OCF Easy Setup Specification 2.2.8

VERSION 2.2.8 | June 2025

CONTACT **admin@openconnectivity.org**

Legal Disclaimer

CONTENTS

Figures

Tables

Introduction

This document, and all the other parts associated with this document, were developed in response to worldwide demand for smart home focused Internet of Things (IoT) devices, such as appliances, door locks, security cameras, sensors, and actuators; these to be modelled and securely controlled, locally and remotely, over an IP network.

While some inter-device communication existed, no universal language had been developed for the IoT. Device makers instead had to choose between disparate frameworks, limiting their market share, or developing across multiple ecosystems, increasing their costs. The burden then falls on end users to determine whether the products they want are compatible with the ecosystem they bought into, or find ways to integrate their devices into their network, and try to solve interoperability issues on their own.

In addition to the smart home, IoT deployments in commercial environments are hampered by a lack of security. This issue can be avoided by having a secure IoT communication framework, which this standard solves.

The goal of these documents is then to connect the next 25 billion devices for the IoT, providing secure and reliable device discovery and connectivity across multiple OSs and platforms. There are multiple proposals and forums driving different approaches, but no single solution addresses the majority of key requirements. This document and the associated parts enable industry consolidation around a common, secure, interoperable approach.

The OCF specification suite is made up of nineteen discrete documents, the documents fall into logical groupings as described herein:

- Core framework
  - Core Specification
  - Security Specification
  - Onboarding Tool Specification
- Bridging framework and bridges
  - Bridging Specification
  - Resource to Alljoyn Interface Mapping Specification
  - OCF Resource to oneM2M Resource Mapping Specification
  - OCF Resource to BLE Mapping Specification
  - OCF Resource to EnOcean Mapping Specification
  - OCF Resource to LWM2M Mapping Specification
  - OCF Resource to UPlus Mapping Specification
  - OCF Resource to Zigbee Cluster Mapping Specification
  - OCF Resource to Z-Wave Mapping Specification
- Resource and Device models
  - Resource Type Specification
  - Device Specification
- Core framework extensions
  - Easy Setup Specification
  - Core Optional Specification
- OCF Cloud
  - Cloud API for Cloud Services Specification

214 – Device to Cloud Services Specification
215 – Cloud Security Specification

Easy Setup Specification

Scope

This document defines functional extensions to the capabilities defined in ISO/IEC 30118-1 to meet the requirements of Easy Setup. It specifies new Resource Types to enable the functionality and any extensions to the existing capabilities defined in ISO/IEC 30118-1.

Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 30118-1 Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 1: Core specification
https://www.iso.org/standard/53238.html
Latest version available at: https://openconnectivity.org/specs/OCF_Core_Specification.pdf

ISO/IEC 30118-2 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 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

IEEE 802.11, IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, December 2016
https://standards.ieee.org/findstds/standard/802.11-2016.html

IETF RFC 5646, *Tags for Identifying Languages*, September 2009
https://www.rfc-editor.org/info/rfc5646

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

GSMA RSP Technical Specification, Version 2.2.2, June 2020
https://www.gsma.com/esim/wp-content/uploads/2020/06/SGP.22-v2.2.2.pdf

Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1 and 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
Easy Setup
process of configuring an *Enrollee* (3.1.3) using a *Mediator* (3.1.5) by transferring of essential information to the *Enrollee* (3.1.3)

3.1.2
Easy Setup Enrollment
step during Easy Setup in which the *Enrollee* (3.1.3) is contacted by the *Mediator* (3.1.5) to configure the *Enroller's* (3.1.4) information by means of accessing *Easy Setup* (3.1.1) Resources

3.1.3
Enrollee
device that needs to be configured and connected. E.g. Air-conditioner, Printer

3.1.4
Enroller
target network entity to which the *Enrollee* (3.1.3) connects. E.g. Wi-Fi AP

3.1.5
Mediator
logical function that enables the *Enrollee* (3.1.3) to connect to the target network (i.e. *Enroller* (3.1.4)

Note 1 to Entry: The Mediator transfers configuration information to the Enrollee. E.g. Mobile Phone

3.1.6
Activation Code
information used by an end user to request the download of an *eSIM Profile* (3.1.8) from an *SM-DP+* (3.1.11) server as defined in the GSMA RSP Technical Specification

3.1.7
Local Profile Assistant (LPA)
functional element in the device or in the eUICC that provides *Remote SIM Provisioning* (3.1.9) features to the device as defined in the GSMA RSP Technical Specification

3.1.8
eSIM Profile
combination of data and applications to be provisioned on an eUICC for the purpose of providing service.

Note 1 to Entry: eSIM Profile is considered as the Profile defined in the GSMA RSP Technical Specification

3.1.9
Remote SIM Provisioning (RSP)
downloading, installing, enabling, disabling, and deleting of an *eSIM Profile* (3.1.8) on an eUICC as defined in the GSMA RSP Technical Specification

3.1.10
Subscription
commercial relationship between an end user and a service provider as defined in the GSMA RSP Technical Specification

3.1.11
Subscription Manager Data Preparation+ (SM-DP+)
eSIM Profile (3.1.8) preparation server which securely downloads *eSIM Profile* (3.1.8) to the *LPA* (3.1.7) of the respective eUICC in the device as defined in the GSMA RSP Technical Specification

3.1.12
Easy Setup Mode
mode that enables OCF setup and configuration to an IoT Device

3.1.13
eSIM Easy Setup Mode
mode that enables cellular network setting and configuration of *Remote SIM Provisioning* (3.1.9)

3.2 Symbols and abbreviated terms

CID	Company Identifier (ID)
eSIM	Embedded Subscriber Identification Module
eUICC	Embedded Universal Integrated Circuit Card
IE	Information Element
SIM	Subscriber Identification Module
Soft AP	Software Enabled Access Point
TLV	type-length-value

Document conventions and organization

4.1 Conventions

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

In this document, to be consistent with the IETF usages for RESTful operations, the RESTful operation words CRUDN, CREATE, RETRIVE, UPDATE, DELETE, and NOTIFY will have all letters capitalized. 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 document has no effect on the implementation's operation and does not produce any error conditions. Backward compatibility may require that a feature is implemented and functions as specified but it shall never be used by implementations compliant with this 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.

Overview

5.1 Introduction

This document describes a way to setup and configure a new Device, using an already configured Device or onboarding tool.

The described setup and configure mechanism is optional and other mechanisms are allowed to be used.

Specifically, this method allows the transferring of essential information to the new Device, which includes:

– Local network connection information, e.g. in case of Wi-Fi it will be Wi-Fi access point information.

– Device Configuration: Additional Device configuration information.

Easy Setup can be enhanced in future by incorporating other suitable technologies.

Annex A specifies the Resource Type definitions using the schema defined in the OpenAPI specification as the API definition language that shall be followed by a Device realizing the Resources specified in this document.

5.2 Architecture

Figure 1 shows the deployment architectural approach.

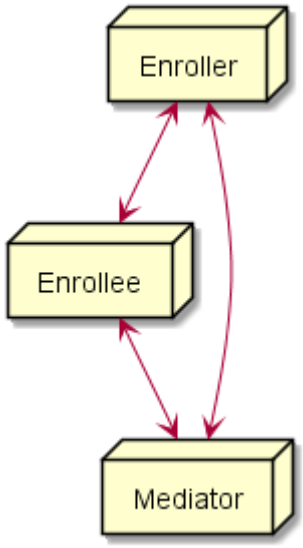


Figure 1 – Easy Setup deployment architecture

Easy Setup defines the following roles: Enrollee, Enroller, and Mediator. Please refer to clause 3 for the definitions thereof.

5.3 Example scenario

The following scenario presents a typical setup case.

The configuration information and steps taken may vary depending on the Device's type and status.

1) The Enrollee enters Easy Setup mode (when the Device is unboxed for the first time, it may be in this mode by default).
2) The Mediator discovers and connects to the Enrollee.
3) The Mediator performs Security Provisioning of the Enrollee.
4) The Mediator transmits Wi-Fi Setting Information to the Enrollee.
5) Using the information received from the Mediator, the Enrollee connects to the Enroller (Wi-Fi AP).

Easy setup overview

6.1 Introduction

Devices capable of Easy Setup shall support the following Resource Types.

1) EasySetup Resource Type
2) WiFiConf Resource Type
3) DevConf Resource Type

Instances of these Resources Type (Resources) shall be excluded in the IDD for the Introspection Resource (see clause 11.4 in ISO/IEC 30118-1).

The EasySetup Resource Type is a Collection Resource and shall contain Links to instances of at least WiFiConf and DevConf. A vendor may add links to other Resource Types. The relationship between the EasySetup Resource Type and linked Resources is shown in Figure 2.

NOTE The EasySetup Resource Type supports the batch Interface ("oic.if.b") which allows for efficient data delivery with a single request rather than multiple requests to each linked Resource.

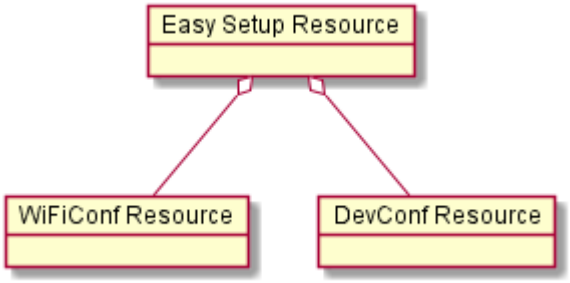


Figure 2 – Easy Setup Resource Types

6.2 EasySetup Resource

6.2.1 Overview

The EasySetup Resource stores useful information including current status of Enrollee and last error code which was produced in the process of Easy Setup.

6.2.2 Resource

The Easy Setup Resource Type is as defined in Table 1.

412 **Table 1 – EasySetup Resource Type**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/EasySetupResURI	EasySetup	oic.r.easysetup, oic.wk.col	oic.if.baseline, oic.if.ll, oic.if.b	Top level Resource for Easy Setup. Indicates easy setup status. The Resource properties exposed are listed in Table 2.	N/A

413

414 Table 2 defines the details for the "oic.r.easysetup" Resource Type. Complete details are
415 provided in annex A.3.

416 **Table 2 – "oic.r.easysetup" Resource Type definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Easy Setup Provisioning Status	ps	integer	enum	N/A	R	Yes	Easy setup provisioning status of the Device 0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF
Last Error Code	lec	integer	enum	N/A	R	Yes	Indicates a failure reason if it fails to connect to Enroller 0: No error, 1: Given SSID is not found, 2: Wi-Fi password is wrong, 3: IP address is not allocated, 4: NO internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown error.
Connect	cn	array of integer	N/A	N/A	RW	Yes	Array of connection types to trigger Enrollee to initiate connection: 1: Wi-Fi,

							2: Other transport to be added in a future (e.g. BLE))
Links	links	array	N/A	N/A	R	Yes	Array of links that are WiFiConf and DevConf Resource.

Enrollee shall set the following as default values (for example, when Device is unboxed first time):

– "ps" equal to 0.

– "lec" equal to 0.

– "cn" equal to an empty array.

6.3 WiFiConf Resource Type

6.3.1 Introduction

The WiFiConf Resource Type stores information to help an Enrollee to connect to an existing Wi-Fi Access Point.

6.3.2 Resource Type

The WiFiConf Resource Type is as defined in Table 3.

Table 3 – WiFiConf Resource Type

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/WiFiConfResURI	WiFiConf	oic.r.wificonf	oic.if.baseline, oic.if.rw	Contains Wi-Fi related properties The Resource properties exposed are listed in Table 4.	N/A

Table 4 defines the details for the "oic.r.wificonf" Resource Type. Complete details are provided in annex A.4.

Table 4 – "oic.r.wificonf" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Supported Wi-Fi Mode Type	swmt	array of string	enum	N/A	R	Yes	Supported Wi-Fi modes by Enrollee. Can be multiple. ("A", "B", "G","N", "AC")
Supported Wi-Fi Frequency	swf	array of string	Refer to description for valid values.	N/A	R	Yes	Supported Wi-Fi frequencies by Enrollee. Can be multiple. ("2.4G", "5G")
Target Network Name	tnn	string	N/A	N/A	RW	Yes	Target network name (SSID of Wi-Fi AP i.e. enroller)
Credential	cd	string	N/A	N/A	RW	No	Credential information of Wi-Fi AP (Password

							used to connect to enroller).
Wi-Fi Auth Type	wat	string	enum	N/A	RW	Yes	Wi-Fi auth type ("None", "WEP", "WPA_PSK", "WPA2_PSK")
Wi-Fi Encryption Type	wet	string	enum	N/A	RW	Yes	Wi-Fi encryption type ("None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES")
Supported Wi-Fi Auth Type	swat	array of string	enum	N/A	R	Yes	Supported Wi-Fi Auth types. Can be multiple. ("None", "WEP", "WPA_PSK", "WPA2_PSK")
Supported Wi-Fi Encryption Type	swet	array of string	enum	N/A	R	Yes	Supported Wi-Fi Encryption types. Can be multiple. ("None", "WEP-64", "WEP_128", "TKIP", "AES", "TKIP_AES")

6.4 DevConf Resource Type

6.4.1 Introduction

The DevConf Resource Type stores Device configuration information required in Wi-Fi Easy Setup.

