OCF Core Specification
OCF核心规范（双语版本）

VERSION 2.1.0 | April 2020

OPEN CONNECTIVITY FOUNDATION™

CONTACT admin@openconnectivity.org

Legal Disclaimer

免责声明

NOTHING CONTAINED IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU ANY KIND OF LICENSE IN ITS CONTENT, EITHETable-Annex A.1R EXPRESSLY OR IMPLIEDLY, OR TO ANY INTELLECTUAL PROPERTY OWNED OR CONTROLLED BY ANY OF THE AUTHORS OR DEVELOPERS OF THIS DOCUMENT. THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE AUTHORS AND DEVELOPERS OF THIS SPECIFICATION HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OPEN CONNECTIVITY FOUNDATION, INC. FURTHER DISCLAIMS ANY AND ALL WARRANTIES OF NON-INFRINGEMENT, ACCURACY OR LACK OF VIRUSES.

本文档中包含的任何内容、明示或默示的EITHETable-Annex A.1R或本文档的所有作者和开发者拥有或控制的任何知识产权均不应被视为授予您任何形式的许可。本文档中所含的任何内容是在适用法律允许的最大范围内按照“原文”提供的。本文档的所有作者和开发者在此声明不对无论是明示或默示、法规或普通法下的所有担保和条件承担任何责任，包括但不限于适销性的默示担保或对特定目标的适用性。OPEN CONNECTIVITY FOUNDATION, INC.进一步声明不对侵犯第三方权利、内容准确性和计算机病毒所产生的任何损失承担任何责任。

Translation Disclaimer

翻译免责声明

For Translation to Local Language

对于翻译成当地语言：

- This translation of the OCF specification from the original, normative, English language version is made available to encourage and assist with development of OCF-based products. While every effort has been made to ensure an accurate translation of the English language specification, this translation shall not be considered normative. The OCF Certification program is developed explicitly against the English language document and any claim for waivers or exemptions will only be evaluated against the text of that document.
- 本译文是从原始的、规范性的和英语版本的OCF规范翻译而来，旨在鼓励和帮助开发基于OCF的产品。虽然已尽一切努力确保对原文规范（英语）的准确翻译，但本译文不应视为规范性翻译。OCF认证计划是根据原文规范（英语）明确制定的，任何豁免或豁免申请将仅根据原文规范（英语）文本进行评估。

- There will be a delay between the publication of the latest English language version of specifications and the subsequent translations.
- 规范的最新英文版本与后续翻译版本间会存在延迟。

- For the latest English language and translated versions of OCF specifications please visit https://openconnectivity.org/developer/specifications
- 有关OCF规范的最新英语版本和翻译版本，请访问 https://openconnectivity.org/developer/specifications 。

关于本文档的说明 About this document

为了使中国国内OCF技术的使用者、爱好者能更好地理解IoTivity的技术细节，更加便捷将此技术应用在不同的产品上，更方便地构筑一个真正的互联互通的应用，OCFC组织了相关资源对OCF现行的技术规范进行了翻译。这是整个系列规范中最重要、亦是首先翻译完成的规范；相关的规范列表如下：

For these users and developers in China who use OCF technology to better understand the technical details of IoTivity, apply this type of technology to different products more conveniently, and build a truly interconnected application, OCFC organized related resources to translate the current technical specifications of OCF. Core specification is the most important in the entire series of specifications, and is the first translated specification; the list of related specifications are as follows:

OCF_Bridging_Specification

OCF_Cloud_Security_Specification

OCF_Core_Optional_Specification

OCF_Core_Specification（本文档，this document）

OCF_Device_Specification

OCF_Device_To_Cloud_Services_Specification

OCF_Onboarding_Tool_Specification

OCF_Resource_to_AllJoyn_Interface_Mapping

OCF_Resource_to_BLE_Mapping_Specification

OCF_Resource_to_EnOcean_Mapping_Specification

OCF_Resource_to_OneM2M_Module_Class_Mapping_Specification

OCF_Resource_to_UPlus_Mapping_Specification

OCF_Resource_to_Z-Wave_Mapping_Specification

OCF_Resource_to_ZigBee_Cluster_Mapping_Specification

OCF_Resource_Type_Specification

OCF_Security_Specification（翻译中，in translation progress）

OCF_Wi-Fi_Easy_Setup_Specification

本文档主要译者：

张明珠、孔睿迅、赵牧、黄伟彬、许蕴盈、李永华、茹昭、马龙飞、王海廷、宋子豪、Viktor Ariel

Main translator of this document：

Mingzhu Zhang, Miles Kong, Betty Zhao, Weibin Huang, Winnie Xu, Yonghua Li, Zhao Ru, Mark Ma, Haiting Wang, Answer Sung, Viktor Ariel

Contents 目录

1 Scope 范围

The OCF Core specifications are divided into a set of documents:
OCF核心规范可分为一系列文件：

- Core specification (this document): The Core specification document specifies 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. This document is mandatory for all Devices to implement.
  核心规范（本文件）：核心规范文档规定了框架，即OCF核心体系结构、接口、协议和服务，以使能针对物联网（IoT）用法和生态系统的OCF配置文件的实施。本文档（的要求）对于所有设备是强制要求实现的。
- Core optional specification: The Core optional specification document specifies 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 that can optionally be implemented by any Device.
  核心可选规范：核心可选规范文档规定了框架，即OCF核心架构、接口、协议和服务，以使能针对物联网(IoT)用法和生态系统的OCF配置文件的实现，这些实现对于设备而言是可选的。
- Core extension specification(s): The Core extension specification(s) document(s) specifies optional OCF Core functionality that are significant in scope (e.g., Wi-Fi easy setup, Cloud).
  核心扩展规范：核心扩展规范文档规定了适用范围内的、重要的可选OCF核心功能（例如，Wi-Fi简易设置、云）。

2 Normative references 引用标准

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.
下列全部或部分文件在本文件中均有规范的引用，对其应用是必不可少的。对于注明日期的参考文献，仅适用所引用的版本。对于未注明日期的引用，引用文件的最新版本（包括任何修订）适用。

ISO 8601, Data elements and interchange formats - Information interchange -Representation of dates and times, International Standards Organization, December 3, 2004
ISO 8601 数据元素和交换格式-信息交换 日期和时间的表示 国际标准组织 2004年12月3日

ISO/IEC DIS 20924, Information Technology - Internet of Things - Vocabulary, June 2018
https://www.iso.org/standard/69470.html
信息技术-物联网-词汇 2018年6月
https://www.iso.org/standard/69470.html

ISO/IEC 30118-2:2018, Information technology - Open Connectivity Foundation (OCF) Specification - Part 2: Security specification
https://www.iso.org/standard/74239.html
ISO/IEC 30118-2:2018 信息技术 开放互连基础（OCF）规范 第2部分：安全规范
https://www.iso.org/standard/74239.html

Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf
最新版本载于：https://openconnectivity.org/specs/OCF_Security_Specification.pdf

IETF RFC 768 User Datagram Protocol, August 1980
https://www.rfc-editor.org/info/rfc768
IETF RFC 768 （UDP）用户数据报协议 1980年8月

IETF RFC 3339, Date and Time on the Internet: Timestamps, July 2002
https://www.rfc-editor.org/info/rfc3339
IETF RFC 3339 互联网的日期和时间：时间戳记 2002年7月
https://www.rfc-editor.org/info/rfc3339

IETF RFC 3986, Uniform Resource Identifier (URI): General Syntax, January 2005
https://www.rfc-editor.org/info/rfc3986
IETF RFC 3986 统一资源标识符（URI）：通用语法 2005年1月
https://www.rfc-editor.org/info/rfc3986

IETF RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace, July 2005
https://www.rfc-editor.org/info/rfcfse4122
IETF RFC 4122 一个通用的唯一标识符（UUID） URN名称空间 2005年7月
https://www.rfc-editor.org/info/rfcfse4122

IETF RFC 4287, The Atom Syndication Format, December 2005
https://www.rfc-editor.org/info/rfc4287

IETF RFC 4287Atom联合格式，2005年12月
https://www.rfc-editor.org/info/rfc4287

IETF RFC 4941, Privacy Extensions for Stateless Address Autoconfiguration in IPv6, September 2007
https://www.rfc-editor.org/info/rfc4941
IETF RFC 4941 隐私扩展无状态地址自动配置在IPv6 2007年9月
https://www.rfc-editor.org/info/rfc4941

IETF RFC 5646, Tags for Identifying Languages, September 2009
https://www.rfc-editor.org/info/rfc5646
IETF RFC 5646 用于识别语言的标签 2009年9月
https://www.rfc-editor.org/info/rfc5646

IETF RFC 6347, Datagram Transport Layer Security Version 1.2, January 2012
https://www.rfc-editor.org/info/rfc6347
IETF RFC 6347 数据包传输层安全 版本1.2 2012年1月
https://www.rfc-editor.org/info/rfc6347

IETF RFC 6434, IPv6 Node Requirements, December 2011
https://www.rfc-editor.org/info/rfc6434
IETF RFC 6434 IPv6节点要求 2011年12月
https://www.rfc-editor.org/info/rfc6434

IETF RFC 6573, The Item and Collection Link Relations, April 2012
https://www.rfc-editor.org/info/rfc6573
IETF RFC 6573 项目和集合的链接关系
2012年4月 https://www.rfc-editor.org/info/rfc6573

IETF RFC 6690, Constrained RESTful Environments (CoRE) Link Format, August 2012
https://www.rfc-editor.org/info/rfc6690
IETF RFC 6690，受限的RESTful环境(CoRE)链接格式，2012年8月
https://www.rfc-editor.org/info/rfc6690

IETF RFC 7049, Concise Binary Object Representation (CBOR), October 2013
https://www.rfc-editor.org/info/rfc7049
IETF RFC 7049，简明的二进制对象表示(CBOR)，2013年10月
https://www.rfc-editor.org/info/rfc7049

IETF RFC 7084, Basic Requirements for IPv6 Customer Edge Routers, November 2013
https://www.rfc-editor.org/info/rfc7084
IETF RFC 7084,IPv6客户边缘路由器的基本要求，2013年11月
https://www.rfc-editor.org/info/rfc7084

IETF RFC 7159, The JavaScript Object Notation (JSON) Data Interchange Format, March 2014
https://www.rfc-editor.org/info/rfc7159

IETF RFC 7159，JavaScript对象表示法(JSON)数据交换格式，2014年3月
https://www.rfc-editor.org/info/rfc7159

IETF RFC 7252, The Constrained Application Protocol (CoAP), June 2014
https://www.rfc-editor.org/info/rfc725

IETF RFC 7252，约束应用程序协议(CoAP)，2014年6月
https://www.rfc-editor.org/info/rfc725

IETF RFC 7301, Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension, July 2014
https://www.rfc-editor.org/info/rfc7301
IETF RFC 7301，传输层安全(TLS)应用层协议协商扩展，2014年7
https://www.rfc-editor.org/info/rfc7301

IETF RFC 7346, IPv6 Multicast Address Scopes, August 2014
https://www.rfc-editor.org/info/rfc7346
IETF RFC 7346，IPv6多播地址范围，2014年8月
https://www.rfc-editor.org/info/rfc7346

IETF RFC 7595, Guidelines and Registration Procedures for URI Schemes, June 2015
https://www.rfc-editor.org/info/rfc7595
IETF RFC 7595，URI计划的指引及登记程序，2015年6月
https://www.rfc-editor.org/info/rfc7595

IETF RFC 7641, Observing Resources in the Constrained Application Protocol (CoAP), September 2015
https://www.rfc-editor.org/info/rfc7641
IETF RFC 7641，在受限的应用程序协议中观察资源（CoAP），2015年9月。
https://www.rfc-editor.org/info/rfc7641

IETF RFC 7721, Security and Privacy Considerations for IPv6 Address Generation Mechanisms, March 20016https://www.rfc-editor.org/info/rfc7721
IETF RFC 7721，IPv6地址生成机制的安全和隐私考虑，20016年3月
https://www.rfc-editor.org/info/rfc7721

IETF RFC 7959, Block-Wise Transfers in the Constrained Application Protocol (CoAP), August2016
https://www.rfc-editor.org/info/rfc7959

IETF RFC 7959，受限应用程序协议(CoAP)中的块级传输，2016年8月。
https://www.rfc-editor.org/info/rfc7959

IETF RFC 8075, Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP), February 2017
https://www.rfc-editor.org/info/rfc8075
IETF RFC 8075，映射实现的指南：到受约束的应用程序的HTTP协议(CoAP)，2017年2月。
https://www.rfc-editor.org/info/rfc8075

IETF RFC 8288, Web Linking, October 2017
https://www.rfc-editor.org/info/rfc8288
IETF RFC 8288，网站链接，2017年10月
https://www.rfc-editor.org/info/rfc8288

IETF RFC 8323, CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets, February 2018
https://www.rfc-editor.org/info/rfc8323
IETF RFC 8323，基于TCP、TLS和WebSockets的CoAP(受约束的应用程序协议)，
https://www.rfc-editor.org/info/rfc8323

IANA ifType-MIB Definitions
https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib
IANA ifType-MIB定义
https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib

IANA IPv6 Multicast Address Space Registry
http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml
IANA IPv6多播地址空间注册表
http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xhtml

IANA Link Relations, October 2017
http://www.iana.org/assignments/link-relations/link-relations.xhtml
IANA链接关系 2017年10月
http://www.iana.org/assignments/link-relations/link-relations.xhtml

JSON Schema Validation, JSON Schema: interactive and non-interactive validation, January 2013
http://json-schema.org/draft-04/json-schema-validation.html
JSON模式验证，JSON模式:交互式和非交互式验证，2013年1月

http://json-schema.org/draft-04/json-schema-validation.html

OpenAPI specification, fka Swagger RESTful API Documentation Specification, Version 2.0
https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md
OpenAPI规范，fka Swagger RESTful API文档规范，2.0版
https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md

3 Terms, definitions, and abbreviated terms 术语、定义和缩略语

3.1 Terms and definitions 术语和定义

For the purposes of this document, the terms and definitions given in the following apply.
就本文件而言，下列术语与定义适用于本文件。

ISO and IEC maintain terminological databases for use in standardization at the following addresses:
国际标准化组织ISO和国际电工委员会IEC维护用于标准化的术语数据库，地址如下所示：

- ISO Online browsing platform: available at https://www.iso.org/obp.
  ISO在线浏览平台：https://www.iso.org/obp
- IEC Electropedia: available at http://www.electropedia.org/.
  IEC电工百科：http://www.electropedia.org/

3.1.1 Atomic Measurement 原子测量

a design pattern that ensures that the Client (3.1.6) can only access the Properties (3.1.33) of linked Resources (3.1.31) atomically, that is as a single group
一种确保客户端（3.1.6）只能以原子方式访问链接资源（3.1.31）的属性（3.1.33）的设计模式，即作为单个组。

