

OCF Wi-Fi Easy Setup Specification

VERSION 2.2.1 | December 2020



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

Copyright Open Connectivity Foundation, Inc. © 2020
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 © 2017-2020 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	Wi-Fi 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	12
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		

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 **1 Scope**

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

194 **2 Normative references**

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

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

203 ISO/IEC 30118-2 Information technology -- Open Connectivity Foundation (OCF) Specification --
204 Part 2: Security specification
205 <https://www.iso.org/standard/74239.html>
206 Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

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

211 IEEE 802.11, IEEE Standard for Information technology—Telecommunications and information
212 exchange between systems Local and metropolitan area networks—Specific requirements - Part
213 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)
214 Specifications, December 2016
215 <https://standards.ieee.org/findstds/standard/802.11-2016.html>

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

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

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

222 3 Terms, definitions, and abbreviated terms

223 3.1 Terms and definitions

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

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

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

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

230 3.1.1

231 Easy Setup

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

234 3.1.2

235 Easy Setup Enrollment

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

238 3.1.3

239 Enrollee

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

241 3.1.4

242 Enroller

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

244 3.1.5

245 Mediator

246 logical function that enables the *Enrollee* (3.1.3) to connect to the target network (i.e. *Enroller*
247 (3.1.4))

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

249 3.1.6

250 Activation Code

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

253 3.1.7

254 Local Profile Assistant (LPA)

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

257 3.1.8

258 eSIM Profile

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

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

262 3.1.9

263 Remote SIM Provisioning (RSP)

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

266 **3.1.10**
267 **Subscription**
268 commercial relationship between an end user and a service provider as defined in the GSMA RSP
269 Technical Specification

270 **3.1.11**
271 **Subscription Manager Data Preparation+ (SM-DP+)**
272 *eSIM Profile* (3.1.8) preparation server which securely downloads *eSIM Profile* (3.1.8) to the *LPA*
273 (3.1.7) of the respective eUICC in the device as defined in the GSMA RSP Technical Specification

274 **3.1.12**
275 **Easy Setup Mode**
276 mode that enables OCF setup and configuration to an IoT Device

277 **3.1.13**
278 **eSIM Easy Setup Mode**
279 mode that enables cellular network setting and configuration of *Remote SIM Provisioning* (3.1.9)

280 **3.2 Symbols and abbreviated terms**

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

288 **4 Document conventions and organization**

289 **4.1 Conventions**

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

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

297 **4.2 Notation**

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

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

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

304 Recommended (or should)(S).

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

311 Allowed (may or allowed)(O).

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

315 DEPRECATED.

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

322 Conditionally allowed (CA)

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

325 Conditionally required (CR)

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

329

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

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

332 **5 Overview**

333 **5.1 Introduction**

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

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

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

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

342 – Device Configuration: Additional Device configuration information.

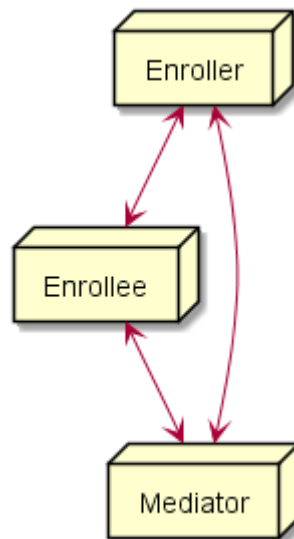
343 Easy Setup can be enhanced in future by incorporating other suitable technologies.

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

347 **5.2 Architecture**

348 Figure 1 shows the deployment architectural approach.

349



350

351

Figure 1 – Easy Setup deployment architecture

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

354 **5.3 Example Scenario**

355 The following scenario presents a typical setup case.

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

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

364 **6 Easy Setup Overview**

365 **6.1 Introduction**

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

- 367 1) EasySetup Resource Type
- 368 2) WiFiConf Resource Type
- 369 3) DevConf Resource Type

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

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

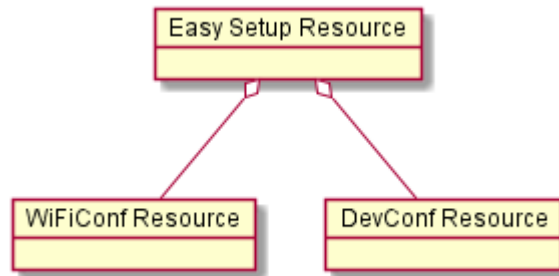
372 The EasySetup Resource Type is a Collection Resource and shall contain Links to instances of at

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

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

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

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



377

378

Figure 2 – Easy Setup Resource Types

379 **6.2 EasySetup Resource**

380 **6.2.1 Overview**

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

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

383 **6.2.2 Resource**

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

385

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

386

387 Table 2 defines the details for the "oic.r.easyssetup" Resource Type.

388

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.

389

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

- 391 – "ps" equal to 0.
- 392 – "lec" equal to 0.
- 393 – "cn" equal to an empty array.

394 6.3 WiFiConf Resource Type

395 6.3.1 Introduction

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

398 6.3.2 Resource Type

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

400

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

401

402 Table 4 defines the details for the "oic.r.wificonf" Resource Type.

403

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")

404

405 6.4 DevConf Resource Type

406 6.4.1 Introduction

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

408 6.4.2 Resource Type

409 The DevConf Resource Type is as defined in Table 5

410

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

411

412 Table 6 defines the details for the "oic.r.devconf" Resource Type.

413

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 array of object	N/A	N/A	R	Yes	Indicates a pre-configured device name in language indicated by "dl" in "/oic/con". or An array of objects where each object has a language field (containing an IETF RFC 5646 language tag) and a value field

							containing the pre-configured device name in the indicated language. The pre-configured device name is presented by enrollee to mediator during easy-setup process.
--	--	--	--	--	--	--	--

414

415 **7 eSIM Easy Setup Overview**

416 **7.1 Introduction**

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

419 If the Enrollee has no IP connectivity, Wi-Fi Easy Setup may be used to connect the Enrollee to
420 the Mediator’s Soft AP for IP connectivity. This method allows transferring subscription related
421 information between an Enrollee and a Mediator, which includes:

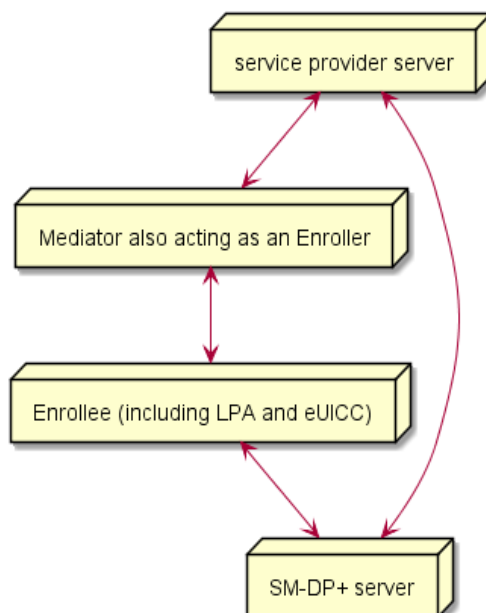
- 422 – Device and eUICC information, used to provide cellular plans to an end user
- 423 – Subscription information, comprising, e.g. Activation Code
- 424 – Progress information, indicating the status of the eSIM Easy Setup

425 **7.2 Architecture**

426 Figure 3 shows the deployment architectural approach.

427

428



429

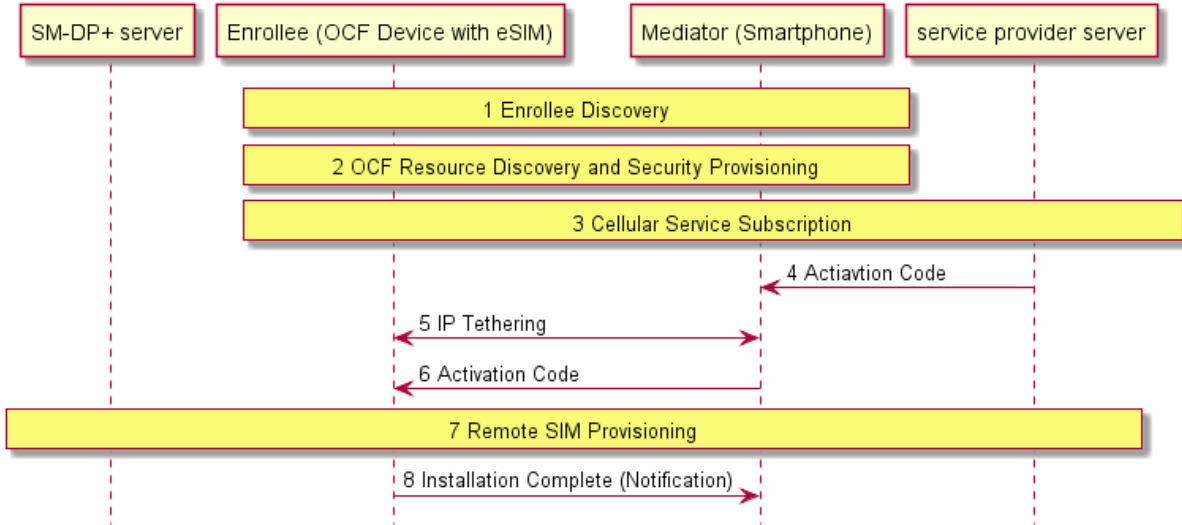
430 **Figure 3 – eSIM Easy Setup deployment architecture**

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

434 **7.3 Example Scenario**

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

437



438

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

- 440 1. When an Enrollee (e.g. OCF Device with eSIM) is unboxed for the first time, the Enrollee
 441 creates SoftAP to make it discoverable. A Mediator (e.g. smartphone) discovers and connects
 442 to the Enrollee.
- 443 2. The Mediator discovers OCF Resources of the Enrollee and performs Security Provisioning
 444 (e.g. Ownership Transfer) of the Enrollee. If eSIM Easy Setup Resources are found, the
 445 Mediator may enter eSIM Easy Setup Mode as a default and displays a certain menu (e.g.
 446 activate cellular plan) on the screen.
- 447 3. An end user enters to buy a cellular plan (e.g. when the user clicks the button) for its Enrollee.
 448 The Enrollee may deliver its Device and eUICC information to the Mediator so that the Mediator
 449 forwards that information to a service provider server. Based on the information, the service
 450 provider provides cellular plans to select.
- 451 4. Once the end user finishes the contract on his or her cellular plan, the service provider server
 452 sends an Activation Code to the Mediator.
- 453 5. The Mediator transmits its Wi-Fi Setting Information to the Enrollee. Using the Wi-Fi Setting
 454 Information received from the Mediator, the Enrollee connects to the Mediator which is acting
 455 as an Enroller (i.e. IP tethering).
- 456 6. The Mediator now transmits the Activation Code to the Enrollee
- 457 7. Via the Mediator's IP network, Enrollee sends Activation Code to a SM-DP+ server. As a return,
 458 Enrollee downloads an eSIM Profile from the SM-DP+ server, and then installs the eSIM Profile
 459 onto the eUICC in the Enrollee. While downloading the eSIM Profile, any progress information
 460 required to display to the end user is notified to the Mediator.
- 461 8. The Enrollee notifies to the Mediator once the eSIM Profile installation is completed. The
 462 Enrollee connects to the cellular network directly. The Enrollee and The Mediator disconnect
 463 its local network connection (i.e. IP tethering) if necessary.

