BS ISO/IEC 30118-1:2018

BSI Standards Publication

Information technology — Open Connectivity Foundation (OCF) Specification

Part 1: Core specification

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

National foreword

This British Standard is the UK implementation of ISO/IEC 30118-1:2018.

The UK participation in its preparation was entrusted to Technical Committee IST/-/99, Participating IST members.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2018
Published by BSI Standards Limited 2018

ISBN 978 0 580 99409 8

ICS 35.200

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2018.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

INTERNATIONAL STANDARD

ISO/IEC 30118-1

First edition
2018-11

Information technology — Open Connectivity Foundation (OCF) Specification —

Part 1: Core specification

Technologies de l'information — Spécification de la Fondation pour la connectivité ouverte (Fondation OCF) —

Partie 1: Spécification du cœur

Reference number
ISO/IEC 30118-1:2018(E)

Documents
Document Sharing Hub
© ISO/IEC 2018

Full document details: BS ISO/IEC 30118-1:2018

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2018

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement. © ISO/IEC 2018 – All rights reserved

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by the Open Connectivity Foundation (OCF) (as the OCF Core Specification v1.0.0) and drafted in accordance with its editorial rules. It was adopted, under the JTC 1 PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

A list of all parts in the ISO/IEC 30118 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30118-1:2018

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

CONTENTS

1 Scope 15
2 Normative references 15
3 Terms, definitions, symbols and abbreviations 18
3.1 Terms and definitions 18
3.2 Symbols and abbreviations 21
3.3 Conventions 22
3.4 Data types 22
4 Document conventions and organization 23
5 Architecture 24
5.1 Overview 24
5.2 Principle 25
5.3 Functional block diagram 26
5.4 Framework 27
5.5 Example Scenario with roles 27
5.6 Example Scenario: Bridging to Non- OCF ecosystem 28
6 Identification and addressing 29
6.1 Introduction 29
6.2 Identification 30
6.2.1 Resource identification and addressing 30
6.3 Namespace: 31
6.4 Network addressing 31
7 Resource model 31
7.1 Introduction 31
7.2 Resource 32
7.3 Property 33
7.3.1 Introduction 33
7.3.2 Common Properties 34
7.4 Resource Type 35
7.4.1 Introduction 35
7.4.2 Resource Type Property 36
7.4.3 Resource Type definition 36
7.4.4 Multi-value "rt" Resource 38
7.5 Device Type 38
7.6 Interface 39
7.6.1 Introduction 39
7.6.2 Interface Property 39
7.6.3 Interface methods 40
7.7 Resource representation 52
7.8 Structure 52
7.8.1 Introduction 52
7.8.2 Resource Relationships 52

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

7.8.3 Collections ... 57
7.9 Third (3rd) party specified extensions ... 60
8 CRUDN ... 61
8.1 Overview ... 61
8.2 CREATE ... 62
8.2.1 CREATE request ... 63
8.2.2 Processing by the Server ... 63
8.2.3 CREATE response ... 63
8.3 RETRIEVE ... 63
8.3.1 RETRIEVE request ... 64
8.3.2 Processing by the Server ... 64
8.3.3 RETRIEVE response ... 64
8.4 UPDATE ... 64
8.4.1 UPDATE request ... 65
8.4.2 Processing by the Server ... 65
8.4.3 UPDATE response ... 65
8.5 DELETE ... 65
8.5.1 DELETE request ... 66
8.5.2 Processing by the Server ... 66
8.5.3 DELETE response ... 66
8.6 NOTIFY ... 66
9 Network and connectivity ... 67
9.1 Introduction ... 67
9.2 Architecture ... 67
9.3 IPv6 network layer requirements ... 68
9.3.1 Introduction ... 68
9.3.2 IPv6 node requirements ... 69
10 Endpoint ... 69
10.1 Endpoint definition ... 69
10.2 Endpoint information ... 70
10.2.1 Introduction ... 70
10.2.2 “ep” ... 70
10.2.3 “pri” ... 70
10.2.4 Endpoint information in "eps" Parameter ... 71
10.3 Endpoint discovery ... 71
10.3.1 Introduction ... 71
10.3.2 Implicit discovery ... 71
10.3.3 Explicit discovery with “/oic/res” response ... 71
10.4 CoAP based Endpoint discovery ... 75
11 Functional interactions ... 76
11.1 Introduction ... 76
11.2 Onboarding, Provisioning and Configuration ... 76
11.3 Resource discovery ... 78
11.3.1 Introduction ... 78

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

11.3.2 Resource based discovery: mechanisms ... 78
11.3.3 Resource based discovery: Information publication process ... 80
11.3.4 Resource based discovery: Finding information ... 81
11.3.5 Resource discovery using “/oic/res” ... 87
11.3.6 Resource directory (RD) based discovery ... 89
11.4 Notification ... 103
11.4.1 Overview ... 103
11.4.2 Observe ... 103
11.5 Device management ... 105
11.5.1 Overview ... 105
11.5.2 Diagnostics and maintenance ... 105
11.6 Scenes ... 106
11.6.1 Introduction ... 106
11.6.2 Scenes ... 106
11.6.3 Security considerations ... 110
11.7 Icons ... 110
11.7.1 Overview ... 110
11.7.2 Resource ... 111
11.8 Introspection ... 111
11.8.1 Overview ... 111
11.8.2 Usage of introspection ... 113
12 Messaging ... 114
12.1 Introduction ... 114
12.2 Mapping of CRUDN to CoAP ... 115
12.2.1 Overview ... 115
12.2.2 URIs ... 115
12.2.3 CoAP method with request and response ... 115
12.2.4 Content-Format negotiation ... 117
12.2.5 OCF-Content-Format-Version information ... 118
12.2.6 Content-Format policy ... 118
12.2.7 CRUDN to CoAP response codes ... 119
12.2.8 CoAP block transfer ... 119
12.3 CoAP serialization over TCP ... 120
12.4 Payload Encoding in CBOR ... 121
13 Security ... 121
Annex A (informative) Operation Examples ... 123
A.1 Introduction ... 123
A.2 When at home: From smartphone turn on a single light ... 123
A.3 GroupAction execution ... 124
A.4 When garage door opens, turn on lights in hall; also notify smartphone ... 124
A.5 Device management ... 124
Annex B (informative) OCF interaction scenarios and deployment models ... 126
B.1 OCF interaction scenarios ... 126
B.2 Deployment model ... 127

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

Annex C (informative) Other Resource Models and OCF Mapping 129
C.1 Multiple resource models 129
C.2 OCF approach for support of multiple resource models 129
C.3 Resource model indication 130
C.4 An Example Profile (IPSO profile) 130
C.4.1 Conceptual equivalence 130
Annex D (normative) Resource Type definitions 133
D.1 List of Resource Type definitions 133
D.2 OCF Collection 134
D.2.1 Introduction 134
D.2.2 Example URI 134
D.2.3 Resource Type 134
D.2.4 RAML Definition 134
D.2.5 Property Definition 139
D.2.6 CRUDN behavior 140
D.2.7 Referenced JSON schemas 140
D.2.8 oic.oic-link-schema.json 140
D.3 Device Configuration 142
D.3.1 Introduction 142
D.3.2 Example URI 142
D.3.3 Resource Type 142
D.3.4 RAML Definition 142
D.3.5 Property Definition 147
D.3.6 CRUDN behavior 147
D.4 Platform Configuration 147
D.4.1 Introduction 147
D.4.2 Example URI 147
D.4.3 Resource Type 147
D.4.4 RAML Definition 147
D.4.5 Property Definition 150
D.4.6 CRUDN behavior 150
D.5 Device 150
D.5.1 Introduction 150
D.5.2 Wellknown URI 150
D.5.3 Resource Type 150
D.5.4 RAML Definition 151
D.5.5 Property Definition 153
D.5.6 CRUDN behavior 153
D.6 Maintenance 154
D.6.1 Introduction 154
D.6.2 Wellknown URI 154
D.6.3 Resource Type 154
D.6.4 RAML Definition 154
D.6.5 Property Definition 156

