

# OCF Easy Setup Specification 2.2.8

VERSION 2.2.8 | June 2025



**OPEN** CONNECTIVITY  
FOUNDATION™

**CONTACT** [admin@openconnectivity.org](mailto:admin@openconnectivity.org)  
Copyright Open Connectivity Foundation, Inc. © 2025.  
All Rights Reserved.



## Legal Disclaimer

2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19

NOTHING CONTAINED IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU ANY KIND OF LICENSE IN ITS CONTENT, EITHER EXPRESSLY OR IMPLIEDLY, OR TO ANY INTELLECTUAL PROPERTY OWNED OR CONTROLLED BY ANY OF THE AUTHORS OR DEVELOPERS OF THIS DOCUMENT. THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE AUTHORS AND DEVELOPERS OF THIS SPECIFICATION HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OPEN CONNECTIVITY FOUNDATION, INC. FURTHER DISCLAIMS ANY AND ALL WARRANTIES OF NON-INFRINGEMENT, ACCURACY OR LACK OF VIRUSES.

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

Copyright © 2017-2022 Open Connectivity Foundation, Inc. All rights reserved.

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

# CONTENTS

20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62

Introduction.....	viii
1 Scope.....	1
2 Normative references .....	1
3 Terms, definitions and abbreviated terms .....	2
3.1 Terms and definitions.....	2
3.2 Symbols and abbreviated terms .....	3
4 Document conventions and organization.....	4
4.1 Conventions.....	4
4.2 Notation .....	4
5 Overview .....	5
5.1 Introduction.....	5
5.2 Architecture .....	5
5.3 Example scenario .....	6
6 Easy setup overview.....	6
6.1 Introduction.....	6
6.2 EasySetup Resource .....	6
6.2.1 Overview .....	6
6.2.2 Resource.....	6
6.3 WiFiConf Resource Type .....	8
6.3.1 Introduction .....	8
6.3.2 Resource Type .....	8
6.4 DevConf Resource Type .....	9
6.4.1 Introduction .....	9
6.4.2 Resource Type .....	9
7 eSIM Easy Setup Overview .....	10
7.1 Introduction.....	10
7.2 Architecture .....	10
7.3 Example Scenario.....	11
7.4 eSIM Easy Setup Resource Model .....	12
7.4.1 Introduction .....	12
7.5 eSIMEasySetup Resource Type.....	12
7.5.1 Introduction .....	12
7.5.2 Resource Type Definition.....	12
7.6 RSPCapability Resource Type .....	15
7.6.1 Introduction .....	15
7.6.2 Resource Type Definition.....	15
7.7 RSPConf Resource Type .....	16
7.7.1 Introduction .....	16
7.7.2 Resource Type Definition.....	16
8 Network and connectivity.....	18
9 Functional interactions .....	18

63	9.1	Onboarding, Provisioning and Configuration .....	18
64	9.2	Resource discovery .....	18
65	9.3	Retrieving and updating Easy Setup Resources .....	18
66	9.4	Error handling .....	18
67	9.4.1	Easy Setup error handling .....	18
68	9.4.2	eSIM Easy Setup error handling .....	19
69	9.5	Example Easy Setup flow.....	19
70	9.6	Easy Setup SSID tags.....	24
71	9.7	Easy Setup information element.....	24
72	9.7.1	Overview .....	24
73	9.7.2	OCF Device information element (IE).....	24
74	10	Security .....	27
75	Annex A (normative)	OpenAPI 2.0 specification definitions .....	28
76	A.1	List of Resource Type definitions .....	28
77	A.2	Device Configuration.....	28
78	A.2.1	Introduction .....	28
79	A.2.2	Example URI .....	28
80	A.2.3	Resource type .....	28
81	A.2.4	OpenAPI 2.0 definition.....	28
82	A.2.5	Property definition .....	30
83	A.2.6	CRUDN behaviour .....	30
84	A.3	Easy Setup Collection.....	31
85	A.3.1	Introduction .....	31
86	A.3.2	Example URI .....	31
87	A.3.3	Resource type .....	31
88	A.3.4	OpenAPI 2.0 definition.....	31
89	A.3.5	Property definition .....	40
90	A.3.6	CRUDN behaviour .....	42
91	A.4	Wi-Fi Configuration .....	42
92	A.4.1	Introduction .....	42
93	A.4.2	Example URI .....	42
94	A.4.3	Resource type .....	42
95	A.4.4	OpenAPI 2.0 definition.....	42
96	A.4.5	Property definition .....	47
97	A.4.6	CRUDN behaviour .....	48
98	A.5	eSIM Easy Setup Collection.....	48
99	A.5.1	Introduction .....	48
100	A.5.2	Example URI .....	48
101	A.5.3	Resource type .....	48
102	A.5.4	OpenAPI 2.0 definition.....	48
103	A.5.5	Property definition .....	56
104	A.5.6	CRUDN behaviour .....	58
105	A.6	Remote SIM Provisioning Capability .....	58
106	A.6.1	Introduction .....	58
107	A.6.2	Example URI .....	58

108	A.6.3	Resource type .....	58
109	A.6.4	OpenAPI 2.0 definition.....	58
110	A.6.5	Property definition .....	60
111	A.6.6	CRUDN behaviour .....	60
112	A.7	RSP Configuration .....	60
113	A.7.1	Introduction .....	60
114	A.7.2	Example URI .....	60
115	A.7.3	Resource type .....	61
116	A.7.4	OpenAPI 2.0 definition.....	61
117	A.7.5	Property definition .....	63
118	A.7.6	CRUDN behaviour .....	64
119			
120			

121  
122  
123

## Figures

124	Figure 1 – Easy Setup deployment architecture .....	5
125	Figure 2 – Easy Setup Resource Types .....	6
126	Figure 3 – eSIM Easy Setup deployment architecture .....	10
127	Figure 4 – eSIM Easy Setup example scenario .....	11
128	Figure 5 – eSIM Easy Setup Resource Types .....	12
129	Figure 6 – RSP Procedure Status Transition .....	14
130	Figure 7 – Easy Setup Flow (Informative) .....	20
131	Figure 8 – eSIM Easy Setup Flow (Informative) .....	22
132	Figure 9 – Easy Setup information element definition.....	25
133	Figure 10 – Type-Length-Value structure .....	25

134  
135

## Tables

136		
137		
138	Table 1 – EasySetup Resource Type .....	7
139	Table 2 – "oic.r.easyssetup" Resource Type definition.....	7
140	Table 3 – WiFiConf Resource Type.....	8
141	Table 4 – "oic.r.wificonf" Resource Type definition.....	8
142	Table 5 – DevConf Resource Type .....	9
143	Table 6 – "oic.r.devconf" Resource Type definition .....	9
144	Table 7 – eSIMEasySetup Resource Type .....	13
145	Table 8 – "oic.r.esimeasyssetup" Resource Type Definition.....	13
146	Table 9 – GSMA RSP procedure mapping to the OCF RSP Procedure Status .....	15
147	Table 10 – Example of LPA received Error Message during RSP Procedure .....	15
148	Table 11 – RSPCapability Resource Type.....	15
149	Table 12 – "oic.r.rspcapability" Resource Type definition .....	16
150	Table 13 – RSPConf Resource Type.....	16
151	Table 14 – "oic.r.rspconf" Resource Type Definition.....	17
152	Table 15 – Easy Setup information element TLVs .....	25
153	Table A.1 – Alphabetized list of resources .....	28
154	Table A.2 – The Property definitions of the Resource with type "rt" = "oic.r.devconf". .....	30
155	Table A.3 – The CRUDN operations of the Resource with type "rt" = "oic.r.devconf". .....	31
156	Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.easyssetup,	
157	oic.wk.col". .....	40
158	Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.easyssetup,	
159	oic.wk.col". .....	42
160	Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.wificonf". .....	47
161	Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.wificonf". .....	48
162	Table A.8 – The Property definitions of the Resource with type "rt" =	
163	"oic.r.esimeasyssetup". .....	56
164	Table A.9 – The CRUDN operations of the Resource with type "rt" = "oic.r.esimeasyssetup". ..	58
165	Table A.10 – The Property definitions of the Resource with type "rt" = "oic.r.rspcapability". ..	60
166	Table A.11 – The CRUDN operations of the Resource with type "rt" = "oic.r.rspcapability". ..	60
167	Table A.12 – The Property definitions of the Resource with type "rt" = "oic.r.rspconf". .....	63
168	Table A.13 – The CRUDN operations of the Resource with type "rt" = "oic.r.rspconf". .....	64
169		
170		

171 **Introduction**

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

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

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

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

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

- 192 – Core framework
  - 193 – Core Specification
  - 194 – Security Specification
  - 195 – Onboarding Tool Specification
- 196 – Bridging framework and bridges
  - 197 – Bridging Specification
  - 198 – Resource to Alljoyn Interface Mapping Specification
  - 199 – OCF Resource to oneM2M Resource Mapping Specification
  - 200 – OCF Resource to BLE Mapping Specification
  - 201 – OCF Resource to EnOcean Mapping Specification
  - 202 – OCF Resource to LWM2M Mapping Specification
  - 203 – OCF Resource to UPlus Mapping Specification
  - 204 – OCF Resource to Zigbee Cluster Mapping Specification
  - 205 – OCF Resource to Z-Wave Mapping Specification
- 206 – Resource and Device models
  - 207 – Resource Type Specification
  - 208 – Device Specification
- 209 – Core framework extensions
  - 210 – Easy Setup Specification
  - 211 – Core Optional Specification
- 212 – OCF Cloud
  - 213 – Cloud API for Cloud Services Specification

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

216 **Easy Setup Specification**

217 **Scope**

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

221 **Normative references**

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

226 ISO/IEC 30118-1 Information technology -- Open Connectivity Foundation (OCF) Specification --  
 227 Part 1: Core specification  
 228 <https://www.iso.org/standard/53238.html>  
 229 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Core\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Core_Specification.pdf)

230 ISO/IEC 30118-2 Information technology -- Open Connectivity Foundation (OCF) Specification --  
 231 Part 2: Security specification  
 232 <https://www.iso.org/standard/74239.html>  
 233 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Security\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Security_Specification.pdf)

234 ISO/IEC 30118-5 Information technology -- Open Connectivity Foundation (OCF) Specification --  
 235 Part 5: Smart home device specification  
 236 <https://www.iso.org/standard/74242.html>  
 237 Latest version available at: [https://openconnectivity.org/specs/OCF\\_Device\\_Specification.pdf](https://openconnectivity.org/specs/OCF_Device_Specification.pdf)

238 IEEE 802.11, IEEE Standard for Information technology—Telecommunications and information  
 239 exchange between systems Local and metropolitan area networks—Specific requirements - Part  
 240 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)  
 241 Specifications, December 2016  
 242 <https://standards.ieee.org/findstds/standard/802.11-2016.html>

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

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

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

## 249 **Terms, definitions and abbreviated terms**

### 250 **3.1 Terms and definitions**

251 For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1 and the  
252 following apply.

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

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

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

#### 257 **3.1.1**

##### 258 **Easy Setup**

259 process of configuring an *Enrollee* (3.1.3) using a *Mediator* (3.1.5) by transferring of essential  
260 information to the *Enrollee* (3.1.3)

#### 261 **3.1.2**

##### 262 **Easy Setup Enrollment**

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

#### 265 **3.1.3**

##### 266 **Enrollee**

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

#### 268 **3.1.4**

##### 269 **Enroller**

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

#### 271 **3.1.5**

##### 272 **Mediator**

273 logical function that enables the *Enrollee* (3.1.3) to connect to the target network (i.e. *Enroller*  
274 (3.1.4))

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

#### 276 **3.1.6**

##### 277 **Activation Code**

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

#### 280 **3.1.7**

##### 281 **Local Profile Assistant (LPA)**

282 functional element in the device or in the eUICC that provides *Remote SIM Provisioning* (3.1.9)  
283 features to the device as defined in the GSMA RSP Technical Specification

#### 284 **3.1.8**

##### 285 **eSIM Profile**

286 combination of data and applications to be provisioned on an eUICC for the purpose of providing  
287 service.

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

#### 289 **3.1.9**

##### 290 **Remote SIM Provisioning (RSP)**

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

293 **3.1.10**  
294 **Subscription**  
295 commercial relationship between an end user and a service provider as defined in the GSMA RSP  
296 Technical Specification

297 **3.1.11**  
298 **Subscription Manager Data Preparation+ (SM-DP+)**  
299 *eSIM Profile* (3.1.8) preparation server which securely downloads *eSIM Profile* (3.1.8) to the *LPA*  
300 (3.1.7) of the respective eUICC in the device as defined in the GSMA RSP Technical Specification

301 **3.1.12**  
302 **Easy Setup Mode**  
303 mode that enables OCF setup and configuration to an IoT Device

304 **3.1.13**  
305 **eSIM Easy Setup Mode**  
306 mode that enables cellular network setting and configuration of *Remote SIM Provisioning* (3.1.9)

307 **3.2 Symbols and abbreviated terms**

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

## 315 Document conventions and organization

### 316 4.1 Conventions

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

321 In this document, to be consistent with the IETF usages for RESTful operations, the RESTful  
322 operation words CRUDN, CREATE, RETRIVE, UPDATE, DELETE, and NOTIFY will have all letters  
323 capitalized. Any lowercase uses of these words have the normal technical English meaning.