6.4.2 Resource Type

The DevConf Resource Type is as defined in Table 5

Table 5 – DevConf Resource Type

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/DevConfResURI	DevConf	oic.r.devconf	oic.if.baseline, "oic.if.r"	Stores device configuration information required in Easy Setup process The Resource properties exposed are listed in Table 6.	N/A

Table 6 defines the details for the "oic.r.devconf" Resource Type. Complete details are provided in annex A.2.

Table 6 – "oic.r.devconf" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Device Name	dn	one of: string or	N/A	N/A	R	Yes	Indicates a pre-configured device name in language indicated by "dl" in "/oic/con". or

		array of object					An array of objects where each object has a language field (containing an IETF RFC 5646 language tag) and a value field containing the pre-configured device name in the indicated language. The pre-configured device name is presented by enrollee to mediator during easy-setup process.

eSIM Easy Setup Overview

7.1 Introduction

eSIM Easy Setup describes a way to setup cellular network setting and to configure Remote SIM Provisioning to an OCF Device with an eUICC.

If the Enrollee has no IP connectivity, the mechanism defined in clause 6 may be used to connect the Enrollee to the Mediator's Soft AP for IP connectivity. This method allows transferring subscription related information between an Enrollee and a Mediator, which includes:

– Device and eUICC information, used to provide cellular plans to an end user

– Subscription information, comprising, e.g. Activation Code

– Progress information, indicating the status of the eSIM Easy Setup

7.2 Architecture

Figure 3 shows the deployment architectural approach.

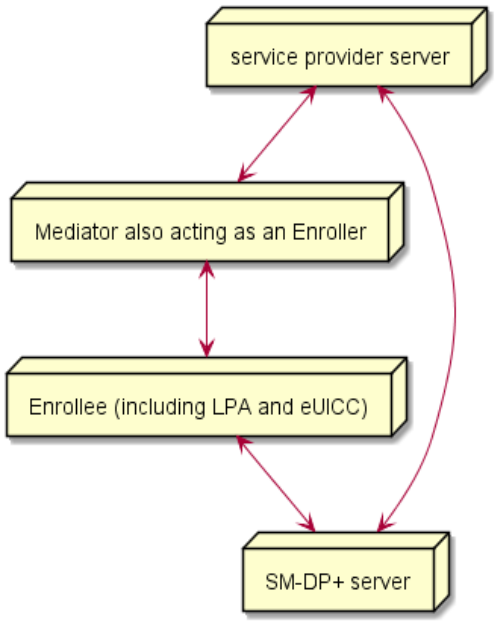


Figure 3 – eSIM Easy Setup deployment architecture

eSIM Easy Setup defines the following roles: Enrollee, Enroller, Mediator, SM-DP+ server, and service provider server. Enrollee to support eSIM Easy Setup includes both the LPA and the eUICC. LPA acts as a module interacting with the OCF Server and the eUICC in the Enrollee.

7.3 Example Scenario

Figure 4 presents a typical eSIM Easy Setup case. The configuration information and steps taken may vary depending on the Device's type and status.

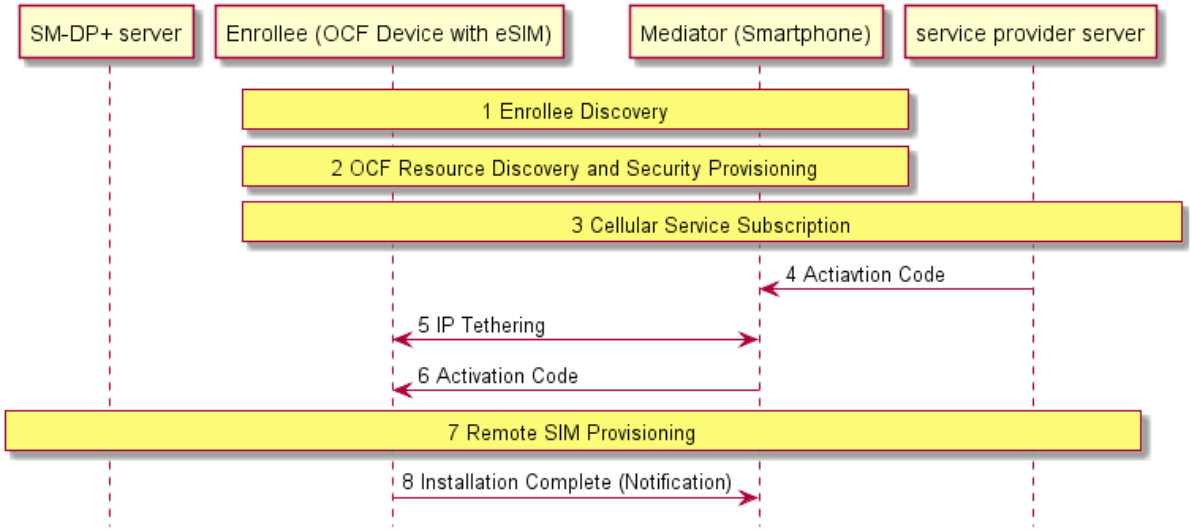


Figure 4 – eSIM Easy Setup example scenario

1. When an Enrollee (e.g. OCF Device with eSIM) is unboxed for the first time, the Enrollee creates SoftAP to make it discoverable. A Mediator (e.g. smartphone) discovers and connects to the Enrollee.
2. The Mediator discovers OCF Resources of the Enrollee and performs Security Provisioning (e.g. Ownership Transfer) of the Enrollee. If eSIM Easy Setup Resources are found, the Mediator may enter eSIM Easy Setup Mode as a default and displays a certain menu (e.g. activate cellular plan) on the screen.
3. An end user enters to buy a cellular plan (e.g. when the user clicks the button) for its Enrollee. The Enrollee may deliver its Device and eUICC information to the Mediator so that the Mediator forwards that information to a service provider server. Based on the information, the service provider provides cellular plans to select.
4. Once the end user finishes the contract on his or her cellular plan, the service provider server sends an Activation Code to the Mediator.
5. The Mediator transmits its Wi-Fi Setting Information to the Enrollee. Using the Wi-Fi Setting Information received from the Mediator, the Enrollee connects to the Mediator which is acting as an Enroller (i.e. IP tethering).
6. The Mediator now transmits the Activation Code to the Enrollee
7. Via the Mediator's IP network, Enrollee sends Activation Code to a SM-DP+ server. As a return, Enrollee downloads an eSIM Profile from the SM-DP+ server, and then installs the eSIM Profile onto the eUICC in the Enrollee. While downloading the eSIM Profile, any progress information required to display to the end user is notified to the Mediator.

8. The Enrollee notifies to the Mediator once the eSIM Profile installation is completed. The Enrollee connects to the cellular network directly. The Enrollee and The Mediator disconnect its local network connection (i.e. IP tethering) if necessary.

NOTE OCF defines connectivity-agnostic protocol. Figure 4 used Wi-Fi for IP tethering for the purpose to illustrate End-to-End on device activation procedure.

7.4 eSIM Easy Setup Resource Model

7.4.1 Introduction

Devices capable of eSIM Easy Setup shall support the following Resource Types.

1) eSIMEasySetup Resource Type
2) RSPCapability Resource Type
3) RSPConf Resource Type

The eSIMEasySetup Resource Type is a Collection Resource and shall contain Links to instances of at least RSPCapability Resource and RSPConf Resource. A vendor may add links to other Resources.

The relationship between the eSIMEasySetup Resource Type and linked Resources is shown in Figure 5.

NOTE The eSIMEasySetup Resource Type supports the batch Interface ("oic.if.b") which allows for efficient data delivery with a single request rather than multiple requests to each linked Resource.

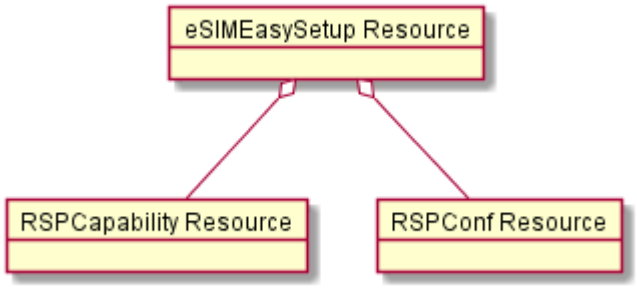


Figure 5 – eSIM Easy Setup Resource Types

7.5 eSIMEasySetup Resource Type

7.5.1 Introduction

The eSIMEasySetup Resource Type stores useful information including Remote SIM Provisioning (RSP) status, and RSP last error code which was produced in the process of eSIM Easy Setup.

7.5.2 Resource Type Definition

The eSIMEasySetup Resource Type is as defined in Table 7.

518 **Table 7 – eSIMEasySetup Resource Type**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/eSIMEasySetupResURI	eSIMEasySetup	oic.r.esimeasysetup	oic.if.baseline, oic.if.ll, oic.if.b	Top level Resource for eSIM Easy Setup. Indicates eSIM Easy Setup status. The Resource properties exposed are listed in Table 8.	N/A

519 Table 8 defines the details for the "oic.r.esimeasysetup" Resource Type. Complete details are
520 provided in annex A.5.

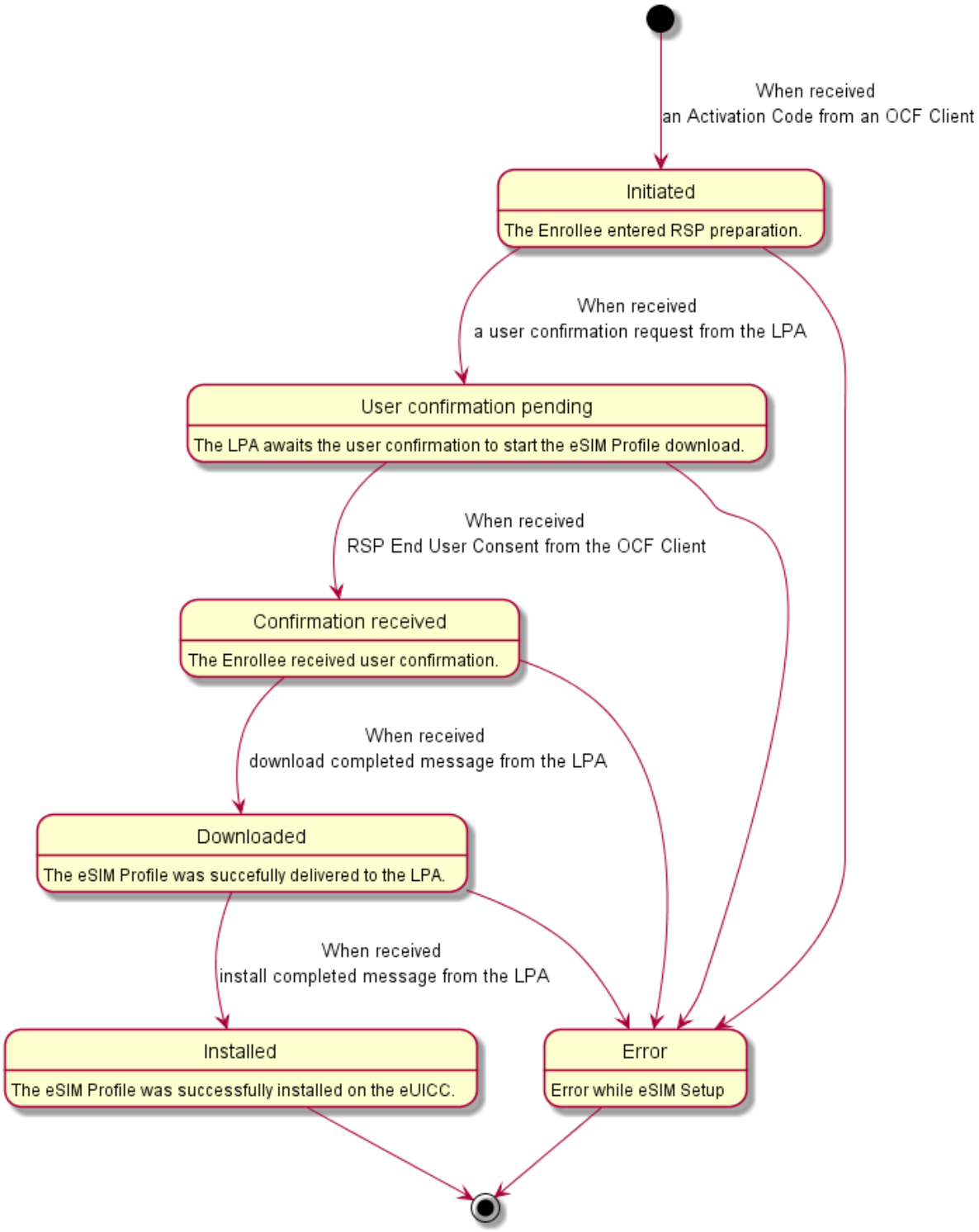
521 **Table 8 – "oic.r.esimeasysetup" Resource Type Definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
RSP Procedure Status	ps	string	enum	N/A	R	Yes	Steps in Remote SIM Provisioning. ("Undefined", "Initiated", "User confirmation pending", "Confirmation received", "Downloaded", "Installed", "Error")
RSP Last Error Reason	ler	string	N/A	N/A	R	Yes	Error Reason returned during eSIM Easy Setup. It indicates where it occurred. (e.g.,ES9+.GetBoundProfilePackage(Fail), ES10b.LoadBoundProfilePackage(Fail))
RSP Last Error Code	lec	string	N/A	N/A	R	Yes	Error Code returned during eSIM Easy Setup. It indicates why it occurred. (e.g., "8.8.1–3.8", "7", "6A 80") See more details in the Table X4
RSP Last Error Description	led	string	N/A	N/A	R	No	Optional error description returned during eSIM Easy Setup. (e.g., Invalid SM-DP+ Address)
RSP End User Consent	euc	string	enum	N/A	RW	Yes	End User Consent for RSP ("Undefined", "Timeout", "Download Reject", "Download Postponed", "Download OK", "Download and Enable OK")
Links	links	array	N/A	N/A	R	Yes	Array of web links that are RSPCapability Resource and RSPConf Resource