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

D.6.6 CRUDN behavior ... 156
D.7 Platform ... 157
D.7.1 Introduction ... 157
D.7.2 Wellknown URI ... 157
D.7.3 Resource Type ... 157
D.7.4 RAML Definition ... 157
D.7.5 Property Definition ... 159
D.7.6 CRUDN behavior ... 160
D.8 Ping ... 160
D.8.1 Introduction ... 160
D.8.2 Wellknown URI ... 160
D.8.3 Resource Type ... 160
D.8.4 RAML Definition ... 160
D.8.5 Property Definition ... 162
D.8.6 CRUDN behavior ... 162
D.9 Discoverable Resources Baseline Interface ... 162
D.9.1 Introduction ... 162
D.9.2 Wellknown URI ... 162
D.9.3 Resource Type ... 162
D.9.4 RAML Definition ... 162
D.9.5 Property Definition ... 164
D.9.6 CRUDN behavior ... 165
D.10 Discoverable Resources Link List interface ... 165
D.10.1 Introduction ... 165
D.10.2 Wellknown URI ... 165
D.10.3 Resource Type ... 165
D.10.4 RAML Definition ... 165
D.10.5 Property Definition ... 166
D.10.6 CRUDN behavior ... 167
D.10.7 Referenced JSON schemas ... 168
D.10.8 oic.oic-link-schema.json ... 168
D.11 Scenes (Top level) ... 170
D.11.1 Introduction ... 170
D.11.2 Example URI ... 170
D.11.3 Resource Type ... 170
D.11.4 RAML Definition ... 170
D.11.5 Property Definition ... 172
D.11.6 CRUDN behavior ... 172
D.12 Scene Collections ... 172
D.12.1 Introduction ... 172
D.12.2 Example URI ... 173
D.12.3 Resource Type ... 173
D.12.4 RAML Definition ... 173
D.12.5 Property Definition ... 176

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

D.12.6 CRUDN behavior ... 177
D.13 Scene Member ... 177
D.13.1 Introduction ... 177
D.13.2 Example URI ... 177
D.13.3 Resource Type ... 177
D.13.4 RAML Definition ... 177
D.13.5 Property Definition ... 179
D.13.6 CRUDN behavior ... 179
D.14 Resource directory resource ... 179
D.14.1 Introduction ... 179
D.14.2 Wellknown URI ... 180
D.14.3 Resource Type ... 180
D.14.4 RAML Definition ... 180
D.14.5 Property Definition ... 185
D.14.6 CRUDN behavior ... 186
D.15 Icon ... 186
D.15.1 Introduction ... 186
D.15.2 Example URI ... 186
D.15.3 Resource Type ... 186
D.15.4 RAML Definition ... 186
D.15.5 Property Definition ... 187
D.15.6 CRUDN behavior ... 187
D.16 Introspection Resource ... 188
D.16.1 Introduction ... 188
D.16.2 Example URI ... 188
D.16.3 Resource Type ... 188
D.16.4 RAML Definition ... 188
D.16.5 Property Definition ... 189
D.16.6 CRUDN behavior ... 190
Annex E (informative) Swagger2.0 definitions ... 191
E.1 Icon ... 191
E.1.1 Introduction ... 191
E.1.2 Example URI ... 191
E.1.3 Resource Type ... 191
E.1.4 Swagger2.0 Definition ... 191
E.1.5 Property Definition ... 193
E.1.6 CRUDN behavior ... 193
E.2 Introspection Resource ... 194
E.2.1 Introduction ... 194
E.2.2 Example URI ... 194
E.2.3 Resource Type ... 194
E.2.4 Swagger2.0 Definition ... 194
E.2.5 Property Definition ... 196
E.2.6 CRUDN behavior ... 197

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

E.3 OCF Collection ... 197
E.3.1 Introduction ... 197
E.3.2 Example URI ... 197
E.3.3 Resource Type ... 197
E.3.4 Swagger2.0 Definition ... 197
E.3.5 Property Definition ... 210
E.3.6 CRUDN behavior ... 213
E.4 Platform Configuration ... 213
E.4.1 Introduction ... 213
E.4.2 Example URI ... 213
E.4.3 Resource Type ... 213
E.4.4 Swagger2.0 Definition ... 213
E.4.5 Property Definition ... 217
E.4.6 CRUDN behavior ... 217
E.5 Device Configuration ... 217
E.5.1 Introduction ... 217
E.5.2 Example URI ... 217
E.5.3 Resource Type ... 217
E.5.4 Swagger2.0 Definition ... 217
E.5.5 Property Definition ... 223
E.5.6 CRUDN behavior ... 224
E.6 Device ... 224
E.6.1 Introduction ... 224
E.6.2 Wellknown URI ... 224
E.6.3 Resource Type ... 224
E.6.4 Swagger2.0 Definition ... 224
E.6.5 Property Definition ... 227
E.6.6 CRUDN behavior ... 228
E.7 Maintenance ... 228
E.7.1 Introduction ... 228
E.7.2 Wellknown URI ... 228
E.7.3 Resource Type ... 229
E.7.4 Swagger2.0 Definition ... 229
E.7.5 Property Definition ... 231
E.7.6 CRUDN behavior ... 231
E.8 Platform ... 231
E.8.1 Introduction ... 231
E.8.2 Wellknown URI ... 231
E.8.3 Resource Type ... 231
E.8.4 Swagger2.0 Definition ... 232
E.8.5 Property Definition ... 235
E.8.6 CRUDN behavior ... 236
E.9 Ping ... 236
E.9.1 Introduction ... 236

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

E.9.2 Wellknown URI 236
E.9.3 Resource Type 236
E.9.4 Swagger2.0 Definition 236
E.9.5 Property Definition 238
E.9.6 CRUDN behavior 239
E.10 Resource directory resource 239
E.10.1 Introduction 239
E.10.2 Wellknown URI 239
E.10.3 Resource Type 239
E.10.4 Swagger2.0 Definition 239
E.10.5 Property Definition 248
E.10.6 CRUDN behavior 249
E.11 Discoverable Resources 249
E.11.1 Introduction 249
E.11.2 Wellknown URI 249
E.11.3 Resource Type 249
E.11.4 Swagger2.0 Definition 249
E.11.5 Property Definition 256
E.11.6 CRUDN behavior 258
E.12 Scenes 258
E.12.1 Introduction 258
E.12.2 Example URI 258
E.12.3 Resource Type 258
E.12.4 Swagger2.0 Definition 258
E.12.5 Property Definition 272
E.12.6 CRUDN behavior 275

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

Figures

Figure 1: Architecture - concepts 25
Figure 2: Functional block diagram 26
Figure 3: Communication layering model 27
Figure 4: Example illustrating the Roles 28
Figure 5: Framework - Architecture Detail 28
Figure 6: Server bridging to Non- OCF device 29
Figure 7: Example of a Resource 33
Figure 8: Example - "Heater" Resource (for illustration only) 50
Figure 9: Example - Actuator Interface 50
Figure 10: Example of a Link 52
Figure 11: Example of distinct Links 52
Figure 12: Example of use of anchor in Link 53
Figure 13: Example of “eps Parameter 56
Figure 14: List of Links in a Resource 57
Figure 15: Example showing Collection and Links 59
Figure 16. CREATE operation 63
Figure 17. RETRIEVE operation 64
Figure 18. UPDATE operation 65
Figure 19. DELETE operation 66
Figure 20. High Level Network & Connectivity Architecture 68
Figure 21: Example of "ep" 70
Figure 22: Example of Link with "eps" Parameter 71
Figure 23: Example of “/oic/res” with Endpoint information 75
Figure 24. Resource based discovery: Information publication process 81
Figure 25. Resource based discovery: Finding information 81
Figure 26. Indirect discovery of resource by resource directory 90
Figure 27. RD discovery and RD supported query of resources support 92
Figure 28. Resource Direction Deployment Scenarios 93
Figure 29. Example of POST request payload 97
Figure 30. Example of POST response payload 98
Figure 31. Example of DELETE request with "di" or "ins" query 99
Figure 32. Observe Mechanism 104
Figure 33 Generic scene resource structure 107
Figure 34 Interactions to check Scene support and setup of specific scenes 108
Figure 35 Client interactions on a specific scene 109
Figure 36 Interaction overview due to a Scene change 110