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

466 **7.4 eSIM Easy Setup Resource Model**

467 **7.4.1 Introduction**

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

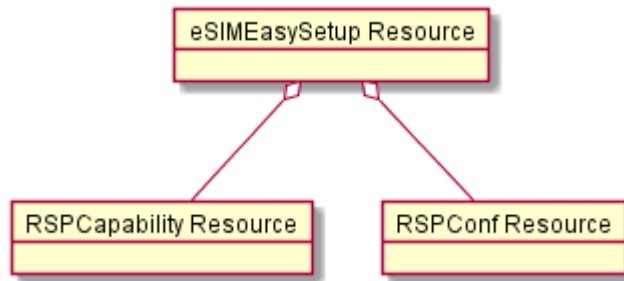
- 469 1) eSIMEasySetup Resource Type
- 470 2) RSPCapability Resource Type
- 471 3) RSPConf Resource Type

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

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

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

479



480

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

482 **7.5 eSIMEasySetup Resource Type**

483 **7.5.1 Introduction**

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

486 **7.5.2 Resource Type Definition**

487 The eSIMEasySetup Resource Type is as defined in Table 7.

488

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

489 Table 8 defines the details for the "oic.r.esimeasyssetup" Resource Type.

490

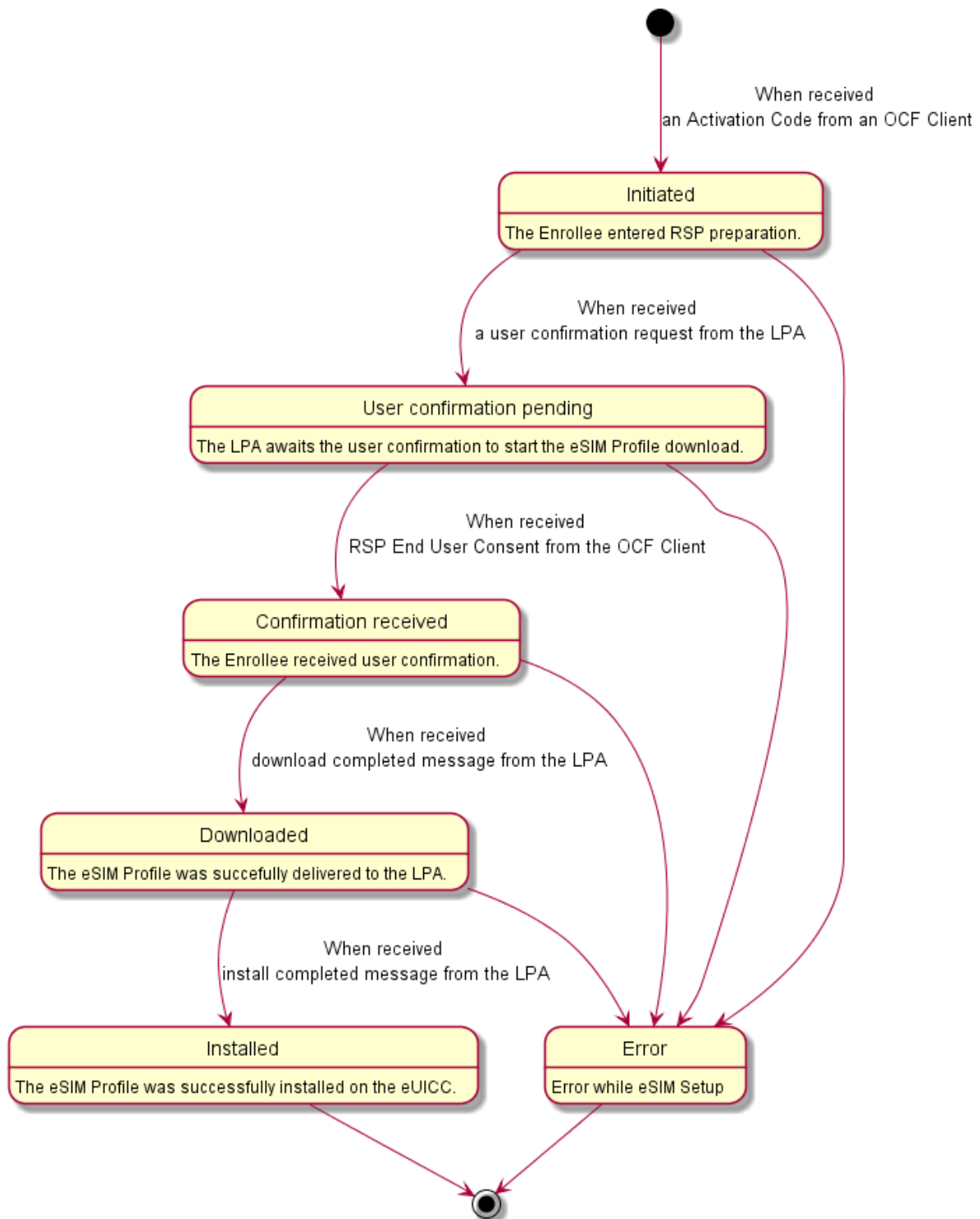
Table 8 – "oic.r.esimeasyssetup" Resource Type Definition

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

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

- 493 – "ps" equal to "Undefined".
- 494 – "ler" equal to an empty string.
- 495 – "lec" equal to an empty string.
- 496 – "led" equal to an empty string if "led" is presented.
- 497 – "euc" equal to "Undefined".

498 Figure 6 shows the RSP Procedure Status transition.



499

500

Figure 6 – RSP Procedure Status Transition

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

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

505 **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

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

508 **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)

509 **7.6 RSPCapability Resource Type**

510 **7.6.1 Introduction**

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

513 **7.6.2 Resource Type Definition**

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

515 **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.	
--	--	--	--	---	--

516 Table 12 defines the details for the "oic.r.rspcapability" Resource Type.

517 **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.

518 **7.7 RSPConf Resource Type**

519 **7.7.1 Introduction**

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

522 **7.7.2 Resource Type Definition**

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

524 **Table 13 – RSPConf Resource Type**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/RSPConfResURI	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

525 Table 14 defines the details for the "oic.r.rspconf" Resource Type.

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

529 **8 Network and connectivity**

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

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

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

536 **9 Functional interactions**

537 **9.1 Onboarding, Provisioning and Configuration**

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

540 **9.2 Resource discovery**

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

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

545 **9.3 Retrieving and Updating Easy Setup Resources**

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

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

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

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

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

559 **9.4 Error Handling**

560 **9.4.1 Wi-Fi Easy Setup Error Handling**

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

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

- 568 – The Enrollee should set "lec" Property to 7, if the Encryption type is not supported by the
569 Enrollee.
- 570 – The Enrollee should set "lec" Property to 8, if it fails to connect due to wrong Auth type
571 information (even though it's supported by the Enrollee).
- 572 – The Enrollee should set "lec" Property to 9, if it fails to connect due to wrong Encryption type
573 information (even though it's supported by the Enrollee).
- 574 When using Wi-Fi as the connectivity between the Enrollee and Mediator, if the Enrollee fails to
575 connect to the Enroller, it shall again make itself discoverable as a Soft AP (in case it destroyed
576 its Soft AP earlier).

577 **9.4.2 eSIM Easy Setup Error Handling**

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

- 580 – The Enrollee shall set "ps" Property to "Error" if it fails to download and install an eSIM Profile.
- 581 – "ler" and "lec" Properties shall be used to indicate the detailed failure reason and error code
582 within eSIM Profile download and installation.
- 583 – "led" Property may be used to indicate additional error description.
- 584 – "euc". Property shall be used to indicate an end user consent. If an end user rejects RSP
585 procedure, Enrollee shall set "ps" Value to "Error", "euc" Value to "Download Reject", and then
586 shall terminate the eSIM Easy Setup Procedure.