### 324 4.2 Notation

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

327 Required (or shall or mandatory)(M).

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

331 Recommended (or should)(S).

332 – These features add functionality supported by Core Architecture and should be implemented.  
333 Recommended features take advantage of the capabilities Core Architecture, usually without  
334 imposing major increase of complexity. Notice that for compliance testing, if a recommended  
335 feature is implemented, it shall meet the specified requirements to be in compliance with these  
336 guidelines. Some recommended features could become requirements in the future. The phrase  
337 "should not" indicates behaviour that is permitted but not recommended.

338 Allowed (may or allowed)(O).

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

342 DEPRECATED.

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

349 Conditionally allowed (CA)

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

352 Conditionally required (CR)

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

356

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

358 Words that are emphasized are printed in *italic*.

## 359 **Overview**

### 360 **5.1 Introduction**

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

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

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

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

369 – Device Configuration: Additional Device configuration information.

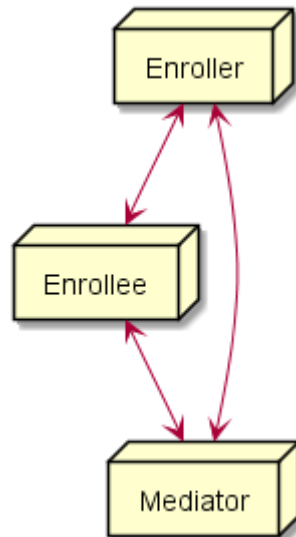
370 Easy Setup can be enhanced in future by incorporating other suitable technologies.

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

### 374 **5.2 Architecture**

375 Figure 1 shows the deployment architectural approach.

376



377

378

**Figure 1 – Easy Setup deployment architecture**

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

381 **5.3 Example scenario**

382 The following scenario presents a typical setup case.

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

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

391 **Easy setup overview**

392 **6.1 Introduction**

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

- 394 1) EasySetup Resource Type
- 395 2) WiFiConf Resource Type
- 396 3) DevConf Resource Type

397 Instances of these Resources Type (Resources) shall be excluded in the IDD for the Introspection

398 Resource (see clause 11.4 in ISO/IEC 30118-1).

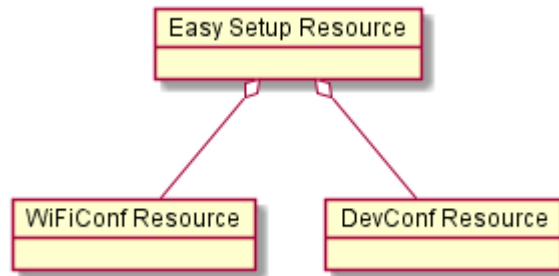
399 The EasySetup Resource Type is a Collection Resource and shall contain Links to instances of at

400 least WiFiConf and DevConf. A vendor may add links to other Resource Types. The relationship

401 between the EasySetup Resource Type and linked Resources is shown in Figure 2.

402 NOTE The EasySetup Resource Type supports the batch Interface ("oic.if.b") which allows for efficient data delivery

403 with a single request rather than multiple requests to each linked Resource.



404

405

**Figure 2 – Easy Setup Resource Types**

406 **6.2 EasySetup Resource**

407 **6.2.1 Overview**

408 The EasySetup Resource stores useful information including current status of Enrollee and last

409 error code which was produced in the process of Easy Setup.

410 **6.2.2 Resource**

411 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.easyssetup, 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.easyssetup" Resource Type. Complete details are  
415 provided in annex A.3.

416

**Table 2 – "oic.r.easyssetup" 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.

417

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

419 – "ps" equal to 0.

420 – "lec" equal to 0.

421 – "cn" equal to an empty array.

### 422 6.3 WiFiConf Resource Type

#### 423 6.3.1 Introduction

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

#### 426 6.3.2 Resource Type

427 The WiFiConf Resource Type is as defined in Table 3.

428

**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

429

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

431

432

**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")

433

## 434 6.4 DevConf Resource Type

### 435 6.4.1 Introduction

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

### 437 6.4.2 Resource Type

438 The DevConf Resource Type is as defined in Table 5

439

**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

440

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

443

**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					<p>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.</p> <p>The pre-configured device name is presented by enrollee to mediator during easy-setup process.</p>
--	--	-----------------	--	--	--	--	---

444

445 **eSIM Easy Setup Overview**

446 **7.1 Introduction**

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

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

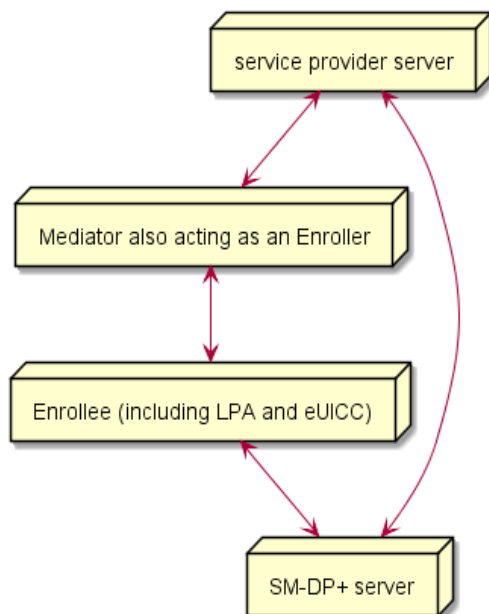
- 452 – Device and eUICC information, used to provide cellular plans to an end user
- 453 – Subscription information, comprising, e.g. Activation Code
- 454 – Progress information, indicating the status of the eSIM Easy Setup

455 **7.2 Architecture**

456 Figure 3 shows the deployment architectural approach.

457

458



459

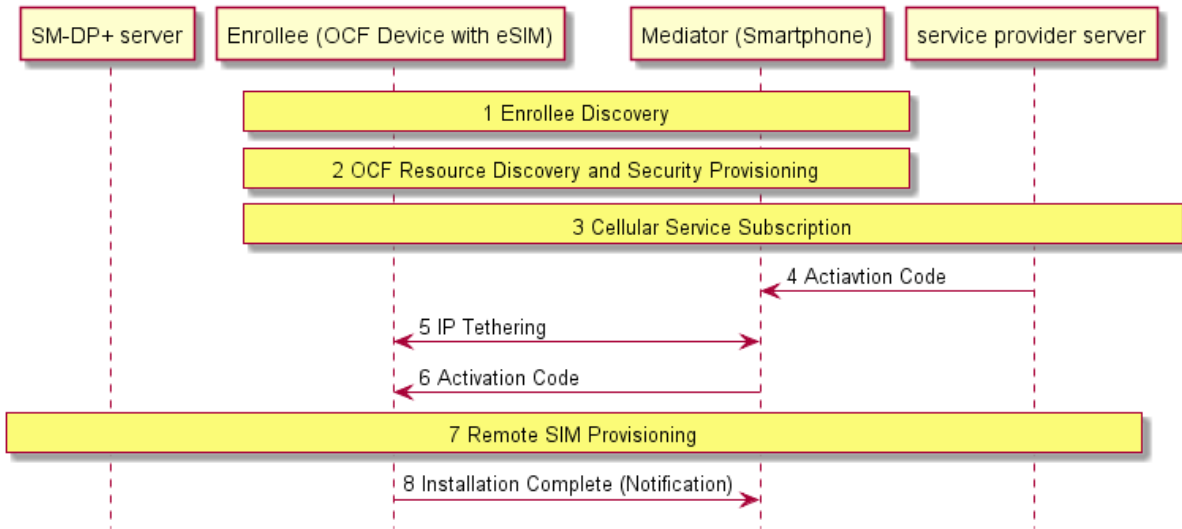
460 **Figure 3 – eSIM Easy Setup deployment architecture**

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

464 **7.3 Example Scenario**

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

467



468

469 **Figure 4 – eSIM Easy Setup example scenario**

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

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

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

## 496 7.4 eSIM Easy Setup Resource Model

### 497 7.4.1 Introduction

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

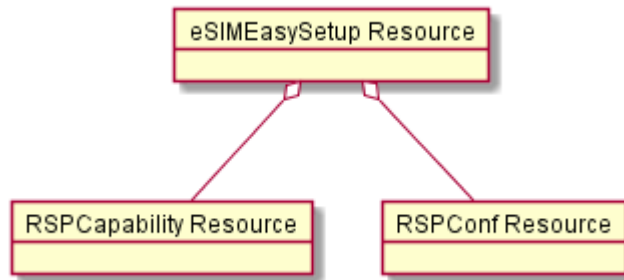
- 499 1) eSIMEasySetup Resource Type
- 500 2) RSPCapability Resource Type
- 501 3) RSPConf Resource Type

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

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

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

509



510

511 **Figure 5 – eSIM Easy Setup Resource Types**

## 512 7.5 eSIMEasySetup Resource Type

### 513 7.5.1 Introduction

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

### 516 7.5.2 Resource Type Definition

517 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

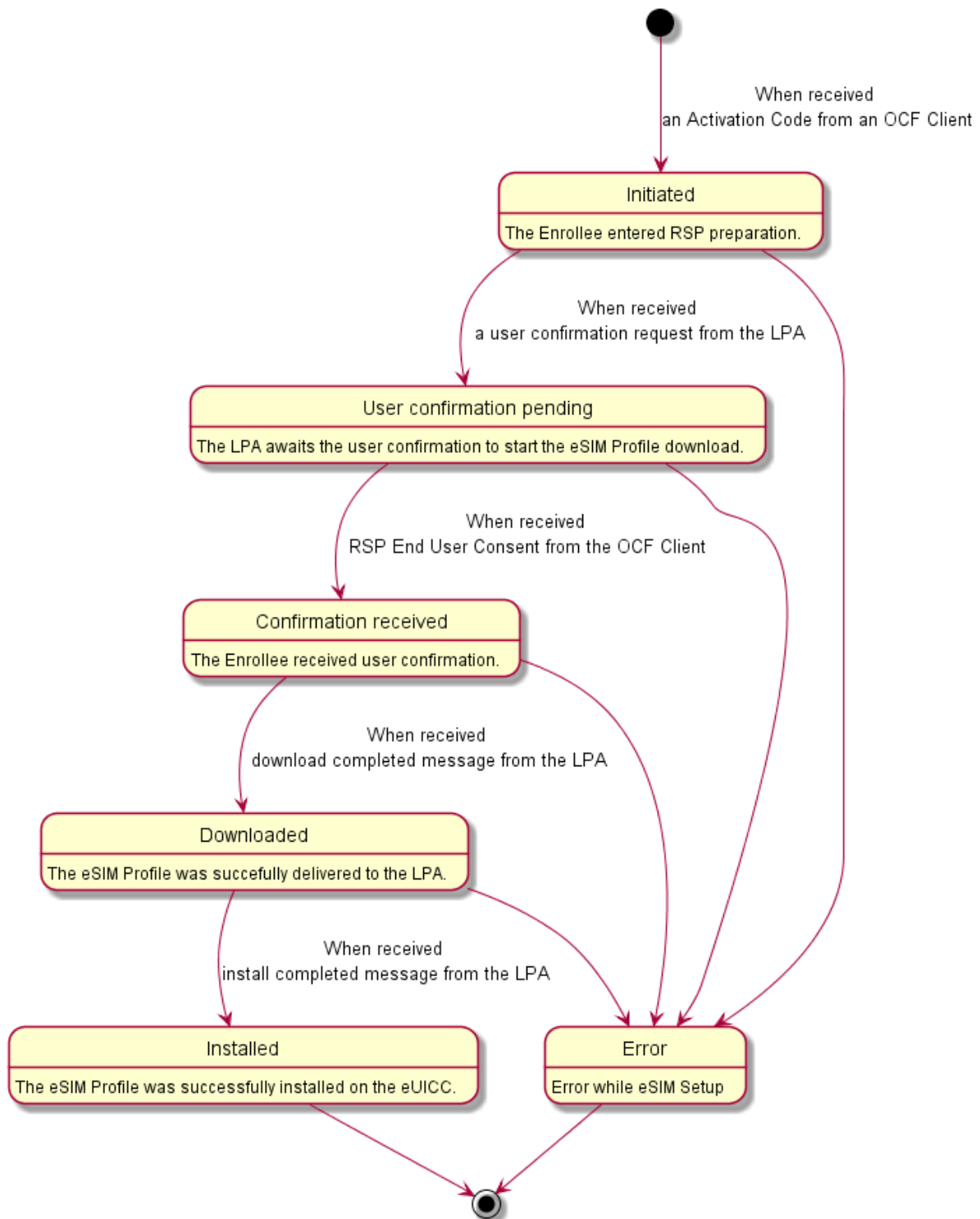
**Table 8 – "oic.r.esimeasysetup" Resource Type Definition**

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

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

536 **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

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

539 **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)

540 **7.6 RSPCapability Resource Type**

541 **7.6.1 Introduction**

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

544 **7.6.2 Resource Type Definition**

545 The RSPCapability Resource Type is as defined in Table 11.

546 **Table 11 – RSPCapability Resource Type**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/RSP CapabilityRes URI	RSPCapability	oic.r.rsppcapability	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.	
--	--	--	--	---	--

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

549 **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.