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

Figure 37 Interactions to check Introspection support and download the Introspection Device Data. 114
Figure 38 Content-Format Policy 119
Figure 39. When at home: from smartphone turn on a single light 124
Figure 40. Device management (maintenance) 125
Figure 41. Direct interaction between Server and Client 126
Figure 42. Interaction between Client and Server using another Server 126
Figure 43. Interaction between Client and Server using Intermediary 126
Figure 44. Interaction between Client and Server using support from multiple Servers and Intermediary 127
Figure 45. Example of Devices 127

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

Tables

Table 1. Additional OCF Types 22
Table 2. Name Property Definition 35
Table 3. Resource Identity Property Definition 35
Table 4. Resource Type Common Property definition 36
Table 5. Example foobar Resource Type 37
Table 6. Example foobar properties 37
Table 7. Resource Interface Property definition 39
Table 8. OCF standard Interfaces 40
Table 9. Common Properties for Collections (in addition to Common Properties defined in section 7.3.2) 60
Table 10. 3rd party defined Resource elements 61
Table 11. Parameters of CRUDN messages 62
Table 12. “ep” value for Transport Protocol Suite 70
Table 13. List of Core Resources 76
Table 14. Configuration Resource 76
Table 15. “oic.wk.con” Resource Type definition 77
Table 16. “oic.wk.con.p” Resource Type definition 78
Table 17. Mandatory discovery Core Resources 82
Table 18. “oic.wk.res” Resource Type definition 83
Table 19. Protocol scheme registry 83
Table 20. “oic.wk.d” Resource Type definition 84
Table 21. “oic.wk.p” Resource Type definition 86
Table 22. “oic.wk.rd” Resource Type definition 90
Table 23. “oic.wk.rd” Properties 91
Table 24: Selection parameters 94
Table 25. Optional diagnostics and maintenance device management Core Resources 105
Table 26. “oic.wk.mnt” Resource Type definition 106
Table 27 list of Resource Types for Scenes 110
Table 28. Optional Icon Core Resource 111
Table 29. “oic.r.icon” Resource Type definition 111
Table 30. Introspection Resource 113
Table 31. “oic.wk.introspection” Resource Type definition 113
Table 32. CoAP request and response 115
Table 33. OCF Content-Formats 117
Table 34. OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Option Numbers 118
Table 35. OCF-Accept-Content-Format-Version and the OCF-Content-Format-Version Representation 118

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

Table 36. Examples of OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Representation 118
Table 37. Ping resource 120
Table 38. “oic.wk.ping” Resource Type definition 121
Table 39. oic.example.light Resource Type definition 123
Table 40. oic.example.garagedoor Resource Type definition 123
Table 41. Light control Resource Type definition 131
Table 42. Light control Resource Type definition 131
Table 43. Alphabetized list of core resources 133

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

1 Scope

The OCF specifications are divided into two sets of documents:

- Core Specification documents: The Core Specification documents specify the Framework, i.e., the OCF core architecture, interfaces, protocols and services to enable OCF profiles implementation for Internet of Things (IoT) usages and ecosystems.
- Vertical Profiles Specification documents: The Vertical Profiles Specification documents specify the OCF profiles to enable IoT usages for different market segments such as smart home, industrial, healthcare, and automotive. The Application Profiles Specification is built upon the interfaces and network security of the OCF core architecture defined in the Core Specification.

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

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

IEEE 754, *IEEE Standard for Floating-Point Arithmetic*, August 2008

IETF RFC 1981, *Path MTU Discovery for IP version 6*, August 1996
https://tools.ietf.org/rfc/rfc1981.txt

IETF RFC 2460, *Internet Protocol, version 6 (IPv6), December, 1998*
https://tools.ietf.org/rfc/rfc2460.txt

IETF RFC 2616, *Hypertext Transfer Protocol – HTTP/1.1*, June 1999.
http://www.ietf.org/rfc/rfc2616.txt

IETF RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*, June 2004
http://www.ietf.org/rfc/rfc3810.txt

IETF RFC 3986, *Uniform Resource Identifier (URI): General Syntax, January 2005*.
http://www.ietf.org/rfc/rfc3986.txt

IETF RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*, July 2005
http://www.ietf.org/rfc/rfc4122.txt

IETF RFC 4287, *The Atom Syndication Format*, December 2005,
http://www.ietf.org/rfc/rfc4287.txt

IETF RFC 4193, *Unique Local IPv6 Unicast Addresses*, October 2005
http://www.ietf.org/rfc/rfc4193.txt

IETF RFC 4291, IP Version 6 Addressing Architecture, February 2006
http://www.ietf.org/rfc/rfc4291.txt

IETF RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, March 2006
http://www.ietf.org/rfc/rfc4443.txt

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

IETF RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*, September 2007
http://www.ietf.org/rfc/rfc4861.txt

IETF RFC 4862, *IPv6 Stateless Address Autoconfiguration*, September 2007
http://www.ietf.org/rfc/rfc4862.txt

IETF RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*, September 2007
http://www.ietf.org/rfc/rfc4941.txt

IETF RFC 4944, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, September 2007
http://www.ietf.org/rfc/rfc4944.txt

IETF RFC 5646, *Tags for Identifying Languages*, September 2009
http://www.ietf.org/rfc/rfc5646.txt

IETF RFC 5988, *Web Linking: General Syntax, October 2010*
http://www.ietf.org/rfc/rfc5988.txt

IETF RFC 6434, *IPv6 Node Requirements*, December 2011
http://www.ietf.org/rfc/rfc6434.txt

IETF RFC 6455, *The WebSocket Protocol, December 2011*
https://www.ietf.org/rfc/rfc6455.txt

IETF RFC 6573, *The Item and Collection Link Relations*, April 2012
http://www.ietf.org/rfc/rfc6573.txt

IETF RFC 6690, *Constrained RESTful Environments (CoRE) Link Format*, August 2012
http://www.ietf.org/rfc/rfc6690.txt

IETF RFC 6762, *Multicast DNS* February 2013
http://www.ietf.org/rfc/rfc6762.txt

IETF RFC 6763, *DNS-Based Service Discovery*, February 2013
http://www.ietf.org/rfc/rfc6763.txt

IETF RFC 6775, *Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)*, November 2012
http://www.ietf.org/rfc/rfc6775.txt

IETF RFC 7049, *Concise Binary Object Representation (CBOR)*, October 2013
http://www.ietf.org/rfc/rfc7049.txt

IETF RFC 7084, *Basic Requirements for IPv6 Customer Edge Routers*, November 2013
http://www.ietf.org/rfc/rfc7084.txt

IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014
http://tools.ietf.org/rfc/rfc7159.txt

IETF RFC 7252, *The Constrained Application Protocol (CoAP)*, June 2014
http://tools.ietf.org/rfc/rfc7252.txt

IETF RFC 7301, *Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension*, July 2014
https://tools.ietf.org/html/rfc7301

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

IETF RFC 7428, *Transmission of IPv6 Packets over ITU-T G.9959 Networks*, February 2015
http://www.ietf.org/rfc/rfc7428.txt

IETF RFC 7641, *Observing Resources in the Constrained Application Protocol (CoAP)*, September 2015
https://tools.ietf.org/html/rfc7641

IETF RFC 7668, *IPv6 over BLUETOOTH(r) Low Energy, October 2015*
https://tools.ietf.org/html/rfc7668