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

589 **9.5 Example Easy Setup Flow**

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

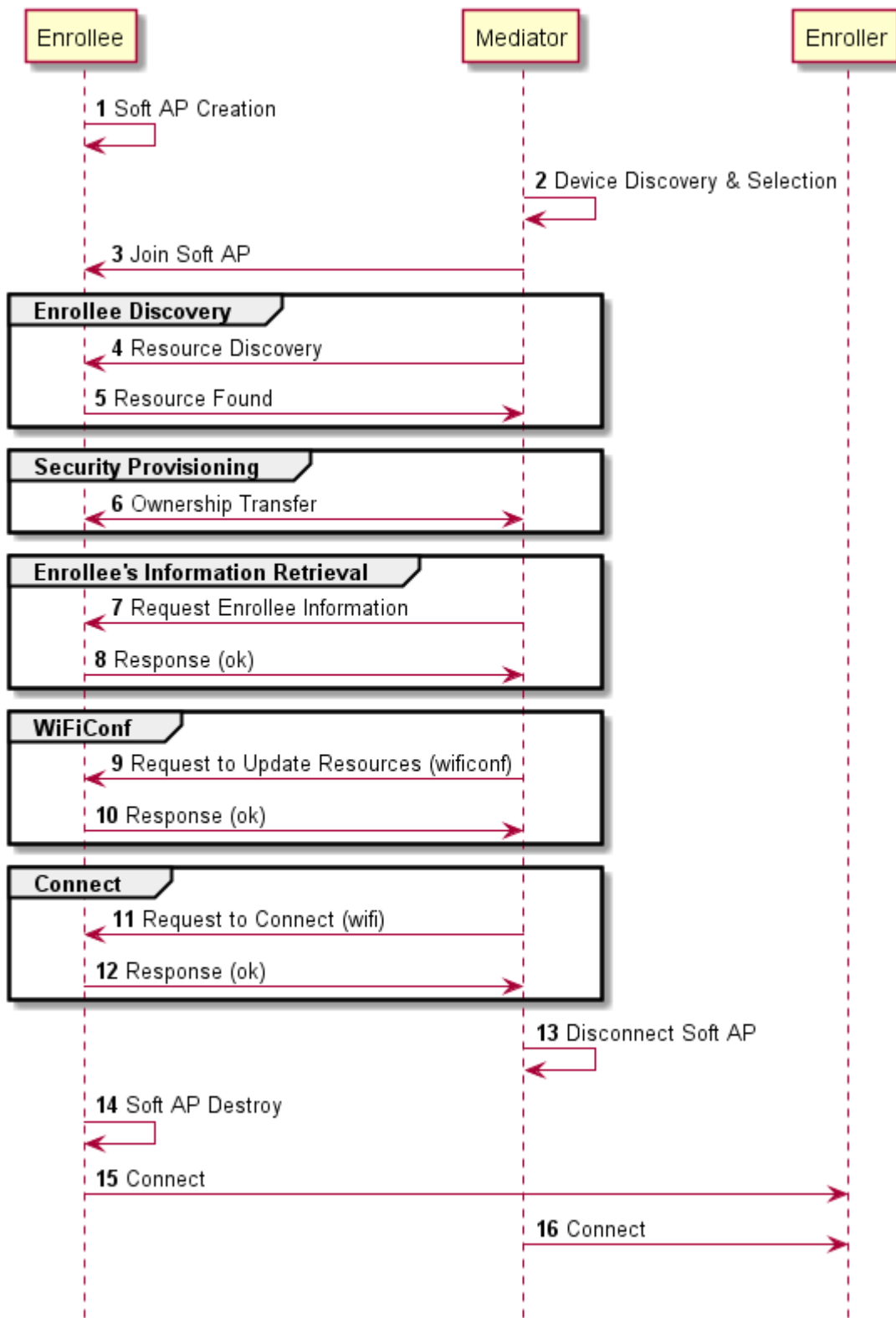
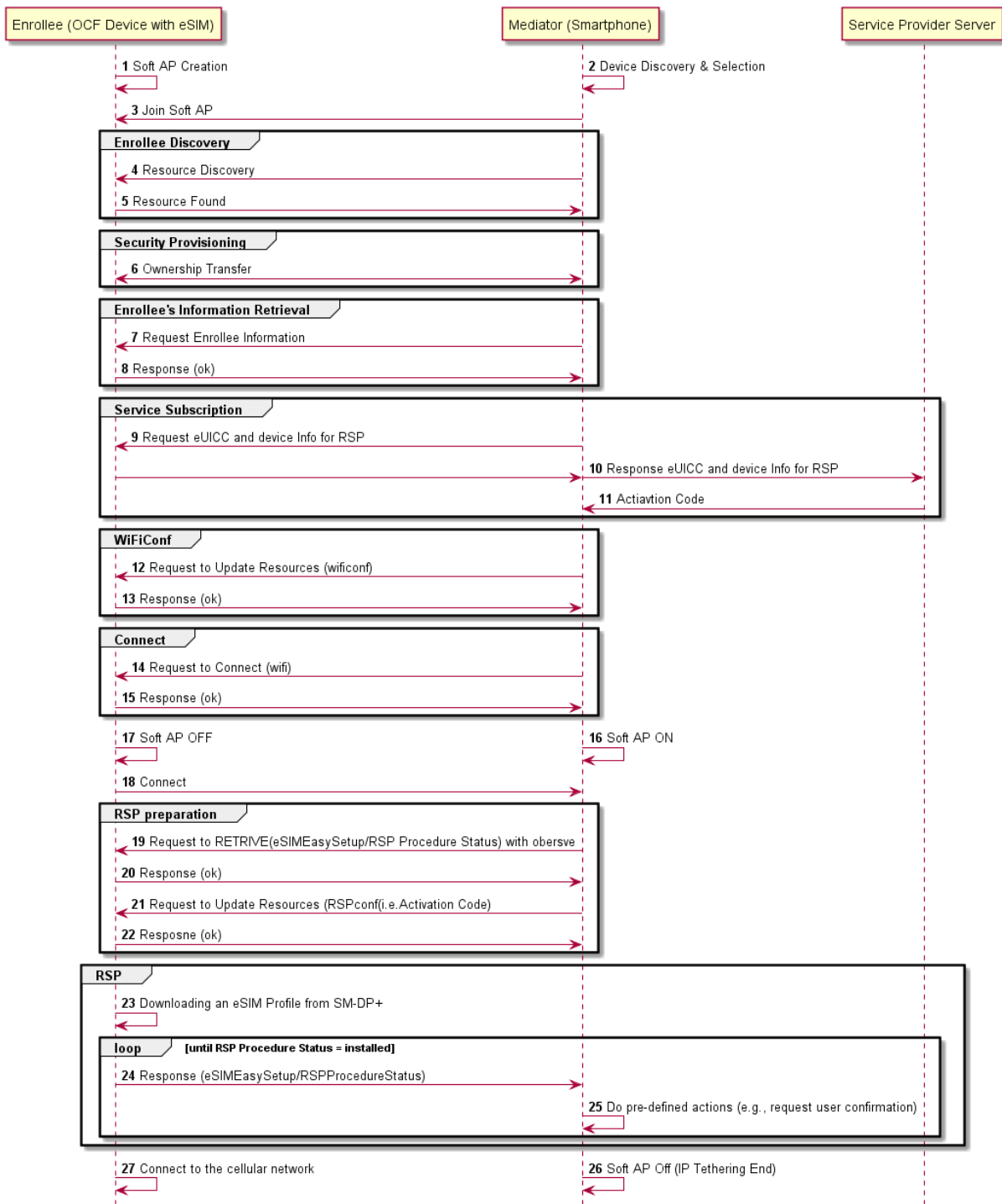


Figure 7 – Easy Setup Flow (Informative)

591
592
593

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

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



599

600

Figure 8 – eSIM Easy Setup Flow (Informative)

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

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

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

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

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

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

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

674 9.6 Easy Setup SSID Tags

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

- 677 – "OCF_"
678 – Prefix tag that has to be at the beginning of the SSID.
679 – Example: OCF_MySSID
- 680 – "_OCF"
681 – Suffix tag that has to be at the end of the SSID.
682 – Example: MySSID_OCf

683 These tags are case sensitive.

684 9.7 Easy Setup Information Element

685 9.7.1 Overview

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

691 9.7.2 OCF Device Information Element (IE)

692 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

- 693
- 694 – Type is a unique id allocated by the IEEE registrar to identify different information elements
 - 695 from each other. The Easy Setup Information Element shall have a Type value of 221 which is
 - 696 standard vendor specific information element.
 - 697 – Length shall indicate the total size of CID, OCF IE Type, and Data in bytes.
 - 698 – Company ID (CID) is a unique 24-bit identifier for a specific company or organization. The Easy
 - 699 Setup Information Element shall have a CID value of 6A 40 65.
 - 700 – OCF IE Type is the identifier of the specific IE within OCF. The OCF IE Type shall be set to 0
 - 701 for Easy Setup.
 - 702 – Data is a set of type-length-value (TLV) structures that represent the device information in Table
 - 703 1. The length of this field shall be less than 252 bytes.
 - 704
 - 705 Each TLV has the structure shown in Figure 10.

1 byte	1 byte	<250 bytes
Type	Length	Value

Figure 10 – Type-Length-Value Structure

- 706
- 707 – Type shall indicate the type of the field from Table 15.
 - 708 – Length shall indicate the length of the Value in bytes.
 - 709 – Value shall represent the corresponding information for specific TLV type from Table 15.
 - 710 Data is a set of TLVs as defined in Table 15.

Table 15 – Easy Setup Information Element TLVs

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

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

717 **9.7.2.1 Device Friendly Name (Type 1)**

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

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

721 **9.7.2.2 Device Type (Type 2)**

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

- 724
 - 725 – Device Type as specified in the ISO/IEC 30118-5: "oic.d.airconditioner"
 - 726 – Device Type as specified in a type 2 TLV: "airconditioner"
- 727 In cases where the device supports multiple functions, several type 2 TLVs may be included to
728 represent each function of the device.

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

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

734 **9.7.2.3 Device Manufacturer Name (Type 3)**

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

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

739 **9.7.2.4 Language Tag (Type 4)**

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

744 Please see 9.7.2.8 for information on supporting multiple languages.

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

747 **9.7.2.5 Protocol Independent ID (Type 5)**

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

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

750 **9.7.2.6 Multiple Information Elements**

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

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

754 – Support for multiple languages is necessary.

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

757 **9.7.2.7 IEC for Large TLV Size Support**

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

761 – Type

762 – Length

763 – CID

764 – OCF IE Type

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

767 **9.7.2.8 IEC for Multiple Language Support**

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

772 **10 Security**

773 A Device shall meet the Wi-Fi 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" : {
```

```

811     "get": {
812         "description": "The Device configuration Resource stores Device settings such as the Device
813 name. Vendor-specific information can be added to the Resource.\nThe Device name is a human-friendly
814 name read by a Mediator during easy setup.\n",
815         "parameters": [
816             { "$ref": "#/parameters/interface" }
817         ],
818         "responses": {
819             "200": {
820                 "description": "",
821                 "x-example": {
822                     "rt": ["oic.r.devconf"],
823                     "dn": "My Refrigerator"
824                 },
825                 "schema": { "$ref": "#/definitions/DevConf" }
826             }
827         }
828     }
829 },
830 },
831 "parameters": {
832     "interface": {
833         "in": "query",
834         "name": "if",
835         "type": "string",
836         "enum": ["oic.if.r", "oic.if.baseline"]
837     }
838 },
839 "definitions": {
840     "DevConf": {
841         "properties": {
842             "rt": {
843                 "description": "Resource Type of the Resource",
844                 "items": {
845                     "enum": ["oic.r.devconf"],
846                     "maxLength": 64,
847                     "type": "string"
848                 },
849                 "minItems": 1,
850                 "readOnly": true,
851                 "uniqueItems": true,
852                 "type": "array"
853             },
854             "n": {
855                 "$ref":
856 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
857 schema.json#/definitions/n"
858             },
859             "id": {
860                 "$ref":
861 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
862 schema.json#/definitions/id"
863             },
864             "if": {
865                 "description": "The OCF Interfaces supported by this Resource",
866                 "items": {
867                     "enum": [
868                         "oic.if.r",
869                         "oic.if.baseline"
870                     ],
871                     "type": "string",
872                     "maxLength": 64
873                 },
874                 "minItems": 2,
875                 "readOnly": true,
876                 "uniqueItems": true,
877                 "type": "array"
878             },
879             "dn": {
880                 "oneOf": [

```

```

881     {
882         "type": "string",
883         "description": "Indicates a pre-configured Device name in language indicated by 'dl'
884 in /oic/con: presented by an Enrollee Device to a Mediator Device during the easy-setup process",
885         "pattern": "^.*$",
886         "readOnly": true
887     },
888     {
889         "type": "array",
890         "items": {
891             "type": "object",
892             "properties": {
893                 "language": {
894 schema.json#/definitions/language-tag",
895                 "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
896                 "readOnly": true,
897                 "description": "An RFC 5646 language tag."
898             },
899             "value": {
900                 "type": "string",
901                 "description": "Pre-configured Device name in the indicated language.",
902                 "pattern": "^.*$",
903                 "readOnly": true
904             }
905         }
906     },
907     "minItems" : 1,
908     "readOnly": true,
909     "description": "Localized device name."
910 }
911 ]
912 }
913 },
914 "type" : "object",
915 "required": ["dn"]
916 }
917 }
918 }
919

```

920 **A.2.5 Property definition**

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

922 **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	

923 **A.2.6 CRUDN behaviour**

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

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

Create	Read	Update	Delete	Notify
	get			observe

926 **A.3 Easy Setup Collection**

927 **A.3.1 Introduction**

928 The Easy Setup Resource stores useful information including the current status of unboxing a
 929 Device and the last error code which are produced in the process of easy setup.
 930 Note that the Easy Setup Resource is a Collection Resource, which contains Links to WiFiConf,
 931 and DevConf Resources and may additionally contain Links to other Resources.
 932

933 **A.3.2 Example URI**

934 /EasySetupResURI

935 **A.3.3 Resource type**

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

937 **A.3.4 OpenAPI 2.0 definition**

```

938 {
939   "swagger": "2.0",
940   "info": {
941     "title": "Easy Setup Collection",
942     "version": "2019-03-27",
943     "license": {
944       "name": "OCF Data Model License",
945       "url":
946         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
947         CENSE.md",
948       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
949     },
950     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
951   },
952   "schemes": ["http"],
953   "consumes": ["application/json"],
954   "produces": ["application/json"],
955   "paths": {
956     "/EasySetupResURI?if=oic.if.ll" : {
957       "get": {
958         "description": "The Easy Setup Resource stores useful information including the current
959         status of unboxing a Device and the last error code which are produced in the process of easy
960         setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
961         WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
962         "parameters": [
963           {"$ref": "#/parameters/interface-all"}
964         ],
965         "responses": {
966           "200": {
967             "description": "",
968             "x-example":
969               [
970                 {
971                   "href": "/EasySetupResURI",
972                   "rt": ["oic.r.easyssetup", "oic.wk.col"],
973                   "if": ["oic.if.b"],
974                   "p":{"bm":3},
975                   "eps": [
976                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
977                   ],
978                   "rel":["self", "item"]
979                 }
980               ]
981           }
982         }
983       }
984     }
985   }
986 }
    