3.1.2 Bridged Client 桥接客户端

logical entity that accesses data via a Bridged Protocol (3.1.4)
通过桥接协议访问数据的逻辑实体（3.1.4）

Note 1 to entry: For example, an AllJoyn Consumer application is a Bridged Client (3.1.2).
注：例如，AllJoyn消费者应用程序是一种桥接客户端（3.1.2）。

3.1.3 Bridged Device 桥接设备

Bridged Client (3.1.2) or Bridged Server (3.1.5)

桥接客户端（3.1.2） 或者桥接服务器（3.1.5）

3.1.4 Bridged Protocol 桥接协议

another protocol (e.g., AllJoyn) that is being translated to or from OCF protocols
转换为OCF协议或从OCF协议转换的另一种协议（例如AllJoyn）

3.1.5 Bridged Server 桥接服务器

logical entity that provides data via a Bridged Protocol (3.1.4)
通过桥接协议提供数据的逻辑实体（3.1.4）

Note 1 to entry: For example an AllJoyn Producer is a Bridged Server (3.1.5).
注1：例如，AllJoyn生成器是一种桥接服务器（3.1.5）。
Note 2 to entry: More than one Bridged Server (3.1.5) can exist on the same physical platform.
注2：同一物理平台上可以存在多个桥接服务器（3.1.5）。

3.1.6 Client 客户端

a logical entity that accesses a Resource (3.1.31) on a Server (3.1.37)
访问服务器（3.1.36）上资源（3.1.31）的逻辑实体

3.1.7 Collection 集合

a Resource (3.1.31) that contains zero or more Links (3.1.21)
包含0个或多个链接 (3.1.21）的资源（3.1.31）

3.1.8 Common Properties 公共属性

Properties (3.1.33) specified for all Resources (3.1.31)
为所有资源（3.1.31）指定的属性（3.1.33）

3.1.9 Composite Device 复合设备

a Device (3.1.13) that is modelled as multiple Device Types (3.1.14); with each component Device Type (3.1.14) being exposed as a Collection (3.1.7)
建模为多种设备类型（3.1.14）的设备（3.1.13）；将每个组件设备类型（3.1.14）公开为一个集合（3.1.7）

3.1.10 Configuration Source 配置源

a cloud or service network or a local read-only file which contains and provides configuration related information to the Devices (3.1.13)
包含并向设备（3.1.13）提供配置相关信息的云或服务网络或本地只读文件

3.1.11 Core Resources 核心资源

those Resources (3.1.31) that are defined in this document
本文件中定义的资源（3.1.31）

3.1.12 Default OCF Interface 默认OCF接口

an OCF Interface (3.1.18) used to generate the response when an OCF Interface (3.1.18) is omitted in a request
用来针对省略了OCF接口（3.1.18）的请求而产生响应的OCF接口（3.1.18）

3.1.13 Device 设备

a logical entity that assumes one or more roles, e.g., Client (3.1.6), Server (3.1.37)
承担一个或多个角色的逻辑实体，如客户端（3.1.6）、服务器（3.1.36）

Note 1 to entry: More than one Device (3.1.13) can exist on a Platform (3.1.30).
注：一个平台（3.1.30）上可以存在多个设备（3.1.13）。

3.1.14 Device Type 设备类型

a uniquely named definition indicating a minimum set of Resource Types (3.1.34) that a Device(3.1.13) supports
唯一命名的定义，指示设备（3.1.13）所支持的最小资源类型集（3.1.34）

Note 1 to entry: A Device Type (3.1.14) provides a hint about what the Device (3.1.13) is, such as a light or a fan, for use during Resource (3.1.31) discovery.
注：设备类型（3.1.14）提供有关在资源（3.1.31）发现期间所使用的设备（3.1.13）的提示，例如电灯或风扇。

3.1.15 Discoverable Resource 可发现资源

a Resource (3.1.31) that is listed in "/oic/res"
在“/oic/res”中列出的资源（3.1.31）

3.1.16 OCF Endpoint OCF端点

entity participating in the OCF protocol, further identified as the source or destination of a request and response messages for a given Transport Protocol Suite
参与OCF协议的实体，进一步标示为请求的源或目的地及针对给定传输协议套件的响应消息

Note 1 to entry: Example of a Transport Protocol Suite would be CoAP over UDP over IPv6.
注：传输协议族的例子是基于UDP和IPv6协议的CoAP。

3.1.17 Framework 框架

a set of related functionalities and interactions defined in this document, which enable interoperability across a wide range of networked devices, including IoT
本文档中定义的一组相关的功能和交互，可在包括物联网在内的各种网络设备之间实现互操作性。

3.1.18 OCF Interface OCF 接口

interface description in accordance with IETF RFC 6690 and as defined by OCF that provides a view to and permissible responses from a Resource (3.1.31)
符合IETF RFC 6690的描述和OCF定义的接口，提供对资源的查看和允许的响应（3.1.31）。

3.1.19 Introspection 内省

mechanism to determine the capabilities of the hosted Resources (3.1.31) of a Device (3.1.13)
确定设备所承载资源（3.1.31）的能力的机制（3.1.13）

3.1.20 Introspection Device Data (IDD) 内省设备数据

data that describes the payloads per implemented method of the Resources (3.1.31) that make up the Device (3.1.13)
通过构成设备（3.1.13）的资源（3.1.31）的实现方法对有效载荷进行描述的数据

Note 1 to entry: See 11.4 for all requirements and exceptions.
注：所有要求和异常见11.4。

3.1.21 Links 链接

extends typed web links according to IETF RFC 8288
根据IETF RFC 8288扩展类型化web链接

3.1.22 Non-Discoverable Resource 不可发现的资源

a Resource (3.1.31) that is not listed in "/oic/res"
“/oic/res”中未列出的资源（3.1.31）

Note 1 to entry: The Resource (3.1.31) can be reached by a Link (3.1.21) which is conveyed by another Resource(3.1.31). For example a Resource (3.1.31) linked in a Collection (3.1.7) does not have to be listed in "/oic/res", since traversing the Collection (3.1.7) would discover the Resource (3.1.31) implemented on the Device (3.1.13).
注：资源（3.1.31）可以通过由另一个资源（3.1.31）传递的链接（3.1.21）来访问。例如，在集合（3.1.7）中链接的资源（3.1.31）不必在“/oic/res”中列出，因为遍历集合（3.1.7）会发现在设备（3.1.13）上实现的资源（3.1.31）。

3.1.23 Notification 通知

the mechanism to make a Client (3.1.6) aware of state changes in a Resource (3.1.31)
让客户端（3.1.6）知道资源状态变化的机制（3.1.31）

3.1.24 Observe 观察

the act of monitoring a Resource (3.1.31) by sending a RETRIEVE operation which is cached by the Server (3.1.36) hosting the Resource (3.1.31) and reprocessed on every change to that Resource (3.1.31)
通过发送检索操作来监视资源（3.1.31）的操作，该操作由托管资源（3.1.31）的服务器（3.1.37）缓存，并在对资源（3.1.31）的每次更改时重新处理。

3.1.25 OpenAPI 2.0 OpenAPI 2.0

Resource (3.1.31) and Introspection Device Data (3.1.20) definitions used in this document as defined in the OpenAPI specification
本文档中使用的资源（3.1.31）和内省设备数据（3.1.20）的定义与OpenAPI规范中定义的一样

3.1.26 Parameter 参数

an element that provides metadata about a Resource (3.1.31) referenced by the target URI of a Link (3.1.21)
提供关于由链接（3.1.21）的目标URI引用的资源（3.1.31）的元数据的元素

3.1.27 Partial UPDATE 部分更新

an UPDATE operation to a Resource (3.1.31) that includes a subset of the Properties (3.1.33) that are visible via the OCF Interface (3.1.18) being applied for the Resource Type (3.1.34)

对资源（3.1.31）的更新操作，其中包括应用于资源类型（3.1.34）的OCF接口（3.1.18）可见的属性子集（3.1.33）。

3.1.28 Permanent Immutable ID 永久不变 ID

an identity for a Device (3.1.13) that cannot be altered
设备（3.1.13）不可改变的身份

3.1.29 Physical Device 物理设备

the physical thing on which a Device(s) (3.1.13) is exposed
设备（3.1.13）公开在其上的物理物体

3.1.30 Platform 平台

a Physical Device (3.1.29) containing one or more Devices (3.1.13)
包含一个或多个设备（3.1.13）的物理设备（3.1.29）

3.1.31 Resource 资源

represents an entity modelled and exposed by the Framework (3.1.17)
通过框架（3.1.17）的建模和公开来表示的实体

3.1.32 Resource Interface 资源接口

a qualification of the permitted requests on a Resource (3.1.31)
对资源（3.1.31）的许可请求进行限定

3.1.33 Property 属性

a significant aspect or Parameter (3.1.26) of a Resource (3.1.31), including metadata, that is exposed through the Resource (3.1.31)
包括元数据在内的、通过资源（3.1.31）公开的重要方面或参数（3.1.26）

3.1.34 Resource Type 资源类型

a uniquely named definition of a class of Properties (3.1.33) and the interactions that are supported by that class
一类属性（3.1.33）的唯一命名定义，以及该类所支持的交互。

Note 1 to entry: Each Resource (3.1.31) has a Property (3.1.33) "rt" whose value is the unique name of the Resource Type (3.1.34).
注：每个资源（3.1.31）都有一个属性（3.1.33）“rt”，其值是资源类型（3.1.34）的唯一名称。

3.1.35 Secure OCF Endpoint 安全的OCF端点

an OCF Endpoint (3.1.16) with a secure connection (e.g., CoAPS)
具有安全连接（例如，CoAPS）的OCF端点（3.1.16）。

3.1.36 Semantic Tag 语义标签

meta-information that provides additional contextual information with regard to the Resource(3.1.31) that is the target of a Link (3.1.21)
元信息，提供关于链接目标资源（3.1.31）的附加上下文信息（3.1.21）。

3.1.37 Server 服务器

a Device (3.1.13) with the role of providing Resource (3.1.31) state information and facilitating remote interaction with its Resources (3.1.31)
一种设备（3.1.13），其作用是提供资源（3.1.31）状态信息并促进其资源的远程交互（3.1.31）。

3.1.38 Unsecure OCF Endpoint 不安全的OCF端点

an OCF Endpoint (3.1.16[1]) with an unsecure connection (e.g., CoAP)
具有不安全连接的OCF端点（3.1.16[1]）（例如，CoAP）

3.1.39 Vertical Resource Type 垂直的资源类型

a Resource Type (3.1.34) in a vertical domain specification
垂直域规范中的资源类型（3.1.34）

Note 1 to entry: An example of a Vertical Resource Type (3.1.39) would be "oic.r.switch.binary".
注：垂直资源类型（3.1.39）的一个例子是“oic.r.switch.binary”。

3.1.40 Virtual OCF Client 虚拟OCF客户端

logical representation of a Bridged Client (3.1.2), which an Bridged Device (3.1.3) exposes to Servers (3.1.36)
桥接客户端（3.1.2）的逻辑表示，桥接设备（3.1.3）将其公开给服务器（3.1.36）。

3.1.41 Virtual OCF Device (or VOD) 虚拟OCF设备（or VOD）

Virtual OCF Client (3.1.40) or Virtual OCF Server (3.1.42)
虚拟OCF客户端（3.1.40）或虚拟OCF服务器（3.1.42）

[1] 英文版原文为()，疑似编辑错误，应为（3.1.16）；本译本中已更改。

3.1.42 Virtual OCF Server 虚拟OCF服务器

logical representation of a Bridged Server (3.1.5), which an Bridged Device (3.1.3) exposes to Clients (3.1.6)
桥接服务器（3.1.5）的逻辑表示，桥接设备（3.1.3）将其公开给客户端（3.1.6）。

3.2 Abbreviated terms 缩略语

3.2.1 ACL

Access Control List
访问控制列表

Note 1 to entry: The details are defined in ISO/IEC 30118-2:2018.
注：具体规定见ISO/IEC 30118-2:2018。

3.2.2 BLE

Bluetooth Low Energy
低功耗蓝牙

3.2.3 CBOR

Concise Binary Object Representation
简明的二进制对象表示

3.2.4 CoAP

Constrained Application Protocol
受限应用协议

3.2.5 CoAPS

Secure Constrained Application Protocol
安全受限应用协议

3.2.6 DTLS

Datagram Transport Layer Security
数据报传输层安全

Note 1 to entry: The details are defined in IETF RFC 6347.
注：详细信息在IETF RFC 6347中定义。

3.2.7 EXI

Efficient XML Interchange
有效的XML交换

3.2.8 IP

Internet Protocol
互联网协议

3.2.9 IRI

Internationalized Resource Identifiers
国际化资源标识符

3.2.10 ISP

Internet Service Provider
互联网服务提供商

3.2.11 JSON

JavaScript Object Notation
JavaScript对象表示法

3.2.12 mDNS

Multicast Domain Name Service
多播域名服务

3.2.13 MTU

Maximum Transmission Unit
最大传输单元

3.2.14 NAT

Network Address Translation
网络地址转换

3.2.15 OCF

Open Connectivity Foundation
开放互联基金会

the organization that created this document
创建此文档的组织

3.2.16 REST

Representational State Transfer
表述性状态转移

3.2.17 RESTful

REST-compliant Web services
符合REST的Web服务

3.2.18 UDP

User Datagram Protocol
用户数据报协议

Note 1 to entry: The details are defined in IETF RFC 768.
注：详细信息在IETF RFC 768中定义

3.2.19 URI

Uniform Resource Identifier
统一资源标识符

3.2.20 URN

Uniform Resource Name
统一资源名称

3.2.21 UTC

Coordinated Universal Time
协调的通用时间

3.2.22 UUID

Universal Unique Identifier
通用唯一识别码

3.2.23 XML

Extensible Markup Language
可扩展标记语言

4 Document conventions and organization 文档惯例和组织

4.1 Conventions 惯例

In this document a number of terms, conditions, mechanisms, sequences, parameters, events,states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Network Architecture). Any lowercase uses of these words have the normal technical English meaning.
在本文件中，许多术语、条件、机制、序列、参数、事件、状态或类似的术语都是用大写每个单词的第一个字母且其他字母小写来表示的（例如，Network Architecture）。这些单词的任何小写用法都具有正常的技术英语含义。

4.2 Notation 符号

In this document, features are described as required, recommended, allowed or DEPRECATED as follows:
在本文件中，特性按照下述的必选、推荐、允许或不推荐使用来进行描述：

- Required (or shall or mandatory)(M).
  必选（或须填或强制填）(M)。
    - These basic features shall be implemented to comply with Core Architecture. The phrases "shall not", and "PROHIBITED" indicate behavior that is prohibited, i.e. that if performed means the implementation is not in compliance.
      这些基本特性应该按照核心架构来实现。短语“不得”、“禁止”表示禁止的行为，即如果实施，则表示该实施不符合要求。
- Recommended (or should)(S).
  推荐（或应该）(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 behavior that is permitted but not recommended.
      这些特性添加了核心体系结构支持的功能，应该加以实现。推荐的特性利用核心架构的功能，通常不会增加复杂性。请注意，对于符合性测试，如果实现了推荐的特性，则它应满足满足本指南的指定要求。一些推荐的特性在将来可能成为要求。短语“should not”表示允许但不建议的行为。