550 **7.7 RSPConf Resource Type**

551 **7.7.1 Introduction**

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

554 **7.7.2 Resource Type Definition**

555 The RSPConf Resource Type is as defined in Table 13.

556 **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

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

**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

## 562 **Network and connectivity**

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

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

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

## 569 **Functional interactions**

### 570 **9.1 Onboarding, Provisioning and Configuration**

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

### 573 **9.2 Resource discovery**

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

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

### 578 **9.3 Retrieving and updating Easy Setup Resources**

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

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

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

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

591 For details of Easy Setup Resources refer to clause 6.

### 592 **9.4 Error handling**

#### 593 **9.4.1 Easy Setup error handling**

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

- 597 – The Enrollee shall set "lec" Property to 1, if it fails to connect because it can't find the SSID.
- 598 – The Enrollee shall set "lec" Property to 2, if it fails to connect due to wrong credential (password)  
599 information.
- 600 – The Enrollee should set "lec" Property to 6, if the Auth type is not supported by the Enrollee.

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

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

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

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

#### 610 **9.4.2 eSIM Easy Setup error handling**

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

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

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

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

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

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

#### 622 **9.5 Example Easy Setup flow**

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

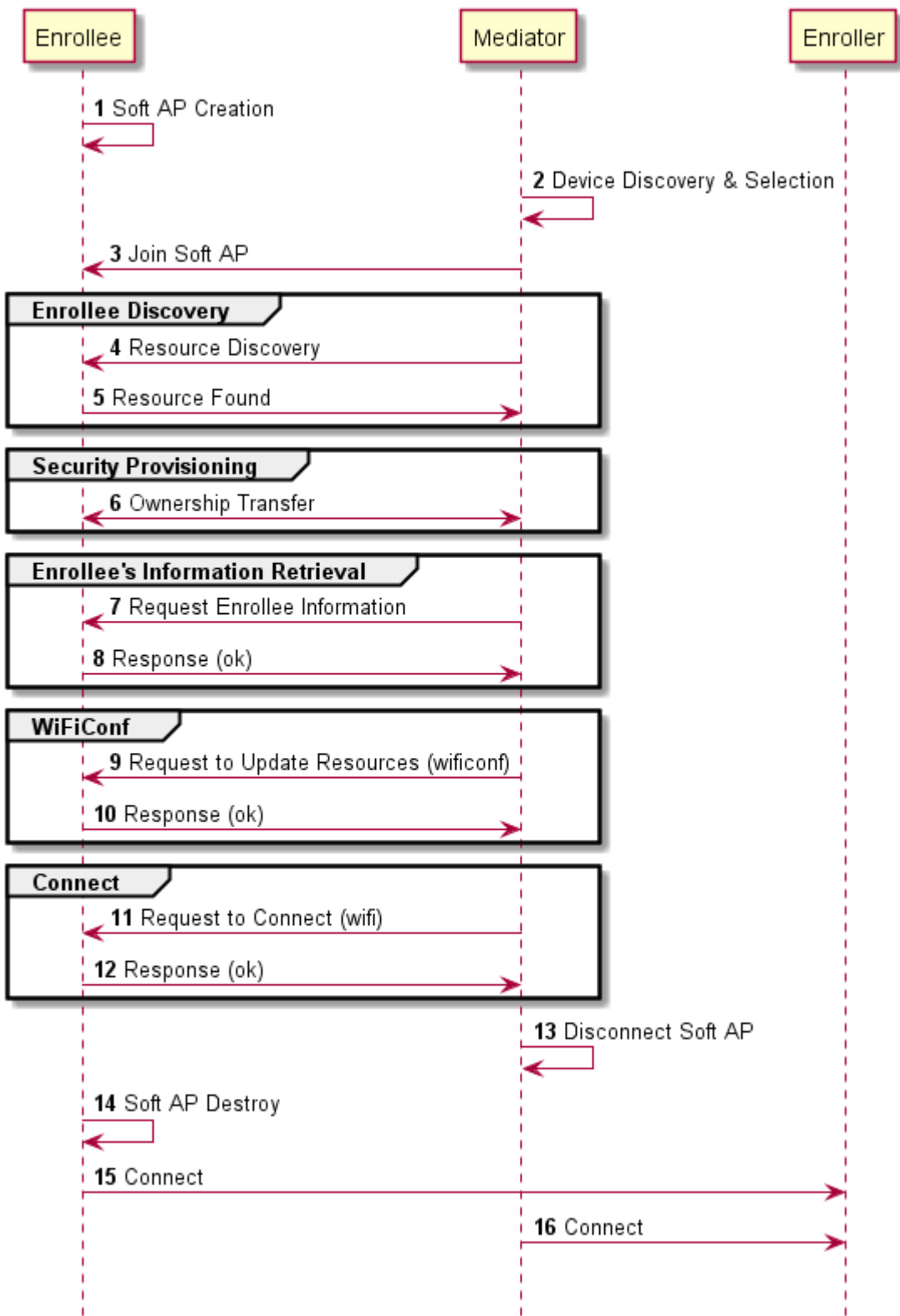
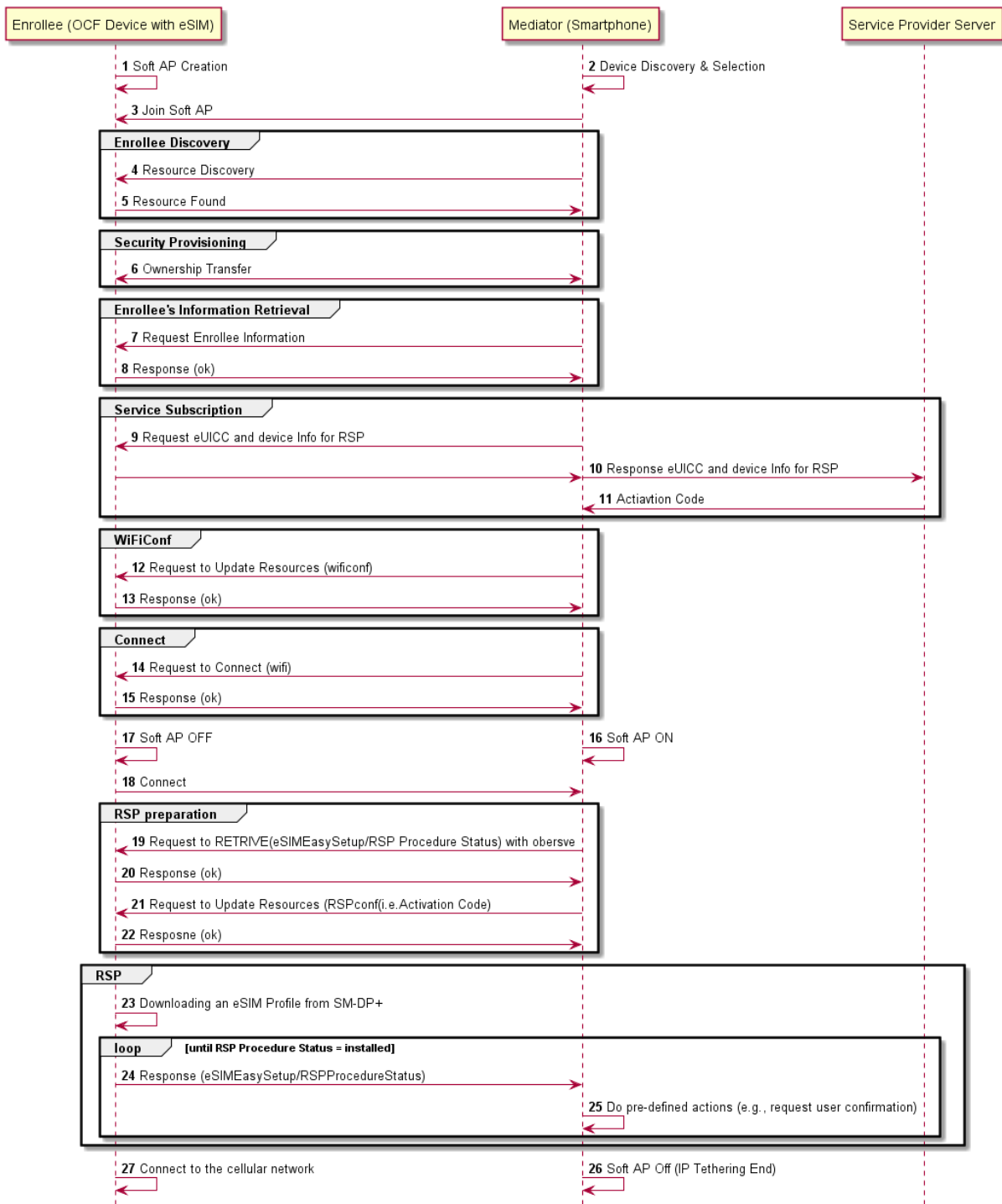


Figure 7 – Easy Setup Flow (Informative)

624  
625  
626

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

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



632

633

**Figure 8 – eSIM Easy Setup Flow (Informative)**

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

635 1. Enrollee turns on Soft AP for Easy Setup.

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

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

- 683 25. On receiving the NOTIFICATION, Mediator performs predefined actions. This is the expected  
684 procedure in the "loop" until RSP Procedure Status is set to "Installed":
- 685 a) After ES9+.AuthenticateClient(Success) returns to LPA, RSP Procedure Status ("ps")  
686 changes to "User confirmation pending", and the value change is Notified to the Mediator.
  - 687 b) The Mediator sends a RETRIEVE message to the RSPConf Resource to get Confirmation  
688 Code Required ("ccr") and eSIM Profile Metadata ("pm") Property values.
  - 689 c) Enrollee returns Confirmation Code Required ("ccr") value and eSIM Profile Metadata ("pm")  
690 value.
  - 691 d) Mediator displays the eSIM Profile Metadata ("pm") to get the end user consent, and request  
692 for Confirmation Code input if Confirmation Code Required ("ccr") sets to "True" in the  
693 RSPConf Resource.
  - 694 e) Mediator sends an UPDATE operation to the eSIMEasySetup Resource using the batch  
695 OCF Interface: RSP End User Consent ("euc") to the either "Download OK" or "Download  
696 and Enable OK", and Confirmation Code ("cc") to what the user entered if a confirmation  
697 code is required. Enrollee sets RSP Procedure Status ("ps") to "Confirmation received"  
698 when RSP End User Consent ("euc") is written.
  - 699 f) Download proceeds until it terminates at which point the Enrollee changes the RSP  
700 Procedure Status to "Downloaded" followed by "Installed" when the LPA receives  
701 ES9+.GetBoundProfilePackage(Success),ES9+.HandleNotification(Success) respectively.

702 26. If successfully "Installed", Mediator terminates the Soft AP, and then leaves eSIM Easy Setup  
703 mode.

704 27. Enrollee connects to the cellular network of the contracted mobile network operator.

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

## 707 **9.6 Easy Setup SSID tags**

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

- 710 – "OCF\_"  
711 – Prefix tag that has to be at the beginning of the SSID.  
712 – Example: OCF\_MySSID
- 713 – "\_OCF"  
714 – Suffix tag that has to be at the end of the SSID.  
715 – Example: MySSID\_OCf

716 These tags are case sensitive.

## 717 **9.7 Easy Setup information element**

### 718 **9.7.1 Overview**

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

### 724 **9.7.2 OCF Device information element (IE)**

725 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**

- 726
- 727 – Type is a unique id allocated by the IEEE registrar to identify different information elements
  - 728 from each other. The Easy Setup Information Element shall have a Type value of 221 which is
  - 729 standard vendor specific information element.
  - 730 – Length shall indicate the total size of CID, OCF IE Type, and Data in bytes.
  - 731 – Company ID (CID) is a unique 24-bit identifier for a specific company or organization. The Easy
  - 732 Setup Information Element shall have a CID value of 6A 40 65.
  - 733 – OCF IE Type is the identifier of the specific IE within OCF. The OCF IE Type shall be set to 0
  - 734 for Easy Setup.
  - 735 – Data is a set of type-length-value (TLV) structures that represent the device information in Table
  - 736 1. The length of this field shall be less than 252 bytes.

