
      "rt": ["oic.r.pstat"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/sec/cred",
      "rt": ["oic.r.cred"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/sec/acl2",
      "rt": ["oic.r.acl2"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/myFanIntrospection",
      "rt": ["oic.wk.introspection"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    }
  ]
```

The exact format of the "/oic/res" response and a way for a Client to acquire a "/oic/res" response message is specified in Annex D and 11.3.5 respectively.

10.4 CoAP based OCF Endpoint discovery

The following describes CoAP based OCF Endpoint discovery:

– Devices shall join the *All OCF Nodes* multicast groups (as defined in [IANA IPv6 Multicast Address Space Registry]) with scopes 2, 3, and 5 (i.e., ff02::158, ff03::158 and ff05::158) and shall listen on the port 5683. For compliance to IETF RFC 7252 a Device may additionally join the *All CoAP Nodes* multicast groups.

– Clients intending to discover Resources shall join the multicast groups as defined in a).

– Devices shall expose "/oic/res" via an unsecured OCF Endpoint.

– Clients shall send discovery requests (GET request) to the *All OCF Nodes* multicast group address with scope 2 ("ff02::158") at port "5683". The requested URI shall be "/oic/res". For compliance to IETF RFC 7252 a Client may additionally send to the *All CoAP Nodes* multicast groups.

– If the discovery request is intended for a specific Resource Type, the query parameter "rt" shall be included in the request (see 6.2.2) with its value set to the desired Resource Type. Only Devices hosting the Resource Type shall respond to the discovery request.

– When the "rt" query parameter is omitted, all Devices shall respond to the discovery request.

– Handling of multicast requests shall be as described in clause 8 of IETF RFC 7252 and clause 4.1 in IETF RFC 6690.

– Devices which receive the request shall respond using CBOR payload encoding. A Device shall indicate support for CBOR payload encoding for multicast discovery as described in 12.4.

11 Functional interactions

11.1 Introduction

The functional interactions between a Client and a Server are described in 11.2 through 11.9 respectively. The functional interactions use CRUDN messages (clause 8) and include Discovery, Notification, and Device management. These functions require support of core defined Resources as defined in Table 20.

Table 20 – List of Core Resources

Pre-defined URI	Resource Name	Resource Type	Related Functional Interaction	Mandatory
"/oic/res"	Default	"oic.wk.res"	Discovery	Yes
"/oic/p"	Platform	"oic.wk.p"	Discovery	Yes
"/oic/d"	Device	"oic.wk.d"	Discovery	Yes
(none)	Configuration	"oic.wk.con"	Device management	No
"/oic/mnt"	Maintenance	"oic.wk.mnt"	Device management	No

11.2 Onboarding, Provisioning and Configuration

Onboarding and Provisioning are fully defined by the ISO/IEC 30118-2:2018.

Should a Device support Client update of configurable information it shall do so via exposing an oic.wk.con Core Resource (Table 21) in "/oic/res".

Table 21 – Configuration Resource

Example URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/example/oic/con"	Device Configuration	"oic.wk.con"	"oic.if.rw"	The Resource Type through which configurable information specific to the Device is exposed. The Resource Properties exposed in "oic.wk.con" are listed in Table 22.	Configuration

"/example/oic/con"	Platform Configuration	"oic.wk.con.p"	"oic.if.rw"	The optional Resource Type through which configurable information specific to the Platform is exposed. The Properties exposed in "oic.wk.con.p" are listed in Table 23.	Configuration

Table 22 defines the "oic.wk.con" Resource Type.

Table 22 – "oic.wk.con" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
(Device) Name	"n" (Common Property of "/example/oic/con")	"string"	N/A	N/A	R, W	Yes	Human friendly name configurable by the end user (e.g. Bob's thermostat). The "n" Common Property of the oic.wk.con Core Resource and the "n" Common Property of the "/oic/d" Core Resource shall have the same Value. When the "n" Common Property Value of the oic.wk.con Core Resource is modified, it shall be reflected to the "n" Common Property of "/oic/d" Core Resource.
Location	"loc"	array of float (has two elements, the first is latitude, the second is longitude)	N/A	Degrees	R, W	No	Provides location information where available.
Location Name	"locn"	"string"	N/A	N/A	R, W	no	Human friendly name for location For example, "Living Room".
Currency	"c"	"string"	N/A	N/A	R,W	no	Indicates the currency that is used for any monetary transactions
Region	"r"	"string"	N/A	N/A	R,W	no	Free form text Indicating the current region in which the Device is located geographically.
Localized Names	"ln"	"array"	N/A	N/A	R,W	no	Human-friendly name of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the Device name in the indicated language. If this Property and the Device Name (n) Property are both supported, the Device Name (n) value shall be included in this array.
Default Language	"dl"	"language-tag"	N/A	N/A	R,W	no	The default language supported by the Device, specified as an IETF RFC 5646 language tag. By default, clients can treat any string Property as being in this

							language unless the Property specifies otherwise.

Table 23 defines the "oic.wk.con.p" Resource Type.

Table 23 – "oic.wk.con.p" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Platform Names	"mnpn"	"array"	N/A	N/A	R,W	No	Friendly name of the Platform. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the platform friendly name in the indicated language. For example, [{"language":"en", "value":"Dave's Laptop"}]

11.3 Resource discovery

11.3.1 Introduction

Discovery is a function which enables OCF Endpoint discovery as well as Resource based discovery. OCF Endpoint discovery is described in detail in clause 10. This clause mainly describes the Resource based discovery.

11.3.2 Resource based discovery: mechanisms

11.3.2.1 Overview

As part of discovery, a Client may find appropriate information about other OCF peers. This information could be instances of Resources, Resource Types or any other information represented in the Resource model that an OCF peer would want another OCF peer to discover.

At the minimum, Resource based discovery uses the following:

- A Resource to enable discovery shall be defined. The representation of that Resource shall contain the information that can be discovered.
- The Resource to enable discovery shall be specified and commonly known a-priori. A Device for hosting the Resource to enable discovery shall be identified.
- A mechanism and process to publish the information that needs to be discovered with the Resource to enable discovery.
- A mechanism and process to access and obtain the information from the Resource to enable discovery. A query may be used in the request to limit the returned information.
- A scope for the publication.
- A scope for the access.
- A policy for visibility of the information.

Depending on the choice of the base aspects, the Framework defines three Resource based discovery mechanisms:

- Direct discovery, where the Resources are published locally at the Device hosting the Resources and are discovered through peer inquiry.
- Indirect discovery, where Resources are published at a third party assisting with the discovery and peers publish and perform discovery against the Resource to enable discovery on the assisting 3rd party.
- Advertisement discovery, where the Resource to enable discovery is hosted local to the initiator of the discovery inquiry but remote to the Devices that are publishing discovery information.

A Device shall support direct discovery.

11.3.2.2 Direct discovery

In direct discovery,

- The Device that is providing the information shall host the Resource to enable discovery.
- The Device publishes the information available for discovery with the local Resource to enable discovery (i.e. local scope).
- Clients interested in discovering information about this Device shall issue RETRIEVE requests directly to the Resource. The request may be made as a unicast or multicast. The request may be generic or may be qualified or limited by using appropriate queries in the request.
- The Server Device that receives the request shall send a response with the discovered information directly back to the requesting Client Device.
- The information that is included in the request is determined by the policies set for the Resource to be discovered locally on the responding Device.

11.3.2.3 Indirect discovery of Resources (Resource Directory based discovery)

In indirect discovery the information about the resource to be discovered is hosted on a Server that is not hosting the Resource. See 11.3.6 for details on Resource Directory based discovery.

In indirect discovery:

- The Resource to be discovered is hosted on a Device that is neither the Client initiating the discovery nor the Device that is providing or publishing the information to be discovered. This Device may use the same Resource to provide discovery for multiple agents looking to discover and for multiple agents with information to be discovered.
- The Device to be discovered or with information to discover, publishes that information with Resource to be discovered on a different Device. The policies on the information shared including the lifetime/validity are specified by the publishing Device. The publishing Device may modify these policies as required.
- The Client doing the discovery may send a unicast discovery request to the Device hosting the discovery information or send a multicast request that shall be monitored and responded to by the Device. In both cases, the Device hosting the discovery information is acting on behalf of the publishing Device.
- The discovery policies may be set by the Device hosting the discovery information or by the party that is publishing the information to be discovered. The discovery information that is returned in the discovery response shall adhere to the policies that are in effect at the time of the request.

11.3.2.4 Advertisement Discovery

In advertisement discovery:

- The Resource to enable discovery is hosted local to the Device that is initiating the discovery request (Client). The Resource to enable discovery may be a Core Resource or discovered as part of a bootstrap.
- The request could be an implementation dependent lookup or be a local RETRIEVE request against the Resource that enables discovery.
- The Device with information to be discovered shall publish the appropriate information to the Resource that enables discovery.
- The publishing Device is responsible for the published information. The publishing Device may UPDATE the information at the resource to enable discovery based on its needs by sending additional publication requests. The policies on the information that is discovered including lifetime is determined by the publishing Device.

11.3.3 Resource based discovery: Information publication process

The mechanism to publish information with the Resource to enable discovery can be done either locally or remotely. The publication process is depicted in Figure 15. The Device which has discovery information to publish shall a) either update the Resource that enables discovery if hosted locally or b) issue an UPDATE request with the information to the Device which hosts the Resource that enables discovery. The Device hosting the Resource to enable discovery adds/updates the Resource to enable discovery with the provided information and then responds to the Device which has requested the publication of the Resource with an UPDATE response.

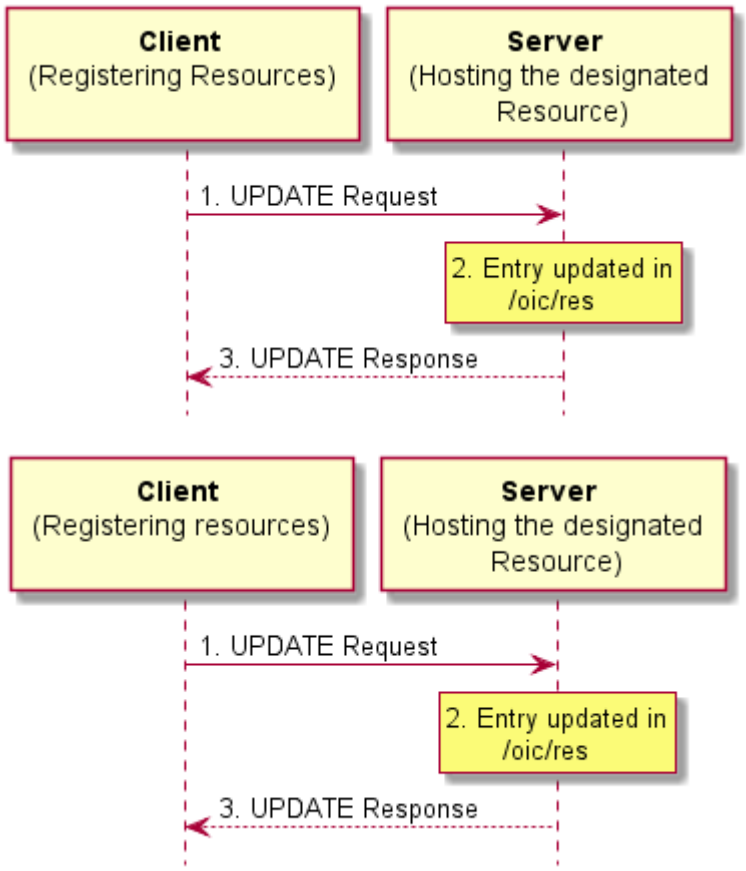


Figure 15 – Resource based discovery: Information publication process

11.3.4 Resource based discovery: Finding information

The discovery process (Figure 16) is initiated as a RETRIEVE request to the Resource to enable discovery. The request may be sent to a single Device (as in a Unicast) or to multiple Devices (as in Multicast). The specific mechanisms used to do Unicast or Multicast are determined by the support in the data connectivity layer. The response to the request has the information to be discovered based on the policies for that information. The policies can determine which information is shared, when and to which requesting agent. The information that can be discovered can be Resources, types, configuration and many other standards or custom aspects depending on the request to appropriate Resource and the form of request. Optionally the requester may narrow the information to be returned in the request using query parameters in the URI query.

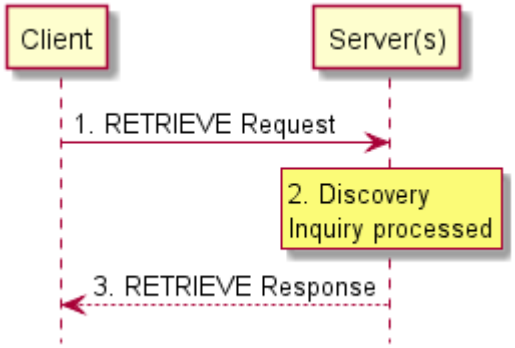


Figure 16 – Resource based discovery: Finding information

Discovery Resources

The following Core Resources shall be implemented on all Devices to support discovery:

– "/oic/res" for discovery of Resources.

– "/oic/p" for discovery of Platform.

– "/oic/d" for discovery of Device information.

Devices shall expose each of "/oic/res", "/oic/d", and "/oic/p" via an unsecured OCF Endpoint. Further details for these mandatory Core Resources are described in Table 24.

Platform Resource

The OCF recognizes that more than one instance of Device may be hosted on a single Platform. Clients need a way to discover and access the information on the Platform. The Core Resource, "/oic/p" exposes Platform specific Properties. All instances of Device on the same Platform shall have the same values of any Properties exposed (i.e. a Device may choose to expose optional Properties within "/oic/p" but when exposed the value of that Property should be the same as the value of that Property on all other Devices on that Platform).

Device Resource

The Device Resource shall have the pre-defined URI "/oic/d". The Resource "/oic/d" exposes the Properties pertaining to a Device as defined in Table 24. The Properties exposed are determined by the specific instance of Device and defined by the Resource Type(s) of "/oic/d" on that Device. Since all the Resource Types of "/oic/d" are not known a priori, the Resource Type(s) of "/oic/d" shall be determined by discovery through the Core Resource "/oic/res". The Device Resource "/oic/d" shall have a default Resource Type that helps in bootstrapping the interactions with this Device (the default type is described in Table 24).

Protocol indication

A Device may need to support different messaging protocols depending on requirements for different vertical domain profiles. For example, a Smart Home profile may use CoAP and an Industrial profile may use DDS. To enable interoperability, a Device uses the protocol indication to indicate the transport protocols they support and can communicate over.

Table 24 – Mandatory discovery Core Resources

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/oic/res"	Default	"oic.wk.res"	"oic.if.ll"	The Resource through which the corresponding Server is discovered and introspected for available Resources. "/oic/res" shall expose the Resources that are discoverable on a Device. When a Server receives a RETRIEVE request targeting "/oic/res" (e.g., "GET /oic/res"), it shall respond with the links list of all the Discoverable Resources of itself. The "/oic/d" and "/oic/p" are Discoverable Resources, hence their links are included in "/oic/res" response. The Properties exposed by "/oic/res" are listed in Table 25.	Discovery
"/oic/p"	Platform	"oic.wk.p"	"oic.if.r"	The Discoverable Resource through which Platform specific information is discovered. The Properties exposed by "/oic/p" are listed in Table 28	Discovery
"/oic/d"	Device	"oic.wk.d" and/or one or more Device Specific Resource Type ID(s)	"oic.if.r"	The discoverable via "/oic/res" Resource which exposes Properties specific to the Device instance. The Propert*ies* exposed by "/oic/d" are listed in Table 27 "/oic/d" may have one or more Resource Type(s) that are specific to the Device in addition to the default Resource Type or if present overriding the default Resource Type. The base type "oic.wk.d" defines the Properties that shall be exposed by all Devices. The Device specific Resource Type(s) exposed are dependent on the class of Device (e.g. air conditioner, smoke alarm, and combined light/fan); applicable values are defined by ISO/IEC 30118-5:2018.	Discovery

Table 25 defines "oic.wk.res" Resource Type.

Table 25 – "oic.wk.res" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	"n"	string	N/A	N/A	R	No	Human-friendly name defined by the vendor

Links	"links"	array	See 7.8.2	N/A	R	Yes	The array of Links describes the URI, supported Resource Types and OCF Interfaces, and access policy.

A Device shall support CoAP based discovery as the baseline discovery mechanism (see 10.4).

The "/oic/res" shall list all Resources that are indicated as discoverable (see 11.3). Also the following architecture Resource Types shall be listed:

– Introspection Resource indicated with an "rt" value of "oic.wk.introspection".
– "/oic/p" indicated with an "rt" value of "oic.wk.p".
– "/oic/d" indicated with an "rt" value of "oic.wk.d"
– "/oic/sec/doxm" indicated with an "rt" value of "oic.r.doxm" as defined in ISO/IEC 30118-2:2018.
– "/oic/sec/pstat" indicated with an "rt" value of "oic.r.pstat" as defined in ISO/IEC 30118-2:2018.
– "/oic/sec/acl2" indicated with an "rt" value of "oic.r.acl2" as defined in ISO/IEC 30118-2:2018.
– "/oic/sec/cred" indicated with an "rt" value of "oic.r.cred" as defined in ISO/IEC 30118-2:2018.

Conditionally required:

– "/oic/res" with an "rt" value of "oic.wk.res" as self-reference, on the condition that "oic/res" has to signal that it is Observable by a Client.

The Introspection Resource is only applicable for Devices that host Vertical Resource Types (e.g. "oic.r.switch.binary") or vendor-defined Resource Types. Devices that only host Resources required to onboard the Device as a Client do not have to implement the Introspection Resource.

Table 26 provides an OCF registry for protocol schemes.

Table 26 – Protocol scheme registry

SI Number	Protocol
1	"coap"
2	"coaps"
3	"http"
4	"https"
5	"coap+tcp"
6	"coaps+tcp"

NOTE The discovery of an OCF Endpoint used by a specific protocol is out of scope. The mechanism used by a Client to form requests in a different messaging protocol other than discovery is out of scope.

The following applies to the use of "/oic/d":

– A Device may choose to expose its Device Type(s) (e.g., refrigerator or A/C or composite of multiple Device Types) by adding the Device Type to the list of Resource Types associated with "/oic/d".

3084 – For example; "rt" of "/oic/d" becomes ["oic.wk.d", "oic.d.<thing1>", "oic.d.<thing2>"]; where
3085 "oic.d.<thing1>" and "oic.d.<thing2>" are defined in another specification such as
3086 ISO/IEC 30118-5:2018.

3087 – This implies that the Properties exposed by "/oic/d" are by default the mandatory Properties
3088 in Table 27.

3089 – A vertical may choose to extend the list of Properties defined by the Resource Type "oic.wk.d".
3090 In that case, the vertical shall assign a new Device Type specific Resource Type ID. The
3091 mandatory Properties defined in Table 27 shall always be present.

3092 – A Device may choose to expose a separate, Discoverable Resource with its Resource Type ID
3093 set to an OCF defined Device Type. In this case the Resource is equivalent to an instance of
3094 "oic.wk.d" and adheres to the definition thereof. As such the Resource shall at a minimum
3095 expose the mandatory Properties of "oic.wk.d". In the case where the Resource tagged in this
3096 manner is defined to be an instance of a Collection in accordance with 7.8.3 then the Resources
3097 that are part of that Collection shall at a minimum include the Resource Types mandated for
3098 the Device Type. For example, if a Collection Resource has an "rt" value of ["oic.d.light"], the
3099 Collection includes an instance of "oic.r.switch.binary" which is mandatory for an "oic.d.light"
3100 as per ISO/IEC 30118-5:2018.

3101 Table 27 "oic.wk.d" Resource Type definition defines the base Resource Type for the "/oic/d"
3102 Resource.

3103 **Table 27 – "oic.wk.d" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
(Device) Name	"n"	"string:	N/A	N/A	R	Yes	Human friendly name defined by the vendor. In the presence of "n" Property of "/oic/con", both have the same Property Value. When "n" Property Value of "/oic/con" is modified, it shall be reflected to "n" Property Value of "/oic/d".
Spec Version	"icv"	"string"	N/A	N/A	R	Yes	Spec version of this document this Device is implemented to, The syntax is "ocf.<major>.<minor>.<sub-version>" where <major>, <minor, and <sub-version> are the major, minor and sub-version numbers of the document respectively. For this version of the document, the string value shall be "ocf.2.0.2".
Device ID	"di"	"uuid"	N/A	N/A	R	Yes	Unique identifier for Device. This value shall be the same value (i.e. mirror) as the doxm.deviceuuid Property as defined in ISO/IEC 30118-2:2018. Handling privacy-sensitivity for the "di" Property, refer to clause 13.16 in ISO/IEC 30118-2:2018.
Data Model Version	"dmv"	"csv"	N/A	N/A	R	Yes	Spec version of the Resource specification to which this Device data model is implemented; if implemented against a Vertical specific Device specification(s), then the Spec version of the vertical specification this Device model is implemented to. The syntax is a comma separated list of <res>.<major>.<minor>.<sub-

							version> or <vertical>.<major>.<minor>.<sub-version>. <res> is the string "ocf.res" and <vertical> is the name of the vertical defined in the Vertical specific Resource specification. The <major>, <minor>, and <sub-version> are the major, minor and sub-version numbers of the specification respectively. One entry in the csv string shall be the applicable version of the Resource Type Specification for the Device (e.g "ocf.res.1.0.0"). If applicable, additional entry(-ies) in the csv shall be the vertical(s) being realized (e.g. "ocf.sh.1.0.0"). This value may be extended by the vendor. The syntax for extending this value, as a comma separated entry, by the vendor shall be by adding x.<Domain_Name>.<vendor_string>. For example "ocf.res.1.0.0, ocf.sh.1.0.0, x.com.example.string", The order of the values in the comma separated string can be in any order (i.e. no prescribed order). This Property shall not exceed 256 octets.
Protocol Independent ID	"piid"	"uuid"	N/A	N/A	R	Yes	A unique and immutable Device identifier. A Client can detect that a single Device supports multiple communication protocols if it discovers that the Device uses a single Protocol Independent ID value for all the protocols it supports. Handling privacy-sensitivity for the "piid" Property, refer to clause 13.16 in ISO/IEC 30118-2:2018.
Localized Descriptions	"ld"	"array"	N/A	N/A	R	No	Detailed description of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the Device description in the indicated language.
Software Version	"sv"	"string"	N/A	N/A	R	No	Version of the Device software.
Manufacturer Name	"dmn"	"array"	N/A	N/A	R	No	Name of manufacturer of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the manufacturer name in the indicated language.
Model Number	"dmno"	"string"	N/A	N/A	R	No	Model number as designated by manufacturer.

3104 The additional Resource Type(s) of the "/oic/d" Resource are defined by ISO/IEC 30118-5:2018.

3105 Table 28 defines "oic.wk.p" Resource Type.

3106 **Table 28 – "oic.wk.p" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Platform ID	"pi"	"string"	N/A	N/A	R	Yes	Unique identifier for the physical Platform (UIUID); this shall be a UUID in accordance with IETF RFC 4122. It is recommended that the UUID be created using the random generation scheme (version 4 UUID) specific in the RFC. Handling privacy-sensitivity for the "pi" Property, refer to clause 13.16 in ISO/IEC 30118-2:2018.
Manufacturer Name	"mnmn"	"string"	N/A	N/A	R	Yes	Name of manufacturer.
Manufacturer Details Link	"mnml"	"uri"	N/A	N/A	R	No	Reference to manufacturer, represented as a URI.
Model Number	"mnmo"	"string"	N/A	N/A	R	No	Model number as designated by manufacturer.
Date of Manufacture	"mndt"	"date"	N/A	Time	R	No	Manufacturing date of Platform.
Serial number	"mnsel	"string"	N/A	s	R	No	Serial number of the Platform, may be unique for each Platform of the same model number.
Platform Version	"mnpv"	"string"	N/A	N/A	R	No	Version of Platform – string (defined by manufacturer).
OS Version	"mnos"	"string"	N/A	N/A	R	No	Version of Platform resident OS – string (defined by manufacturer).
Hardware Version	"mnhw"	"string"	N/A	N/A	R	No	Version of Platform hardware.
Firmware version	"mnfv"	"string"	N/A	N/A	R	No	Version of Platform firmware.
Support link	"mnsl"	"uri"	N/A	N/A	R	No	URI that points to support information from manufacturer.
SystemTime	"st"	"date-time"	N/A	N/A	R	No	Reference time for the Platform.
Vendor ID	"vid"	"string"	N/A	N/A	R	No	Vendor defined string for the Platform. The string is freeform and up to the vendor on what text to populate it.

3107 ### 11.3.5 Resource discovery using "/oic/res"

3108 Discovery using "/oic/res" is the default discovery mechanism that shall be supported by all Devices
3109 as follows:

3110 – Every Device updates its local "/oic/res" with the Resources that are discoverable (see 7.3.2.2).
3111 Every time a new Resource is instantiated on the Device and if that Resource is discoverable

by a remote Device then that Resource is published with the "/oic/res" Resource that is local to the Device (as the instantiated Resource).

– A Device wanting to discover Resources or Resource Types on one or more remote Devices makes a RETRIEVE request to the "/oic/res" on the remote Devices. This request may be sent multicast (default) or unicast if only a specific host is to be probed. The RETRIEVE request may optionally be restricted using appropriate clauses in the query portion of the request. Queries may select based on Resource Types, OCF Interfaces, or Properties.

– The query applies to the representation of the Resources. "/oic/res" is the only Resource whose representation has "rt". So "/oic/res" is the only Resource that can be used for Multicast discovery at the transport protocol layer.

– The Device receiving the RETRIEVE request responds with a list of Resources, the Resource Type of each of the Resources and the OCF Interfaces that each Resource supports. Additionally, information on the policies active on the Resource can also be sent. The policy supported includes Observability and discoverability.

– The receiving Device may do a deeper discovery based on the Resources returned in the request to "/oic/res".

The information that is returned on discovery against "/oic/res" is at the minimum:

– The URI (relative or fully qualified URL) of the Resource.

– The Resource Type(s) of each Resource. More than one Resource Type may be returned if the Resource enables more than one type. To access Resources of multiple types, the specific Resource Type that is targeted shall be specified in the request.

– The OCF Interfaces supported by that Resource. Multiple OCF Interfaces may be returned. To access a specific OCF Interface that OCF Interface shall be specified in the request. If the OCF Interface is not specified, then the Default OCF Interface is assumed.

Different "/oic/res" responses are returned according to requesting Clients, which indicate their preference via inclusion or otherwise of an OCF-Accept-Content–Format-Version option.

For Clients that do not include the OCF-Accept-Content-Format-Version option, an "/oic/res" response shall use "sec" and "port" to provide the information for an encrypted connection. See Annex E for the schema for the Link.

For Clients that do include the OCF-Accept-Content-Format-Version option, an "/oic/res" response includes an array of Links to conform to IETF RFC 6690. Each Link shall use an "eps" Parameter to provide the information for an encrypted connection and carry "anchor" of the value OCF URI where the authority component of <deviceID> indicates the Device hosting the target Resource.

The OpenAPI 2.0 file for discovery using "/oic/res" is described in Annex D the schema that is applicable to requesting Clients that do not include an OCF-Accept-Content-Format-Version option is described in Annex E. Also refer to clause 10 (OCF Endpoint discovery) for details of Multicast discovery using "/oic/res" on a CoAP transport.

For example, a Light Device might return the following to OIC 1.1 Clients:

```
[
  {
    "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
    "links": [
      {
        "href": "coaps://[fe80::b1d6]:44444/oic/res",
        "rel": "self",
        "rt": ["oic.wk.res"],
        "if": ["oic.if.ll", "oic.if.baseline"],
        "p": {"bm": 3}
```



```
        },
        {
          "href": "/oic/p",
          "rt": ["oic.wk.p"],
          "if": ["oic.if.r", "oic.if.baseline"],
          "p": {"bm": 3, "sec": true, "port": 11111}
        },
        {
          "href": "/oic/d",
          "rt": ["oic.wk.d", "oic.d.light"],
          "if": ["oic.if.r", "oic.if.baseline"],
          "p": {"bm": 3, "sec": true, "port": 11111}
        },
        {
          "href": "/myLight",
          "rt": ["oic.r.switch.binary"],
          "if": ["oic.if.a", "oic.if.baseline"],
          "p": {"bm": 3, "sec": true, "port": 11111}
        }
      ]
    }
]
```

The light Device might return the following to Clients that request with the Content Format of "application/vnd.ocf+cbor" in Accept Option:

```
[
  {
    "href": "/oic/res",
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989/oic/res",
    "rel": "self",
    "rt": ["oic.wk.res"],
    "if": ["oic.if.ll", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [{"ep": "coap://[fe80::b1d6]:44444"}]
  },
  {
    "href": "/oic/p",
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989,
    "rt": ["oic.wk.p"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [{"ep": "coap://[fe80::b1d6]:44444"},
            {"ep": "coaps://[fe80::b1d6]:11111"}
           ]
  },
  {
    "href": "/oic/d",
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989,
    "rt": ["oic.wk.d", "oic.d.light"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [{"ep": "coap://[fe80::b1d6]:44444"},
            {"ep": "coaps://[fe80::b1d6]:11111"}
           ]
  },
  {
    "href": "/myLight",
    "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989,
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "p": {"bm": 3},
```