```

```

979         },
980         {
981             "href": "/WiFiConfResURI",
982             "rt": ["oic.r.wificonf"],
983             "if": ["oic.if.baseline"],
984             "p":{"bm":3},
985             "eps": [
986                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
987             ]
988         },
989         {
990             "href": "/DevConfResURI",
991             "rt": ["oic.r.devconf"],
992             "if": ["oic.if.baseline"],
993             "p":{"bm":3},
994             "eps": [
995                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
996             ]
997         }
998     ],
999     "schema": { "$ref": "#/definitions/slinks" }
1000 }
1001 }
1002 }
1003 },
1004 "/EasySetupResURI?if=oic.if.b" : {
1005     "get": {
1006         "description": "The Easy Setup Resource stores useful information including the current
1007 status of unboxing a Device and the last error code which are produced in the process of easy
1008 setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
1009 WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
1010         "parameters": [
1011             {"$ref": "#/parameters/interface-all"}
1012         ],
1013         "responses": {
1014             "200": {
1015                 "description" : "",
1016                 "x-example":
1017                 [
1018                     {
1019                         "href": "/EasySetupResURI",
1020                         "rep":{"
1021                             "ps" : 0,
1022                             "lec": 0,
1023                             "cn": [1]
1024                         }
1025                     },
1026                     {
1027                         "href": "/WiFiConfResURI",
1028                         "rep":{"
1029                             "swmt" : ["A", "B", "G"],
1030                             "swf": ["2.4G", "5G"],
1031                             "tnn": "Home_AP_SSID",
1032                             "cd": "Home_AP_PWD",
1033                             "wat": "WPA2_PSK",
1034                             "wet": "AES",
1035                             "swat": ["WPA_PSK", "WPA2_PSK"],
1036                             "swet": ["TKIP", "AES", "TKIP_AES"]
1037                         }
1038                     },
1039                     {
1040                         "href": "/DevConfResURI",
1041                         "rep":{"
1042                             "dn" : "My Refrigerator"
1043                         }
1044                     }
1045                 ],
1046                 "schema": { "$ref": "#/definitions/sbatch" }
1047             }
1048         }

```



```

1049     },
1050     "post": {
1051         "description": "Able to deliver Wi-Fi, Device configuration and other
1052 configuration\ninformation in a batch by utilizing 'batch' OCF Interface.\nIf you want to deliver
1053 Wi-Fi and Device configuration information in a batch,\nyou can write all Properties you want to
1054 send with a 'batch' OCF Interface.\nThe below example is the case to send Easy Setup and Wi-Fi
1055 configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",
1056         "parameters": [
1057             { "$ref": "#/parameters/interface-update" },
1058             {
1059                 "name": "body",
1060                 "in": "body",
1061                 "required": true,
1062                 "schema": { "$ref": "#/definitions/sbatch-update" },
1063                 "x-example":
1064                     [
1065                         {
1066                             "href": "/EasySetupResURI",
1067                             "rep": {
1068                                 "cn": [1]
1069                             }
1070                         },
1071                         {
1072                             "href": "/WiFiConfResURI",
1073                             "rep": {
1074                                 "tnn": "Home_AP_SSID",
1075                                 "cd": "Home_AP_PWD",
1076                                 "wat": "WPA2_PSK",
1077                                 "wet": "AES"
1078                             }
1079                         }
1080                     ]
1081             }
1082         ],
1083         "responses": {
1084             "200": {
1085                 "description": "",
1086                 "x-example":
1087                     [
1088                         {
1089                             "href": "/EasySetupResURI",
1090                             "rep": {
1091                                 "ps": 0,
1092                                 "lec": 0,
1093                                 "cn": [1]
1094                             }
1095                         },
1096                         {
1097                             "href": "/WiFiConfResURI",
1098                             "rep": {
1099                                 "swmt": ["A", "B", "G"],
1100                                 "swf": ["2.4G", "5G"],
1101                                 "tnn": "Home_AP_SSID",
1102                                 "cd": "Home_AP_PWD",
1103                                 "wat": "WPA2_PSK",
1104                                 "wet": "AES",
1105                                 "swat": ["WPA_PSK", "WPA2_PSK"],
1106                                 "swet": ["TKIP", "AES", "TKIP_AES"]
1107                             }
1108                         },
1109                         {
1110                             "href": "/DevConfResURI",
1111                             "rep": {
1112                                 "dn": "My Refrigerator"
1113                             }
1114                         }
1115                     ],
1116                 "schema": { "$ref": "#/definitions/sbatch" }
1117             }
1118         }

```

```

1119     }
1120   },
1121   "/EasySetupResURI?if=oic.if.baseline" : {
1122     "get": {
1123       "description": "The Easy Setup Resource stores useful information including the current
1124 status of unboxing a Device and the last error code which are produced in the process of easy
1125 setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
1126 WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
1127       "parameters": [
1128         { "$ref": "#/parameters/interface-all" }
1129       ],
1130       "responses": {
1131         "200": {
1132           "description" : "",
1133           "x-example":
1134             {
1135               "rt" : ["oic.r.easyssetup", "oic.wk.col"],
1136               "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
1137               "ps" : 0,
1138               "lec": 0,
1139               "cn": [1],
1140               "links": [
1141                 {
1142                   "href": "/EasySetupResURI",
1143                   "rt": ["oic.r.easyssetup", "oic.wk.col"],
1144                   "if": ["oic.if.b"],
1145                   "p":{"bm":3},
1146                   "eps": [
1147                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1148                   ],
1149                   "rel":["self", "item"]
1150                 },
1151                 {
1152                   "href": "/WiFiConfResURI",
1153                   "rt": ["oic.r.wificonf"],
1154                   "if": ["oic.if.baseline"],
1155                   "p":{"bm":3},
1156                   "eps": [
1157                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1158                   ]
1159                 },
1160                 {
1161                   "href": "/DevConfResURI",
1162                   "rt": ["oic.r.devconf"],
1163                   "if": ["oic.if.baseline"],
1164                   "p":{"bm":3},
1165                   "eps": [
1166                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1167                   ]
1168                 }
1169               ]
1170             },
1171         "schema": { "$ref": "#/definitions/EasySetup" }
1172       }
1173     }
1174   },
1175   "post": {
1176     "description": "Able to update connection type to attempt to connect to the Enroller to
1177 start during while posting to /EasySetupResURI\nThe below example is the case to send Easy Setup
1178 configuration\n(i.e. connection type) in a post.\n",
1179     "parameters": [
1180       { "$ref": "#/parameters/interface-update" },
1181       {
1182         "name": "body",
1183         "in": "body",
1184         "required": true,
1185         "schema": { "$ref": "#/definitions/EasySetupUpdate" },
1186         "x-example":
1187           {
1188             "cn": [1]

```

```

1189     }
1190   }
1191 ],
1192 "responses": {
1193   "200": {
1194     "description" : "",
1195     "x-example":
1196     {
1197       "rt" : ["oic.r.easyssetup", "oic.wk.col"],
1198       "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
1199       "ps" : 0,
1200       "lec" : 0,
1201       "cn" : [1],
1202       "links": [
1203         {
1204           "href": "/EasySetupResURI",
1205           "rt": ["oic.r.easyssetup", "oic.wk.col"],
1206           "if": ["oic.if.b", "oic.if.ll", "oic.if.baseline"],
1207           "p":{"bm":3},
1208           "eps": [
1209             {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1210           ],
1211           "rel":["self", "item"]
1212         },
1213         {
1214           "href": "/WiFiConfResURI",
1215           "rt": ["oic.r.wificonf"],
1216           "if": ["oic.if.rw", "oic.if.baseline"],
1217           "p":{"bm":3},
1218           "eps": [
1219             {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1220           ]
1221         },
1222         {
1223           "href": "/DevConfResURI",
1224           "rt": ["oic.r.devconf"],
1225           "if": ["oic.if.r", "oic.if.baseline"],
1226           "p":{"bm":3},
1227           "eps": [
1228             {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1229           ]
1230         }
1231       ]
1232     },
1233     "schema": { "$ref": "#/definitions/EasySetup" }
1234   }
1235 }
1236 }
1237 }
1238 },
1239 "parameters": {
1240   "interface-all" : {
1241     "in" : "query",
1242     "name" : "if",
1243     "type" : "string",
1244     "enum" : ["oic.if.ll","oic.if.b","oic.if.baseline"]
1245   },
1246   "interface-update" : {
1247     "in" : "query",
1248     "name" : "if",
1249     "type" : "string",
1250     "enum" : ["oic.if.b", "oic.if.baseline"]
1251   }
1252 },
1253 "definitions": {
1254   "oic.oic-link": {
1255     "type": "object",
1256     "properties": {
1257       "anchor": {
1258         "$ref":

```

```

1259 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1260 schema.json#/definitions/anchor"
1261 },
1262 "di": {
1263   "$ref":
1264 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1265 schema.json#/definitions/di"
1266 },
1267 "eps": {
1268   "$ref":
1269 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1270 schema.json#/definitions/eps"
1271 },
1272 "href": {
1273   "$ref":
1274 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1275 schema.json#/definitions/href"
1276 },
1277 "ins": {
1278   "$ref":
1279 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1280 schema.json#/definitions/ins"
1281 },
1282 "p": {
1283   "$ref":
1284 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1285 schema.json#/definitions/p"
1286 },
1287 "rel": {
1288   "$ref":
1289 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1290 schema.json#/definitions/rel_array"
1291 },
1292 "title": {
1293   "$ref":
1294 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1295 schema.json#/definitions/title"
1296 },
1297 "type": {
1298   "$ref":
1299 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1300 schema.json#/definitions/type"
1301 },
1302 "if": {
1303   "description": "The OCF Interfaces supported by the target Resource",
1304   "items": {
1305     "enum": [
1306       "oic.if.baseline",
1307       "oic.if.ll",
1308       "oic.if.b",
1309       "oic.if.r",
1310       "oic.if.rw"
1311     ],
1312     "type": "string",
1313     "maxLength": 64
1314   },
1315   "minItems": 1,
1316   "uniqueItems": true,
1317   "type": "array"
1318 },
1319 "rt": {
1320   "description": "Resource Type of the target Resource",
1321   "items": {
1322     "maxLength": 64,
1323     "type": "string"
1324   },
1325   "minItems": 1,
1326   "uniqueItems": true,
1327   "type": "array"
1328 }

```

```

1329     },
1330     "required": [
1331         "href",
1332         "rt",
1333         "if"
1334     ]
1335 },
1336 "slinks" : {
1337     "type": "array",
1338     "items": {
1339         "$ref": "#/definitions/oic.oic-link"
1340     }
1341 },
1342 "sbatch" : {
1343     "minItems" : 1,
1344     "items" : {
1345         "additionalProperties": true,
1346         "properties": {
1347             "href": {
1348                 "$ref":
1349 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1350 schema.json#/definitions/href"
1351             },
1352             "rep": {
1353                 "description": "The response payload from a single Resource",
1354                 "type": "object",
1355                 "anyOf": [
1356                     {
1357                         "$ref": "#/definitions/EasySetup"
1358                     },
1359                     {
1360                         "$ref": "https://openconnectivityfoundation.github.io/core-
1361 extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConf"
1362                     },
1363                     {
1364                         "$ref": "https://openconnectivityfoundation.github.io/core-
1365 extensions/swagger2.0/oic.r.devconf.swagger.json#/definitions/DevConf"
1366                     }
1367                 ]
1368             }
1369         },
1370         "required": [
1371             "href",
1372             "rep"
1373         ],
1374         "type": "object"
1375     },
1376     "type" : "array"
1377 },
1378 "sbatch-update" : {
1379     "minItems" : 1,
1380     "items" : {
1381         "additionalProperties": true,
1382         "description": "Array of Resource representations to apply to the batch Collection, using
1383 href to indicate which resource(s) in the batch to update. If the href Property is empty,
1384 effectively making the URI reference to the Collection itself, the representation is to be applied
1385 to all Resources in the batch",
1386         "properties": {
1387             "href": {
1388                 "$ref":
1389 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1390 schema.json#/definitions/href"
1391             },
1392             "rep": {
1393                 "description": "The response payload from a single Resource",
1394                 "type": "object",
1395                 "anyOf": [
1396                     {
1397                         "$ref": "#/definitions/EasySetupUpdate"
1398                     }

```