- Allowed (may or allowed)(O).
  允许（可以或允许）(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.
      这些特性既不是核心架构所要求的，也不是核心架构所推荐的，但是如果实现了这些特性，它应该满足特定的需求，以符合这些指南。

- DEPRECATED.
  不推荐使用

    - Although these features are still described in this document, they should not be implemented except for backward compatibility. The occurrence of a deprecated feature during operation of an implementation compliant with the current documentas[2] no effect on the implementation's operation and does not produce any error conditions. Backward compatibility may require that a feature is implemented and functions as specified but it shall never be used by implementations compliant with this document.
      尽管这些特性在本文件中仍然有描述，但是除了向后兼容之外，不应该实现它们。在与当前文档兼容的实现的操作过程中，不支持的特性的出现不会影响实现的操作，也不会产生任何错误条件。向后兼容可能需要实现某种特性并按规定运行，但与本文件兼容的实现决不能使用该功能。

- Conditionally allowed (CA).
  有条件地允许（CA）

    - The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is allowed, otherwise it is not allowed.
      取决于某一条件的定义或行为。如果满足指定的条件，则允许定义或行为，否则不允许。

- Conditionally required (CR).
  有条件的要求（CR）

    - The definition or behavior depends on a condition. If the specified condition is met, then the definition or behavior is required. Otherwise the definition or behavior is allowed as default unless specifically defined as not allowed.
      取决于某一条件的定义或行为。如果满足指定的条件，则需要定义或行为。否则，除非明确定义为不允许，否则允许将定义或行为作为默认值。

Strings that are to be taken literally are enclosed in "double quotes".
按字面意思理解的字符串包含在“双引号”中。

---

[2] 原文如此（documentas），应为document has的笔误；中文版按document has进行翻译。

Words that are emphasized are printed in italic.
强调的词用斜体印刷。

In all of the Property and Resource definition tables that are included throughout this document the "Mandatory" column indicates that the item detailed is mandatory to implement; the mandating of inclusion of the item in a Resource Payload associated with a CRUDN action is dependent on the applicable schema for that action.
在本文件中包含的所有属性和资源定义表中，“必填”列所详述的项目必须实现；是否将项目包含在与CRUDN动作关联的资源有效负载中取决于该操作的适用模式。

4.3 Data types 数据类型

Resources are defined using data types derived from JSON values as defined in IETF RFC 7159.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.
资源是使用IETF RFC 7159中定义的JSON值派生的数据类型定义的。但是，资源可以使用JSON模式验证中定义的验证关键字重载JSON定义的值，以指定JSON值的特定子集。

Among other validation keywords, clause 7 in 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 clause defines patterns that are available for use in describing OCF Resources. The pattern names can be used in documenttext where JSON format names can occur. The actual JSON schemas shall use the JSON type and pattern instead.
在其他验证关键字中，JSON模式验证中的条款7定义了一个“format”关键字，该关键字具有许多格式属性，如“uri”和“date-time”，以及一个“pattern”关键字，该关键字具有一个正则表达式，可用于验证字符串。此条款定义可用于描述OCF资源的模式。可以在文档文本中使用模式名，其中可以出现JSON格式的名称。实际的JSON模式应该使用JSON类型和模式。

For all rows defined in Table 1, the JSON type is string.
对于表1中定义的所有行，JSON类型都是字符串型。

Strings shall be encoded as UTF-8 unless otherwise specified.
除非另有说明，否则字符串应使用UTF-8编码。

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.
在JSON模式中，字符串的“maxLength”表示最大的字符数而非八位元组数。但是，“maxLength”也应该表示最大的字节数。如果没有为字符串定义“maxLength”，那么最大长度应为64字节。

Table 1 – Additional OCF Types
表1 - 附加OCF类型

Pattern Name 模式类型	Pattern 模式	Description 描述
"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. 用逗号分隔的、编码于字符串内的值列表。csv中的值类型由使用csv的属性描述。例如，一个整数的csv。 NOTE csv is considered deprecated and an array of strings should be used instead for new Resources. 注:csv被认为已弃用，而对于新资源，应当使用字符串组代替。
"date"	^([0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|2[0-9]|1[0-9]|0[1-9])$	The full-date format pattern according to IETF RFC 3339 根据IETF RFC 3339的完整格式模式
"duration"	^(P(?!$)([0-9]+Y)?([0-9]+M)?([0-9]+W)?([0-9]+D)?((T(?=[0-9]+[HMS])([0-9]+H)?([0-9]+M)?([0-9]+S)?)?))$|^(P[0-9]+W)$\|^(P[0-9]{4})-(1[0-2]\|0[1-9])-(3[0-1]\|2[0-9]\|1[0-9]\|0[1-9])T(2[0-3]\|1[0-9]\|0[1-9]):([0-5][0-9]):([0-5][0-9])$|^(P[0-9]{4})(1[0-2]|0[1-9])(3[0-1]|2[0-9]|1[0-9]|0[1-9])T(2[0-3]|1[0-9]|0[1-9])([0-5][0-9])([0-5][0-9])$	A string representing duration formatted as defined in ISO 8601. 表示持续时间的字符串，其格式如ISO 8601所定义。 Allowable formats are: 允许的格式为： P[n]Y[n]M[n]DT[n]H[n]M[n]S, P[n]W,P[n]Y[n]-M[n]-DT[0-23]H[0-59]:M[0-59]:S, and P[n]W, P[n]Y[n]M[n]DT[0-23]H[0-59]M[0-59]S. P is mandatory,all other elements are optional, time elements must follow a T. P是强制的，所有其他元素都是可选的，时间元素必须跟随T。
"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] 如果字符串实例包含[-（2 ** 63），（2 ** 63）-1]范围内的整数，则对此属性有效。 NOTE IETF RFC 7159 clause 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. 注：IETF RFC 7159条款6解释了[-（2 ** 53）1，（2 ** 53）-1]范围之外的JSON整数不可互操作，因此JSON数字不能用于64位数字。
"language-tag"	^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$	An IETF language tag formatted according to IETF RFC 5646 clause 2.1. 根据IETF RFC 5646条款2.1格式化的IETF语言标签。

"unit64"	^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] 如果字符串实例包含[0,（2 ** 64）-1]范围内的整数，则对此属性有效。 Also see note for "int64" 另请参见“ int64”的注释
"uuid"	^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fAF0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}$	A UUID string representation formatted according to IETF RFC 4122 clause 3. 根据IETF RFC 4122条款3格式化的UUID字符串表示形式。

4.4 Resource notation syntax 资源符号语法

When it is desired to describe the Property of a Resource Type or the "anchor" Parameter value in an abbreviated notation, it can be described as follows:
当需要用缩写表示法描述资源类型的属性或"锚定"参数值时，可以描述如下：

- A value of the "rt" Property of the Resource Type or "anchor" Parameter value ":" Property name.
  资源类型的“rt”属性值或“锚”参数值":"“属性名。
- e.g., "oic.wk.d:di", which is the "di" Property of the Device Resource Type.
  例如“oic.wk.d:di”，这是设备资源类型的“di”属性。

If Property name is a composite type (a type that is composed of several Properties), it can be described in recursive way. The following expression describes this as a regular expression format:
如果属性名是复合类型（由多个属性组成的类型），则可以递归方式对其进行描述。以下表达式将其描述为正则表达式格式：

- A value of the "rt" Property of the Resource Type or "anchor" Parameter value (":" Property name)+
  资源类型的“rt”属性值或“anchor”参数值（":"属性名字）+
- e.g., "oic.r.pstat:dos:s", which is the "s" Property of the "dos" Property of the "pstat" Resource Type (see 13.8 of ISO/IEC 30118-2:2018).
  例如，“oic.r.pstat:dos:s”，这是“pstat”资源类型的“dos”属性的“s”属性（参见ISO/IEC 30118-2-2018的13.8）。

If there is a Resource URI (i.e., The Resource instance for a specific Resource Type), it can be used instead of using a value of "rt" Property of Resource Type or the “anchor" Parameter value as follows:
如果存在资源URI（即特定资源类型的资源实例），则可以使用该URI，而不使用资源类型的“rt”属性值或“anchor”参数值，如下所示：

- A Resource URI (":" Property name)+
  资源URI（":"属性名）+

- e.g., "/oic/d:di", which is the "di" Property of the Device Resource Type instance.
  例如，“/oic/d:di”，这是设备资源类型实例的“di”属性。
- e.g. "/oic/sec/pstat:dos:s", which is the "s" Property of the "dos" Property of the "oic.r.pstat" Resource Type instance.
  例如“/oic/sec/pstat:dos:s”，这是“oic.r.pstat”的“dos”属性的“s”属性资源类型实例。

In the auto-generated Annex's Property definition tables for Resource Types, the Property names can be noted as belonging to the RETRIEVE schema or to the UPDATE schema by prefixing the Property name with "RETRIEVE" or "UPDATE" followed with the ":" separator. This is to avoid duplicate Property names appearing in the Property definition tables that are auto-generated. The following are examples using this notation with the "locn" Property of the "oic.wk.con" Resource Type:
在自动生成的附件的资源类型属性定义表中，可以通过在属性名称前面加上“RETRIEVE”或“UPDATE”并后跟“:”分隔符，将属性名称标记为属于检索架构或更新架构。这是为了避免在自动生成的特性定义表中出现重复的特性名称。下面是将此符号与“oic.wk.con”资源类型的“locn”属性一起使用的示例：

- "RETRIEVE:locn"
- "UPDATE:locn"

5 Architecture 架构

5.1 Overview 概述

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.
该架构支持物联网人工物品（即物理设备或应用程序）之间基于资源的交互。该架构利用现有的行业标准和技术，为建立连接（无线或有线）和管理设备之间的信息流提供解决方案，且不必考虑这些设备的形式因素、操作系统或服务提供商如何。

Specifically, the architecture provides:
具体而言，该架构提供：

- A communication and interoperability framework for multiple market segments (Consumer, Enterprise, Industrial, Automotive, Health, etc.), OSs, platforms, modes of communication, transports and use cases.
  针对多个细分市场（消费者、企业、工业、汽车、健康等）、操作系统、平台以及不同的通信方式、传输和用例的通信和互操作性框架。

- A common and consistent model for describing the environment and enabling information and semantic interoperability.
  用于描述环境、使信息和语义具备互操作性的通用和一致的模型。
- Common communication protocols for discovery and connectivity.
  用于发现和连接的通用通信协议。
- Common security and identification mechanisms.
  常见的安全和识别机制。
- Opportunity for innovation and product differentiation.
  创新和产品差异化的机会。
- A scalable solution addressing different Device capabilities, applicable to smart devices as well as the smallest connected things and wearable devices.
  针对不同设备功能的可扩展解决方案，其适用于智能设备以及最小的连接物品和可穿戴设备。

The architecture is based on the Resource Oriented Architecture design principles and described in the 5.2 through 5.4 respectively. 5.2 presents the guiding principles for OCF operations. 5.3 defines the functional block diagram and Framework.
该架构基于面向资源的架构设计原则，对此在5.2至5.4中有相关描述。5.2介绍了OCF操作的指导原则。5.3定义了功能框图和框架。

5.2 Principle 原则

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 (see 7.6.3.9) using operations that adhere to Representational State Transfer (REST) architectural style, i.e., RESTful interactions.
在该架构中，物理世界中的实体（例如温度传感器、电灯或家用电器）被表示为资源。与实体的交互通过其资源表示（请参见7.6.3.9）并使用符合表示状态转移（REST）体系结构样式的操作来实现，即RESTful交互。

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. 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 clause 8, which include representations of Resources.
该体系结构将框架的总体结构定义为一个信息系统，以及构成OCF的实体之间的相互关系。实体以资源的形式被公开，具有其唯一标识符（URI）和支持接口，这些接口使能对资源的RESTful操作。每个RESTful

操作都有操作的发起方（客户端）和操作的响应方（服务器）。在该框架中，通过角色实现客户端和服务器的概念。任何设备都可以充当客户端，并可以在充当服务器的任何设备上启动RESTful操作。同样，任何将实体公开为资源的设备都可以充当服务器。与REST架构风格一致，每个RESTful操作都包含理解交互上下文所需的所有信息，并使用一组小的通用操作（即第8章中定义的创建、检索、更新、删除和通知（CRUDN））来驱动，其中包括资源的表示。

Figure 1 depicts the architecture.
图1描述了该架构。

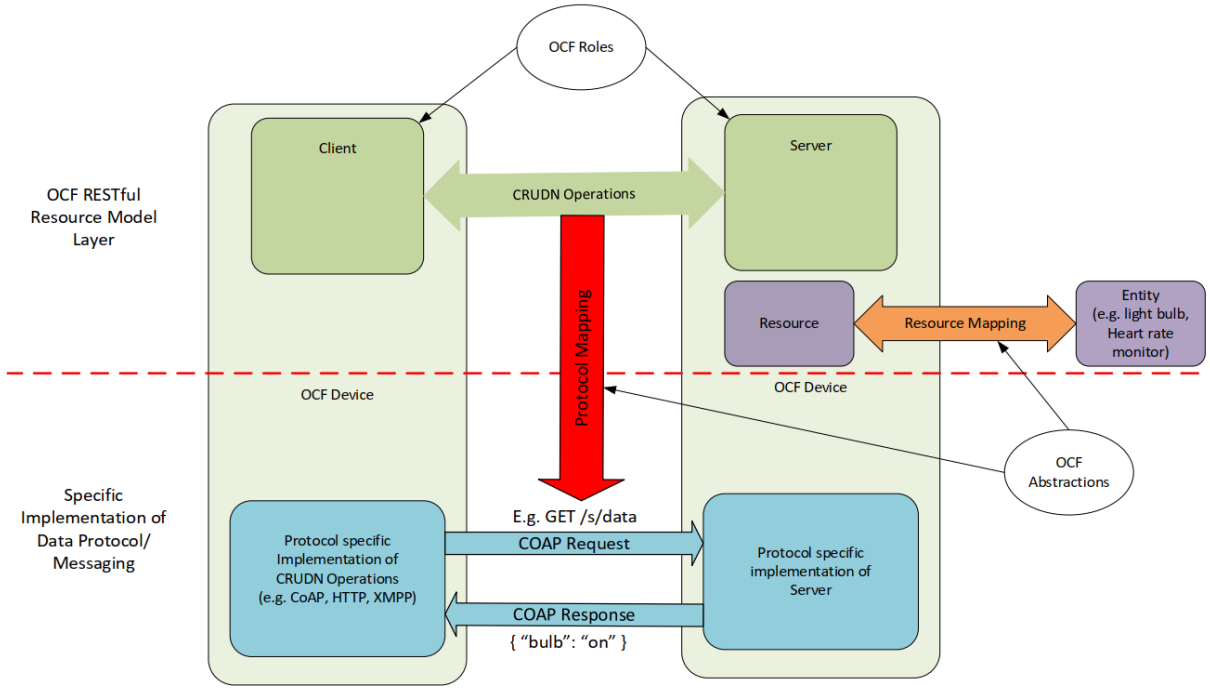

Figure 1 – Architecture - concepts 图1 – 架构-概念

The architecture is organized conceptually into three major aspects that provide overall separation of concern: Resource model, RESTful operations and abstractions.
该架构在概念上分为三个主要方面，提供了整体的独立关注点：资源模型、RESTful操作和抽象。