```
    "eps": [{"ep": "coap://[fe80::b1d6]:44444"},
            {"ep": "coaps://[fe80::b1d6]:11111"}
           ]
  }
]
```

After performing discovery using "/oic/res", Clients may discover additional details about Server by performing discovery using "/oic/p", "/oic/rts" etc. If a Client already knows about Server it may discover using other Resources without going through the discovery of "/oic/res".

11.3.6 Resource Directory (RD) based discovery

11.3.6.1 Introduction

11.3.6.1.1 Indirect discovery for lookup of the Resources

Direct discovery is the mechanism used currently to find Resources in the network. When needed, Resources are queried at a particular Device directly or a multicast packet is sent to all Devices. Each queried Device responds directly with its Resources to the discovering Device. Resources available locally are registered on the same Device.

In some situations, one of the other mechanisms described in 11.3.2.3, called indirect discovery, may be required. Indirect discovery is when a 3rd party Device, other than the discovering Device and the discovered Device, assists with the discovery process. The 3rd party Device, called Resource Directory (RD), only provides information on Resources on behalf of another Device but does not host Resources on part of that Device.

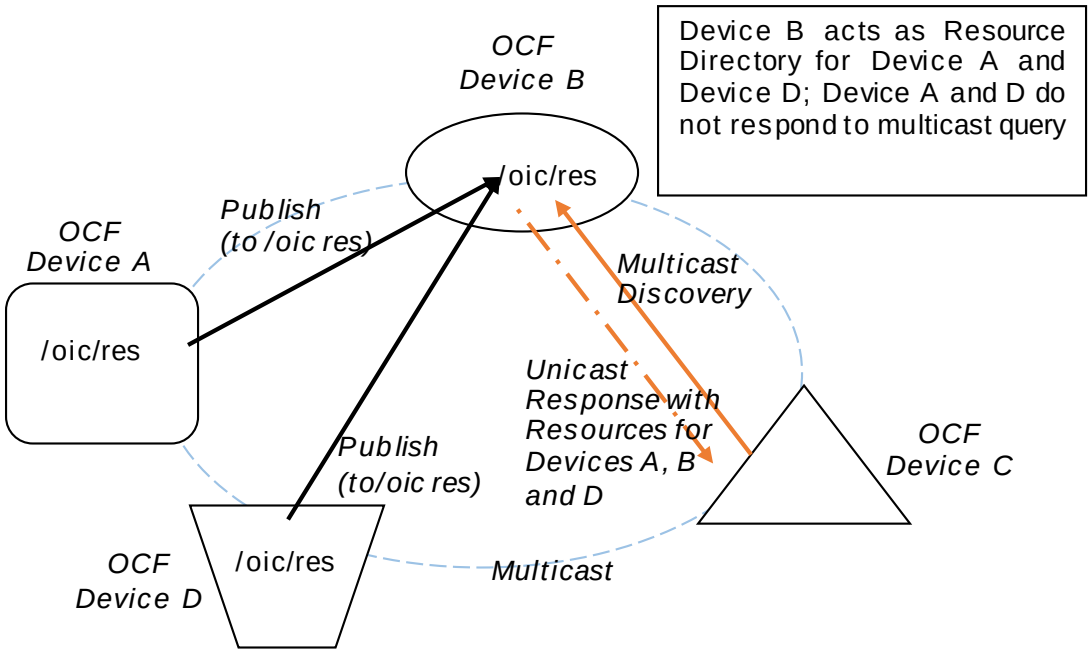


Figure 17 – Indirect discovery of Resources by via an RD

In Figure 17, Device B acts as Resource Directory for Device A and Device D. Device A and Device D publish their Resource information to Device B. Device C may query Deice B to acquire the Resource information of Device A and Device D. Device A and Device D may not respond to a multicast query when Device B, as a Resource Directory, responds to the query on their behalf.

Indirect discovery is useful for a constrained Device that needs to sleep to manage power and cannot process every discovery request, or when Devices may not be on the same network and requires optimization for discovery. Once Resources are discovered using indirect discovery, i.e.,

RD query, then the access to the Resource is done by a request sent directly to the Device that hosts that Resource.

11.3.6.1.2 Resource Directory

A Resource Directory (RD) is a Device that assists with indirect discovery. A Device which acts as an RD will be involved in the following operations.

- *RD discovery* – the procedure with which publishing Devices discover an RD and acquire the criteria to select from among multiple detected RDs.
- *Resource publish* – the procedures with which Devices publish their Resource information, i.e. Links. Future revision of this documentwill allow modifying RD entries with UPDATE and DELETE operations. Any UPDATE or DELETE operations performed on an RD in this documentshould be either silently ignored or generate an error.
- *Resource exposure* – the feature with which RDs expose the Links hosted by the 3rd party Devices via their own "/oic/res".

The RDs make use of Resource Type "oic.wk.rd" defined in Table 29 and Table 30. A Device that supports the capability to host indirect discovery shall expose an instance of "oic.wk.rd" in its "/oic/res" to announce that it serves as an RD. The discoverable instance of "oic.wk.rd" shall allow only secure connections (e.g. OCF Endpoint with a scheme of "coaps" or "coaps+tcp"). A publishing Device may send a RETRIEVE request to "/oic/rd" to acquire the selection criteria among multiple RDs. Then it may send an UPDATE request to "/oic/rd" with its Links in the payload to publish the Links in "/oic/res" of the RD. A publishing Device is responsible to insure an RD has the correct published Links to expose via its "/oic/res".

Table 29 – "oic.wk.rd" Resource Type definition

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/oic/rd"	Resource Directory	"oic.wk.rd"	"oic.if.baseline"	The Discoverable Resource Type through with which an RD 1) facilitates its discovery and provides the criteria to select an RD and 2) allows Devices to publish their Links in "/oic/res" of the RD.	Discovery

Table 30 – "oic.wk.rd" Properties

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Selector	"sel"	"integer"	N/A	N/A	R	Yes	Provides the criteria for RD selection. An integer representing a value calculated by the RD. The value is in the range of 0 to 100. The lower the value, the more preferable the RD is.

An RD may be queried at its "/oic/res" Resource to find Resources hosted on other Devices. These Devices can be sleepy nodes or any other Device that cannot or may not respond to discovery requests. A publishing Device may publish all or a partial list of Resources they host to an RD. The RD then responds to queries for Resource discovery on behalf of the publishing Device (for example: when a Device may go to sleep). For general Resource discovery, the RD behaves like any other Server in responding to requests to "/oic/res".

The remainder of 11.3.6 is divided into three parts. The first part covers "RD Discovery" (see 11.3.6.2), i.e., discovering and selecting of an RD. The second part covers "Resource publish" (see 11.3.6.3), i.e., publishing of Resources. The third part covers "Resource exposure" (see 11.3.6.4) where the RD replies to queries from Devices looking to discover Resources.

11.3.6.2 RD discovery

11.3.6.2.1 Discovering an RD

An RD shall support RD discovery.

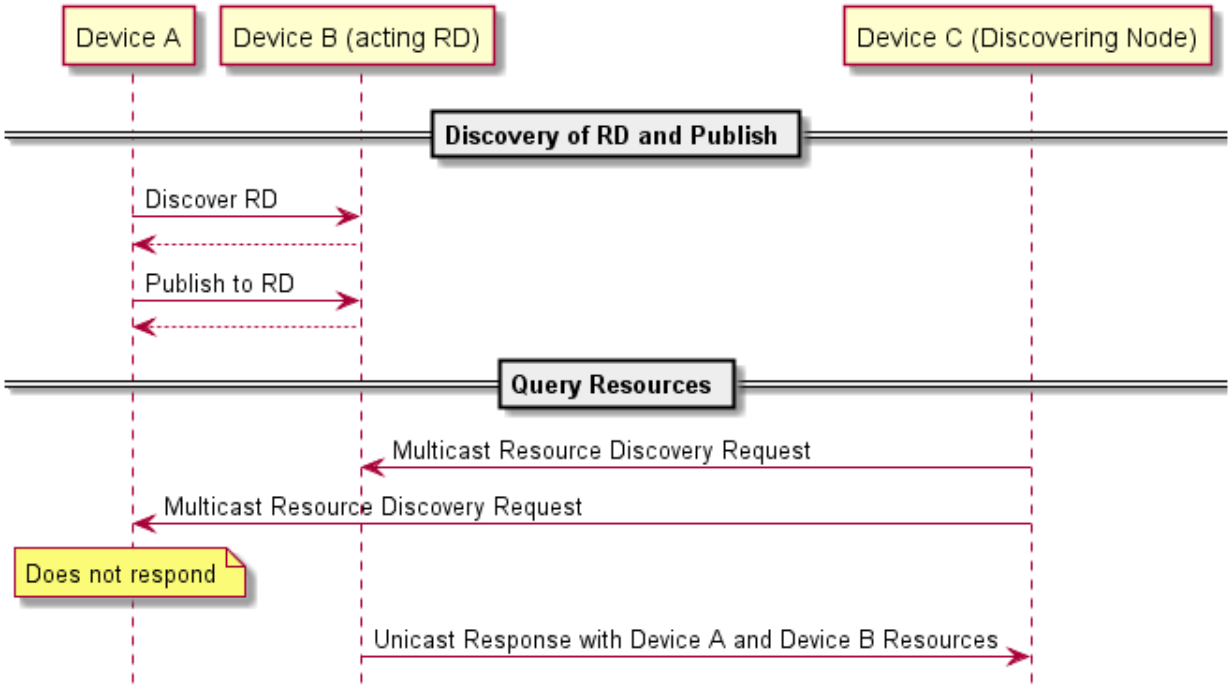


Figure 18 – RD discovery and RD supported query of Resources support

In Figure 18 and Figure 19, a Device that wishes to publish its Resources first discovers an RD and then publishes the desired Resource information. Once a set of Resources have been published to an RD then the publishing Device should not respond to multicast Resource discovery queries for those published Resources when the RD is on the same multicast domain. In that case, only the RD should respond to multicast Resource discovery requests on the Resource published to it.

It is allowed for more than one Device to act as an RD. The reason to have multiple RD support is to make networks scalable, handle network failures and prevent centralized Device failure bottlenecks. This does not preclude a scenario where a use case or deployment environment may require a single Device in the environment to be deployed as the only RD (e.g. gateway model).

Discovering an RD may result in responses from more than one RD. If more than one RD responds, the discovering Device may select one of them based on the weighting parameter(s) provided in the response from the RD.

A Client that performs Resource discovery uses an RD just like it uses any other Server for discovery. It may send a unicast request to the RD when it needs only the Resources published on the RD or do a multicast query when it does not require or have explicit knowledge of an RD.

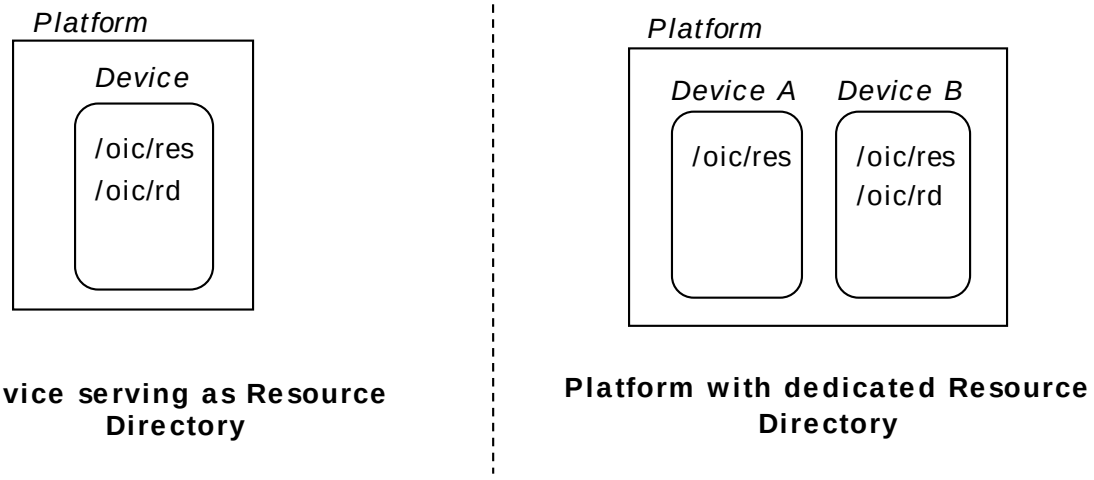


Figure 19 – Resource Direction Deployment Scenarios

RDs may also be discovered in the following ways:

– Pre-configuration: Devices wishing to publish Resource information may be configured a priori with the information (e.g. IP address, port, transport etc.) of a specific RD. This pre-configuration may be done at onboarding or may be updated on the Device using an out-of-band method. This pre-configuration may be done by the manufacturer.

– Query-oriented: A publishing Device wanting to discover Resource Directories using query-oriented discovery may issue a multicast Resource discovery request for "/oic/res?rt=oic.wk.rd". Only and all Devices that can be an RD shall respond to this query. The "/oic/rd" response shall include information about the RD i.e., the presence of "oic.wk.rd" Link (as defined by the Resource Type) and a subsequent query to "/oic/rd" would produce weighting parameters to allow the discovering Device to select between RDs (see details in RD selection 11.3.6.2.2). The "oic.wk.rd" Resource shall be instantiated on the Devices acting as RDs. The "oic.wk.rd" schema is as defined in Annex D.

11.3.6.2.2 RD selection process

The Device that wants to use an RD will find zero or more RDs on the network. There may not be an RD within the network. When discovering RDs, the Device needs to select an RD of all RDs found on the network. The Device may send a RETRIEVE request to "/oic/rd" of a specific RD, the RD shall respond with the representation of "/oic/rd/" containing selection criteria as defined by the "sel" Property. The lower the "sel" Property value is, the more preferable the responding RD is. The creation of the "sel" value is vendor defined.

For example an "/oic/rd" response may return the following.

```
{
  "rt": ["oic.wk.rd"],
```



```
  "if": ["oic.if.baseline"],
  "sel": 50
}
```

The selection based on the "sel" Property value will ensure that a Device can judge if the found RD is suitable for its needs.

The following situations may occur during the selection of an RD:

– A single or multiple RDs are present in the network.

– No RD is present in the network.

– An additional RD arrives on the network.

In the first scenario, the RDs are already present. If a single RD is detected then that RD may be used. When multiple RDs are detected the Device may use the "sel" Property value to select the RD.

In the second scenario, the publishing Device may continue looking for an RD until one is found or give up using an RD altogether.

In the third scenario, the Device has already published its Resources to an existing RD, then discovers a new RD on the network. After judging the "sel" Property value, the Device may choose to move to the new RD. The Device should delete its Resource information from the currently used RD and publish the information to the new RD.

11.3.6.3 Resource publish

11.3.6.3.1 Overview

An RD shall provide the facility to allow Devices to publish their Resource information to an RD.

11.3.6.3.2 Publish Resources

11.3.6.3.2.1 Overview

After the selection process of an RD, a Device may push its Resource information to the selected RD, i.e., publish the Links in its "/oic/res" to the "/oic/res" of the RD.

The publishing Device may decide to publish all Resources or just a few of the Resources on the RD. The publishing Device should only publish Resources that are otherwise published to its own "/oic/res"; a publishing Device should not publish non-Discoverable Resources or Resources hosted by some other Device. A publishing Device shall respond to discovery requests on its "/oic/res" Resource unless all its Discoverable Resources have been published in an RD.

11.3.6.3.2.2 Publish: Push Resource information

Resource information may be published using an UPDATE request sent to "/oic/rd".

A Device which hosts a Resource may publish the Resource information, i.e. the Link targeting the Resource, to an RD by sending an UPDATE request with the Link in the payload. The published Link shall be exposed through the "/oic/res" of the RD.

When a Device first publishes a Link or Links, it shall send an UPDATE request to the "/oic/rd" Resource of the RD including the following key-value pairs in the payload:

– "di" –its value shall be the Device ID of the publishing Device, i.e. the "di" value of "/oic/d".

– "links" –its value shall be the array of Links to be published. Links may omit the "ins" Parameter in which case the RD will assign a value for each Link. The supplied "ins" Parameter by the Client is allowed to be overruled by the RD, e.g. an RD can ignore the supplied "ins" value.

– "ttl" –its value indicates how long (in seconds) the publishing Device requests the RD to keep this published Link.

Notice that the payload shall carry the appropriate Content-Format of "application/vnd.ocf+cbor".

```
{
  "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
  "links": [
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9"
      "href": "/myLightSwitch",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
        {"ep": "coaps://[fe80::b1d6]:1122"},
        {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
      ]
    },
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/myLightBrightness",
      "rt": ["oic.r.brightness"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[[2001:db8:a::123]:2222"}
      ]
    }
  ],
  "ttl": 600
}
```

When an RD receives this initial UPDATE request, it determines whether to grant the request or not. Upon granting the request, the RD shall send back an UPDATE response to the publishing Device. The response shall include a payload with the same information as the original UPDATE request with the following possible differences:

– For each Link, an "ins" Parameter shall be included in the response. The RD shall assign a unique "ins" value identifying the Link among all the Links it advertises. If the publishing Device included an "ins" value in the UPDATE request, the RD may use it as long as it doesn't match any existing "ins" value in the published Links.

– The "ttl" Property Value shall be assigned by the RD and it shall be included in the response. The RD should use the value included in the UPDATE request but may assign a value that is lower if it is not able to honour the requested "ttl" value. After this time elapses, the RD shall remove the Links. To keep a Link alive the publishing Device may update the "ttl" using the UPDATE schema.

The RD shall add the new Links to its "/oic/res" and expose them to a valid discovery query, i.e. RETRIEVE request:

```
{
  "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
  "links": [
    {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/myLightSwitch",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
```



```
          {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
          {"ep": "coaps://[fe80::b1d6]:1122"},
          {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
        ],
        "ins": 11235
      },
      {
        "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
        "href": "/myLightBrightness",
        "rt": ["oic.r.brightness"],
        "if": ["oic.if.a", "oic.if.baseline"],
        "p": {"bm": 3},
        "eps": [
          {"ep": "coaps://[[2001:db8:a::123]:2222"}
        ],
        "ins": 112358
       }
      ].
    "ttl": 600
}
```

Once a publishing Device has published Resources to an RD, it may choose not respond to the multicast discovery queries for the same Resources against its own "/oic/res", especially when on the same multicast domain as the RD. After publishing Resources, primarily it is the RDs responsibility to reply to the queries for the published Resources.

There is another possibility that the RD and the publishing Device both respond to the multicast query from the discovering Device. This will create a duplication of the information but is an alternative that may be used for non-robust networks. It is not a recommended option but for industrial scenarios, this is one of the possibilities. Either way, discovering Clients need to always be prepared to process duplicate information in responses to multicast discovery request. The "/oic/rd" schema is as defined in Annex D to specify publishing to the "/oic/rd" Resource.

11.3.6.4 Resource exposure

11.3.6.4.1 "/oic/res" and retrieving of the Resources

The "/oic/res" based discovery process remains the same as that in the absence of an RD. Resources may be discovered by retrieving the "/oic/res" Resource by sending a multicast or unicast request. In the case of a multicast discovery request, an RD shall include in its response any published Resources on behalf of the Device that hosts the Resources. Clients should be prepared to process duplicate Resource information from more than one RD responding with the same information or from an RD and the hosting Device (publishing the Resource information) both responding to the request. Interaction with Resources discovered using the RD is done using the same mechanism and methods as with Resources discovered by retrieving the "/oic/res" Resource of the Device hosting the Resources (e.g., connect to the hosting Device and perform CRUDN operations on the Resource).

Resource Directories provide different "/oic/res" responses according to the requesting Clients, which indicate their preference with content format. OCF 1.0 Clients request with a Content Format of "application/vnd.ocf+cbor" in the Accept Option, whereas the Content-Format "application/cbor" in the Accept Option indicates OIC 1.1 Clients. For OIC 1.1 Clients, the "/oic/res" response includes Links conforming to OIC 1.1 specification, which OIC 1.1 Clients can understand. In this case the Resources hosted by the same Device shall be grouped together within a single JSON Object with "di" indicating the hosting Device. For a 3rd party Resource, i.e., a Resource which doesn't belong to the responding RD, its "href" value shall be a fully qualified transfer protocol URI with an IP address and port number as its authority component (e.g., coaps://[2001:db8:b::c2e5]:22222/myLightSwitch).

For example, an RD might return the following to an OIC 1.1 Clients:


```
[
  {
    "di": "88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "links": [
      {
        "href": "/oic/res",
        "rel": "self",
        "rt": ["oic.wk.res"],
        "if": ["oic.if.ll", "oic.if.baseline"],
        "p": {"bm": 3, "sec": false}
      },
      {
        "href": "/oic/d",
        "rt": ["oic.wk.d", "oic.d.fan"],
        "if": ["oic.if.r", "oic.if.baseline"],
        "p": {"bm": 3, "sec": false}
      },
      {
        "href": "/oic/p",
        "rt": ["oic.wk.p"],
        "if": ["oic.if.r", "oic.if.baseline"],
        "p": {"bm": 3, "sec": true, "port": 33333}
      },
      {
        "href": "/myFanIntrospection",
        "rt": ["oic.wk.introspection"],
        "if": ["oic.if.r", "oic.if.baseline"],
        "p": {"bm": 3, "sec": true, "port": 33333}
      },
      {
        "href": "/oic/rd",
        "rt": ["oic.wk.rd"],
        "if": ["oic.if.baseline"],
        "p": {"bm": 3, "sec": true, "port": 33333}
      },
      {
        "href": "/myFanSwitch",
        "rt": ["oic.r.switch.binary"],
        "if": ["oic.if.a", "oic.if.baseline"],
        "p": {"bm": 3, "sec": true, "port": 33333}
      },
      {
        "href": "/oic/sec/doxm",
        "rt": ["oic.r.doxm"],
        "if": ["oic.if.baseline"],
        "p": {"bm": 1, "sec": false}
      },
      {
        "href": "/oic/sec/pstat",
        "rt": ["oic.r.pstat"],
        "if": ["oic.if.baseline"],
        "p": {"bm": 1, "sec": true, "port": 33333}
      },
      {
        "href": "/oic/sec/cred",
        "rt": ["oic.r.cred"],
        "if": ["oic.if.baseline"],
        "p": {"bm": 1, "sec": true, "port": 33333}
      },
      {
        "href": "/oic/sec/acl2",
        "rt": ["oic.r.acl2"],
        "if": ["oic.if.baseline"],
```



```
          "p": {"bm": 1, "sec": true, "port": 33333}
        }
      ]
    },
    {
      "di": "dc70373c-1e8d-4fb3-962e-017eaa863989",
      "links": [
         {
          "href": "coap://[2001:db8:b::c2e5]:66666/oic/d",
          "rt": ["oic.wk.d", "oic.d.light", "oic.d.virtual"],
          "if": ["oic.if.r", "oic.if.baseline"],
          "p": {"bm": 3, "sec": false}
        },
        {
          "href": "coaps://[2001:db8:b::c2e5]:22222/oic/p",
          "rt": ["oic.wk.p"],
          "if": ["oic.if.r", "oic.if.baseline"],
          "p": {"bm": 3, "sec": true, "port": 22222}
        },
        {
          "href": "coaps://[2001:db8:b::c2e5]:22222/myLightSwitch",
          "rt": ["oic.r.switch.binary"],
          "if": ["oic.if.a", "oic.if.baseline"],
          "p": {"bm": 3, "sec": true, "port": 22222}
        },
        {
          "href": "coaps://[2001:db8:b::c2e5]:22222/myLightBrightness",
          "rt": ["oic.r.brightness"],
          "if": ["oic.if.a", "oic.if.baseline"],
          "p": {"bm": 3, "sec": true, "port": 22222}
        }
      ]
    }
  ]
```

For OCF 1.0 Clients, the "/oic/res" response includes the OCF 1.0 Links with the "anchor" Parameter containing an OCF URI. The "/oic/res" response has a single array of Links to conform to IETF RFC 6690. Each Link shall contain the "anchor" Parameter of the value OCF URI where the authority component of <deviceID> indicates the Device hosting the target Resource.

For example, an RD may return the following to an OCF 1.0 Client.

```
[
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/res",
    "rel": "self",
    "rt": ["oic.wk.res"],
    "if": ["oic.if.ll", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
      {"ep": "coap://[2001:db8:a::b1d4]:77777"},
      {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
    ]
  },
  {
    "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
    "href": "/oic/d",
    "rt": ["oic.wk.d", "oic.d.fan"],
    "if": ["oic.if.r", "oic.if.baseline"],
    "p": {"bm": 3},
    "eps": [
```



```
        {"ep": "coap://[2001:db8:a::b1d4]:77777"},
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/p",
      "rt": ["oic.wk.p"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/myFanIntrospection",
      "rt": ["oic.wk.introspection"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
   {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/rd",
      "rt": ["oic.wk.rd"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/myFanSwitch",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/sec/doxm",
      "rt": ["oic.r.doxm"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coap://[2001:db8:a::b1d4]:77777"},
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/sec/pstat",
      "rt": ["oic.r.pstat"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
```