522 Enrollee shall set the following as default values (for example, when a Device is unboxed the first
523 time):

524 – "ps" equal to "Undefined".
525 – "ler" equal to an empty string.
526 – "lec" equal to an empty string.
527 – "led" equal to an empty string if "led" is presented.
528 – "euc" equal to "Undefined".

529 Figure 6 shows the RSP Procedure Status transition.



530

531 **Figure 6 – RSP Procedure Status Transition**

532 LPA-returned RSP procedure message to the OCF Server is out of scope in this document.
533 However, when LPA receives value(s) indicated in Table 9, the Server changes the RSP Procedure

Status ("ps") value, and shall send NOTIFICATION on any observe transaction(s) that may exist for the RSP Procedure Status ("ps") value change(s).

Table 9 – GSMA RSP procedure mapping to the OCF RSP Procedure Status

LPA received Value while RSP procedure	Mapping OCF RSP Procedure Status Property
ES9+.AuthenticateClient(Success)	User confirmation pending
ES9+.GetBoundProfilePackage(Success)	Downloaded
ES9+.HandleNotification(Success)	Installed
See Table X4	Error

Table 10 shows the example of error messages the LPA could receive while RSP procedure. Enrollee shall notify LPA-received error message to the Mediator.

Table 10 – Example of LPA received Error Message during RSP Procedure

Last Error Reason	Last Error Code	Last Error Description
ES9+.InitiateAuthentication(Fail)	8.8.1–3.8	Invalid SM-DP+ Address
ES9+.AuthenticateClient(Fail)	8.2.6–3.8	MatchingID is refused
ES9+.AuthenticateClient (Fail)	8.2–1.2	Profile has not yet been released
ES9+.AuthenticateClient(Fail)	8.8.5–4.10	The download order has expired
ES10b.PrepareDownload(Fail)	1	invalid certificate
ES10b.PrepareDownload(Fail)	2	invalid signature
ES9+.GetBoundProfilePackage(Fail)	8.2.7–2.2	Confirmation Code is missing
ES9+.GetBoundProfilePackage(Fail)	8.2.7–3.8	Confirmation Code is refused
ES10b.LoadBoundProfilePackage(Fail)	6A 80	Incorrect values in command data
ES10b.LoadBoundProfilePackage(Fail)	69 85	Conditions of use not satisfied (wrong TLV in Bound Profile Package)

7.6 RSPCapability Resource Type

7.6.1 Introduction

RSPCapability Resource Type stores information to help a service provider to provide appropriate cellular plans to an end user.

7.6.2 Resource Type Definition

The RSPCapability Resource Type is as defined in Table 11.

Table 11 – RSPCapability Resource Type

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/RSPCapabilityResURI	RSPCapability	oic.r.rspcapability	oic.if.baseline, oic.if.r	Contains eUICC and/or device configuration information required in eSIM Easy Setup process.	N/A

				The Resource properties exposed are listed in Table 12.	

Table 12 defines the details for the "oic.r.rspcapability" Resource Type. Complete details are provided in annex A.6.

Table 12 – "oic.r.rspcapability" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
eUICC Information	euiccinfo	string	Max.1024 octets	N/A	R	Yes	eUICC information used for the eSIM Profile download and installation procedure. Refers to "EUICCInfo2" defined in the GSMA RSP Technical Specification Annex H. This value type shall be encoded as Major Type 2.
Device Information for RSP	deviceinfo	string	Max.128 octets	N/A	R	Yes	Device information used for the eSIM Profile download and installation procedure. Refers to "DeviceInfo" defined in the GSMA RSP Technical Specification Annex H. This value type shall be encoded as Major Type 2.

7.7 RSPConf Resource Type

7.7.1 Introduction

RSPConf Resource Type stores the information used to download and install an eSIM Profile to an eSIM capable OCF device.

7.7.2 Resource Type Definition

The RSPConf Resource Type is as defined in Table 13.

Table 13 – RSPConf Resource Type

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example /RSPConfRes URI	RSPConf	oic.r.rspconf	oic.if.baseline, oic.if.rw	Contains Properties used to download and install an eSIM Profile. The Resource Properties exposed are listed in Table 14.	N/A

Table 14 defines the details for the "oic.r.rspconf" Resource Type. Complete details are provided in annex A.7.

559 **Table 14 – "oic.r.rspconf" Resource Type Definition**

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Activation Code	ac	string	Max.256 characters	N/A	RW	Yes	The information needed to provision an eSIM device. Comprises SM-DP+ server FQDN and Activation Code Token binding to a specific subscription as defined by the GSMA RSP Technical Specification
eSIM Profile Metadata	pm	string	Max. 2048 octets	N/A	R	Yes	Refers to "ProfileInfo" in the GSMA RSP Technical Specification Annex H. This value type shall be encoded as Major Type 2.
Confirmation Code	cc	string	N/A	N/A	RW	No	A code entered by an end user required by the SM-DP+ to confirm the download and installation of an eSIM Profile. The Confirmation Code is provided from a service provider to the end user.
Confirmation Code Required	ccr	boolean	N/A	N/A	R	Yes	Indicates whether a Confirmation Code is required. Set to "true" if Confirmation Code is required and required a user to enter Confirmation Code

560

561

Network and connectivity

Both the Mediator and Enrollee communicate via a common connectivity (e.g. Wi-Fi).

If using Wi-Fi for Easy Setup then the Enrollee shall have capability to act as a Soft AP. If an Enrollee uses IP tethering via Wi-Fi for eSIM Easy Setup, the Mediator shall have the capability to act as a SoftAP. A Soft AP shall support the access point requirements defined by IEEE 802.11.

Once the eSIMEasySetup procedure is completed, the IP connection (i.e., IP tethering) between an Enroller and an Enrollee should be destroyed.

Functional interactions

9.1 Onboarding, Provisioning and Configuration

The Mediator may be present as a standalone function or in conjunction with other functions or services such as AMS as part of an OBT (Onboarding Tool); please refer to the ISO/IEC 30118-2.

9.2 Resource discovery

The Mediator connects to the Enrollee via a mutually supported connection.

When in Easy Setup phase, if using Wi-Fi as the connectivity between the Enrollee and the Mediator then the Enrollee shall make itself discoverable as a Soft AP. The Soft AP has additional availability constraints which are documented in ISO/IEC 30118-2.

9.3 Retrieving and updating Easy Setup Resources

The Enrollee shall expose Easy Setup Resources (i.e. EasySetup Resource, eSIMEasySetup Resource) such that a Mediator is able to discover them using standard Resource discovery methods (i.e. via a RETRIEVE on /oic/res); see ISO/IEC 30118-1, clause 11.3.

Easy Setup Resources shall expose only secure Endpoints (e.g. CoAPS); see ISO/IEC 30118-1, clause 10.

The Mediator may RETRIEVE a Resource within the Easy Setup Collection or the Collection itself to check the Enrollee’s status at any stage of Easy Setup. This applies only when the Enrollee and the Mediator are on a common network.

The Mediator may UPDATE Resource Property(-ies) on the Enrollee. Upon receipt of the request from the Mediator the Enrollee shall update its current Resource Property Values, and shall perform any required action. For example, if the "cn" Property of "EasySetup" Resource is updated by the Mediator, to indicate connection to Wi-Fi, the Enrollee shall start the connection to Enroller.

For details of Easy Setup Resources refer to clause 6.

9.4 Error handling

9.4.1 Easy Setup error handling

The "lec" Property of the EasySetup Resource (i.e. "oic.r.easysetup") is used to indicate the error that occurred in the Easy Setup process while trying to connect to the Enroller (using the information provided by the Mediator in WiFiConf Resource):

– The Enrollee shall set "lec" Property to 1, if it fails to connect because it can’t find the SSID.

– The Enrollee shall set "lec" Property to 2, if it fails to connect due to wrong credential (password) information.

– The Enrollee should set "lec" Property to 6, if the Auth type is not supported by the Enrollee.

– The Enrollee should set "lec" Property to 7, if the Encryption type is not supported by the Enrollee.

– The Enrollee should set "lec" Property to 8, if it fails to connect due to wrong Auth type information (even though it’s supported by the Enrollee).

– The Enrollee should set "lec" Property to 9, if it fails to connect due to wrong Encryption type information (even though it’s supported by the Enrollee).

When using Wi-Fi as the connectivity between the Enrollee and Mediator, if the Enrollee fails to connect to the Enroller, it shall again make itself discoverable as a Soft AP (in case it destroyed its Soft AP earlier).

9.4.2 eSIM Easy Setup error handling

The "Error" in the "ps" Property of the eSIMEasySetup Resource (i.e. "oic.r.esimeasysetup") is used to indicate that an error occurred in the eSIM Easy Setup process while RSP procedure:

– The Enrollee shall set "ps" Property to "Error" if it fails to download and install an eSIM Profile.

– "ler" and "lec" Properties shall be used to indicate the detailed failure reason and error code within eSIM Profile download and installation.

– "led" Property may be used to indicate additional error description.

– "euc". Property shall be used to indicate an end user consent. If an end user rejects RSP procedure, Enrollee shall set "ps" Value to "Error", "euc" Value to "Download Reject", and then shall terminate the eSIM Easy Setup Procedure.

For more detailed Error handling within the Remote SIM Provisioning Procedure, please refer to the GSMA RSP Technical Specification.

9.5 Example Easy Setup flow

Figure 7 shows an example Easy Setup flow for informative purposes.