737

738 Each TLV has the structure shown in Figure 10.

1 byte	1 byte	<250 bytes
Type	Length	Value

**Figure 10 – Type-Length-Value structure**

739

- 740 – Type shall indicate the type of the field from Table 15.
  - 741 – Length shall indicate the length of the Value in bytes.
  - 742 – Value shall represent the corresponding information for specific TLV type from Table 15.
- 743 Data is a set of TLVs as defined in Table 15.

744

**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

745

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

#### 750 **9.7.2.1 Device friendly name (Type 1)**

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

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

#### 754 **9.7.2.2 Device Type (Type 2)**

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

757

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

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

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

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

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

#### 767 **9.7.2.3 Device manufacturer name (Type 3)**

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

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

#### 772 **9.7.2.4 Language tag (Type 4)**

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

777 Please see 9.7.2.8 for information on supporting multiple languages.

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

#### 780 **9.7.2.5 Protocol Independent ID (Type 5)**

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

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

#### 783 **9.7.2.6 Multiple information elements**

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

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

787 – Support for multiple languages is necessary.

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

#### 790 **9.7.2.7 IEC for large TLV size support**

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

794 – Type

795 – Length

796 – CID

797 – OCF IE Type

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

#### 800 **9.7.2.8 IEC for multiple language support**

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

### 805 **Security**

806 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.easyssetup"	A.3
Wi-Fi Configuration		"oic.r.wificonf"	A.4
eSIM Easy Setup		"oic.r. esimeasyssetup"	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" : {
```

```

844     "get": {
845         "description": "The Device configuration Resource stores Device settings such as the Device
846 name. Vendor-specific information can be added to the Resource.\nThe Device name is a human-friendly
847 name read by a Mediator during easy setup.\n",
848         "parameters": [
849             {"$ref": "#/parameters/interface"}
850         ],
851         "responses": {
852             "200": {
853                 "description" : "",
854                 "x-example": {
855                     "rt": ["oic.r.devconf"],
856                     "dn" : "My Refrigerator"
857                 },
858                 "schema": { "$ref": "#/definitions/DevConf" }
859             }
860         }
861     }
862 },
863 },
864 "parameters": {
865     "interface" : {
866         "in" : "query",
867         "name" : "if",
868         "type" : "string",
869         "enum" : ["oic.if.r", "oic.if.baseline"]
870     }
871 },
872 "definitions": {
873     "DevConf" : {
874         "properties": {
875             "rt" : {
876                 "description": "Resource Type of the Resource",
877                 "items": {
878                     "enum": ["oic.r.devconf"],
879                     "maxLength": 64,
880                     "type": "string"
881                 },
882                 "minItems": 1,
883                 "readOnly": true,
884                 "uniqueItems": true,
885                 "type": "array"
886             },
887             "n" : {
888                 "$ref":
889 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
890 schema.json#/definitions/n"
891             },
892             "id" : {
893                 "$ref":
894 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
895 schema.json#/definitions/id"
896             },
897             "if" : {
898                 "description": "The OCF Interfaces supported by this Resource",
899                 "items": {
900                     "enum": [
901                         "oic.if.r",
902                         "oic.if.baseline"
903                     ],
904                     "type": "string",
905                     "maxLength": 64
906                 },
907                 "minItems": 2,
908                 "readOnly": true,
909                 "uniqueItems": true,
910                 "type": "array"
911             },
912             "dn": {
913                 "oneOf": [

```

```

914     {
915         "type": "string",
916         "description": "Indicates a pre-configured Device name in language indicated by 'dl'
917 in /oic/con: presented by an Enrollee Device to a Mediator Device during the easy-setup process",
918         "pattern": "^.*$",
919         "readOnly": true
920     },
921     {
922         "type": "array",
923         "items": {
924             "type": "object",
925             "properties": {
926                 "language": {
927                     "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
928 schema.json#/definitions/language-tag",
929                     "readOnly": true,
930                     "description": "An RFC 5646 language tag."
931                 },
932                 "value": {
933                     "type": "string",
934                     "description": "Pre-configured Device name in the indicated language.",
935                     "pattern": "^.*$",
936                     "readOnly": true
937                 }
938             }
939         },
940         "minItems" : 1,
941         "readOnly": true,
942         "description": "Localized device name."
943     }
944 ]
945 }
946 },
947 "type" : "object",
948 "required": ["dn"]
949 }
950 }
951 }
952

```

### 953 A.2.5 Property definition

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

955 **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	

### 956 A.2.6 CRUDN behaviour

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

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

Create	Read	Update	Delete	Notify
	get			observe

959 **A.3 Easy Setup Collection**

960 **A.3.1 Introduction**

961 The Easy Setup Resource stores useful information including the current status of unboxing a  
 962 Device and the last error code which are produced in the process of easy setup.  
 963 Note that the Easy Setup Resource is a Collection Resource, which contains Links to WiFiConf,  
 964 and DevConf Resources and may additionally contain Links to other Resources.  
 965

966 **A.3.2 Example URI**

967 /EasySetupResURI

968 **A.3.3 Resource type**

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

970 **A.3.4 OpenAPI 2.0 definition**

```

971 {
972   "swagger": "2.0",
973   "info": {
974     "title": "Easy Setup Collection",
975     "version": "2019-03-27",
976     "license": {
977       "name": "OCF Data Model License",
978       "url":
979         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
980         CENSE.md",
981       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
982     },
983     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
984   },
985   "schemes": ["http"],
986   "consumes": ["application/json"],
987   "produces": ["application/json"],
988   "paths": {
989     "/EasySetupResURI?if=oic.if.ll" : {
990       "get": {
991         "description": "The Easy Setup Resource stores useful information including the current
992         status of unboxing a Device and the last error code which are produced in the process of easy
993         setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
994         WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
995         "parameters": [
996           {"$ref": "#/parameters/interface-all"}
997         ],
998         "responses": {
999           "200": {
1000             "description": "",
1001             "x-example":
1002               [
1003                 {
1004                   "href": "/EasySetupResURI",
1005                   "rt": ["oic.r.easyssetup", "oic.wk.col"],
1006                   "if": ["oic.if.b"],
1007                   "p": {"bm": 3},
1008                   "eps": [
1009                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1010                   ],
1011                   "rel": ["self", "item"]

```

```

1012     },
1013     {
1014         "href": "/WiFiConfResURI",
1015         "rt": ["oic.r.wificonf"],
1016         "if": ["oic.if.baseline"],
1017         "p":{"bm":3},
1018         "eps": [
1019             {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1020         ]
1021     },
1022     {
1023         "href": "/DevConfResURI",
1024         "rt": ["oic.r.devconf"],
1025         "if": ["oic.if.baseline"],
1026         "p":{"bm":3},
1027         "eps": [
1028             {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1029         ]
1030     }
1031 ],
1032 "schema": { "$ref": "#/definitions/slinks" }
1033 }
1034 }
1035 }
1036 },
1037 "/EasySetupResURI?if=oic.if.b" : {
1038     "get": {
1039         "description": "The Easy Setup Resource stores useful information including the current
1040 status of unboxing a Device and the last error code which are produced in the process of easy
1041 setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
1042 WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
1043         "parameters": [
1044             {"$ref": "#/parameters/interface-all"}
1045         ],
1046         "responses": {
1047             "200": {
1048                 "description" : "",
1049                 "x-example":
1050                 [
1051                     {
1052                         "href": "/EasySetupResURI",
1053                         "rep":{"
1054                             "ps" : 0,
1055                             "lec": 0,
1056                             "cn": [1]
1057                         }
1058                     },
1059                     {
1060                         "href": "/WiFiConfResURI",
1061                         "rep":{"
1062                             "swmt" : ["A", "B", "G"],
1063                             "swf": ["2.4G", "5G"],
1064                             "tnn": "Home_AP_SSID",
1065                             "cd": "Home_AP_PWD",
1066                             "wat": "WPA2_PSK",
1067                             "wet": "AES",
1068                             "swat": ["WPA_PSK", "WPA2_PSK"],
1069                             "swet": ["TKIP", "AES", "TKIP_AES"]
1070                         }
1071                     },
1072                     {
1073                         "href": "/DevConfResURI",
1074                         "rep":{"
1075                             "dn" : "My Refrigerator"
1076                         }
1077                     }
1078                 ],
1079                 "schema": { "$ref": "#/definitions/sbatch" }
1080             }
1081         }

```

```

1082     },
1083     "post": {
1084         "description": "Able to deliver Wi-Fi, Device configuration and other
1085 configuration\ninformation in a batch by utilizing 'batch' OCF Interface.\nIf you want to deliver
1086 Wi-Fi and Device configuration information in a batch,\nyou can write all Properties you want to
1087 send with a 'batch' OCF Interface.\nThe below example is the case to send Easy Setup and Wi-Fi
1088 configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",
1089         "parameters": [
1090             { "$ref": "#/parameters/interface-update" },
1091             {
1092                 "name": "body",
1093                 "in": "body",
1094                 "required": true,
1095                 "schema": { "$ref": "#/definitions/sbatch-update" },
1096                 "x-example":
1097                     [
1098                         {
1099                             "href": "/EasySetupResURI",
1100                             "rep": {
1101                                 "cn": [1]
1102                             }
1103                         },
1104                         {
1105                             "href": "/WiFiConfResURI",
1106                             "rep": {
1107                                 "tnn": "Home_AP_SSID",
1108                                 "cd": "Home_AP_PWD",
1109                                 "wat": "WPA2_PSK",
1110                                 "wet": "AES"
1111                             }
1112                         }
1113                     ]
1114             }
1115         ],
1116         "responses": {
1117             "200": {
1118                 "description": "",
1119                 "x-example":
1120                     [
1121                         {
1122                             "href": "/EasySetupResURI",
1123                             "rep": {
1124                                 "ps": 0,
1125                                 "lec": 0,
1126                                 "cn": [1]
1127                             }
1128                         },
1129                         {
1130                             "href": "/WiFiConfResURI",
1131                             "rep": {
1132                                 "swmt": ["A", "B", "G"],
1133                                 "swf": ["2.4G", "5G"],
1134                                 "tnn": "Home_AP_SSID",
1135                                 "cd": "Home_AP_PWD",
1136                                 "wat": "WPA2_PSK",
1137                                 "wet": "AES",
1138                                 "swat": ["WPA_PSK", "WPA2_PSK"],
1139                                 "swet": ["TKIP", "AES", "TKIP_AES"]
1140                             }
1141                         },
1142                         {
1143                             "href": "/DevConfResURI",
1144                             "rep": {
1145                                 "dn": "My Refrigerator"
1146                             }
1147                         }
1148                     ],
1149                 "schema": { "$ref": "#/definitions/sbatch" }
1150             }
1151         }

```

```

1152     }
1153   },
1154   "/EasySetupResURI?if=oic.if.baseline" : {
1155     "get": {
1156       "description": "The Easy Setup Resource stores useful information including the current
1157 status of unboxing a Device and the last error code which are produced in the process of easy
1158 setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
1159 WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
1160       "parameters": [
1161         { "$ref": "#/parameters/interface-all" }
1162       ],
1163       "responses": {
1164         "200": {
1165           "description" : "",
1166           "x-example":
1167             {
1168               "rt" : ["oic.r.easyssetup", "oic.wk.col"],
1169               "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
1170               "ps" : 0,
1171               "lec": 0,
1172               "cn": [1],
1173               "links": [
1174                 {
1175                   "href": "/EasySetupResURI",
1176                   "rt": ["oic.r.easyssetup", "oic.wk.col"],
1177                   "if": ["oic.if.b"],
1178                   "p":{"bm":3},
1179                   "eps": [
1180                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1181                   ],
1182                   "rel":["self", "item"]
1183                 },
1184                 {
1185                   "href": "/WiFiConfResURI",
1186                   "rt": ["oic.r.wificonf"],
1187                   "if": ["oic.if.baseline"],
1188                   "p":{"bm":3},
1189                   "eps": [
1190                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1191                   ]
1192                 },
1193                 {
1194                   "href": "/DevConfResURI",
1195                   "rt": ["oic.r.devconf"],
1196                   "if": ["oic.if.baseline"],
1197                   "p":{"bm":3},
1198                   "eps": [
1199                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1200                   ]
1201                 }
1202               ]
1203             },
1204           "schema": { "$ref": "#/definitions/EasySetup" }
1205         }
1206       }
1207     },
1208     "post": {
1209       "description": "Able to update connection type to attempt to connect to the Enroller to
1210 start during while posting to /EasySetupResURI\nThe below example is the case to send Easy Setup
1211 configuration\n(i.e. connection type) in a post.\n",
1212       "parameters": [
1213         { "$ref": "#/parameters/interface-update" },
1214         {
1215           "name": "body",
1216           "in": "body",
1217           "required": true,
1218           "schema": { "$ref": "#/definitions/EasySetupUpdate" },
1219           "x-example":
1220             {
1221               "cn": [1]

```