```

1399         {
1400             "$ref": "https://openconnectivityfoundation.github.io/core-
1401 extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConfUpdate"
1402         }
1403     ]
1404 }
1405 },
1406 "required": [
1407     "href",
1408     "rep"
1409 ],
1410 "type": "object"
1411 },
1412 "type" : "array"
1413 },
1414 "EasySetup" : {
1415     "properties": {
1416         "n" : {
1417             "$ref":
1418 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1419 schema.json#/definitions/n"
1420         },
1421         "rts" : {
1422             "description": "Resource Type of the Resources within the Collection",
1423             "items": {
1424                 "maxLength": 64,
1425                 "type": "string"
1426             },
1427             "minItems": 1,
1428             "uniqueItems": true,
1429             "readOnly": true,
1430             "type": "array"
1431         },
1432         "id" : {
1433             "$ref":
1434 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1435 schema.json#/definitions/id"
1436         },
1437         "rts-m" : {
1438             "description": "Resource Type of the mandatory Resources within the Collection",
1439             "items": {
1440                 "maxLength": 64,
1441                 "type": "string"
1442             },
1443             "minItems": 1,
1444             "uniqueItems": true,
1445             "readOnly": true,
1446             "type": "array"
1447         },
1448         "if" : {
1449             "description": "The OCF Interfaces supported by this Resource",
1450             "items": {
1451                 "enum": [
1452                     "oic.if.ll",
1453                     "oic.if.baseline",
1454                     "oic.if.b"
1455                 ],
1456                 "type": "string",
1457                 "maxLength": 64
1458             },
1459             "minItems": 2,
1460             "uniqueItems": true,
1461             "readOnly": true,
1462             "type": "array"
1463         },
1464         "rt" : {
1465             "items": {
1466                 "enum": [
1467                     "oic.r.easysetup",
1468                     "oic.wk.col"

```

```

1469         ],
1470         "type": "string",
1471         "maxLength": 64
1472     },
1473     "minItems": 2,
1474     "type": "array",
1475     "uniqueItems": true
1476 },
1477 "ps" : {
1478     "description": "Indicates the easy setup status of the Device. (0: Need to Setup, 1:
1479 Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved,
1480 255: EOF)",
1481     "enum": [
1482         0,
1483         1,
1484         2,
1485         3
1486     ],
1487     "readOnly": true,
1488     "type": "integer"
1489 },
1490 "lec" : {
1491     "description": "Indicates a failure reason (0: No error, 1: A given SSID is not found, 2:
1492 Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6:
1493 Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the
1494 Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi
1495 Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown
1496 error)",
1497     "enum": [
1498         0,
1499         1,
1500         2,
1501         3,
1502         4,
1503         5,
1504         6,
1505         7,
1506         8,
1507         9,
1508         255
1509     ],
1510     "readOnly": true,
1511     "type": "integer"
1512 },
1513 "cn" : {
1514     "description": "Indicates an array of connection types that trigger an attempt to connect
1515 to the Enroller to start.",
1516     "items": {
1517         "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1518 to be added in future (e.g. Connect to cloud / BLE))",
1519         "type": "integer"
1520     },
1521     "type": "array"
1522 },
1523 "links" : {
1524     "type": "array",
1525     "description": "A set of OCF Links.",
1526     "items": {
1527         "$ref": "#/definitions/oic.oic-link"
1528     }
1529 }
1530 },
1531 "type" : "object",
1532 "required": ["ps", "lec", "cn"]
1533 },
1534 "EasySetupUpdate" : {
1535     "additionalProperties": true,
1536     "description": "Update to writeable values in EasySetupResURI",
1537     "properties": {
1538         "cn" : {

```

```

1539         "description": "Indicates an array of connection types that trigger an attempt to connect
1540 to the Enroller to start.",
1541         "items": {
1542             "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1543 to be added in future (e.g. Connect to cloud / BLE))",
1544             "type": "integer"
1545         },
1546         "type": "array"
1547     },
1548 },
1549     "required": [
1550         "cn"
1551     ],
1552     "type": "object"
1553 }
1554 }
1555 }
1556

```

1557 A.3.5 Property definition

1558 Table A.4 defines the Properties that are part of the "oic.r.easyssetup, oic.wk.col" Resource Type.

1559 **Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.easyssetup,**
1560 **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.
----	-------------------	-----	------------	---

1561 **A.3.6 CRUDN behaviour**

1562 Table A.5 defines the CRUDN operations that are supported on the "oic.r.easyssetup, oic.wk.col"
1563 Resource Type.

1564 **Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.easyssetup,**
1565 **oic.wk.col".**

Create	Read	Update	Delete	Notify
	get	post		observe

1566 **A.4 Wi-Fi Configuration**

1567 **A.4.1 Introduction**

1568 WiFiConf Resource stores essential information to help an unboxing Device
1569 to connect to an existing Wi-Fi AP.

1570

1571 **A.4.2 Example URI**

1572 /WiFiConfResURI

1573 **A.4.3 Resource type**

1574 The Resource Type is defined as: "oic.r.wificonf".

1575 **A.4.4 OpenAPI 2.0 definition**

```

1576 {
1577   "swagger": "2.0",
1578   "info": {
1579     "title": "Wi-Fi Configuration",
1580     "version": "2019-03-27",
1581     "license": {
1582       "name": "OCF Data Model License",
1583       "url":
1584         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
1585         CENSE.md",
1586       "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
1587     },
1588     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
1589   },
1590   "schemes": ["http"],
1591   "consumes": ["application/json"],
1592   "produces": ["application/json"],
1593   "paths": {
1594     "/WiFiConfResURI?if=oic.if.rw" : {
1595       "get": {
1596         "description": "The WiFiConf Resource stores essential information to help an unboxing
1597         Device connect to an existing Wi-Fi AP.\n",
1598         "parameters": [
1599           {"$ref": "#/parameters/interface-all"}
1600         ],
1601         "responses": {
1602           "200": {
1603             "description" : "",
1604             "x-example":

```

```

1605         {
1606             "tnn": "Home_AP_SSID",
1607             "swmt" : ["A", "B", "G"],
1608             "swf": ["2.4G", "5G"],
1609             "cd": "Home_AP_PWD",
1610             "wat": "WPA2_PSK",
1611             "wet": "AES",
1612             "swat": ["WPA_PSK", "WPA2_PSK"],
1613             "swet": ["TKIP", "AES", "TKIP_AES"]
1614         },
1615         "schema": { "$ref": "#/definitions/WiFiConf" }
1616     }
1617 },
1618 },
1619 "post": {
1620     "description": "Deliver Wi-Fi AP's information for an unboxing Device to connect to it.\n",
1621     "parameters": [
1622         {"$ref": "#/parameters/interface-all"},
1623         {
1624             "name": "body",
1625             "in": "body",
1626             "required": true,
1627             "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1628             "x-example":
1629                 {
1630                     "tnn": "Home_AP_SSID",
1631                     "cd": "Home_AP_PWD",
1632                     "wat": "WPA2_PSK",
1633                     "wet": "AES"
1634                 }
1635         }
1636     ],
1637     "responses": {
1638         "200": {
1639             "description" : "",
1640             "x-example":
1641                 {
1642                     "tnn": "Home_AP_SSID",
1643                     "swmt" : ["A", "B", "G"],
1644                     "swf": ["2.4G", "5G"],
1645                     "cd": "Home_AP_PWD",
1646                     "wat": "WPA2_PSK",
1647                     "wet": "AES",
1648                     "swat": ["WPA_PSK", "WPA2_PSK"],
1649                     "swet": ["TKIP", "AES", "TKIP_AES"]
1650                 },
1651             "schema": { "$ref": "#/definitions/WiFiConf" }
1652         }
1653     }
1654 },
1655 },
1656 "/WiFiConfResURI?if=oic.if.baseline" : {
1657     "get": {
1658         "description": "WiFiConf Resource stores essential information to help an unboxing
1659 Device\nto connect to an existing Wi-Fi AP.\n",
1660         "parameters": [
1661             {"$ref": "#/parameters/interface-all"}
1662         ],
1663         "responses": {
1664             "200": {
1665                 "description" : "",
1666                 "x-example":
1667                     {
1668                         "rt": ["oic.r.wificonf"],
1669                         "if": ["oic.if.rw", "oic.if.baseline"],
1670                         "swmt" : ["A", "B", "G"],
1671                         "swf": ["2.4G", "5G"],
1672                         "tnn": "Home_AP_SSID",
1673                         "cd": "Home_AP_PWD",
1674                         "wat": "WPA2_PSK",

```

```

1675         "wet": "TKIP",
1676         "swat": ["WPA_PSK", "WPA2_PSK"],
1677         "swet": ["TKIP", "AES", "TKIP_AES"]
1678     },
1679     "schema": { "$ref": "#/definitions/WiFiConf" }
1680 }
1681 },
1682 },
1683 "post": {
1684     "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
1685     "parameters": [
1686         {"$ref": "#/parameters/interface-all"},
1687         {
1688             "name": "body",
1689             "in": "body",
1690             "required": true,
1691             "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1692             "x-example":
1693                 {
1694                     "tnn": "Home_AP_SSID",
1695                     "cd": "Home_AP_PWD",
1696                     "wat": "WPA2_PSK",
1697                     "wet": "AES"
1698                 }
1699         }
1700     ],
1701     "responses": {
1702         "200": {
1703             "description": "",
1704             "x-example":
1705                 {
1706                     "rt": ["oic.r.wificonf"],
1707                     "if": ["oic.if.rw", "oic.if.baseline"],
1708                     "tnn": "Home_AP_SSID",
1709                     "swmt": ["A", "B", "G"],
1710                     "swf": ["2.4G", "5G"],
1711                     "cd": "Home_AP_PWD",
1712                     "wat": "WPA2_PSK",
1713                     "wet": "AES",
1714                     "swat": ["WPA_PSK", "WPA2_PSK"],
1715                     "swet": ["TKIP", "AES", "TKIP_AES"]
1716                 },
1717             "schema": { "$ref": "#/definitions/WiFiConf" }
1718         }
1719     }
1720 }
1721 },
1722 },
1723 "parameters": {
1724     "interface-all": {
1725         "in": "query",
1726         "name": "if",
1727         "type": "string",
1728         "enum": ["oic.if.rw", "oic.if.baseline"]
1729     }
1730 },
1731 "definitions": {
1732     "WiFiConf": {
1733         "properties": {
1734             "rt": {
1735                 "description": "Resource Type of the Resource",
1736                 "items": {
1737                     "enum": ["oic.r.wificonf"],
1738                     "type": "string",
1739                     "maxLength": 64
1740                 },
1741                 "minItems": 1,
1742                 "uniqueItems": true,
1743                 "readOnly": true,
1744                 "type": "array"

```