- 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, 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 behavior.
  资源模型：资源模型提供逻辑上建模并在应用程序及其环境上进行逻辑操作所需的抽象和概念。核心资源模型对任何特定的应用领域（如智能家居、工业或汽车）都是通用的和不可知的。例如，资

源模型定义了实体抽象化后的资源，资源的表示形式映射了实体的状态。其它资源模型概念可用于其他方面的建模，例如行为。

- 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 11.4.
  RESTful操作：通用CRUDN操作是使用RESTful范式定义，以便以与协议和技术无关的方式对与资源的交互进行建模。特定的通信或消息传递协议是协议抽象化的一部分，11.4节中介绍了资源到特定协议的映射。
- 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 on 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.
  抽象化：使用抽象原语将资源模型中的抽象符号和RESTful操作映射到具体元素中。使用实体处理程序将实体映射向资源，并且连接抽象原语用于将逻辑RESTful操作映射到数据连接协议或技术。实体处理程序也可用于将资源映射到实体，这种映射是通过非OCF原生支持的协议而达到的。

5.3 Functional block diagram 功能框图

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.
功能框图包含操作所需的所有功能。这些功能分为L2连接、联网、传输、框架和应用程序配置文件。功能块如图2所示。

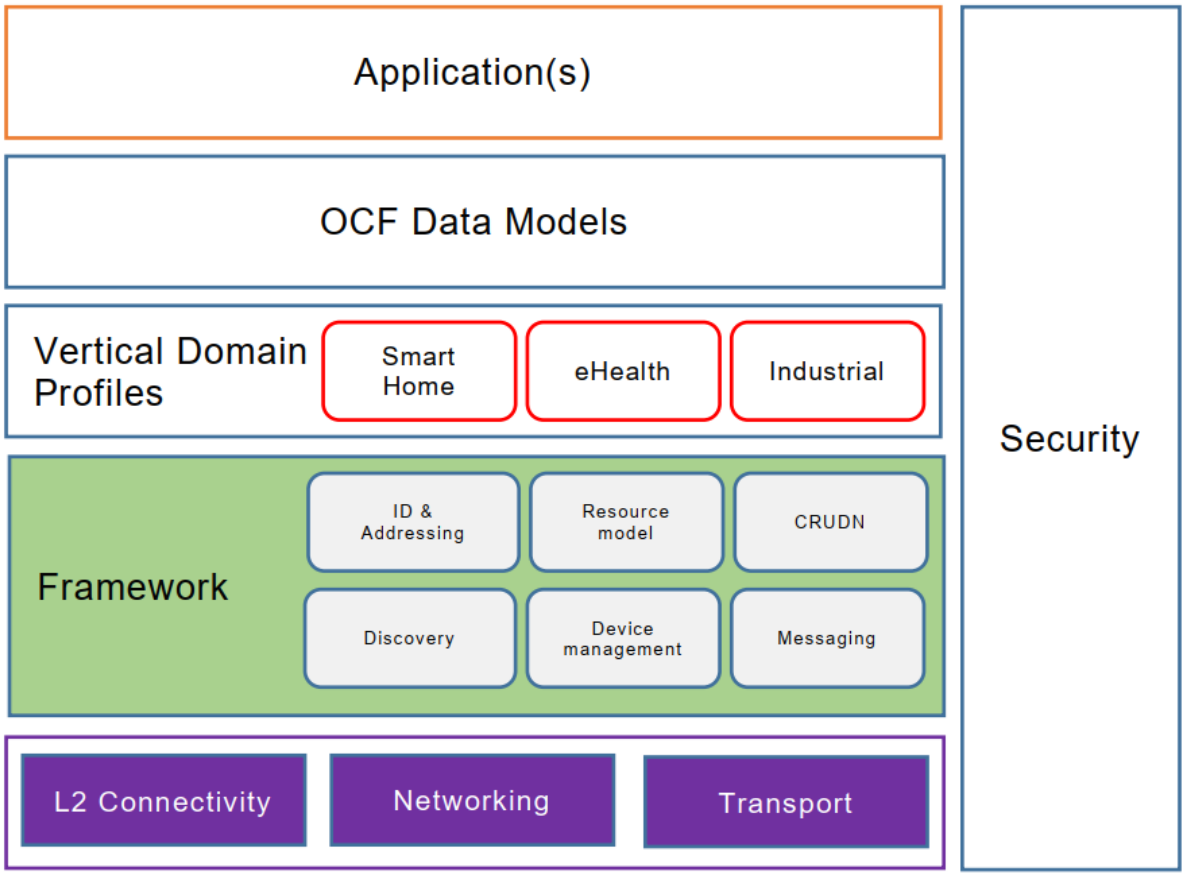

Figure 2 – Functional block diagram 图2 – 功能框图

- L2 connectivity: Provides the functionalities required for establishing physical and data link layer connections (e.g., Wi-Fi™ or Bluetooth® connection) to the network.
  L2连接：提供为网络建立物理和数据链路层连接（例如，Wi-Fi™或蓝牙连接）所需的功能。
- Networking: Provides functionalities required for Devices to exchange data among themselves over the network (e.g., Internet).
  联网：提供设备之间通过网络（例如因特网）交换数据所需的功能。
- 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).
  传输：提供具有特定QoS约束的端到端流量传输。传输协议的示例包括 TCP和UDP，或在IETF中正在开发的新传输协议，例如，时滞容错网络（DTN）。
- Framework: Provides the core functionalities as defined in this document. The functional block is the source of requests and responses that are the content of the communication between two Devices.
  框架：提供本文档中定义的核心功能。功能块是请求和响应的来源，请求和响应是两个设备之间的通信内容。

- Vertical Domain profile: Provides market segment specific functionalities, e.g., functions for the smart home market segment.
  垂直域配置文件：提供细分市场的特定功能，例如智能家居市场部分的功能。

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.
当两个设备相互通信时，设备中的每个功能块与对等设备中的对应程序进行交互，如图3所示。

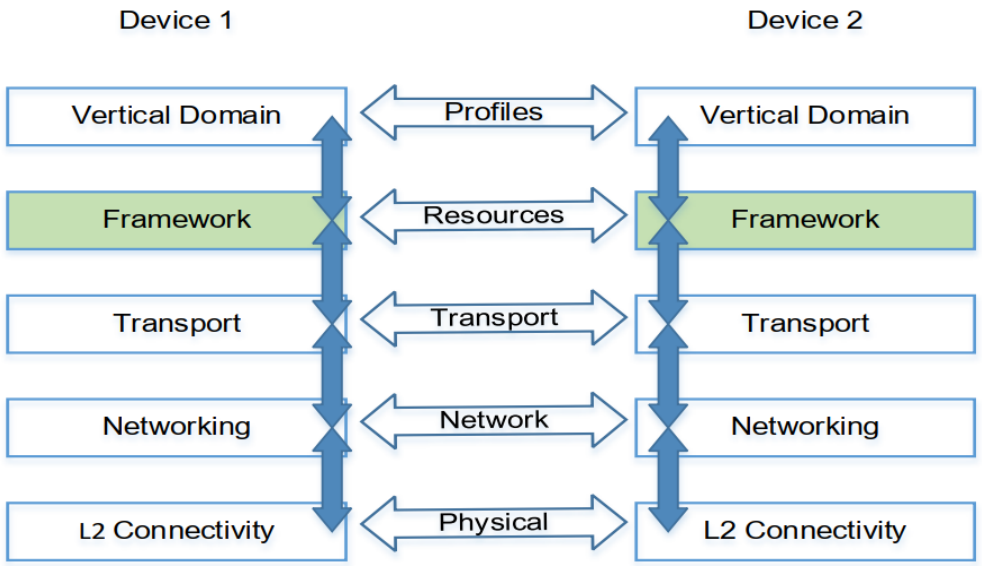

Figure 3 – Communication layering model 图3 – 通信分层模型

5.4 Framework 框架

Framework consists of functions which provide core functionalities for operation.
框架由为操作提供核心功能的功能组件组成。

- Identification and addressing. Defines the identifier and addressing capability. The Identification and addressing function is defined in clause 6.
  识别和寻址。定义标识符和寻址功能。识别和寻址功能见第6章。
- Discovery. Defines the process for discovering available.
  发现。定义发现可用项的过程。
- Devices (OCF Endpoint Discovery in clause 10) and
  设备（第10章中的OCF端点发现项）和
- Resources (Resource discovery in 11.2).
  资源（第11.2条款中的资源发现项）。
- Resource model. Specifies the capability for representation of entities in terms of Resources And defines mechanisms for manipulating the Resources. The Resource model function isdefined in

clause 7.
资源模型。指定了用资源表示实体的能力，并定义了操作资源的机制。在第7章中定义了资源模型函数。

- CRUDN. Provides a generic scheme for the interactions between a Client and Server as defined in clause 8.
  CRUDN。为客户端和服务器之间的交互提供一个通用方案，如第8章所定义。
- 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 11.5.
  信息。为RESTful操作提供特定的消息协议，即CRUDN。例如，CoAP是一种消息传递协议。在11.5条款定义了消息传递功能。
- Security. Includes authentication, authorization, and access control mechanisms required for secure access to Entities. The security function is defined in clause 13.
  安全。包括安全地访问实体所需的身份验证、授权和访问控制机制。在第13章中定义了安全功能。

6 Identification and addressing 识别和寻址

6.1 Introduction 引言

Facilitating proper and efficient interactions between elements in the Framework, requires a means to identify, name and address these elements.
为使框架中元素之间达到适当和有效的交互，需要一种能够识别、命名和处理这些元素的方法。

The identifier unambiguously 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.
标识符可以明确地标识上下文或域中的元素。可以通过用法或应用方式决定上下文或域。我们希望在该元素的生命周期中该标识符不可变，并且在上下文或域中可以明确确定。

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.
地址可以用于定义元素的位置、作为获得或访问元素的方式或方法，以便与元素进行交互。根据上下文，地址可以是可变的。

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.
名称可以作为一种称呼，可以将某种元素与框架中的其它元素区分开来。在该元素的生命周期中，可以更改名称。

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).

可能有一些方法或解决方案允许根据一个或多个其它知识来确定地址或名称中任何一个（例如，从地址确定名称或从名称确定地址）。

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.
这些方面中的每一个都可以针对多个上下文分别定义（例如，一个上下文可以是堆栈中的一层）。因此，地址可以是用于寻址资源的URL和用于在连接层寻址的IP地址。在某些情况下，这两个地址都是必需的。

For example, to do RETRIEVE (see 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.
例如，要对特定资源表示执行检索（请参见8.3）操作，客户端需要知道目标资源的地址和公开资源的服务器地址。

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.
在使用的上下文或域中，名称或地址可以用作标识符，反之亦然。例如，URL可以用作资源的标识符并可以被指定为URI。

The remainder of this clause 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 (clause 8) on a Resource. Also The mapping of these to transport protocols, e.g., CoAP is described.
本章的其余部分从资源模型和资源模型支持的交互的角度讨论标识符、地址和命名。关于交互的举例则是RESTful交互，即资源上的CRUDN操作（第8章）。此外，到传输协议的映射，例如文中所描述的CoAP，也是与交互有关的一个例子。

6.2 Identification 识别

6.2.1 Device and Platform identification 设备和平台识别

This document defines three identifiers that are used for identification of the Device. All identifiers are exposed via Resources that are also defined within this document (see clause 11.2).
本文件定义了三个用于识别设备的标识符。所有标识符都是通过本文件中定义的资源公开的（参见第11.2条）。

The Permanent Immutable ID ("piid" Property of "/oic/d") is the immutable identity of the Device ,the persistent valid value of this property is typically only visible after the Device is on-boarded(when not on-boarded the Device typically exposes a temporary value). This value does not change across the life-cycle of the Device.

永久不可变ID（“/oic/d”的“piid”属性）是设备的不可变标识，该属性的持久有效值通常只有在设备安装后才可见（未安装时，设备通常公开一个临时值）。该值在设备的整个生命周期中不会改变。

The Device ID ("di" Property of "/oic/d") is a mutable identity. The value changes each time the Device is on-boarded. It reflects a specific on-boarded instance of the Device.
设备ID（“/oic/d”的“di”属性）是一个可变标识。每次设备安装时，该值都会更改。它反映了设备的特定安装实例。

The Platform ID ("pi" Property of "/oic/p") is the immutable identity of the Platform on which the Device is resident. When multiple logical Devices are exposed on a single Platform (for example,on a Bridge) then the "pi" exposed by each Device should be the same.
平台ID（“/oic/p”的“pi”属性）是设备所在平台的不可变标识。当多个逻辑设备出现在一个平台上（例如，在桥接时），那么每个设备出现的“pi”应该是相同的。

6.2.2 Resource identification and addressing 资源标识和寻址

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.
资源可以使用URI标识，如果URI是URL，则可以使用相同的URI寻址。在某些情况下，资源可能需要与URI不同的标识符；在这种情况下，资源可能有一个属性，其值是标识符。当URI以URL的形式存在时，可以使用URI来寻址资源。

An OCF URI is based on the general form of a URI as defined in IETF RFC 3986 as follows:
OCF URI基于IETF RFC 3986中定义的URI的一般形式，如下所示:

```
<scheme>://<authority>/<path>?<query>
```

Specifically the OCF URI is specified in the following form:
具体来说，OCF URI以以下形式指定：

```
ocf://<authority>/<path>?<query>
```

The following is a description of values that each component takes.
下面是对每个组件采用的值的描述。

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.
URI的模式是“ocf”。“ocf”模式表示了按本文件定义的语义、定义和使用。如果URI省略了“//”（双斜杠）前面的部分，那么应假定采用了“ocf”模式。

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.
每个传输绑定负责在请求者通过网络发送之前指定OCF URI如何转换为传输协议URI。类似地，在接收端，每个传输绑定负责指定OCF URI在移交给接收端的资源模型层之前如何从传输协议URI转换为OCF URI。

The authority of an OCF URI shall be the Device ID ("di") value, as defined in [OCF Security], of the Server.
OCF URI的权限应该是服务器的设备ID（“di”）值，如[OCF Security]中定义的那样。

The path is a string that unambiguously identifies or references a Resource within the context of the Server. In this version of the document, 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.
路径是一个字符串，它明确地标识或引用服务器上下文中的资源。在这个版本的文件中，路径不能包含pct编码的非ascii字符或NUL字符。路径的前面应该有一个“/”（斜线）。路径包含以“/”分隔的段以便于阅读。在OCF上下文中，“/”（斜杠）分隔的段被视为一个单独的字符串，它直接引用资源（即一个平面结构），而不是作为层次结构进行解析。在服务器端，如果结果引用在主机上下文中是惟一的，则可以使用散列法或其他方案缩短路径或路径中的某些子字符串。

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.
一旦生成了路径，访问资源的客户端或URI的接受者应该将该路径用作不透明的字符串，而不应该通过解析来推断结构、组织或语义。

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).
查询字符串应该包含一个“<name>=<value>”段（又称为名值对）列表，每个段之间用“&”（& and）分隔。查询字符串将映射到用于消息传递的协议的适当语法。(例如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 OCF Interface, based on some rules or with respect to different roots or bases.