```
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/sec/cred",
      "rt": ["oic.r.cred"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://88b7c7f0-4b51-4e0a-9faa-cfb439fd7f49",
      "href": "/oic/sec/acl2",
      "rt": ["oic.r.acl2"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
        {"ep": "coaps://[2001:db8:a::b1d4]:33333"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/d",
      "rt": ["oic.wk.d", "oic.d.light"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coap://[2001:db8:b::c2e5]:66666"},
        {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/oic/p",
      "rt": ["oic.wk.p"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/myLightSwitch",
      "rt": ["oic.r.switch.binary"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
      ]
    },
    {
      "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
      "href": "/myLightBrightness",
      "rt": ["oic.r.brightness"],
      "if": ["oic.if.a", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
        {"ep": "coaps://[2001:db8:b::c2e5]:22222"}
      ]
    }
```


]

11.4 Notification

11.4.1 Overview

A Server shall support NOTIFY operation to enable a Client to request and be notified of desired states of one or more Resources in an asynchronous manner. 11.4.2 specifies the Observe mechanism in which updates are delivered to the requester.

11.4.2 Observe

11.4.2.1 Overview

In the Observe mechanism the Client utilizes the RETRIEVE operation to require the Server for updates in case of Resource state changes. The Observe mechanism consists of five steps which are depicted in Figure 20.

NOTE the Observe mechanism can only be used for a resource with a Property of Observable (see 7.3.2.2).

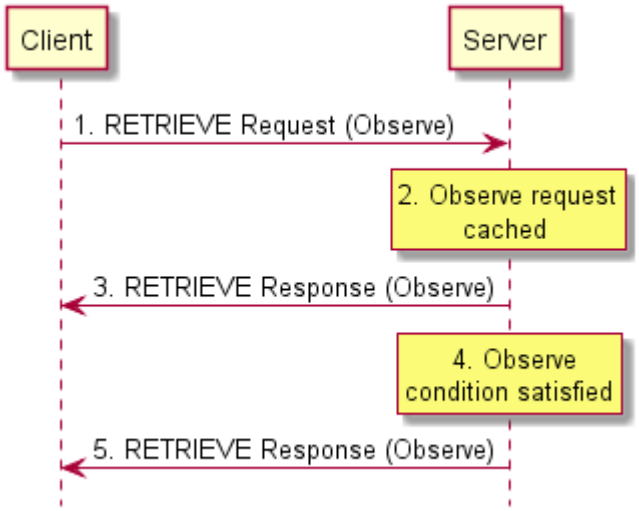


Figure 20 – Observe Mechanism

11.4.2.2 RETRIEVE request with Observe indication

The Client transmits a RETRIEVE request message to the Server to request updates for the Resource on the Server if there is a state change. The RETRIEVE request message carries the following parameters:

– *fr*: Unique identifier of the Client.
– *to*: Resource that the Client is requesting to Observe.
– *ri*: Identifier of the RETRIEVE operation.
– *op*: RETRIEVE.
– *obs*: Indication for Observe operation.

11.4.2.3 Processing by the Server

Following the receipt of the RETRIEVE request, the Server may validate if the Client has the appropriate rights for the requested operation and the Properties are readable and Observable. If

the validation is successful, the Server caches the information related to the Observe request. The Server caches the value of the *ri* parameter from the RETRIEVE request for use in the initial response and future responses in case of a change of state.

11.4.2.4 RETRIEVE response with Observe indication

The Server shall transmit a RETRIEVE response message in response to a RETRIEVE request message from a Client. The RETRIEVE response message shall include the following parameters. If validation succeeded, the response includes an Observe indication. If not, the Observe indication is omitted from the response which signals to the requesting Client that registration for notification was not allowed.

The RETRIEVE response message shall include the following parameters:

– *fr*: Unique identifier of the Server.

– *to*: Unique identifier of the Client.

– *ri*: Identifier included in the RETRIEVE operation.

– *cn*: Information Resource representation as requested by the Client.

– *rs*: The result of the RETRIEVE operation.

– *obs*: Indication that the response is made to an Observe operation.

11.4.2.5 Resource monitoring by the Server

The Server shall monitor the state the Resource identified in the Observe request from the Client. Anytime there is a change in the state of the Observed Resource, the Server sends another RETRIEVE response with the Observe indication. The mechanism does not allow the client to specify any bounds or limits which trigger a notification, the decision is left entirely to the server.

11.4.2.6 Additional RETRIEVE responses with Observe indication

The Server shall transmit updated RETRIEVE response messages following Observed changes in the state of the Resources indicated by the Client. The RETRIEVE response message shall include the parameters listed in 11.4.2.4.

11.4.2.7 Cancelling Observe

The Client can explicitly cancel Observe by sending a RETRIEVE request without the Observe indication field to the same Resource on the Server which it was Observing. For certain protocol mappings, the Client may also be able to cancel an Observe by ceasing to respond to the RETRIEVE responses.

11.5 Device management

11.5.1 Overview

The Device management includes the following functions:

– Diagnostics and maintenance.

The Device management functionalities specified in this version of document are intended to address the basic Device management features. Addition of new Device management features in the future versions of the documentis expected.

11.5.2 Diagnostics and maintenance

The Diagnostics and Maintenance function is intended for use by administrators to resolve issues encountered with the Devices while operating in the field. If diagnostics and maintenance is supported by a Device, the Core Resource "/oic/mnt" shall be supported as described in Table 31.

3799 **Table 31 – Optional diagnostics and maintenance Device management Core Resources**

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/oic/mnt"	Maintenance	"oic.wk.mnt"	"oic.if.rw"	The Resource through which the Device is maintained and can be used for diagnostic purposes. The Properties exposed by "/oic/mnt" are listed in Table 32.	Device management

3800 Table 32 defines the "oic.wk.mnt" Resource Type. At least one of the Factory Reset, Reboot or last
3801 error Properties shall be implemented.

3802 **Table 32 – "oic.wk.mnt" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Factory_Reset	"fr"	"boolean"	N/A	N/A	R, W	No	When writing to this Property: false – No action (Default*) true – Start Factory Reset After factory reset all configuration and state data will be lost. When reading this Property, a value of true indicates a pending factory reset. Once the factory reset has been completed, the Device shall set the value back to false.
Reboot	"rb"	"boolean"	N/A	N/A	R, W	No	When writing to this Property: false – No action (Default) true – Start Reboot After Reboot, this value shall be changed back to the default value (i.e., false)
Last error	"err"	"integer"	HTTP error code	N/A	R	No	Last occurred error code, shall be cleared to 503 (service unavailable), when doing a Factory Reset or Reboot. All HTTP errors outside the 100, 200 or 300 range shall be stored.

3803

3804 NOTE Default indicates the value of this Property as soon as the Device is rebooted or factory reset.

3805 ### 11.5.3 Network monitoring

3806 Network monitoring is used for monitoring the current network state of the Device.

3807 The network monitoring Resource Type is "oic.wk.nmon" and is described in Table 33. The
3808 Resource Type may occur multiple times if more than 1 network interface is implemented. The
3809 Common Property "n" may be used to distinguish the different network interfaces, like distinguish
3810 the 2.4 and 5G Wi-Fi network interfaces.

3811 **Table 33 – Optional monitoring Device management Core Resources**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/example/oic/nmon"	Network Monitoring	"oic.wk.nmon"	"oic.if.rw oic.if.baseline"	The Resource through which the Device is monitored. The Resource exposes Properties relevant to aspects that may be monitored. The Resource Properties exposed by Resource Type "oic.wk.nmon" are listed in Table 34	Device management

3812

3813 Table 34 defines "oic.wk.nmon" Resource Type.

3814 **Table 34 – "oic.wk.nmon" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Network indicator	"ianaifType"	"integer"	The integer value of the ianaifType	N/A	R	Yes	The network type this Resource is collecting information from as defined by: https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib
reset	"reset"	"boolean"	True, all collected values should be reset. The server should reset the value automatically to false after the reset occurred.	N/A	RW	Yes	Reset of the collected values
Collecting status indication	"col"	"boolean"	True: collecting data. False: not collecting data	N/A	RW	Yes	Boolean to start/stop collecting data.
Transmission bytes	"tx"	"integer"	N/A	kilobytes	R	No	Amount of transmitted kilo bytes from the collection
Reception bytes	"rx"	"integer"	N/A	kilobytes	R	No	Amount of received kilo bytes from the collection.
Maximum message size tx	"mmstx"	"integer"	bytes	bytes	R	No	Maximum transmitted message, e.g. Max(tx) in the collection period

Maximum message size rx	"mmsrx"	"integer"	bytes	bytes	R	No	Maximum received message, e.g. Max(rx) in the collection period
Average message size -tx	"amstx"	"integer"	bytes	bytes	R	No	Average transmitted message size, e.g AVG(tx) in the collection period.
Average message size -rx	"amsrx"	"integer"	bytes	bytes	R	No	Average received message size e.g AVT(rx) in the collection period.

Examples of typical used values for ianaifType are 71 (ieee80211) for Wi-Fi and 6 (ethernetCsmacd) for Ethernet.

A Device should start collecting network monitoring data when receiving an UPDATE operation with the parameter "col" = true. A Device should stop collecting network data when receiving an UPDATE operation with parameter "col" = false. The collected network data should be reset when an UPDATE operation with parameter "reset" = true is received, if the parameter "reset" is false then the values should not be reset. Figure 21 illustrates the interactions with the network monitoring Resource.

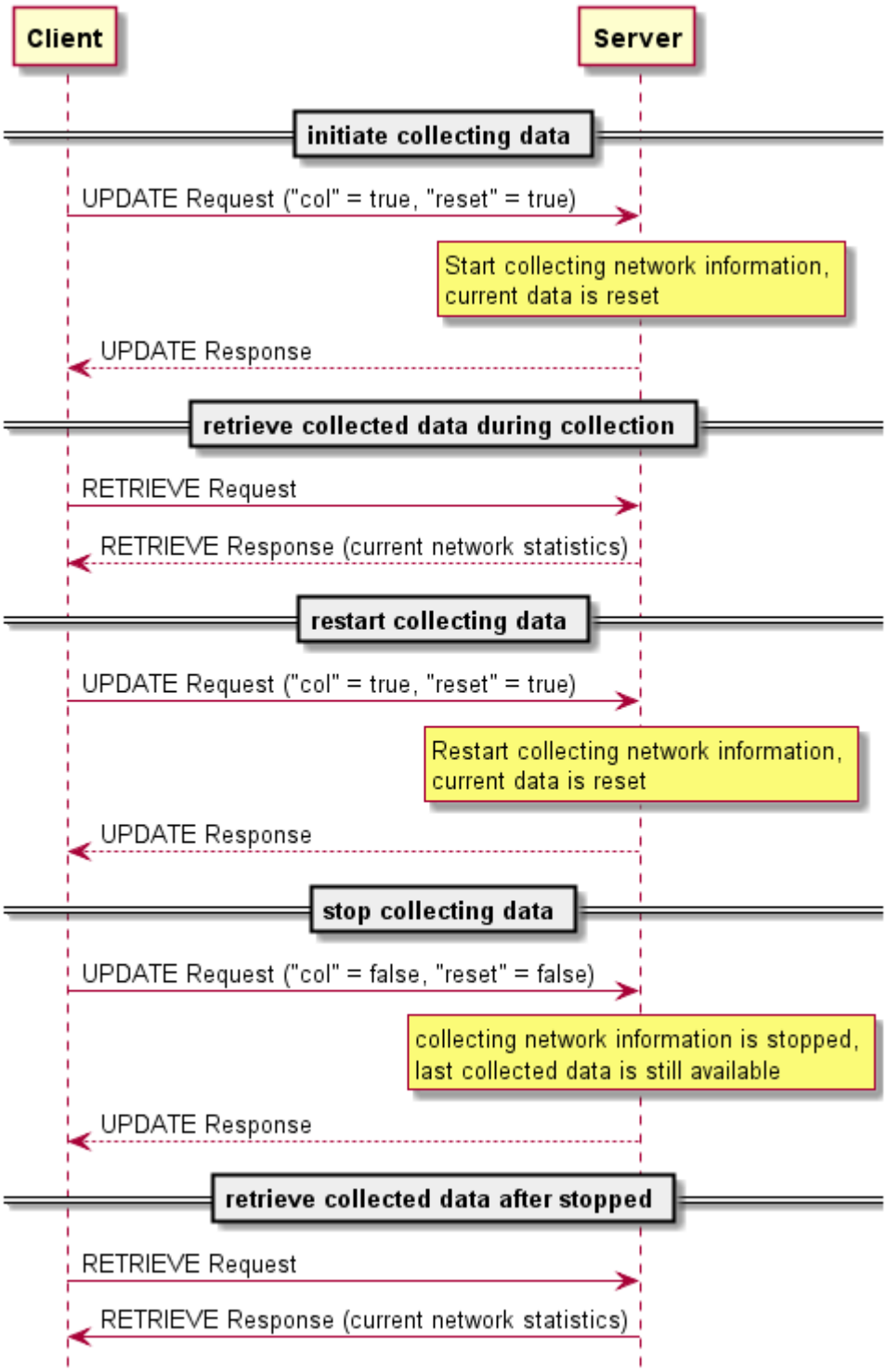


Figure 21 – Interactions with the network monitoring Resource

The state transition diagram for collecting or not collecting network information is described by Figure 22.

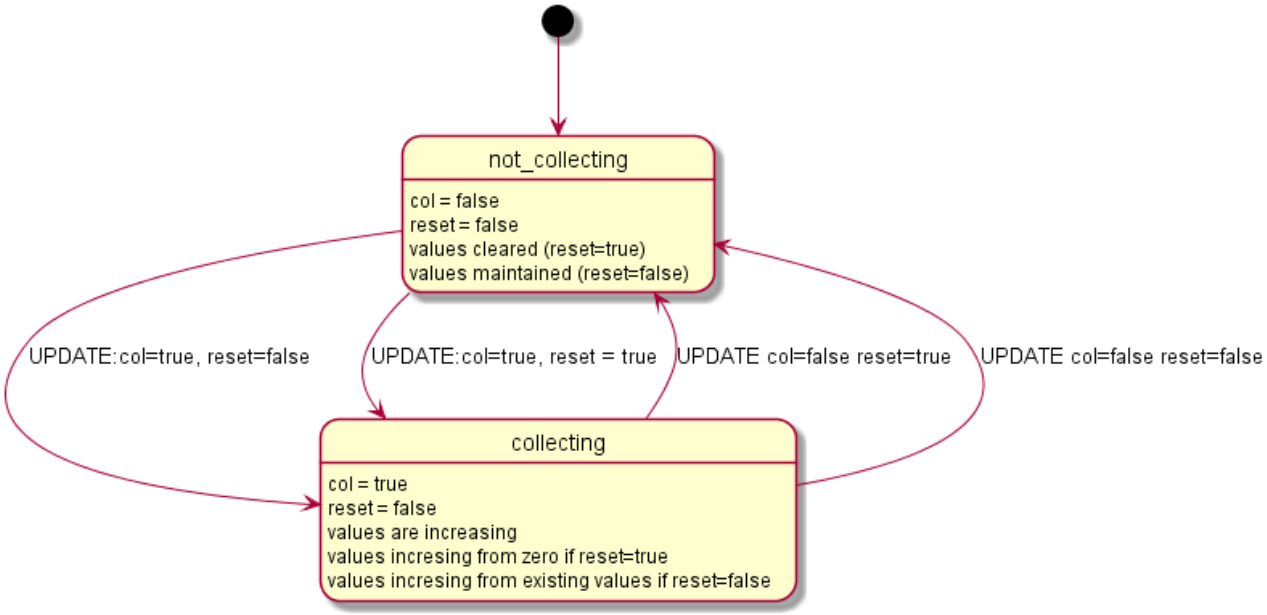


Figure 22 – State transition diagram of collecting network information

11.6 Scenes

11.6.1 Introduction

Scenes are a mechanism for automating certain operations.

A Scene is a static entity that stores a set of defined Property values for a Collection of Resources. Scenes provide a mechanism to store a setting over multiple Resources that may be hosted by multiple separate Servers. Scenes, once set up, can be used by multiple Clients to recall a setup.

Scenes can be grouped and reused, a group of Scenes is also a Scene.

In short, Scenes are bundled user settings.

11.6.2 Scenes

11.6.2.1 Introduction

Scenes are described by means of Resources. The Scene Resources are hosted by a Server and the top level Resource is listed in "/oic/res". This means that a Client can determine if the Scene functionality is hosted on a Server via Resource discovery as defined in 11.3. The setup of Scenes is driven by Client interactions. This includes creating new Scenes, and mappings of Server Properties that are part of a Scene.

The Scene functionality is created by multiple Resources and has the structure depicted in Figure 23. The sceneList and sceneCollection Resources are overloaded Collection Resources. The sceneCollection Resource contains a list of Scenes. This list contains zero or more Scenes. The sceneMember Resource contains the mapping between a Scene and what needs to happen according to that Scene on an indicated Resource.

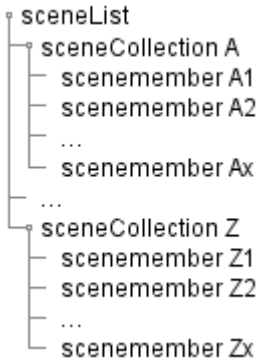


Figure 23 – Generic Scene Resource structure

11.6.2.2 Scene creation

A Client desiring to interact with Scenes needs to first determine if the Server supports the Scene feature; the sceneMembers of a Scene that are Resources of end Device being updated by the Scene change do not have to be co-located on the Server supporting the Scene feature. This can be done by checking if "/oic/res" contains the "rt" of the sceneList Resource. This is depicted in first steps of Figure 24. The sceneCollection Resource is created by the Server using some out of bound mechanism, Client creation of Scenes is not supported at this time. This will entail defining the Scene with an applicable list of Scene Values and the mappings for each Resource being part of the Scene. The mapping for each Resource being part of the sceneCollection Resource is described by a Resource called sceneMember. The sceneMember Resource contains the link to a Resource and the mapping between the Scene listed in the "sceneValues" Property and the actual Property value of the Resource indicated by the Link.

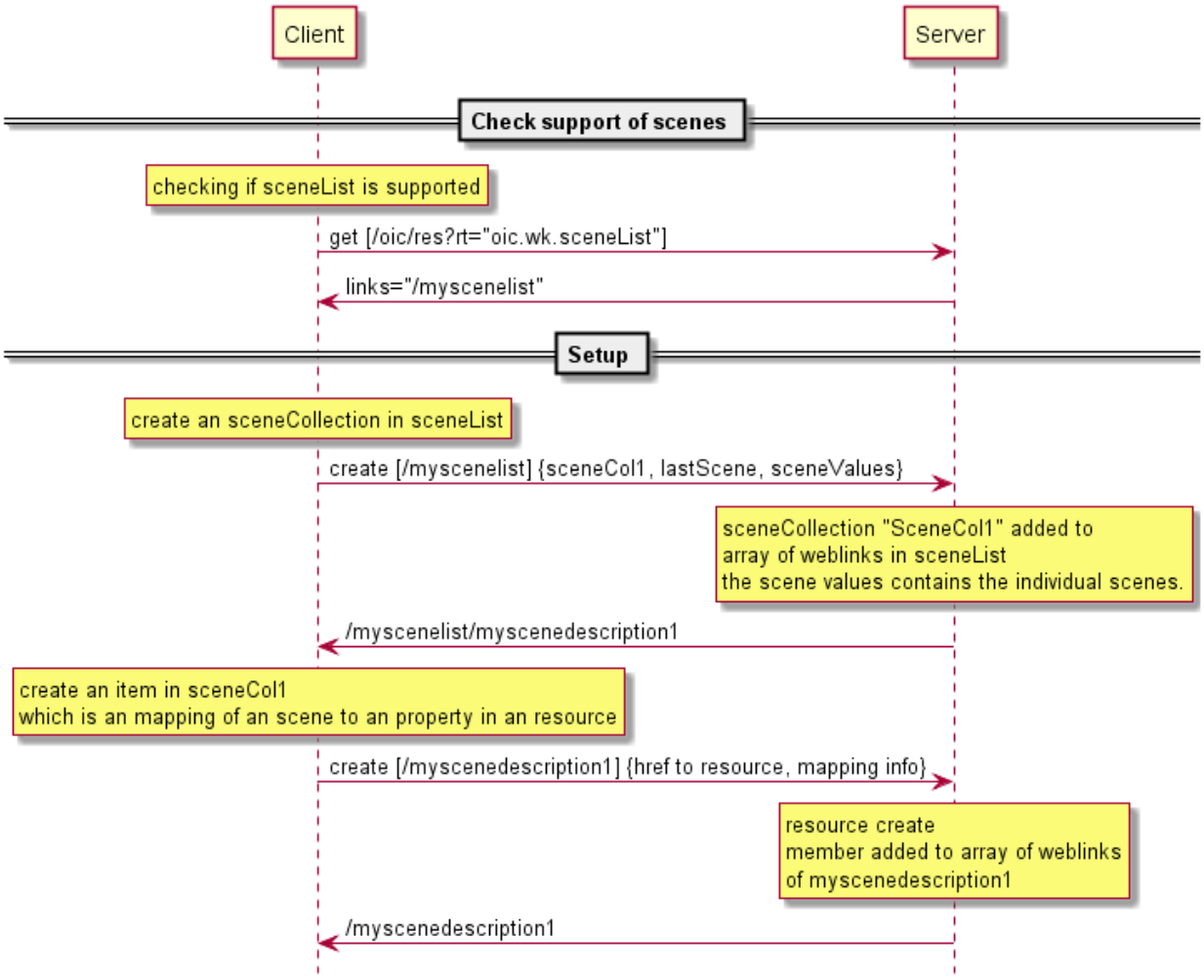


Figure 24 – Interactions to check Scene support and setup of specific Scenes

11.6.2.3 Interacting with Scenes

All capable Clients can interact with Scenes. The allowed Scene Values and the last applied Scene Value can be retrieved from the Server hosting the Scene. The Scene Value shall be changed by issuing an UPDATE operation with a payload that sets the "lastScene" Property to one of the listed allowed Scene Values. These steps are depicted in Figure 25. Note that the "lastScene" Property value does not imply that the current state of all Resources that are part of the Scene will be at the mapped value. This is due to that the setting the Scene Values are not modelled as actual states of the system. This means that another Client can change just one Resource being part of the Scene without having feedback that the state of the Scene is changed.

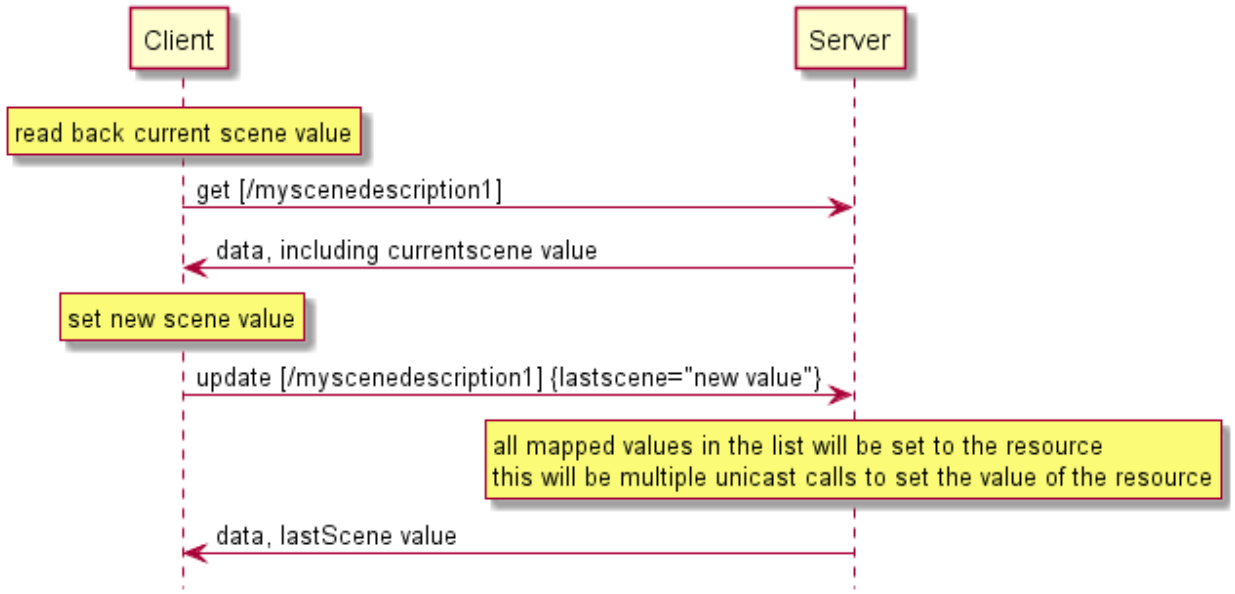
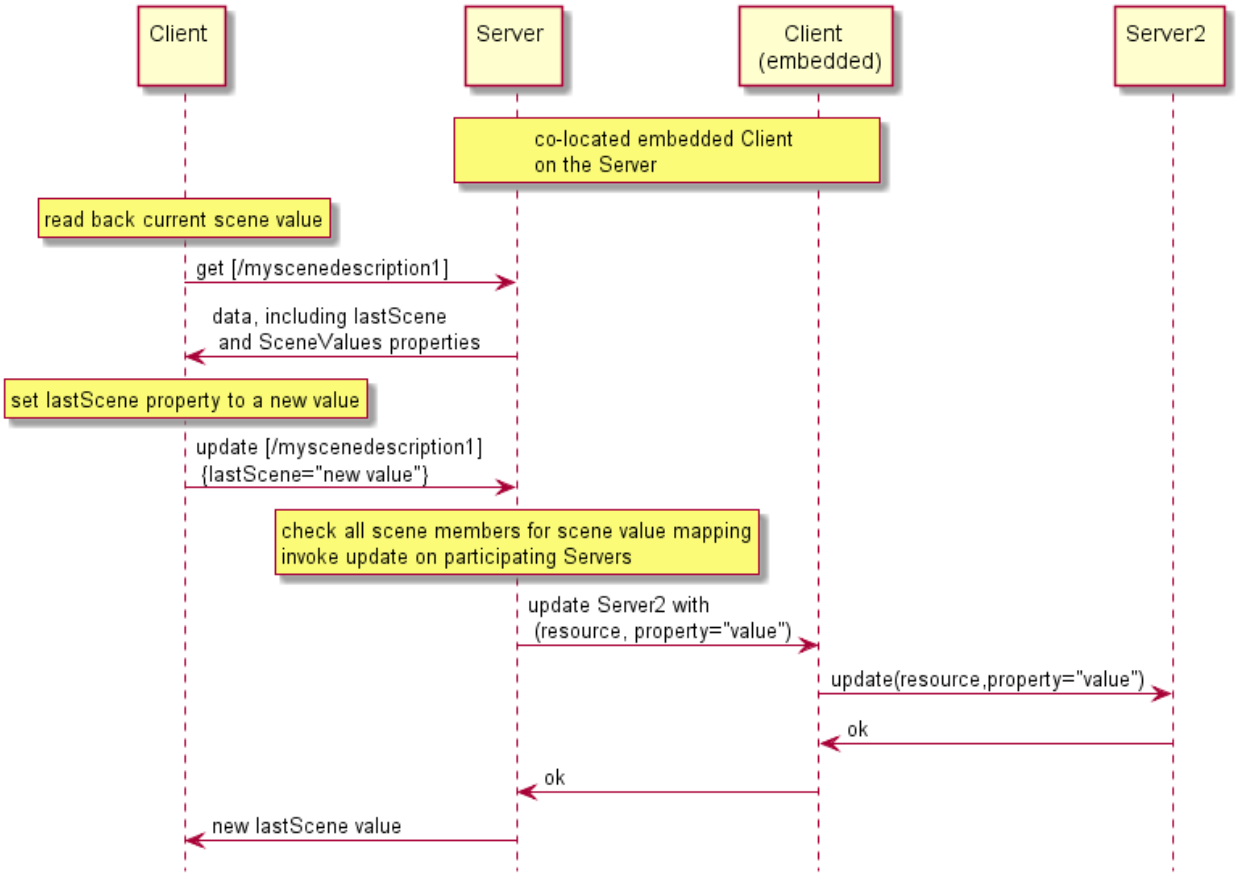


Figure 25 – Client interactions on a specific Scene

As described previously, a Scene can reference one or more Resources (i.e., sceneMembers) that are present on one or more Servers. The Scene Members are re-evaluated each time a Scene change takes place. This evaluation is triggered by a Client that is either embedded as part of the Server hosting the Scene, or separate to the Server having knowledge of the Scene via a RETRIEVE operation, Observing the referenced Resources using the mechanism described in 11.4.2. The embedded Client located in the same Device with the Server is a general Client but interacts only with Scene functionalities. During the evaluation the mappings for the new Scene Value will be applied to the Servers which contain sceneMembers from the Scene that is being updated. This behaviour is depicted in Figure 26.



3887

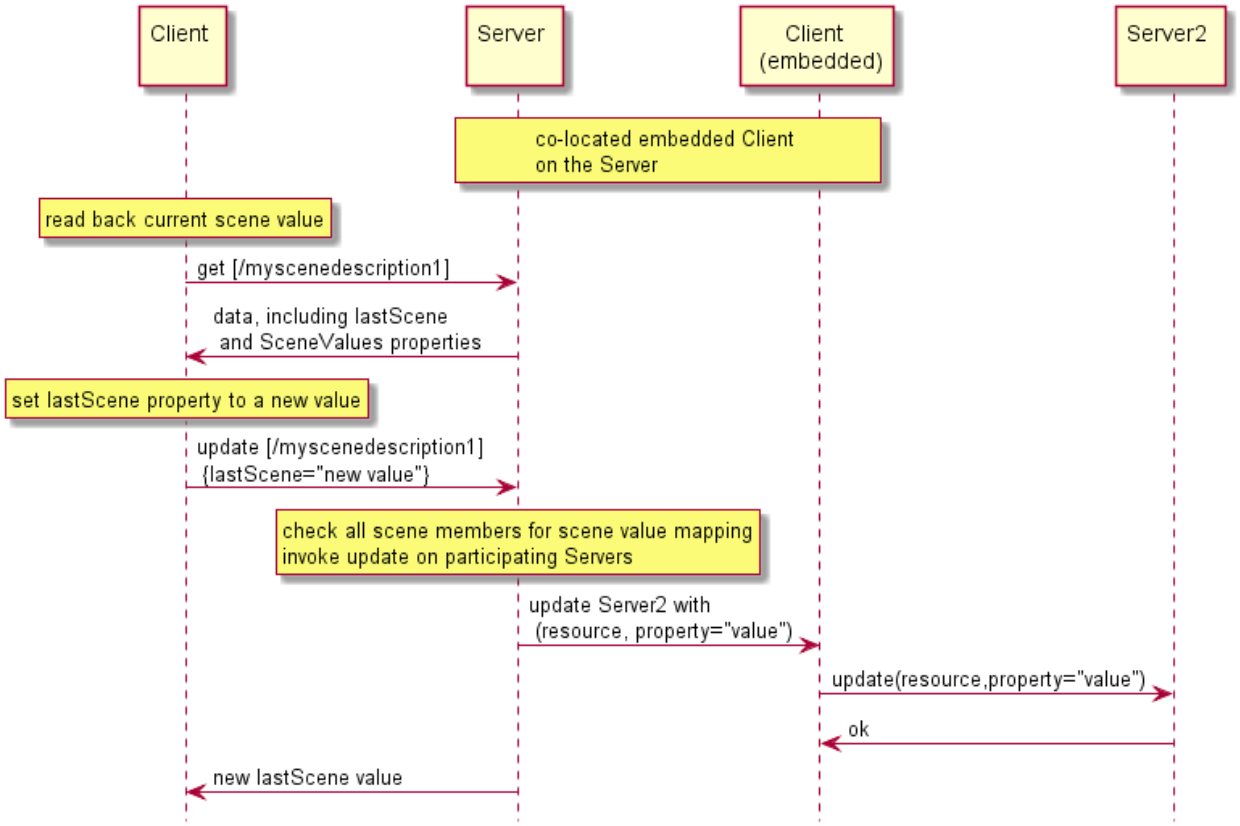


Figure 26 – Interaction overview due to a Scene change

11.6.2.4 Summary of Resource Types defined for Scene functionality

Table 35 summarizes the list of Resource Types that are part of Scenes.

Table 35 – list of Resource Types for Scenes

Friendly Name (informative)	Resource Type (rt)	Short Description	Clause
sceneList	"oic.wk.scenelist"	Top Level Collection containing sceneCollections	N/A
sceneCollection	"oic.wk.scenecollection"	Description of zero or more scenes	N/A
sceneMember	"oic.wk.scenemember"	Description of mappings for each specific Resource part of the sceneCollection	N/A

11.6.3 Security considerations

Creation of Scenes on a Server that is capable of this functionality is dependent on the ACLs applied to the Resources and the Client having the appropriate permissions. Interaction between a Client (embedded or separate) and a Server that hosts the Resource that is referenced as a Scene Member is contingent on the Client having appropriate permissions to access the Resource on the host Server.

See ISO/IEC 30118-2:2018 for details on the use of ACLs and also the mechanisms around Device Authentication that are necessary to ensure that the correct permissions exist for the Client to access the Scene Member Resource(s) on the Server.

11.7 Icons

11.7.1 Overview

Icons are a primitive that are needed by various OCF subsystems, such as bridging. An optional Resource Type of "oic.r.icon" has been defined to provide a common representation of an icon Resource that can be used by Devices.

11.7.2 Resource

The icon Resource is as defined in Table 36.

Table 36 – Optional Icon Core Resource

Example URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
"/example/oic/icon"	Icon	"oic.r.icon"	"oic.if.r"	The Resource through which the Device can obtain icon images. The Properties exposed by "/example/oic/mnt" are listed in Table 37.	Icon

Table 37 defines the details for the "oic.r.icon" Resource Type.

Table 37 – "oic.r.icon" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Mime Type	"mimetype"	"string"	N/A	N/A	R	Yes	Specifies the format (media type) of the icon. It should be a template string as specified in IANA Media Types Assignment
Width	"width"	"integer"	>= 1	pixels	R	Yes	Width of the icon in pixels greater than or equal to 1.
Height	"height"	"integer"	>= 1	pixels	R	Yes	Height of the icon in pixels greater than or equal to 1.
Icon	"media"	"uri"	N/A	N/A	R	Yes	URI to the location of the icon image.

11.8 Introspection

11.8.1 Overview

Introspection is a mechanism to announce the capabilities of Resources hosted on the Device.

The intended usage of the Introspection Device Data (IDD) is to enable dynamic Clients e.g. Clients that can use the IDD) to generate dynamically a UI or dynamically create translations of the hosted Resources to another eco-system. Other usages of Introspection is that the information can be used to generate Client code. The IDD is designed to augment the existing data already on the

wire. This means that existing mechanisms need to be used to get a full overview of what is implemented in the Device. For example, the IDD does not convey information about Observability, since that is already conveyed with the "p" Property on the Links in "/oic/res" (see 7.8.2.2.2).

The IDD is recommended to be conveyed as static data. Meaning that the data does not change during the uptime of a Device. However, when the IDD is not static, the Introspection Resource shall be Observable and the url Property Value of "oic.wk.introspection" Resource shall change to indicate that the IDD is changed.

The IDD describes the Resources that make up the Device. For the complete list of included Resources Table 20. The IDD is described as a OpenAPI 2.0 in JSON format file. The text in the following bulleted list contains OpenAPI 2.0 terms, such as paths, methods etc. The OpenAPI 2.0 file shall contain the description of the Resources:

- The IDD will use the HTTP syntax, e.g., define the CRUDN operation as HTTP methods and use the HTTP status codes.
- The IDD does not have to define all the status codes that indicate an error situation.
- The IDD does not have to define a schema when the status code indicates that there is no payload (see HTTP status code 204 as an example).
- The paths (URLs) of the Resources in the IDD shall be without the OCF Endpoint description, e.g. it shall not be a fully-qualified URL but only the relative path from the OCF Endpoint, aka the "href". The relative path may include a query parameter (e.g. "?if=oic.if.ll"), in such cases the text following (and including) the "?" delimiter shall be removed before equating to the "href" that is conveyed by "/oic/res".
- The following Resources shall be excluded in the IDD:
  - Resource with Resource Type: "oic.wk.res" unless 3rd party defined or optional Properties are implemented.
  - Resource with Resource Type: "oic.wk.introspection".
  - Resources that handle Wi-Fi Easy Setup, see OCF Easy Wi-Fi Setup.
- The following Resources shall be included in the IDD when optional or 3rd party defined Properties are implemented:
  - Resources with type: "oic.wk.p" and "oic.wk.d" (e.g. discovery related Resources).
  - Security Virtual Resources from ISO/IEC 30118-2:2018.
- When the Device does not expose instances of Vertical Resource Types, and does not have any 3rd party defined Resources (see 7.8.4.4), and does not need to include Resources in the IDD due to other clauses in this clause, then the IDD shall be an empty OpenAPI 2.0 file. An example of an empty OpenAPI 2.0 file can be found in found in Annex **F.2**.
- All other Resources that are individually addressable by a Client (i.e. the "href" can be resolved and at least one operation is supported with a success path response) shall be listed in the IDD.
- Per Resource the IDD shall include:
  - All implemented methods
    - For an OCF defined Resource Type, only the methods that are listed in the OpenAPI 2.0 definition are allowed to exist in the IDD. For an OCF defined Resource Type, methods not listed in the OpenAPI 2.0 definition shall not exist in the IDD. The supported methods contained in the IDD shall comply with the listed OCF Interfaces. For example, if the POST method is listed in the IDD, then an OCF Interface that allows UPDATE will be listed in the IDD.
  - Per supported method:
    - Implemented query parameters per method.

- This includes the supported OCF Interfaces ("if") as enum values.
- Schemas of the payload for the request and response bodies of the method.
- Where the schema provides the representation of a batch request or response ("oic.if.b") the schema shall contain the representations for all Resource Types that may be included within the batch representation. The representations shall be provided within the IDD itself.
- The schema data shall be conveyed by the OpenAPI 2.0 schema.
- The OpenAPI 2.0 schema object shall comply with:
  - The schemas shall be fully resolved, e.g. no references shall exist outside the OpenAPI 2.0 file.
  - The schemas shall list which OCF Interfaces are supported on the method.
  - The schemas shall list if a Property is optional or required.
  - The schemas shall include all Property validation keywords. Where an enum is defined the enum shall contain the values supported by the Device. When vendor defined extensions exist to the enum (defined in accordance to 7.8.4.4) these shall be included in the enum.
  - The schemas shall indicate if an Property is read only or read-write.
    - By means of the readOnly schema tag belonging to the Property.
    - Default value of readOnly is false as defined by OpenAPI 2.0.
  - The default value of the "rt" Property shall be used to indicate the supported Resource Types.
  - oneOf and anyOf constructs are allowed to be used as part of a OpenAPI 2.0 schema object. The OpenAPI 2.0 schema with oneOf and anyOf constructs can be found in Annex **F.1**.

– For Atomic Measurements (see clause 7.8.4), the following apply:
  - The "rts" Property Value in the IDD shall include only the Resource Types the instance contains and not the theoretical maximal set allowed by the schema definition.
  - The Resources that are part of an Atomic Measurement, excluding the Atomic Measurement Resource itself, shall not be added to their own individual path in the IDD, as they are not individually addressable; however, the schemas for the composed Resource Types shall be provided in the IDD as part of the batch response definition along with the "href" for the Resource.

Dynamic Resources (e.g. Resources that can be created on a request by a Client) shall have a URL definition which contains a URL identifier (e.g. using the {} syntax). A URL with {} identifies that the Resource definition applies to the whole group of Resources that may be created. The actual path may contain the Collection node that links to the Resource.

Example of a URL with identifiers:

```
/SceneListResURI/{SceneCollectionResURI}/{SceneMemberResURI}:
```

When different Resource Types are allowed to be created in a Collection, then the different schemas for the CREATE method shall define all possible Resource Types that may be created. The schema construct oneOf allows the definition of a schema with selectable Resources. The oneOf construct allows the integration of all schemas and that only one existing sub schema shall be used to indicate the definition of the Resource that may be created.

Example usage of oneOf JSON schema construct is shown in Figure 27:

```
{
```



```
  "oneOf": [
    { <<subschema 1 definition>> },
    { << sub schema 2 definition >> }
…
  ]
}
```

Figure 27 – Example usage of oneOf JSON schema

A Client using the IDD of a Device should check the version of the supported IDD of the Device. The OpenAPI 2.0 version is indicated in each file with the tag "swagger". Example of the 2.0 supported version of the tag is: "swagger": "2.0". Later versions of the specification may reference newer versions of the OpenAPI specification, for example 3.0.

A Device shall support one Resource with a Resource Type of "oic.wk.introspection" as defined in Table 38. The Resource with a Resource Type of "oic.wk.introspection" shall be included in the Resource "/oic/res".

An empty IDD file, e.g. no URLs are exposed, shall still have the mandatory OpenAPI 2.0 fields. See OpenAPI specification. An example of an empty OpenAPI 2.0 file can be found in found in Annex F.2.

Table 38 – Introspection Resource

Pre-defined URI	Resource Type Title	Resource Type ID ("rt" value)	OCF Interfaces	Description	Related Functional Interaction
none	Introspection	"oic.wk.introspection"	"oic.if.r"	The Resource that announces the URL of the Introspection file.	Introspection

Table 39 defines "oic.wk.introspection" Resource Type.

Table 39 – "oic.wk.introspection" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
urlInfo	"urlInfo"	"array"	N/A	N/A	R	Yes	array of objects
url	"url"	"string"	"uri"	N/A	R	Yes	URL to the hosted payload
protocol	"protocol"	"string"	"enum"	N/A	R	Yes	Protocol definition to retrieve the Introspection Device Data from the url.
content-type	"content-type"	"string"	"enum"	N/A	R	No	content type of the url.
version	"version"	"integer"	"enum"	N/A	R	No	Version of the Introspection protocol, indicates which rules are applied on the Introspection Device Data regarding the content of the OpenAPI 2.0 file. Current value is 1.

11.8.2 Usage of Introspection

The Introspection Device Data is retrieved in the following steps and as depicted in Figure 28:

– Check if the Introspection Resource is supported and retrieve the URL of the Resource.

– Retrieve the contents of the Introspection Resource
– Download the Introspection Device Data from the URL specified the Introspection Resource.
– Usage of the Introspection Device Data by the Client

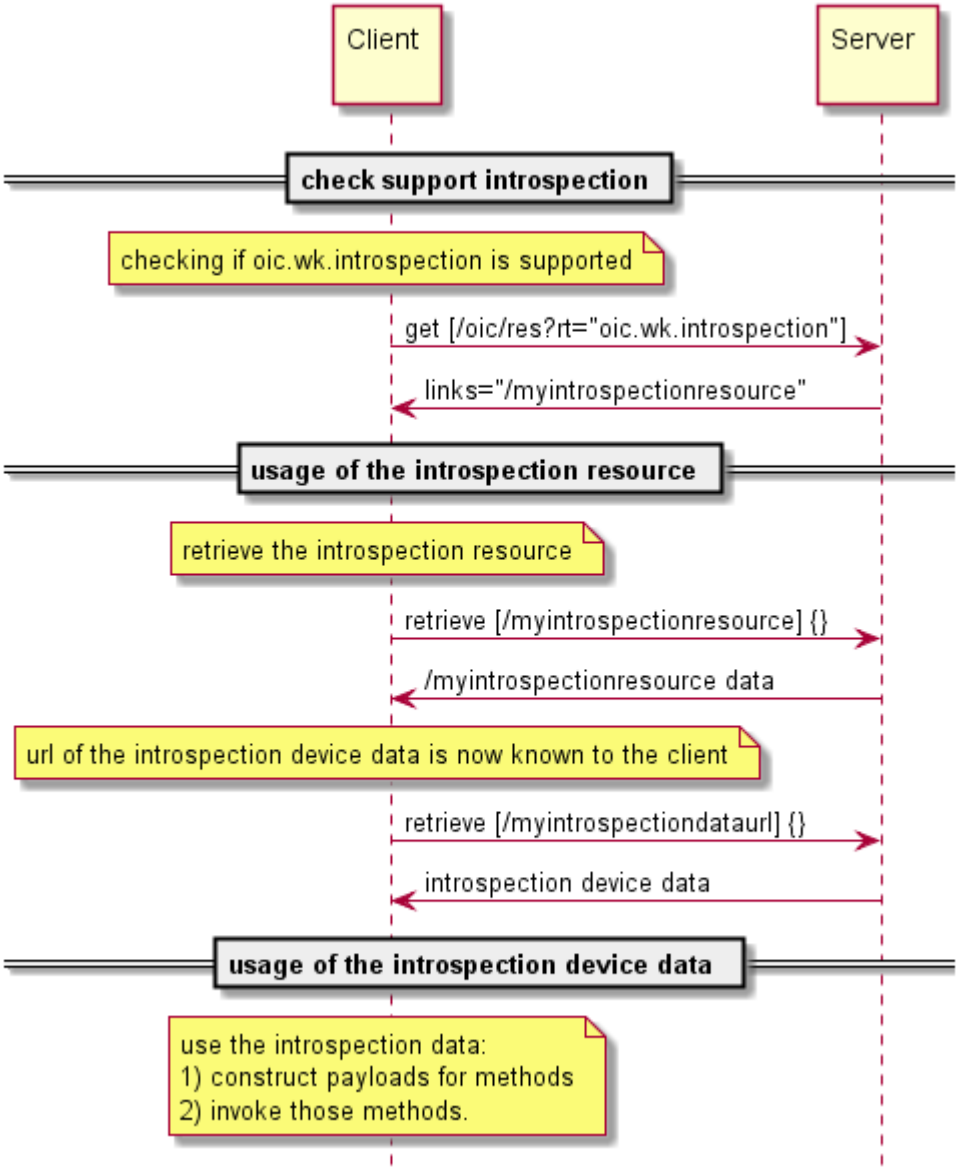


Figure 28 – Interactions to check Introspection support and download the Introspection Device Data.

11.9 Alerts

11.9.1 Overview

Alerts provide a means by which a Device provides information to an interested party with regard to error or other conditions that the Device is experiencing. An Alert contains human readable text that is dependent on the Device itself and the condition being reported. A Device may expose discrete instances of an Alert Resource Type ("oic.r.alert") or may also expose Alerts within an

Alert Collection ("oic.r.alertcollection"). If the instance of "oic.r.alertcollection" is Observable (see clause 7.8.2.2.2) then a Client may Observe the Collection using the mechanisms defined in clause 11.4. As the Device adds and removes Alerts from the Collection notifications may be generated for any registered Observers, the format of which is dependent upon the OCF Interface used for the initial Observe, see clause 7.6.3.

11.9.2 Resource Types

The Alert and Alert Collection Resource Types are as defined in Table 40.

Table 40 – Optional Alert Core Resources

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
"/example/alertURI"	Alert	"oic.r.alert"	"oic.if.r", "oic.if.baseline"	The Resource through which the Device exposes Alerts. The Properties exposed by "oic.r.alert" are listed in Table 41.	Alerts
"/example/alertcollectionURI"	Alert Collection	"oic.r.alertcollection"	"oic.if.ll", "oic.if.b", "oic.if.baseline"	A specialisation of a Collection that contains only instances of "oic.r.alert" that may be Observed by a Client in order to consume Alerts as they are created by the Device.	Alerts

Table 41 defines the details for the "oic.r.alert" Resource Type.

Table 41 – "oic.r.alert" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Category	"category"	"string"			R	Yes	Device defined category for the Alert (e.g. "System", "I/O")
Generated Time	"generatedtime"	"date-time"			R	Yes	IETF RFC 3339 formatted time at which the Alert was generated.
Originator ID	"originatorid"	"string"			R	Yes	Identity of the originator of the Alert. May be the Device ID of the Device or some other Device defined identity.
Severity	"severity"	"integer"	[0,7]		R	Yes	IETF RFC 5424 defined severity value
Subject	"subject"	"array"			R	No	Human-friendly subject of the Alert in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the subject of the Alert name in the indicated language.
Account ID	"accounted"	"string"			R	No	Identity of the account with which the Device

							generating this Alert is associated.

The Alert Collection ("oic.r.alertcollection") Resource Type defines no Properties additional to those defined for all instances of a Collection in Table 9. However the Alert Collection does impose restrictions of the values that shall be populated in the "rt" and "rts" Properties. These are described in Table 42.

Table 42 – "oic.r.alertcollection" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Links	"links"	"array"	See Table 11		R	Yes	See Table 11
Resource Type	"rt"	"array"	["oic.r.alertcollection"]		R	Yes	See Table 4.
Resource Types	"rts"	"array"	["oic.r.alert"] or ["oic.r.alert","oic.r.value.conditional"]		R	Yes	See Table 11

11.9.3 Example of Use

Consider a Device that is capable of generating Alerts; it exposes an empty instance of an Alert Collection ("oic.r.alertcollection"); that is the array of Links (the "links" Property) contains no items.

As the Device under whatever conditions generates Alerts, the Device adds a Link to the Alert Resource in the instance of an Alert Collection. A Client that has discovered the Device and is Observing the Alert Collection using the links list OCF Interface ("oic.if.ll") will receive a notification containing the complete Alert Collection (not just any Links that were added). It is the responsibility of the Client to determine which Links were added (or removed if the Alert was removed); noting that the "generatedtime" Property may be used to determine the generated order. The Client then retrieves the Alert itself via a RETRIEVE to the "href" Link Parameter in the newly added Link to the Collection.

See D.17 for an example of an Alert Resource and the applicable schema.

12 Messaging

12.1 Introduction

This clause specifies the protocol messaging mapping to the CRUDN messaging operations (clause 8) for each messaging protocol specified (e.g., CoAP.). Mapping to additional protocols is expected in later version of this document. All the Property information from the Resource model shall be carried within the message payload. This payload shall be generated in the Resource model layer and shall be encapsulated in the data connectivity layer. The message header shall only be used to describe the message payload (e.g., verb, mime-type, message payload format), in addition to the mandatory header fields defined in messaging protocol (e.g., CoAP) specification. If the message header does not support this, then this information shall also be carried in the message payload. Resource model information shall not be included in the message header structure unless the message header field is mandatory in the messaging protocol specification.

When a Resource is specified with a RESTful description language like OpenAPI 2.0 then the HTTP syntax definitions are used in the description (e.g., HTTP syntax for the CRUDN operations, status codes, etc). The HTTP syntax will be mapped to the actual used web transfer protocol (e.g., CoAP).

12.2 Mapping of CRUDN to CoAP

12.2.1 Overview

A Device implementing CoAP shall conform to IETF RFC 7252 for the methods specified in clause 12.2.3. A Device implementing CoAP shall conform to IETF RFC 7641 to implement the CoAP Observe option. Support for CoAP block transfer when the payload is larger than the MTU is defined in 12.2.8.

12.2.2 URIs

An OCF: URI is mapped to a coap: URI by replacing the scheme name "ocf" with "coap" if unsecure or "coaps" if secure before sending over the network by the requestor. Similarly on the receiver side, the scheme name is replaced with "ocf".

Any query string that is present within the URI is encoded as one or more URI-Query Options as defined in IETF RFC 7252 clause 6.4.

12.2.3 CoAP method with request and response

12.2.3.1 Overview

Every request has a CoAP method that realizes the request. The primary methods and their meanings are shown in Table 43, which provides the mapping of GET/PUT/POST/DELETE methods to CREATE, RETRIEVE, UPDATE, and DELETE operations. The associated text provides the generic behaviours when using these methods, however Resource OCF Interfaces may modify these generic semantics. The HTTP codes in the RESTful descriptions will be translated as described in IETF RFC 8075 clause 7 Response Code Mapping.

Table 43 – CoAP request and response

Method for CRUDN	(mandatory) Request data	(mandatory) Response data
GET for RETRIEVE	- **Method code**: GET (0.01). - **Request URI**: an existing URI for the Resource to be retrieved	- **Response code**: success (2.xx) or error (4.xx or 5.xx). - **Payloa**d: Resource representation of the target Resource (when successful).
POST for CREATE	- **Method code**: POST (0.02). - **Request URI**: an existing URI for the Resource responsible for the creation. - **Payload**: Resource presentation of the Resource to be created.	- **Response code**: success (2.xx) or error (4.xx or 5.xx). - **Payload**: the URI of the newly created Resource (when successful).
PUT for CREATE	- **Method code**: PUT (0.03). - **Request URI**: a new URI for the Resource to be created. - **Payload**: Resource presentation of the Resource to be created.	- **Response code**: success (2.xx) or error (4.xx or 5.xx).
POST for UPDATE	- **Method code**: POST (0.02). - **Request URI**: an existing URI for the Resource to be updated. - **Payload**: representation of the Resource to be updated.	- **Response Code**: success (2.xx) or error (4.xx or 5.xx).
DELETE for DELETE	**- Method code**: DELETE (0.04). - **Request URI**: an existing URI for the Resource to be deleted.	- **Response code**: success (2.xx) or error (4.xx or 5.xx).

12.2.3.2 CREATE with POST or PUT

12.2.3.2.1 With POST

POST shall be used only in situations where the request URI is valid, that is it is the URI of an existing Resource on the Server that is processing the request. If no such Resource is present, the Server shall respond with an error response code of 4.xx. The use of POST for CREATE shall use an existing request URI which identifies the Resource on the Server responsible for creation. The URI of the created Resource is determined by the Server and provided to the Client in the response.

A Client shall include the representation of the new Resource in the request payload. The new resource representation in the payload shall have all the necessary Properties to create a valid Resource instance, i.e. the created Resource should be able to properly respond to the valid Request with mandatory OCF Interface (e.g., "GET with ?if=oic.if.baseline").

Upon receiving the POST request, the Server shall either:

– Create the new Resource with a new URI, respond with the new URI for the newly created Resource and a success response code (2.xx); or

– respond with an error response code (4.xx or 5.xx).

POST is unsafe and is the supported method when idempotent behaviour cannot be expected or guaranteed.

12.2.3.2.2 With PUT

PUT shall be used to create a new Resource or completely replace the entire representation of an existing Resource. The Resource representation in the payload of the PUT request shall be the complete representation. PUT for CREATE shall use a new request URI identifying the new Resource to be created.

The new Resource representation in the payload shall have all the necessary Properties to create a valid Resource instance, i.e. the created Resource should be able to properly respond to the valid Request with mandatory OCF Interface (e.g. "GET with ?if=oic.if.baseline").

Upon receiving the PUT request, the Server shall either:

– Create the new Resource with the request URI provided in the PUT request and send back a response with a success response code (2.xx); or

– respond with an error response code (4.xx or 5.xx).

PUT is an unsafe method but it is idempotent, thus when a PUT request is repeated the outcome is the same each time.

12.2.3.3 RETRIEVE with GET

GET shall be used for the RETRIEVE operation. The GET method retrieves the representation of the target Resource identified by the request URI.

Upon receiving the GET request, the Server shall either:

– Send back the response with the representation of the target Resource with a success response code (2.xx); or

– respond with an error response code (4.xx or 5.xx) or ignore it (e.g. non-applicable multicast GET).

GET is a safe method and is idempotent.

12.2.3.4 UPDATE with POST

POST shall be used only in situations where the request URI is valid, that is it is the URI of an existing Resource on the Server that is processing the request. If no such Resource is present, the Server shall respond with an error response code of 4.xx. A client shall use POST to UPDATE Property values of an existing Resource.

Upon receiving the request, the Server shall either:

– Apply the request to the Resource identified by the request URI in accordance with the applied OCF Interface (i.e. POST for non-existent Properties is ignored) and send back a response with a success response code (2.xx); or

– respond with an error response code (4.xx or 5.xx). Note that if the representation in the payload is incompatible with the target Resource for POST using the applied OCF Interface (i.e. the overwrite semantic cannot be honored because of read-only Property in the payload), then the error response code 4.xx shall be returned.

POST is unsafe and is the supported method when idempotent behaviour cannot be expected or guaranteed.

12.2.3.5 DELETE with DELETE

DELETE shall be used for DELETE operation. The DELETE method requests that the Resource identified by the request URI be deleted.

Upon receiving the DELETE request, the Server shall either:

– Delete the target Resource and send back a response with a success response code (2.xx); or

– respond with an error response code (4.xx or 5.xx).

DELETE is unsafe but idempotent (unless URIs are recycled for new instances).

12.2.4 Content-Format negotiation

The Framework mandates support of CBOR, however it allows for negotiation of the payload body if more than one Content-Format (e.g. CBOR and JSON) is supported by an implementation. In this case the Accept Option defined in clause 5.10.4 of IETF RFC 7252 shall be used to indicate which Content–Format (e.g. JSON) is requested by the Client.

The Content-Formats supported are shown in Table 44.

Table 44 – OCF Content-Formats

Media Type	ID
"application/cbor"	60
"application/vnd.ocf+cbor"	10000

Clients shall include a Content-Format Option in every message that contains a payload. Servers shall include a Content-Format Option for all success (2.xx) responses with a payload body. Per IETF RFC 7252 clause 5.5.1, Servers shall include a Content-Format Option for all error (4.xx or 5.xx) responses with a payload body unless they include a Diagnostic Payload; error responses with a Diagnostic Payload do not include a Content-Format Option. The Content-Format Option shall use the ID column numeric value from Table 44. An OCF vertical may mandate a specific Content-Format Option.

Clients shall also include an Accept Option in every request message. The Accept Option shall indicate the required Content-Format as defined in Table 44 for response messages. The Server

shall return the required Content-Format if available. If the required Content-Format cannot be returned, then the Server shall respond with an appropriate error message.

12.2.5 OCF-Content-Format-Version information

Servers and Clients shall include the OCF-Content-Format-Version Option in both request and response messages with a payload. Clients shall include the OCF-Accept-Content-Format-Version Option in request messages. The OCF-Content-Format-Version Option and OCF-Accept-Content-Format-Version Option are specified as Option Numbers in the CoAP header as shown in Table 45.

Table 45 – OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Option Numbers

CoAP Option Number	Name	Format	Length (bytes)
2049	OCF-Accept-Content-Format-Version	uint	2
2053	OCF-Content-Format-Version	uint	2

The value of both the OCF-Accept-Content-Format-Version Option and the OCF-Content-Format-Version Option is a two-byte unsigned integer that is used to define the major, minor and sub versions. The major and minor versions are represented by 5 bits and the sub version is represented by 6 bits as shown in Table 46.

Table 46 – OCF-Accept-Content-Format-Version and OCF-Content-Format-Version Representation

	Major Version					Minor Version					Sub Version					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 47 illustrates several examples:

Table 47 – Examples of OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Representation

OCF version	Binary representation	Integer value
"1.0.0"	"0000 1000 0000 0000"	2048
"1.1.0"	"0000 1000 0100 0000"	2112

The OCF-Accept-Content-Format-Version Option and OCF-Content-Format-Version Option for this version of the document shall be "1.0.0" (i.e. "0b0000 1000 0000 0000").

12.2.6 Content-Format policy

All Devices shall support the current Content-Format Option, "application/vnd.ocf+cbor", and OCF-Content-Format-Version "1.0.0".

For backward compatibility with previous OCF-Content-Format-Version Options:

– All Client Devices shall support OCF-Content-Format-Version Option set to "1.0.0" and higher.

- All Client Devices shall support OCF-Accept-Content-Format-Version Option set to "1.0.0" and higher.
- A Client shall send a discovery request message with its Accept Option set to "application/vnd.ocf+cbor", and its OCF-Accept-Content-Format-Version Option matching its highest supported version.
- A Server shall respond to a Client's discovery request that is higher than its OCF-Content-Format-Version by responding with its Content-Format Option set to "application/vnd.ocf+cbor", and OCF-Content-Format-Version matching its highest supported version. The response representation shall be encoded with the OCF-Content-Format-Version matching the Server's highest supported version.
- A Server may support previous Content-Formats and OCF-Content-Format-Versions to support backward compatibility with previous versions.
- For a Server that supports multiple OCF-Content-Format-Version Options, the Server should attempt to respond with an OCF-Content-Format-Version that matches the OCF-Accept-Content-Format-Version of the request.

For optional backward compatibility with OIC 1.1:

- All Devices that claim backward compatibility to the OIC 1.1 specification shall support the "application/cbor" media type.
- For a Client supporting backward compatibility with OIC 1.1, the Client shall send a discovery request with its Accept Option set to "application/cbor" in response to an error from an OIC 1.1 Server.
- A Server supporting OIC 1.1 compatibility shall handle a Client request containing the Accept Option ="application/cbor" by responding with its Content-Format Option set to "application/cbor" and no OCF-Content-Format-Version Option.
- For more OIC 1.1 information see Annex E.

To maintain compatibility between Devices implemented to different versions of this document, Devices should follow the policy as described in Figure 29, Figure 30 and Figure 31.

The OIC 1.1 Clients and Servers represented in Figure 29 and Figure 30 support sending Content-Format Option set to "application/cbor" and Accept Option set to "application/cbor". The OIC 1.1 Clients and Servers do not support OCF-Content-Format-Version Option, nor the OCF-Accept-Content-Format-Version Option. The OCF Clients in Figure 30 and Figure 31 support sending Content-Format Option set to "application/vnd.ocf+cbor", Accept Option set to "application/vnd.ocf+cbor", OCF-Content-Format-Version Option set to "1.0.0", and OCF-Accept-Content-Format-Version Option set to "1.0.0" (representing OCF 1.0 and later Clients). The OCF Servers in Figure 29 and Figure 31 support sending Content-Format Option set to "application/vnd.ocf+cbor" and OCF-Content-Format-Version Option set to "1.0.0" (representing OCF 1.0 and later Servers).

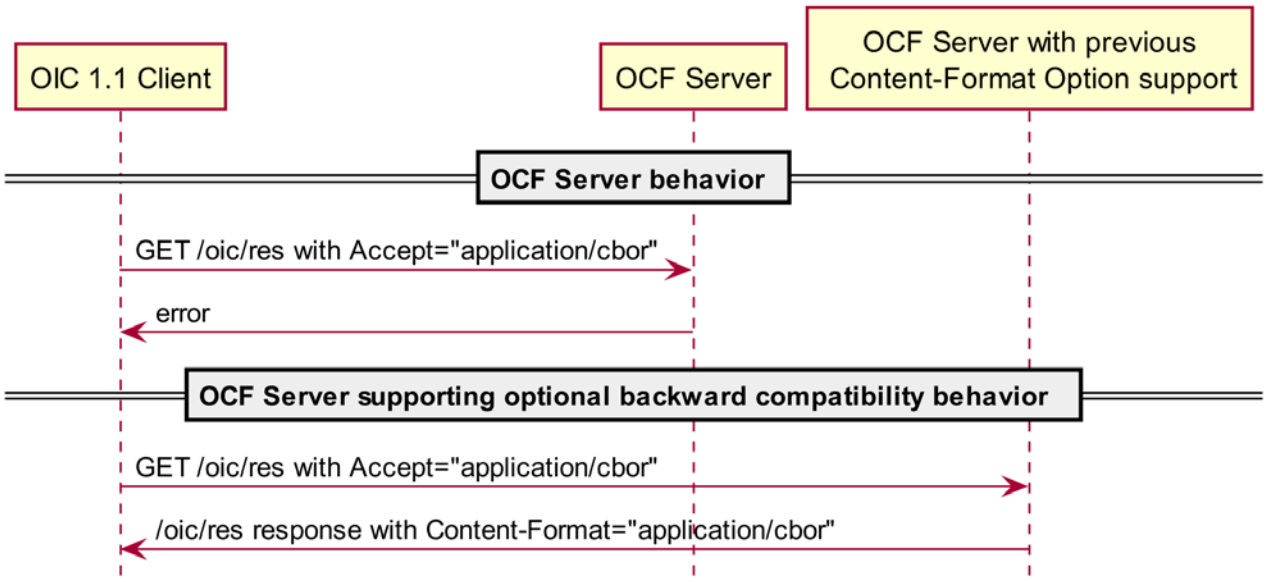


Figure 29 – Content-Format Policy for OCF Servers supporting error responses and backward compatibility responses

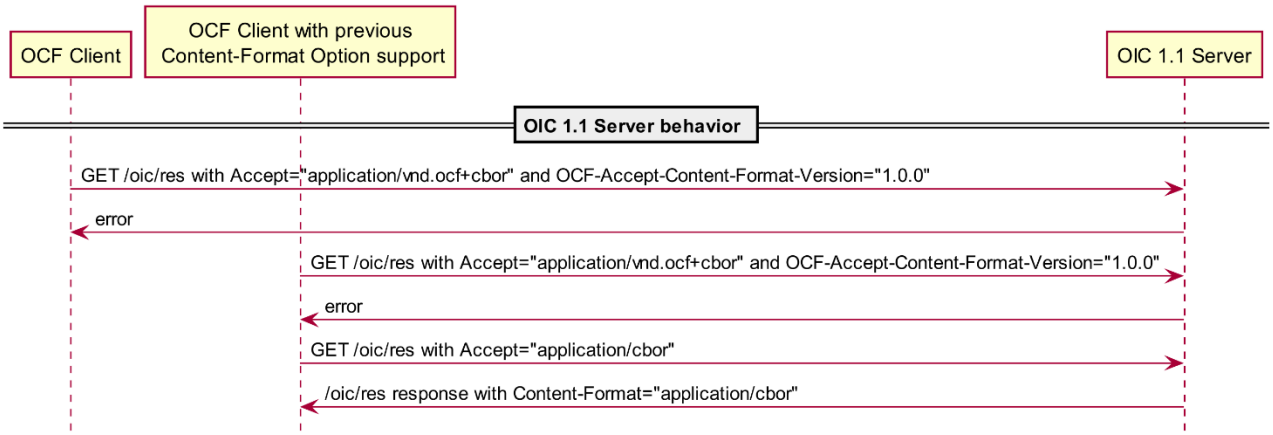


Figure 30 – Content-Format Policy for OCF Clients supporting error responses and backward compatibility responses

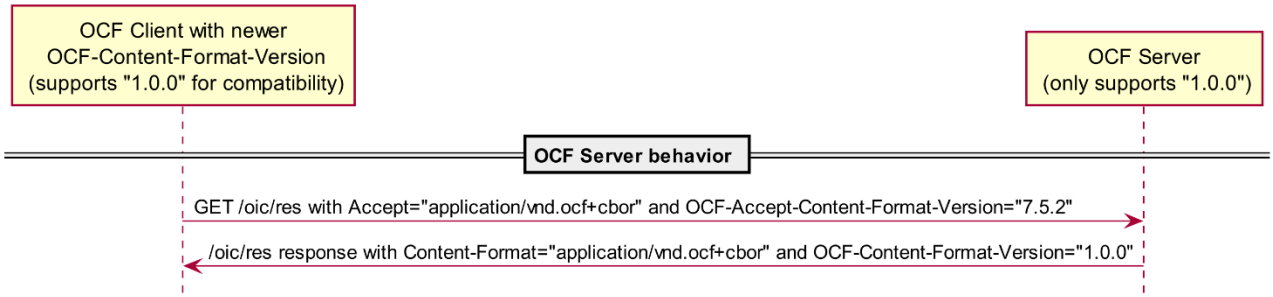


Figure 31 – Content-Format Policy for backward compatible OCF Clients negotiating lower OCF Content-Format-Version

12.2.7 CRUDN to CoAP response codes

The mapping of CRUDN operations response codes to CoAP response codes are identical to the response codes defined in IETF RFC 7252.

12.2.8 CoAP block transfer

Basic CoAP messages work well for the small payloads typical of light-weight, constrained IoT devices. However scenarios can be envisioned in which an application needs to transfer larger payloads.

CoAP block-wise transfer as defined in IETF RFC 7959 shall be used by all Servers which generate a content payload that would exceed the size of a CoAP datagram as the result of handling any defined CRUDN operation.

Similarly, CoAP block-wise transfer as defined in IETF RFC 7959 shall be supported by all Clients. The use of block-wise transfer is applied to both the reception of payloads as well as transmission of payloads that would exceed the size of a CoAP datagram.

All blocks that are sent using this mechanism for a single instance of a transfer shall all have the same reliability setting (i.e. all confirmable or all non-confirmable).

A Client may support both the block1 (as descriptive) and block2 (as control) options as described by IETF RFC 7959. A Server may support both the block1 (as control) and block2 (as descriptive) options as described by IETF RFC 7959.

12.3 Mapping of CRUDN to CoAP serialization over TCP

12.3.1 Overview

In environments where TCP is already available, CoAP can take advantage of it to provide reliability. Also in some environments UDP traffic is blocked, so deployments may use TCP. For example, consider a cloud application acting as a Client and the Server is located at the user's home. A Server which already support CoAP as a messaging protocol could easily support CoAP serialization over TCP rather than utilizing another messaging protocol. A Device implementing CoAP Serialization over TCP shall conform to IETF RFC 8323.

12.3.2 URIs

When UDP is blocked, Clients are dependent on pre-configured details of the Device to determine if the Device supports CoAP serialization over TCP. When UDP is not-blocked, a Device which supports CoAP serialization over TCP shall populate the "eps" Parameter in the "/oic/res" response, as defined in 10.2, with the URI scheme(s) as defined in clause 8.1 or 8.2 of IETF RFC 8323. For the "coaps+tcp" URI scheme, as defined in clause 8.2 of IETF RFC 8323, IETF RFC 7301 shall be used. In addition, the URIs used for CoAP serialization over TCP shall conform to 12.2.2 by

substituting the scheme names with the scheme names defined in clauses 8.1 and 8.2 of IETF RFC 8323 respectively.

12.3.3 CoAP method with request and response

The CoAP methods used for CoAP serialization over TCP shall conform to 12.2.3.

12.3.4 Content-Format negotiation

The Content Format negotiation used for CoAP serialization over TCP shall conform to 12.2.4.

12.3.5 OCF-Content-Format-Version information

The OCF Content Format Version information used for CoAP serialization over TCP shall conform to 12.2.5.

12.3.6 Content-Format policy

The Content Format policy used for CoAP serialization over TCP shall conform to 12.2.6.

12.3.7 CRUDN to CoAP response codes

The CRUDN to CoAP response codes for CoAP serialization over TCP shall conform to 12.2.7.

12.3.8 CoAP block transfer

The CoAP block transfer for CoAP serialization over TCP shall conform to clause 6 of IETF RFC 8323.

12.3.9 Keep alive (connection health)

The Device that initiated the CoAP over TCP connection shall send a Ping message as described in clause 5.4 in IETF RFC 8323. The Device to which the connection was made may send a Ping message. The recipient of any Ping message shall send a Pong message as described in clause 5.4 in IETF RFC 8323.

Both sides of an established CoAP over TCP connection may send subsequent Ping (and corresponding Pong) messages.

12.4 Payload Encoding in CBOR

OCF implementations shall perform the conversion to CBOR from JSON defined schemas and to JSON from CBOR in accordance with IETF RFC 7049 clause 4 unless otherwise specified in this clause.

Properties defined as a JSON integer shall be encoded in CBOR as an integer (CBOR major types 0 and 1). Properties defined as a JSON number shall be encoded as an integer, single- or double-precision floating point (CBOR major type 7, sub-types 26 and 27); the choice is implementation dependent. Half-precision floating point (CBOR major 7, sub-type 25) shall not be used. Integer numbers shall be within the closed interval [-2^53, 2^53]. Properties defined as a JSON number should be encoded as integers whenever possible; if this is not possible Properties defined as a JSON number should use single-precision if the loss of precision does not affect the quality of service, otherwise the Property shall use double-precision.

On receipt of a CBOR payload, an implementation shall be able to interpret CBOR integer values in any position. If a Property defined as a JSON integer is received encoded other than as an integer, the implementation may reject this encoding using a final response as appropriate for the underlying transport (e.g. 4.00 for CoAP) and thus optimise for the integer case. If a Property is defined as a JSON number an implementation shall accept integers, single- and double-precision floating point.

13 Security

The details for handling security and privacy are specified in ISO/IEC 30118-2:2018.

Annex A
(informative)

Operation Examples

A.1 Introduction

This clause describes some example scenarios using sequence of operations between the entities involved. In all the examples illustrated in Figure A.1 *Light* is a Server and *Smartphone* is a Client. In one of the scenario *Garage* additionally acts as a Server. All the examples are based on the following example Resource definitions:

"rt=oic.example.light" with Resource Type definition as illustration in Figure A.1.

Table A-1 – "oic.example.light" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	"n"	"string"	N/A	N/A	R, W	No	N/A
on-off	"of"	"boolean"	N/A	N/A	R, W	Yes	On/Off Control: "0" = Off "1" = On
dim	"dm"	"integer"	0-255	N/A	R, W	Yes	Resource which can take a range of values minimum being "0" and maximum being "255"

rt=oic.example.garagedoor with Resource Type definition as illustration in Table A-2.

Table A-2 – oic.example.garagedoor Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	"n"	"string"	N/A	N/A	R, W	No	N/A
open-close	"oc"	"boolean"	N/A	N/A	R, W	Yes	Open/Close Control: 0 = Open 1 = Close

"/oic/mnt" ("rt=oic.wk.mnt") used in the examples in Figure A.2 is defined in 11.5.2.

A.2 When at home: From smartphone turn on a single light

This sequence highlights (Figure A.1) the discovery and control of an OCF light Resource from an OCF smartphone.

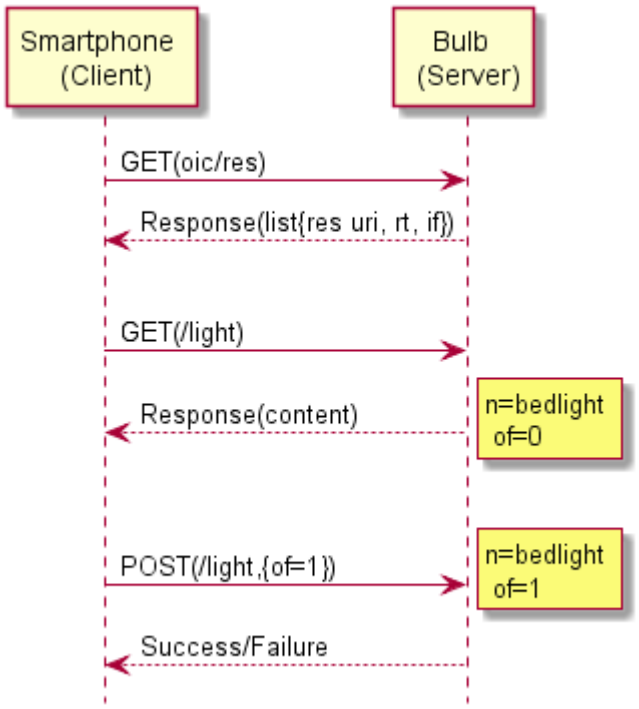


Figure A.1 – When at home: from smartphone turn on a single light

Discovery request can be sent to *All OCF Nodes* Multicast address FF0X::158 or can be sent directly to the IP address of Device hosting the light Resource.

- Smartphone sends a GET request to "/oic/res" Resource to discover all Resources hosted on targeted end point.
- The end point (bulb) responds with the list of Resource URI, Resource Type and OCF Interfaces supported on the end point (one of the Resource is "/light" whose "rt=oic.example.light").
- Smartphone sends a GET request to "/light" Resource to know its current state.
- The end point responds with representation of light Resource ({n=bedlight;of=0}).
- Smartphone changes the "of" Property of the light Resource by sending a POST request to "/light" Resource ("{of=1}").
- On Successful execution of the request, the end point responds with the changed Resource representation. Else, error code is returned. Details of the error codes are defined in 12.2.7.

A.3 GroupAction execution

This example will be added when groups feature is added in later version of the document.

A.4 When garage door opens, turn on lights in hall; also notify smartphone

This example will be added when scripts feature is added in later version of the document.

A.5 Device management

This sequence highlights (Figure A.2) the Device management function of maintenance.

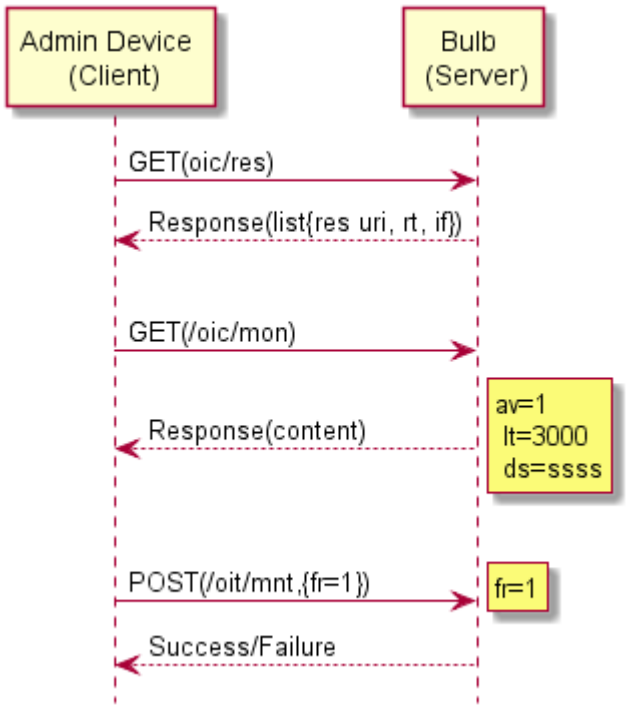


Figure A.2 – Device management (maintenance)

Pre-Condition: Admin Device has different security permissions and hence can perform Device management operations on the Device.

- Admin Device sends a GET request to "/oic/res" Resource to discover all Resources hosted on a targeted end point (in this case Bulb).
- The end point (bulb) responds with the list of Resource URI, Resource Type and OCF Interfaces supported on the end point (one of the Resources is "/oic/mnt" whose "rt=oic.wk.mnt").
- Admin Device changes the "fr" Property of the maintenance Resource by sending a POST request to "/oic/mnt" Resource ("{fr=1}"). This triggers a factory reset of the end point (bulb).
- On successful execution of the request, the end point responds with the changed Resource representation. Else, error code is returned. Details of the error codes are defined in 12.2.7.

Annex B
(informative)

OCF interaction scenarios and deployment models

B.1 OCF interaction scenarios

A Client connects to one or multiple Servers in order to access the Resources provided by those Servers. The following are scenarios representing possible interactions among roles:

– Direct interaction between Client and Server (Figure B.1). In this scenario the Client and the Server directly communicate without involvement of any other Device. A smartphone which controls an actuator directly uses this scenario.

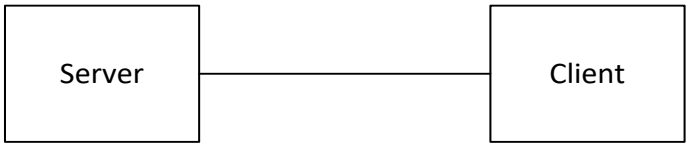


Figure B.1 – Direct interaction between Server and Client

– Interaction between Client and Server using another server (Figure B.). In this scenario, another Server provides the support needed for the Client to directly access the desired Resource on a specific Server. This scenario is used for example, when a smartphone first accesses a discovery server to find the addressing information of a specific appliance, and then directly accesses the appliance to control it.

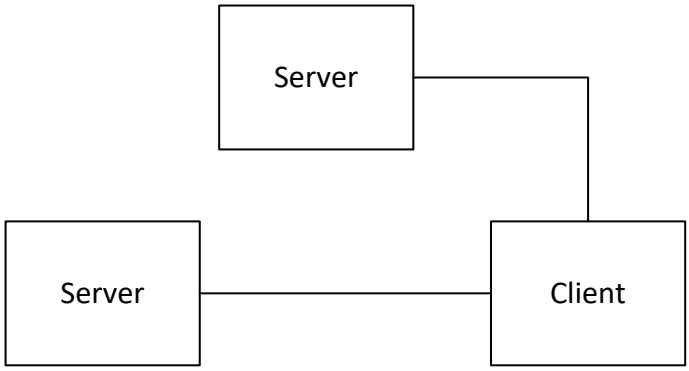


Figure B.2– Interaction between Client and Server using another Server

– Interaction between Client and Server using Intermediary (Figure B.3). In this scenario an Intermediary facilitates the interaction between the Client and the Server. A smartphone which controls appliances in a smart home via MQTT broker uses this scenario.

Figure B.3 – Interaction between Client and Server using Intermediary

– Interaction between Client and Server using support from multiple Servers and intermediary (Figure B.4). In this scenario, both Server and Intermediary roles are present to facilitate the transaction between the Client and a specific Server. An example scenario is when a

smartphone first accesses a Resource Directory (RD) server to find the address to a specific appliance, then utilizes MQTT broker to deliver a command message to the appliance. The smartphone can utilize the mechanisms defined in CoRE Resource Directory such as default location, anycast address or DHCP to discover the Resource Directory information.

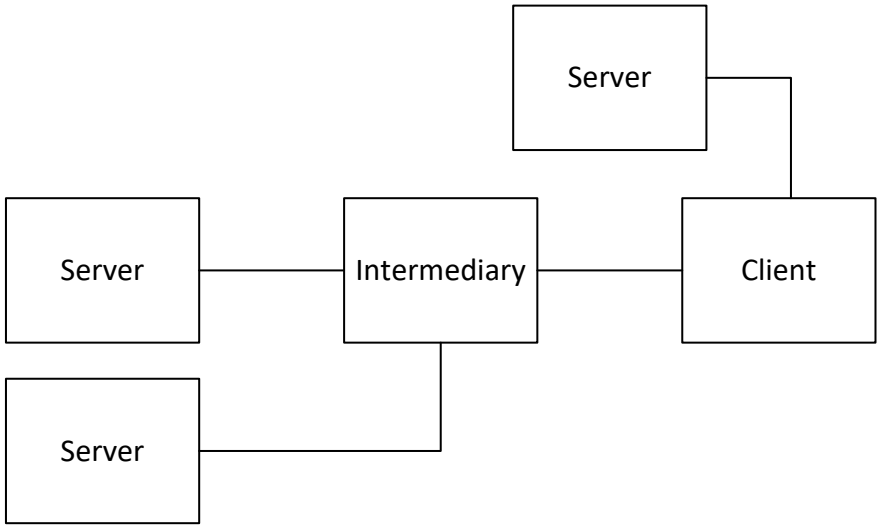


Figure B.4 – Interaction between Client and Server using support from multiple Servers and Intermediary

B.2 Deployment model

In deployment, Devices are deployed and interact via either wired or wireless connections. Devices are the physical entities that may host Resources and play one or more roles. There is no constraint on the structure of a deployment or number of Devices in it. Architecture is flexible and scalable and capable of addressing large number of Devices with different Device capabilities, including constrained Devices which have limited memory and capabilities. Constrained Devices are defined and categorized in [TCNN].

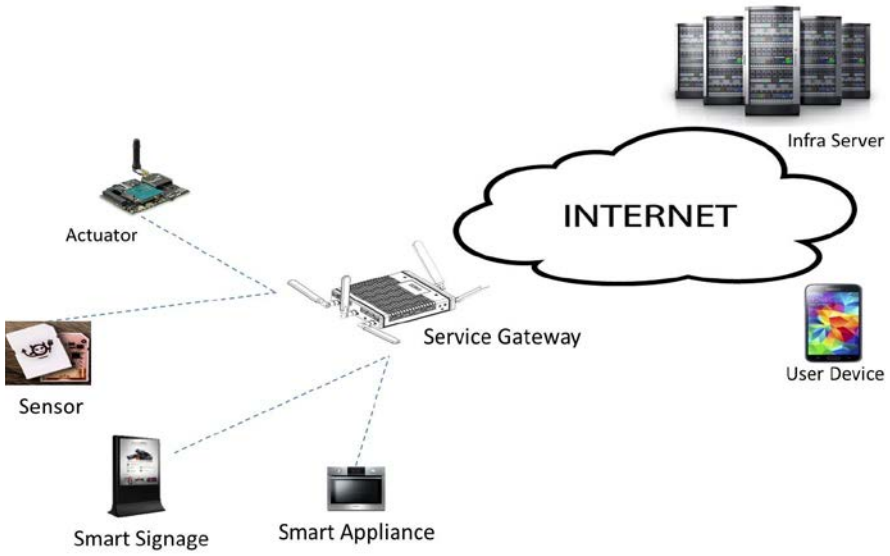


Figure B.5 – Example of Devices

Figure B.5 depicts a typical deployment and set of Devices, which may be divided in the following categories:

- *Things*: Networked Devices which are able to interface with physical environments. Things are the Devices which are primarily controlled and monitored. Examples include smart appliances, sensors, and actuators. Things mostly take the role of Sever but they may also take the role of Client, for example in machine-to-machine communications.
- *User Devices*: Devices employed by the users enabling the users to access Resources and services. Examples include smart phones, tablets, and wearable devices. User Devices mainly take the role of Client, but may also take the role of Server or Intermediary.
- *Service Gateways*: Network equipment which take the role of Intermediary. Examples are home gateways.
- *Infra Servers*: Data centers residing in cloud infrastructure, which facilitate the interaction among Devices by providing network services such as AAA, NAT traversal or discovery. It can also play the role of Client or Intermediary.

Annex C
(informative)

Other Resource models and OCF mapping

C.1 Multiple Resource models

RESTful interactions are defined dependent on the Resource model; hence, Devices require a common understanding of the Resource model for interoperability.

There are multiple Resource models defined by different organizations including OCF, OMA SpecWorks and oneM2M used in the industry, which may restrict interoperability among respective ecosystems. The main differences from Resource model are as follows:

– *Resource structure*: Resources may be defined to have Properties (e.g., oneM2M defined Resources), or may be defined as an atomic entity and not be decomposable into Properties (e.g., OMA SpecWorks defined Resources). For example, a smart light may be represented as a Resource with an on-off Property or a Resource Collection containing an on-off Resource. In the former, on-off Property doesn't have a URI of its own and can only be accessed indirectly via the Rresource. In the latter, being a Resource itself, on-off Resource is assigned its own URI and can be directly manipulated.

– *Resource name & type*: Resources may be allowed to be named freely and have their characteristics indicated using a Resource Type Property (e.g., as defined in oneM2M). Alternatively, the name of Resources may be defined a priori in a way that the name by itself is indicative of its characteristic (e.g., as defined by OMA SpecWorks). For example, in oneM2M Resource model, a smart light can be named with no restrictions, such as "LivingRoomLight_1" but in OMA SpecWorks Resource model it is required to have the fixed Object name with numerical Object ID of OMA SpecWorks Light Control ("3311"). Consequently, it's likely that in the former case the data path in URI is freely defined and in the latter case it is predetermined.

– *Resource hierarchy*: Resources may be allowed to be organized in hierarchy where a Resource contains another Resource with a parent-child relationship (e.g., in oneM2M definition of Resource model). Resources may also be required to have a flat structure and associate with other Resources only by referencing their links.

In addition, different organizations use different syntax and define different features (e.g., Resource OCF Interface), which preclude interoperability.

C.2 OCF approach for support of multiple Resource models

In order to expand the IoT ecosystem the Framework takes an inclusive approach for interworking with existing Resource models. Specifically, the Framework defines a Resource model while providing a mechanism to easily map to other models. By embracing existing Resource models OCF is inclusive of existing ecosystems while allowing for the transition toward definition of a comprehensive Resource model integrating all ecosystems.

The following OCF characteristics enable support of other Resource models:

– *Resource model is the superset of multiple models*: the Resource model is defined as the superset of existing Resource models. In other words, any existing Resource model can be mapped to a subset of Resource model concepts.

– *Framework may allow for Resource model negotiation*: the Client and Server exchange the information about what Resource model(s) each supports. Based on the exchanged information, the Client and Server choose a Resource model to perform RESTful interactions or to perform translation. This feature is out of scope of the current version of this document, however, the following is a high level description for Resource model negotiation.

C.3 Resource model indication

The Client and server exchange the information about what Resource model(s) each supports. Based on the exchanged information, the Client and Server choose a Resource model to perform RESTful interactions or to perform translation. The exchange could be part of discovery and negotiation. Based on the exchange, the Client and Server follow a procedure to ensure interoperability among them. They may choose a common Resource model or execute translation between Resource models.

– *Resource model schema exchange*: The Client and Server may share the Resource model information when they initiate a RESTful interaction. They may exchange the information about which Resource model they support as part of session establishment procedures. Alternatively, each request or response message may carry the indication of which Resource model it is using. For example, [COAP] defines Content-Format option to indicate the representation format such as "application/json". It's possible to extend the Content-Format Option to indicate the Resource model used with the representation format such as "application/ipso-json".

– *Ensuing procedures*: After the Client and Server exchange the Resource model information, they perform a suitable procedure to ensure interoperability among them. The simplest way is to choose a Resource model supported by both the Client and Server. In case there is no common Resource model, the Client and Server may interact through a 3rd party.

In addition to translation which can be resource intensive, a method based on profiles can be used in which an OCF implementation can accommodate multiple profiles and hence multiple ecosystems.

– *Resource Model Profile*: the Framework defines Resource model profiles and implementers or users choose the active profile. The chosen profile constraints the Device to strict rules in how Resources are defined, instantiated and interacted with. This would allow for interoperation with devices from the ecosystem identified by the profile (e.g., OMA SpecWorks, OneM2M etc.). Although this enables a Device to participate in and be part of any given ecosystem, this scheme does not allow for generic interoperability at runtime. While this approach may be suitable for resource constrained devices, more resource capable devices are expected to support more than one profile.

C.4 An Example Profile (OMA SpecWorks profile)

C.4.1 Overview

OMA SpecWorks defines smart objects that have specific Resources and they take values determined by the data type of that Resource. The smart object specification defines a category of such objects. Each Resource represents a characteristic of the smart object being modelled.

While the terms may be different, there are equivalent concepts in OCF to represent these terms. This clause provides the equivalent OCF terms and then frames the OMA SpecWorks smart object in OCF terms.

The OMA SpecWorks object Light Control defined in clause 16 of the OMA SpecWorks Smart Objects 1.0 is used as the reference example.

C.5 Conceptual equivalence

The OMA SpecWorks smart object definition is equivalent to a Resource Type definition which defines the relevant characteristics of an entity being modelled. The specific OMA SpecWorks Resource is equivalent to a Property that like an OMA SpecWorks Resource has a defined data type, enumeration of acceptable values, units, a general description and access modes (based on the OCF Interface).

The general method for developing the equivalent Resource Type from an OMA SpecWorks Smart Object definition is to ignore the Object ID and replace the Object URN with and OCF "." (dot) separated name that incorporates the OMA SpecWorks object. Alternatively the Object URN can be used as the Resource Type ID as is (as long as the URN does not contain any "." (dots)) – using the same Object URN as the Resource Type ID allows for compatibility when interacting with an OMA SpecWorks compliant device. The object URN based naming does not have any bearing for OCF to OCF interoperability and so the OCF format is preferred – for OCF to OCF interoperability only the data model consistency is required.

Two models are available to render OMA SpecWorks objects into OCF.

– One is where the OMA SpecWorks Smart Object represents a Resource. In this case, the IP Smart Object is regarded as a Resource with the Resource Type matching the description of the Smart Object. Furthermore, each Resource in the OMA SpecWorks definition is represented as a Property in the Resource Type (the OMA SpecWorks Resource ID is replaced with a string representing the Property). This is the preferred approach when the OMA SpecWorks Data Model is expressed in the Resource Model.

– The other approach is to model an OMA SpecWorks Smart Object as a Collection. Each OMA SpecWorks Resource is then modelled as a Resource with a Resource Type that matches the definition of the OMA SpecWorks Resource. Each of these Resource instances are then bound to the Collection that represents this OMA SpecWorks Smart Object.

C.5.1 is an example showing how an OMA SpecWorks LightControl Object is modelled as a Resource.

C.5.1 Resource Type: Light Control

Description: This Object is used to control a light source, such as a LED or other light. It allows a light to be turned on or off and its dimmer setting to be controlled as a percentage value between 0 and 100. An optional colour setting enables a string to be used to indicate the desired colour. Table C-1 and Table C-2 define the Resource Type and its Properties, respectively.

Table C-1 – Light control Resource Type definition

Resource Type	Resource Type ID	Multiple Instances	Description
Light Control	"oic.light.control" or "urn:oma:lwm2m:ext:3311"	Yes	Light control object with on/off and optional dimming and energy monitor

Table C-2 – Light control Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
On/Off	"on-off"	"boolean"	N/A	N/A	R, W	Yes	On/Of Control: 0 = Off 1 = On
Dimmer	"dim"	"integer"	N/A	%	R, W	No	Proportional Control, integer value between 0 and 100 as percentage
Color	"color"	"string"	0 – 100	Defined by "units" Property	R, W	No	String representing some value in color space
Units	"units"	"string"	N/A	N/A	R	No	Measurement Units Definition e.g., "Cel"

							for Temperature in Celsius.
On Time	"ontime"	"integer"	N/A	s	R, W	No	The time in seconds that the light has been on. Writing a value of "0" resets the counter
Cumulative active power	"cumap"	"float"	N/A	Wh	R	No	The cumulative active power since the last cumulative energy reset or device start
Power Factor	"powfact"	"float"	N/A	N/A	R	No	The power factor of the load

4584

4585

Annex D
(normative)

Resource Type definitions

D.1 List of Resource Type definitions

All the clauses in Annex D and Annex E describe the Resource Types with a RESTful API definition language. The Resource Type definitions presented in Annex D and Annex E are formatted for readability, and so may appear to have extra line breaks. Table D-1 contains the list of defined Core Common Resources in this document.

Table D-1 – Alphabetized list of Core Resources

Friendly Name (informative)	Resource Type (rt)	Clause
Alerts	"oic.r.alert"	D.17
Alerts Collection	"oic.r.alertcollection"	D.18
Atomic Measurement	"oic.wk.atomicmeasurement"	D.2
Collections	"oic.wk.col"	D.3
Device Configuration	"oic.wk.con"	D.4
Platform Configuration	"oic.wk.con.p"	D.5
Device	"oic.wk.d"	D.6
Discoverable Resource	"oic.wk.res"	D.13
Icon	"oic.r.icon"	D.7
Introspection	"oic.wk.introspection"	D.8
Maintenance	"oic.wk.mnt"	D.9
Network Monitoring	"oic.wk.nmon"	D.10
Platform	"oic.wk.p"	D.11
Resource Directory	"oic.wk.rd"	D.12
Scenes (Top Level)	"oic.wk.scenelist"	D.14
Scenes Collections	"oic.wk.scenecollection"	D.15
Scene Member	"oic.wk.scenemember"	D.16

D.2 Atomic Measurement links list representation

D.2.1 Introduction

The oic.if.baseline OCF Interface exposes a representation of the links and the Common Properties of the Atomic Measurement Resource.

D.2.2 Example URI

/AtomicMeasurementResURI

D.2.3 Resource type

The Resource Type is defined as: "oic.wk.atomicmeasurement".

D.2.4 OpenAPI 2.0 definition