```

1745     },
1746     "tnn" : {
1747         "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1748         "pattern": "^.*$",
1749         "type": "string"
1750     },
1751     "swmt" : {
1752         "description": "Indicates supported Wi-Fi mode types. It can be multiple",
1753         "items": {
1754             "description": "Supported Wi-Fi Mode Type.",
1755             "enum": [
1756                 "A",
1757                 "B",
1758                 "G",
1759                 "N",
1760                 "AC"
1761             ],
1762             "type": "string"
1763         },
1764         "readOnly": true,
1765         "type": "array"
1766     },
1767     "wat" : {
1768         "description": "Indicates Wi-Fi Auth Type",
1769         "enum": [
1770             "None",
1771             "WEP",
1772             "WPA_PSK",
1773             "WPA2_PSK"
1774         ],
1775         "type": "string"
1776     },
1777     "n" : {
1778         "$ref":
1779         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1780         schema.json#/definitions/n"
1781     },
1782     "swat" : {
1783         "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
1784         "items": {
1785             "description": "Indicates Wi-Fi Auth Type",
1786             "enum": [
1787                 "None",
1788                 "WEP",
1789                 "WPA_PSK",
1790                 "WPA2_PSK"
1791             ],
1792             "type": "string"
1793         },
1794         "readOnly": true,
1795         "type": "array"
1796     },
1797     "swf" : {
1798         "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
1799 Valid values are ('2.4G', '5G')",
1800         "items": {
1801             "pattern": "^(2\\.4|5)G$",
1802             "type": "string"
1803         },
1804         "readOnly": true,
1805         "type": "array"
1806     },
1807     "swet" : {
1808         "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
1809         "items": {
1810             "description": "Indicates Wi-Fi Encryption Type",
1811             "enum": [
1812                 "None",
1813                 "WEP_64",
1814                 "WEP_128",

```

```

1815         "TKIP",
1816         "AES",
1817         "TKIP_AES"
1818     ],
1819     "type": "string"
1820 },
1821 "readOnly": true,
1822 "type": "array"
1823 },
1824 "wet" : {
1825     "description": "Indicates Wi-Fi Encryption Type",
1826     "enum": [
1827         "None",
1828         "WEP_64",
1829         "WEP_128",
1830         "TKIP",
1831         "AES",
1832         "TKIP_AES"
1833     ],
1834     "type": "string"
1835 },
1836 "cd" : {
1837     "description": "Indicates credential information of Wi-Fi AP",
1838     "pattern": "^.*$",
1839     "type": "string"
1840 },
1841 "id" : {
1842     "$ref":
1843     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1844     schema.json#/definitions/id"
1845 },
1846 "if" : {
1847     "description": "The OCF Interfaces supported by this Resource",
1848     "items": {
1849         "enum": [
1850             "oic.if.rw",
1851             "oic.if.baseline"
1852         ],
1853         "type": "string",
1854         "maxLength": 64
1855     },
1856     "minItems": 2,
1857     "uniqueItems": true,
1858     "readOnly": true,
1859     "type": "array"
1860 }
1861 },
1862 "type" : "object",
1863 "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
1864 },
1865 "WiFiConfUpdate" : {
1866     "properties": {
1867         "wat" : {
1868             "description": "Indicates Wi-Fi Auth Type",
1869             "enum": [
1870                 "None",
1871                 "WEP",
1872                 "WPA_PSK",
1873                 "WPA2_PSK"
1874             ]
1875         },
1876         "cd" : {
1877             "description": "Indicates credential information of Wi-Fi AP",
1878             "pattern": "^.*$",
1879             "type": "string"
1880         },
1881         "wet" : {
1882             "description": "Indicates Wi-Fi Encryption Type",
1883             "enum": [
1884                 "None",

```

```

1885         "WEP_64",
1886         "WEP_128",
1887         "TKIP",
1888         "AES",
1889         "TKIP_AES"
1890     ]
1891 },
1892     "tnn" : {
1893         "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1894         "pattern": "^.*$",
1895         "type": "string"
1896     }
1897 },
1898     "type" : "object",
1899     "required":["tnn", "wat", "wet"]
1900 }
1901 }
1902 }
1903

```

1904 A.4.5 Property definition

1905 Table A.6 defines the Properties that are part of the "oic.r.wificonf" Resource Type.

1906 **Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.wificonf".**

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

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

1907 **A.4.6 CRUDN behaviour**

1908 Table A.7 defines the CRUDN operations that are supported on the "oic.r.wificonf" Resource Type.

1909 **Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.wificonf".**

Create	Read	Update	Delete	Notify
	get	post		observe

1910 **A.5 eSIM Easy Setup Collection**

1911 **A.5.1 Introduction**

1912 The eSIMEasySetup Resource Type stores useful information including Remote SIM Provisioning (RSP) status,

1913 and RSP last error code which was produced in the process of eSIM Easy Setup.

1914 Note that the eSIM Easy Setup Resource is a Collection Resource, which contains Links to RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.

1917

1918 **A.5.2 Example URI**

1919 /eSIMEasySetupResURI

1920 **A.5.3 Resource type**

1921 The Resource Type is defined as: "oic.r.esimeasysetup".

1922 **A.5.4 OpenAPI 2.0 definition**

```

1923 {
1924   "swagger": "2.0",
1925   "info": {
1926     "title": "eSIM Easy Setup Collection",
1927     "version": "2020-09-01",
1928     "license": {
1929       "name": "OCF Data Model License",
1930       "url":
1931 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
1932 CENSE.md",
1933       "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
1934     },
1935     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
1936   },
1937   "schemes": ["http"],
1938   "consumes": ["application/json"],
1939   "produces": ["application/json"],
1940   "paths": {

```



```

1941     "/eSIMEasySetupResURI?if=oic.if.ll" : {
1942         "get": {
1943             "description": "The eSIMEasySetup Resource Type stores useful information including Remote
1944 SIM Provisioning (RSP) status, and RSP last error which was produced in the process of eSIM Easy
1945 Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links to
1946 RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
1947             "parameters": [
1948                 {"$ref": "#/parameters/interface-all"}
1949             ],
1950             "responses": {
1951                 "200": {
1952                     "description": "",
1953                     "x-example":
1954                         [
1955                             {
1956                                 "href": "/eSIMEasySetupResURI",
1957                                 "rt": ["oic.r.esimeasysetup"],
1958                                 "if": ["oic.if.b", "oic.if.baseline", "oic.if.ll"],
1959                                 "p":{"bm":3},
1960                                 "eps": [
1961                                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1962                                 ],
1963                                 "rel":["self", "item"]
1964                             },
1965                             {
1966                                 "href": "/RSPConfResURI",
1967                                 "rt": ["oic.r.rspconf"],
1968                                 "if": ["oic.if.baseline", "oic.if.rw"],
1969                                 "p":{"bm":3},
1970                                 "eps": [
1971                                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1972                                 ]
1973                             },
1974                             {
1975                                 "href": "/RSPCapabilityResURI",
1976                                 "rt": ["oic.r.rspcapability"],
1977                                 "if": ["oic.if.baseline", "oic.if.r"],
1978                                 "p":{"bm":3},
1979                                 "eps": [
1980                                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1981                                 ]
1982                             }
1983                         ],
1984                     "schema": { "$ref": "#/definitions/slinks" }
1985                 }
1986             }
1987         },
1988     },
1989     "/eSIMEasySetupResURI?if=oic.if.b" : {
1990         "get": {
1991             "description": "The eSIMEasySetup Resource Type stores useful information including Remote
1992 SIM Provisioning (RSP) status, and RSP last error code which was produced in the process of eSIM
1993 Easy Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links
1994 to RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
1995             "parameters": [
1996                 {"$ref": "#/parameters/interface-all"}
1997             ],
1998             "responses": {
1999                 "200": {
2000                     "description": "",
2001                     "x-example":
2002                         [
2003                             {
2004                                 "href": "/eSIMEasySetupResURI",
2005                                 "rep":{"
2006                                     "ps" : "User confirmation pending",
2007                                     "ler": "",
2008                                     "lec": "",
2009                                     "led": "",
2010                                     "euc": "Undefined"

```

```

2011         }
2012     },
2013     {
2014         "href": "/RSPConfResURI",
2015         "rep": {
2016             "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
2017             "pm":
2018 "vyU4WgqJAQIDBAUGBwgJkRNTZXJ2aWNlUHJvdmlkZXJOYWllkgtQcm9maWxlTmFtZZMBAJQCAACVAQI=",
2019             "cc": "",
2020             "ccr": true
2021         }
2022     },
2023     {
2024         "href": "/RSPCapabilityURI",
2025         "rep": {
2026             "euccinfo":
2027 "vyJ7gQMCAACCAwICAYMDQQEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDagMAiAIEkKkWBRRmWhQzlnwaLF24tSyWfxCgV7p
2028 csqoWBRRmWhQzlnwaLF24tSyWfxCgV7pcsosBAGQDAQAADBxMDAwMDAwMDAwMDAwMDAw",
2029             "deviceinfo":
2030 "oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMBAUGhAMFBgeFAwYHCIYDBwgJhwMICQo="
2031         }
2032     }
2033 ],
2034     "schema": { "$ref": "#/definitions/sbatch" }
2035 }
2036 },
2037 },
2038 "post": {
2039     "description": "Able to deliver RSP Configuration, RSP Capability and other
2040 configuration\ninformation in a batch by utilizing 'batch' OCF Interface.\nIf you want to deliver in
2041 a batch,\nyou can write all Properties you want to send with a 'batch' OCF Interface.\nThe below
2042 example is the case to send eSIMEasySetup and RSP configuration\n(i.e., RSP Procedure Status,
2043 Activation Code, Confirmation Code required) in a batch.\n",
2044     "parameters": [
2045         { "$ref": "#/parameters/interface-update" },
2046         {
2047             "name": "body",
2048             "in": "body",
2049             "required": true,
2050             "schema": { "$ref": "#/definitions/sbatch-update" },
2051             "x-example":
2052             [
2053                 {
2054                     "href": "/eSIMEasySetupResURI",
2055                     "rep": {
2056                         "euc": "Download OK"
2057                     }
2058                 },
2059                 {
2060                     "href": "/RSPConfResURI",
2061                     "rep": {
2062                         "cc": "102030405"
2063                     }
2064                 }
2065             ]
2066         }
2067     ],
2068     "responses": {
2069         "200": {
2070             "description": "",
2071             "x-example":
2072             [
2073                 {
2074                     "href": "/eSIMEasySetupResURI",
2075                     "rep": {
2076                         "ps": "Confirmation received",
2077                         "ler": "",
2078                         "lec": "",
2079                         "led": "",
2080                         "euc": "Download OK"

```