URI可以是完全限定的URI，也可以是URI的相对生成。URI可以由创建该资源的客户机定义。这样的URI可以是相对的或绝对的（完全限定的）。一个相对URI应该相对于它所在的设备。或者，该资源的服务器可以根据预定义的约定或资源的组织、OCF接口、一些规则或不同的根或基自动生成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 addressesand Resource b cannot be construed as a child of Resource a).
URI的绝对路径引用将被视为不透明的字符串，客户端不应该推断URI中的任何显式或隐含的结构——URI只是一个地址。还建议承载资源的设备将每个资源的URI视为只处理该资源的不透明字符串。（例如，URI的“/a”和“/a/b”被认为是不同的地址，而资源b不能被解释为资源a的子资源。）

6.3 Namespace：命名空间：

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. The prefix "oic." used for OCF Interfaces and Resource Types is reserved for OCF specification usage.
相对URI前缀“/oic/”保留为OCF规范中定义的URI的名称空间，不应用于OCF规范中未定义的URI。用于OCF接口和资源类型的前缀“oic.”保留给OCF规范使用。

6.4 Network addressing 网络寻址

The following are the addresses used in this document:
以下是本文件所使用的地址：

IP address
IP 地址

- An IP address is used when the Device is using an IP configured interface.
  当设备使用配置好的IP接口时，使用IP地址。
- When a Device only has the identity information of its peer, a resolution mechanism is needed to map the identifier to the corresponding address.
  当一个设备只有它的对等设备的身份信息时，需要一个解析机制来将标识符映射到相应的地址。

7 Resource model 资源模型

7.1 Introduction 引言

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.
资源模型定义了在OCF生态系统中提供设备之间的一致性和核心互操作性的概念和机制。然后，将资源模型概念和机制映射到传输协议，以支持设备之间的通信——每个传输协议提供通信协议互操作性。因此，资源模型允许独立于传输而定义互操作性。

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.
此外，资源模型中的概念支持对主要人工制品及其彼此之间的关系进行建模，并捕获上下文中互操作性所需的语义信息。通过这种方式，OCF超越了简单的协议互操作性，获得了可穿戴和物联网生态系统中真正互操作性所需的丰富语义。

The primary concepts in the Resource model are: entity, Resources, Uniform Resource Identifiers (URI), Resource Types, Properties, Representations, OCF 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 Framework is applied to.
资源模型中的主要概念有:实体、资源、统一资源标识符（URI）、资源类型、属性、表示形式、OCF接口、集合和链接。此外，一般机制是创建（CREATE）、检索（RETRIEVE）、更新（UPDATE）、删除（DELETE）和通知（NOTIFY）。这些概念和机制可以以各种方式组合，以定义框架所应用的不同用例集所需的丰富语义和互操作性。

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.
在OCF资源模型框架中，实体需可见、能交互或背操作，它由称为资源的抽象来表示。资源封装并表示实体的状态。使用URI标识、寻址和命名资源。

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 OCF Interfaces. Interactions with a Resource are done as Requests and Responses containing Representations.
属性是“key=value”对，表示资源的状态。这些属性的快照是资源的表示。表示的特定视图和该视图中适用的机制被指定为OCF接口。与资源的交互是通过包含了表示的请求和响应来完成的。

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.

资源实例派生自资源类型。一个资源和另一个资源之间的单向关系被定义为链接。具有属性和链接的资源是一个集合。

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.
一组属性可用于定义资源的状态。可以在来自该资源的响应和对该资源的请求中使用适当的表示来检索或更新此状态。

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.
资源（和资源类型）可以表示并用于公开功能。与该资源的交互可用于锻炼或使用该功能。这些功能可用于定义发现、管理、公告等流程。例如：在设备上发现资源可以定义为对特定资源表示的检索，其中一个或多个属性具有描述或引用设备上资源的值。

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 11.4 for transport protocols supported.
带有表示的请求或响应的信息可以通过使用传输协议进行序列化或封装在传输协议的有效负载中进行在线通信—具体方法由请求或响应到传输协议的规范映射确定。支持的传输协议见11.4。

The OpenAPI 2.0 definitions (Annex A) used in this document are normative. This includes that all defined JSON payloads shall comply with the indicated OpenAPI 2.0 definitions. Annex A contains all of the OpenAPI 2.0 definitions for Resource Types defined in this document.

本文件中使用的OpenAPI 2.0定义（附录A）是规范的。这包括所有定义的JSON有效载荷都必须符合指定的OpenAPI 2.0定义。附录A包含本文件中定义的所有资源类型的OpenAPI 2.0定义。

7.2 Resource 资源

A Resource shall be defined by one or more Resource Type(s) - see Annex A for Resource Type. A request to CREATE a Resource shall specify one or more Resource Types that define that Resource.
资源应由一个或多个资源类型定义——资源类型见附录A。创建资源的请求应指定定义了该资源的一个或多个资源类型。

A Resource is hosted in a Device. A Resource shall have a URI as defined in clause 6. The URI may be assigned by the Authority at the creation of the Resource or may be pre-defined by the definition of the Resource Type. An example Resource representation is depicted in Figure 4.

资源托管在设备中。资源应该具有第6章中定义的URI。URI可以在创建资源时由权威机构分配，也可以由资源类型的定义预先定义。图4描述了一个示例资源表示。

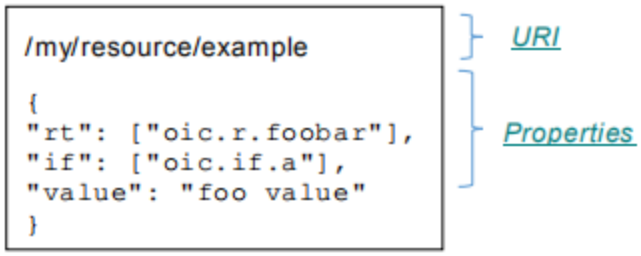

Figure 4 – Example Resource 图4 – 资源示例

Core Resources are the Resources defined in this document to enable functional interactions as defined in clause 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.
核心资源是本文件中定义的资源，用于实现第10章中定义的功能性交互（例如，发现、设备管理等）。在核心资源中，所有设备应支持“/oic/res”、“/oic/p”和“/oic/d”。设备可能支持其他核心资源，这取决于它们支持的功能性交互。

7.3 Property 属性

7.3.1 Introduction 引言

A Property describes an aspect that is exposed through a Resource including meta-information related to that Resource.
属性描述了通过资源进行公开的一个方面，包括与该资源相关的元信息。

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).
属性应该有名称，即属性名和值，也即属性值。属性表示为键值对，其中键是属性名，值是属性值，例如<属性名> = <属性值>。例如，如果“temperature”属性有一个属性名“temp”和一个属性值“30F”，那么该属性表示为“temp=30F”。属性的特定格式取决于编码方案。例如，在JSON中，属性表示为“key”:value（例：“temp”:30）。

In addition, the Property definition shall have a
另外，属性定义应该有

- 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 4.3 or may be a complex data type defined with a schema. The Value Type may define
  值类型——值类型定义了属性值可以取的值。值类型可以是4.3中定义的简单数据类型（例如字符串、布尔型），也可以是模式定义的复杂数据类型。值类型可以定义为
  - 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, the 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.
    值规则定义属性值可能采用的值集的规则。这些规则可以定义值的范围、最小最大值、函数、枚举值集、模式、条件值，甚至是对其他属性值的依赖。规则可用于验证属性值的特定值和标记错误。
- Mandatory - specifies if the Property is mandatory or not for a given Resource Type.
  强制性——指定给定资源类型的属性是否是必需的。
- 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.
  访问模式——指定属性是可读、可写或两者均可。更新等同于写操作。“r”用于读取，“w”用于写入——两者都可以指定。写入并不自动意味着读取。

The definition of a Property may include the following additional information - these items are informative:
属性的定义可以包括以下附加信息——这些信息是资料性的：

- Property Title - a human-friendly name to designate the Property; usually not sent over the wire.
  属性名称——用来指明属性的人性化名称；通常不通过线路发送。
- Description - descriptive text defining the purpose and expected use of this Property.
  描述——定义此属性的目的和预期用途的描述性文本。

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 for all Common Properties are defined in 7.3.2.
通常，属性仅在与其关联的资源中有意义。然而，所有资源都支持的一组基本属性（称为公共属性）在资源之间保持语义不变，即它们的“key=value”对在任何资源中都是相同的。所有公共属性的详细表在7.3.2中定义。

7.3.2 Common Properties 公共属性

7.3.2.1 Introduction 引言

The Common Properties defined in this clause may be specified for all Resources. The following Properties are defined as Common Properties:
可以为所有资源指定此条款中定义的公共属性。以下属性定义为公共属性：

- Resource Type 资源类型
- Resource Interface 资源接口
- Name 名称
- Resource Identity. 资源标识

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 ResourceTypes 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 name 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.
公共属性的名称应当是唯一的，不应被其他属性使用。定义新资源类型时，其非公共属性不能使用现有公共属性的名称（如“rt”、“if”、“n”、“id”）。在定义一个新的“公共属性”时，应该确保它的名称没有被任何其他属性使用。可以通过检查所有现有OCF定义的资源类型的所有属性以验证新公共属性名的唯一性。然而，随着资源类型数量的增加，这可能变得很麻烦。为了防止将来出现这样的名称冲突，OCF可能为公共属性保留特定的命名空间。可能的方法是（1）指定一个特定的前缀（例如“oic”），名称前加此前缀（例如“oic.psize”）仅用于公共属性；（2）由一个或两个字母构成的名字保留为公共属性，其他属性的名称长度应当大于两个字母；（3）公共属性可以嵌套在特定对象中以区分自己。

The ability to UPDATE a Common Property (that supports write as an access mode) is restricted to the "oic.if.rw" (read-write) OCF Interface; thus a Common Property shall be updatable using the read-write OCF Interface if and only if the Property supports write access as defined by the Property definition and the associated schema for the read-write OCF Interface.
更新公共属性的能力（支持将写入作为访问模式）仅限于“oic.if.rw”（读写）OCF接口；因此，当且仅当该属性支持由属性定义和读写OCF接口的关联模式定义的写访问权限时，才可以使用读写OCF接口更新公共属性。

The following Common Properties for all Resources are specified in 7.3.2.2 through 7.3.2.6 and summarized as follows:
7.3.2.2至7.3.2.6中规定了所有资源的以下公共属性，并总结如下：

- Resource Type ("rt") - this Property is used to declare the Resource Type of that Resource. Since a Resource could be defined by more than one Resource Type the Property Value of the Resource Type Property can be used to declare more than one Resource type (see clause7.4.4). See 7.3.2.3 for details.
  资源类型(“rt”)——此属性用于声明该资源的资源类型。由于资源可以由多个资源类型定义，因此可以使用资源类型属性的属性值声明多个资源类型（参见第7.4.4章节）。详见7.3.2.3。
- OCF Interface ("if") - this Property declares the OCF Interfaces supported by the Resource.The Property Value of the OCF Interface Property can be multi-valued and lists all the OCF Interfaces supported. See 7.3.2.4 for details.
  OCF接口(“if”)——此属性声明资源支持的OCF接口。OCF接口属性的属性值可以是多值的，并列出所有受支持的OCF接口。详见7.3.2.4。
- Name ("n") - the Property declares human-readable name assigned to the Resource. See7.3.2.5.
  名称-("n")——属性声明分配给资源的易读名称。见7.3.2.5。
- 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 7.3.2.6 for details.
  资源标识 ("id")：它的属性值应该是特定资源实例的唯一实例标识符（跨越主机服务器范围）。该标识符的编码依赖于设备和实现。详见7.3.2.6。

7.3.2.2 Property Name and Property Value definitions 属性名称和属性值定义

The Property Name and Property Value as used in this document:
本文件中使用的属性名和属性值：

- Property Name - the key in “key=value” pair. Property Name is case sensitive and its data typeis "string". Property names shall contain only letters A to Z, a to z, digits 0 to 9, hyphen, anddot, and shall not begin with a digit.
  属性名——“键=值”对中的键。属性名区分大小写，其数据类型为“字符串”。属性名只能包含字母A到Z、a到z、数字0到9、连字符和点，不能以数字开头。
- Property Value - the value in "key=value" pair. Property Value is case sensitive when its datatype is "string".
  属性值——“key= Value”对中的值。属性值在其数据类型为“字符串”时区分大小写。

7.3.2.3 Resource Type 资源类型

Resource Type Property is specified in 7.4.
资源类型属性在7.4中指定。

7.3.2.4 OCF Interface OCF接口

OCF Interface Property is specified in 7.6.
OCF接口属性在7.6中指定.。

7.3.2.5 Name 名称

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
一个人性化的资源名称，即特定的资源实例名（例：MyLivingRoomLight），名称属性定义在表2中

Table 2 – Name Property Definition
表2 – 名称属性定义

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
Name 名称	"n"	"string"	N/A	N/A	R,W	No	Human understandable name for the Resource. 人类可以理解的资源名称。

The Name Property is read-write unless otherwise restricted by the Resource Type (i.e. theResource Type does not support UPDATE or does not support UPDATE using read-write).
除非受资源类型的限制，名称属性是可读写的（即资源类型不支持更新或不支持使用可读写的更新）。

7.3.2.6 Resource Identity 资源标识

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 as long as the uniqueness constraint is met, noting that an implementation may use a uuid as defined in 4.3. The Resource Identity Property is as defined in Table 3.
资源标识属性应为特定资源实例的唯一实例标识符（跨越主机服务器范围）。只要满足唯一性约束，这个标识符的编码就依赖于设备和实现，4.3中定义的uuid可以作为一个具体的实现。资源标识属性如表3中定义的那样。

Table 3 - Resource Identity Property Definition
表3 – 资源标识属性定义

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
Resource Identity 资源标识	"id"	"string" or uuid	Implementation Dependent 独立实现	N/A	R	No	Unique identifier of the Resource (over all Resources in the Device) 资源的唯一标识符（设备中的所有资源）

7.4 Resource Type 资源类型

7.4.1 Introduction 引言

Resource Type is a class or category of Resources and a Resource is an instance of one or more Resource Types.
资源类型是资源的一个类或类别，资源是一个或多个资源类型的实例。