IETF RFC 7721, *Security and Privacy Considerations for IPv6 Address Generation Mechanisms*, March 20016
https://tools.ietf.org/html/rfc7721

IETF RFC 7959, *Block-Wise Transfers in the Constrained Application Protocol (CoAP)*, August 2016
https://tools.ietf.org/html/rfc7959

IETF draft-ietf-core-coap-tcp-tls-07, *CoAP over TCP, TLS, and WebSockets*, June 10 2015
https://datatracker.ietf.org/doc/draft-ietf-core-coap-tcp-tls/

ECMA-4-4, *The JSON Data Interchange Format*, October 2013.
http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

OCF Security, *Open Connectivity Foundation Security Capabilities*, Version 1.0,

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

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

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

OCF Resource Type Definitions, *API Definition Language for OCF Resource Type Definitions,* Release OCF-v1.0.0
https://github.com/openconnectivityfoundation/core

W3C XML character escaping, *Extensible Markup Language (XML) 1.0,* November 2008
http://www.w3.org/TR/2008/REC-xml-20081126/#syntax

3 Terms, definitions, symbols and abbreviations

3.1 Terms and definitions

3.1.1
Client
a logical entity that accesses a Resource on a Server

3.1.2
Collection
a Resource that contains zero or more Links

3.1.3
Configuration Source

a cloud or service network or a local read-only file which contains and provides configuration related information to the Devices

3.1.4
Core Resources
those Resources that are defined in this specification

3.1.5
Default Interface
an Interface used to generate the response when an Interface is omitted in a request

3.1.6
Device
a logical entity that assumes one or more Roles (e.g., Client, Server)

Note 1 to entry: More than one Device can exist on a physical platform.

3.1.7
Device Type
a uniquely named definition indicating a minimum set of Resource Types that a Device supports

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

3.1.8
Endpoint
the source or destination of a request and response messages for a given Transport Protocol Suite

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

3.1.9
Entity
an aspect of the physical world that is exposed through a Device

Note 1 to entry: Example of an entity is an LED.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

3.1.10
Framework
a set of related functionalities and interactions defined in this specification, which enable interoperability across a wide range of networked devices, including IoT

3.1.11
Interface
provides a view and permissible responses on a Resource

3.1.12
Introspection
mechanism to determine the capabilities of the hosted Resources of a Device

3.1.13
Introspection Device Data
data that describes the payloads per implemented method of the Resources that makes up the Device

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

3.1.14
Links
extends typed web links according to IETF RFC 5988

3.1.15
Non-OCF Device
A device which does not comply with the OCF Device requirements

3.1.16
Notification
the mechanism to make a Client aware of resource state changes in a Resource

3.1.17
Observe
the act of monitoring a Resource by sending a RETRIEVE request which is cached by the Server hosting the Resource and reprocessed on every change to that Resource

3.1.18
Parameter
an element that provides metadata about a Resource referenced by the target URI of a Link

3.1.19
Partial UPDATE
an UPDATE request to a Resource that includes a subset of the Properties that are visible via the Interface being applied for the Resource Type

3.1.20
Platform
a physical device containing one or more Devices

3.1.21
Remote Access Endpoint (RAE) Client
a Client which supports XMPP functionality in order to access a Server from a remote location

3.1.22
Remote Access Endpoint (RAE) Server
a Server which supports XMPP and can publish its resource(s) to an XMPP server in the cloud, thus becoming remotely addressable and accessible

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

Note 1 to entry: An RAE Server also supports ICE/STUN/TURN.

3.1.23
Resource
represents an Entity modelled and exposed by the Framework

3.1.24
Resource Directory
a set of descriptions of Resources where the actual Resources are held on Servers external to the Device hosting the Resource Directory, allowing lookups to be performed for those resources

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

3.1.25
Resource Interface
a qualification of the permitted requests on a Resource

3.1.26
Resource Property
a significant aspect or parameter of a resource, including metadata, that is exposed through the Resource

3.1.27
Resource Type
a uniquely named definition of a class of Resource Properties and the interactions that are supported by that class

Note 1 to entry: Each Resource has a Property “rt” whose value is the unique name of the Resource Type.

3.1.28
Scene
a static entity that stores a set of defined Resource property values for a collection of Resources

Note 1 to entry: A Scene is a prescribed setting of a set of resources with each having a predetermined value for the property that has to change.

3.1.29
Scene Collection
a collection Resource that contains an enumeration of possible Scene Values and the current Scene Value

Note 1 to entry: The member values of the Scene collection Resource are Scene Members.

3.1.30
Scene Member
a Resource that contains mappings of Scene Values to values of a property in the resource

3.1.31
Scene Value
a Scene enumerator representing the state in which a Resource can be

3.1.32
Server
a Device with the role of providing resource state information and facilitating remote interaction with its resources

Note 1 to entry: A Server can be implemented to expose non-OCF Device resources to Clients (section 5.6)

3.1.33
Vertical Resource Type
a Resource Type in a vertical domain specification

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

Note 1 to entry: An example of a Vertical Resource Type would be “oic.r.switch.binary”.

3.2 Symbols and abbreviations

3.2.1
ACL
Access Control List

Note 1 to entry: The details are defined in OCF Security.

3.2.2
CBOR
Concise Binary Object Representation

3.2.3
CoAP
Constrained Application Protocol

3.2.4
EXI
Efficient XML Interchange

3.2.5
IRI
Internationalized Resource Identifiers

3.2.6
ISP
Internet Service Provider

3.2.7
JSON
JavaScript Object Notation

3.2.8
mDNS
Multicast Domain Name Service

3.2.9
MTU
Maximum Transmission Unit

3.2.10
NAT
Network Address Translation

3.2.11
OCF
Open Connectivity Foundation

the organization that created this specification

3.2.12
RAML
RESTful API Modeling Language

3.2.13
REST
Representational State Transfer

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

3.2.14
RESTfull
REST-compliant Web services

3.2.15
URI
Uniform Resource Identifier

3.2.16
URN
Uniform Resource Name

3.2.17
UTC
Coordinated Universal Time

3.2.18
UUID
Universal Unique Identifier

3.2.19
XML
Extensible Markup Language

3.3 Conventions

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

3.4 Data types

Resources are defined using data types derived from JSON values as defined in ECMA-4-4. However, a Resource can overload a JSON defined value to specify a particular subset of the JSON value, using validation keywords defined in [JSON Schema Validation].

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

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

Table 1. Additional OCF Types

Pattern Name	Pattern	Description
csv	<none>	A comma separated list of values encoded within a string. The value type in the csv is described by the property where the csv is used. For example a csv of integers.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

		Note: csv is considered deprecated and an array of strings should be used instead for new Resources.
date	^([0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|2[0-9]|1[0-9]|0[1-9])$	As defined in ISO 8601. The format is [yyyy]-[mm]-[dd].
int64	^0|(-?[1-9][0-9]{0,18})$	A string instance is valid against this attribute if it contains an integer in the range [-(2**63), (2**63)-1] Note: IETF RFC 7159 section 6 explains that JSON integers outside the range [-(2**53)+1, (2**53)-1] are not interoperable and so JSON numbers cannot be used for 64-bit numbers.
language-tag	^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$	An IETF language tag formatted according to IETF RFC 5646 section 2.1.
uint64	^0|([1-9][0-9]{0,19})$	A string instance is valid against this attribute if it contains an integer in the range [0, (2**64)-1] Also see note for int64
uuid	^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$	A UUID string representation formatted according to IETF RFC 4122 section 3.

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

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

4 Document conventions and organization

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

Required (or shall or mandatory)(M).

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

Recommended (or should)(S).

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

Allowed (may or allowed)(O).

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

DEPRECATED.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

Conditionally allowed (CA)

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

Conditionally required (CR)

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

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

Words that are emphasized are printed in italic.