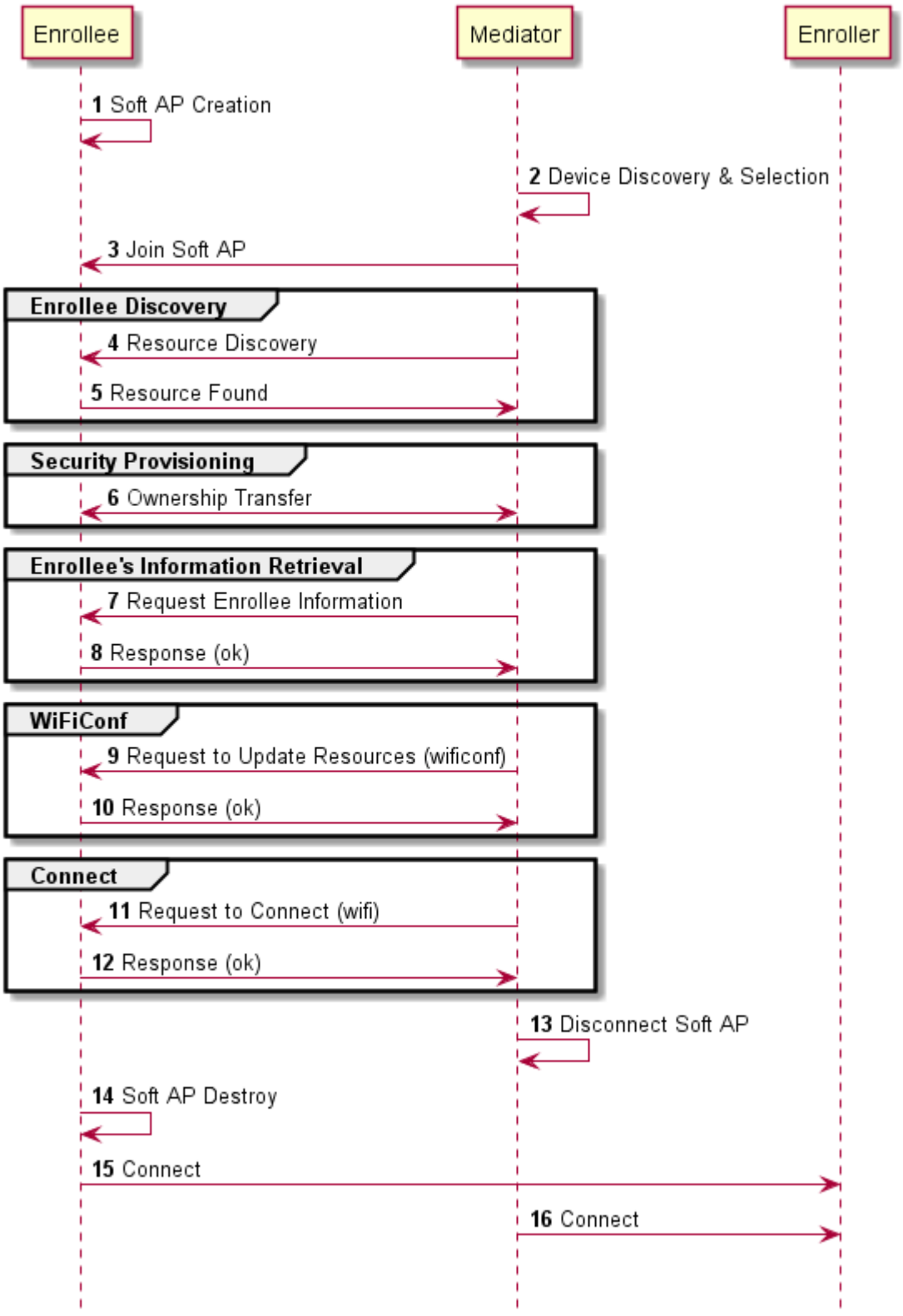


Figure 7 – Easy Setup Flow (Informative)

The example flow in Figure 7 undergoes security provisioning (step 6) during Easy Setup. Alternatively, security provisioning can be done before Enrollee Discovery (steps 4 and 5) if preferred. Please refer to ISO/IEC 30118-2 for more information on the different scenarios.

Figure 8 shows an example of an eSIM Easy Setup flow based on clause 7.3 for informative purposes.

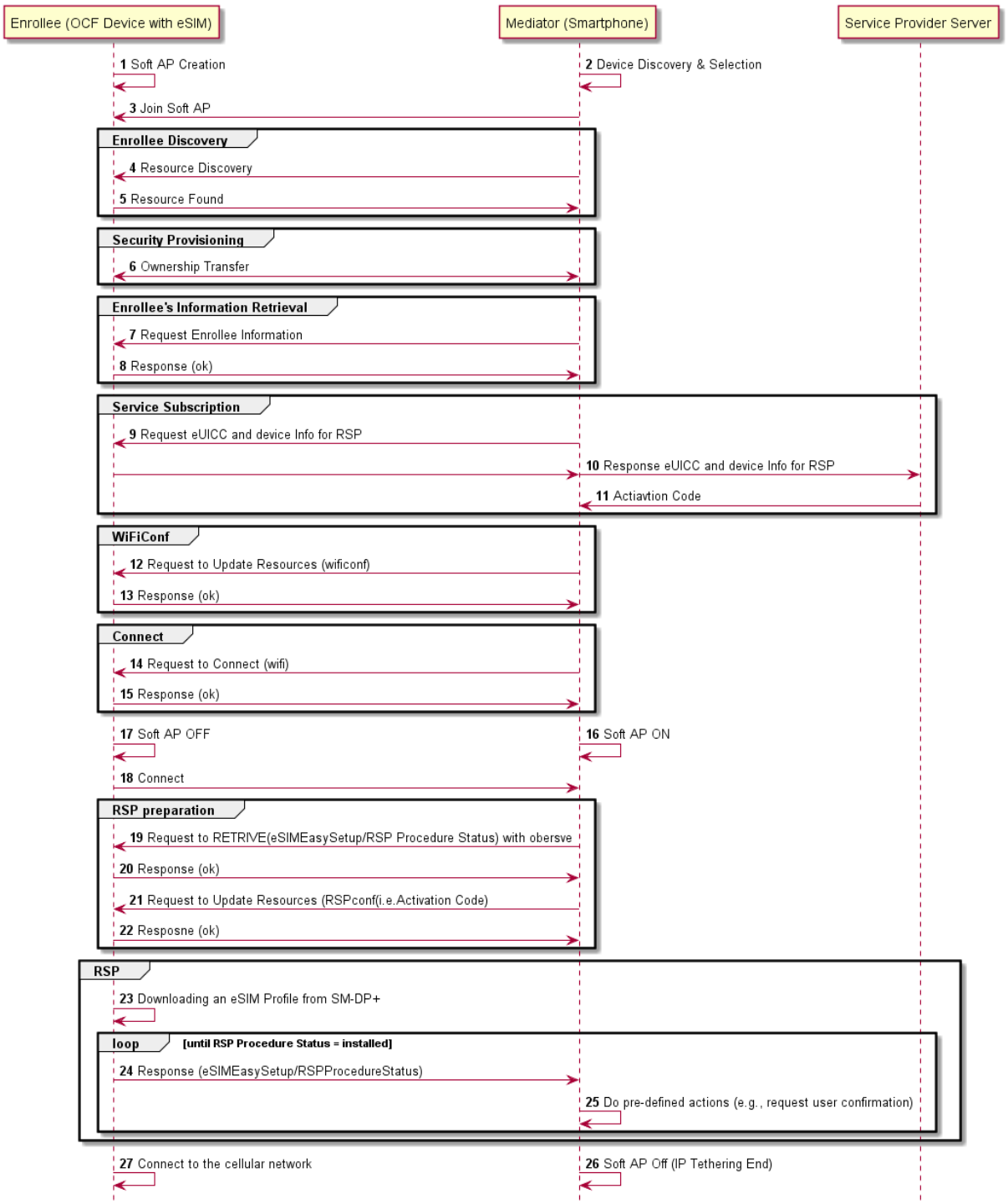


Figure 8 – eSIM Easy Setup Flow (Informative)

The individual elements in the flow are further described as follows:

1. Enrollee turns on Soft AP for Easy Setup.
2. Mediator starts searching for the AP, and finds an Enrollee on the scanned list. An Enrollee may be identified using Easy Setup SSID tag as defined in clause 8.6

3. The Soft AP of the Enrollee supports a passphrase for connection by the Mediator. Please refer to ISO/IEC 30118-2 for more information.
4. Mediator discovers the Enrollee’s Resources by doing a RETRIEVE operation on the known "/oic/res" Resource.
5. The "/oic/res" response from all Enrollees includes all supported Resource Types, including the eSIM Easy Setup Resource and Wi-Fi Easy Setup Resource. Detailed Resource information (e.g. Value rule, Value type) is not discoverable at this stage.
6. Security Provisioning occurs by doing Ownership Transfer. At this stage, the Enrollee is onboarded to the OCF Ecosystem. Please refer to ISO/IEC 30118-2 for more detailed information.
7. Mediator RETRIEVES the eSIM Easy Setup and Easy Setup Resources.
8. Enrollee responds with Resource Representations via secure connection.
9. The end user indicates a desire to buy a cellular plan via an on-device service activation application of the Mediator; Mediator enters eSIM Easy Setup Mode. Mediator requests eUICC Information ("euiccinfo") and Device Information for RSP ("deviceinfo") from the Enrollee for capability negotiation and eligibility check. The Enrollee retrieves corresponding values from LPA (i.e.EUICCInfo2, DeviceInfo), and then returns those values to the Mediator.
10. Mediator forwards eUICC Information ("euiccinfo") and Device Information for RSP ("deviceinfo") to the service operator server. Based on this information, the service provider provides cellular plans to select from.
11. Once the end user finishes the contract on their cellular plan, the service provider server sends an Activation Code to the Mediator.
12. When using Wi-Fi for IP tethering, Mediator sends a unicast UPDATE operation to the Wi-FiConf Resource. Under eSIM Easy Setup Mode, Mediator updates Wi-FiConf Resource in the Enrollee to the Mediator’s own SoftAP information (e.g. SSID, Password) to provide IP tethering.
13. Enrollee sends Response (ok) message to the Mediator.
14. To request connection, Mediator sends an UPDATE operation to the Enrollee to change the Connect ("cn") Property value to "1" in the EasySetup Resource.
15. Enrollee sends Response (ok) message to Mediator.
16. Mediator turns on its mobile hotspot, and acts as a Soft AP.
17. Enrollee turns off Soft AP.
18. Enrollee joins to the Mediator’s AP using provided information in the Step 12.
19. The NOTIFY operation is used to provide asynchronous notification of state changes; this is enabled via the sending of a RETRIEVE containing an "observe" indication to the eSIMEasySetup Resource. Refer to ISO/IEC 30118-1 for more detailed information.
20. Enrollee sends a RETRIEVE response including an Observe indication.
21. Mediator sends an UPDATE operation to the Enrollee to set the Activation Code ("ac") Property in the RSPConf Resource. Enrollee sets RSP Procedure Status ("ps") to "Initiated" when Activation Code is written.
22. Enrollee sends Response (ok) message to Mediator.
23. Internal to the Enrollee, the Activation Code ("ac") is delivered to LPA, and as receiving Activation Code, Enrollee starts downloading an eSIM Profile from SM-DP+ server using IP connectivity provided by Mediator (e.g. IP Tethering).
24. When the RSP Procedure Status ("ps") Resource value changes according to the input(s) from LPA, Enrollee sends NOTIFICATION operation to the Mediator.

25. On receiving the NOTIFICATION, Mediator performs predefined actions. This is the expected procedure in the "loop" until RSP Procedure Status is set to "Installed":
    a) After ES9+.AuthenticateClient(Success) returns to LPA, RSP Procedure Status ("ps") changes to "User confirmation pending", and the value change is Notified to the Mediator.
    b) The Mediator sends a RETRIEVE message to the RSPConf Resource to get Confirmation Code Required ("ccr") and eSIM Profile Metadata ("pm") Property values.
    c) Enrollee returns Confirmation Code Required ("ccr") value and eSIM Profile Metadata ("pm") value.
    d) Mediator displays the eSIM Profile Metadata ("pm") to get the end user consent, and request for Confirmation Code input if Confirmation Code Required ("ccr") sets to "True" in the RSPConf Resource.
    e) Mediator sends an UPDATE operation to the eSIMEasySetup Resource using the batch OCF Interface: RSP End User Consent ("euc") to the either "Download OK" or "Download and Enable OK", and Confirmation Code ("cc") to what the user entered if a confirmation code is required. Enrollee sets RSP Procedure Status ("ps") to "Confirmation received" when RSP End User Consent ("euc") is written.
    f) Download proceeds until it terminates at which point the Enrollee changes the RSP Procedure Status to "Downloaded" followed by "Installed" when the LPA receives ES9+.GetBoundProfilePackage(Success),ES9+.HandleNotification(Success) respectively.
26. If successfully "Installed", Mediator terminates the Soft AP, and then leaves eSIM Easy Setup mode.
27. Enrollee connects to the cellular network of the contracted mobile network operator.

NOTE OCF defines connectivity-agnostic protocol. Figure 8 used Wi-Fi for IP tethering for the purpose to illustrate End-to-End on device activation procedure.

9.6 Easy Setup SSID tags

If using Wi-Fi as the connectivity between the Enrollee and the Mediator, then the Enrollee's Soft AP SSID should contain exactly one of the following Easy Setup SSID tags:

- "OCF_"
  - Prefix tag that has to be at the beginning of the SSID.
  - Example: OCF_MySSID
- "_OCF"
  - Suffix tag that has to be at the end of the SSID.
  - Example: MySSID_OCF

These tags are case sensitive.

9.7 Easy Setup information element

9.7.1 Overview

If using Wi-Fi as the connectivity between the Enrollee and the Mediator, then the Enrollee's Soft AP beacon should contain the Easy Setup Information Element. The information element provides additional information about the device such as a friendly name or device manufacturer for the mediator application. The mediator application can then use this information to provide a better user experience.

9.7.2 OCF Device information element (IE)

The Easy Setup Information Element has the structure shown in Figure 9

1 byte	1 byte	3 bytes	1 byte	<252 bytes
Type = 221	Length	CID = 6A 40 65	OCF IE Type = 0	Data

Figure 9 – Easy Setup information element definition

– Type is a unique id allocated by the IEEE registrar to identify different information elements from each other. The Easy Setup Information Element shall have a Type value of 221 which is standard vendor specific information element.