```

1222     }
1223   }
1224 ],
1225 "responses": {
1226   "200": {
1227     "description" : "",
1228     "x-example":
1229     {
1230       "rt" : ["oic.r.easyssetup", "oic.wk.col"],
1231       "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
1232       "ps" : 0,
1233       "lec": 0,
1234       "cn": [1],
1235       "links": [
1236         {
1237           "href": "/EasySetupResURI",
1238           "rt": ["oic.r.easyssetup", "oic.wk.col"],
1239           "if": ["oic.if.b", "oic.if.ll", "oic.if.baseline"],
1240           "p":{"bm":3},
1241           "eps": [
1242             {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1243           ],
1244           "rel":["self", "item"]
1245         },
1246         {
1247           "href": "/WiFiConfResURI",
1248           "rt": ["oic.r.wificonf"],
1249           "if": ["oic.if.rw", "oic.if.baseline"],
1250           "p":{"bm":3},
1251           "eps": [
1252             {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1253           ]
1254         },
1255         {
1256           "href": "/DevConfResURI",
1257           "rt": ["oic.r.devconf"],
1258           "if": ["oic.if.r", "oic.if.baseline"],
1259           "p":{"bm":3},
1260           "eps": [
1261             {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1262           ]
1263         }
1264       ]
1265     },
1266     "schema": { "$ref": "#/definitions/EasySetup" }
1267   }
1268 }
1269 }
1270 }
1271 },
1272 "parameters": {
1273   "interface-all" : {
1274     "in" : "query",
1275     "name" : "if",
1276     "type" : "string",
1277     "enum" : ["oic.if.ll","oic.if.b","oic.if.baseline"]
1278   },
1279   "interface-update" : {
1280     "in" : "query",
1281     "name" : "if",
1282     "type" : "string",
1283     "enum" : ["oic.if.b", "oic.if.baseline"]
1284   }
1285 },
1286 "definitions": {
1287   "oic.oic-link": {
1288     "type": "object",
1289     "properties": {
1290       "anchor": {
1291         "$ref":

```

```

1292 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1293 schema.json#/definitions/anchor"
1294   },
1295   "di": {
1296     "$ref":
1297     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1298     schema.json#/definitions/di"
1299   },
1300   "eps": {
1301     "$ref":
1302     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1303     schema.json#/definitions/eps"
1304   },
1305   "href": {
1306     "$ref":
1307     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1308     schema.json#/definitions/href"
1309   },
1310   "ins": {
1311     "$ref":
1312     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1313     schema.json#/definitions/ins"
1314   },
1315   "p": {
1316     "$ref":
1317     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1318     schema.json#/definitions/p"
1319   },
1320   "rel": {
1321     "$ref":
1322     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1323     schema.json#/definitions/rel_array"
1324   },
1325   "title": {
1326     "$ref":
1327     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1328     schema.json#/definitions/title"
1329   },
1330   "type": {
1331     "$ref":
1332     "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1333     schema.json#/definitions/type"
1334   },
1335   "if": {
1336     "description": "The OCF Interfaces supported by the target Resource",
1337     "items": {
1338       "enum": [
1339         "oic.if.baseline",
1340         "oic.if.ll",
1341         "oic.if.b",
1342         "oic.if.r",
1343         "oic.if.rw"
1344       ],
1345       "type": "string",
1346       "maxLength": 64
1347     },
1348     "minItems": 1,
1349     "uniqueItems": true,
1350     "type": "array"
1351   },
1352   "rt": {
1353     "description": "Resource Type of the target Resource",
1354     "items": {
1355       "maxLength": 64,
1356       "type": "string"
1357     },
1358     "minItems": 1,
1359     "uniqueItems": true,
1360     "type": "array"
1361   }

```

```

1362     },
1363     "required": [
1364         "href",
1365         "rt",
1366         "if"
1367     ]
1368 },
1369 "slinks" : {
1370     "type": "array",
1371     "items": {
1372         "$ref": "#/definitions/oic.oic-link"
1373     }
1374 },
1375 "sbatch" : {
1376     "minItems" : 1,
1377     "items" : {
1378         "additionalProperties": true,
1379         "properties": {
1380             "href": {
1381                 "$ref":
1382 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1383 schema.json#/definitions/href"
1384             },
1385             "rep": {
1386                 "description": "The response payload from a single Resource",
1387                 "type": "object",
1388                 "anyOf": [
1389                     {
1390                         "$ref": "#/definitions/EasySetup"
1391                     },
1392                     {
1393                         "$ref": "https://openconnectivityfoundation.github.io/core-
1394 extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConf"
1395                     },
1396                     {
1397                         "$ref": "https://openconnectivityfoundation.github.io/core-
1398 extensions/swagger2.0/oic.r.devconf.swagger.json#/definitions/DevConf"
1399                     }
1400                 ]
1401             }
1402         },
1403         "required": [
1404             "href",
1405             "rep"
1406         ],
1407         "type": "object"
1408     },
1409     "type" : "array"
1410 },
1411 "sbatch-update" : {
1412     "minItems" : 1,
1413     "items" : {
1414         "additionalProperties": true,
1415         "description": "Array of Resource representations to apply to the batch Collection, using
1416 href to indicate which resource(s) in the batch to update. If the href Property is empty,
1417 effectively making the URI reference to the Collection itself, the representation is to be applied
1418 to all Resources in the batch",
1419         "properties": {
1420             "href": {
1421                 "$ref":
1422 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1423 schema.json#/definitions/href"
1424             },
1425             "rep": {
1426                 "description": "The response payload from a single Resource",
1427                 "type": "object",
1428                 "anyOf": [
1429                     {
1430                         "$ref": "#/definitions/EasySetupUpdate"
1431                     }

```

```

1432         {
1433             "$ref": "https://openconnectivityfoundation.github.io/core-
1434 extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConfUpdate"
1435         }
1436     ]
1437 }
1438 },
1439 "required": [
1440     "href",
1441     "rep"
1442 ],
1443 "type": "object"
1444 },
1445 "type" : "array"
1446 },
1447 "EasySetup" : {
1448     "properties": {
1449         "n" : {
1450             "$ref":
1451 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1452 schema.json#/definitions/n"
1453         },
1454         "rts" : {
1455             "description": "Resource Type of the Resources within the Collection",
1456             "items": {
1457                 "maxLength": 64,
1458                 "type": "string"
1459             },
1460             "minItems": 1,
1461             "uniqueItems": true,
1462             "readOnly": true,
1463             "type": "array"
1464         },
1465         "id" : {
1466             "$ref":
1467 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1468 schema.json#/definitions/id"
1469         },
1470         "rts-m" : {
1471             "description": "Resource Type of the mandatory Resources within the Collection",
1472             "items": {
1473                 "maxLength": 64,
1474                 "type": "string"
1475             },
1476             "minItems": 1,
1477             "uniqueItems": true,
1478             "readOnly": true,
1479             "type": "array"
1480         },
1481         "if" : {
1482             "description": "The OCF Interfaces supported by this Resource",
1483             "items": {
1484                 "enum": [
1485                     "oic.if.ll",
1486                     "oic.if.baseline",
1487                     "oic.if.b"
1488                 ],
1489                 "type": "string",
1490                 "maxLength": 64
1491             },
1492             "minItems": 2,
1493             "uniqueItems": true,
1494             "readOnly": true,
1495             "type": "array"
1496         },
1497         "rt" : {
1498             "items": {
1499                 "enum": [
1500                     "oic.r.easysetup",
1501                     "oic.wk.col"

```

```

1502     ],
1503     "type": "string",
1504     "maxLength": 64
1505   },
1506   "minItems": 2,
1507   "type": "array",
1508   "uniqueItems": true
1509 },
1510 "ps" : {
1511   "description": "Indicates the easy setup status of the Device. (0: Need to Setup, 1:
1512 Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved,
1513 255: EOF)",
1514   "enum": [
1515     0,
1516     1,
1517     2,
1518     3
1519   ],
1520   "readOnly": true,
1521   "type": "integer"
1522 },
1523 "lec" : {
1524   "description": "Indicates a failure reason (0: No error, 1: A given SSID is not found, 2:
1525 Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6:
1526 Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the
1527 Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi
1528 Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown
1529 error)",
1530   "enum": [
1531     0,
1532     1,
1533     2,
1534     3,
1535     4,
1536     5,
1537     6,
1538     7,
1539     8,
1540     9,
1541     255
1542   ],
1543   "readOnly": true,
1544   "type": "integer"
1545 },
1546 "cn" : {
1547   "description": "Indicates an array of connection types that trigger an attempt to connect
1548 to the Enroller to start.",
1549   "items": {
1550     "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1551 to be added in future (e.g. Connect to cloud / BLE))",
1552     "type": "integer"
1553   },
1554   "type": "array"
1555 },
1556 "links" : {
1557   "type": "array",
1558   "description": "A set of OCF Links.",
1559   "items": {
1560     "$ref": "#/definitions/oic.oic-link"
1561   }
1562 }
1563 },
1564 "type" : "object",
1565 "required": ["ps","lec","cn"]
1566 },
1567 "EasySetupUpdate" : {
1568   "additionalProperties": true,
1569   "description": "Update to writeable values in EasySetupResURI",
1570   "properties": {
1571     "cn" : {

```

```

1572         "description": "Indicates an array of connection types that trigger an attempt to connect
1573 to the Enroller to start.",
1574         "items": {
1575             "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1576 to be added in future (e.g. Connect to cloud / BLE))",
1577             "type": "integer"
1578         },
1579         "type": "array"
1580     },
1581 },
1582     "required": [
1583         "cn"
1584     ],
1585     "type": "object"
1586 }
1587 }
1588 }
1589

```

### 1590 A.3.5 Property definition

1591 Table A.4 defines the Properties that are part of the "oic.r.easyssetup, oic.wk.col" Resource Type.

1592 **Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.easyssetup,**  
1593 **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.
----	-------------------	-----	------------	---

1594 **A.3.6 CRUDN behaviour**

1595 Table A.5 defines the CRUDN operations that are supported on the "oic.r.easyssetup, oic.wk.col"  
1596 Resource Type.

1597 **Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.easyssetup,**  
1598 **oic.wk.col".**

Create	Read	Update	Delete	Notify
	get	post		observe

1599 **A.4 Wi-Fi Configuration**

1600 **A.4.1 Introduction**

1601 WiFiConf Resource stores essential information to help an unboxing Device  
1602 to connect to an existing Wi-Fi AP.

1604 **A.4.2 Example URI**

1605 /WiFiConfResURI

1606 **A.4.3 Resource type**

1607 The Resource Type is defined as: "oic.r.wificonf".

1608 **A.4.4 OpenAPI 2.0 definition**

```

1609 {
1610   "swagger": "2.0",
1611   "info": {
1612     "title": "Wi-Fi Configuration",
1613     "version": "2019-03-27",
1614     "license": {
1615       "name": "OCF Data Model License",
1616       "url":
1617 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
1618 CENSE.md",
1619       "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
1620     },
1621     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
1622   },
1623   "schemes": ["http"],
1624   "consumes": ["application/json"],
1625   "produces": ["application/json"],
1626   "paths": {
1627     "/WiFiConfResURI?if=oic.if.rw" : {
1628       "get": {
1629         "description": "The WiFiConf Resource stores essential information to help an unboxing
1630 Device connect to an existing Wi-Fi AP.\n",
1631         "parameters": [
1632           {"$ref": "#/parameters/interface-all"}
1633         ],
1634         "responses": {
1635           "200": {
1636             "description" : "",
1637             "x-example":

```