5 Architecture

5.1 Overview

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

Specifically, the architecture provides:

- A communication and interoperability framework for multiple market segments (Consumer, Enterprise, Industrial, Automotive, Health, etc.), OSs, platforms, modes of communication, transports and use cases
- A common and consistent model for describing the environment and enabling information and semantic interoperability
- Common communication protocols for discovery and connectivity
- Common security and identification mechanisms
- Opportunity for innovation and product differentiation
- A scalable solution addressing different device capabilities, applicable to smart devices as well as the smallest connected things and wearable devices

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

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

5.2 Principle

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

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

Figure 1 depicts the architecture.

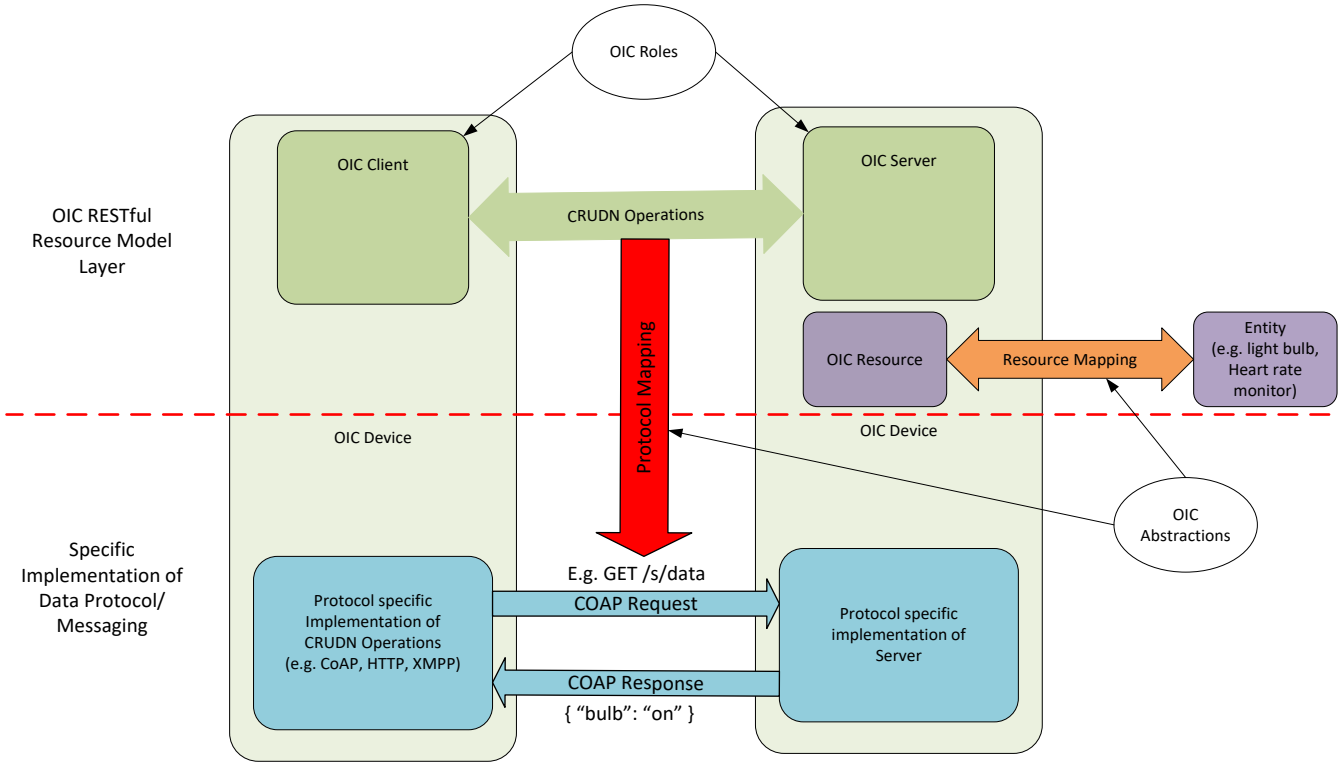

Figure 1: Architecture - concepts

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

- Resource model: The resource model provides the abstractions and concepts required to logically model, and logically operate on the application and its environment. The core resource model is common and agnostic to any specific application domain such as smart home,

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

industrial or automotive. For example, the resource model defines a Resource which abstracts an Entity and the representation of a Resource maps the Entity's state. Other resource model concepts can be used to model other aspects, for example behaviour.

- RESTful operations: The generic CRUDN operations are defined using the RESTful paradigm to model the interactions with a Resource in a protocol and technology agnostic way. The specific communication or messaging protocols are part of the protocol abstraction and mapping of Resources to specific protocols is provided in section 11.8.
- Abstraction: The abstractions in the resource model and the RESTful operations are mapped to concrete elements using abstraction primitives. An entity handler is used to map an Entity to a Resource and connectivity abstraction primitives are used to map logical RESTful operations to data connectivity protocols or technologies. Entity handlers may also be used to map Resources to Entities that are reached over protocols that are not natively supported by OCF.

5.3 Functional block diagram

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

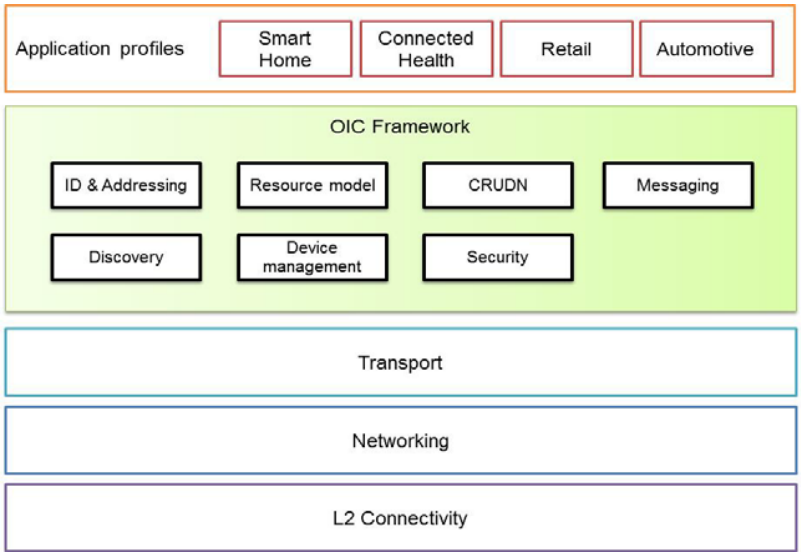

Figure 2: Functional block diagram

- **L2 connectivity:** Provides the functionalities required for establishing physical and data link layer connections (e.g., Wi-Fi™ or Bluetooth® connection) to the network.
- **Networking**: Provides functionalities required for Devices to exchange data among themselves over the network (e.g., Internet).
- **Transport**: Provides end-to-end flow transport with specific QoS constraints. Examples of a transport protocol include TCP and UDP or new Transport protocols under development in the IETF, e.g., Delay Tolerant Networking (DTN).
- **Framework**: Provides the core functionalities as defined in this specification. The functional block is the source of requests and responses that are the content of the communication between two Devices.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 30118-1:2018

- **Application profile**: Provides market segment specific data model and functionalities, e.g., smart home data model and functions for the smart home market segment.

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

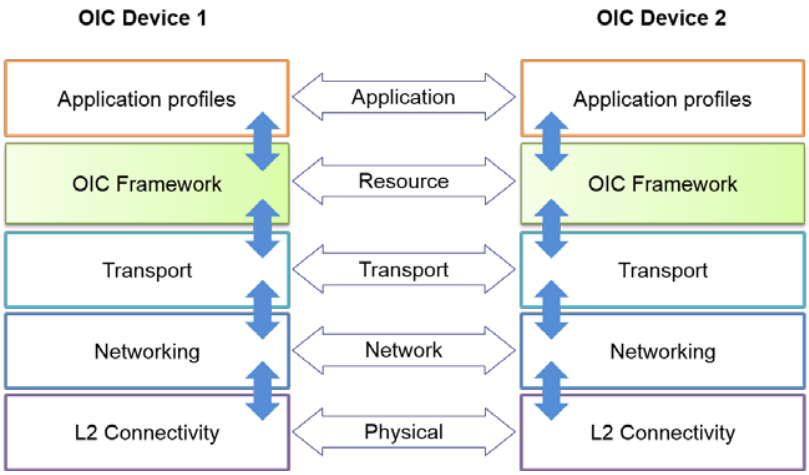

Figure 3: Communication layering model

5.4 Framework

Framework consists of functions which provide core functionalities for operation.