The Resource Types of a Resource is declared using the Resource Type Common Property as described in 7.3.2.3 or in a Link using the Resource Type Parameter.
资源的资源类型是使用7.3.2.3中描述的资源类型公共属性声明的，或者在使用资源类型参数的链接中声明的。

A Resource Type may either be pre-defined by OCF 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 OpenAPI 2.0 as the specification method for OCF's RESTful interfaces and Resource definitions.
资源类型可以由OCF预先定义，也可以由设备的制造商、最终用户或开发人员（供应商定义的资源类型）自定义。资源类型及其定义细节可以在带外通信（即在文档中）、也可以使用可被API或应用程序下载和使用的元语言来显式定义。OCF采用OpenAPI 2.0作为OCF RESTful接口和资源定义的规范方法。

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 (clause 2 for ABNF and clause 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 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.
每个资源类型都应该用一个资源类型ID来标识，该ID应该使用IETF RFC 6690中控制资源类型属性的需求和ABNF来表示（第2条对ABNF的规定和第3.1条对要求的规定），但需要注意的是，段之间用一个“.”分隔。整个字符串表示资源类型ID。在定义ID时，每个段可以表示适合资源类型的任何语义。例如，每个段可以表示一个命名空间。一旦定义了ID，就应该不透明地使用ID，实现不应该从各个段推断任何信息。字符串“oic”用作资源类型ID定义中的第一个段时，是为OCF定义的资源类型保留的。所有定义的OCF资源类型都要像IETF RFC 6690中描述的那样，在IANA核心参数注册表中注册。

7.4.2 Resource Type Property 资源类型属性

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 as defined in Table 4. The Property Value for the"rt" Common Property shall be the list of Resource Type IDs for the Resource Types used astemplates (i.e., "rt"=<list of Resource Type IDs>).
实例化的或创建的资源应该有一个或多个资源类型，它们是该资源的模板。资源符合的资源类型应使用表4中定义的资源的“rt”公共属性声明。“rt”公共属性的属性值应为用作模板的资源类型的资源类型id列表（即，“rt”=<资源类型id >的列表）。

Table 4 - Resource Type Common Property definition
表4 –资源类型公共属性定义

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
Resource Type 资源类型	"rt"	"array"	Array of strings, conveying Resource Type IDs	N/A	R	Yes	The Property name rt is as described in IETF RFC 6690 属性名rt在IETF RFC 6690中有描述

Resource Types may be explicitly discovered or implicitly shared between the user (i.e. Client) and the host (i.e. Server) of the Resource.
资源类型可以在资源的用户（即客户端）和主机（即服务器）之间显式地发现或隐式地共享。

7.4.3 Resource Type definition 资源类型定义

Resource Type is specified as follows:
资源类型指定如下：

- 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.
  预定义URI(可选)——可以在OCF规范中为特定资源类型指定预定义URI。当资源类型具有预定义URI时，该资源类型的所有实例只能使用预定义的URI。不同资源类型的实例不能使用预定义的URI。
- Resource Type Title (optional) - a human friendly name to designate the Resource Type.
  资源类型标题（可选）——用于指定资源类型的人性化名称。

- Resource Type ID - the value of "rt" Property which identifies the Resource Type, (e.g.,"oic.wk.p").
  资源类型ID ——“rt”属性的值，它标识资源类型（例：“oic.wk.p”）。
- Resource Interfaces - list of the OCF Interfaces that may be supported by the Resource Type.
  资源接口——资源类型可能支持的OCF接口列表。
- 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.
  属性——应用于资源类型的所有属性的定义。资源类型定义应定义属性是强制性的、条件强制性的还是可选的。
- Related Resource Types (optional) - the definition of other Resource Types that may be referenced as part of the Resource Type, applicable to Collections.
  相关资源类型（可选）——可作为资源类型一部分引用的其他资源类型的定义，适用于集合。
- Mime Types (optional) - mime types supported by the Resource including serializations (e.g.,application/cbor, application/json, application/xml).
  Mime类型（可选）——资源支持的Mime类型，包括序列化（例：application/cbor, application/json，application/xml）。

Table 5 and Table 6 provides an example description of an illustrative foobar Resource Type and its associated Properties.
表5和表6提供了一个说明性foobar资源类型及其相关属性的示例描述。

Table 5 – Example foobar Resource Type
表5 – foobar资源类型示例

Pre-defined URI 预定义URI	Resource Type Title 资源类型标题	Resource Type ID (“rt” value) 资源类型ID (“rt” 值)	OCF Interfaces OCF接口	Description 描述	Related Functional Interaction 关系功能联系	M/CR/O
None	"foobar"	"oic.r.foobar"	"oic.if.a"	Example "foobar" Resource 示例"foobar" 资源	Actuation 驱动	O

Table 6 - Example foobar Properties
表6 – foobar属性示例

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
Resource Type 资源类型	"rt"	"array"	N/A	N/A	R	Yes	Resource Type 资源类型
OCF Interface	"if"	"array"	N/A	N/A	R	Yes	OCF Interface OCF接口

OCF接口							
Foo value foo 值	value 值	"array"	N/A	N/A	R	Yes	Foo value Foo 值

For example, an instance of the foobar Resource Type.
例如，foobar资源类型的实例。

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

For example, a schema representation for the foobar Resource Type.
例如，foobar资源类型的模式表示。

```
{
  "$schema": "http://json-schema.org/draft-04/schema",
  "type": "object",
    "properties": {
      "rt": {
      "type": "array",
      "items" : {
        "type" :
        "string",
        "maxLength": 64
      },
      "minItems" : 1,
      "readOnly": true,
      "description": "Resource Type of the Resource"
    },
    "if": {
      "type": "array",
      "items": {
      "type" : "string",
      "enum" : ["oic.if.baseline", "oic.if.ll", "oic.if.b", "oic.if.lb", "oic.if.rw",
"oic.if.r", "oic.if.a", "oic.if.s"]
    },
    "value": {"type": "string"}
  },
  "required": ["rt", "if", "value"]
}
```

7.4.4 Multi-value "rt" Resource 多值"rt"资源

Multi-value "rt" Resource means a Resource with multiple Resource Types where none of the included Resource Types denote a well-known Resource Type (i.e. "oic.wk.<thing>"). 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.
多值“rt”资源是指具有多种资源类型的资源，其中所包含的资源类型中没有一个表示已知的资源类型（即“oic.wk.<thing>”）。这样的资源与多个资源类型相关联，因此具有多个资源类型id的“rt”属性值（例如，“rt”:["oic.r.switch.binary”、“oic.r.light.brightness”]）。rt”属性值中资源类型id的顺序没有意义。例如，“rt”:[“oic.r.switch”.binary”、“oic.r.light.brightness]和“rt”:[“oic.r.light”.brightness”、“oic.r.switch.binary]有相同的含义。

Resource Types for multi-value "rt" Resources shall satisfy the following conditions:
多值“rt”资源的资源类型应满足以下条件：

- 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.
  属性名——除公共属性外，每个资源类型的属性名应该是唯一的（在多值“rt”资源范围内），否则属性语义将会冲突。如果两个资源类型具有具有相同属性“Name”的属性，则不应由这些资源类型组成多值“rt”资源。

A multi-value "rt" Resource satisfies all the requirements for each Resource Type and conforms to the OpenAPI 2.0 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".
多值“rt”资源满足每种资源类型的所有需求，并且符合每种组件资源类型的OpenAPI 2.0定义。因此，多值“rt”资源的强制属性应该是每种资源类型的所有强制属性的联合。例如，具有“rt”的资源的强制属性:
["oic.r.switch.binary”、“oic.r.light.brightness”]是“值”和“亮度”，前者是“oic.r.switch.binary”的强制值，后者表示“oic.r.light.brightness”。

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

多值“rt”资源接口集应该是来自组件资源类型的OCF接口集的联合。响应OCF接口上CRUDN操作的资源表示应该是为该OCF接口定义的模式的联合。多值“rt”资源的默认OCF接口应该是基线OCF接口（“oic.if.baseline”），因为这是资源类型之间唯一得到保证的公共OCF接口。

For clarity if each Resource Type supports the same set of OCF Interfaces, then the resultant multi-value "rt" Resource has that same set of OCF Interfaces with a Default OCF Interface of baseline ("oic.if.baseline").
为了清晰起见，如果每个资源类型都支持相同的OCF接口集，那么得到的多值“rt”资源具有相同的OCF接口集，并且默认OCF接口为baseline（“oic.if.baseline”）。

See 7.9.3 for the handling of query parameters as applied to a multi-value "rt" Resource.
有关应用于多值“rt”资源的查询参数的处理，请参见7.9.3。

7.5 Device Type 设备类型

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 Typeis used in Resource discovery as specified in 11.2.3.
设备类型是设备的一个类。定义的每个设备类型将包括一个设备为该设备类型实现的最小资源类型列表。设备可以公开最小列表之外的其他标准和供应商定义的资源类型。设备类型用于11.2.3中指定的资源发现。

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.
与资源类型一样，设备类型可以在资源类型公共属性中使用，也可以在使用资源类型参数的链接中使用。

A Device Type may either be pre-defined by an ecosystem that builds on this document, or in custom definitions by manufacturers, end users, or developers of Devices (vendor-defined DeviceTypes). Device Types and their definition details may be communicated out of band (like in documentation).
设备类型可以由构建在此文件上的生态系统预先定义，也可以由设备的制造商、最终用户或开发人员（供应商定义的设备类型）自定义。设备类型及其定义细节可以在带外通信（如在文档中）。

Every Device Type shall be identified with a Resource Type ID using the same syntax constraints as a Resource Type.
每个设备类型都应该使用与资源类型相同的语法约束来标识资源类型ID。

7.6 OCF Interface OCF 接口

7.6.1 Introduction 引言

An OCF 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 OCF Interface defines the context

for requests and responses on a Resource. Therefore, the same request to a Resource when targeted to different OCF Interfaces may result in different responses.
OCF接口首先提供资源视图，然后定义该资源视图上允许的请求和响应。因此，OCF接口提供的这个视图定义了资源上的请求和响应的上下文。因此，针对不同OCF接口时对资源的相同请求可能导致不同的响应。

An OCF Interface may be defined by either this document (a Core OCF Interface), manufacturers,end users or developers of Devices (a vendor-defined OCF Interface).
OCF接口可以由本文档（核心OCF接口）、制造商、终端用户或设备开发人员（供应商定义的OCF接口）定义。

The OCF Interface Property lists all the OCF Interfaces the Resource support. All Resources shall have at least one OCF Interface. The Default OCF Interface shall be defined by the Resource Type definition. The Default OCF Interface associated with all OCF-defined Resource Types shall be the supported OCF Interface listed first within the applicable enumeration in the definition of theResource Type (see Annex A for the OCF-defined Resource Types defined in this document). Theapplicable enumeration is in the "parameters" enumeration referenced from the first "get" method in the first "path" in the OpenAPI 2.0 file ("post" method if no "get" exists) for the Resource Type.All Default OCF Interfaces specified in an OCF specification shall be mandatory.
OCF接口属性列出了资源支持的所有OCF接口。所有资源应该至少有一个OCF接口。默认的OCF接口应该由资源类型定义来定义。与所有OCF定义的资源类型相关联的缺省OCF接口应是在资源类型定义的适用枚举中首先列出的受支持的OCF接口（本文件中定义的OCF定义的资源类型请参见附件A）。适用的枚举在资源类型的OpenAPI 2.0文件的第一个“路径”中的第一个“get”方法引用的“parameters”枚举中（如果不存在“get”，则为“post”方法）。OCF规范中指定的所有默认OCF接口都是强制性的。

In addition to any defined OCF Interface in this document, all Resources shall support the baseline OCF Interface ("oic.if.baseline") as defined in 7.6.3.2.
除了本文档中定义的OCF接口外，所有资源都应支持7.6.3.2中定义的OCF基线接口（“oic.if.baseline”）。

See 7.9.4 for the use of queries to enable selection of a specific OCF Interface in a request.
请参阅7.9.4，了解如何使用查询来支持在请求中选择特定的OCF接口。

An OCF Interface may accept more than one media type. An OCF 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 OCF Interface shall have at least one media type.
OCF接口可以接受多种媒体类型。OCF接口可以以多个媒体类型进行响应。可接受的媒体类型可能与响应媒体类型不同。在传输协议中使用适当的头参数指定媒体类型。（注：必须审慎地使用此功能，并允许优化线路上的表示）每个OCF接口应该至少有一个媒体类型。

7.6.2 OCF Interface Property OCF接口属性

The OCF Interfaces supported by a Resource shall be declared using the OCF Interface Common Property (Table 7), e.g., ""if": ["oic.if.ll", "oic.if.baseline"]". The Property Value of an OCF 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 OCF Interface Property Value, is reserved for OCF-defined OCF Interfaces. The OCF Interface Property Value may also be a reference to an authority similar to IANA that may be used to find the definition of an OCF Interface. A Resource Type shall support one or more of the OCF Interfaces defined in 7.6.3.

资源支持的OCF接口应使用OCF接口公共属性（表7）进行声明，如：""if": ["oic.if.ll", "oic.if.baseline"]"。OCF接口属性的属性值应该是小写字符串，段之间用“.”分隔。。字符串“oic”用作OCF接口属性值中的第一个段时，保留给OCF定义的OCF接口。OCF接口属性值也可以是一个类似于IANA的权威的引用，可用于查找OCF接口的定义。资源类型应支持7.6.3中定义的一个或多个OCF接口。

Table 7 - Resource Interface Property definition

表7 – 资源接口属性定义

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
OCF Interface OCF接口	"if"	"array"	Array of strings, conveying OCF Interfaces 字符串数组，包含OCF接口	N/A	R	Yes	Property to declare the OCF Interfaces supported by a Resource. 属性声明资源支持的OCF接口。

7.6.3 OCF Interface methods OCF接口方法

7.6.3.1 Overview 综述

OCF Interface methods shall not violate the defined OpenAPI 2.0 definitions for the Resources as defined in Annex A.