```

1638         {
1639             "tnn": "Home_AP_SSID",
1640             "swmt" : ["A", "B", "G"],
1641             "swf": ["2.4G", "5G"],
1642             "cd": "Home_AP_PWD",
1643             "wat": "WPA2_PSK",
1644             "wet": "AES",
1645             "swat": ["WPA_PSK", "WPA2_PSK"],
1646             "swet": ["TKIP", "AES", "TKIP_AES"]
1647         },
1648         "schema": { "$ref": "#/definitions/WiFiConf" }
1649     }
1650 },
1651 },
1652 "post": {
1653     "description": "Deliver Wi-Fi AP's information for an unboxing Device to connect to it.\n",
1654     "parameters": [
1655         { "$ref": "#/parameters/interface-all" },
1656         {
1657             "name": "body",
1658             "in": "body",
1659             "required": true,
1660             "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1661             "x-example":
1662                 {
1663                     "tnn": "Home_AP_SSID",
1664                     "cd": "Home_AP_PWD",
1665                     "wat": "WPA2_PSK",
1666                     "wet": "AES"
1667                 }
1668         }
1669     ],
1670     "responses": {
1671         "200": {
1672             "description" : "",
1673             "x-example":
1674                 {
1675                     "tnn": "Home_AP_SSID",
1676                     "swmt" : ["A", "B", "G"],
1677                     "swf": ["2.4G", "5G"],
1678                     "cd": "Home_AP_PWD",
1679                     "wat": "WPA2_PSK",
1680                     "wet": "AES",
1681                     "swat": ["WPA_PSK", "WPA2_PSK"],
1682                     "swet": ["TKIP", "AES", "TKIP_AES"]
1683                 },
1684             "schema": { "$ref": "#/definitions/WiFiConf" }
1685         }
1686     }
1687 },
1688 },
1689 "/WiFiConfResURI?if=oic.if.baseline" : {
1690     "get": {
1691         "description": "WiFiConf Resource stores essential information to help an unboxing
1692 Device\nto connect to an existing Wi-Fi AP.\n",
1693         "parameters": [
1694             { "$ref": "#/parameters/interface-all" }
1695         ],
1696         "responses": {
1697             "200": {
1698                 "description" : "",
1699                 "x-example":
1700                     {
1701                         "rt": ["oic.r.wificonf"],
1702                         "if": ["oic.if.rw", "oic.if.baseline"],
1703                         "swmt" : ["A", "B", "G"],
1704                         "swf": ["2.4G", "5G"],
1705                         "tnn": "Home_AP_SSID",
1706                         "cd": "Home_AP_PWD",
1707                         "wat": "WPA2_PSK",

```

```

1708         "wet": "TKIP",
1709         "swat": ["WPA_PSK", "WPA2_PSK"],
1710         "swet": ["TKIP", "AES", "TKIP_AES"]
1711     },
1712     "schema": { "$ref": "#/definitions/WiFiConf" }
1713 }
1714 },
1715 },
1716 "post": {
1717     "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
1718     "parameters": [
1719         {"$ref": "#/parameters/interface-all"},
1720         {
1721             "name": "body",
1722             "in": "body",
1723             "required": true,
1724             "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1725             "x-example":
1726                 {
1727                     "tnn": "Home_AP_SSID",
1728                     "cd": "Home_AP_PWD",
1729                     "wat": "WPA2_PSK",
1730                     "wet": "AES"
1731                 }
1732         }
1733     ],
1734     "responses": {
1735         "200": {
1736             "description": "",
1737             "x-example":
1738                 {
1739                     "rt": ["oic.r.wificonf"],
1740                     "if": ["oic.if.rw", "oic.if.baseline"],
1741                     "tnn": "Home_AP_SSID",
1742                     "swmt": ["A", "B", "G"],
1743                     "swf": ["2.4G", "5G"],
1744                     "cd": "Home_AP_PWD",
1745                     "wat": "WPA2_PSK",
1746                     "wet": "AES",
1747                     "swat": ["WPA_PSK", "WPA2_PSK"],
1748                     "swet": ["TKIP", "AES", "TKIP_AES"]
1749                 },
1750             "schema": { "$ref": "#/definitions/WiFiConf" }
1751         }
1752     }
1753 },
1754 },
1755 },
1756 "parameters": {
1757     "interface-all": {
1758         "in": "query",
1759         "name": "if",
1760         "type": "string",
1761         "enum": ["oic.if.rw", "oic.if.baseline"]
1762     }
1763 },
1764 "definitions": {
1765     "WiFiConf": {
1766         "properties": {
1767             "rt": {
1768                 "description": "Resource Type of the Resource",
1769                 "items": {
1770                     "enum": ["oic.r.wificonf"],
1771                     "type": "string",
1772                     "maxLength": 64
1773                 },
1774                 "minItems": 1,
1775                 "uniqueItems": true,
1776                 "readOnly": true,
1777                 "type": "array"

```

```

1778     },
1779     "tnn" : {
1780         "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1781         "pattern": "^.*$",
1782         "type": "string"
1783     },
1784     "swmt" : {
1785         "description": "Indicates supported Wi-Fi mode types. It can be multiple",
1786         "items": {
1787             "description": "Supported Wi-Fi Mode Type.",
1788             "enum": [
1789                 "A",
1790                 "B",
1791                 "G",
1792                 "N",
1793                 "AC"
1794             ],
1795             "type": "string"
1796         },
1797         "readOnly": true,
1798         "type": "array"
1799     },
1800     "wat" : {
1801         "description": "Indicates Wi-Fi Auth Type",
1802         "enum": [
1803             "None",
1804             "WEP",
1805             "WPA_PSK",
1806             "WPA2_PSK"
1807         ],
1808         "type": "string"
1809     },
1810     "n" : {
1811         "$ref":
1812         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1813         schema.json#/definitions/n"
1814     },
1815     "swat" : {
1816         "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
1817         "items": {
1818             "description": "Indicates Wi-Fi Auth Type",
1819             "enum": [
1820                 "None",
1821                 "WEP",
1822                 "WPA_PSK",
1823                 "WPA2_PSK"
1824             ],
1825             "type": "string"
1826         },
1827         "readOnly": true,
1828         "type": "array"
1829     },
1830     "swf" : {
1831         "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
1832         Valid values are ('2.4G', '5G')",
1833         "items": {
1834             "pattern": "^(2\\.4|5)G$",
1835             "type": "string"
1836         },
1837         "readOnly": true,
1838         "type": "array"
1839     },
1840     "swet" : {
1841         "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
1842         "items": {
1843             "description": "Indicates Wi-Fi Encryption Type",
1844             "enum": [
1845                 "None",
1846                 "WEP_64",
1847                 "WEP_128",

```

```

1848         "TKIP",
1849         "AES",
1850         "TKIP_AES"
1851     ],
1852     "type": "string"
1853 },
1854     "readOnly": true,
1855     "type": "array"
1856 },
1857 "wet" : {
1858     "description": "Indicates Wi-Fi Encryption Type",
1859     "enum": [
1860         "None",
1861         "WEP_64",
1862         "WEP_128",
1863         "TKIP",
1864         "AES",
1865         "TKIP_AES"
1866     ],
1867     "type": "string"
1868 },
1869 "cd" : {
1870     "description": "Indicates credential information of Wi-Fi AP",
1871     "pattern": "^.*$",
1872     "type": "string"
1873 },
1874 "id" : {
1875     "$ref":
1876     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1877     schema.json#/definitions/id"
1878 },
1879 "if" : {
1880     "description": "The OCF Interfaces supported by this Resource",
1881     "items": {
1882         "enum": [
1883             "oic.if.rw",
1884             "oic.if.baseline"
1885         ],
1886         "type": "string",
1887         "maxLength": 64
1888     },
1889     "minItems": 2,
1890     "uniqueItems": true,
1891     "readOnly": true,
1892     "type": "array"
1893 }
1894 },
1895 "type" : "object",
1896 "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
1897 },
1898 "WiFiConfUpdate" : {
1899     "properties": {
1900         "wat" : {
1901             "description": "Indicates Wi-Fi Auth Type",
1902             "enum": [
1903                 "None",
1904                 "WEP",
1905                 "WPA_PSK",
1906                 "WPA2_PSK"
1907             ]
1908         },
1909         "cd" : {
1910             "description": "Indicates credential information of Wi-Fi AP",
1911             "pattern": "^.*$",
1912             "type": "string"
1913         },
1914         "wet" : {
1915             "description": "Indicates Wi-Fi Encryption Type",
1916             "enum": [
1917                 "None",

```

```

1918         "WEP_64",
1919         "WEP_128",
1920         "TKIP",
1921         "AES",
1922         "TKIP_AES"
1923     ]
1924 },
1925     "tnn" : {
1926         "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1927         "pattern": "^.*$",
1928         "type": "string"
1929     }
1930 },
1931     "type" : "object",
1932     "required":["tnn", "wat", "wet"]
1933 }
1934 }
1935 }
1936

```

#### 1937 A.4.5 Property definition

1938 Table A.6 defines the Properties that are part of the "oic.r.wificonf" Resource Type.

1939 **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
ttn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP)

1940 **A.4.6 CRUDN behaviour**

1941 Table A.7 defines the CRUDN operations that are supported on the "oic.r.wificonf" Resource Type.

1942 **Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.wificonf".**

Create	Read	Update	Delete	Notify
	get	post		observe

1943 **A.5 eSIM Easy Setup Collection**

1944 **A.5.1 Introduction**

1945 The eSIMEasySetup Resource Type stores useful information including Remote SIM Provisioning (RSP) status,

1946 and RSP last error code which was produced in the process of eSIM Easy Setup.

1947 Note that the eSIM Easy Setup Resource is a Collection Resource, which contains Links to

1948 RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.

1949

1951 **A.5.2 Example URI**

1952 /eSIMEasySetupResURI

1953 **A.5.3 Resource type**

1954 The Resource Type is defined as: "oic.r.esimeasysetup".

1955 **A.5.4 OpenAPI 2.0 definition**

```

1956 {
1957   "swagger": "2.0",
1958   "info": {
1959     "title": "eSIM Easy Setup Collection",
1960     "version": "2020-09-01",
1961     "license": {
1962       "name": "OCF Data Model License",
1963       "url":
1964         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
1965         CENSE.md",
1966       "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
1967     },
1968     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
1969   },
1970   "schemes": ["http"],
1971   "consumes": ["application/json"],
1972   "produces": ["application/json"],
1973   "paths": {

```

```

1974     "/eSIMEasySetupResURI?if=oic.if.ll" : {
1975         "get": {
1976             "description": "The eSIMEasySetup Resource Type stores useful information including Remote
1977 SIM Provisioning (RSP) status, and RSP last error which was produced in the process of eSIM Easy
1978 Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links to
1979 RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
1980             "parameters": [
1981                 {"$ref": "#/parameters/interface-all"}
1982             ],
1983             "responses": {
1984                 "200": {
1985                     "description": "",
1986                     "x-example":
1987                         [
1988                             {
1989                                 "href": "/eSIMEasySetupResURI",
1990                                 "rt": ["oic.r.esimeasysetup"],
1991                                 "if": ["oic.if.b", "oic.if.baseline", "oic.if.ll"],
1992                                 "p":{"bm":3},
1993                                 "eps": [
1994                                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1995                                 ],
1996                                 "rel":["self", "item"]
1997                             },
1998                             {
1999                                 "href": "/RSPConfResURI",
2000                                 "rt": ["oic.r.rspconf"],
2001                                 "if": ["oic.if.baseline", "oic.if.rw"],
2002                                 "p":{"bm":3},
2003                                 "eps": [
2004                                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
2005                                 ]
2006                             },
2007                             {
2008                                 "href": "/RSPCapabilityResURI",
2009                                 "rt": ["oic.r.rspcapability"],
2010                                 "if": ["oic.if.baseline", "oic.if.r"],
2011                                 "p":{"bm":3},
2012                                 "eps": [
2013                                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
2014                                 ]
2015                             }
2016                         ],
2017                 "schema": { "$ref": "#/definitions/slinks" }
2018             }
2019         }
2020     },
2021     "/eSIMEasySetupResURI?if=oic.if.b" : {
2022         "get": {
2023             "description": "The eSIMEasySetup Resource Type stores useful information including Remote
2024 SIM Provisioning (RSP) status, and RSP last error code which was produced in the process of eSIM
2025 Easy Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links
2026 to RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
2027             "parameters": [
2028                 {"$ref": "#/parameters/interface-all"}
2029             ],
2030             "responses": {
2031                 "200": {
2032                     "description": "",
2033                     "x-example":
2034                         [
2035                             {
2036                                 "href": "/eSIMEasySetupResURI",
2037                                 "rep":{"
2038                                     "ps" : "User confirmation pending",
2039                                     "ler": "",
2040                                     "lec": "",
2041                                     "led": "",
2042                                     "euc": "Undefined"
2043                                 }

```