– Length shall indicate the total size of CID, OCF IE Type, and Data in bytes.

– Company ID (CID) is a unique 24-bit identifier for a specific company or organization. The Easy Setup Information Element shall have a CID value of 6A 40 65.

– OCF IE Type is the identifier of the specific IE within OCF. The OCF IE Type shall be set to 0 for Easy Setup.

– Data is a set of type-length-value (TLV) structures that represent the device information in Table 1. The length of this field shall be less than 252 bytes.

Each TLV has the structure shown in Figure 10.

1 byte	1 byte	<250 bytes
Type	Length	Value

Figure 10 – Type-Length-Value structure

– Type shall indicate the type of the field from Table 15.

– Length shall indicate the length of the Value in bytes.

– Value shall represent the corresponding information for specific TLV type from Table 15.

Data is a set of TLVs as defined in Table 15.

Table 15 – Easy Setup information element TLVs

Type	Length (bytes)	Value	Description of TLV	# of Occurrences in IE or IEC	Required
1	<65	Friendly name of the device	Device Friendly Name	1	Y
2	<27	Device Type	Device type/Class	>=1	Y
3	<65	Name of Device Manufacturer	Manufacturer Name	1	Y
4	<43	Language tag for strings	See IETF RFC 5646	1	Y
5	16	Permanent Immutable ID in network byte order	See ISO/IEC 30118-1	1	Y
101	<65	Device Type/Class	Device Type as string	>=0	N

The TLVs may be set in any order inside an IE or IEC. All strings shall be UTF-8 encoded and shall not include a null terminator. All TLVs in Table 15 with a required value of "Y" shall be included in the IE or IEC (if multiple IEs are required). The value of each TLV shall meet the length requirements specified in Table 1.

9.7.2.1 Device friendly name (Type 1)

User readable string representing the friendly name of the device that is beaconing and ready to undergo Easy Setup. This should match "n" from "oic.wk.d" as defined in the ISO/IEC 30118-1.

This string is in the same language specified in the type 4 TLV.

9.7.2.2 Device Type (Type 2)

Device type shall be the shortened form of Device Type as specified in the ISO/IEC 30118-5. For example:

– Device Type as specified in the ISO/IEC 30118-5: "oic.d.airconditioner"

– Device Type as specified in a type 2 TLV: "airconditioner"

In cases where the device supports multiple functions, several type 2 TLVs may be included to represent each function of the device.

If the device does not support any of the functions as specified in the ISO/IEC 30118-5, at least one type 101 TLV shall be included. Type 101 TLV contains a user readable string in the same language specified in the type 4 TLV. (Ex: "Lock").

If the device supports more than one function, a mix of type 2 and type 101 TLVs may be used depending on which functions are defined in the ISO/IEC 30118-5.

9.7.2.3 Device manufacturer name (Type 3)

User readable string representing the manufacturer name of the device that is beaconing and ready to undergo Easy Setup. This should match "mnmn" Property from "oic.wk.p" as defined in the ISO/IEC 30118-1.

This string is in the same language specified in the type 4 TLV.

9.7.2.4 Language tag (Type 4)

The language of all strings shall be specified in a type 4 TLV. The value of the type 4 TLV shall contain a language tag as described in IETF RFC 5646 (Ex: "en-us"). If the actual length of the language tag exceeds 42 bytes, the manufacturer shall exclude subtags on the language tag until it is less than 43 bytes.

Please see 9.7.2.8 for information on supporting multiple languages.

If an IE contains a TLV that is a string (i.e. type 1, type 3 or type 101), then a type 4 TLV corresponding to the language of the string(s) shall also be present in the IE.

9.7.2.5 Protocol Independent ID (Type 5)

This shall match "piid" from "oic.wk.d" as defined in the ISO/IEC 30118-1.

The piid in the TLV shall be in network byte order.

9.7.2.6 Multiple information elements

Additional Easy Setup IEs may be present in the Soft AP beacon in the following situations:

– The total size of the TLVs is larger than the size of Data as defined in an Easy Setup Information Element.

– Support for multiple languages is necessary.

Two or more Easy Setup Information Elements are referred to as an Information Element Collection (IEC).

9.7.2.7 IEC for large TLV size support

If a TLV or set of TLVs will not fit into the current IE, a manufacturer may add additional Easy Setup IEs to contain the TLV/s thereby creating or extending an IEC. The additional IE shall contain the following fields as described in 9.7.2:

– Type

– Length

– CID

– OCF IE Type

If an IE contains a TLV that is a string (i.e. type 1, type 3 or type 101), then a type 4 TLV corresponding to the language of the string(s) shall also be present in the IE.

9.7.2.8 IEC for multiple language support

A manufacturer may include additional Easy Setup IEs to support multiple languages in the Soft AP beacon. In the case that a manufacturer needs to provide device information in more than one language, they shall include an additional copy of the IE/IEC for each additional language. Each additional IE/IEC shall include all of the mandatory TLVs defined in 9.7.2.

Security

A Device shall meet the Easy Setup security requirements specified in ISO/IEC 30118-2.

Annex A(normative)

OpenAPI 2.0 specification definitions

A.1 List of Resource Type definitions

Table A.1 contains the list of defined resources in this document.

Table A.1 – Alphabetized list of resources

Friendly Name (informative)		Resource Type (rt)	Clause
Device Configuration		"oic.r.devconf"	A.2
Easy Setup		"oic.r.easysetup"	A.3
Wi-Fi Configuration		"oic.r.wificonf"	A.4
eSIM Easy Setup		"oic.r. esimeasysetup"	A.5
Remote SIM Provisioning Capability		"oic.r.rspcapability"	A.6
RSP Configuration		"oic.r.rspconf"	A.7

A.2 Device Configuration

A.2.1 Introduction

The Device configuration Resource stores Device settings such as the Device name. Vendor-specific information can be added to the Resource.
The Device name is a human-friendly name read by a Mediator during easy setup.

A.2.2 Example URI

/example/DevConfResURI

A.2.3 Resource type

The Resource Type is defined as: "oic.r.devconf".

A.2.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Device Configuration",
    "version": "2019-03-06",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.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": {
    "/example/DevConfResURI" : {
```



```
      "get": {
        "description": "The Device configuration Resource stores Device settings such as the Device name. Vendor-specific information can be added to the Resource.\nThe Device name is a human-friendly name read by a Mediator during easy setup.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": {
              "rt": ["oic.r.devconf"],
              "dn" : "My Refrigerator"
            },
            "schema": { "$ref": "#/definitions/DevConf" }
          }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.r", "oic.if.baseline"]
    }
  },
  "definitions": {
    "DevConf" : {
      "properties": {
        "rt" : {
          "description": "Resource Type of the Resource",
          "items": {
            "enum": ["oic.r.devconf"],
            "maxLength": 64,
            "type": "string"
          },
          "minItems": 1,
          "readOnly": true,
          "uniqueItems": 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"
        },
        "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"
        },
        "dn": {
          "oneOf": [
```