OCF接口方法不得违反附录A中定义的资源OpenAPI 2.0定义。

The defined OCF Interfaces are listed in Table 8:

表8中列出了已定义的OCF接口：

Table 8 OCF standard OCF interfaces

表8 – OCF标准接口

OCF Interface OCF接口	Name 名称	Applicable Operations 适用的操作	Description 描述

Baseline 基线	"oic.if.baseline"	RETRIECE, NOTIFY, UPDATE[3]	The baseline OCF Interface defines a view into all Properties of a Resource including the Meta Properties. 基准OCF接口定义了对资源的所有属性（包括元属性）的视图。 This OCF Interface is used to operate on the full Representation of a Resource. 此OCF接口用于对资源的完整表示形式进行操作。
links list 链接列表	"oic.if.ll"	RETRIECE, NOTIFY,	The links list OCF Interface provides a view into Links in a Collection (Resource). 。链接列表OCF接口提供了指向集合(资源)中的链接的视图。 Since Links represent relationships to other Resources, the links list OCF 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 OCF Interface to allow discovery of Resource hosted on a Device. 由于链接表示与其他资源的关系，所以可以使用链接列表OCF接口来发现与上下文相关的资源。发现是通过检索到这些资源的链接来完成的。例如:核心资源“/oic/res”使用这个OCF接口来发现设备上的资源。
Batch 批量	"oic.if.b"	RETRIEVE, NOTIFY, UPDATE	The batch OCF 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 批处理OCF接口用于同时与一组资源交互。这也消除了客户机首先发现它正在操作的资源的需要——服务器转发请求并聚合响应。
read-only 只读	"oic.if.r"	RETRIEVE, NOTIFY	The read-only OCF Interface exposes the Properties of a Resource that may be read. This OCF Interface does not provide methods to update Properties, so can only be used to read Property Values. 只读OCF接口公开可读资源的属性。此OCF接口不提供更新属性的方法，因此只能用于读取属性值。
read-write 读写	"oic.if.rw"	RETRIEVE, NOTIFY, UPDATE	The read-write OCF Interface exposes only those Properties that may be read from a Resource during a RETRIEVE operation and only those Properties that may be written to a Resource during and[4] UPDATE operation. 读写OCF接口只公开那些可能在检索操作期间从资源读取的属性，也只公开那些可能在更新操作期间写入资源的属性。
actuator 执行器	"oic.if.a"	RETRIEVE, NOTIFY, UPDATE	The actuator OCF Interface is used to read or write the Properties of an actuator Resource. 执行器OCF接口用于读取或写入执行器资源的属性。
sensor 传感器	"oic.if.s"	RETRIEVE, NOTIFY	The sensor OCF Interface is used to read the Properties of a sensor Resource. 传感器OCF接口用于读取传感器资源的属性。
create	"oic.if.create"	CREATE	The create OCF Interface is used to create new

[3] The use of UPDATE with the baseline OCF Interface is not recommended, see clause 7.6.3.2.3.
不建议使用基线OCF接口的UPDATE，参见第7.6.3.2.3条。
[4] 疑似笔误，“and”应为“a”；中文译文按“a”处理。

创建			Resources in a Collection. Both the Resource and the Link pointing to it are created in a single atomic operation. 创建OCF接口用于在集合中创建新资源。资源和指向它的链接都是在单个原子操作中创建的。

7.6.3.2 Baseline OCF Interface 基线OCF 接口

7.6.3.2.1 Overview 综述

The Representation that is visible using the baseline OCF Interface includes all the Properties of the Resource including the Common Properties. The baseline OCF Interface shall be defined for all Resource Types. All Resources shall support the baseline OCF Interface.
使用基线OCF接口可见的表示包括资源的所有属性，包括公共属性。应为所有资源类型定义基线OCF接口。所有资源都应支持基线OCF接口。

7.6.3.2.2 Use of RETRIEVE 检索（RETRIEVE）的使用

The baseline OCF Interface is used when a Client wants to retrieve all Properties of a Resource; that is 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.
当客户端想要检索资源的所有属性时，使用基线OCF接口；也就是说，服务器应使用资源表示进行响应，该资源表示包含该资源的所有实现属性。当服务器无法发送回整个资源表示形式时，它将返回一条错误消息。服务器不应返回部分资源表示。

An example response to a RETRIEVE request using the baseline OCF Interface:
使用基线OCF接口对RETRIEVE请求的响应示例：

```
{
"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 更新（UPDATE）的使用

Support for the UPDATE operation using the baseline OCF Interface should not be provided by a Resource Type. Where a Resource Type needs to support the ability to be UPDATED this should only be supported using one of the other OCF Interfaces defined in Table 8 that supports the UPDATE operation.

资源类型不应该支持使用基线OCF接口的UPDATE操作。如果某个资源类型需要支持更新功能，那么只能使用表8中定义的支持UPDATE操作的其他OCF接口来支持该功能。

If a Resource Type is required to support UPDATE using the baseline OCF Interface, then all Properties of a Resource with the exception of Common Properties may be modified using an UPDATE operation only if the Resource Type defines support for UPDATE using baseline in the applicable OpenAPI 2.0 schema for the Resource Type. If the OCF Interfaces exposed by a Resource in addition to the baseline OCF Interface do not support the UPDATE operation, then UPDATE using the baseline OCF Interface shall not be supported.
如果要求使用基线OCF接口支持UPDATE的资源类型，则仅当资源类型在适用的OpenAPI 2.0中使用基线定义了对UPDATE的支持时，才可以使用UPDATE操作修改资源的所有属性（通用属性除外）资源类型的架构。如果除了基线OCF接口外，由资源公开的OCF接口不支持更新操作，则不支持使用基线OCF接口进行更新。

7.6.3.3 Links list OCF Interface 链接列表OCF接口

7.6.3.3.1 Overview 综述

The Links list OCF Interface is used to provide a view into a Collection, Atomic Measurement, or"/oic.res" Resource. This view shall be an array of all Links for those Resources subject to any applied filtering being applied. The Links list OCF Interface name is "oic.if.ll".
链接列表OCF接口用于提供到集合、原子测量或“/oic.res”资源的视图。这个视图应是那些应用了任一过滤后所得的所有链接的一个数组。链接列表OCF接口名称为“oic.if.ll”。

7.6.3.3.2 Use with RETRIEVE 检索（RETRIEVE）的使用

The RETRIEVE operation is supported with the Links list OCF Interface. A successful RETRIEVE operation shall return a status code indicating success (i.e. "Content") with a payload with the Resource representation as an array of Links. If there are no Links present in a Resource Representation, then an empty array list shall be returned in response to a RETRIEVE operation request.
RETRIEVE操作由链接列表OCF接口支持。一个成功的检索操作应该返回一个状态码来表示成功（例如“Content”），这个状态码与以链接的数组形式表示的资源一起存在于有效载荷。如果资源表示中没有链接，则应返回空数组列表以响应检索操作请求。

An example of a RETRIEVE operation request using the Links list OCF Interface for a Collection is as illustrated:
使用“链接”列表的OCF集合接口进行RETRIEVE操作请求的示例如下：

RETRIEVE /scenes/scene1?if=oic.if.ll

The RETRIEVE operation response will be the array of Links to all Resources in the Collection as illustrated:
RETRIEVE操作的响应将是指向集合中所有资源的链接的数组，如下所示：

```
Response: Content
Payload:
[
  {
    "href": "/the/light/1",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "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"}]
  }
]
```

7.6.3.3.3 Use with NOTIFY 通知的使用

The NOTIFY operation is supported with the Links list OCF Interface. A successful NOTIFY operation shall return a status code indicating success (i.e. "Content") with a payload with the Resource representation as an array of Links. If there are no Links present in a Resource representation, then an empty array list shall be returned in response to a NOTIFY operation request. Future events that change the Resource representation (e.g. UPDATE operation) shall return a status code indicating success (i.e. "Content") with a payload with the newly updated Resource representation as an array of Links.
链接列表OCF接口支持NOTIFY操作。一个成功的通知操作应该返回一个状态码来表示成功（例如“Content”），这个状态码与以链接的数组形式表示的资源一起存在于有效载荷。。如果资源表示中没有链接，则应返回空数组列表以响应NOTIFY操作请求。改变资源表示的未来事件(例如，UPDATE操作)应该返

回一个状态码来表示成功（例如“Content”），这个状态码与以链接的数组形式表示的新更新的资源一起存在于有效载荷。。

An example of a NOTIFY operation request using the Links list OCF Interface for a Collection is as illustrated:
使用集合的链接列表OCF接口的NOTIFY操作请求的示例如下：

```
NOTIFY /scenes/scene1?if=oic.if.ll
```

The NOTIFY operation response will be the array of Links to all Resources in the Collection as illustrated:
NOTIFY操作响应将是指向集合中所有资源的链接数组，如下所示：

```
Response:Content
Payload:
[
  {
    "href": "/the/light/1",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "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"}]
  }
]
```

Later when the M/his/fan/2M Link is removed (e.g., UPDATE operation with the Link remove OCF Interface) the response to the NOTIFY operation request is as illustrated:
稍后移除M/his/fan/2M链路（例如，使用Link remove OCF接口更新操作）时，对NOTIFY操作请求的响应如下所示：

```
Response:Content
Payload:
[
  {
    "href": "/the/light/1",
    "rt": ["oic.r.switch.binary"],
    "if": ["oic.if.a", "oic.if.baseline"],
    "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"}]
  }
]
```

If the result of removing a Link results in no Links being present, then an empty array list shall be sent in a notification. An example of a response with no Links being present is as illustrated:
如果删除链接的结果导致没有链接存在，则应在通知中发送一个空数组列表。下面是一个没有链接的响应示例：