```

2081     }
2082     },
2083     {
2084         "href": "/RSPConfResURI",
2085         "rep": {
2086             "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
2087             "pm":
2088 "vyU4WgqJAQIDBAUGBwgJkRNTZXJ2aWNlUHJvdmlkZXZJOYw1lkggtQcm9maWxlTmFtZmZMBAJQCAACVAQI=",
2089             "cc": "102030405",
2090             "ccr": true
2091         }
2092     },
2093     {
2094         "href": "/RSPCapabilityResURI",
2095         "rep": {
2096             "euicinfo":
2097 "vyJ7gQMCAACCAwICAYMDQqEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDagMAiAIEkKkWBRRmWhQz1nwaLF24tSyWfxCgV7p
2098 csqoWBRRmWhQz1nwaLF24tSyWfxCgV7pcsosBAGQDAQAADBxMDAwMDAwMDAwMDAwMDAw",
2099             "deviceinfo":
2100 "oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMDBAUGhAMFBGefAWYHClYDBwgJhwMICQo="
2101         }
2102     }
2103 ],
2104     "schema": { "$ref": "#/definitions/sbatch" }
2105 }
2106 }
2107 }
2108 },
2109 "/eSIMEasySetupResURI?if=oic.if.baseline": {
2110     "get": {
2111         "description": "The eSIMEasySetup Resource Type stores useful information including Remote
2112 SIM Provisioning (RSP) status,\n and RSP last error code which was produced in the process of eSIM
2113 Easy Setup.\nNote that the eSIM Easy Setup Resource is a Collection Resource, which contains Links
2114 to RSPConf, and RSPCapability Resources and may additionally contain Links to other Resources.\n",
2115         "parameters": [
2116             { "$ref": "#/parameters/interface-all" }
2117         ],
2118         "responses": {
2119             "200": {
2120                 "description": "",
2121                 "x-example":
2122                 {
2123                     "rt": ["oic.r.esimeasyssetup"],
2124                     "if": ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
2125                     "ps": ["Undefined", "Initiated", "User confirmation pending", "Confirmation
2126 received", "Downloaded", "Installed", "Error"],
2127                     "ler": "",
2128                     "lec": "",
2129                     "led": "",
2130                     "euc": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download
2131 OK", "Download and Enable OK"],
2132                     "links": [
2133                         {
2134                             "href": "/eSIMEasySetupResURI",
2135                             "rt": ["oic.r.esimeasyssetup", "oic.wk.col"],
2136                             "if": ["oic.if.b", "oic.if.baseline", "oic.if.ll"],
2137                             "p":{"bm":3},
2138                             "eps": [
2139                                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
2140                             ],
2141                             "rel":["self", "item"]
2142                         },
2143                         {
2144                             "href": "/RSPConfResURI",
2145                             "rt": ["oic.r.rspsconf"],
2146                             "if": ["oic.if.baseline", "oic.if.rw"],
2147                             "p":{"bm":3},
2148                             "eps": [
2149                                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
2150                             ]
2151                         }
2152                     ]
2153                 }
2154             }
2155         }
2156     }
2157 }

```

```

2151         },
2152         {
2153             "href": "/RSPCapabilityResURI",
2154             "rt": ["oic.r.rspcapability"],
2155             "if": ["oic.if.baseline", "oic.if.r"],
2156             "p": {"bm": 3},
2157             "eps": [
2158                 {"ep": "coaps://[fe80:b1d6]:1111", "pri": 2}
2159             ]
2160         }
2161     ],
2162 },
2163     "schema": { "$ref": "#/definitions/eSIMEasySetup" }
2164 }
2165 }
2166 }
2167 }
2168 },
2169 "parameters": {
2170     "interface-all" : {
2171         "in" : "query",
2172         "name" : "if",
2173         "type" : "string",
2174         "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
2175     },
2176     "interface-update" : {
2177         "in" : "query",
2178         "name" : "if",
2179         "type" : "string",
2180         "enum" : ["oic.if.b"]
2181     }
2182 },
2183 "definitions": {
2184     "oic.oic-link": {
2185         "type": "object",
2186         "properties": {
2187             "anchor": {
2188                 "$ref":
2189 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2190 schema.json#/definitions/anchor"
2191             },
2192             "di": {
2193                 "$ref":
2194 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2195 schema.json#/definitions/di"
2196             },
2197             "eps": {
2198                 "$ref":
2199 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2200 schema.json#/definitions/eps"
2201             },
2202             "href": {
2203                 "$ref":
2204 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2205 schema.json#/definitions/href"
2206             },
2207             "ins": {
2208                 "$ref":
2209 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2210 schema.json#/definitions/ins"
2211             },
2212             "p": {
2213                 "$ref":
2214 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2215 schema.json#/definitions/p"
2216             },
2217             "rel": {
2218                 "$ref":
2219 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2220 schema.json#/definitions/rel_array"

```

```

2221     },
2222     "title": {
2223       "$ref":
2224       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2225       schema.json#/definitions/title"
2226     },
2227     "type": {
2228       "$ref":
2229       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2230       schema.json#/definitions/type"
2231     },
2232     "if": {
2233       "description": "The OCF Interfaces supported by the target Resource",
2234       "items": {
2235         "enum": [
2236           "oic.if.baseline",
2237           "oic.if.ll",
2238           "oic.if.b",
2239           "oic.if.r",
2240           "oic.if.rw"
2241         ],
2242         "type": "string",
2243         "maxLength": 64
2244       },
2245       "minItems": 1,
2246       "uniqueItems": true,
2247       "type": "array"
2248     },
2249     "rt": {
2250       "description": "Resource Type of the target Resource",
2251       "items": {
2252         "maxLength": 64,
2253         "type": "string"
2254       },
2255       "minItems": 1,
2256       "uniqueItems": true,
2257       "type": "array"
2258     }
2259   },
2260   "required": [
2261     "href",
2262     "rt",
2263     "if"
2264   ]
2265 },
2266 "slinks" : {
2267   "type": "array",
2268   "items": {
2269     "$ref": "#/definitions/oic.oic-link"
2270   }
2271 },
2272 "sbatch" : {
2273   "minItems" : 1,
2274   "items" : {
2275     "additionalProperties": true,
2276     "properties": {
2277       "href": {
2278         "$ref":
2279         "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2280         schema.json#/definitions/href"
2281       },
2282       "rep": {
2283         "description": "The response payload from a single Resource",
2284         "type": "object",
2285         "anyOf": [
2286           {
2287             "$ref": "#/definitions/eSIMEasySetup"
2288           },
2289           {
2290             "$ref": "https://openconnectivityfoundation.github.io/core-

```

```

2291 extensions/swagger2.0/oic.r.rspconf.swagger.json#/definitions/RSPConf"
2292     },
2293     {
2294         "$ref": "https://openconnectivityfoundation.github.io/core-
2295 extensions/swagger2.0/oic.r.rspcapability.swagger.json#/definitions/RSPCapability"
2296     }
2297 ]
2298 }
2299 },
2300 "required": [
2301     "href",
2302     "rep"
2303 ],
2304 "type": "object"
2305 },
2306 "type" : "array"
2307 },
2308 "sbatch-update" : {
2309     "minItems" : 1,
2310     "items" : {
2311         "additionalProperties": true,
2312         "description": "Array of Resource representations to apply to the batch Collection, \nusing
2313 href to indicate which resource(s) in the batch to update. \nIf the href Property is empty,
2314 effectively making the URI reference to the Collection itself, \nthe representation is to be applied
2315 to all Resources in the batch\n",
2316         "properties": {
2317             "href": {
2318                 "$ref":
2319 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
2320 schema.json#/definitions/href"
2321             },
2322             "rep": {
2323                 "description": "The response payload from a single Resource",
2324                 "type": "object",
2325                 "anyOf": [
2326                     {
2327                         "$ref": "#/definitions/eSIMEasySetupUpdate"
2328                     },
2329                     {
2330                         "$ref": "https://openconnectivityfoundation.github.io/core-
2331 extensions/swagger2.0/oic.r.rspconf.swagger.json#/definitions/RSPConfUpdate"
2332                     }
2333                 ]
2334             }
2335         },
2336         "required": [
2337             "href",
2338             "rep"
2339         ],
2340         "type": "object"
2341     },
2342     "type" : "array"
2343 },
2344 "eSIMEasySetup" : {
2345     "properties": {
2346         "n" : {
2347             "$ref":
2348 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2349 schema.json#/definitions/n"
2350         },
2351         "rts" : {
2352             "description": "Resource Type of the Resources within the Collection",
2353             "items": {
2354                 "maxLength": 64,
2355                 "type": "string"
2356             },
2357             "minItems": 1,
2358             "uniqueItems": true,
2359             "readOnly": true,
2360             "type": "array"

```

```

2361     },
2362     "id" : {
2363         "$ref":
2364         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2365         schema.json#/definitions/id"
2366     },
2367     "rts-m" : {
2368         "description": "Resource Type of the mandatory Resources within the Collection",
2369         "items": {
2370             "maxLength": 64,
2371             "type": "string"
2372         },
2373         "minItems": 1,
2374         "uniqueItems": true,
2375         "readOnly": true,
2376         "type": "array"
2377     },
2378     "if" : {
2379         "description": "The OCF Interfaces supported by this Resource",
2380         "items": {
2381             "enum": [
2382                 "oic.if.ll",
2383                 "oic.if.baseline",
2384                 "oic.if.b"
2385             ],
2386             "type": "string",
2387             "maxLength": 64
2388         },
2389         "minItems": 3,
2390         "uniqueItems": true,
2391         "readOnly": true,
2392         "type": "array"
2393     },
2394     "rt" : {
2395         "items": {
2396             "enum": [
2397                 "oic.r.esimeasysetup"
2398             ],
2399             "type": "string",
2400             "maxLength": 64
2401         },
2402         "minItems": 1,
2403         "type": "array",
2404         "uniqueItems": true,
2405         "readOnly": true
2406     },
2407     "ps" : {
2408         "description": "Indicates the steps in Remote SIM Provisioning.\n",
2409         "enum": ["Undefined", "Initiated", "User confirmation pending", "Confirmation received",
2410 "Downloaded", "Installed", "Error"],
2411         "readOnly": true,
2412         "type": "string"
2413     },
2414     "ler" : {
2415         "description": "Error Reason returned by the LPA while eSIM Easy Setup. \nIt indicates
2416 where it was occurred.\n(e.g., ES9+.GetBoundProfilePackage(Fail),
2417 ES10b.LoadBoundProfilePackage(Fail))\n",
2418         "readOnly": true,
2419         "type": "string"
2420     },
2421     "lec" : {
2422         "description": "Error Code returned by the LPA while eSIM Easy Setup. \nIt indicates why
2423 it was occurred.\nIt is mapped to the GSMA error status (e.g., \"8.8.1-3.8\", \"7\", \"6A 80\")\n",
2424         "readOnly": true,
2425         "type": "string"
2426     },
2427     "led" : {
2428         "description": "Optional error description \nreturned by the LPA while eSIM Easy Setup.
2429 (e.g., Invalid SM-DP+ Address)\n",
2430         "readOnly": true,

```

```

2431         "type": "string"
2432     },
2433     "euc" : {
2434         "description": "End User Consent for RSP.\n",
2435         "enum": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download OK",
2436 "Download and Enable OK"],
2437         "type": "string"
2438     },
2439     "links" : {
2440         "type": "array",
2441         "description": "A set of OCF Links.",
2442         "items": {
2443             "$ref": "#/definitions/oic.oic-link"
2444         },
2445         "readOnly":true
2446     }
2447 },
2448 "type" : "object",
2449 "required": ["ps","ler","lec","euc"]
2450 },
2451 "eSIMEasySetupUpdate" : {
2452     "additionalProperties": true,
2453     "description": "Update to writeable values in eSIMEasySetupResURI",
2454     "properties": {
2455         "euc" : {
2456             "description": "End User Consent for RSP.\n",
2457             "enum": ["Undefined", "Timeout", "Download Reject", "Download Postponed", "Download OK",
2458 "Download and Enable OK"],
2459             "type": "string"
2460         }
2461     },
2462     "type": "object",
2463     "required": ["euc"]
2464 }
2465 }
2466 }
2467

```

2468 A.5.5 Property definition

2469 Table A.8 defines the Properties that are part of the "oic.r.esimeasyssetup" Resource Type.

2470 **Table A.8 – The Property definitions of the Resource with type "rt" =**
2471 **"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.

2472 **A.5.6 CRUDN behaviour**

2473 Table A.9 defines the CRUDN operations that are supported on the "oic.r.esimeasyssetup" Resource
2474 Type.

2475 **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

2476 **A.6 Remote SIM Provisioning Capability**

2477 **A.6.1 Introduction**

2478 RSPCapability Resource stores information to help a service provider to provide appropriate
2479 cellular plans to an end user.
2480

2481 **A.6.2 Example URI**

2482 /RSPCapabilityResURI

2483 **A.6.3 Resource type**

2484 The Resource Type is defined as: "oic.r.rspcapability".

2485 **A.6.4 OpenAPI 2.0 definition**