```
            {
              "type": "string",
              "description": "Indicates a pre-configured Device name in language indicated by 'dl'
in /oic/con;  presented by an Enrollee Device to a Mediator Device during the easy-setup process",
              "pattern": "^.*$",
              "readOnly": true
            },
            {
              "type": "array",
              "items": {
                "type": "object",
                "properties": {
                  "language": {
                    "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/language-tag",
                    "readOnly": true,
                    "description": "An RFC 5646 language tag."
                  },
                  "value": {
                    "type": "string",
                    "description": "Pre-configured Device name in the indicated language.",
                    "pattern": "^.*$",
                    "readOnly": true
                  }
                }
              },
              "minItems" : 1,
              "readOnly": true,
              "description": "Localized device name."
            }
          ]
        }
      },
      "type" : "object",
      "required": ["dn"]
    }
  }
}
```

A.2.5 Property definition

Table A.2 defines the Properties that are part of the "oic.r.devconf" Resource Type.

Table A.2 – The Property definitions of the Resource with type "rt" = "oic.r.devconf".

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	
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource
dn	multiple types: see schema	Yes	Read Write	

A.2.6 CRUDN behaviour

Table A.3 defines the CRUDN operations that are supported on the "oic.r.devconf" Resource Type.

Table A.3 – The CRUDN operations of the Resource with type "rt" = "oic.r.devconf".

Create	Read	Update	Delete	Notify
	get			observe

A.3 Easy Setup Collection

A.3.1 Introduction

The Easy Setup Resource stores useful information including the current status of unboxing a Device and the last error code which are produced in the process of easy setup.
Note that the Easy Setup Resource is a Collection Resource, which contains Links to WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.

A.3.2 Example URI

/EasySetupResURI

A.3.3 Resource type

The Resource Type is defined as: "oic.r.easysetup, oic.wk.col".

A.3.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Easy Setup Collection",
    "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": {
    "/EasySetupResURI?if=oic.if.ll" : {
      "get": {
        "description": "The Easy Setup Resource stores useful information including the current
status of unboxing a Device and the last error code which are produced in the process of easy
setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "href": "/EasySetupResURI",
                    "rt": ["oic.r.easysetup", "oic.wk.col"],
                    "if": ["oic.if.b"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ],
                    "rel":["self", "item"]
```

```
                    },
                    {
                      "href": "/WiFiConfResURI",
                      "rt":   ["oic.r.wificonf"],
                      "if":   ["oic.if.baseline"],
                      "p":{"bm":3},
                      "eps": [
                        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                      ]
                    },
                    {
                      "href": "/DevConfResURI",
                      "rt":   ["oic.r.devconf"],
                      "if":   ["oic.if.baseline"],
                      "p":{"bm":3},
                      "eps": [
                        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                      ]
                    }
                  ],
                "schema": { "$ref": "#/definitions/slinks" }
              }
          }
        }
      },
      "/EasySetupResURI?if=oic.if.b" : {
        "get": {
          "description": "The Easy Setup Resource stores useful information including the current
status of unboxing a Device and the last error code which are produced in the process of easy
setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"}
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "href": "/EasySetupResURI",
                    "rep":{
                      "ps" : 0,
                      "lec": 0,
                      "cn": [1]
                    }
                  },
                  {
                    "href": "/WiFiConfResURI",
                    "rep":{
                      "swmt" : ["A", "B", "G"],
                      "swf": ["2.4G", "5G"],
                      "tnn": "Home_AP_SSID",
                      "cd": "Home_AP_PWD",
                      "wat": "WPA2_PSK",
                      "wet": "AES",
                      "swat": ["WPA_PSK", "WPA2_PSK"],
                      "swet": ["TKIP", "AES", "TKIP_AES"]
                    }
                  },
                  {
                    "href": "/DevConfResURI",
                    "rep":{
                        "dn" : "My Refrigerator"
                    }
                  }
                ],
              "schema": { "$ref": "#/definitions/sbatch" }
            }
          }
```



```
      },
      "post": {
        "description": "Able to deliver Wi-Fi, Device configuration and other
configuration\ninformation in a batch by utilizing 'batch' OCF Interface.\nIf you want to deliver
Wi-Fi and Device configuration information in a batch,\nyou can write all Properties you want to
send with a 'batch' OCF Interface.\nThe below example is the case to send Easy Setup and Wi-Fi
configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-update"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/sbatch-update" },
            "x-example":
              [
                {
                  "href": "/EasySetupResURI",
                  "rep":{
                    "cn": [1]
                  }
                },
                {
                  "href": "/WiFiConfResURI",
                  "rep":{
                    "tnn": "Home_AP_SSID",
                    "cd": "Home_AP_PWD",
                    "wat": "WPA2_PSK",
                    "wet": "AES"
                  }
                }
              ]
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example":
              [
                {
                  "href": "/EasySetupResURI",
                  "rep" : {
                    "ps" : 0,
                    "lec": 0,
                    "cn": [1]
                  }
                },
                {
                  "href": "/WiFiConfResURI",
                  "rep" : {
                    "swmt" : ["A", "B", "G"],
                    "swf": ["2.4G", "5G"],
                    "tnn": "Home_AP_SSID",
                    "cd": "Home_AP_PWD",
                    "wat": "WPA2_PSK",
                    "wet": "AES",
                    "swat": ["WPA_PSK", "WPA2_PSK"],
                    "swet": ["TKIP", "AES", "TKIP_AES"]
                  }
                },
                {
                  "href": "/DevConfResURI",
                  "rep" : {
                    "dn" : "My Refrigerator"
                  }
                }
              ],
            "schema": { "$ref": "#/definitions/sbatch" }
          }
        }
```



```
      }
    },
    "/EasySetupResURI?if=oic.if.baseline" : {
      "get": {
        "description": "The Easy Setup Resource stores useful information including the current
status of unboxing a Device and the last error code which are produced in the process of easy
setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example":
              {
                "rt" : ["oic.r.easysetup", "oic.wk.col"],
                "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
                "ps" : 0,
                "lec": 0,
                "cn": [1],
                "links": [
                  {
                    "href": "/EasySetupResURI",
                    "rt": ["oic.r.easysetup", "oic.wk.col"],
                    "if": ["oic.if.b"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ],
                    "rel":["self", "item"]
                  },
                  {
                    "href": "/WiFiConfResURI",
                    "rt":   ["oic.r.wificonf"],
                    "if":   ["oic.if.baseline"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ]
                  },
                  {
                    "href": "/DevConfResURI",
                    "rt":   ["oic.r.devconf"],
                    "if":   ["oic.if.baseline"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ]
                  }
                ]
              },
            "schema": { "$ref": "#/definitions/EasySetup" }
          }
        }
      },
      "post": {
        "description": "Able to update connection type to attempt to connect to the Enroller to
start during while posting to /EasySetupResURI\nThe below example is the case to send Easy Setup
configuration\n(i.e. connection type) in a post.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-update"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/EasySetupUpdate" },
            "x-example":
             {
               "cn": [1]
```



```
              }
            }
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example":
               {
                 "rt" : ["oic.r.easysetup", "oic.wk.col"],
                 "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
                 "ps" : 0,
                 "lec": 0,
                 "cn": [1],
                 "links": [
                   {
                     "href": "/EasySetupResURI",
                     "rt": ["oic.r.easysetup", "oic.wk.col"],
                     "if": ["oic.if.b", "oic.if.ll", "oic.if.baseline"],
                     "p":{"bm":3},
                     "eps": [
                       {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                     ],
                     "rel":["self", "item"]
                   },
                   {
                     "href": "/WiFiConfResURI",
                     "rt":   ["oic.r.wificonf"],
                     "if":   ["oic.if.rw", "oic.if.baseline"],
                     "p":{"bm":3},
                     "eps": [
                       {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                     ]
                   },
                   {
                     "href": "/DevConfResURI",
                     "rt":   ["oic.r.devconf"],
                     "if":   ["oic.if.r", "oic.if.baseline"],
                     "p":{"bm":3},
                     "eps": [
                       {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                     ]
                   }
                 ]
               },
               "schema": { "$ref": "#/definitions/EasySetup" }
             }
           }
         }
       }
    },
    "parameters": {
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.ll","oic.if.b","oic.if.baseline"]
      },
      "interface-update" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.b", "oic.if.baseline"]
      }
    },
    "definitions": {
      "oic.oic-link": {
        "type": "object",
        "properties": {
          "anchor": {
            "$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.baseline",
              "oic.if.ll",
              "oic.if.b",
              "oic.if.r",
              "oic.if.rw"
            ],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "uniqueItems": true,
          "type": "array"
        },
        "rt": {
          "description": "Resource Type of the target Resource",
          "items": {
            "maxLength": 64,
            "type": "string"
          },
          "minItems": 1,
          "uniqueItems": true,
          "type": "array"
        }
```



```
        },
        "required": [
          "href",
          "rt",
          "if"
        ]
      },
      "slinks" : {
        "type": "array",
        "items": {
          "$ref": "#/definitions/oic.oic-link"
        }
      },
      "sbatch" : {
        "minItems" : 1,
        "items" : {
          "additionalProperties": true,
          "properties": {
            "href": {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
            },
            "rep": {
              "description": "The response payload from a single Resource",
              "type": "object",
              "anyOf": [
                {
                    "$ref": "#/definitions/EasySetup"
                },
                {
                    "$ref": "https://openconnectivityfoundation.github.io/core-
extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConf"
                },
                {
                    "$ref": "https://openconnectivityfoundation.github.io/core-
extensions/swagger2.0/oic.r.devconf.swagger.json#/definitions/DevConf"
                }
              ]
            }
          },
          "required": [
            "href",
            "rep"
          ],
          "type": "object"
        },
        "type" : "array"
      },
      "sbatch-update" : {
        "minItems" : 1,
        "items" : {
          "additionalProperties": true,
          "description": "Array of Resource representations to apply to the batch Collection, using
href to indicate which resource(s) in the batch to update. If the href Property is empty,
effectively making the URI reference to the Collection itself, the representation is to be applied
to all Resources in the batch",
          "properties": {
            "href": {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
            },
            "rep": {
              "description": "The response payload from a single Resource",
              "type": "object",
              "anyOf": [
                {
                  "$ref": "#/definitions/EasySetupUpdate"
                },
```

```
                {
                  "$ref": "https://openconnectivityfoundation.github.io/core-
extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConfUpdate"
                }
              ]
            }
          },
          "required": [
            "href",
            "rep"
          ],
          "type": "object"
        },
        "type" : "array"
      },
      "EasySetup" : {
        "properties": {
          "n" :  {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
          },
          "rts" : {
            "description": "Resource Type of the Resources within the Collection",
            "items": {
              "maxLength": 64,
              "type": "string"
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "id" : {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
          },
          "rts-m" : {
            "description": "Resource Type of the mandatory Resources within the Collection",
            "items": {
              "maxLength": 64,
              "type": "string"
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "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"
          },
          "rt" : {
            "items": {
              "enum": [
                "oic.r.easysetup",
                "oic.wk.col"
```



```
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 2,
            "type": "array",
            "uniqueItems": true
          },
          "ps" : {
            "description": "Indicates the easy setup status of the Device. (0: Need to Setup, 1:
Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved,
255: EOF)",
            "enum": [
              0,
              1,
              2,
              3
            ],
            "readOnly": true,
            "type": "integer"
          },
          "lec" : {
            "description": "Indicates a failure reason (0: No error, 1: A given SSID is not found, 2:
Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6:
Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the
Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi
Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown
error)",
            "enum": [
              0,
              1,
              2,
              3,
              4,
              5,
              6,
              7,
              8,
              9,
              255
            ],
            "readOnly": true,
            "type": "integer"
          },
          "cn" : {
            "description": "Indicates an array of connection types that trigger an attempt to connect
to the Enroller to start.",
            "items": {
              "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
to be added in future (e.g. Connect to cloud / BLE))",
              "type": "integer"
            },
            "type": "array"
          },
          "links" : {
            "type": "array",
            "description": "A set of OCF Links.",
            "items": {
              "$ref": "#/definitions/oic.oic-link"
            }
          }
        },
        "type" : "object",
        "required": ["ps","lec","cn"]
      },
      "EasySetupUpdate" : {
        "additionalProperties": true,
        "description": "Update to writeable values in EasySetupResURI",
        "properties": {
          "cn" : {
```

```
          "description": "Indicates an array of connection types that trigger an attempt to connect
to the Enroller to start.",
          "items": {
            "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
to be added in future (e.g. Connect to cloud / BLE))",
            "type": "integer"
          },
          "type": "array"
        }
      },
      "required": [
        "cn"
      ],
      "type": "object"
    }
  }
}
```

A.3.5 Property definition

Table A.4 defines the Properties that are part of the "oic.r.easysetup, oic.wk.col" Resource Type.

Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.easysetup, oic.wk.col".

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	
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 Write	The OCF Interfaces supported by the target Resource
rt	array: see schema	Yes	Read Write	Resource Type of the target Resource
href	multiple types: see schema	Yes	Read Write	
rep	object: see schema	Yes	Read Write	The response payload from a single Resource
href	multiple types: see schema	Yes	Read Write	

rep	object: see schema	Yes	Read Write	The response payload from a single Resource
n	multiple types: see schema	No	Read Write	
rts	array: see schema	No	Read Only	Resource Type of the Resources within the Collection
id	multiple types: see schema	No	Read Write	
rts-m	array: see schema	No	Read Only	Resource Type of the mandatory Resources within the Collection
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource
rt	array: see schema	No	Read Write	
ps	integer	Yes	Read Only	Indicates the easy setup status of the Device. (0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF)
lec	integer	Yes	Read Only	Indicates a failure reason (0: No error, 1: A given SSID is not found, 2: Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown error)
cn	array: see schema	Yes	Read Write	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.
links	array: see schema	No	Read Write	A set of OCF Links.

cn	array: see schema	Yes	Read Write	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.

A.3.6 CRUDN behaviour

Table A.5 defines the CRUDN operations that are supported on the "oic.r.easysetup, oic.wk.col" Resource Type.

Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.easysetup, oic.wk.col".

Create	Read	Update	Delete	Notify
	get	post		observe

A.4 Wi-Fi Configuration

A.4.1 Introduction

WiFiConf Resource stores essential information to help an unboxing Device to connect to an existing Wi-Fi AP.

A.4.2 Example URI

/WiFiConfResURI

A.4.3 Resource type

The Resource Type is defined as: "oic.r.wificonf".

A.4.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Wi-Fi Configuration",
    "version": "2019-03-27",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.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": {
    "/WiFiConfResURI?if=oic.if.rw" : {
      "get": {
        "description": "The WiFiConf Resource stores essential information to help an unboxing
Device connect to an existing Wi-Fi AP.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example":
```

```
                {
                  "tnn": "Home_AP_SSID",
                  "swmt" : ["A", "B", "G"],
                  "swf": ["2.4G", "5G"],
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "AES",
                  "swat": ["WPA_PSK", "WPA2_PSK"],
                  "swet": ["TKIP", "AES", "TKIP_AES"]
                },
              "schema": { "$ref": "#/definitions/WiFiConf" }
            }
          }
        },
        "post": {
          "description": "Deliver Wi-Fi AP's information for an unboxing Device to connect to it.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"},
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
              "x-example":
                {
                  "tnn": "Home_AP_SSID",
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "AES"
                }
            }
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "tnn": "Home_AP_SSID",
                  "swmt" : ["A", "B", "G"],
                  "swf": ["2.4G", "5G"],
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "AES",
                  "swat": ["WPA_PSK", "WPA2_PSK"],
                  "swet": ["TKIP", "AES", "TKIP_AES"]
                },
              "schema": { "$ref": "#/definitions/WiFiConf" }
            }
          }
        }
      },
      "/WiFiConfResURI?if=oic.if.baseline" : {
        "get": {
          "description": "WiFiConf Resource stores essential information to help an unboxing
Device\nto connect to an existing Wi-Fi AP.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"}
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt": ["oic.r.wificonf"],
                  "if": ["oic.if.rw", "oic.if.baseline"],
                  "swmt" : ["A", "B", "G"],
                  "swf": ["2.4G", "5G"],
                  "tnn": "Home_AP_SSID",
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
```



```
                  "wet": "TKIP",
                  "swat": ["WPA_PSK", "WPA2_PSK"],
                  "swet": ["TKIP", "AES", "TKIP_AES"]
                },
              "schema": { "$ref": "#/definitions/WiFiConf" }
            }
          }
        },
        "post": {
          "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"},
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
              "x-example":
                {
                  "tnn": "Home_AP_SSID",
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "AES"
                }
            }
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt": ["oic.r.wificonf"],
                  "if": ["oic.if.rw", "oic.if.baseline"],
                  "tnn": "Home_AP_SSID",
                  "swmt" : ["A", "B", "G"],
                  "swf": ["2.4G", "5G"],
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "AES",
                  "swat": ["WPA_PSK", "WPA2_PSK"],
                  "swet": ["TKIP", "AES", "TKIP_AES"]
                },
              "schema": { "$ref": "#/definitions/WiFiConf" }
            }
          }
        }
      }
    },
    "parameters": {
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.rw", "oic.if.baseline"]
      }
    },
    "definitions": {
      "WiFiConf" : {
        "properties": {
          "rt" : {
            "description": "Resource Type of the Resource",
            "items": {
              "enum": ["oic.r.wificonf"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
```



```
        },
        "tnn" : {
          "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
          "pattern": "^.*$",
          "type": "string"
        },
        "swmt" : {
          "description": "Indicates supported Wi-Fi mode types. It can be multiple",
          "items": {
            "description": "Supported Wi-Fi Mode Type.",
            "enum": [
              "A",
              "B",
              "G",
              "N",
              "AC"
            ],
            "type": "string"
          },
          "readOnly": true,
          "type": "array"
        },
        "wat" : {
          "description": "Indicates Wi-Fi Auth Type",
          "enum": [
            "None",
            "WEP",
            "WPA_PSK",
            "WPA2_PSK"
          ],
          "type": "string"
        },
        "n" : {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
        },
        "swat" : {
          "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
          "items": {
            "description": "Indicates Wi-Fi Auth Type",
            "enum": [
              "None",
              "WEP",
              "WPA_PSK",
              "WPA2_PSK"
            ],
            "type": "string"
          },
          "readOnly": true,
          "type": "array"
        },
        "swf" : {
          "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
Valid values are ('2.4G', '5G')",
          "items": {
            "pattern": "^(2\\.4|5)G$",
            "type": "string"
          },
          "readOnly": true,
          "type": "array"
        },
        "swet" : {
          "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
          "items": {
            "description": "Indicates Wi-Fi Encryption Type",
            "enum": [
              "None",
              "WEP_64",
              "WEP_128",
```