```
{
       "swagger": "2.0",
       "info": {
              "title": "Atomic Measurement links list representation",
              "version": "2019-03-04",
              "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
       },
       "schemes": ["http"],
       "consumes": ["application/json"],
       "produces": ["application/json"],
       "paths": {
              "/AtomicMeasurementResURI?if=oic.if.ll": {
                     "get": {
                            "description": "The oic.if.ll OCF Interface exposes a representation
of the Links",
                            "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
                            "responses": {
                                   "200": {
                                          "description": "",
                                          "x-example": [{
                                                 "href": "/temperature",
                                                 "rt": ["oic.r.temperature"],
                                                 "if": ["oic.if.s", "oic.if.baseline"]
                                          },
                                          {
                                                 "href": "/bodylocation",
                                                 "rt": ["oic.r.body.location.temperature"],
                                                 "if": ["oic.if.s", "oic.if.baseline"]
                                          },
                                          {
                                                 "href": "/timestamp",
                                                 "rt": ["oic.r.time.stamp"],
                                                 "if": ["oic.if.s", "oic.if.baseline"]
                                          }],
                                          "schema": {
                                                 "$ref": "#/definitions/links"
                                          }
                                   }
                            }
                     }
              },
              "/AtomicMeasurementResURI?if=oic.if.b": {
                     "get": {
                            "description": "The oic.if.b OCF Interface returns data items
retrieved from Resources pointed to by the Links.\n",
                            "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
                            "responses": {
                                   "200": {
                                          "description": "Normal response, no errors, all
Properties are returned correctly\n",
                                          "x-example": [{
                                                 "href": "/temperature",
                                                 "rep": {
                                                        "temperature": 38,
                                                        "units": "C",
                                                        "range": [25,  45]
```

```
                                                        }
                                                },
                                                {
                                                        "href": "/bodylocation",
                                                        "rep": {
                                                                "bloc": "ear"
                                                        }
                                                },
                                                {
                                                        "href": "/timestamp",
                                                        "rep": {
                                                                "timestamp": "2007-04-05T14:30+09:00"
                                                        }
                                                }],
                                                "schema": {
                                                        "$ref": "#/definitions/batch-retrieve"
                                                }
                                        }
                                }
                        }
                },
                "/AtomicMeasurementResURI?if=oic.if.baseline": {
                        "get": {
                                "description": "The oic.if.baseline OCF Interface exposes a
representation of the links and\nthe Common Properties of the Atomic Measurement Resource.\n",
                                "parameters": [
           {
             "$ref": "#/parameters/interface-all"
           }
         ],
                                "responses": {
                                        "200": {
                                                "description": "",
                                                "x-example": {
                                                        "rt": ["oic.wk.atomicmeasurement"],
                                                        "if": ["oic.if.b", "oic.if.ll",
        "oic.if.baseline"],
                                                        "rts": ["oic.r.temperature",
"oic.r.body.location.temperature",    "oic.r.time.stamp"],
                                                        "rts-m": ["oic.r.temperature",
"oic.r.body.location.temperature",    "oic.r.time.stamp"],
                                                        "links": [{
                                                                "href": "/temperature",
                                                                "rt": ["oic.r.temperature"],
                                                                "if": ["oic.if.s", "oic.if.baseline"]
                                                        },
                                                        {
                                                                "href": "/bodylocation",
                                                                "rt":
["oic.r.body.location.temperature"],
                                                                "if": ["oic.if.s", "oic.if.baseline"]
                                                        },
                                                        {
                                                                "href": "/timestamp",
                                                                "rt": ["oic.r.time.stamp"],
                                                                "if": ["oic.if.s", "oic.if.baseline"]
                                                        }]
                                                },
                                                "schema": {
                                                        "$ref": "#/definitions/baseline"
                                                }
                                        }
                                }
                        }
                }
        },
        "parameters": {
                "interface-all": {
                        "in": "query",
                        "name": "if",
                        "type": "string",
```



```
                        "enum": ["oic.if.b", "oic.if.ll",     "oic.if.baseline"]
                }
        },
        "definitions": {
                "links": {
                        "type": "array",
                        "items": {
                                "$ref": "#/definitions/oic.oic-link"
                        }
                },
                "batch-retrieve": {
                        "title": "Collection Batch Retrieve Format (auto merged)",
                        "minItems": 1,
                        "items": {
                                "additionalProperties": true,
                                "properties": {
                                        "href": {
                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
                                        },
                                        "rep": {
                                                "oneOf": [{
                                                        "description": "The response payload from a
single Resource",
                                                        "type": "object"
                                                },
                                                {
                                                        "description": " The response payload from a
Collection (batch) Resource",
                                                        "items": {
                                                                "properties": {
                                                                        "anchor": {
                                                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
                                                },
                                                "di": {
                                                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
                                                },
                                                "eps": {
                                                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
                                                },
                                                "href": {
                                                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
                                                },
                                                                "if": {
                                                                        "description": "The OCF
Interface set supported by this Resource",
                                                                        "items": {
                                                                                "enum": [

        "oic.if.baseline",
                                                                                        "oic.if.ll",
                                                                                        "oic.if.b",
                                                                                        "oic.if.rw",
                                                                                        "oic.if.r",
                                                                                        "oic.if.a",
                                                                                        "oic.if.s"],
                                                                                        "type":
"string"
                                                                        },
                                                                        "minItems": 1,
                                                                        "uniqueItems": true,
                                                                        "type": "array"
```



```
                                                                    },
                                                                    "ins": {
                                                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
                                                },
                                                "p": {
                                                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
                                                },
                                                "rel": {
                                                  "description": "The relation of the target URI
referenced by the Link to the context URI",
                                                  "oneOf": [
                                                    {
                                                      "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
                                                    },
                                                    {
                                                      "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_string"
                                                    }
                                                  ]
                                                },
                                                                    "rt": {
                                                                           "description":
"Resource Type of the Resource",
                                                                           "items": {
                                                                                  "maxLength":
64,
                                                                                  "type":
"string"
                                                                           },
                                                                           "minItems": 1,
                                                                           "uniqueItems": true,
                                                                           "type": "array"
                                                                    },
                                                                    "title": {
                                                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
                                                },
                                                "type": {
                                                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
                                                }
                                                                    },
                                                                    "required": [
                                                                      "href",
                                                                      "rt",
                                                                      "if"
                                                                    ],
                                                                    "type": "object"
                                                             },
                                                             "type": "array"
                                                }]
                                         }
                                  },
                                  "required": [
                                    "href",
                                    "rep"
                                  ],
                                  "type": "object"
                           },
                           "type": "array"
                    },
                    "baseline": {
```

```
                        "properties": {
                                "links": {
                                        "description": "A set of simple or individual Links.",
                                        "items": {
                                                "$ref": "#/definitions/oic.oic-link"
                                        },
                                        "type": "array"
                                },
                                "n": { "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"},
                        "id": { "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"},
                                "rt": {
          "description": "Resource Type of this Resource",
          "items": {
            "enum": ["oic.wk.atomicmeasurement"],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "readOnly": true,
          "uniqueItems": true,
          "type": "array"
        },
                                "rts": {
                                        "description": "An array of Resource Types that are supported
within an array of Links exposed by the Resource",
                                        "items": {
                                                "maxLength": 64,
                                                "type": "string"
                                        },
                                        "minItems": 1,
                                        "readOnly": true,
                                        "uniqueItems": true,
                                        "type": "array"
                                },
                                "rts-m": {
                                        "description": "An array of Resource Types that are mandatory
to be exposed within an array of Links exposed by the Resource",
                                        "items": {
                                                "maxLength": 64,
                                                "type": "string"
                                        },
                                        "minItems": 1,
                                        "readOnly": true,
                                        "uniqueItems": true,
                                        "type": "array"
                                },
                                "if": {
                                        "description": "The OCF Interface set supported by this
Resource",
                                        "items": {
                                                "enum": ["oic.if.b", "oic.if.ll", "oic.if.baseline"],
                                                "type": "string"
                                        },
                                        "minItems": 3,
          "readOnly": true,
          "uniqueItems": true,
          "type": "array"
                                }
                        },
                        "type": "object",
                        "required": [
        "rt",
        "if",
        "links"
      ]
                },
                "oic.oic-link": {
```



```
                        "properties": {
                                "anchor": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
        },
        "di": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
        },
        "eps": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
        },
        "href": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
        },
                                "if": {
                                        "description": "The OCF Interface set supported by this
Resource",
                                        "items": {
                                                "enum": [
                                                "oic.if.baseline",
                                                "oic.if.ll",
                                                "oic.if.b",
                                                "oic.if.rw",
                                                "oic.if.r",
                                                "oic.if.a",
                                                "oic.if.s"],
                                                "type": "string"
                                        },
                                        "minItems": 1,
                                        "uniqueItems": true,
                                        "type": "array"
                                },
                                "ins": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
        },
        "p": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
        },
        "rel": {
          "description": "The relation of the target URI referenced by the Link to the context URI",
          "oneOf": [
            {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
            },
            {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_string"
            }
          ]
        },
                                "rt": {
                                        "description": "Resource Type of the Resource",
                                        "items": {
                                                "maxLength": 64,
                                                "type": "string"
                                        },
                                        "minItems": 1,
```

```
                                        "uniqueItems": true,
                                        "type": "array"
                                },
                                "title": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
          },
          "type": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
          }
                        },
                        "required": [
                                "href",
                          "rt",
                          "if"
                        ],
                        "type": "object"
                }
        }
}
```

D.2.5 Property definition

Table D-2 defines the Properties that are part of the "oic.wk.atomicmeasurement" Resource Type.

Table D-2 – The Property definitions of the Resource with type "rt" = "oic.wk.atomicmeasurement".

Property name	Value type	Mandatory	Access mode	Description
href	multiple types: see schema	Yes	Read Write	
rep	multiple types: see schema	Yes	Read Write	
links	array: see schema	Yes	Read Write	A set of simple or individual Links.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
rt	array: see schema	Yes	Read Only	Resource Type of this Resource.
rts	array: see schema	No	Read Only	An array of Resource Types that are supported within an array of Links exposed by the Resource.
rts-m	array: see schema	No	Read Only	An array of Resource Types that are mandatory to be exposed within an array of Links exposed by the Resource.
if	array: see schema	Yes	Read Only	The OCF Interface set supported by this Resource.

anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Write	The OCF Interface set supported by this Resource.
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the Link to the context URI.
rt	array: see schema	Yes	Read Write	Resource Type of the Resource.
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

D.2.6 CRUDN behaviour

Table D-3 defines the CRUDN operations that are supported on the "oic.wk.atomicmeasurement" Resource Type.

Table D-3 – The CRUDN operations of the Resource with type "rt" = "oic.wk.atomicmeasurement".

Create	Read	Update	Delete	Notify
	get			observe

D.3 Collection

D.3.1 Introduction

Collection Resource Type contains Properties and Links. The oic.if.baseline OCF Interface exposes a representation of the Links and the Properties of the Collection Resource itself

D.3.2 Example URI

/CollectionResURI

D.3.3 Resource type

The Resource Type is defined as: "oic.wk.col".

D.3.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
```



```
  "info": {
    "title": "Collection",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/CollectionResURI?if=oic.if.ll" : {
      "get": {
        "description": "Collection Resource Type contains Properties and Links.\nThe oic.if.ll OCF
Interface exposes a representation of the Links\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": [
              {
                "href": "/switch",
                "rt":   ["oic.r.switch.binary"],
                "if":   ["oic.if.a", "oic.if.baseline"],
                "eps": [
                  {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                  {"ep": "coaps://[fe80::b1d6]:1122"},
                  {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                ]
              },
              {
                "href": "/airFlow",
                "rt":   ["oic.r.airflow"],
                "if":   ["oic.if.a", "oic.if.baseline"],
                "eps": [
                  {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                  {"ep": "coaps://[fe80::b1d6]:1122"},
                  {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                ]
              }
            ],
            "schema": {
              "$ref": "#/definitions/slinks"
            }
          }
        }
      }
    },
    "/CollectionResURI?if=oic.if.baseline" : {
      "get": {
        "description": "Collection Resource Type contains Properties and Links.\nThe oic.if.baseline
OCF Interface exposes a representation of\nthe Links and the Properties of the Collection Resource
itself\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
```

```
          "responses": {
            "200": {
              "description" : "",
              "x-example": {
                "rt": ["oic.wk.col"],
                "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
                "rts": [ "oic.r.switch.binary", "oic.r.airflow" ],
                "rts-m": [ "oic.r.switch.binary" ],
                "links": [
                  {
                    "href": "/switch",
                    "rt":   ["oic.r.switch.binary"],
                    "if":   ["oic.if.a", "oic.if.baseline"],
                    "eps": [
                        {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                        {"ep": "coaps://[fe80::b1d6]:1122"},
                        {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
                    ]
                  },
                  {
                    "href": "/airFlow",
                    "rt":   ["oic.r.airflow"],
                    "if":   ["oic.if.a", "oic.if.baseline"],
                    "eps": [
                        {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                        {"ep": "coaps://[fe80::b1d6]:1122"},
                        {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
                    ]
                  }
                ]
              },
              "schema": {
                "$ref": "#/definitions/sbaseline"
              }
            }
          }
        },
        "post": {
          "description": "Update on Baseline OCF Interface\n",
          "parameters": [
            {
              "$ref": "#/parameters/interface-update"
            },
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": {
                "$ref": "#/definitions/sbaseline-update"
              }
            }
          ],
          "responses": {
            "200": {
              "description" : "",
              "schema": {
                "$ref": "#/definitions/sbaseline"
              }
            }
          }
        }
      },
      "/CollectionResURI?if=oic.if.b" : {
        "get": {
          "description": "Collection Resource Type contains Properties and Links.\nThe oic.if.b OCF
Interfacce exposes a composite representation of the\nResources pointed to by the Links\n",
          "parameters": [
            {
              "$ref": "#/parameters/interface-all"
            }
          ],
```



```
          "responses": {
            "200": {
              "description" : "All targets returned OK status",
              "x-example": [
                {
                  "href": "/switch",
                  "rep": {
                    "value": true
                  }
                },
                {
                  "href": "/airFlow",
                  "rep": {
                    "direction": "floor",
                    "speed":     3
                  }
                }
              ],
              "schema": {
                "$ref": "#/definitions/sbatch-retrieve"
              }
            },
            "404": {
              "description" : "One or more targets did not return an OK status, return a
representation containing returned Properties from the targets that returned OK",
              "x-example": [
                {
                  "href": "/switch",
                  "rep": {
                    "value": true
                  }
                }
              ],
              "schema": {
                "$ref": "#/definitions/sbatch-retrieve"
              }
            }
          }
        },
        "post": {
          "description": "Update on Batch OCF Interface\n",
          "parameters": [
            {
              "$ref": "#/parameters/interface-update"
            },
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": {
                "$ref": "#/definitions/sbatch-update"
              },
              "x-example": [
                {
                  "href": "/switch",
                  "rep": {
                    "value": true
                  }
                },
                {
                  "href": "/airFlow",
                  "rep": {
                    "direction": "floor",
                    "speed": 3
                  }
                }
              ]
            }
          ],
          "responses": {
            "200": {
```

```
              "description" : "All targets returned OK status, return a representation of the current
    state of all targets",
              "x-example": [
                {
                  "href": "/switch",
                  "rep": {
                    "value": true
                  }
                },
                {
                  "href": "/airFlow",
                  "rep": {
                    "direction": "demist",
                    "speed": 5
                  }
                }
              ],
              "schema": {
                "$ref": "#/definitions/sbatch-retrieve"
              }
            },
            "403": {
              "description" : "One or more targets did not return OK status; return a retrieve
    representation of the current state of all targets in the batch",
              "x-example": [
                {
                  "href": "/switch",
                  "rep": {
                    "value": true
                  }
                },
                {
                  "href": "/airFlow",
                  "rep": {
                    "direction": "floor",
                    "speed": 3
                  }
                }
              ],
              "schema": {
                "$ref": "#/definitions/sbatch-retrieve"
              }
            }
          }
        }
      },
      "parameters": {
        "interface-all" : {
          "in" : "query",
          "name" : "if",
          "type" : "string",
          "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
        },
        "interface-update" : {
          "in" : "query",
          "name" : "if",
          "type" : "string",
          "enum" : ["oic.if.b", "oic.if.baseline"]
        }
      },
      "definitions": {
        "sbaseline" : {
          "properties": {
            "links" : {
              "description": "A set of simple or individual Links.",
              "items": {
                "$ref": "#/definitions/oic.oic-link"
              },
              "type": "array"
            },
```



```
          "n": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "id": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
                "rt": {
            "$ref": "#/definitions/oic.core.rt-col"
          },
                "rts": {
            "$ref": "#/definitions/oic.core.rt"
          },
                "rts-m": {
            "$ref": "#/definitions/oic.core.rt"
          },
                "if": {
                  "description": "The OCF Interfaces supported by this Resource",
                  "items": {
                         "enum": [
                "oic.if.ll",
                "oic.if.baseline",
                                "oic.if.b"
              ],
                                                "type": "string",
              "maxLength": 64
                                        },
                                        "minItems": 2,
            "uniqueItems": true,
            "readOnly": true,
                                        "type": "array"
                                  }
        },
        "additionalProperties": true,
        "type" : "object",
        "required": [
          "rt",
          "if",
          "links"
        ]
      },
      "sbaseline-update": {
        "additionalProperties": true
      },
            "oic.core.rt-col": {
        "description": "Resource Type of the Resource",
                        "items": {
                                "enum": ["oic.wk.col"],
                                "type": "string",
          "maxLength": 64
                        },
                        "minItems": 1,
        "uniqueItems": true,
                        "readOnly": true,
                        "type": "array"
                },
      "oic.core.rt": {
        "description": "Resource Type or set of Resource Types",
                        "items": {
                                "type": "string",
          "maxLength": 64
                        },
                        "minItems": 1,
        "uniqueItems": true,
                        "readOnly": true,
                        "type": "array"
                },
      "sbatch-retrieve" : {
```



```
        "minItems" : 1,
        "items" : {
          "additionalProperties": true,
          "properties": {
            "href": {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
            },
            "rep": {
              "oneOf": [
                {
                  "description": "The response payload from a single Resource",
                  "type": "object"
                },
                {
                  "description": " The response payload from a Collection (batch) Resource",
                  "items": {
                    "$ref": "#/definitions/oic.oic-link"
                  },
                  "type": "array"
                }
              ]
            }
          },
          "required": [
            "href",
            "rep"
          ],
          "type": "object"
        },
        "type" : "array"
      },
      "sbatch-update" : {
        "title" : "Collection Batch Update Format",
        "minItems" : 1,
        "items" :  {
          "$ref": "#/definitions/sbatch-update.item"
        },
        "type" :  "array"
      },
      "sbatch-update.item" : {
        "additionalProperties": true,
        "description": "Array of Resource representations to apply to the batch Collection, using href
to indicate which Resource(s) in the batch to update. If the href Property is empty, effectively
making the URI reference to the Collection itself, the representation is to be applied to all
Resources in the batch",
        "properties": {
          "href": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
          },
          "rep": {
            "oneOf": [
              {
                "description": "The payload for a single Resource",
                "type": "object"
              },
              {
                "description": " The payload for a Collection (batch) Resource",
                "items": {
                  "$ref": "#/definitions/oic.oic-link"
                },
                "type": "array"
              }
            ]
          }
        },
        "required": [
          "href",
```

```
          "rep"
        ],
        "type": "object"
      },
      "slinks" : {
        "type" : "array",
        "items" : {
          "$ref": "#/definitions/oic.oic-link"
        }
      },
      "oic.oic-link": {
        "properties": {
          "if": {
            "description": "The OCF Interfaces supported by the Linked target",
            "items": {
              "enum": [
                "oic.if.baseline",
                "oic.if.ll",
                "oic.if.b",
                "oic.if.rw",
                "oic.if.r",
                "oic.if.a",
                "oic.if.s"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "rt": {
            "$ref": "#/definitions/oic.core.rt"
          },
          "anchor": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
          },
          "di": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
          },
          "eps": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
          },
          "href": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
          },
          "ins": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
          },
          "p": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
          },
          "rel": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
          },
          "title": {
```



```
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
        },
        "type": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
        }
      },
      "required": [
        "href",
        "rt",
        "if"
      ],
      "type": "object"
    }
  }
}
```

D.3.5 Property definition

Table D-4 defines the Properties that are part of the "oic.wk.col" Resource Type.

Table D-4 – The Property definitions of the Resource with type "rt" = "oic.wk.col".

Property name	Value type	Mandatory	Access mode	Description
links	array: see schema	Yes	Read Write	A set of simple or individual Links.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
rt	multiple types: see schema	Yes	Read Write	
rts	multiple types: see schema	No	Read Write	
rts-m	multiple types: see schema	No	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
href	multiple types: see schema	Yes	Read Write	
rep	multiple types: see schema	Yes	Read Write	
href	multiple types: see schema	Yes	Read Write	
rep	multiple types: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by the Linked target.
rt	multiple types: see schema	Yes	Read Write	
anchor	multiple types: see schema	No	Read Write	

di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

D.3.6 CRUDN behaviour

Table D-5 defines the CRUDN operations that are supported on the "oic.wk.col" Resource Type.

