

**OCF 2.3 – IDD Clarification – CTWG CR 2655**

## Legal Disclaimer

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

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

Copyright © 2018 Open Connectivity Foundation, Inc. All rights reserved.

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

\*\*\*\*\* **Change #1 (changed text)** \*\*\*\*\*

## 1.1 Introspection

### 1.1.1 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 section 7.8.2.1.2).

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 18. The IDD is described as a Swagger2.0 in JSON format file. The text below contains also Swagger2.0 terms, like paths, methods etc. The Swagger2.0 file shall contain the description of the Resources as defined below:

- 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 3<sup>rd</sup> party defined or optional Properties are implemented.
  - Resource with Resource Type: “oic.wk.introspection”.
  - Resources that handle Wi-Fi Easy Setup, see OCF Wi-Fi Easy Setup specification.
- The following Resources shall be included in the IDD when optional or 3<sup>rd</sup> party defined Properties are implemented:
  - Resources with type: “oic.wk.p” and “oic.wk.d” (e.g. discovery related Resources).
  - Security Virtual Resources from the OCF Security specification.
- When the Device does not expose instances of Vertical Resource Types, and does not have any 3<sup>rd</sup> party defined Resources (see section 7.8.4.4), and does not need to include Resources in the

IDD due to other clauses in this section, then the IDD shall be an empty Swagger2.0 file. An example of an empty Swagger2.0 file can be found in found in Annex G.2:

- All other Resources shall be listed in the IDD.
- Per Resource the IDD shall include:
  - All implemented methods
    - For an OCF defined Resource, only the methods that are standardized are allowed to exist in the IDD. The supported methods shall comply with the listed OCF Interfaces. For example, if an OCF Interface is listed that allows updates, then the update method shall be listed. It is not allowed to have methods listed for OCF defined Resources that does not have this method defined.
  - 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 swagger schema object as defined in the parameters section.
    - The swagger2.0 schema object shall comply with:
      - The schemas shall be fully resolved, e.g. no references shall exist outside the swagger 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 section 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 Swagger2.0.
      - 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 Swagger2.0 schema object. The Swagger2.0 schema with oneOf and anyOf constructs can be found in Annex G.1.

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 26. Example usage of oneOf JSON schema:

```

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

```

**Figure 1. 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 swagger 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 (swagger specification), for example 3.0.

A Device shall support one Resource with a Resource Type of "oic.wk.introspection" as defined in Table 36. 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 Swagger2.0 fields. See OpenAPI specification (swagger specification). An example of an empty Swagger2.0 file can be found in Annex G.2.

**Table 1. 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 37 defines "oic.wk.introspection" Resource Type.

**Table 2. "oic.wk.introspection" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
urlInfo	urlInfo	array			R	yes	array of objects
url	url	string	uri		R	yes	URL to the hosted payload

<b>protocol</b>	protocol	string	enum		R	yes	Protocol definition to retrieve the Introspection Device Data from the url.
<b>content-type</b>	content-type	string	enum		R	no	content type of the url.
<b>version</b>	version	integer	enum		R	no	Version of the Introspection protocol, indicates which rules are applied on the Introspection Device Data regarding the content of the RAML file. Current value is 1.

DRAFT