```
                "TKIP",
                "AES",
                "TKIP_AES"
              ],
              "type": "string"
            },
            "readOnly": true,
            "type": "array"
          },
          "wet" : {
            "description": "Indicates Wi-Fi Encryption Type",
            "enum": [
              "None",
              "WEP_64",
              "WEP_128",
              "TKIP",
              "AES",
              "TKIP_AES"
            ],
            "type": "string"
          },
          "cd" : {
            "description": "Indicates credential information of Wi-Fi AP",
            "pattern": "^.*$",
            "type": "string"
          },
          "id" : {
            "$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": 2,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          }
        },
        "type" : "object",
        "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
      },
      "WiFiConfUpdate" : {
        "properties": {
          "wat" : {
            "description": "Indicates Wi-Fi Auth Type",
            "enum": [
              "None",
              "WEP",
              "WPA_PSK",
              "WPA2_PSK"
            ]
          },
          "cd" : {
            "description": "Indicates credential information of Wi-Fi AP",
            "pattern": "^.*$",
            "type": "string"
          },
          "wet" : {
            "description": "Indicates Wi-Fi Encryption Type",
            "enum": [
              "None",
```



```
              "WEP_64",
              "WEP_128",
              "TKIP",
              "AES",
              "TKIP_AES"
            ]
          },
          "tnn" : {
            "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
            "pattern": "^.*$",
            "type": "string"
          }
        },
        "type" : "object",
        "required":["tnn", "wat", "wet"]
      }
    }
}
```

A.4.5 Property definition

Table A.6 defines the Properties that are part of the "oic.r.wificonf" Resource Type.

Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.wificonf".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource
tnn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP)
swmt	array: see schema	Yes	Read Only	Indicates supported Wi-Fi mode types. It can be multiple
wat	string	Yes	Read Write	Indicates Wi-Fi Auth Type
n	multiple types: see schema	No	Read Write	
swat	array: see schema	Yes	Read Only	Indicates supported Wi-Fi Auth types. It can be multiple
swf	array: see schema	Yes	Read Only	Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple. Valid values are ('2.4G', '5G')
swet	array: see schema	Yes	Read Only	Indicates supported Wi-Fi Encryption types. It can be multiple
wet	string	Yes	Read Write	Indicates Wi-Fi Encryption Type
cd	string	No	Read Write	Indicates credential information of Wi-Fi AP
id	multiple types: see schema	No	Read Write	

if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource
wat	multiple types: see schema	Yes	Read Write	Indicates Wi-Fi Auth Type
cd	string	No	Read Write	Indicates credential information of Wi-Fi AP
wet	multiple types: see schema	Yes	Read Write	Indicates Wi-Fi Encryption Type
tnn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP)

A.4.6 CRUDN behaviour

Table A.7 defines the CRUDN operations that are supported on the "oic.r.wificonf" Resource Type.

Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.wificonf".

Create	Read	Update	Delete	Notify
	get	post		observe

A.5 eSIM Easy Setup Collection

A.5.1 Introduction

The eSIMEasySetup Resource Type stores useful information including Remote SIM Provisioning (RSP) status,
and RSP last error code which was produced in the process of eSIM Easy Setup.
Note that the eSIM Easy Setup Resource is a Collection Resource, which contains Links to RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.

A.5.2 Example URI

/eSIMEasySetupResURI

A.5.3 Resource type

The Resource Type is defined as: "oic.r.esimeasysetup".

A.5.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "eSIM Easy Setup Collection",
    "version": "2020-09-01",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
```



```
    "/eSIMEasySetupResURI?if=oic.if.ll" : {
      "get": {
        "description": "The eSIMEasySetup Resource Type stores useful information including Remote
SIM Provisioning (RSP) status, and RSP last error which was produced in the process of eSIM Easy
Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links to
RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "href": "/eSIMEasySetupResURI",
                    "rt": ["oic.r.esimeasysetup"],
                    "if": ["oic.if.b", "oic.if.baseline", "oic.if.ll"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ],
                    "rel":["self", "item"]
                  },
                  {
                    "href": "/RSPConfResURI",
                    "rt":   ["oic.r.rspconf"],
                    "if":   ["oic.if.baseline", "oic.if.rw"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ]
                  },
                  {
                    "href": "/RSPCapabilityResURI",
                    "rt":   ["oic.r.rspcapability"],
                    "if":   ["oic.if.baseline", "oic.if.r"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ]
                  }
                ],
              "schema": { "$ref": "#/definitions/slinks" }
            }
        }
      }
    },
    "/eSIMEasySetupResURI?if=oic.if.b" : {
      "get": {
        "description": "The eSIMEasySetup Resource Type stores useful information including Remote
SIM Provisioning (RSP) status, and RSP last error code which was produced in the process of eSIM
Easy Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links
to RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example":
              [
                {
                  "href": "/eSIMEasySetupResURI",
                  "rep":{
                    "ps" : "User confirmation pending",
                    "ler": "",
                    "lec": "",
                    "led": "",
                    "euc": "Undefined"
```



```
                    }
                  },
                  {
                    "href": "/RSPConfResURI",
                    "rep":{
                      "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
                      "pm":
"vyU4WgqJAQIDBAUGBwgJkRNTZXJ2aWNlUHJvdmlkZXJOYW1lkgtQcm9maWxlTmFtZZMBAJQCAACVAQI=",
                      "cc": "",
                      "ccr": true
                    }
                  },
                  {
                    "href": "/RSPCapabilityURI",
                    "rep":{
                     "euiccinfo":
"vyJ7gQMCAACCAwICAYMDQQEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDAgMAiAIEkKkWBBRmWhQz1nwaLF24tSyWfxCgV7p
csqoWBBRmWhQz1nwaLF24tSyWfxCgV7pcsosBAgQDAQAADBAxMDAwMDAwMDAwMDAwMDAw",
                      "deviceinfo" :
"oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMDBAUGhAMFBgeFAwYHCIYDBwgJhwMICQo="
                    }
                  }
                ],
              "schema": { "$ref": "#/definitions/sbatch" }
            }
          }
        },
        "post": {
          "description": "Able to deliver RSP Configuration, RSP Capability and other
configuration\ninformation in a batch by utilizing 'batch' OCF Interface.\nIf you want to deliver in
a batch,\nyou can write all Properties you want to send with a 'batch' OCF Interface.\nThe below
example is the case to send eSIMEasySetup and RSP configuration\n(i.e., RSP Procedure Status,
Activation Code, Confirmation Code required) in a batch.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-update"},
            {
              "name": "body",
              "in": "body",
              "required": true,
              "schema": { "$ref": "#/definitions/sbatch-update" },
              "x-example":
                [
                  {
                    "href": "/eSIMEasySetupResURI",
                    "rep":{
                      "euc": "Download OK"
                    }
                  },
                  {
                    "href": "/RSPConfResURI",
                    "rep":{
                      "cc": "102030405"
                    }
                  }
                ]
            }
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "href": "/eSIMEasySetupResURI",
                    "rep" : {
                      "ps" : "Confirmation received",
                      "ler": "",
                      "lec": "",
                      "led": "",
                      "euc": "Download OK"
```



```
                    }
                  },
                  {
                    "href": "/RSPConfResURI",
                    "rep" : {
                      "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
                      "pm":
"vyU4WgqJAQIDBAUGBwgJkRNTZXJ2aWNlUHJvdmlkZXJOYW1lkgtQcm9maWxlTmFtZZMBAJQCAACVAQI=",
                      "cc": "102030405",
                      "ccr": true
                    }
                  },
                  {
                    "href": "/RSPCapabilityResURI",
                    "rep" : {
                      "euiccinfo":
"vyJ7gQMCAACCAwICAYMDQQEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDAgMAiAIEkKkWBBRmWhQz1nwaLF24tSyWfxCgV7p
csqoWBBRmWhQz1nwaLF24tSyWfxCgV7pcsosBAgQDAQAADBAxMDAwMDAwMDAwMDAwMDAw",
                      "deviceinfo" :
"oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMDBAUGhAMFBgeFAwYHCIYDBwgJhwMICQo="
                    }
                  }
                ],
              "schema": { "$ref": "#/definitions/sbatch" }
            }
          }
        }
      },
      "/eSIMEasySetupResURI?if=oic.if.baseline" : {
        "get": {
          "description": "The eSIMEasySetup Resource Type stores useful information including Remote
SIM Provisioning (RSP) status,\n and RSP last error code which was produced in the process of eSIM
Easy Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links
to RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
          "parameters": [
            {"$ref": "#/parameters/interface-all"}
          ],
          "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt" : ["oic.r.esimeasysetup"],
                  "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
                  "ps" : ["Undefined", "Initiated", "User confirmation pending", "Confirmation
received", "Downloaded", "Installed", "Error"],
                  "ler": "",
                  "lec": "",
                  "led": "",
                  "euc": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download
OK", "Download and Enable OK"],
                  "links": [
                    {
                      "href": "/eSIMEasySetupResURI",
                      "rt": ["oic.r.esimeasysetup", "oic.wk.col"],
                      "if": ["oic.if.b", "oic.if.baseline", "oic.if.ll"],
                      "p":{"bm":3},
                      "eps": [
                        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                      ],
                      "rel":["self", "item"]
                    },
                    {
                      "href": "/RSPConfResURI",
                      "rt":   ["oic.r.rspconf"],
                      "if":   ["oic.if.baseline", "oic.if.rw"],
                      "p":{"bm":3},
                      "eps": [
                        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                      ]
```



```
                    },
                    {
                      "href": "/RSPCapabilityResURI",
                      "rt":   ["oic.r.rspcapability"],
                      "if":   ["oic.if.baseline", "oic.if.r"],
                      "p":{"bm":3},
                      "eps": [
                        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                      ]
                    }
                  ]
                },
              "schema": { "$ref": "#/definitions/eSIMEasySetup" }
            }
          }
        }
       }
    },
    "parameters": {
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.ll","oic.if.b","oic.if.baseline"]
      },
      "interface-update" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.b"]
      }
    },
    "definitions": {
      "oic.oic-link": {
        "type": "object",
        "properties": {
          "anchor": {
            "$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": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
          },
          "ins": {
            "$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.baseline",
                "oic.if.ll",
                "oic.if.b",
                "oic.if.r",
                "oic.if.rw"
              ],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "type": "array"
          },
          "rt": {
            "description": "Resource Type of the target Resource",
            "items": {
              "maxLength": 64,
              "type": "string"
            },
            "minItems": 1,
            "uniqueItems": true,
            "type": "array"
          }
        },
        "required": [
          "href",
          "rt",
          "if"
        ]
      },
      "slinks" : {
        "type": "array",
        "items": {
          "$ref": "#/definitions/oic.oic-link"
        }
      },
      "sbatch" : {
        "minItems" : 1,
        "items" : {
          "additionalProperties": true,
          "properties": {
            "href": {
              "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
            },
            "rep": {
              "description": "The response payload from a single Resource",
              "type": "object",
              "anyOf": [
                {
                    "$ref": "#/definitions/eSIMEasySetup"
                },
                {
                    "$ref": "https://openconnectivityfoundation.github.io/core-
```

```
extensions/swagger2.0/oic.r.rspconf.swagger.json#/definitions/RSPConf"
              },
              {
                  "$ref": "https://openconnectivityfoundation.github.io/core-
extensions/swagger2.0/oic.r.rspcapability.swagger.json#/definitions/RSPCapability"
              }
            ]
          }
        },
        "required": [
          "href",
          "rep"
        ],
        "type": "object"
      },
      "type" : "array"
    },
    "sbatch-update" : {
      "minItems" : 1,
      "items" : {
        "additionalProperties": true,
        "description": "Array of Resource representations to apply to the batch Collection, \nusing
href to indicate which resource(s) in the batch to update. \nIf the href Property is empty,
effectively making the URI reference to the Collection itself, \nthe representation is to be applied
to all Resources in the batch\n",
        "properties": {
          "href": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
          },
          "rep": {
            "description": "The response payload from a single Resource",
            "type": "object",
            "anyOf": [
              {
                  "$ref": "#/definitions/eSIMEasySetupUpdate"
              },
              {
                  "$ref": "https://openconnectivityfoundation.github.io/core-
extensions/swagger2.0/oic.r.rspconf.swagger.json#/definitions/RSPConfUpdate"
              }
            ]
          }
        },
        "required": [
          "href",
          "rep"
        ],
        "type": "object"
      },
      "type" : "array"
    },
    "eSIMEasySetup" : {
      "properties": {
        "n" :  {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
        },
        "rts" : {
          "description": "Resource Type of the Resources within the Collection",
          "items": {
            "maxLength": 64,
            "type": "string"
          },
          "minItems": 1,
          "uniqueItems": true,
          "readOnly": true,
          "type": "array"
```



```
        },
        "id" : {
          "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
        },
        "rts-m" : {
          "description": "Resource Type of the mandatory Resources within the Collection",
          "items": {
            "maxLength": 64,
            "type": "string"
          },
          "minItems": 1,
          "uniqueItems": true,
          "readOnly": true,
          "type": "array"
        },
        "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": 3,
          "uniqueItems": true,
          "readOnly": true,
          "type": "array"
        },
        "rt" : {
          "items": {
            "enum": [
              "oic.r.esimeasysetup"
            ],
            "type": "string",
            "maxLength": 64
          },
          "minItems": 1,
          "type": "array",
          "uniqueItems": true,
          "readOnly": true
        },
        "ps" : {
          "description": "Indicates the steps in Remote SIM Provisioning.\n",
          "enum": ["Undefined", "Initiated", "User confirmation pending", "Confirmation received",
"Downloaded", "Installed", "Error"],
          "readOnly": true,
          "type": "string"
        },
        "ler" : {
          "description": "Error Reason returned by the LPA while eSIM Easy Setup. \nIt indicates
where it was occurred.\n(e.g.,ES9+.GetBoundProfilePackage(Fail),
ES10b.LoadBoundProfilePackage(Fail))\n",
          "readOnly": true,
          "type": "string"
        },
        "lec" : {
          "description": "Error Code returned by the LPA while eSIM Easy Setup. \nIt indicates why
it was occurred.\nIt is mapped to the GSMA error status (e.g., \"8.8.1-3.8\", \"7\", \"6A 80\"')\n",
          "readOnly": true,
          "type": "string"
        },
        "led" : {
          "description": "Optional error description \nreturned by the LPA while eSIM Easy Setup.
(e.g., Invalid SM-DP+ Address)\n",
          "readOnly": true,
```



```
            "type": "string"
          },
          "euc" : {
            "description": "End User Consent for RSP.\n",
            "enum": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download OK",
"Download and Enable OK"],
            "type": "string"
          },
          "links" : {
            "type": "array",
            "description": "A set of OCF Links.",
            "items": {
              "$ref": "#/definitions/oic.oic-link"
            },
            "readOnly":true
          }
        },
        "type" : "object",
        "required": ["ps","ler","lec","euc"]
      },
      "eSIMEasySetupUpdate" : {
        "additionalProperties": true,
        "description": "Update to writeable values in eSIMEasySetupResURI",
        "properties": {
          "euc" : {
            "description": "End User Consent for RSP.\n",
            "enum": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download OK",
"Download and Enable OK"],
            "type": "string"
          }
       },
        "type": "object",
        "required": ["euc"]
      }
    }
}
```