Table D-5 – The CRUDN operations of the Resource with type "rt" = "oic.wk.col".

Create	Read	Update	Delete	Notify
	get	post		observe

D.4 Device Configuration

D.4.1 Introduction

Resource that allows for Device specific information to be configured.

D.4.2 Example URI

/exampleDeviceConfigurationResURI

D.4.3 Resource type

The Resource Type is defined as: "oic.wk.con".

D.4.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Device Configuration",
    "version": "2019-02-28",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
```

```
      "application/json"
    ],
    "paths": {
      "/exampleDeviceConfigurationResURI" : {
        "get": {
          "description": "Resource that allows for Device specific information to be configured.\n",
          "parameters": [
            {
              "$ref": "#/parameters/interface-all"
            }
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example": {
                "n":    "My Friendly Device Name",
                "rt":   ["oic.wk.con"],
                "loc":  [32.777,-96.797],
                "locn": "My Location Name",
                "c":    "USD",
                "r":    "MyRegion",
                "dl":   "en"
              },
              "schema": {
                "$ref": "#/definitions/Configuration"
              }
            }
          }
        },
        "post": {
          "description": "Update the information about the Device\n",
          "parameters": [
            {
              "$ref": "#/parameters/interface-rw"
            },
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": {
                "$ref": "#/definitions/Update"
              },
              "x-example": {
                "n": "Nuevo Nombre Amistoso",
                "r": "MyNewRegion",
                "ln": [ { "language": "es", "value": "Nuevo Nombre Amistoso" } ],
                "dl": "es"
              }
            }
          ],
          "responses": {
            "200": {
              "description": "",
              "x-example":  {
                "n": "Nuevo Nombre Amistoso",
                "r": "MyNewRegion",
                "ln": [ { "language": "es", "value": "Nuevo Nombre Amistoso" } ],
                "dl":   "es"
              },
              "schema": {
                "$ref": "#/definitions/Update"
              }
            }
          }
        }
      }
    },
    "parameters": {
      "interface-rw" : {
        "in" : "query",
        "name" : "if",
```



```
        "type" : "string",
        "enum" : ["oic.if.rw"]
      },
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.rw", "oic.if.baseline"]
      }
    },
    "definitions": {
      "Configuration": {
        "properties": {
          "rt":  {
            "description": "Resource Type of the Resource",
            "items": {
              "enum": ["oic.wk.con"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "loc": {
            "description": "Location information (lat, long)",
            "items": {
              "type": "number"
            },
            "maxItems": 2,
            "minItems": 2,
            "type": "array"
          },
          "c": {
            "description": "Currency",
            "maxLength": 64,
            "type": "string"
          },
          "ln": {
            "description": "Localized names",
            "items": {
              "properties": {
                "language": {
                  "allOf": [
                    {
                      "description": "Format pattern according to IETF RFC 5646 (language tag).",
                      "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
                      "type": "string"
                    },
                    {
                      "description": "An RFC 5646 language tag."
                    }
                  ]
                },
                "value": {
                  "description": "The Device name in the indicated language.",
                  "maxLength": 64,
                  "type": "string"
                }
              },
              "type": "object"
            },
            "minItems": 1,
            "type": "array"
          },
          "locn": {
            "description": "Human Friendly Name for location",
            "maxLength": 64,
            "type": "string"
          },
```



```
          "dl": {
            "allOf": [
              {
                "description": "Format pattern according to IETF RFC 5646 (language tag).",
                "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
                "type": "string"
              },
              {
                "description": "Default Language as an RFC 5646 language tag."
              }
            ]
          },
          "n": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "id": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "r": {
            "description": "Region",
            "maxLength": 64,
            "type": "string"
          },
          "if" : {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.baseline",
                "oic.if.rw"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["n"]
      },
      "Update" : {
        "properties": {
          "loc": {
            "description": "Location information (lat, long)",
            "items": {
              "type": "number"
            },
            "maxItems": 2,
            "minItems": 2,
            "type": "array"
          },
          "c": {
            "description": "Currency",
            "maxLength": 64,
            "type": "string"
          },
          "ln": {
            "description": "Localized names",
            "items": {
              "properties": {
                "language": {
                  "allOf": [
                    {
                      "description": "Format pattern according to IETF RFC 5646 (language tag).",
                      "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
```

```
                    "type": "string"
                  },
                  {
                    "description": "An RFC 5646 language tag."
                  }
                ]
              },
              "value": {
                "description": "The Device name in the indicated language.",
                "maxLength": 64,
                "type": "string"
              }
            },
            "type": "object"
          },
          "minItems": 1,
          "type": "array"
        },
        "locn": {
          "description": "Human Friendly Name for location",
          "maxLength": 64,
          "type": "string"
        },
        "dl": {
          "allOf": [
            {
              "description": "Format pattern according to IETF RFC 5646 (language tag).",
              "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
              "type": "string"
            },
            {
              "description": "Default Language as an RFC 5646 language tag."
            }
          ]
        },
        "n": {
          "description": "The human friendly name to be set on the Resource, this is also reflected
in the same Property in oic.wk.d",
          "maxLength": 64,
          "type": "string"
        },
        "r": {
          "description": "Region",
          "maxLength": 64,
          "type": "string"
        }
      },
      "anyOf": [
        {
          "required": ["loc"]
        },
        {
          "required": ["locn"]
        },
        {
          "required": ["c"]
        },
        {
          "required": ["r"]
        },
        {
          "required": ["ln"]
        },
        {
          "required": ["dl"]
        },
        {
          "required": ["n"]
        }
      ],
      "type" : "object"
```



```
      }
    }
}
```

D.4.5 Property definition

Table D-6 defines the Properties that are part of the "oic.wk.con" Resource Type.

Table D-6 – The Property definitions of the Resource with type "rt" = "oic.wk.con".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
loc	array: see schema	No	Read Write	Location information (lat, long).
c	string	No	Read Write	Currency.
ln	array: see schema	No	Read Write	Localized names.
locn	string	No	Read Write	Human Friendly Name for location.
dl	multiple types: see schema	No	Read Write	
n	multiple types: see schema	Yes	Read Write	
id	multiple types: see schema	No	Read Write	
r	string	No	Read Write	Region.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
loc	array: see schema	No	Read Write	Location information (lat, long).
c	string	No	Read Write	Currency.
ln	array: see schema	No	Read Write	Localized names.
locn	string	No	Read Write	Human Friendly Name for location.
dl	multiple types: see schema	No	Read Write	
n	string	Yes	Read Write	The human friendly name to be set on the Resource, this is also reflected in the same Property in oic.wk.d.
r	string	No	Read Write	Region.

D.4.6 CRUDN behaviour

Table D-7 defines the CRUDN operations that are supported on the "oic.wk.con" Resource Type.

Table D-7 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con".

Create	Read	Update	Delete	Notify
	get	post		observe

D.5 Platform Configuration

D.5.1 Introduction

Resource that allows for Platform specific information to be configured.

D.5.2 Example URI

/examplePlatformConfigurationResURI

D.5.3 Resource type

The Resource Type is defined as: "oic.wk.con.p".

D.5.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Platform Configuration",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/examplePlatformConfigurationResURI": {
      "get": {
        "description": "Resource that allows for Platform specific information to be configured.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt":   ["oic.wk.con.p"],
              "mnpn": [ { "language": "en", "value": "My Friendly Device Name" } ]
            },
            "schema": { "$ref": "#/definitions/Conf_Platform" }
          }
        }
      },
      "post": {
        "description": "Update the information about the Platform\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-rw"
          },
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/Update_Platform" },
            "x-example": {
```

```
                "n": "Nuevo nombre",
                "mnpn": [ { "language": "es", "value": "Nuevo nombre de Plataforma Amigable" } ]
              }
            }
          ],
          "responses": {
            "200": {
              "description": "",
              "x-example": {
                "n": "Nuevo nombre",
                "mnpn": [ { "language": "es", "value": "Nuevo nombre de Plataforma Amigable" } ]
              },
              "schema": { "$ref": "#/definitions/Update_Platform" }
            }
          }
        }
      }
    },
    "parameters": {
      "interface-rw": {
        "in": "query",
        "name": "if",
        "type": "string",
        "enum" : ["oic.if.rw"]
      },
      "interface-all": {
        "in": "query",
        "name": "if",
        "type": "string",
        "enum": ["oic.if.rw", "oic.if.baseline"]
      }
    },
    "definitions": {
      "Conf_Platform": {
        "properties": {
          "rt":  {
            "description": "Resource Type of the Resource",
            "items": {
              "enum": ["oic.wk.con.p"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "n":  {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "mnpn":  {
            "description": "Platform names",
            "items": {
              "properties": {
                "language": {
                  "allOf": [
                    {
                      "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/language-tag"
                    },
                    {
                      "description": "An RFC 5646 language tag."
                    }
                  ]
                },
                "value": {
                  "description": "The Platform description in the indicated language.",
                  "maxLength": 64,
                  "type": "string"
```



```
              }
            },
            "type": "object"
          },
          "minItems": 1,
          "type": "array"
        },
        "id": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
        },
        "if": {
          "description": "The OCF Interfaces supported by this Resource",
          "items": {
            "enum": [
              "oic.if.rw",
              "oic.if.baseline"
            ],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "readOnly": true,
          "uniqueItems": true,
          "type": "array"
        }
      },
      "type" : "object"
    },
    "Update_Platform": {
      "properties": {
        "n":  {
          "description": "The human friendly name to be set on the Resource, this is also reflected
in the same Property in oic.wk.p",
          "maxLength": 64,
          "type": "string"
        },
        "mnpn" :  {
          "description": "Platform names",
          "items": {
            "properties": {
              "language": {
                "allOf": [
                  {
                    "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/language-tag"
                  },
                  {
                    "description": "An RFC 5646 language tag."
                  }
                ]
              },
              "value": {
                "description": "The Platform description in the indicated language.",
                "maxLength": 64,
                "type": "string"
              }
            },
            "type": "object"
          },
          "minItems": 1,
          "type": "array"
        }
      },
      "type": "object",
      "anyOf": [
        {
          "required": ["mnpn"]
        },
        {
```

```
            "required": ["n"]
          }
        ]
      }
    }
}
```

D.5.5 Property definition

Table D-8 defines the Properties that are part of the "oic.wk.con.p" Resource Type.

Table D-8 – The Property definitions of the Resource with type "rt" = "oic.wk.con.p".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema		Read Only	Resource Type of the Resource.
n	multiple types: see schema		Read Write	
mnpn	array: see schema		Read Write	Platform names.
id	multiple types: see schema		Read Write	
if	array: see schema		Read Only	The OCF Interfaces supported by this Resource.
n	string	Yes	Read Write	The human friendly name to be set on the Resource, this is also reflected in the same Property in oic.wk.p.
mnpn	array: see schema	No	Read Write	Platform names.

D.5.6 CRUDN behaviour

Table D-9 defines the CRUDN operations that are supported on the "oic.wk.con.p" Resource Type.

Table D-9 – The CRUDN operations of the Resource with type "rt" = "oic.wk.con.p".

Create	Read	Update	Delete	Notify
	get	post		observe

D.6 Device

D.6.1 Introduction

Known Resource that is hosted by every Server. Allows for logical Device specific information to be discovered.

D.6.2 Well-known URI

/oic/d

D.6.3 Resource type

The Resource Type is defined as: "oic.wk.d".

D.6.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Device",
    "version": "2019-03-13",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/oic/d" : {
      "get": {
        "description": "Known Resource that is hosted by every Server.\nAllows for logical Device
specific information to be discovered.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface"
          }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example":
              {
                "n":    "Device 1",
                "rt":   ["oic.wk.d"],
                "di":   "54919CA5-4101-4AE4-595B-353C51AA983C",
                "icv":  "ocf.2.0.2",
                "dmv":  "ocf.res.1.0.0, ocf.sh.1.0.0",
                "piid": "6F0AAC04-2BB0-468D-B57C-16570A26AE48"
              },
              "schema": {
                 "$ref": "#/definitions/Device"
               }
            }
         }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in": "query",
      "name": "if",
      "type": "string",
      "enum": ["oic.if.r", "oic.if.baseline"]
    }
  },
  "definitions": {
    "Device": {
      "properties": {
        "rt":  {
          "description": "Resource Type of the Resource",
          "items": {
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
```

```
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          },
          "ld":  {
            "description": "Localized Descriptions.",
            "items": {
              "properties": {
                "language": {
                  "allOf": [
                    {
                      "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/language-tag"
                    },
                    {
                      "description": "An RFC 5646 language tag.",
                      "readOnly": true
                    }
                  ]
                },
                "value": {
                  "description": "Device description in the indicated language.",
                  "maxLength": 64,
                  "readOnly": true,
                  "type": "string"
                }
              },
              "type": "object"
            },
            "minItems": 1,
            "readOnly": true,
            "type": "array"
          },
          "piid": {
            "allOf": [
              {
                "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/uuid"
              },
              {
                "description": "Protocol independent unique identifier for the Device that is
immutable.",
                "readOnly": true
              }
            ]
          },
          "di": {
            "allOf": [
              {
                "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/uuid"
              },
              {
                "description": "Unique identifier for the Device",
                "readOnly": true
              }
            ]
          },
          "dmno":  {
            "description": "Model number as designated by manufacturer.",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "sv":  {
            "description": "Software version.",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "dmn":  {
```

```
            "description": "Manufacturer Name.",
            "items": {
              "properties": {
                "language": {
                  "allOf": [
                    {
                      "$ref" : "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/language-tag"
                    },
                    {
                      "description": "An RFC 5646 language tag.",
                      "readOnly": true
                    }
                  ]
                },
                "value": {
                  "description": "Manufacturer name in the indicated language.",
                  "maxLength": 64,
                  "readOnly": true,
                  "type": "string"
                }
              },
              "type": "object"
            },
            "minItems": 1,
            "readOnly": true,
            "type": "array"
          },
          "icv": {
            "description": "The version of the Device",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "dmv":  {
            "description": "Specification versions of the Resource and Device Specifications to which
this device data model is implemented",
            "maxLength": 256,
            "readOnly": true,
            "type": "string"
          },
          "n":  {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "id": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "if":  {
            "description": "The OCF Interfacces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.r",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 2,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          }
        },
        "type": "object",
        "required": ["n", "di", "icv", "dmv", "piid"]
      }
    }
```


}

D.6.5 Property definition

Table D-10 defines the Properties that are part of the "oic.wk.d" Resource Type.

Table D-10 – The Property definitions of the Resource with type "rt" = "oic.wk.d".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
ld	array: see schema	No	Read Only	Localized Descriptions.
piid	multiple types: see schema	Yes	Read Write	
di	multiple types: see schema	Yes	Read Write	
dmno	string	No	Read Only	Model number as designated by manufacturer.
sv	string	No	Read Only	Software version.
dmn	array: see schema	No	Read Only	Manufacturer Name.
icv	string	Yes	Read Only	The version of the Device
dmv	string	Yes	Read Only	Specification versions of the Resource and Device Specifications to which this device data model is implemented.
n	multiple types: see schema	Yes	Read Write	
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfacces supported by this Resource.

D.6.6 CRUDN behaviour

Table D-11 defines the CRUDN operations that are supported on the "oic.wk.d" Resource Type.

Table D-11 – The CRUDN operations of the Resource with type "rt" = "oic.wk.d".

Create	Read	Update	Delete	Notify
	get			observe

D.7 Icon

D.7.1 Introduction

This Resource describes the attributes associated with an Icon.

D.7.2 Example URI

/IconResURI

D.7.3 Resource type

The Resource Type is defined as: "oic.r.icon".

D.7.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Icon",
    "version": "2019-02-26",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/IconResURI" : {
      "get": {
        "description": "This Resource describes the attributes associated with an Icon.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface"
          }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": {
              "rt": ["oic.r.icon"],
              "mimetype": "image/png",
              "width": 256,
              "height": 256,
              "media": "http://findbetter.ru/public/uploads/1481662800/2043.png"
            },
            "schema": {
              "$ref": "#/definitions/Icon"
            }
          }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.r", "oic.if.baseline"]
    }
  },
  "definitions": {
    "Icon" : {
      "properties": {
        "mimetype": {
          "description": "The Media Type of the icon",
```

```
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "rt": {
            "description": "Resource Type of the Resource",
            "items": {
              "enum": ["oic.r.icon"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "media": {
            "description": "Specifies the URI to the icon",
            "format": "uri",
            "maxLength": 256,
            "readOnly": true,
            "type": "string"
          },
          "n": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "id": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "width": {
            "description": "The width in pixels",
            "minimum": 1,
            "readOnly": true,
            "type": "integer"
          },
          "height": {
            "description": "The height in pixels",
            "minimum": 1,
            "readOnly": true,
            "type": "integer"
          },
          "if": {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.r",
                "oic.if.baseline"
              ],
              "maxLength": 64,
              "type": "string"
            },
            "minItems": 2,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["mimetype", "width", "height", "media"]
      }
    }
  }
```

D.7.5 Property definition

Table D-12 defines the Properties that are part of the "oic.r.icon" Resource Type.

Table D-12 – The Property definitions of the Resource with type "rt" = "oic.r.icon".

Property name	Value type	Mandatory	Access mode	Description
mimetype	string	Yes	Read Only	The Media Type of the icon.
rt	array: see schema	No	Read Only	Resource Type of the Resource.
media	string	Yes	Read Only	Specifies the URI to the icon.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
width	integer	Yes	Read Only	The width in pixels.
height	integer	Yes	Read Only	The height in pixels.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

D.7.6 CRUDN behaviour

Table D-13 defines the CRUDN operations that are supported on the "oic.r.icon" Resource Type.

Table D-13 – The CRUDN operations of the Resource with type "rt" = "oic.r.icon".

Create	Read	Update	Delete	Notify
	get			observe

D.8 Introspection Resource

D.8.1 Introduction

This Resource provides the means to get the Introspection Device Data (IDD) specifying all the OCF Endpoints of the Device.
The url hosted by this Resource is either a local or an external url.

D.8.2 Well-known URI

/IntrospectionResURI

D.8.3 Resource type

The Resource Type is defined as: "oic.wk.introspection".

D.8.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Introspection Resource",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
```

```
      "http"
    ],
    "consumes": [
      "application/json"
    ],
    "produces": [
      "application/json"
    ],
    "paths": {
      "/IntrospectionResURI": {
        "get": {
          "description": "This Resource provides the means to get the Introspection Device Data (IDD)
specifying all the OCF Endpoints of the Device.\nThe url hosted by this Resource is either a local
or an external url.\n",
          "parameters": [
            {
              "$ref": "#/parameters/interface"
            }
          ],
          "responses": {
            "200": {
              "description": "",
              "x-example": {
                "rt": ["oic.wk.introspection"],
                "urlInfo": [
                  {
                    "content-type": "application/cbor",
                    "protocol": "coap",
                    "url": "coap://[fe80::1]:1234/IntrospectionExampleURI"
                  }
                ]
              },
              "schema": {
                "$ref": "#/definitions/oic.wk.introspectionInfo"
              }
            }
          }
        }
      }
    },
    "parameters": {
      "interface": {
        "in": "query",
        "name": "if",
        "type": "string",
        "enum": ["oic.if.r", "oic.if.baseline"]
      }
    },
    "definitions": {
      "oic.wk.introspectionInfo": {
        "properties": {
          "rt":  {
            "description": "Resource Type of the Resource",
            "items": {
                "enum": ["oic.wk.introspection"],
                "type": "string",
                "maxLength": 64
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          },
          "n":  {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "urlInfo":  {
            "description": "Information on the location of the Introspection Device Data (IDD).",
            "items": {
```



```
              "properties": {
                "content-type": {
                  "default": "application/cbor",
                  "description": "content-type of the Introspection Device Data",
                  "enum": [
                    "application/json",
                    "application/cbor"
                  ],
                  "type": "string"
                },
                "protocol": {
                  "description": "Identifier for the protocol to be used to obtain the Introspection
Device Data",
                  "enum": [
                    "coap",
                    "coaps",
                    "http",
                    "https",
                    "coap+tcp",
                    "coaps+tcp"
                  ],
                  "type": "string"
                },
                "url": {
                  "description": "The URL of the Introspection Device Data.",
                  "format": "uri",
                  "type": "string"
                },
                "version": {
                  "default": 1,
                  "description": "The version of the Introspection Device Data that can be
downloaded",
                  "enum": [
                    1
                  ],
                  "type": "integer"
                }
              },
              "required": [
                "url",
                "protocol"
              ],
              "type": "object"
            },
            "minItems": 1,
            "readOnly": true,
            "type": "array"
          },
          "id":  {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "if": {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.r",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 2,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["urlInfo"]
```



```
      }
    }
}
```

D.8.5 Property definition

Table D-14 defines the Properties that are part of the "oic.wk.introspection" Resource Type.

Table D-14 – The Property definitions of the Resource with type "rt" = "oic.wk.introspection".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
n	multiple types: see schema	No	Read Write	
urlInfo	array: see schema	Yes	Read Only	Information on the location of the Introspection Device Data (IDD).
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

D.8.6 CRUDN behaviour

Table D-15 defines the CRUDN operations that are supported on the "oic.wk.introspection" Resource Type.

Table D-15 – The CRUDN operations of the Resource with type "rt" = "oic.wk.introspection".

Create	Read	Update	Delete	Notify
	get			observe

D.9 Maintenance

D.9.1 Introduction

The Resource through which a Device is maintained and can be used for diagnostic purposes.
fr (Factory Reset) is a boolean.
The value 0 means No action (Default), the value 1 means Start Factory Reset
After factory reset, this value shall be changed back to the default value
rb (Reboot) is a boolean.
The value 0 means No action (Default), the value 1 means Start Reboot
After Reboot, this value shall be changed back to the default value

D.9.2 Well-known URI

/oic/mnt

D.9.3 Resource type

The Resource Type is defined as: "oic.wk.mnt".

D.9.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Maintenance",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/oic/mnt" : {
      "get": {
        "description": "The Resource through which a Device is maintained and can be used for
diagnostic purposes.\nfr (Factory Reset) is a boolean.\n  The value 0 means No action (Default), the
value 1 means Start Factory Reset\nAfter factory reset, this value shall be changed back to the
default value\nrb (Reboot) is a boolean.\n  The value 0 means No action (Default), the value 1 means
Start Reboot\nAfter Reboot, this value shall be changed back to the default value\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example": {
                "rt":   ["oic.wk.mnt"],
                "fr":   false,
                "rb":   false,
                "err" : 503
              },
              "schema": { "$ref": "#/definitions/mnt" }
            }
        }
      },
      "post": {
        "description": "Set the maintenance action(s)\n",
        "parameters": [
          {"$ref": "#/parameters/interface-rw"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/mnt-update" },
            "x-example": {
              "fr":   false,
              "rb":   false
            }
          }
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example": {
                "fr":   false,
                "rb":   false
              },
              "schema": { "$ref": "#/definitions/mnt" }
            }
        }
      }
    }
  },
```

```
    "parameters": {
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.rw", "oic.if.baseline"]
      },
      "interface-rw" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.rw"]
      }
    },
    "definitions": {
      "mnt" : {
        "properties": {
          "rt" : {
            "description": "Resource Type of the Resource",
            "items": {
              "enum":   ["oic.wk.mnt"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "fr" : {
            "description": "Factory Reset",
            "type": "boolean"
          },
          "err" : {
            "description": "Last HTTP occurred error",
            "maximum": 599,
            "minimum": 399,
            "readOnly": true,
            "type": "integer"
          },
          "n": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "rb" : {
            "description": "Reboot Action",
            "type": "boolean"
          },
          "id" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "if" : {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.rw",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          }
        },
        "anyOf" : [
```



```
          {
            "required": [ "fr" ]
          },
          {
            "required": [ "rb" ]
          },
          {
            "required": [ "err" ]
          }
        ],
        "type" : "object"
      },
      "mnt-update" : {
        "properties": {
          "fr" : {
            "description": "Factory Reset",
            "type": "boolean"
          },
          "n": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "rb" : {
            "description": "Reboot Action",
            "type": "boolean"
          }
        },
        "anyOf" : [
          {
            "required": [
              "fr"
            ]
          },
          {
            "required": [
              "rb"
            ]
          }
        ],
        "type" : "object"
      }
    }
}
```

D.9.5 Property definition

Table D-16 defines the Properties that are part of the "oic.wk.mnt" Resource Type.

Table D-16 – The Property definitions of the Resource with type "rt" = "oic.wk.mnt".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
fr	boolean	No	Read Write	Factory Reset.
err	integer	Yes	Read Only	Last HTTP occurred error.
n	multiple types: see schema	No	Read Write	
rb	boolean	No	Read Write	Reboot Action.
id	multiple types: see schema	No	Read Write	

if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
fr	boolean	No	Read Write	Factory Reset.
n	multiple types: see schema	No	Read Write	
rb	boolean	Yes	Read Write	Reboot Action.

D.9.6 CRUDN behaviour

Table D-17 defines the CRUDN operations that are supported on the "oic.wk.mnt" Resource Type.

Table D-17 – The CRUDN operations of the Resource with type "rt" = "oic.wk.mnt".

Create	Read	Update	Delete	Notify
	get	post		observe

D.10 Network Monitoring

D.10.1 Introduction

The Resource through which a Device can monitor network traffic.

D.10.2 Example URI

/nmonResURI

D.10.3 Resource type

The Resource Type is defined as: "oic.wk.nmon".

D.10.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Network Monitoring",
    "version": "2019-03-27",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/nmonResURI" : {
      "get": {
        "description": "The Resource through which a Device can monitor network traffic.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example": {
                "rt":   ["oic.wk.nmon"],
                "ianaifType":   71,
                "reset":   false,
```

```
                  "col" : false,
                  "tx" : 10,
                  "rx" : 15,
                  "mmstx" : 50,
                  "amstx" : 35,
                  "mmsrx" : 35,
                  "amsrx" : 20
                },
                "schema": { "$ref": "#/definitions/nmon" }
              }
            }
          },
          "post": {
            "description": "Start/Stop collecting and reset the networking monitor Resource\n",
            "parameters": [
              {"$ref": "#/parameters/interface-rw"},
              {
                "name": "body",
                "in": "body",
                "required": true,
                "schema": { "$ref": "#/definitions/nmon-update" },
                "x-example": {
                  "col":   true,
                  "reset":   true
                }
              }
            ],
            "responses": {
                "200": {
                  "description" : "",
                  "x-example": {
                    "rt":   ["oic.wk.nmon"],
                    "ianaifType":   71,
                    "reset":   false,
                    "col" : true,
                    "tx" : 0,
                    "rx" : 0,
                    "mmstx" : 0,
                    "amstx" : 0,
                    "mmsrx" : 0,
                    "amsrx" : 0
                  },
                  "schema": { "$ref": "#/definitions/nmon" }
                }
            }
          }
        }
      },
      "parameters": {
        "interface-rw" : {
          "in" : "query",
          "name" : "if",
          "type" : "string",
          "enum" : ["oic.if.rw"]
        },
        "interface-all" : {
          "in" : "query",
          "name" : "if",
          "type" : "string",
          "enum" : ["oic.if.rw", "oic.if.baseline"]
        }
      },
      "definitions": {
        "nmon" : {
          "properties": {
            "amstx" : {
              "description": "Average transmitted message size in bytes (tx) in the collection period",
              "readOnly": true,
              "type": "integer"
            },
            "reset" : {
```



```
            "description": "True: reset the collected values",
            "readOnly": false,
            "type": "boolean"
          },
          "mmsrx" : {
            "description": "Maximum received message size in bytes (rx) in the collection period",
            "readOnly": true,
            "type": "integer"
          },
          "mmstx" : {
            "description": "Maximum transmitted message size in bytes (tx) in the collection period",
            "readOnly": true,
            "type": "integer"
          },
          "tx" : {
            "description": "Amount of transmitted kilo bytes from the collection",
            "readOnly": true,
            "type": "integer"
          },
          "rt" : {
            "description": "Resource Type of the Resource",
            "items": {
              "enum": ["oic.wk.nmon"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "ianaifType" : {
            "description": "The type of the network connection, as defined by iana
    https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib",
            "readOnly": true,
            "type": "integer"
          },
          "rx" : {
            "description": "Amount of received kilobytes from the collection",
            "readOnly": true,
            "type": "integer"
          },
          "id" : {
            "$ref":
    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
    schema.json#/definitions/id"
          },
          "amsrx" : {
            "description": "Average received message size in bytes (rx) in the collection period",
            "readOnly": true,
            "type": "integer"
          },
          "n" : {
            "$ref":
    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
    schema.json#/definitions/n"
          },
          "col" : {
            "description": "True: Device is collecting values",
            "readOnly": false,
            "type": "boolean"
          },
          "if" : {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.rw",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
```



```
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["reset", "col", "ianaifType"]
      },
      "nmon-update" : {
        "properties": {
          "reset" : {
            "description": "True: reset the collected values",
            "readOnly": false,
            "type": "boolean"
          },
          "n" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "col" : {
            "description": "True: Device is collecting values",
            "readOnly": false,
            "type": "boolean"
          }
        },
        "type" : "object",
        "required": ["reset", "col"]
      }
    }
}
```

D.10.5 Property definition

Table D-18 defines the Properties that are part of the "oic.wk.nmon" Resource Type.

Table D-18 – The Property definitions of the Resource with type "rt" = "oic.wk.nmon".

Property name	Value type	Mandatory	Access mode	Description
amstx	integer	No	Read Only	Average transmitted message size in bytes (tx) in the collection period
reset	boolean	Yes	Read Write	True: reset the collected values
mmsrx	integer	No	Read Only	Maximum received message size in bytes (rx) in the collection period
mmstx	integer	No	Read Only	Maximum transmitted message size in bytes (tx) in the collection period
tx	integer	No	Read Only	Amount of transmitted kilo bytes from the collection
rt	array: see schema	No	Read Only	Resource Type of the Resource
ianaifType	integer	Yes	Read Only	The type of the network connection, as defined by iana https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib
rx	integer	No	Read Only	Amount of received kilobytes from the collection
id	multiple types: see schema	No	Read Write	
amsrx	integer	No	Read Only	Average received message size in bytes (rx) in the collection period

n	multiple types: see schema	No	Read Write	
col	boolean	Yes	Read Write	True: Device is collecting values
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource
reset	boolean	Yes	Read Write	True: reset the collected values
n	multiple types: see schema	No	Read Write	
col	boolean	Yes	Read Write	True: Device is collecting values

D.10.6 CRUDN behaviour

Table D-19 defines the CRUDN operations that are supported on the "oic.wk.nmon" Resource Type.

Table D-19 – The CRUDN operations of the Resource with type "rt" = "oic.wk.nmon".

Create	Read	Update	Delete	Notify
	get	post		observe

D.11 Platform

D.11.1 Introduction

Known Resource that is defines the Platform on which an Server is hosted. Allows for Platform specific information to be discovered.

D.11.2 Well-known URI

/oic/p

D.11.3 Resource type

The Resource Type is defined as: "oic.wk.p".

D.11.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Platform",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/oic/p" : {
      "get": {
        "description": "Known Resource that is defines the Platform on which an Server is
hosted.\nAllows for Platform specific information to be discovered.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
```

```
            "200": {
              "description" : "",
              "x-example": {
                "pi":   "54919CA5-4101-4AE4-595B-353C51AA983C",
                "rt":   ["oic.wk.p"],
                "mnmn": "Acme, Inc"
              },
              "schema": { "$ref": "#/definitions/Platform" }
            }
          }
        }
      }
    },
    "parameters": {
      "interface" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.r", "oic.if.baseline"]
      }
    },
    "definitions": {
      "Platform" : {
        "properties": {
          "rt" : {
            "description": "Resource Type of the Resource",
            "items": {
              "enum":   ["oic.wk.p"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "pi" : {
            "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
            "type": "string",
            "description": "Platform Identifier",
            "readOnly": true
          },
          "mnfv" : {
            "description": "Manufacturer's firmware version",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "vid" : {
            "description": "Manufacturer's defined information for the Platform. The content is
freeform, with population rules up to the manufacturer",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "mnmn" : {
            "description": "Manufacturer name",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "mnmo" : {
            "description": "Model number as designated by the manufacturer",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "mnhw" : {
            "description": "Platform Hardware Version",
            "maxLength": 64,
```



```
            "readOnly": true,
            "type": "string"
          },
          "mnos" : {
            "description": "Platform Resident OS Version",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "mndt" : {
            "pattern": "^([0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|2[0-9]|1[0-9]|0[1-9])$",
            "type": "string",
            "description": "Manufacturing Date.",
            "readOnly": true
          },
          "id" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "mnsl" : {
            "description": "Manufacturer's Support Information URL",
            "format": "uri",
            "maxLength": 256,
            "readOnly": true,
            "type": "string"
          },
          "mnpv" : {
            "description": "Platform Version",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "st" : {
            "description": "The date-time format pattern according to IETF RFC 3339.",
            "format": "date-time",
            "readOnly": true,
            "type": "string"
          },
          "n" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "mnml" : {
            "description": "Manufacturer's URL",
            "format": "uri",
            "maxLength": 256,
            "readOnly": true,
            "type": "string"
          },
          "mnsel" : {
            "description": "Serial number as designated by the manufacturer",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "if" : {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.r",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 2,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
```



```
          }

        },
        "type" : "object",
        "required": ["pi", "mnmn"]
      }
    }
}
```

D.11.5 Property definition

Table D-20 defines the Properties that are part of the "oic.wk.p" Resource Type.