```

2044     }
2045     },
2046     {
2047         "href": "/RSPConfResURI",
2048         "rep": {
2049             "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
2050             "pm":
2051 "vyU4WgqJAQIDBAUGBwgJkRNTZXJ2aWNlUHJvdmlkZXJOYW1lkgtQcm9maWxlTmFtZZMBAJQCAACVAQI=",
2052             "cc": "",
2053             "ccr": true
2054         }
2055     },
2056     {
2057         "href": "/RSPCapabilityURI",
2058         "rep": {
2059             "euccinfo":
2060 "vyJ7gQMCAACCAwICAYMDQQEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDagMAiAIEkKkWBRRmWhQzlnwaLF24tSyWfxCgV7p
2061 csqoWBRRmWhQzlnwaLF24tSyWfxCgV7pcsosBAGQDAQAADBxMDAwMDAwMDAwMDAwMDAw",
2062             "deviceinfo":
2063 "oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMBAUGhAMFBgeFAwYHCIYDBwgJhwMICQo="
2064         }
2065     }
2066 ],
2067     "schema": { "$ref": "#/definitions/sbatch" }
2068 }
2069 },
2070 },
2071 "post": {
2072     "description": "Able to deliver RSP Configuration, RSP Capability and other
2073 configuration\ninformation in a batch by utilizing 'batch' OCF Interface.\nIf you want to deliver in
2074 a batch,\nyou can write all Properties you want to send with a 'batch' OCF Interface.\nThe below
2075 example is the case to send eSIMEasySetup and RSP configuration\n(i.e., RSP Procedure Status,
2076 Activation Code, Confirmation Code required) in a batch.\n",
2077     "parameters": [
2078         { "$ref": "#/parameters/interface-update" },
2079         {
2080             "name": "body",
2081             "in": "body",
2082             "required": true,
2083             "schema": { "$ref": "#/definitions/sbatch-update" },
2084             "x-example":
2085             [
2086                 {
2087                     "href": "/eSIMEasySetupResURI",
2088                     "rep": {
2089                         "euc": "Download OK"
2090                     }
2091                 },
2092                 {
2093                     "href": "/RSPConfResURI",
2094                     "rep": {
2095                         "cc": "102030405"
2096                     }
2097                 }
2098             ]
2099         }
2100     ],
2101     "responses": {
2102         "200": {
2103             "description": "",
2104             "x-example":
2105             [
2106                 {
2107                     "href": "/eSIMEasySetupResURI",
2108                     "rep": {
2109                         "ps": "Confirmation received",
2110                         "ler": "",
2111                         "lec": "",
2112                         "led": "",
2113                         "euc": "Download OK"

```

```

2114     }
2115     },
2116     {
2117         "href": "/RSPConfResURI",
2118         "rep": {
2119             "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
2120             "pm":
2121 "vyU4WgqJAQIDBAUGBwgJkRNTZXJ2aWNlUHJvdmlkZXZJOYw1lkggtQcm9maWxlTmFtZmZMBAJQCAACVAQI=",
2122             "cc": "102030405",
2123             "ccr": true
2124         }
2125     },
2126     {
2127         "href": "/RSPCapabilityResURI",
2128         "rep": {
2129             "euicinfo":
2130 "vyJ7gQMCAACCAwICAYMDQQEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDagMAiAIEkKkWBRRmWhQzlnwaLF24tSyWfxCgV7p
2131 csqoWBRRmWhQzlnwaLF24tSyWfxCgV7pcsosBAGQDAQAADBxMDAwMDAwMDAwMDAwMDAw",
2132             "deviceinfo":
2133 "oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMDBAUGhAMFBGefAWYHCIYDBwgJhwMICQo="
2134         }
2135     }
2136 ],
2137 "schema": { "$ref": "#/definitions/sbatch" }
2138 }
2139 }
2140 }
2141 },
2142 "/eSIMEasySetupResURI?if=oic.if.baseline": {
2143     "get": {
2144         "description": "The eSIMEasySetup Resource Type stores useful information including Remote
2145 SIM Provisioning (RSP) status,\n and RSP last error code which was produced in the process of eSIM
2146 Easy Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links
2147 to RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
2148         "parameters": [
2149             {"$ref": "#/parameters/interface-all"}
2150         ],
2151         "responses": {
2152             "200": {
2153                 "description": "",
2154                 "x-example":
2155                 {
2156                     "rt": ["oic.r.esimeasyssetup"],
2157                     "if": ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
2158                     "ps": ["Undefined", "Initiated", "User confirmation pending", "Confirmation
2159 received", "Downloaded", "Installed", "Error"],
2160                     "ler": "",
2161                     "lec": "",
2162                     "led": "",
2163                     "euc": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download
2164 OK", "Download and Enable OK"],
2165                     "links": [
2166                         {
2167                             "href": "/eSIMEasySetupResURI",
2168                             "rt": ["oic.r.esimeasyssetup", "oic.wk.col"],
2169                             "if": ["oic.if.b", "oic.if.baseline", "oic.if.ll"],
2170                             "p":{"bm":3},
2171                             "eps": [
2172                                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
2173                             ],
2174                             "rel":["self", "item"]
2175                         },
2176                         {
2177                             "href": "/RSPConfResURI",
2178                             "rt": ["oic.r.rspsconf"],
2179                             "if": ["oic.if.baseline", "oic.if.rw"],
2180                             "p":{"bm":3},
2181                             "eps": [
2182                                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
2183                             ]
2184                         }
2185                     ]
2186                 }
2187             }
2188         }
2189     }
2190 }

```

```

2184         },
2185         {
2186             "href": "/RSPCapabilityResURI",
2187             "rt": ["oic.r.rspcapability"],
2188             "if": ["oic.if.baseline", "oic.if.r"],
2189             "p": {"bm": 3},
2190             "eps": [
2191                 {"ep": "coaps://[fe80:b1d6]:1111", "pri": 2}
2192             ]
2193         }
2194     ],
2195     },
2196     "schema": { "$ref": "#/definitions/eSIMEasySetup" }
2197 }
2198 }
2199 }
2200 }
2201 },
2202 "parameters": {
2203     "interface-all" : {
2204         "in" : "query",
2205         "name" : "if",
2206         "type" : "string",
2207         "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
2208     },
2209     "interface-update" : {
2210         "in" : "query",
2211         "name" : "if",
2212         "type" : "string",
2213         "enum" : ["oic.if.b"]
2214     }
2215 },
2216 "definitions": {
2217     "oic.oic-link": {
2218         "type": "object",
2219         "properties": {
2220             "anchor": {
2221                 "$ref":
2222 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2223 schema.json#/definitions/anchor"
2224             },
2225             "di": {
2226                 "$ref":
2227 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2228 schema.json#/definitions/di"
2229             },
2230             "eps": {
2231                 "$ref":
2232 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2233 schema.json#/definitions/eps"
2234             },
2235             "href": {
2236                 "$ref":
2237 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2238 schema.json#/definitions/href"
2239             },
2240             "ins": {
2241                 "$ref":
2242 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2243 schema.json#/definitions/ins"
2244             },
2245             "p": {
2246                 "$ref":
2247 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2248 schema.json#/definitions/p"
2249             },
2250             "rel": {
2251                 "$ref":
2252 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2253 schema.json#/definitions/rel_array"

```

```

2254     },
2255     "title": {
2256       "$ref":
2257       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2258       schema.json#/definitions/title"
2259     },
2260     "type": {
2261       "$ref":
2262       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2263       schema.json#/definitions/type"
2264     },
2265     "if": {
2266       "description": "The OCF Interfaces supported by the target Resource",
2267       "items": {
2268         "enum": [
2269           "oic.if.baseline",
2270           "oic.if.ll",
2271           "oic.if.b",
2272           "oic.if.r",
2273           "oic.if.rw"
2274         ],
2275         "type": "string",
2276         "maxLength": 64
2277       },
2278       "minItems": 1,
2279       "uniqueItems": true,
2280       "type": "array"
2281     },
2282     "rt": {
2283       "description": "Resource Type of the target Resource",
2284       "items": {
2285         "maxLength": 64,
2286         "type": "string"
2287       },
2288       "minItems": 1,
2289       "uniqueItems": true,
2290       "type": "array"
2291     }
2292   },
2293   "required": [
2294     "href",
2295     "rt",
2296     "if"
2297   ]
2298 },
2299 "slinks" : {
2300   "type": "array",
2301   "items": {
2302     "$ref": "#/definitions/oic.oic-link"
2303   }
2304 },
2305 "sbatch" : {
2306   "minItems" : 1,
2307   "items" : {
2308     "additionalProperties": true,
2309     "properties": {
2310       "href": {
2311         "$ref":
2312         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2313         schema.json#/definitions/href"
2314       },
2315       "rep": {
2316         "description": "The response payload from a single Resource",
2317         "type": "object",
2318         "anyOf": [
2319           {
2320             "$ref": "#/definitions/eSIMEasySetup"
2321           },
2322           {
2323             "$ref": "https://openconnectivityfoundation.github.io/core-

```

```

2324 extensions/swagger2.0/oic.r.rspconf.swagger.json#/definitions/RSPConf"
2325     },
2326     {
2327         "$ref": "https://openconnectivityfoundation.github.io/core-
2328 extensions/swagger2.0/oic.r.rspcapability.swagger.json#/definitions/RSPCapability"
2329     }
2330 ]
2331 }
2332 },
2333 "required": [
2334     "href",
2335     "rep"
2336 ],
2337 "type": "object"
2338 },
2339 "type" : "array"
2340 },
2341 "sbatch-update" : {
2342     "minItems" : 1,
2343     "items" : {
2344         "additionalProperties": true,
2345         "description": "Array of Resource representations to apply to the batch Collection, \nusing
2346 href to indicate which resource(s) in the batch to update. \nIf the href Property is empty,
2347 effectively making the URI reference to the Collection itself, \nthe representation is to be applied
2348 to all Resources in the batch\n",
2349         "properties": {
2350             "href": {
2351                 "$ref":
2352 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2353 schema.json#/definitions/href"
2354             },
2355             "rep": {
2356                 "description": "The response payload from a single Resource",
2357                 "type": "object",
2358                 "anyOf": [
2359                     {
2360                         "$ref": "#/definitions/eSIMEasySetupUpdate"
2361                     },
2362                     {
2363                         "$ref": "https://openconnectivityfoundation.github.io/core-
2364 extensions/swagger2.0/oic.r.rspconf.swagger.json#/definitions/RSPConfUpdate"
2365                     }
2366                 ]
2367             }
2368         },
2369         "required": [
2370             "href",
2371             "rep"
2372         ],
2373         "type": "object"
2374     },
2375     "type" : "array"
2376 },
2377 "eSIMEasySetup" : {
2378     "properties": {
2379         "n" : {
2380             "$ref":
2381 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2382 schema.json#/definitions/n"
2383         },
2384         "rts" : {
2385             "description": "Resource Type of the Resources within the Collection",
2386             "items": {
2387                 "maxLength": 64,
2388                 "type": "string"
2389             },
2390             "minItems": 1,
2391             "uniqueItems": true,
2392             "readOnly": true,
2393             "type": "array"

```

```

2394     },
2395     "id" : {
2396         "$ref":
2397         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2398         schema.json#/definitions/id"
2399     },
2400     "rts-m" : {
2401         "description": "Resource Type of the mandatory Resources within the Collection",
2402         "items": {
2403             "maxLength": 64,
2404             "type": "string"
2405         },
2406         "minItems": 1,
2407         "uniqueItems": true,
2408         "readOnly": true,
2409         "type": "array"
2410     },
2411     "if" : {
2412         "description": "The OCF Interfaces supported by this Resource",
2413         "items": {
2414             "enum": [
2415                 "oic.if.ll",
2416                 "oic.if.baseline",
2417                 "oic.if.b"
2418             ],
2419             "type": "string",
2420             "maxLength": 64
2421         },
2422         "minItems": 3,
2423         "uniqueItems": true,
2424         "readOnly": true,
2425         "type": "array"
2426     },
2427     "rt" : {
2428         "items": {
2429             "enum": [
2430                 "oic.r.esimeasysetup"
2431             ],
2432             "type": "string",
2433             "maxLength": 64
2434         },
2435         "minItems": 1,
2436         "type": "array",
2437         "uniqueItems": true,
2438         "readOnly": true
2439     },
2440     "ps" : {
2441         "description": "Indicates the steps in Remote SIM Provisioning.\n",
2442         "enum": ["Undefined", "Initiated", "User confirmation pending", "Confirmation received",
2443         "Downloaded", "Installed", "Error"],
2444         "readOnly": true,
2445         "type": "string"
2446     },
2447     "ler" : {
2448         "description": "Error Reason returned by the LPA while eSIM Easy Setup. \nIt indicates
2449         where it was occurred.\n(e.g., ES9+.GetBoundProfilePackage(Fail),
2450         ES10b.LoadBoundProfilePackage(Fail))\n",
2451         "readOnly": true,
2452         "type": "string"
2453     },
2454     "lec" : {
2455         "description": "Error Code returned by the LPA while eSIM Easy Setup. \nIt indicates why
2456         it was occurred.\nIt is mapped to the GSMA error status (e.g., \"8.8.1-3.8\", \"7\", \"6A 80\")\n",
2457         "readOnly": true,
2458         "type": "string"
2459     },
2460     "led" : {
2461         "description": "Optional error description \nreturned by the LPA while eSIM Easy Setup.
2462         (e.g., Invalid SM-DP+ Address)\n",
2463         "readOnly": true,

```