```

2486 {
2487   "swagger": "2.0",
2488   "info": {
2489     "title": "Remote SIM Provisioning Capability",
2490     "version": "2020-09-01",
2491     "license": {
2492       "name": "OCF Data Model License",
2493       "url":
2494         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
2495         CENSE.md",
2496       "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
2497     },
2498     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
2499   },
2500   "schemes": ["http"],
2501   "consumes": ["application/json"],
2502   "produces": ["application/json"],
2503   "paths": {
2504     "/RSPCapabilityResURI" : {
2505       "get": {
2506         "description": "RSPCapability Resource stores information to help a service provider to
2507         provide appropriate cellular plans to an end user.\n",
2508         "parameters": [
2509           {"$ref": "#/parameters/interface-all"}
2510         ],
2511         "responses": {
2512           "200": {
2513             "description" : "",
2514             "x-example":
2515               { "rt": ["oic.r.rspcapability"],
2516                 "if": ["oic.if.r", "oic.if.baseline"],
2517                 "euicinfo":
2518                   "vyJ7gQMCAACCAwICAYMDQqEFhAyBAQCCAwVJQIMCFkWFBAV/NuCGAwkCAIcDagMAiAIEkKkWBBrmWhQz1nwaLF24tSyWfxcgV7p
2519                   csqoWBBrmWhQz1nwaLF24tSyWfxcgV7pcsosBAGQDAQAADBxMDAwMDAwMDAwMDAwMDAw",
2520                 "deviceinfo" :
2521                   "oDCABBI0VnihKIADAQIDgQMCAwSCAwMEBYMDBAUGHAMFBgeFAwYHCiYDBwgJhwMICQo="
2522               },
2523           "schema": { "$ref": "#/definitions/RSPCapability" }

```

```

2524     }
2525     }
2526   }
2527 }
2528 },
2529 "parameters": {
2530   "interface-all" : {
2531     "in" : "query",
2532     "name" : "if",
2533     "type" : "string",
2534     "enum" : ["oic.if.r", "oic.if.baseline"]
2535   }
2536 },
2537 "definitions": {
2538   "RSPCapability" : {
2539     "properties": {
2540       "rt" : {
2541         "description": "Resource Type of the Resource",
2542         "items": {
2543           "enum": ["oic.r.rspcapability"],
2544           "type": "string",
2545           "maxLength": 64
2546         },
2547         "minItems": 1,
2548         "uniqueItems": true,
2549         "readOnly": true,
2550         "type": "array"
2551       },
2552       "euicccinfo" : {
2553         "description": "Refers to EUICCCInfo2 defined in GSMA SGP.22 Annex H.This value type shall
2554 be encoded as Major Type 2.",
2555         "type": "string",
2556         "readOnly": true,
2557         "maxLength": 1024
2558       },
2559       "deviceinfo" : {
2560         "description": "Refers to DeviceInfo defined in GSMA SGP.22 Annex H.This value type shall
2561 be encoded as Major Type 2.",
2562         "type": "string",
2563         "readOnly": true,
2564         "maxLength": 128
2565       },
2566       "n" : {
2567         "$ref":
2568 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2569 schema.json#/definitions/n"
2570       },
2571       "id" : {
2572         "$ref":
2573 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2574 schema.json#/definitions/id"
2575       },
2576       "if" : {
2577         "description": "The OCF Interfaces supported by this Resource",
2578         "items": {
2579           "enum": [
2580             "oic.if.r",
2581             "oic.if.baseline"
2582           ],
2583           "type": "string",
2584           "maxLength": 64
2585         },
2586         "minItems": 2,
2587         "uniqueItems": true,
2588         "readOnly": true,
2589         "type": "array"
2590       }
2591     },
2592     "type" : "object",
2593     "required":["euicccinfo", "deviceinfo"]

```

2594 }
 2595 }
 2596 }
 2597 }

2598 **A.6.5 Property definition**

2599 Table A.10 defines the Properties that are part of the "oic.r.rspcapability" Resource Type.

2600 **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

2601 **A.6.6 CRUDN behaviour**

2602 Table A.11 defines the CRUDN operations that are supported on the "oic.r.rspcapability" Resource
 2603 Type.

2604 **Table A.11 – The CRUDN operations of the Resource with type "rt" = "oic.r.rspcapability".**

Create	Read	Update	Delete	Notify
	get			observe

2605 **A.7 RSP Configuration**

2606 **A.7.1 Introduction**

2607 RSPConf Resource stores the information
 2608 used to download and install an eSIM Profile to an eSIM capable IoT device.
 2609 It comprises SM-DP+ server FQDN and Activation Code Token
 2610 binding to a specific subscription as defined by GSMA SGP.22.

2611 **A.7.2 Example URI**

2612 /RSPConfResURI

2613 A.7.3 Resource type

2614 The Resource Type is defined as: "oic.r.rspconf".

2615 A.7.4 OpenAPI 2.0 definition

```
2616 {
2617   "swagger": "2.0",
2618   "info": {
2619     "title": "RSP Configuration",
2620     "version": "2020-09-01",
2621     "license": {
2622       "name": "OCF Data Model License",
2623       "url":
2624         "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
2625         CENSE.md",
2626       "x-copyright": "Copyright 2020 Open Connectivity Foundation, Inc. All rights reserved."
2627     },
2628     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
2629   },
2630   "schemes": ["http"],
2631   "consumes": ["application/json"],
2632   "produces": ["application/json"],
2633   "paths": {
2634     "/RSPConfResURI": {
2635       "get": {
2636         "description": "RSPConf Resource stores the information \nused to download and install an
2637         eSIM Profile to an eSIM capable IoT device.\nIt comprises SM-DP+ server FQDN and Activation Code
2638         Token\n binding to a specific subscription as defined by GSMA SGP.22.",
2639         "parameters": [
2640           { "$ref": "#/parameters/interface" }
2641         ],
2642         "responses": {
2643           "200": {
2644             "description": "",
2645             "x-example": {
2646               "rt": ["oic.r.rspconf"],
2647               "if": ["oic.if.rw", "oic.if.baseline"],
2648               "ac": "",
2649               "pm": "",
2650               "ccr": false
2651             },
2652             "schema": { "$ref": "#/definitions/RSPConf" }
2653           }
2654         }
2655       },
2656       "post": {
2657         "description": "Update Properties of the RSPConf Resource (deliver Activation Code in this
2658         example).\n",
2659         "parameters": [
2660           { "$ref": "#/parameters/interface-rw" },
2661           {
2662             "name": "body",
2663             "in": "body",
2664             "required": true,
2665             "schema": { "$ref": "#/definitions/RSPConfUpdate" },
2666             "x-example": {
2667               "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815"
2668             }
2669           }
2670         ],
2671         "responses": {
2672           "200": {
2673             "description": "",
2674             "x-example": {
2675               "ac": "1$SMDP.GSMA.COM$04386-AGYFT-A74Y8-3F815",
2676               "pm": "",
2677               "ccr": false
2678             },
2679             "schema": { "$ref": "#/definitions/RSPConf" }
2680           }
2681         }
2682       }
2683     }
2684   }
2685 }
```

```

2680     }
2681   }
2682 }
2683 }
2684 },
2685 "parameters": {
2686   "interface": {
2687     "in": "query",
2688     "name": "if",
2689     "type": "string",
2690     "enum": ["oic.if.rw", "oic.if.baseline"]
2691   },
2692   "interface-rw": {
2693     "in": "query",
2694     "name": "if",
2695     "type": "string",
2696     "enum": ["oic.if.rw"]
2697   }
2698 },
2699 "definitions": {
2700   "RSPConf": {
2701     "properties": {
2702       "rt": {
2703         "description": "The Resource Type.",
2704         "items": {
2705           "enum": ["oic.r.rspconf"],
2706           "maxLength": 64,
2707           "type": "string"
2708         },
2709         "minItems": 1,
2710         "uniqueItems": true,
2711         "readOnly": true,
2712         "type": "array"
2713       },
2714       "ac": {
2715         "description": "The information needed to provision an eSIM device.",
2716         "maxLength": 256,
2717         "type": "string"
2718       },
2719       "pm": {
2720         "description": "Refers to ProfileInfo in GSMA SGP.22 Annex H.This value type shall be
2721 encoded as Major Type 2",
2722         "maxLength": 2048,
2723         "type": "string",
2724         "readOnly": true
2725       },
2726       "cc": {
2727         "description": "A code entered by an end user required by the SM-DP+ \nto confirm the
2728 download and installation of an eSIM Profile.\nThe Confirmation Code is provided from a service
2729 provider to the end user.\n",
2730         "maxLength": 64,
2731         "type": "string"
2732       },
2733       "ccr": {
2734         "description": "Indicates whether a Confirmation Code is required.\n",
2735         "maxLength": 64,
2736         "type": "boolean",
2737         "readOnly": true
2738       },
2739       "n": {
2740         "$ref":
2741 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2742 schema.json#/definitions/n"
2743       },
2744       "id": {
2745         "$ref":
2746 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
2747 schema.json#/definitions/id"
2748       },
2749       "if": {

```

```

2750     "description": "The OCF Interface set supported by this Resource.",
2751     "items": {
2752       "enum": [
2753         "oic.if.rw",
2754         "oic.if.baseline"
2755       ],
2756       "type": "string"
2757     },
2758     "minItems": 2,
2759     "uniqueItems": true,
2760     "readOnly": true,
2761     "type": "array"
2762   }
2763 },
2764 "type": "object",
2765 "required": ["ac", "pm", "ccr"]
2766 },
2767 "RSPConfUpdate": {
2768   "properties": {
2769     "ac": {
2770       "description": "The information needed to provision an eSIM device.",
2771       "maxLength": 256,
2772       "type": "string"
2773     },
2774     "cc": {
2775       "description": "A code entered by an end user required by the SM-DP+ \nto confirm the
2776 download and installation of an eSIM Profile.\nThe Confirmation Code is provided from a service
2777 provider to the end user.\n",
2778       "maxLength": 64,
2779       "type": "string"
2780     }
2781   }
2782 }
2783 }
2784 }
2785

```

2786 **A.7.5 Property definition**

2787 Table A.12 defines the Properties that are part of the "oic.r.rspconf" Resource Type.

2788 **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.

2789 **A.7.6 CRUDN behaviour**

2790 Table A.13 defines the CRUDN operations that are supported on the "oic.r.rspconf" Resource Type.

2791 **Table A.13 – The CRUDN operations of the Resource with type "rt" = "oic.r.rspconf".**

Create	Read	Update	Delete	Notify
	get	post		observe

2792

2793