Table D-20 – The Property definitions of the Resource with type "rt" = "oic.wk.p".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
pi	string	Yes	Read Only	Platform Identifier.
mnfv	string	No	Read Only	Manufacturer's firmware version.
vid	string	No	Read Only	Manufacturer's defined information for the Platform. The content is freeform, with population rules up to the manufacturer.
mnmn	string	Yes	Read Only	Manufacturer name.
mnmo	string	No	Read Only	Model number as designated by the manufacturer.
mnhw	string	No	Read Only	Platform Hardware Version.
mnos	string	No	Read Only	Platform Resident OS Version.
mndt	string	No	Read Only	Manufacturing Date.
id	multiple types: see schema	No	Read Write	
mnsl	string	No	Read Only	Manufacturer's Support Information URL.
mnpv	string	No	Read Only	Platform Version
st	string	No	Read Only	The date-time format pattern according to IETF RFC 3339.
n	multiple types: see schema	No	Read Write	
mnml	string	No	Read Only	Manufacturer's URL.
mnsel	string	No	Read Only	Serial number as designated by the manufacturer.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

D.11.6 CRUDN behaviour

Table D-21 defines the CRUDN operations that are supported on the "oic.wk.p" Resource Type.

Table D-21 – The CRUDN operations of the Resource with type "rt" = "oic.wk.p".

Create	Read	Update	Delete	Notify
	get			observe

D.12 Resource directory resource

D.12.1 Introduction

Resource to be exposed by any Device that can act as a Resource Directory.
1) Provides selector criteria (e.g., integer) with GET request
2) Publish a Link in /oic/res with POST request

D.12.2 Well-known URI

/oic/rd

D.12.3 Resource type

The Resource Type is defined as: "oic.wk.rd".

D.12.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Resource directory resource",
    "version": "2019-02-22",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/oic/rd" : {
      "get": {
        "description": "Resource to be exposed by any Device that can act as a Resource
Directory.\n1) Provides selector criteria (e.g., integer) with GET request\n2) Publish a Link in
/oic/res with POST request\n",
        "parameters": [
          {"$ref": "#/parameters/rdgetinterface"}
        ],
        "responses": {
            "200": {
              "description" : "Respond with the selector criteria - either the set of attributes or
the bias factor\n",
              "x-example": {
                "rt": ["oic.wk.rd"],
                "if": ["oic.if.baseline"],
                "sel": 50
              },
              "schema": { "$ref": "#/definitions/rdSelection" }
            }
        }
      },
```

```
      "post": {
        "description": "Publish the Resource information for the first time in /oic/res. Updates to
existing entries are not allowed.\nAppropriates parts of the information, i.e., Links of the
published Resources will be discovered through /oic/res.\n1) When a Device first publishes a Link,
the request payload to RD may include the Links without an \"ins\" Parameter.\n2) Upon granting the
request, the RD assigns a unique instance value identifying the Link among all the Links it
advertises\n   and sends back the instance value in the \"ins\" Parameter in the Link to the
publishing Device.\n",
        "parameters": [
          {"$ref": "#/parameters/rdpostinterface"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/rdPublish" },
            "x-example": {
              "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
              "links": [
                {
                  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
                  "href":   "/myLightSwitch",
                  "rt":     [ "oic.r.switch.binary" ],
                  "if":     [ "oic.if.a", "oic.if.baseline" ],
                  "p":      { "bm": 3 },
                  "eps": [
                    { "ep": "coaps://[2001:db8:a::b1d6]:1111", "pri": 2 },
                    { "ep": "coaps://[2001:db8:a::b1d6]:1122" },
                    { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
                  ]
                },
                {
                  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
                  "href":   "/myLightBrightness",
                  "rt":     [ "oic.r.brightness" ],
                  "if":     [ "oic.if.a", "oic.if.baseline" ],
                  "p":      { "bm": 3 },
                  "eps": [
                    { "ep": "coaps://[[2001:db8:a::123]:2222" }
                  ]
                }
              ],
              "ttl": 600
            }
          }
        ],
        "responses": {
          "200": {
            "description" : "Respond with the same schema as publish with the additional \"ins\"
Parameter in the Link.\n",
            "x-example": {
              "di": "e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
              "links": [
                {
                  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
                  "href":   "/myLightSwitch",
                  "rt":     [ "oic.r.switch.binary" ],
                  "if":     [ "oic.if.a", "oic.if.baseline" ],
                  "p":      { "bm": 3 },
                  "eps": [
                    { "ep": "coaps://[2001:db8:a::b1d6]:1111", "pri": 2 },
                    { "ep": "coaps://[2001:db8:a::b1d6]:1122" },
                    { "ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3 }
                  ],
                  "ins":    11235
                },
                {
                  "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
                  "href":   "/myLightBrightness",
                  "rt":     ["oic.r.brightness"],
                  "if":     ["oic.if.a", "oic.if.baseline"],
                  "p":      {"bm": 3},
```



```
                    "eps": [
                        {"ep": "coaps://[2001:db8:a::123]:2222"}
                    ],
                    "ins":    112358
                  }
                ],
                "ttl": 600
              },
              "schema": { "$ref": "#/definitions/rdPublish" }
            }
          }
        }
      }
    },
    "parameters": {
      "rdgetinterface" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.baseline"]
      },
      "rdpostinterface" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.baseline"]
      }
    },
    "definitions": {
      "rdSelection" : {
        "properties": {
          "rt" : {
            "description": "Resource Type of the Resource",
            "items": {
              "enum": ["oic.wk.rd"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "n" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "sel" : {
            "description": "A bias factor calculated by the Resource Directory",
            "maximum": 100,
            "minimum": 0,
            "readOnly": true,
            "type": "integer"
          },
          "id" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "if" : {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "readOnly": true,
```



```
            "uniqueItems": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["sel"]
      },
      "rdPublish" : {
        "properties": {
          "di" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
          },
          "ttl" : {
            "description": "Time to indicate a RD, i.e. how long to keep this published item.",
            "type": "integer"
          },
          "links" : {
            "description": "A set of simple or individual OCF Links.",
            "items": {
              "properties": {
                "anchor": {
                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
                },
                "di": {
                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
                },
                "eps": {
                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
                },
                "href": {
                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
                },
                "if": {
                  "description": "The interface set supported by the published resource",
                  "items": {
                    "enum": [
                      "oic.if.baseline",
                      "oic.if.ll",
                      "oic.if.b",
                      "oic.if.rw",
                      "oic.if.r",
                      "oic.if.a",
                      "oic.if.s"
                    ],
                    "type": "string",
                    "maxLength": 64
                  },
                  "minItems": 1,
                  "uniqueItems": true,
                  "type": "array"
                },
                "ins": {
                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
                },
                "p": {
                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
                },
```



```
                "rel": {
                  "description": "The relation of the target URI referenced by the Link to the context
URI",
                  "oneOf": [
                    {
                      "default": [
                        "hosts"
                      ],
                      "items": {
                        "maxLength": 64,
                        "type": "string"
                      },
                      "minItems": 1,
                      "type": "array"
                    },
                    {
                      "default": "hosts",
                      "maxLength": 64,
                      "type": "string"
                    }
                  ]
                },
                "rt": {
                  "description": "Resource Type of the published Resource",
                  "items": {
                    "maxLength": 64,
                    "type": "string"
                  },
                  "minItems": 1,
                  "maxItems": 1,
                  "uniqueItems": true,
                  "type": "array"
                },
                "title": {
                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
                },
                "type": {
                  "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
                }
              },
              "required": [
                "href",
                "rt",
                "if"
              ],
              "type": "object"
            },
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["di", "links", "ttl"]
      }
    }
  }
```

D.12.5 Property definition

Table D-22 defines the Properties that are part of the "oic.wk.rd" Resource Type.

7637 **Table D-22 – The Property definitions of the Resource with type "rt" = "oic.wk.rd".**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource.
n	multiple types: see schema	No	Read Write	
sel	integer	Yes	Read Only	A bias factor calculated by the Resource Directory.
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
di	multiple types: see schema	Yes	Read Write	
ttl	integer	Yes	Read Write	Time to indicate a RD, i.e. how long to keep this published item.
links	array: see schema	Yes	Read Write	A set of simple or individual OCF Links.

7638 ### D.12.6 CRUDN behaviour

7639 Table D-23 defines the CRUDN operations that are supported on the "oic.wk.rd" Resource Type.

7640 **Table D-23 – The CRUDN operations of the Resource with type "rt" = "oic.wk.rd".**

Create	Read	Update	Delete	Notify
	get	post		observe

7641 ## D.13 Discoverable Resources

7642 ### D.13.1 Introduction

7643 Baseline representation of /oic/res; list of discoverable Resources
7644

7645 ### D.13.2 Well-known URI

7646 /oic/res

7647 ### D.13.3 Resource type

7648 The Resource Type is defined as: "oic.wk.res".

7649 ### D.13.4 OpenAPI 2.0 definition

```
7650 {
7651   "swagger": "2.0",
7652   "info": {
7653     "title": "Discoverable Resources",
7654     "version": "2019-03-13",
7655     "license": {
7656       "name": "OCF Data Model License",
7657       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
7658       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
7659     },
```

```
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/oic/res?if=oic.if.ll": {
      "get": {
        "description": "Links list representation of /oic/res; list of discoverable Resources\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": [
              {
                "href": "/humidity",
                "rt":   ["oic.r.humidity"],
                "if":   ["oic.if.s", "oic.if.baseline"],
                "p":    {"bm": 3},
                "eps": [
                  {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
                  {"ep": "coaps://[fe80::b1d6]:1122"},
                  {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
                ]
              },
              {
                "href": "/temperature",
                "rt":   ["oic.r.temperature"],
                "if":   ["oic.if.s", "oic.if.baseline"],
                "p":    {"bm": 3},
                "eps": [
                  {"ep": "coaps://[[2001:db8:a::123]:2222"}
                ]
              }
            ],
            "schema": {
              "$ref": "#/definitions/slinklist"
            }
          }
        }
      }
    },
    "/oic/res?if=oic.if.baseline": {
      "get": {
        "description": "Baseline representation of /oic/res; list of discoverable Resources\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
        "responses": {
          "200": {
            "description": "",
            "x-example": [
              {
                "rt": ["oic.wk.res"],
                "if": ["oic.if.ll", "oic.if.baseline"],
                "links": [
                  {
                    "href": "/humidity",
                    "rt":   ["oic.r.humidity"],
```

```
                    "if":   ["oic.if.s", "oic.if.baseline"],
                    "p":    {"bm": 3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
                      {"ep": "coaps://[fe80::b1d6]:1122"},
                      {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                    ]
                  },
                  {
                    "href": "/temperature",
                    "rt":   ["oic.r.temperature"],
                    "if":   ["oic.if.s", "oic.if.baseline"],
                    "p":    {"bm": 3},
                    "eps": [
                      {"ep": "coaps://[[2001:db8:a::123]:2222"}
                    ]
                  }
                ]
              }
            ],
            "schema": {
              "$ref": "#/definitions/sbaseline"
            }
          }
        }
      }
    }
  },
  "parameters": {
    "interface-all": {
      "in": "query",
      "name": "if",
      "type": "string",
      "enum": ["oic.if.ll", "oic.if.baseline"]
    }
  },
  "definitions": {
    "oic.oic-link": {
      "type": "object",
      "properties": {
        "anchor": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
        },
        "di": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
        },
        "eps": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
        },
        "href": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
        },
        "if": {
          "description": "The OCF Interfaces supported by the Linked Resource",
          "items": {
            "enum": [
              "oic.if.baseline",
              "oic.if.ll",
              "oic.if.b",
              "oic.if.rw",
              "oic.if.r",
              "oic.if.a",
              "oic.if.s"
```



```
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "type": "array"
          },
          "ins": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
          },
          "p": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
          },
          "rel": {
            "description": "The relation of the target URI referenced by the Link to the context URI",
            "oneOf": [
              {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
              },
              {
                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_string"
              }
            ]
          },
          "rt": {
            "description": "Resource Type of the Linked Resource",
            "items": {
              "maxLength": 64,
              "type": "string"
            },
            "minItems": 1,
            "uniqueItems": true,
            "type": "array"
          },
          "title": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
          },
          "type": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
          }
        },
        "required": [
          "href",
          "rt",
          "if"
        ]
      },
      "slinklist": {
        "type" : "array",
        "readOnly": true,
        "items": {
          "$ref": "#/definitions/oic.oic-link"
        }
      },
      "sbaseline": {
        "type": "array",
        "minItems": 1,
        "maxItems": 1,
```

```
        "items": {
          "type": "object",
          "properties": {
            "n": {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
            },
            "id": {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
            },
            "rt": {
              "description": "Resource Type of this Resource",
              "items": {
                "enum": ["oic.wk.res"],
                "type": "string",
                "maxLength": 64
              },
              "minItems": 1,
              "readOnly": true,
              "uniqueItems": true,
              "type": "array"
            },
            "if": {
              "description": "The OCF Interfaces supported by this Resource",
              "items": {
                "enum": [
                  "oic.if.ll",
                  "oic.if.baseline"
                ],
                "type": "string",
                "maxLength": 64
              },
              "minItems": 2,
              "readOnly": true,
              "uniqueItems": true,
              "type": "array"
            },
            "links": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/oic.oic-link"
              }
            }
          }
        },
        "required": [
          "rt",
          "if",
          "links"
        ]
      }
    }
  }
```

D.13.5 Property definition

Table D-24 defines the Properties that are part of the "oic.wk.res" Resource Type.

Table D-24 – The Property definitions of the Resource with type "rt" = "None".

Property name	Value type	Mandatory	Access mode	Description
anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	

eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Write	The OCF Interfaces supported by the Linked Resource.
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the Link to the context URI.
rt	array: see schema	Yes	Read Write	Resource Type of the Linked Resource.
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
rt	array: see schema	Yes	Read Only	Resource Type of this Resource.
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
links	array: see schema	Yes	Read Write	

D.13.6 CRUDN behaviour

Table D-25 defines the CRUDN operations that are supported on the "None" Resource Type.

Table D-25 – The CRUDN operations of the Resource with type "rt" = "None".

Create	Read	Update	Delete	Notify
	get			observe

D.14 SceneList

D.14.1 Introduction

Toplevel Scene Resource. This Resource is a generic Collection Resource. The rts value contains oic.wk.scenecollection Resource Types.

D.14.2 Example URI

/SceneListResURI

D.14.3 Resource type

The Resource Type is defined as: "oic.wk.scenelist".

D.14.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Scene List",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/SceneListResURI?if=oic.if.ll": {
      "get": {
        "description": "Toplevel Scene Resource.\nThis Resource is a generic Collection
Resource.\nThe rts value contains oic.wk.scenecollection Resource Types.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": [
              {"href": "/scenecollection1","rt": ["oic.wk.scenecollection"],"if":["oic.if.ll",
"oic.if.baseline"]},
              {"href": "/scenecollection2","rt": ["oic.wk.scenecollection"],"if":["oic.if.ll",
"oic.if.baseline"]}
            ],
            "schema": {
              "$ref": "#/definitions/slinks"
            }
          }
        }
      }
    },
    "/SceneListResURI?if=oic.if.baseline": {
      "get": {
        "description": "Toplevel Scene Resource.\nThis Resource is a generic Collection
Resource.\nThe rts value contains oic.wk.scenecollection Resource Types.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": {
              "rt":  ["oic.wk.scenelist"],
              "if":  ["oic.if.ll", "oic.if.baseline"],
              "n":   "list of scene collections",
              "rts": ["oic.wk.scenecollection"],
              "links": [
```

```
                {"href": "/scenecollection1","rt": ["oic.wk.scenecollection"],"if":["oic.if.ll",
"oic.if.baseline"]},
                {"href": "/scenecollection2","rt": ["oic.wk.scenecollection"],"if":["oic.if.ll",
"oic.if.baseline"]}
              ]
            },
            "schema": { "$ref": "#/definitions/Collection" }
          }
        }
      }
    }
  },
  "parameters": {
    "interface-all" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.ll", "oic.if.baseline"]
    }
  },
  "definitions": {
    "Collection": {
      "properties": {
        "links": {
          "description": "A set of simple or individual OCF Links.",
          "items": {
            "$ref": "#/definitions/oic.oic-link"
          },
          "type": "array"
        },
        "n": {
          "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
        },
        "id": {
          "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
        },
        "if": {
          "type": "array",
          "description": "The OCF Interfaces supported by this Resource",
          "items": {
            "enum": [
              "oic.if.ll",
              "oic.if.baseline"
            ],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 2,
          "uniqueItems": true,
          "readOnly": true
        },
        "rts": {
          "description": "The list of allowable Resource Types in Links included in the Collection",
          "items": {
            "enum": ["oic.wk.scenecollection"],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "uniqueItems": true,
          "readOnly": true,
          "type": "array"
        },
        "rt": {
          "description": "Resource Type of the Resource",
          "items": {
            "enum": ["oic.wk.scenelist"],
```



```
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          }
        },
        "type": "object",
        "required": [
          "rt",
          "if",
          "links"
        ]
      },
      "slinks" : {
        "type" : "array",
        "items" : {
          "$ref": "#/definitions/oic.oic-link"
        }
      },
      "oic.oic-link": {
        "properties": {
          "if": {
            "description": "The OCF Interfaces supported by the Linked Resource",
            "items": {
              "enum": [
                "oic.if.ll",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "rt": {
            "description": "The Resource Type of the Linked Resource",
            "items": {
              "enum": ["oic.wk.scenecollection"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "anchor": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
          },
          "di": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
          },
          "eps": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
          },
          "href": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
          },
```



```
        "ins": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
        },
        "p": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
        },
        "rel": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
        },
        "title": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
        },
        "type": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
        }
      },
      "required": [
        "href",
        "rt",
        "if"
      ],
      "type": "object"
    }
  }
}
```

D.14.5 Property definition

Table D-26 defines the Properties that are part of the "oic.wk.scenelist" Resource Type.

Table D-26 – The Property definitions of the Resource with type "rt" = "oic.wk.scenelist".

Property name	Value type	Mandatory	Access mode	Description
links	array: see schema	Yes	Read Write	A set of simple or individual OCF Links.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
rts	array: see schema	No	Read Only	The list of allowable Resource Types in Links included in the Collection.
rt	array: see schema	Yes	Read Only	Resource Type of the Resource.
if	array: see schema	Yes	Read Only	The OCF Interfaces

				supported by the Linked Resource.
rt	array: see schema	Yes	Read Only	The Resource Type of the Linked Resource.
anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

8193 **D.14.6 CRUDN behaviour**

8194 Table D-27 defines the CRUDN operations that are supported on the "oic.wk.scenelist" Resource
8195 Type.

8196 **Table D-27 – The CRUDN operations of the Resource with type "rt" = "oic.wk.scenelist".**

Create	Read	Update	Delete	Notify
	get			observe

8197 **D.15 Scene Collection**

8198 **D.15.1 Introduction**

8199 Collection that models a set of Scenes.
8200 This Resource is a generic Collection Resource with additional Properties.
8201 The rts value contains oic.scenemember Resource Types.
8202 The additional Properties are
8203 lastScene, this is the Scene Value last set by any Client
8204 sceneValues, this is the list of available Scenes
8205 lastScene shall be listed in sceneValues.
8206

8207 **D.15.2 Example URI**

8208 /SceneCollectionResURI

8209 **D.15.3 Resource type**

8210 The Resource Type is defined as: "oic.wk.scenecollection".

D.15.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Scene Collection",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/SceneCollectionResURI?if=oic.if.ll" : {
      "get": {
        "description": "Collection that models a set of Scenes.\nThis Resource is a generic
Collection Resource with additional Properties.\nThe rts value contains oic.scenemember Resource
Types.\nThe additional Properties are\n  lastScene, this is the Scene Value last set by any Client\n
sceneValues, this is the list of available Scenes\n  lastScene shall be listed in sceneValues.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": [
              {"href": "/scenemember1","rt": ["oic.wk.scenemember"],"if":["oic.if.baseline"]},
              {"href": "/scenemember2","rt": ["oic.wk.scenemember"],"if":["oic.if.baseline"]}
            ],
            "schema": {
              "$ref": "#/definitions/slinks"
            }
          }
        }
      }
    },
    "/SceneCollectionResURI?if=oic.if.baseline" : {
      "get": {
        "description": "Collection that models a set of Scenes.\nThis Resource is a generic
Collection Resource with additional Properties.\nThe rts value contains oic.scenemember Resource
Types.\nThe additional Properties are\n  lastScene, this is the Scene Value last set by any Client\n
sceneValues, this is the list of available Scenes\n  lastScene shall be listed in sceneValues.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-all"
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": {
              "lastScene": "off",
              "sceneValues": ["off","Reading","TVWatching"],
              "rt":       ["oic.wk.scenecollection"],
              "n":        "My Scenes for my living room",
              "rts":      ["oic.wk.scenemember"],
              "links": [
                {"href": "/scenemember1","rt": ["oic.wk.scenemember"],"if":["oic.if.baseline"]},
```

```
                {"href": "/scenemember2","rt": ["oic.wk.scenemember"],"if":["oic.if.baseline"]}
              ]
            },
            "schema": {
              "$ref": "#/definitions/SceneCollection"
            }
          }
        }
      },
      "post": {
        "description": "Provides the action to change the last set Scene selection.\nCalling this
method shall update all Scene Members to the prescribed membervalue.\nWhen this method is called
with the same value as the current lastScene value\nthen all Scene Members shall be updated.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-update"
          },
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": {
              "$ref": "#/definitions/SceneCollectionUpdate"
            },
            "x-example": {
              "lastScene": "Reading"
            }
          }
        ],
        "responses": {
          "200": {
            "description" : "Indicates that the value is changed.\nThe changed Properties are
provided in the response.\n",
            "x-example": {
              "lastScene": "Reading"
            },
            "schema": {
              "$ref": "#/definitions/SceneCollectionUpdate"
            }
          }
        }
      }
    }
  },
  "parameters": {
    "interface-update" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.a"]
    },
    "interface-all" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.ll", "oic.if.baseline"]
    }
  },
  "definitions": {
    "SceneCollection": {
      "properties": {
        "rt": {
          "description": "Resource Type of the Resource",
          "items": {
            "enum": ["oic.wk.scenecollection"],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "readOnly": true,
          "uniqueItems": true,
```

```
            "type": "array"
          },
          "lastScene": {
            "description": "Last selected Scene from the set of sceneValues",
            "type": "string"
          },
          "links": {
            "description": "A set of simple or individual OCF Links.",
            "items": {
              "$ref": "#/definitions/oic.oic-link"
            },
            "type": "array"
          },
          "sceneValues": {
            "description": "All available Scene Values",
            "items": {
              "type": "string"
            },
            "readOnly": true,
            "type": "array"
          },
          "n": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "id": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "rts": {
            "description": "Resource Type of the Resources within the Collection",
            "items": {
              "enum": ["oic.wk.scenemember"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          },
          "if" : {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.ll",
                "oic.if.baseline",
                "oic.if.a"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          }
        },
        "type" : "object"
      },
      "SceneCollectionUpdate": {
        "properties": {
          "lastScene": {
            "description": "Last selected Scene from the set of sceneValues",
            "type": "string"
          }
        },
        "type" : "object"
      },
```



```
      "slinks" : {
        "type" : "array",
        "items" : {
          "$ref": "#/definitions/oic.oic-link"
        }
      },
      "oic.oic-link": {
        "type": "object",
        "properties": {
          "if": {
            "description": "The OCF Interfaces supported by the Linked Resource",
            "items": {
              "enum": [
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "rt": {
            "description": "Resource Type of the Linked Resource",
            "items": {
              "enum": ["oic.wk.scenemember"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "anchor": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
          },
          "di": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
          },
          "eps": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
          },
          "href": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
          },
          "ins": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
          },
          "p": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
          },
          "rel": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
          },
          "title": {
```



```
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
        },
        "type": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
        }
      },
      "required": [
        "href",
        "rt",
        "if"
      ]
    }
  }
}
```

D.15.5 Property definition

Table D-28 defines the Properties that are part of the "oic.wk.scenecollection" Resource Type.

Table D-28 – The Property definitions of the Resource with type "rt" = "oic.wk.scenecollection".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema		Read Only	Resource Type of the Resource.
lastScene	string		Read Write	Last selected Scene from the set of sceneValues.
links	array: see schema		Read Write	A set of simple or individual OCF Links.
sceneValues	array: see schema		Read Only	All available Scene Values.
n	multiple types: see schema		Read Write	
id	multiple types: see schema		Read Write	
rts	array: see schema		Read Only	Resource Type of the Resources within the Collection.
if	array: see schema		Read Only	The OCF Interfaces supported by this Resource.
lastScene	string		Read Write	Last selected Scene from the set of sceneValues.
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by the Linked Resource.
rt	array: see schema	Yes	Read Only	Resource Type of the Linked Resource.
anchor	multiple types: see schema	No	Read Write	

di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

D.15.6 CRUDN behaviour

Table D-29 defines the CRUDN operations that are supported on the "oic.wk.scenecollection" Resource Type.

Table D-29 – The CRUDN operations of the Resource with type "rt" = "oic.wk.scenecollection".

Create	Read	Update	Delete	Notify
	get	post		observe

D.16 Scene Member

D.16.1 Introduction

Single Link that models a Scene Member.

D.16.2 Example URI

/SceneMemberResURI

D.16.3 Resource type

The Resource Type is defined as: "oic.wk.scenemember".

D.16.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Scene Member",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": [
    "http"
  ],
  "consumes": [
    "application/json"
```



```
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/SceneMemberResURI" : {
      "get": {
        "description": "Single Link that models a Scene Member.\n",
        "parameters": [
          {
            "$ref": "#/parameters/interface-baseline"
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": {
              "rt": ["oic.wk.scenemember"],
              "id": "0685B960-FFFF-46F7-BEC0-9E6234671ADC1",
              "n": "my binary switch (for light bulb) mappings",
              "if": ["oic.if.baseline"],
              "link": {
                "href": "binarySwitch",
                "rt":   ["oic.r.switch.binary"],
                "if":   ["oic.if.a", "oic.if.baseline"],
                "eps": [
                  {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                  {"ep": "coaps://[fe80::b1d6]:1122"},
                  {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                ]
              },
              "SceneMappings": [
                {
                  "scene":           "off",
                  "memberProperty":  "value",
                  "memberValue":     "true"
                },
                {
                  "scene":           "Reading",
                  "memberProperty":  "value",
                  "memberValue":     "false"
                },
                {
                  "scene":           "TVWatching",
                  "memberProperty":  "value",
                  "memberValue":     "true"
                }
              ]
            },
            "schema": {
              "$ref": "#/definitions/SceneMember"
            }
          }
        }
      }
    },
    "parameters": {
      "interface-baseline" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.baseline"]
      }
    },
    "definitions": {
      "SceneMember": {
        "properties": {
          "rt": {
            "description": "Resource Type of the Resource",
            "items": {
```



```
              "enum": ["oic.wk.scenemember"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          },
          "SceneMappings": {
            "description": "Array of mappings per Scene, can be one(1)",
            "items": {
              "properties": {
                "memberProperty": {
                  "description": "Property name that will be mapped",
                  "readOnly": true,
                  "type": "string"
                },
                "memberValue": {
                  "description": "Value of the Member Property",
                  "readOnly": true,
                  "type": "string"
                },
                "scene": {
                  "description": "Specifies a Scene Value that will be acted upon",
                  "type": "string"
                }
              },
              "required": [
                "scene",
                "memberProperty",
                "memberValue"
              ],
              "type": "object"
            },
            "type": "array"
          },
          "n": {
            "$ref" :
    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
    schema.json#/definitions/n"
          },
          "id": {
            "$ref" :
    "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
    schema.json#/definitions/id"
          },
          "link": {
            "$ref": "#/definitions/oic.oic-link"
          },
          "if": {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required": [
          "rt",
          "if",
          "SceneMappings"
        ]
```



```
      },
      "oic.oic-link": {
        "properties": {
          "if": {
            "description": "The OCF Interfaces supported by the target Resource",
            "items": {
              "enum": [
                "oic.if.baseline",
                "oic.if.ll",
                "oic.if.b",
                "oic.if.lb",
                "oic.if.rw",
                "oic.if.r",
                "oic.if.a",
                "oic.if.s"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "rt": {
            "description": "Resource Type of the target Resource",
            "items": {
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          },
          "anchor": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
          },
          "di": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
          },
          "eps": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
          },
          "href": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
          },
          "ins": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
          },
          "p": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
          },
          "rel": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
          },
          "title": {
```



```
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
        },
        "type": {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
        }
      },
      "required": [
        "href",
        "rt",
        "if"
      ],
      "type": "object"
    }
  }
}
```

D.16.5 Property definition

Table D-30 defines the Properties that are part of the "oic.wk.scenemember" Resource Type.

Table D-30 – The Property definitions of the Resource with type "rt" = "oic.wk.scenemember".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	Yes	Read Only	Resource Type of the Resource.
SceneMappings	array: see schema	Yes	Read Write	Array of mappings per Scene, can be one(1).
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
link	multiple types: see schema	No	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by the target Resource.
rt	array: see schema	Yes	Read Only	Resource Type of the target Resource.
anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	

p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	

D.16.6 CRUDN behaviour

Table D-31 defines the CRUDN operations that are supported on the "oic.wk.scenemember" Resource Type.

Table D-31 – The CRUDN operations of the Resource with type "rt" = "oic.wk.scenemember".

Create	Read	Update	Delete	Notify
	get			observe

D.17 Alert

D.17.1 Introduction

This Resource provides a mechanism for a Server to expose information to an interested party with regard to error or other conditions that the Device is experiencing (Alerts). category is a string that contains the Device defined category for the Alert. timestamp is an RFC3339 formatted time at which the Alert was generated. originatorid is a string that contains the identity of the originator of the Alert. severity is an integer that contains the RFC5424 defined severity of the Alert. subject is an array containing human readable text in one or more languages. accountid is a string containing the identity of the account with which the Device is associated.

D.17.2 Example URI

/AlertResURI

D.17.3 Resource type

The Resource Type is defined as: "oic.r.alert".