1) **Identification and addressing.** Defines the identifier and addressing capability. The Identification and addressing function is defined in section 6.
2) **Discovery**. Defines the process for discovering available
   a) Devices (Endpoint Discovery in section 10) and
   b) Resources (Resource discovery in section 11.3)
3) **Resource model**. Specifies the capability for representation of Entities in terms of resources and defines mechanisms for manipulating the resources. The resource model function is defined in section 7.
4) **CRUDN**. Provides a generic scheme for the interactions between a Client and Server as defined in section 8.
5) **Messaging**. Provides specific message protocols for RESTful operation, i.e. CRUDN. For example, CoAP is a primary messaging protocol. The messaging function is defined in section 11.8.
6) **Device management.** Specifies the discipline of managing the capabilities of a Device, and includes device provisioning and initial setup as well as device monitoring and diagnostics. The device management function is defined in section 11.5.
7) **Security.** Includes authentication, authorization, and access control mechanisms required for secure access to Entities. The security function is defined in section 13.

5.5 Example Scenario with roles

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

Figure 4 illustrates an example of the Roles in a scenario where a smart phone sends a request message to a thermostat; the original request is sent over HTTP, but is translated into a CoAP

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

request message by a gateway in between, and then delivered to the thermostat. In this example, the smart phone takes the role of a Client, the gateway takes the role of an Intermediary and the thermostat takes the role of a Server.

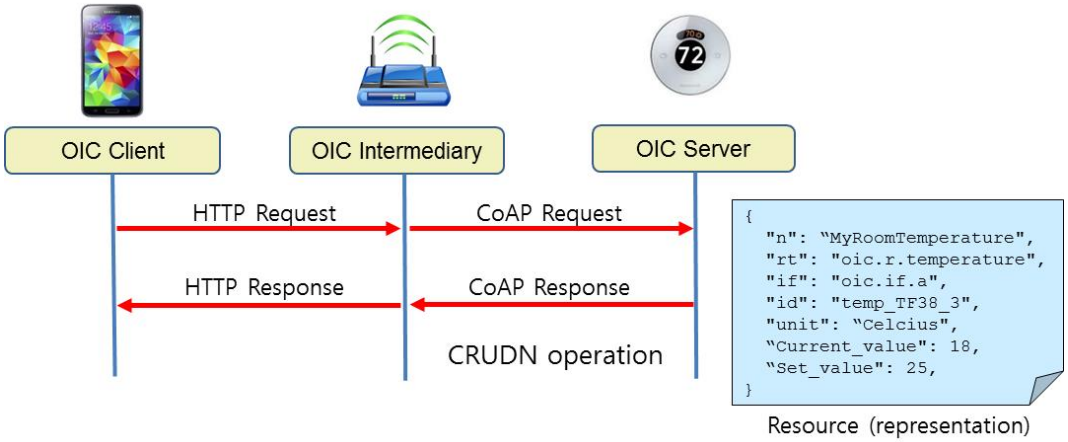

Figure 4: Example illustrating the Roles

5.6 Example Scenario: Bridging to Non- OCF ecosystem

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

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

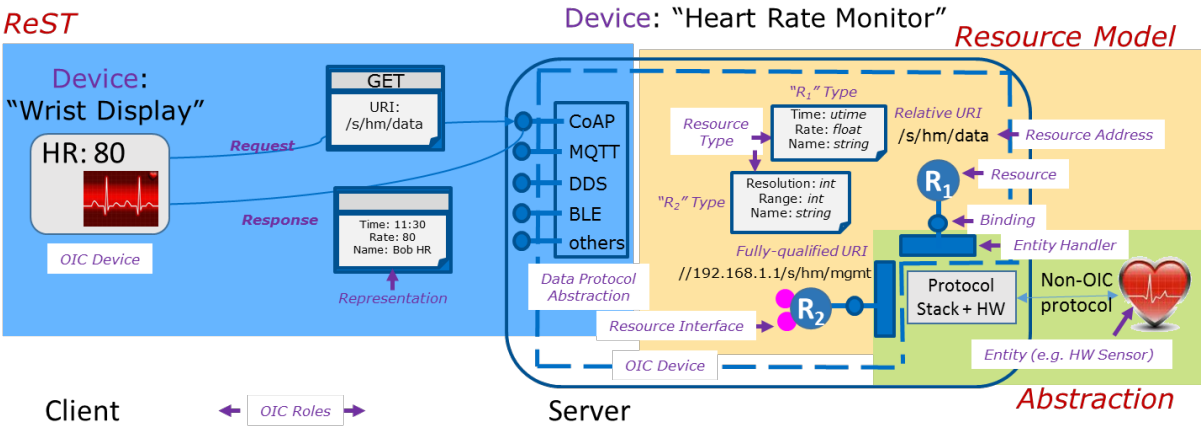

Figure 5: Framework - Architecture Detail

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

© ISO/IEC 2018 – All rights reserved

Documents Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

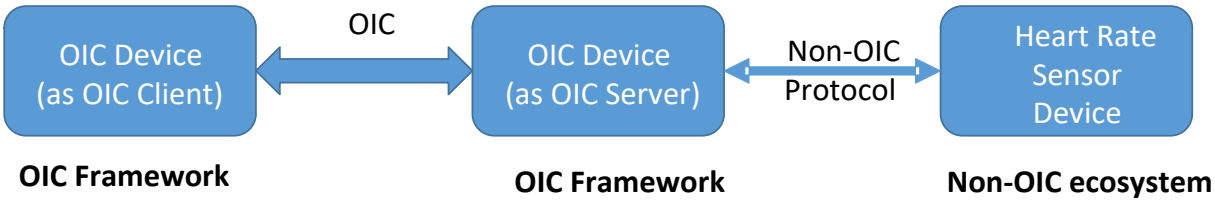

Figure 6: Server bridging to Non- OCF device

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

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

6 Identification and addressing

6.1 Introduction

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

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

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

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

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

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

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

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

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

6.2 Identification

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

6.2.1 Resource identification and addressing

A resource may be identified using a URI and addressed by the same URI if the URI is a URL. In some cases a resource may need an identifier that is different from a URI; in this case, the resource may have a property whose value is the identifier. When the URI is in the form of a URL, then the URI may be used to address the resource.

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

<**scheme>**://<a**uthority**>/<p**ath**>?<q**uery**>

Specifically the OCF URI is specified in the following form:

ocf://<a**uthority**>/<p**ath**>?<q**uery**>

A description of values that each component takes is given below.

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

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

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

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

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

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

A URI may be either

- Fully qualified or
- Relative

Generation of URI:

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

Use of URI:

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

6.3 Namespace:

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

6.4 Network addressing

The following are the addresses used in this specification:

- **IP address**

An IP address is used when the device is using an IP configured interface.

When a Device only has the identity information of its peer, a resolution mechanism is needed to map the identifier to the corresponding address.

7 Resource model

7.1 Introduction

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

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

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

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

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

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

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

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

The information for Request or Response with the Representation may be communicated “on the wire” by serializing using a transfer protocol or encapsulated in the payload of the transport protocol – the specific method is determined by the normative mapping of the Request or Response to the transport protocol. See section 11.8 for transport protocols supported.

The RAML definitions used in this document are normative. This also includes that all defined JSON payloads shall comply with the indicated JSON schema. See Annex D for Resource Types defined in this specification.

7.2 Resource

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

A Resource is hosted in a Device. A Resource shall have a URI as defined in section 6. The URI may be assigned by the Authority at the creation of the Resource or may be pre-defined by the

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

specification of the Resource Type.