```

2464         "type": "string"
2465     },
2466     "euc" : {
2467         "description": "End User Consent for RSP.\n",
2468         "enum": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download OK",
2469 "Download and Enable OK"],
2470         "type": "string"
2471     },
2472     "links" : {
2473         "type": "array",
2474         "description": "A set of OCF Links.",
2475         "items": {
2476             "$ref": "#/definitions/oic.oic-link"
2477         },
2478         "readOnly":true
2479     }
2480 },
2481 "type" : "object",
2482 "required": ["ps", "ler", "lec", "euc"]
2483 },
2484 "eSIMEasySetupUpdate" : {
2485     "additionalProperties": true,
2486     "description": "Update to writeable values in eSIMEasySetupResURI",
2487     "properties": {
2488         "euc" : {
2489             "description": "End User Consent for RSP.\n",
2490             "enum": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download OK",
2491 "Download and Enable OK"],
2492             "type": "string"
2493         }
2494     },
2495     "type": "object",
2496     "required": ["euc"]
2497 }
2498 }
2499 }
2500

```

### 2501 A.5.5 Property definition

2502 Table A.8 defines the Properties that are part of the "oic.r.esimeasyssetup" Resource Type.

2503 **Table A.8 – The Property definitions of the Resource with type "rt" =**  
2504 **"oic.r.esimeasyssetup".**

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.

2505 **A.5.6 CRUDN behaviour**

2506 Table A.9 defines the CRUDN operations that are supported on the "oic.r.esimeasyssetup" Resource  
2507 Type.

2508 **Table A.9 – The CRUDN operations of the Resource with type "rt" = "oic.r.esimeasyssetup".**

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

2509 **A.6 Remote SIM Provisioning Capability**

2510 **A.6.1 Introduction**

2511 RSPCapability Resource stores information to help a service provider to provide appropriate  
2512 cellular plans to an end user.  
2513

2514 **A.6.2 Example URI**

2515 /RSPCapabilityResURI

2516 **A.6.3 Resource type**

2517 The Resource Type is defined as: "oic.r.rspcapability".

2518 **A.6.4 OpenAPI 2.0 definition**

```

2519 {
2520   "swagger": "2.0",
2521   "info": {
2522     "title": "Remote SIM Provisioning Capability",
2523     "version": "2020-09-01",
2524     "license": {
2525       "name": "OCF Data Model License",
2526       "url":
2527         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
2528         CENSE.md",
2529       "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
2530     },
2531     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
2532   },
2533   "schemes": ["http"],
2534   "consumes": ["application/json"],
2535   "produces": ["application/json"],
2536   "paths": {
2537     "/RSPCapabilityResURI" : {
2538       "get": {
2539         "description": "RSPCapability Resource stores information to help a service provider to
2540         provide appropriate cellular plans to an end user.\n",
2541         "parameters": [
2542           {"$ref": "#/parameters/interface-all"}
2543         ],
2544         "responses": {
2545           "200": {
2546             "description": "",
2547             "x-example":
2548               { "rt": ["oic.r.rspcapability"],
2549                 "if": ["oic.if.r", "oic.if.baseline"],
2550                 "euiinfo":
2551                   "vyJ7gQMCAACCAwICAYMDQqEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDagMAiAIEkKkWBBrmWhQz1nwaLF24tSyWfxcGv7p
2552                   csqoWBBrmWhQz1nwaLF24tSyWfxcGv7pcsosBAGQDAQAADBxMDAwMDAwMDAwMDAwMDAw",
2553                   "deviceinfo" :
2554                     "oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMBAUGHAMFBgeFAwYHCiYDBwgJhwMICQo="
2555                   },
2556           "schema": { "$ref": "#/definitions/RSPCapability" }

```

```

2557     }
2558   }
2559 }
2560 }
2561 },
2562 "parameters": {
2563   "interface-all" : {
2564     "in" : "query",
2565     "name" : "if",
2566     "type" : "string",
2567     "enum" : ["oic.if.r", "oic.if.baseline"]
2568   }
2569 },
2570 "definitions": {
2571   "RSPCapability" : {
2572     "properties": {
2573       "rt" : {
2574         "description": "Resource Type of the Resource",
2575         "items": {
2576           "enum": ["oic.r.rspcapability"],
2577           "type": "string",
2578           "maxLength": 64
2579         },
2580         "minItems": 1,
2581         "uniqueItems": true,
2582         "readOnly": true,
2583         "type": "array"
2584       },
2585       "euiccinfo" : {
2586         "description": "Refers to EUICCInfo2 defined in GSMA SGP.22 Annex H.This value type shall
2587 be encoded as Major Type 2.",
2588         "type": "string",
2589         "readOnly": true,
2590         "maxLength": 1024
2591       },
2592       "deviceinfo" : {
2593         "description": "Refers to DeviceInfo defined in GSMA SGP.22 Annex H.This value type shall
2594 be encoded as Major Type 2.",
2595         "type": "string",
2596         "readOnly": true,
2597         "maxLength": 128
2598       },
2599       "n" : {
2600         "$ref":
2601 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2602 schema.json#/definitions/n"
2603       },
2604       "id" : {
2605         "$ref":
2606 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2607 schema.json#/definitions/id"
2608       },
2609       "if" : {
2610         "description": "The OCF Interfaces supported by this Resource",
2611         "items": {
2612           "enum": [
2613             "oic.if.r",
2614             "oic.if.baseline"
2615           ],
2616           "type": "string",
2617           "maxLength": 64
2618         },
2619         "minItems": 2,
2620         "uniqueItems": true,
2621         "readOnly": true,
2622         "type": "array"
2623       }
2624     },
2625     "type" : "object",
2626     "required":["euiccinfo", "deviceinfo"]

```

2627 }  
 2628 }  
 2629 }  
 2630 }

2631 **A.6.5 Property definition**

2632 Table A.10 defines the Properties that are part of the "oic.r.rspcapability" Resource Type.

2633 **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
euicinfo	string	Yes	Read Only	Refers to EUICInfo2 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

2634 **A.6.6 CRUDN behaviour**

2635 Table A.11 defines the CRUDN operations that are supported on the "oic.r.rspcapability" Resource  
 2636 Type.

2637 **Table A.11 – The CRUDN operations of the Resource with type "rt" = "oic.r.rspcapability".**

Create	Read	Update	Delete	Notify
	get			observe

2638 **A.7 RSP Configuration**

2639 **A.7.1 Introduction**

2640 RSPConf Resource stores the information  
 2641 used to download and install an eSIM Profile to an eSIM capable IoT device.  
 2642 It comprises SM-DP+ server FQDN and Activation Code Token  
 2643 binding to a specific subscription as defined by GSMA SGP.22.

2644 **A.7.2 Example URI**

2645 /RSPConfResURI

2646 **A.7.3 Resource type**

2647 The Resource Type is defined as: "oic.r.rspconf".

2648 **A.7.4 OpenAPI 2.0 definition**

```
2649 {
2650   "swagger": "2.0",
2651   "info": {
2652     "title": "RSP Configuration",
2653     "version": "2020-09-01",
2654     "license": {
2655       "name": "OCF Data Model License",
2656       "url":
2657         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
2658         CENSE.md",
2659       "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
2660     },
2661     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
2662   },
2663   "schemes": ["http"],
2664   "consumes": ["application/json"],
2665   "produces": ["application/json"],
2666   "paths": {
2667     "/RSPConfResURI": {
2668       "get": {
2669         "description": "RSPConf Resource stores the information \nused to download and install an
2670         eSIM Profile to an eSIM capable IoT device.\nIt comprises SM-DP+ server FQDN and Activation Code
2671         Token\n binding to a specific subscription as defined by GSMA SGP.22.",
2672         "parameters": [
2673           { "$ref": "#/parameters/interface" }
2674         ],
2675         "responses": {
2676           "200": {
2677             "description": "",
2678             "x-example": {
2679               "rt": ["oic.r.rspconf"],
2680               "if": ["oic.if.rw", "oic.if.baseline"],
2681               "ac": "",
2682               "pm": "",
2683               "ccr": false
2684             },
2685             "schema": { "$ref": "#/definitions/RSPConf" }
2686           }
2687         }
2688       },
2689       "post": {
2690         "description": "Update Properties of the RSPConf Resource (deliver Activation Code in this
2691         example).\n",
2692         "parameters": [
2693           { "$ref": "#/parameters/interface-rw" },
2694           {
2695             "name": "body",
2696             "in": "body",
2697             "required": true,
2698             "schema": { "$ref": "#/definitions/RSPConfUpdate" },
2699             "x-example": {
2700               "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815"
2701             }
2702           }
2703         ],
2704         "responses": {
2705           "200": {
2706             "description": "",
2707             "x-example": {
2708               "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
2709               "pm": "",
2710               "ccr": false
2711             },
2712             "schema": { "$ref": "#/definitions/RSPConf" }
2713           }
2714         }
2715       }
2716     }
2717   }
2718 }
```

```

2713     }
2714   }
2715 }
2716 }
2717 },
2718 "parameters": {
2719   "interface": {
2720     "in": "query",
2721     "name": "if",
2722     "type": "string",
2723     "enum": ["oic.if.rw", "oic.if.baseline"]
2724   },
2725   "interface-rw": {
2726     "in": "query",
2727     "name": "if",
2728     "type": "string",
2729     "enum": ["oic.if.rw"]
2730   }
2731 },
2732 "definitions": {
2733   "RSPConf": {
2734     "properties": {
2735       "rt": {
2736         "description": "The Resource Type.",
2737         "items": {
2738           "enum": ["oic.r.rspconf"],
2739           "maxLength": 64,
2740           "type": "string"
2741         },
2742         "minItems": 1,
2743         "uniqueItems": true,
2744         "readOnly": true,
2745         "type": "array"
2746       },
2747       "ac": {
2748         "description": "The information needed to provision an eSIM device.",
2749         "maxLength": 256,
2750         "type": "string"
2751       },
2752       "pm": {
2753         "description": "Refers to ProfileInfo in GSMA SGP.22 Annex H.This value type shall be
2754 encoded as Major Type 2",
2755         "maxLength": 2048,
2756         "type": "string",
2757         "readOnly": true
2758       },
2759       "cc": {
2760         "description": "A code entered by an end user required by the SM-DP+ \nto confirm the
2761 download and installation of an eSIM Profile.\nThe Confirmation Code is provided from a service
2762 provider to the end user.\n",
2763         "maxLength": 64,
2764         "type": "string"
2765       },
2766       "ccr": {
2767         "description": "Indicates whether a Confirmation Code is required.\n",
2768         "maxLength": 64,
2769         "type": "boolean",
2770         "readOnly": true
2771       },
2772       "n": {
2773         "$ref":
2774 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2775 schema.json#/definitions/n"
2776       },
2777       "id": {
2778         "$ref":
2779 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2780 schema.json#/definitions/id"
2781       },
2782       "if": {

```

```

2783     "description": "The OCF Interface set supported by this Resource.",
2784     "items": {
2785         "enum": [
2786             "oic.if.rw",
2787             "oic.if.baseline"
2788         ],
2789         "type": "string"
2790     },
2791     "minItems": 2,
2792     "uniqueItems": true,
2793     "readOnly": true,
2794     "type": "array"
2795 }
2796 },
2797 "type": "object",
2798 "required": ["ac", "pm", "ccr"]
2799 },
2800 "RSPConfUpdate": {
2801     "properties": {
2802         "ac": {
2803             "description": "The information needed to provision an eSIM device.",
2804             "maxLength": 256,
2805             "type": "string"
2806         },
2807         "cc": {
2808             "description": "A code entered by an end user required by the SM-DP+ \nto confirm the
2809 download and installation of an eSIM Profile.\nThe Confirmation Code is provided from a service
2810 provider to the end user.\n",
2811             "maxLength": 64,
2812             "type": "string"
2813         }
2814     }
2815 }
2816 }
2817 }
2818

```

2819 **A.7.5 Property definition**

2820 Table A.12 defines the Properties that are part of the "oic.r.rspconf" Resource Type.

2821 **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.

2822 **A.7.6 CRUDN behaviour**

2823 Table A.13 defines the CRUDN operations that are supported on the "oic.r.rspconf" Resource Type.

2824 **Table A.13 – The CRUDN operations of the Resource with type "rt" = "oic.r.rspconf".**

Create	Read	Update	Delete	Notify
	get	post		observe

2825

2826