D.17.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Alert",
    "version": "2019-02-28",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/AlertResURI" : {
      "get": {
        "description": "This Resource provides a mechanism for a Server to expose information to
```

```
an\ninterested party with regard to error or other conditions that the Device is experiencing
(Alerts).\ncategory is a string that contains the Device defined category for the Alert.\ntimestamp
is an RFC3339 formatted time at which the Alert was generated.\noriginatorid is a string that
contains the identity of the originator of the Alert.\nseverity is an integer that contains the
RFC5424 defined severity of the Alert.\nsubject is an array containing human readable text in one or
more languages.\naccountid is a string containing the identity of the account with which the Device
is associated.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt":           ["oic.r.alert"],
                  "accountid":    "MyAccountID",
                  "category":     "MyCategory",
                  "timestamp":    "2018-02-28T08:00:00Z",
                  "originatorid": "MyOriginatorID",
                  "severity":     3,
                  "subject":      [{"language":"en-US","value":"System error"}]
                },
              "schema": { "$ref": "#/definitions/Alert" }
            }
        }
      }
    },
    "parameters": {
      "interface" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.r", "oic.if.baseline"]
      }
    },
    "definitions": {
      "Alert" : {
        "properties": {
          "category": {
            "description": "Category into which the notification is classified",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "rt": {
            "description": "Resource Type",
            "items": {
              "maxLength": 64,
              "type": "string",
              "enum": ["oic.r.alert"]
            },
            "minItems": 1,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          },
          "severity": {
            "description": "RFC 5424 severity of the alert",
            "maximum": 7,
            "minimum": 0,
            "readOnly": true,
            "type": "integer"
          },
          "timestamp": {
            "description": "An RFC3339 formated time indicating when the data was observed (e.g.:
2016-02-15T09:19Z, 1996-12-19T16:39:57-08:00)",
            "format": "date-time",
            "readOnly": true,
            "type": "string"
```



```
          },
          "subject": {
            "description": "Alert subject matter.",
            "items": {
              "properties": {
                "language": {
                  "allOf": [
                    {
                      "description": "An identifier formatted according to IETF RFC 5646 (langusge
tag).",
                      "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
                      "type": "string"
                    },
                    {
                      "description": "An RFC 5646 language tag.",
                      "readOnly": true
                    }
                  ]
                },
                "value": {
                  "description": "Alert subject matter in the indicated language.",
                  "maxLength": 255,
                  "readOnly": true,
                  "type": "string"
                }
              },
              "type": "object"
            },
            "minItems": 1,
            "readOnly": true,
            "type": "array"
          },
          "originatorid": {
            "description": "ID of the creator of the event",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "n": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "id": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "accountid": {
            "description": "ID of the account",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
          },
          "if": {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.r",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 2,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
          }
        },
        "type" : "object",
```

```
        "required": ["category", "timestamp", "originatorid", "severity"]
      }
    }
}
```

D.17.5 Property definition

Table D-32 defines the Properties that are part of the "oic.r.alert" Resource Type.

Table D-32 – The Property definitions of the Resource with type "rt" = "oic.r.alert".

Property name	Value type	Mandatory	Access mode	Description
category	string	Yes	Read Only	Category into which the notification is classified.
rt	array: see schema	No	Read Only	Resource Type.
severity	integer	Yes	Read Only	RFC 5424 severity of the alert.
timestamp	string	Yes	Read Only	An RFC3339 formated time indicating when the data was observed (e.g.: 2016-02-15T09:19Z, 1996-12-19T16:39:57-08:00).
subject	array: see schema	No	Read Only	Alert subject matter.
originatorid	string	Yes	Read Only	ID of the creator of the event.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
accountid	string	No	Read Only	ID of the account.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

D.17.6 CRUDN behaviour

Table D-33 defines the CRUDN operations that are supported on the "oic.r.alert" Resource Type.

Table D-33 – The CRUDN operations of the Resource with type "rt" = "oic.r.alert".

Create	Read	Update	Delete	Notify
	get			observe

D.18 Alert Collection

D.18.1 Introduction

This Resource is a Collection containing instances of Alerts (oic.r.alert). This is the response using the baseline interface.

D.18.2 Example URI

/AlertCollectionResURI

D.18.3 Resource type

The Resource Type is defined as: "oic.r.alertcollection".

D.18.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Alert Collection",
    "version": "2019-03-04",
    "license": {
      "name": "OCF Data Model License",
      "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
      "x-copyright": "Copyright 2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/AlertCollectionResURI?if=oic.if.ll" : {
      "get": {
        "description": "This Resource is a Collection containing instances of Alerts
(oic.r.alert).\nThis is the response using the links list OCF Interface.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": [
              {"href": "/myAlert1ResURI", "rt": ["oic.r.alert"], "if":
["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
              {"href": "/myAlert2ResURI", "rt": ["oic.r.alert"], "if":
["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
              {"href": "/myAlert3ResURI", "rt": ["oic.r.alert"], "if":
["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
              {"href": "/myAlert4ResURI", "rt": ["oic.r.alert"], "if":
["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]}
            ],
            "schema": { "$ref": "#/definitions/AlertCollection-ll" }
          }
        }
      }
    },
    "/AlertCollectionResURI?if=oic.if.b" : {
      "get": {
        "description": "This Resource is a Collection containing instances of Alerts
(oic.r.alert).\nThis is the response using the Batch interface.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": [
```

```
                {
                  "href": "/Alert1ResURI",
                  "rep":{
                    "rt":           ["oic.r.alert"],
                    "accountid":    "MyAccountID",
                    "category":     "MyCategory",
                    "timestamp":    "2018-02-28T08:00:00Z",
                    "originatorid": "MyOriginatorID",
                    "severity": 3,
                    "subject": [{"language":"en-US","value":"System error"}]
                  }
                },
                {
                  "href": "/Alert2ResURI",
                  "rep":{
                    "rt":           ["oic.r.alert"],
                    "accountid":    "MyAccountID",
                    "category":     "MyCategory",
                    "timestamp":    "2018-02-28T08:15:00Z",
                    "originatorid": "MyOriginatorID",
                    "severity": 4,
                    "subject": [{"language":"en-US","value":"Network error"}]
                  }
                }
              ],
              "schema": { "$ref": "#/definitions/AlertCollection-b" }
            }
          }
        }
      },
      "/AlertCollectionResURI?if=oic.if.baseline" : {
        "get": {
          "description": "This Resource is a Collection containing instances of Alerts
(oic.r.alert).\nThis is the response using the baseline interface.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"}
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example": {
                "rt":    ["oic.r.alertcollection"],
                "rts":   ["oic.r.alert"],
                "if":    ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
                "links": [
                  {"href": "/myAlert1ResURI", "rt": ["oic.r.alert"], "if":
["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
                  {"href": "/myAlert2ResURI", "rt": ["oic.r.alert"], "if":
["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
                  {"href": "/myAlert3ResURI", "rt": ["oic.r.alert"], "if":
["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]},
                  {"href": "/myAlert4ResURI", "rt": ["oic.r.alert"], "if":
["oic.if.r","oic.if.baseline"], "eps": [{"ep": "coaps://[fe80::b1d6]:1122"}]}
                ]
              },
              "schema": { "$ref": "#/definitions/AlertCollection-baseline" }
            }
          }
        }
      }
    },
    "parameters": {
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
      }
    },
    "definitions": {
      "AlertCollection-b" : {
```

```
        "type": "array",
        "minItems": 0,
        "uniqueItems": true,
        "items": {
          "type": "object",
          "additionalProperties": true,
          "properties": {
            "href": {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
            },
            "rep": {
              "$ref":
"http://openconnectivityfoundation.github.io/core/swagger2.0/oic.r.alert.swagger.json#/definitions/A
lert"
            }
          },
          "required": [
            "href",
            "rep"
          ]
        }
      },
      "AlertCollection-baseline" : {
        "properties": {
          "n": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "id": {
            "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "rt": {
            "items": {
              "type": "string",
              "enum": ["oic.r.alertcollection"],
              "maxLength": 64
            },
            "minItems": 1,
            "type": "array",
            "uniqueItems": true,
            "readOnly": true
          },
          "rts": {
            "items": {
              "type": "string",
              "enum": ["oic.r.alert"],
              "maxLength": 64
            },
            "minItems": 1,
            "type": "array",
            "uniqueItems": true,
            "readOnly": true
          },
          "if": {
            "description": "The OCF Interfaces supported by this Resource",
            "items": {
              "enum": [
                "oic.if.ll",
                "oic.if.b",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 3,
            "readOnly": true,
```



```
            "uniqueItems": true,
            "type": "array"
          },
          "links": {
            "description": "A set of simple or individual Links.",
            "items": {
              "$ref": "#/definitions/oic.oic-link"
            },
            "type": "array"
          }
        },
        "type" : "object",
        "required": ["rt","rts","if","links"]
      },
      "AlertCollection-ll" : {
        "type": "array",
        "items": {
          "$ref": "#/definitions/oic.oic-link"
        }
      },
      "oic.oic-link": {
        "type": "object",
        "properties": {
          "anchor": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
          },
          "di": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
          },
          "eps": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
          },
          "href": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
          },
          "ins": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
          },
          "p": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
          },
          "rel": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
          },
          "title": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
          },
          "type": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
          },
          "if": {
            "description": "The OCF Interfaces supported by the target Resource",
            "items": {
```



```
              "enum": [
                "oic.if.r",
                "oic.if.baseline"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 2,
            "uniqueItems": true,
            "type": "array",
            "readOnly": true
          },
          "rt": {
            "description": "Resource Type of the target Resource",
            "items": {
              "maxLength": 64,
              "type": "string",
              "enum": ["oic.r.alert"]
            },
            "minItems": 1,
            "type": "array",
            "uniqueItems": true,
            "readOnly": true
          }
        },
        "required": [
          "href",
          "rt",
          "if"
        ]
      }
    }
  }
```

D.18.5 Property definition

Table D-34 defines the Properties that are part of the "oic.r.alertcollection" Resource Type.

Table D-34 – The Property definitions of the Resource with type "rt" = "oic.r.alertcollection".

Property name	Value type	Mandatory	Access mode	Description
href	multiple types: see schema	Yes	Read Write	
rep	multiple types: see schema	Yes	Read Write	
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
rt	array: see schema	Yes	Read Only	
rts	array: see schema	Yes	Read Only	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by this Resource.
links	array: see schema	Yes	Read Write	A set of simple or individual Links.
anchor	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	

eps	multiple types: see schema	No	Read Write	
href	multiple types: see schema	Yes	Read Write	
ins	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
type	multiple types: see schema	No	Read Write	
if	array: see schema	Yes	Read Only	The OCF Interfaces supported by the target Resource.
rt	array: see schema	Yes	Read Only	Resource Type of the target Resource.

D.18.6 CRUDN behaviour

Table D-35 defines the CRUDN operations that are supported on the "oic.r.alertcollection" Resource Type.

Table D-35 – The CRUDN operations of the Resource with type "rt" = "oic.r.alertcollection".

Create	Read	Update	Delete	Notify
	get			observe

Annex E
(informative)

OIC 1.1 Resource Type definitions

E.1 List of Resource Type Definitions

Table E.1 contains the list of OIC 1.1 defined Core Resources that are referenced in this documentand so included herein to enable backwards compatibility. These definitions are only to be used when communicating with OIC 1.1 Devices where specifically referenced in this document.

Table E.1 – Alphabetized list of referenced OIC 1.1 Core Resources

Friendly Name (informative)	Resource Type (rt)	Clause
Collections	"oic.wk.col"	E.2
Discoverable Resources	"oic.wk.res"	E.3

E.2 OCF Collection

E.2.1 Introduction

OCF Collection Resource Type contains properties and links.

E.2.2 Wellknown URI

/CollectionResURI

E.2.3 Resource type

The Resource Type is defined as: "oic.wk.col".

E.2.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "OCF Collection",
    "version": "1.0",
    "license": {
      "name": "copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/CollectionResURI?if=oic.if.baseline" : {
      "get": {
```

```
        "description": "OCF Collection Resource Type contains properties and links.\nThe
oic.if.baseline interface exposes a representation of\nthe links and the properties of the
collection resource itself\nRetrieve on Baseline Interface\n",
        "parameters": [
          {"$ref": "#/parameters/interface-baseline"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt": ["oic.wk.col"],
                  "id": "unique_example_id",
                  "rts": [ "oic.r.switch.binary", "oic.r.airflow" ],
                  "rts-m": [ "oic.r.switch.binary" ],
                  "links": [
                    {
                      "href": "switch",
                      "rt":   ["oic.r.switch.binary"],
                      "if":   ["oic.if.a", "oic.if.baseline"],
                      "eps": [
                          {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                          {"ep": "coaps://[fe80::b1d6]:1122"},
                          {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                      ]
                    },
                    {
                      "href": "airFlow",
                      "rt":   ["oic.r.airflow"],
                      "if":   ["oic.if.a", "oic.if.baseline"],
                      "eps": [
                          {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                          {"ep": "coaps://[fe80::b1d6]:1122"},
                          {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                      ]
                    }
                  ]
                },
              "schema": { "$ref": "#/definitions/sbaseline" }
            }
        }
      },
      "post": {
        "description": "Update on Baseline Interface\n",
        "parameters": [
          {"$ref": "#/parameters/interface-baseline"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/sbaseline-update" }
          }
        ],
        "responses": {
            "200": {
              "description" : "",
              "schema": { "$ref": "#/definitions/sbaseline-update" }
            }
        }
      }
    },
    "/CollectionResURI?if=oic.if.b" : {
      "get": {
        "description": "OCF Collection Resource Type contains properties and links.\nThe oic.if.b
interfacce exposes a composite representation of the\nresources pointed to by the links\nRetrieve on
Batch Interface\n",
        "parameters": [
          {"$ref": "#/parameters/interface-b"}
        ],
        "responses": {
            "200": {
```

```
                "description" : "All targets returned OK status (HTTP 200 or CoAP 2.05 Content)",
                "x-example":
                  [
                    {
                      "href": "switch",
                      "rep":
                        {
                          "value": true
                        }
                    },
                    {
                      "href": "airFlow",
                      "rep":
                        {
                          "direction": "floor",
                          "speed":      3
                        }
                    }
                  ],
                "schema": { "$ref": "#/definitions/sbatch-retrieve" }
              },
              "404": {
                "description" : "One or more targets did not return an OK status, return a
representation containing returned properties from the targets that returned OK",
                "x-example":
                  [
                    {
                      "href": "switch",
                      "rep":
                        {
                          "value": true
                        }
                    }
                  ],
                "schema": { "$ref": "#/definitions/sbatch-retrieve" }
              }
          }
        },
        "post": {
          "description": "Update on Batch Interface\n",
          "parameters": [
            {"$ref": "#/parameters/interface-b"},
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": { "$ref": "#/definitions/sbatch-update" },
              "x-example":
                [
                  {
                    "href": "switch",
                    "rep":
                      {
                        "value": true
                      }
                  },
                  {
                    "href": "airFlow",
                    "rep":
                      {
                        "direction": "floor",
                        "speed": 3
                      }
                  }
                ]
            }
          ],
          "responses": {
              "200": {
                "description" : "all targets returned OK status (HTTP 200 or CoAP 2.04 Changed) return
a representation of the current state of all targets",
```



```
                "x-example":
                  [
                    {
                      "href": "switch",
                      "rep":
                        {
                          "value": true
                        }
                    },
                    {
                      "href": "airFlow",
                      "rep":
                        {
                          "direction": "demist",
                          "speed": 5
                        }
                    }
                  ],
                "schema": { "$ref": "#/definitions/sbatch-retrieve" }
              },
              "403": {
                "description" : "one or more targets did not return OK status; return a retrieve
representation of the current state of all targets in the batch",
                "x-example":
                  [
                    {
                      "href": "switch",
                      "rep":
                        {
                          "value": true
                        }
                    },
                    {
                      "href": "airFlow",
                      "rep":
                        {
                          "direction": "floor",
                          "speed": 3
                        }
                    }
                  ],
                "schema": { "$ref": "#/definitions/sbatch-retrieve" }
              }
          }
        }
      },
      "/CollectionResURI?if=oic.if.ll" : {
        "get": {
          "description": "OCF Collection Resource Type contains properties and links.\nThe oic.if.ll
interface exposes a representation of the links\nRetrieve on Link List Interface\n",
          "parameters": [
            {"$ref": "#/parameters/interface-ll"}
          ],
          "responses": {
              "200": {
                "description" : "",
                "x-example":
                  {
                    "links": [
                      {
                        "href": "switch",
                        "rt":   ["oic.r.switch.binary"],
                        "if":   ["oic.if.a", "oic.if.baseline"],
                        "eps": [
                          {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                          {"ep": "coaps://[fe80::b1d6]:1122"},
                          {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                        ]
                      },
                      {
                        "href": "airFlow",
```



```
                        "rt":   ["oic.r.airflow"],
                        "if":   ["oic.if.a", "oic.if.baseline"],
                        "eps": [
                          {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                          {"ep": "coaps://[fe80::b1d6]:1122"},
                          {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                        ]
                      }
                    ]
                  },
                "schema": { "$ref": "#/definitions/slinks" }
              }
          }
        }
      }
    },
    "parameters": {
      "interface-ll" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.ll"]
      },
      "interface-b" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.b"]
      },
      "interface-baseline" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.baseline"]
      },
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"]
      }
    },
    "definitions": {
      "sbaseline" : {
        "properties": {
          "links" : {
              "description": "A set of simple or individual OIC Links.",
              "items": {
              "$ref": "#/definitions/oic.oic-link"
              },
              "type": "array"
          },
          "n": {
            "type": "string",
            "description": "User friendly name of the collection"
          },
          "id": {
            "anyOf": [
              {
                "type": "integer",
                "description": "A number that is unique to that collection; like an ordinal number
that is not repeated"
              },
              {
                "type": "string",
                "description": "A unique string that could be a hash or similarly unique"
              },
              {
                "description": "An identifier formatted according to IETF RFC 4122.",
                "type":"string",
                "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
```



```
9]{12}$"
            }
          ],
          "description": "ID for the collection. Can be an value that is unique to the use context
or a UUIDv4"
        },
        "rt": {"$ref": "#/definitions/oic.core/properties/rt"},
              "rts": { "$ref": "#/definitions/oic.core/properties/rt"},
              "if": {
                "description": "The interface set supported by this resource",
                "items": {
                       "enum": ["oic.if.baseline",
                              "oic.if.ll",
                              "oic.if.b",
                              "oic.if.rw",
                              "oic.if.r",
                                            "oic.if.a",
                                            "oic.if.s"],
                                          "type": "string"
                                    },
                                    "minItems": 1,
                                    "type": "array"
                                }
      },
      "type" : "object"
    },
    "sbaseline-update": {
      "additionalProperties": true
    },
         "oic.core": {
                       "properties": {
                              "rt": {
                                    "description": "Resource Type of the Resource",
                                    "items": {
                                           "maxLength": 64,
                                           "type": "string"
                                    },
                                    "minItems": 1,
                                    "readOnly": true,
                                    "type": "array"
                              }
                       },
                       "type": "object"
                },
    "sbatch-retrieve" : {
      "title" : "Collection Batch Retrieve Format (auto merged)",
      "minItems" : 1,
      "items" : {
        "additionalProperties": true,
        "properties": {
          "href": {
            "description": "URI of the target resource relative assuming the collection URI as
anchor",
            "format": "uri",
            "maxLength": 256,
            "type": "string"
          },
          "rep": {
            "oneOf": [
              {
                "description": "The response payload from a single resource",
                "type": "object"
              },
              {
                "description": " The response payload from a collection (batch) resource",
                "items": {
                  "properties": {
                    "anchor": {
                      "description": "This is used to override the context URI e.g. override the URI
of the containing collection.",
                      "format": "uri",
```



```
                        "maxLength": 256,
                        "type": "string"
                      },
                      "di": {
                        "allOf": [
                          {
                            "description": "Format pattern according to IETF RFC 4122.",
                            "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
[a-fA-F0-9]{12}$",
                            "type": "string"
                          },
                          {
                            "description": "The device ID"
                          }
                        ]
                      },
                      "eps": {
                        "description": "the Endpoint information of the target Resource",
                        "items": {
                          "properties": {
                            "ep": {
                              "description": "Transport Protocol Suite + Endpoint Locator",
                              "format": "uri",
                              "type": "string"
                            },
                            "pri": {
                              "description": "The priority among multiple Endpoints",
                              "minimum": 1,
                              "type": "integer"
                            }
                          },
                          "type": "object"
                        },
                        "type": "array"
                      },
                      "href": {
                        "description": "This is the target URI, it can be specified as a Relative
Reference or fully-qualified URI.",
                        "format": "uri",
                        "maxLength": 256,
                        "type": "string"
                      },
                      "if": {
                        "description": "The interface set supported by this resource",
                        "items": {
                          "enum": [
                            "oic.if.baseline",
                            "oic.if.ll",
                            "oic.if.b",
                            "oic.if.rw",
                            "oic.if.r",
                            "oic.if.a",
                            "oic.if.s"
                          ],
                          "type": "string"
                        },
                        "minItems": 1,
                        "type": "array"
                      },
                      "ins": {
                        "description": "The instance identifier for this web link in an array of web
links - used in collections",
                        "type": "integer"
                      },
                      "p": {
                        "description": "Specifies the framework policies on the Resource referenced by
the target URI",
                        "properties": {
                          "bm": {
                            "description": "Specifies the framework policies on the Resource
referenced by the target URI for e.g. observable and discoverable",
```

```
                            "type": "integer"
                          }
                        },
                        "required": [
                          "bm"
                        ],
                        "type": "object"
                      },
                      "rel": {
                        "description": "The relation of the target URI referenced by the link to the
context URI",
                        "oneOf": [
                          {
                            "default": [
                              "hosts"
                            ],
                            "items": {
                              "maxLength": 64,
                              "type": "string"
                            },
                            "minItems": 1,
                            "type": "array"
                          },
                          {
                            "default": "hosts",
                            "maxLength": 64,
                            "type": "string"
                          }
                        ]
                      },
                      "rt": {
                        "description": "Resource Type of the Resource",
                        "items": {
                          "maxLength": 64,
                          "type": "string"
                        },
                        "minItems": 1,
                        "type": "array"
                      },
                      "title": {
                        "description": "A title for the link relation. Can be used by the UI to
provide a context.",
                        "maxLength": 64,
                        "type": "string"
                      },
                      "type": {
                        "default": "application/cbor",
                        "description": "A hint at the representation of the resource referenced by the
target URI. This represents the media types that are used for both accepting and emitting.",
                        "items": {
                          "maxLength": 64,
                          "type": "string"
                        },
                        "minItems": 1,
                        "type": "array"
                      }
                    },
                    "required": [
                      "href",
                      "rt",
                      "if"
                    ],
                    "type": "object"
                  },
                  "type": "array"
                }
              ]
            }
          },
          "required": [
            "href",
```



```
            "rep"
          ],
          "type": "object"
        },
        "type" : "array"
      },
      "sbatch-update" : {
        "title" : "Collection Batch Update Format (auto merged)",
        "minItems" : 1,
        "items" :  { "$ref": "#/definitions/oic.batch-update.item" },
        "type" :  "array"
      },
      "slinks" : {
        "type": "object",
        "properties": {
          "links": {
            "type" : "array",
            "items" : {
              "$ref": "#/definitions/oic.oic-link"
            }
          }
        }
      },
      "oic.batch-update.item" : {
        "additionalProperties": true,
        "description": "array of resource representations to apply to the batch collection, using href
to indicate which resource(s) in the batch to update. If the href property is empty, effectively
making the URI reference to the collection itself, the representation is to be applied to all
resources in the batch",
        "properties": {
          "href": {
            "description": "URI of the target resource relative assuming the collection URI as
anchor",
            "format": "uri",
            "maxLength": 256,
            "type": "string"
          },
          "rep": {
            "oneOf": [
              {
                "description": "The response payload from a single resource",
                "type": "object"
              },
              {
                "description": " The response payload from a collection (batch) resource",
                "items": {
                  "$ref": "#/definitions/oic.oic-link"
                },
                "type": "array"
              }
            ]
          }
        },
        "required": [
          "href",
          "rep"
        ],
        "type": "object"
      },
      "oic.oic-link" : {
        "properties": {
          "anchor": {
            "description": "This is used to override the context URI e.g. override the URI of the
containing collection.",
            "format": "uri",
            "maxLength": 256,
            "type": "string"
          },
          "di": {
            "description": "The Device ID formatted according to IETF RFC 4122.",
            "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
```



```
9]{12}$",
          "type": "string"
        },
        "eps": {
          "description": "the Endpoint information of the target Resource",
          "items": {
            "properties": {
              "ep": {
                "description": "Transport Protocol Suite + Endpoint Locator",
                "format": "uri",
                "type": "string"
              },
              "pri": {
                "description": "The priority among multiple Endpoints",
                "minimum": 1,
                "type": "integer"
              }
            },
            "type": "object"
          },
          "type": "array"
        },
        "href": {
          "description": "This is the target URI, it can be specified as a Relative Reference or
fully-qualified URI.",
          "format": "uri",
          "maxLength": 256,
          "type": "string"
        },
        "if": {
          "description": "The interface set supported by this resource",
          "items": {
            "enum": [
              "oic.if.baseline",
              "oic.if.ll",
              "oic.if.b",
              "oic.if.rw",
              "oic.if.r",
              "oic.if.a",
              "oic.if.s"
            ],
            "type": "string"
          },
          "minItems": 1,
          "type": "array"
        },
        "ins": {
          "description": "The instance identifier for this web link in an array of web links - used
in collections",
          "type": "integer"
        },
        "p": {
          "description": "Specifies the framework policies on the Resource referenced by the target
URI",
          "properties": {
            "bm": {
              "description": "Specifies the framework policies on the Resource referenced by the
target URI for e.g. observable and discoverable",
              "type": "integer"
            }
          },
          "required": [
            "bm"
          ],
          "type": "object"
        },
        "rel": {
          "description": "The relation of the target URI referenced by the link to the context URI",
          "oneOf": [
            {
              "default": [
```

```
                "hosts"
              ],
              "items": {
                "maxLength": 64,
                "type": "string"
              },
              "minItems": 1,
              "type": "array"
            },
            {
              "default": "hosts",
              "maxLength": 64,
              "type": "string"
            }
          ]
        },
        "rt": {
          "description": "Resource Type of the Resource",
          "items": {
            "maxLength": 64,
            "type": "string"
          },
          "minItems": 1,
          "type": "array"
        },
        "title": {
          "description": "A title for the link relation. Can be used by the UI to provide a
context.",
          "maxLength": 64,
          "type": "string"
        },
        "type": {
          "default": "application/cbor",
          "description": "A hint at the representation of the resource referenced by the target URI.
This represents the media types that are used for both accepting and emitting.",
          "items": {
            "maxLength": 64,
            "type": "string"
          },
          "minItems": 1,
          "type": "array"
        }
      },
      "required": [
        "href",
        "rt",
        "if"
      ],
      "type": "object"
    }
  }
}
```

E.2.5 Property definition

Table E.2 defines the Properties that are part of the "oic.wk.col" Resource Type

Table E.2 – The Property definitions of the Resource with type "rt" = "oic.wk.col"

Property name	Value type	Mandatory	Access mode	Description
rep	multiple types: see schema	Yes	Read Write	
href	string	Yes	Read Write	URI of the target resource relative assuming the collection URI as anchor.

rt	array: see schema		Read Only	Resource Type of the Resource.
links	array: see schema		Read Write	
if	array: see schema		Read Write	The interface set supported by this resource.
rts	multiple types: see schema		Read Write	
id	multiple types: see schema		Read Write	ID for the collection. Can be an value that is unique to the use context or a UUIDv4.
rt	multiple types: see schema		Read Write	
n	string		Read Write	User friendly name of the collection.
links	array: see schema		Read Write	A set of simple or individual OIC Links.
di	string	No	Read Write	The Device ID formatted according to IETF RFC 4122.
anchor	string	No	Read Write	This is used to override the context URI e.g. override the URI of the containing collection.
if	array: see schema	Yes	Read Write	The interface set supported by this resource.
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the link to the context URI.
eps	array: see schema	No	Read Write	the Endpoint information of the target Resource.
ins	integer	No	Read Write	The instance identifier for this web link in an array of web links - used in collections.
rt	array: see schema	Yes	Read Write	Resource Type of the Resource.
type	array: see schema	No	Read Write	A hint at the representation of the resource referenced by the target URI. This represents the media types that are used for both accepting and emitting.
title	string	No	Read Write	A title for the link relation. Can be used by the UI to provide a context.

href	string	Yes	Read Write	This is the target URI, it can be specified as a Relative Reference or fully-qualified URI.
p	object: see schema	No	Read Write	Specifies the framework policies on the Resource referenced by the target URI.
rep	multiple types: see schema	Yes	Read Write	
href	string	Yes	Read Write	URI of the target resource relative assuming the collection URI as anchor.

E.2.6 CRUDN behaviour

Table E.3 defines the CRUDN operations that are supported on the ['oic.wk.col'] Resource Type

Table E.3 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.col']

Create	Read	Update	Delete	Notify
	get	post		observe

E.3 Discoverable Resources

E.3.1 Introduction

List of discoverable resources.

E.3.2 Wellknown URI

/oic/res

E.3.3 Resource type

The Resource Type is defined as: "oic.wk.res"

E.3.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Discoverable Resources Link List interface",
    "version": "v1-20160622",
    "license": {
      "name": "copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2. Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
```

```
SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/oic/res?if=oic.if.ll" : {
      "get": {
        "description": "Link list representation of /oic/res; list of discoverable
resources\nRetrieve the discoverable resource set, link list interface\n",
        "parameters": [
          {"$ref": "#/parameters/interface-ll"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1",
                    "links": [
                      {
                        "href": "/humidity",
                        "rt":   ["oic.r.humidity"],
                        "if":   ["oic.if.s"],
                        "p":    {"bm": 3},
                        "eps": [
                          {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
                          {"ep": "coaps://[fe80::b1d6]:1122"},
                          {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
                        ]
                      },
                      {
                        "href": "/temperature",
                        "rt":   ["oic.r.temperature"],
                        "if":   ["oic.if.s"],
                        "p":    {"bm": 3},
                        "eps": [
                          {"ep": "coaps://[[2001:db8:a::123]:2222"}
                        ]
                      }
                    ]
                  }
                ],
              "schema": { "$ref": "#/definitions/slinklist" }
            }
        }
      }
    },
    "/oic/res?if=oic.if.baseline" : {
      "get": {
        "description": "Baseline representation of /oic/res; list of discoverable
resources\nRetrieve the discoverable resource set, baseline interface\n",
        "parameters": [
          {"$ref": "#/parameters/interface-baseline"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "rt": ["oic.wk.res"],
                    "if": ["oic.if.baseline", "oic.if.ll" ],
                    "di": "0685B960-736F-46F7-BEC0-9E6CBD61ADC1",
                    "links": [
                      {
                        "href": "/humidity",
                        "rt":   ["oic.r.humidity"],
                        "if":   ["oic.if.s"],
```

```
                          "p":    {"bm": 3},
                          "eps": [
                              {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
                              {"ep": "coaps://[fe80::b1d6]:1122"},
                              {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                          ]
                        },
                        {
                          "href": "/temperature",
                          "rt":   ["oic.r.temperature"],
                          "if":   ["oic.if.s"],
                          "p":    {"bm": 3},
                          "eps": [
                              {"ep": "coaps://[[2001:db8:a::123]:2222"}
                          ]
                        }
                      ]
                    }
                  ],
                "schema": { "$ref": "#/definitions/sbaseline" }
              }
          }
        }
    },
    "parameters": {
      "interface-ll" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.ll"]
      },
      "interface-baseline" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.baseline"]
      },
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.ll", "oic.if.baseline"]
      }
    },
    "definitions": {
      "slinklist" : {
        "type": "array",
        "items" : {
          "type": "object",
          "properties": {
            "di": {
              "description": "An identifier formatted according to IETF RFC 4122.",
              "type":"string",
              "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
              "readOnly": true
            },
            "links": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/oic.oic-link"
              }
            }
          }
        }
      },
      "sbaseline" : {
        "type": "array",
        "items" : {
          "type": "object",
```

```
          "properties": {
            "n": {
              "description": "Human friendly name",
              "maxLength": 64,
              "readOnly": true,
              "type": "string"
            },
            "rt": {
              "description": "Resource Type of the Resource",
              "items": {
                "maxLength": 64,
                "type": "string"
              },
              "minItems": 1,
              "readOnly": true,
              "type": "array"
            },
            "if": {
              "description": "The interface set supported by this resource",
              "items": {
                "enum": [
                  "oic.if.baseline",
                  "oic.if.ll"
                ],
                "type": "string"
              },
              "minItems": 1,
              "readOnly": true,
              "type": "array"
            },
            "di": {
              "description": "An identifier formatted according to IETF RFC 4122.",
              "type":"string",
              "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
              "readOnly": true
            },
            "mpro": {
              "readOnly": true,
              "description": "Supported messaging protocols",
              "type": "string",
              "maxLength": 64
            },
            "links": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/oic.oic-link"
              }
            }
          },
          "required": [
            "rt",
            "if",
            "links"
          ]
        }
      },
      "oic.oic-link": {
        "type": "object",
        "properties": {
          "anchor": {
            "description": "This is used to override the context URI e.g. override the URI of the
containing collection.",
            "format": "uri",
            "maxLength": 256,
            "type": "string"
          },
          "di": {
            "description": "The Device ID formatted according to IETF RFC 4122.",
            "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
```



```
            "type": "string"
          },
          "eps": {
            "description": "the Endpoint information of the target Resource",
            "items": {
              "properties": {
                "ep": {
                  "description": "Transport Protocol Suite + Endpoint Locator",
                  "format": "uri",
                  "type": "string"
                },
                "pri": {
                  "description": "The priority among multiple Endpoints",
                  "minimum": 1,
                  "type": "integer"
                }
              },
              "type": "object"
            },
            "type": "array"
          },
          "href": {
            "description": "This is the target URI, it can be specified as a Relative Reference or
fully-qualified URI.",
            "format": "uri",
            "maxLength": 256,
            "type": "string"
          },
          "if": {
            "description": "The interface set supported by this resource",
            "items": {
              "enum": [
                "oic.if.baseline",
                "oic.if.ll",
                "oic.if.b",
                "oic.if.rw",
                "oic.if.r",
                "oic.if.a",
                "oic.if.s"
              ],
              "type": "string"
            },
            "minItems": 1,
            "type": "array"
          },
          "ins": {
            "description": "The instance identifier for this web link in an array of web links - used
in collections",
            "type": "integer"
          },
          "p": {
            "description": "Specifies the framework policies on the Resource referenced by the target
URI",
            "properties": {
              "bm": {
                "description": "Specifies the framework policies on the Resource referenced by the
target URI for e.g. observable and discoverable",
                "type": "integer"
              }
            },
            "required": [
              "bm"
            ],
            "type": "object"
          },
          "rel": {
            "description": "The relation of the target URI referenced by the link to the context URI",
            "oneOf": [
              {
                "default": [
                  "hosts"
```

```
                ],
                "items": {
                  "maxLength": 64,
                  "type": "string"
                },
                "minItems": 1,
                "type": "array"
              },
              {
                "default": "hosts",
                "maxLength": 64,
                "type": "string"
              }
            ]
          },
          "rt": {
            "description": "Resource Type of the Resource",
            "items": {
              "maxLength": 64,
              "type": "string"
            },
            "minItems": 1,
            "type": "array"
          },
          "title": {
            "description": "A title for the link relation. Can be used by the UI to provide a
context.",
            "maxLength": 64,
            "type": "string"
          },
          "type": {
            "default": "application/cbor",
            "description": "A hint at the representation of the resource referenced by the target URI.
This represents the media types that are used for both accepting and emitting.",
            "items": {
              "maxLength": 64,
              "type": "string"
            },
            "minItems": 1,
            "type": "array"
          }
        },
        "required": ["href","rt","if"]
      }
    }
  }
```

E.3.5 Property definition

Table E.4 defines the Properties that are part of the "oic.wk.res" Resource Type.

Table E.4 – The Property definitions of the Resource with type "rt" = "oic.wk.res"

Property name	Value type	Mandatory	Access mode	Description
di	string		Read Only	An identifier formatted according to IETF RFC 4122.
links	array: see schema		Read Write	
links	array: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Only	The interface set supported by this resource
rt	array: see schema	Yes	Read Only	Resource Type of the Resource

n	string	No	Read Only	Human friendly name
di	string	No	Read Only	An identifier formatted according to IETF RFC 4122.
mpro	string	No	Read Only	Supported messaging protocols
ins	integer	No	Read Write	The instance identifier for this web link in an array of web links - used in collections
type	array: see schema	No	Read Write	A hint at the representation of the resource referenced by the target URI. This represents the media types that are used for both accepting and emitting.
eps	array: see schema	No	Read Write	the Endpoint information of the target Resource
if	array: see schema	Yes	Read Write	The interface set supported by this resource
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the link to the context URI
rt	array: see schema	Yes	Read Write	Resource Type of the Resource
anchor	string	No	Read Write	This is used to override the context URI e.g. override the URI of the containing collection.
di	string	No	Read Write	The Device ID formatted according to IETF RFC 4122.
href	string	Yes	Read Write	This is the target URI, it can be specified as a Relative Reference or fully-qualified URI.
title	string	No	Read Write	A title for the link relation. Can be used by the UI to provide a context.
p	object: see schema	No	Read Write	Specifies the framework policies on the Resource referenced by the target URI

E.3.6 CRUDN behaviour

Table E.5defines the CRUDN operations that are supported on the None Resource Type

Table E.5 – The CRUDN operations of the Resource with type 'rt' = ['oic.wk.res']

Create	Read	Update	Delete	Notify
	get			observe

Annex F
(informative)

OpenAPI 2.0 Schema Extension

F.1 OpenAPI 2.0 Schema Reference

OpenAPI 2.0 does not support allOf and anyOf JSON schema valiation constructs; this document has extended the underlying OpenAPI 2.0 schema to enable these, all OpenAPI 2.0 files are valid against the extended schema. Reference the following location for a copy of the extended schema:

https://github.com/openconnectivityfoundation/OCFswagger2.0-schema

F.2 OpenAPI 2.0 Introspection empty file

Reference the following location for a copy of an empty OpenAPI 2.0 file:

https://github.com/openconnectivityfoundation/DeviceBuilder/blob/master/examples/introspection-empty.txt