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

URI

Properties

Figure 7: Example of a Resource

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

7.3 Property

7.3.1 Introduction

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

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

In addition, the Property definition shall have a

- **Value Type** – the Value Type defines the values that a Property Value may take. The Value Type may be a simple data type (e.g. string, Boolean) as defined in section 3.4 or may be a complex data type defined with a schema. The Value Type may define
    - Value Rules define the rules for the set of values that the Property Value may take. Such rules may define the range of values, the min-max, formulas, set of enumerated values, patterns, conditional values and even dependencies on values of other Properties. The rules may be used to validate the specific values in a Property Value and flag errors.
- **Mandatory** – specifies if the Property is mandatory or not for a given Resource Type.
- **Access modes** – specifies whether the Property may be read, written or both. Updates are equivalent to a write. “r” is used for read and “w” is used for write – both may be specified. Write does not automatically imply read.

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

- **Property Title** - a human-friendly name to designate the Property; usually not sent over the wire
- **Description** – descriptive text defining the purpose and expected use of this Property.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

A Property may be used in the query part of an URI as one criterion for selection of a particular Resource. This is done by declaring the Property (i.e. <Property Name> = <desired Property Value>) as one of the segments of the query. In this version of the specification, only ASCII strings are permitted in query filters, and NUL characters are disallowed in query filters. This means that only property values with ASCII characters can be matched in a query filter. The Resource is selected when all the declared Properties in the query match the corresponding Properties in the full Representation of the target Resource. The full Representation is the snapshot that includes the union of all Properties in all Resource Types that define the target Resource. If the Property is declared in the “filter” segment of the query then the declared Property is matched to the Representation defined by the Interface to isolate certain parts of that Representation.

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

7.3.2 Common Properties

7.3.2.1 Introduction

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

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

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

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

- Resource Type ("rt") – this Property is used to declare the Resource Type of that Resource. Since a Resource could be define by more than one Resource Type the Property Value of the Resource Type Property can be used to declare more than one Resource type. For example: “rt”: [“oic.wk.d”, “oic.d.airconditioner”] declares that the Resource containing this Property is defined by either the “oic.wk.d” Resource Type or the “oic.d.airconditioner” Resource Type. See section 7.3.2.3 for details.
- Interface ("if") – this Property declares the Interfaces supported by the Resource. The Property Value of the Interface Property can be multi-valued and lists all the Interfaces supported. See section 7.3.2.4 for details.
- Name ("n") – the Property declares “human-readable” name assigned to the Resource. See section 7.3.2.5.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

7.3.2.2 Property Name and Property Value definitions

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

- **Property Name**– the key in "key=value" pair. Property Name is case sensitive and its data type is "string" but only ASCII characters are permitted, and embedded NUL characters are not permitted.
- **Property Value** – the value in "key=value" pair. Property Value is case sensitive when its data type is "string". Any enum values shall be ASCII only.

7.3.2.3 Resource Type

Resource Type Property is specified in section 7.4.

7.3.2.4 Interface

Interface Property is specified in Section 7.5.

7.3.2.5 Name

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

Table 2. Name Property Definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Name	n	string			R, W	no	Human understandable name for the resource.

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

7.3.2.6 Resource Identity

The Resource Identity Property shall be a unique (across the scope of the host Server) instance identifier for a specific instance of the Resource. The encoding of this identifier is device and implementation dependent. The Resource Identity Property is as defined in Table 3.

Table 3. Resource Identity Property Definition

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

7.4 Resource Type

7.4.1 Introduction

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

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

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

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

7.4.2 Resource Type Property

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

Table 4. Resource Type Common Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource type	rt	array	Array of strings, conveying resource Type IDs		R	yes	The property name rt is as described in IETF RFC 6690

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

7.4.3 Resource Type definition

Resource Type is specified as follows:

- **Pre-defined URI** (optional) – a pre-defined URI may be specified for a specific Resource Type in an OCF specification. When a Resource Type has a pre-defined URI, all instances of that Resource Type shall use only the pre-defined URI. An instance of a different Resource Type shall not use the pre-defined URI.
- **Resource Type Title (optional)** – a human friendly name to designate the Resource Type**.**
- **Resource Type ID** – the value of "rt" property which identifies the Resource Type, (e.g., "oic.wk.p").
  - **Resource Interfaces** – list of the interfaces that may be supported by the Resource Type.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

- **Resource Properties** – definition of all the properties that apply to the Resource Type. The Resource Type definition shall define whether a property is mandatory, conditional mandatory, or optional.
- **Related Resource Types** (optional) – the specification of other Resource Types that may be referenced as part of the Resource Type, applicable to collections.
- **Mime Types** (optional) – mime types supported by the resource including serializations (e.g., application/cbor, application/json, application/xml).

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

Table 5. Example foobar Resource Type

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

Table 6. Example foobar properties

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Resource Type	**rt**	**array**			**R**	**yes**	**Resource Type**
Interface	**if**	**array**			**R**	**yes**	**Interface**
Foo value	**value**	**string**			**R**	**yes**	**Foo value**

An instance of the foobar Resource Type is as shown below

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

An example schema for the foobar Resource Type is shown below

```
{
  "$schema": "http://json-schema.org/draft-04/schema",
  "type": "object",
  "properties": {
    "rt": {"type": "string"},
    "if": {"type": "string"},
    "value": {"type": "string"}
  },
  "required": ["rt", "if", "value"]
}
```


© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

7.4.4 Multi-value "rt" Resource

Multi-value "rt" Resource means a Resource with multiple Resource Types. Such a Resource is associated with multiple Resource Types and so has an "rt" Property Value of multiple Resource Type IDs (e.g. "rt": ["oic.r.switch.binary", "oic.r.light.brightness"]). The order of the Resource Type IDs in the “rt” Property Value is meaningless. For example, "rt": ["oic.r.switch.binary", "oic.r.light.brightness"] and "rt": ["oic.r.light.brightness", "oic.r.switch.binary"] have the same meaning.

Resource Types for multi-value “rt” Resources shall satisfy the following conditions.

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

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

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

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

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

7.5 Device Type

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

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

A Device Type may either be pre-defined (in vertical domain specifications) or in custom definitions by manufacturers, end users, or developers of Devices (vendor-defined Device Types). Device Types and their definition details may be communicated out of band (like in documentation).

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

7.6 Interface

7.6.1 Introduction

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

An Interface may be defined by either this specification (a Core Interface), the OCF vertical domain specifications (a “vertical Interface) or manufacturers, end users or developers of Devices (a “vendor-defined Interface”).

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

In addition to any OCF specification defined interface, all Resources shall support the Baseline Interface (“oic.if.baseline”) as defined in section 7.6.3.2.

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

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

7.6.2 Interface Property

Table 7. Resource Interface Property definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Interface	if	array	Array of strings, conveying interfaces		R	yes	Property to declare the Interfaces supported by a Resource.

The Interfaces supported by a Resource shall be declared using the Interface Common Property (Table 7) as "`if=`<array of Interfaces>". The Property Value of an Interface Property shall be a lower case string with segments separated by a "." (dot). The string "`oic`", when used as the first segment in the Interface Property Value, is reserved for OCF-defined Interfaces. The Interface Property Value may also be a reference to an authority similar to IANA that may be used to find the definition of an Interface. A Resource Type shall support one or more of the Interfaces defined in section 7.6.3.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

7.6.3 Interface methods

7.6.3.1 Overview

The OCF -defined Interfaces are listed in the table below:

Table 8. OCF standard Interfaces

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

7.6.3.2 Baseline Interface

7.6.3.2.1 Overview

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

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

7.6.3.2.2 Use of RETRIEVE

The "baseline" Interface is used when a Client wants to retrieve all Properties of a Resource. The Client includes the URI query parameter definition "?if=oic.if.baseline" in a RETRIEVE request. When this query parameter definition is included the Server shall respond with a Resource representation that includes all of the implemented Properties of the Resource. When the Server is unable to send back the whole Resource representation, it shall reply with an error message. The Server shall not return a partial Resource representation.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

An example response to a RETRIEVE request using the baseline Interface is shown below:

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

7.6.3.2.3 Use of UPDATE

Using the baseline Interface, all Properties of a Resource with the exception of Common Properties may be modified using an UPDATE request with a list of Properties and their desired values if a Resource Type has an associated schema for UPDATE using baseline.