A.5.5 Property definition

Table A.8 defines the Properties that are part of the "oic.r.esimeasysetup" Resource Type.

Table A.8 – The Property definitions of the Resource with type "rt" = "oic.r.esimeasysetup".

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	
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 Write	The OCF Interfaces supported by the target Resource
rt	array: see schema	Yes	Read Write	Resource Type of the target Resource
href	multiple types: see schema	Yes	Read Write	
rep	object: see schema	Yes	Read Write	The response payload from a single Resource
href	multiple types: see schema	Yes	Read Write	
rep	object: see schema	Yes	Read Write	The response payload from a single Resource
n	multiple types: see schema	No	Read Write	
rts	array: see schema	No	Read Only	Resource Type of the Resources within the Collection
id	multiple types: see schema	No	Read Write	
rts-m	array: see schema	No	Read Only	Resource Type of the mandatory Resources within the Collection
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource
rt	array: see schema	No	Read Only	
ps	string	Yes	Read Only	Indicates the steps in Remote SIM Provisioning.
ler	string	Yes	Read Only	Error Reason returned by the LPA while eSIM Easy Setup. It indicates where it was occurred. (e.g.,ES9+.GetBoundProfilePackage(Fail), ES10b.LoadBoundProfilePackage(Fail))
lec	string	Yes	Read Only	Error Code returned by the LPA while eSIM Easy Setup. It indicates why it was occurred. It is mapped to the GSMA error status (e.g., "8.8.1-3.8", "7", "6A 80"')
led	string	No	Read Only	Optional error description returned by the LPA while eSIM Easy Setup. (e.g., Invalid SM-DP+ Address)
euc	string	Yes	Read Write	End User Consent for RSP.
links	array: see schema	No	Read Only	A set of OCF Links.
euc	string	Yes	Read Write	End User Consent for RSP.

A.5.6 CRUDN behaviour

Table A.9 defines the CRUDN operations that are supported on the "oic.r.esimeasysetup" Resource Type.

Table A.9 – The CRUDN operations of the Resource with type "rt" = "oic.r.esimeasysetup".

Create	Read	Update	Delete	Notify
	Get	Post ("oic.if.b" only)		observe

A.6 Remote SIM Provisioning Capability

A.6.1 Introduction

RSPCapability Resource stores information to help a service provider to provide appropriate cellular plans to an end user.

A.6.2 Example URI

/RSPCapabilityResURI

A.6.3 Resource type

The Resource Type is defined as: "oic.r.rspcapability".

A.6.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Remote SIM Provisioning Capability",
    "version": "2020-09-01",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/RSPCapabilityResURI" : {
      "get": {
        "description": "RSPCapability Resource stores information to help a service provider to
provide appropriate cellular plans to an end user.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-all"}
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example":
              { "rt": ["oic.r.rspcapability"],
                "if": ["oic.if.r", "oic.if.baseline"],
                "euiccinfo":
"vyJ7gQMCAACCAwICAYMDQQEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDAgMAiAIEkKkWBBRmWhQz1nwaLF24tSyWfxCgV7p
csqoWBBRmWhQz1nwaLF24tSyWfxCgV7pcsosBAgQDAQAADBAxMDAwMDAwMDAwMDAwMDAw",
                "deviceinfo" :
"oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMDBAUGhAMFBgeFAwYHCIYDBwgJhwMICQo="
              },
            "schema": { "$ref": "#/definitions/RSPCapability" }
```

```
            }
          }
        }
      }
    },
    "parameters": {
      "interface-all" : {
        "in" : "query",
        "name" : "if",
        "type" : "string",
        "enum" : ["oic.if.r", "oic.if.baseline"]
      }
    },
    "definitions": {
      "RSPCapability" : {
        "properties": {
          "rt" : {
            "description": "Resource Type of the Resource",
            "items": {
              "enum": ["oic.r.rspcapability"],
              "type": "string",
              "maxLength": 64
            },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "euiccinfo" : {
            "description": "Refers to EUICCInfo2 defined in GSMA SGP.22 Annex H.This value type shall
be encoded as Major Type 2.",
            "type": "string",
            "readOnly": true,
            "maxLength": 1024
          },
          "deviceinfo" : {
            "description": "Refers to DeviceInfo defined in GSMA SGP.22 Annex H.This value type shall
be encoded as Major Type 2.",
            "type": "string",
            "readOnly": true,
            "maxLength": 128
          },
          "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 Interfaces 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":["euiccinfo", "deviceinfo"]
```



```
      }
    }
}
```

A.6.5 Property definition

Table A.10 defines the Properties that are part of the "oic.r.rspcapability" Resource Type.

Table A.10 – The Property definitions of the Resource with type "rt" = "oic.r.rspcapability".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource
euiccinfo	string	Yes	Read Only	Refers to EUICCInfo2 defined in GSMA SGP.22 Annex H.This value type shall be encoded as Major Type 2.
deviceinfo	string	Yes	Read Only	Refers to DeviceInfo defined in GSMA SGP.22 Annex H.This value type shall be encoded as Major Type 2.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource

A.6.6 CRUDN behaviour

Table A.11 defines the CRUDN operations that are supported on the "oic.r.rspcapability" Resource Type.

Table A.11 – The CRUDN operations of the Resource with type "rt" = "oic.r.rspcapability".

Create	Read	Update	Delete	Notify
	get			observe

A.7 RSP Configuration

A.7.1 Introduction

RSPConf Resource stores the information
used to download and install an eSIM Profile to an eSIM capable IoT device.
It comprises SM-DP+ server FQDN and Activation Code Token
binding to a specific subscription as defined by GSMA SGP.22.

A.7.2 Example URI

/RSPConfResURI

A.7.3 Resource type

The Resource Type is defined as: "oic.r.rspconf".

A.7.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info":  {
    "title": "RSP Configuration",
    "version": "2020-09-01",
    "license": {
        "name": "OCF Data Model License",
        "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
        "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/RSPConfResURI": {
      "get": {
        "description": "RSPConf Resource stores the information \nused to download and install an
eSIM Profile to an eSIM capable IoT device.\nIt comprises SM-DP+ server FQDN and Activation Code
Token\n binding to a specific subscription as defined by GSMA SGP.22.",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": {
              "rt": ["oic.r.rspconf"],
              "if": ["oic.if.rw", "oic.if.baseline"],
              "ac":"",
              "pm": "",
              "ccr": false
            },
            "schema": { "$ref": "#/definitions/RSPConf" }
          }
        }
      },
      "post": {
        "description": "Update Properties of the RSPConf Resource (deliver Activation Code in this
example).\n",
        "parameters": [
          {"$ref": "#/parameters/interface-rw"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/RSPConfUpdate" },
            "x-example": {
              "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815"
            }
          }
        ],
        "responses": {
          "200": {
            "description" : "",
            "x-example": {
              "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
              "pm": "",
              "ccr": false
            },
            "schema": { "$ref": "#/definitions/RSPConf" }
```



```
            }
          }
        }
      }
    },
    "parameters": {
      "interface": {
        "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": {
      "RSPConf": {
        "properties": {
          "rt": {
            "description": "The Resource Type.",
            "items": {
              "enum": ["oic.r.rspconf"],
              "maxLength": 64,
              "type": "string"
          },
            "minItems": 1,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
          },
          "ac": {
            "description": "The information needed to provision an eSIM device.",
            "maxLength": 256,
            "type": "string"
          },
          "pm": {
            "description": "Refers to ProfileInfo in GSMA SGP.22 Annex H.This value type shall be
encoded as Major Type 2",
            "maxLength": 2048,
            "type": "string",
            "readOnly": true
          },
          "cc": {
            "description": "A code entered by an end user required by the SM-DP+ \nto confirm the
download and installation of an eSIM Profile.\nThe Confirmation Code is provided from a service
provider to the end user.\n",
            "maxLength": 64,
            "type": "string"
          },
          "ccr": {
            "description": "Indicates whether a Confirmation Code is required.\n",
            "maxLength": 64,
            "type": "boolean",
            "readOnly": true
          },
          "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 Interface set supported by this Resource.",
            "items": {
              "enum": [
                "oic.if.rw",
                "oic.if.baseline"
              ],
              "type": "string"
            },
            "minItems": 2,
            "uniqueItems": true,
            "readOnly": true,
            "type": "array"
            }
          },
          "type": "object",
          "required": ["ac","pm","ccr"]
      },
      "RSPConfUpdate": {
        "properties": {
          "ac": {
            "description": "The information needed to provision an eSIM device.",
            "maxLength": 256,
            "type": "string"
          },
          "cc": {
            "description": "A code entered by an end user required by the SM-DP+ \nto confirm the
download and installation of an eSIM Profile.\nThe Confirmation Code is provided from a service
provider to the end user.\n",
            "maxLength": 64,
            "type": "string"
          }
        }
      }
    }
}
```

A.7.5 Property definition

Table A.12 defines the Properties that are part of the "oic.r.rspconf" Resource Type.

Table A.12 – The Property definitions of the Resource with type "rt" = "oic.r.rspconf".

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	The Resource Type.
ac	string	Yes	Read Write	The information needed to provision an eSIM device.
pm	string	Yes	Read Only	Refers to ProfileInfo in GSMA SGP.22 Annex H.This value type shall be encoded as Major Type 2
cc	string	No	Read Write	A code entered by an end user required by the SM-DP+ to confirm the download and installation of an eSIM Profile. The Confirmation Code is provided from a service provider to the end

				user.
ccr	boolean	Yes	Read Only	Indicates whether a Confirmation Code is required.
n	multiple types: see schema	No	Read Write	
id	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interface set supported by this Resource.
ac	string		Read Write	The information needed to provision an eSIM device.
cc	string		Read Write	A code entered by an end user required by the SM-DP+ to confirm the download and installation of an eSIM Profile. The Confirmation Code is provided from a service provider to the end user.

A.7.6 CRUDN behaviour

Table A.13 defines the CRUDN operations that are supported on the "oic.r.rspconf" Resource Type.

Table A.13 – The CRUDN operations of the Resource with type "rt" = "oic.r.rspconf".

Create	Read	Update	Delete	Notify
	get	post		observe