7.6.3.3 Link List Interface

7.6.3.3.1 Overview

The links list Interface provides a view into the list of Links in a Collection (Resource). The Representation visible through this Interface has only the Links defined in the Property Value of the “links” Property – so this Interface is used to manipulate or interact with the list of Links in a Collection. The Links list may be RETRIEVEd using this Interface.

The Interface definition and semantics are given as follows:

- The links list Interface name shall be “oic.if.ll”.
- If specified in a request (usually in the request header), the serialization in the response shall be in the format expected in the request.
- In response to a RETRIEVE request on the “links list” Interface, the URIs of the referenced Resources shall be returned as a URI reference.
- If there are no links present in a Resource, then an empty list shall be returned.
- The Representation determined by this Interface view only includes the Property Value of the “links” Property. Hence Collection or /oic/res response with oic.if.ll is an array of OCF Links.

7.6.3.3.2 Example: “links list” Interface

Example: Request to a Collection

Request to RETRIEVE the Links in room
 	
(the Links could be referencing lights, fans, electric sockets etc)
	GET ocf://<devID>/a/room/1?if=oic.if.ll
	
	The response would be the array of OCF Links
	
	[
	  {
	  "href": "/the/light/1",
	  "rt": ["oic.r.switch.binary"],
	  "if": ["oic.if.a", "oic.if.baseline"],

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

7.6.3.4 Batch Interface

7.6.3.4.1 Overview

The batch Interface is used to interact with a collection of Resources using a single/same Request. The batch Interface supports methods of Resources in the Links of the Collection, and can be used to RETRIEVE or UPDATE the Properties of the “linked” Resources with a single Resource representation.

The batch Interface selects a view into the Links in a Collection – the Request is sent to all the Links in this view with potential modifications defined in the Parameters of the Link

© ISO/IEC 2018 – All rights reserved

Documents Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

The batch Interface is defined as follows:

- The batch Interface name shall be “oic.if.b”
- A Collection Resource with a batch Interface has Links that have Resource references that may be URIs (fully qualified for remote Resources) or relative references (for local Resources).
- The original request is modified to create new requests targeting each of the targets in the Resource Links by substituting the URI in the original request with the URI of the target Resource in the Link. The payload in the original request is replicated in the payload of the new Requests.
- The Requests shall be forwarded assuming use of the Default Interface of the referenced Resources.
- Requests shall only be forwarded to link targets that are identified as items in the collection by relation types "item" or "hosts" ("hosts" is the default relation type). Requests shall not be forwarded to targets of links that do not contain the "item" or "hosts" relation type values. The default relation type "hosts" shall be allowed for relative and absolute links.
- Resources of the collection itself may be included in the batch by using the link relation "self" along with "item" and insuring that the default interface of the collection does not expose the links property, i.e. not “oic.if.baseline” or “oic.if.ll”.
- All the Responses from the linked item Resources shall be aggregated into single Response to the Client. The Server may timeout the Response to a time window (if a time window has been negotiated with the Client then the Server shall not timeout within that window; in the absence of negotiated window, the Server may choose any appropriate window based on conditions). If the target Resources cannot process the new request, an empty response or error response shall be returned. These empty/error Responses shall be included in aggregated Response to the original Client Request.
- The batch representation is an array of objects representing the responses from the linked resources. Each object in the batch response shall include at least two items: (1) the URI of the linked resource (fully qualified for remote resources, or a relative reference for local resources) as “href”: <URI> and (2) the individual response object as “rep” as the key i.e. “rep”: { < representation of individual response> }.
- Resources referenced by links in the collection may be observed using the batch interface of the collection. The observe mechanism shall work as defined in 11.4.2. Specifically, the representations and status codes shall be the same as for RETRIEVE operations using the Batch interface.
- Properties of the collection resource itself may be observed by using the appropriate interface. For example, a collection may be observed on its linked list or baseline interface to receive notifications of changes to its links.
- The Client may choose to restrict the list of Links to which the Request is forwarded by including query parameters in the URI of the Collection to which this original ‘batch’ Interface Request is made. The Server should process query parameters in a request that includes “oic.if.b”, as selectors for links in the Collection that are to be processed in the batch.
- Batch UPDATE operations are performed by creating a payload according to the same schema of the Batch RETRIEVE payload. A set of link-specific UPDATE requests is created according to the "href" tags in the included items, and the payload contained in the value of the "rep" property is applied to the corresponding "href" referenced item.
- If requested property for UPDATE does not exist in linked resource, it shall silently ignore the request.
- If the "href" value is the empty URI, denoted by a zero length string or "" in JSON, the payload in the value of the "rep" property is applied to all batch items in the Collection.

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

- Items with the empty "href" and link-specific "href" shall not be mixed in the same UPDATE payload.
- The Representation in the Link-specific Request may not match the Representation exposed by the Default Interface on the target Resource. In such cases, the 'subset' semantics apply where Properties in the Request which match Properties in the Resource view exposed shall be modified in the target Resource if the Property is writeable.
- The response to POST shall contain the updated values using the same payload schema as RETRIEVE operations, along with the appropriate status code. The response payload shall reflect the known state of the updated resource properties after the batch update was completed.

7.6.3.4.2 Use of Query Parameters with Batch

Additional query parameters may be used with the batch interface in order to select particular items in the batch for retrieval or update. When additional parameters are included which are not interpreted in other ways, these parameters are used to select items in the batch by matching link attribute values.

A particular item in a batch is selected by additional query parameters in a request if, and only if, all of the link selection query parameters contained values which match corresponding values in the link attributes of that item.

When link selection query parameters are used with RETRIEVE operations, only the items with matching link attributes are returned.

When link selection query parameters are used with UPDATE operations, only the items having matching link attributes are updated.

See 7.6.3.4.3 for examples of RETRIEVE and UPDATE operations that use link selection query parameters.

7.6.3.4.3 Examples: Batch Interface

Example 1

Resources	/a/room/1 { "rt": ["oic.wk.col","x.org.example.rt.room"], "if": ["oic.if.baseline","oic.if.b","oic.if.ll", "oic.if.r"], "x.org.example.color": "blue", "x.org.example.dimension": "15bx15wx10h", "links": [{"href": "/a/room/1", "rel": ["self", "item"], "rt": ["x.org.example.rt.room"], "if": ["oic.if.r", "oic.if.baseline","oic.if.b","oic.if.ll"] }, {"href": "/the/light/1", "rel": ["item"], "rt": ["oic.r.switch.binary"], "if": ["oic.if.a","oic.if.baseline"]},

© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018

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

     {"href": "/my/fan/1", "rel": ["item"], "rt":
["oic.r.switch.binary"], if=[","oic.if.a", "oic.if.baseline"]},

     {"href": "/his/fan/2", "rel": ["item"], "rt":
["oic.r.switch.binary"], if=["oic.if.a", "oic.if.baseline"]}

  ]

}


/the/light/1

{

  "rt": ["oic.r.switch.binary"],

  "ins": "light-1",

  "if": ["oic.if.a", "oic.if.baseline"],

  "value": false

}


/the/light/2

{

  "rt": ["oic.r.switch.binary"],

  "ins": :"light-2",

  "if": ["oic.if.a", "oic.if.baseline"],

  "value": true

}


/the/fan/1

{

  "rt": ["oic.r.switch.binary"],

  "ins": "fan-1",

  "if": ["oic.if.a", "oic.if.baseline"],

  "value": true
```


© ISO/IEC 2018 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 30118-1:2018