```
Response:   Content
Payload:
[
]
```

7.6.3.3.4 Use with CREATE, UPDATE, and DELETE 创建（CREATE）、更新（UPDATE）和删除（DELETE）的使用

The CREATE, UPDATE and DELETE operations are not allowed by the Links list OCF Interface. Attempts to perform CREATE, UPDATE or DELETE operations using the Links list OCF Interface shall return an appropriate error status code, for example "Method Not Allowed".
链接列表OCF接口不允许CREATE、UPDATE和DELETE操作。尝试使用链表OCF接口执行CREATE、UPDATE或DELETE操作时，将返回适当的错误状态码，例如“方法不允许”。

7.6.3.4 Batch OCF Interface 批量OCF接口

7.6.3.4.1 Overview 综述

The batch OCF Interface is used to interact with a Collection of Resources using a single/same Request. The batch OCF Interface can be used to RETRIEVE or UPDATE the Properties of the linked Resources with a single request.
批量OCF接口用于使用单个/相同的请求与资源集合进行交互。批量OCF接口可通过单个请求用于RETRIEVE或UPDATE链接资源的属性。

7.6.3.4.2 General requirements for realizations of the batch OCF Interface 批量OCF接口实现的一般要求

All realiation[5] of the batch OCF Interface adhere to the following:
所有批量OCF接口的实现均应符合以下要求：

- The batch OCF Interface name is "oic.if.b"
  批量OCF接口名称为“oic.if.b”
- A Collection Resource has linked Resources that are represented as URIs. In the "href" Property of the batch payload the URI shall be fully qualified for remote Resources and a relative reference for local Resources.
  集合资源具有以URI表示的链接资源。在“href”上，批量有效负载的属性URI对于远程资源应该是完全合格的，对于本地资源应该是相对引用。
- The original request is modified to create new requests targeting each of the linked Resources in the Collection by substituting the URI in the original request with the URI of the linked Resource. The payload in the original request is replicated in the payload of the new requests.
  通过将原始请求中的URI替换为链接资源的URI，可以修改原始请求以创建针对集合中每个链接资源的新请求。。原始请求中的有效负载被复制到新请求的有效负载中。
- The requests shall be forwarded assuming use of the Default OCF Interface of the linked Resources.
  假设使用链接资源的默认OCF接口，则应转发请求。
- Requests shall only be forwarded to linked Resources that are identified by relation types "item" or "hosts" ("hosts" is the default relation type value should the "rel" Link Parameter not be present). Requests shall not be forwarded to linked Resources that do not contain the "item" or "hosts" relation type values.
  请求只能被转发到由关系类型“项目”或“主机”标识的链接资源（如果“rel”链接参数不存在，则“主机”是默认的关系类型值）。请求不应转发给不包含“项目”或“主机”关系类型值的链接资源。
- Properties of the Collection Resource itself may be included in payloads using "oic.if.b" OCF Interface by exposing a single Link with the link relation "self" along with "item" within the Collection, and ensuring that Link resolution cannot become an infinite loop due to recursive references. For example, if the Default OCF Interface of the Collection is "oic.if.b", then the

[5] 应为“realizations”的笔误，中文译文按“realizations”处理。

Server might recursively include its batch representation within its batch representation, in an endless loop. See 7.6.3.4.5 for an example of use of a Link containing "rel": ["self","item"] to include Properties of the Collection Resource, along with linked Resources, in "oic.if.b" payloads.
通过在集合中使用链接关系“自身”和“项目”来暴露单个链接，集合资源本身的属性可以使用“oic.if”包含在有效载荷中的"oic.if.b"的OCF接口，确保链接解析不会因为递归引用而变成无限循环。例如，如果集合的默认OCF接口是“oic.if”。然后服务器可能递归地将其批处理表示包含在其批处理表示中，形成一个无限循环。请参见7.6.3.4.5，这是一个使用包含“rel”:[“自身”，“项目”]的链接的示例，该链接将集合资源的属性与链接资源一起包含在“oic.if.b”有效载荷。

- If the Default OCF Interface of a Collection Resource is exposed using the Link relation "self", and the Default OCF Interface contains Properties that expose any Links, those Properties shall not be included in a batch representation which includes the "self" Link.
  如果集合资源的缺省OCF接口使用链接关系“自身”公开，并且默认OCF接口包含公开任何链接的属性，那么这些属性不应包含在包含“self”链接的批处理表示中。
- Any request forwarded to a linked Resource that is a Collection (including a "self" Link reference) shall have the Default OCF Interface of the linked Collection Resource applied.
  任何转发给集合的链接资源（包括“自”链接引用）的请求都应应用该链接集合资源的默认OCF接口。
- All the responses from the linked Resources shall be aggregated into a single Response to the Client. The Server may timeout the response to a time window, the Server may choose any appropriate window based on conditions.
  来自链接资源的所有响应应聚合为对客户端的单个响应。服务器可neng hui 超时响应一个时间窗口，服务器可以根据条件选择任何适当的窗口。
- If a linked Resource cannot process the request, an empty response, i.e. a JSON object with no content ("{}") as the representation for the "rep" Property, or error response should the linked Resource Type provide an error schema or diagnostic payload, shall be returned by the linked Resource. These empty or error responses for all linked Resources that exhibit an error shall be included in the aggregated response to the original Client request. See the example in 7.6.3.4.5.
  如果一个链接的资源不能处理请求，一个空响应，即一个JSON对象，没有内容（“{}”）作为“rep”属性的表示，或错误响应，如果链接的资源类型提供一个错误模式或诊断负载，应由链接的资源返回。所有显示错误的链接资源的这些空响应或错误响应应包含在对原始客户端请求的聚合响应中。参见7.6.3.4.5中的示例。
- If any of the linked Resources returns an error response, the aggregated response sent to the Client shall also indicate an error (e.g. 4.xx in CoAP). If all of the linked Resources return successful responses, the aggregated response shall include the success response code.
  如果任何链接资源返回错误响应，发送给客户端的聚合响应也应指示错误（例如，在CoAP 的4.xx）。如果所有链接的资源都返回成功响应，则聚合的响应应包括成功响应代码。
- The aggregated response shall be an array of objects representing the responses from each linked Resource. Each object in the 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 or array of objects if the linked Resource is itself a Collection using "rep" as the key, e.g. "rep": {〈representation of individual response> }.

聚合的响应应该是表示来自每个链接资源的响应的对象数组。响应中的每个对象应包括至少两个项目:(1)链接资源的URI（完全限定为远程资源,或对当地资源的相对参考）作为“href”: < URI >,(2)个人响应对象或数组的对象如果链接资源本身就是一个集合使用“代表”的关键,如“rep”:{<单个响应>}的表示。

- The Client may choose to restrict the linked Resources to which the request is forwarded by including additional query parameters in the request. The Server should process any additional query parameters in a request that includes "oic.if.b" as selectors for linked Resources that are to be processed by the request.
  客户端可以通过在请求中包含额外的查询参数来限制请求转发到的链接资源。服务器应该处理请求中包含“oic.if”的任何其他查询参数。作为将由请求处理的链接资源的选择器。

7.6.3.4.3 Observability of the batch OCF Interface 批量OCF接口的可观察性

When a Collection supports the ability to be observed using the batch OCF Interface the following apply:
当一个集合支持使用批量OCF接口观察的能力时，应用如下：

- If the Collection Resource is marked as Observable, linked Resources referenced in the Collection may be Observed using the batch OCF Interface. If the Collection Resource is not marked as Observable then the Collection cannot be Observed and Observe requests to the Collection shall be handled as defined for the case where request validation fails in clause 11.3.2.4. The Observe mechanism shall work as defined in 11.3.2 with the Observe request forwarded to each of the linked Resources. All responses to the request shall be aggregated into a single response to the Client using the same representations and status codes as for RETRIEVE operations using the batch OCF Interface.
  如果集合资源被标记为可观察，则可以使用批量OCF接口观察集合中引用的链接资源。如果集合资源没有被标记为可观察的，那么集合就不能被观察，对于集合的观察请求应按照第11.3.2.4条中定义的请求验证失败的情况进行处理。观察机制应按照11.3.2中定义的工作，将观察请求转发给每个链接的资源。对请求的所有响应应使用与使用批OCF接口进行检索操作相同的表示形式和状态代码聚合为对客户端的单个响应。
- Should any one of the Observable linked Resources fail to honour the Observe request the response to the batch Observe request shall also indicate that the entire request was not honoured using the mechanism described in 11.3.2.4.
  如果任何一个被观察到的链接资源没有兑现观察请求，对批量观察请求的响应也应该表明，使用11.3.2.4中描述的机制，整个请求没有兑现。
- If any of the Observable Resources in a request to a Collection using the batch OCF Interface Replies with an error or Observe Cancel, the Observations of all other linked Resources shall be cancelled and the error or Observe Cancel status shall be returned to the Observing Client.
  如果一个使用批量OCF接口的集合请求中的任何一个被观察到的资源返回一个错误或观察取消，所有其他被链接资源的观察将被取消，错误或观察取消状态将被返回到观察客户端。

NOTE Behavior may be different for Links that do network requests vs. local Resources.
注：执行网络请求的链接与执行本地资源的链接的行为可能不同。

- All notifications to the Client that initiated an Observe request using the batch OCF Interface shall use the batch representation for the Collection. This is the aggregation of any individual Observe notifications received by the Device hosting the Collection from the individual Observed Requests that were forwarded to the linked Resources.
  所有使用批量OCF接口发起观察请求的客户端通知均应使用集合的批量表示。这是托管了来自被转发到链接资源的各个观察者请求的集合的设备所接收到的任何单个观察者通知的聚合。
- Linked Resources which are not marked Observable in the Links of a Collection shall not trigger Notifications, but may be included in the response to, and subsequent Notifications resulting from, an Observe request to the batch OCF Interface of a Collection.
  在集合的链接中未标记为Observable的链接资源不应触发通知，但可以包含在对集合的批OCF接口的一个观察请求的响应中，以及由此产生的后续通知中。
- Each notification shall contain the most current values for all of the Linked Resources that would be included if the original Observe request were processed again. The Server hosting the Collection may choose to RETRIEVE all of the linked Resources each time, or may choose to employ caching to avoid retrieving linked Resources on each Notification.
  每个通知应包含所有链接资源的最新值，如果再次处理原始的观察请求，将包含这些资源。承载集合的服务器可以选择每次检索所有链接的资源，也可以选择使用缓存来避免在每次通知时检索链接的资源。
- If a Linked Resource is Observable and has responded with a successful Observe response,the most recently reported value of that Resource is considered to be the most current value and may be reported in all subsequent Notifications.
  如果一个被链接的资源是可观察到的，并且已成功响应了一个观察响应，则该资源最近报告的值将被认为是最近报告的值，并可能在所有后续通知中被报告。
- Links in the Collection should be Observed by using the "oic.if.ll" OCF Interface. A notification shall be sent any time the contents of the "oic.if.ll" OCF Interface representation are changed;that is, if a Link is added, if a Link is removed, or if a Link is updated. Notifications on the"oic.if.ll" OCF Interface shall contain all of the Links in the"oic.if.ll" OCF Interface representation.
  应使用“oic.if.ll”OCF接口来观察集合中的链接。每当“oic.if.ll”OCF接口表示的内容发生更改时，都应发送通知；也就是说，如果一个链接被添加、删除、或更新。"oic.if.ll" OCF 接口的通知应包含"oic.if.ll" OCF接口表示的所有链接。
- Other Properties of the Collection Resource, if present, may be Observed by using the OCF Interfaces defined in the definition for the Resource Type, including using the "oic.if.baseline"OCF Interface.
  如果存在集合资源的其他属性，可以使用资源类型定义中定义的OCF接口（包括”oic.if.baseline”OCF 接口）来观察。

7.6.3.4.4 UPDATE using the batch OCF Interface 使用批量OCF接口的更新

When a Collection supports the ability for the linked Resources to be the subject of the UPDATE operation using the batch OCF Interface the following apply:
当一个集合支持使用批量OCF接口将链接的资源作为更新操作的主体时，应用如下：

- A Client shall perform UPDATE operations using the batch OCF Interface by creating a payload that is similar to a RETRIEVE response payload from a batch OCF Interface request. The Servershall send a separate UPDATE request to each of the linked Resources according to each "href"Property and the corresponding value of the "rep" Property.
  客户端通过创建一个类似于从批量OCF接口请求中检索响应有效负载的有效负载来使用批OCF接口执行更新操作。服务器应根据每个“href”属性和对应的“rep”属性值向每个链接资源发送单独的更新请求。
- Items shall always contain a link-specific "href".
  项目应始终包含一个链接特定的“href”
- An UPDATE received by a Server with an empty "href" shall be rejected with a response indicating an appropriate error (e.g. bad request).
  当一个服务器接收到一个空的“href”更新时，它将被拒绝，并给出一个适当的错误响应（例如，错误的请求）。
- Each linked Resource shall follow the requirements for an UPDATE request may not be supported by the linked Resource. In such cases, writable Properties in the UPDATE operation as defined in clause 8.4.
  每个连结的资源必须符合该连结资源可能不支援的更新要求。在这种情况下，更新操作中的可写属性如第8.4条中定义的那样。
- The UPDATE response shall contain the updated values using the same payload schema as RETRIEVE operations if provided by the linked Resource, along with the appropriate statuscode. The aggregated response payload shall reflect the known state of the updated Properties after the batch update was completed. If no payload is provided by the updated Resource, thenan empty response (i.e. "rep": {}) shall be provided for that Resource.
  更新响应应包含更新的值，使用与检索操作相同的有效载荷模式（如果由链接的资源提供），以及适当的状态码。聚合的响应负载应反映批量更新完成后更新属性的已知状态。如果更新后的资源没有提供有效负载，则为空响应（即“rep”:{}）应提供该资源。
- A Collection shall not support the use of the UPDATE operation to add, modify, or remove Links In an existing Collection using the "oic.if.baseline", "oic.if.rw" or "oic.if.a" OCF Interfaces.
  集合不支持使用“oic.if”更新操作来添加、修改或删除现有集合中的链接。基线”、“oic.if。rw”或“oic.if。一个“OCF接口。
- A Collection shall not support the use of the UPDATE operation using the batch OCF Interface When the Collection contains Links that resolve to Resources that are not hosted on the Device that also hosts the Collection. If such a Collection receives an UPDATE operation, the operation shall be rejected with a response indicating an appropriate error (e.g. method not allowed). If The ability to UPDATE linked remote Resources is desired, the use of the optional scene feature(see

clause 11.6 in [1]) to effect the UPDATE could be utilizied.
当集合包含解析到资源的链接时，集合不支持使用batch OCF接口进行更新操作，这些资源不在同时承载集合的设备上。如果此类集合接收到更新操作，则应拒绝该操作，并给出指示适当错误的响应（例如不允许使用方法）。如果需要更新链接的远程资源，可以使用可选的场景特性（参见[1]中的第11.6条）来实现更新。

7.6.3.4.5 Examples: Batch OCF Interface 实例：批量OCF接口

Note that the examples provided in Table 9 are illustrative and do not include all mandatory schema elements in all cases. It is assumed that the Default OCF Interface for the Resource Type"x.org.example.rt.room" is specified in its Resource Type definition file as "oic.if.rw", which exposes the Properties "x.org.example.colour" and "x.org.example.size".
请注意，表9中提供的示例是说明性的，并不包含所有情况下的所有强制模式元素。假设资源类型“x.org.example.rt.room”的默认OCF接口在其资源类型定义文件中指定为“oic.if”。其中显示属性“x.org.example.color”和“x.org.example.size”。

Table 9 - Batch OCF Interface Example
表9 - 批量OCF接口实例

Resources	/a/room/1
	{
	"rt": "x.org.example.rt.room"],
	"if": ["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],
	"x.org.example.colour": "blue",
	"x.org.example.dimension": "15bx15wx10h",
	"links": [
	{"href": "/a/room/1", "rel": ["self", "item"], "rt":
	["x.org.example.rt.room"], "if":
	["oic.if.rw","oic.if.baseline","oic.if.b","oic.if.ll"],"p": {"bm": 2} },
	{"href": "/the/light/1", "rel": ["item"], "rt":
	["oic.r.switch.binary"],
	"if": ["oic.if.a","oic.if.baseline"], "ins": "11111", "p": {"bm": 2} },
	{"href": "/the/light/2", "rel": ["item"], "rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a" ,"oic.if.baseline"], "ins": "22222", "p": {"bm": 2} },
	{"href": "/my/fan/1", "rel": ["item"], "rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a", "oic.if.baseline"], "ins": "33333", "p": {"bm": 2} },
	{"href": "/his/fan/2", "rel": ["item"], "rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a", "oic.if.baseline"], "ins": "44444", "p": {"bm": 2} },
	{"href": "/the/switches/1", "rel": ["item"], "rt": ["oic.wk.col"],
	"if":["oic.if.ll", "oic.if.b", "oic.if.baseline"], "ins": "55555", "p":
	{"bm": 2}
	}
	]
	}
	
	/the/light/1
	{
	"rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a", "oic.if.baseline"],
	"value": false
	}
	
	/the/light/2
	{
	"rt": ["oic.r.switch.binary"],

	<pre>"if": ["oic.if.a", "oic.if.baseline"],
	"value": true
	}
	
	/my/fan/1
	{
	"rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a", "oic.if.baseline"],
	"value": true
	}
	
	/his/fan/2
	{
	"rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a", "oic.if.baseline"],
	"value": false
	}
	
	/the/switches/1
	{
	{
	"rt": ["oic.wk.col"],
	"if":["oic.if.ll", "oic.if.b", "oic.if.baseline"],
	"links": [
	{
	"href": "/switch-1a",
	"rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a","oic.if.baseline"],
	"p": {"bm": 2}
	}
	{
	"href": "/switch-1b",
	"rt": ["oic.r.switch.binary"],
	"if": ["oic.if.a","oic.if.baseline"],
	"p": {"bm": 2 }
	}
	]
	}</pre>

Use of batch, successful response	Request: GET /a/room/1?if=oic.if.b
	Becomes the following individual request messages issued by the Device in the Client role
	GET /a/room/1 (NOTE: uses the Default OCF Interface as specified for the Collection Resource, in this example oic.if.rw)
	GET /the/light/1 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /the/light/2 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /my/fan/1 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /his/fan/2 (NOTE: Uses the Default OCF Interface as specified for this Resource)
	GET /the/switches/1 (NOTE: Uses the Default OCF Interface for the Collection that is within the Collection)
	Response:
	<pre>[
	{
	"href": "/a/room/1",
	"rep": {"x.org.example.colour": "blue","x.org.example.dimension": "15bx15wx10h"}
	},
	{
	"href": "/the/light/1",
	"rep": {"value": false}
	},
	{
	"href": "/the/light/2",</pre>

<table>
<tr><td></td><td>

<pre>
    "rep": {"value": true}
  },
  {
    "href": "/my/fan/1",
    "rep": {"value": true}
  },
  {
    "href": "/his/fan/1",
    "rep": {"value": false}
  },
  {
    "href":
    "/the/switches/1",
    "rep": [
      {
        "href": "/switch-1a",
        "rt": ["oic.r.switch.binary"],
        "if": ["oic.if.a","oic.if.baseline"],
        "p": {"bm": 2},
        "eps":[
          {"ep": "coaps://[2001:db8:a::b1d4]:55555"}
        ]
      },
      {
        "href": "/switch-1b",
        "rt": ["oic.r.switch.binary"],
        "if": ["oic.if.a","oic.if.baseline"],
        "p": {"bm": 2 },
        "eps":[
          {"ep": "coaps://[2001:db8:a::b1d4]:55555"}
        ]
      }   ]
   }
]
</pre>

</td></tr>
<tr><td>Use of batch, error response</td><td>

Should any of the RETRIEVE requests in the previous example fail then the response includes an empty payload for that Resource instance and an error code is sent. The following example assumes errors from "/my/fan/1" and "/the/switches/1"

Error Response:

<pre>
[
  {
    "href": "/a/room/1",
   "rep":   {"x.org.example.colour":   "blue","x.org.example.dimension":
"15bx15wx10h"}
  },
  {
    "href": "/the/light/1",
   "rep": {"value": false}
  },
  {
    "href": "/the/light/2",
   "rep": {"value": true}
  },
  {
    "href": "/my/fan/1",
    "rep": {}
  },
  {
     "href": "/his/fan/2",
    "rep": {"value": false}
  },
  {
    "href": "/the/switches/1",
</pre>

</td></tr>
</table>

	<pre>"rep": {}
	}
	]</pre>

Use of batch (UPDATE has POST semantics)	<pre>UPDATE
	/a/room/1?if=oic.if.b [
	{
	"href": "",
	"rep": {
	"value": false
	}
	}
	]</pre>
	Since the "href" value in the UPDATE request is empty, the request is forwarded to all Resources in the Collection and becomes:
	<pre>UPDATE /a/room/1 { "value": false }
	UPDATE /the/light/1 { "value": false }
	UPDATE /the/light/2 { "value": false }
	UPDATE /my/fan/1 { "value": false }
	UPDATE /his/fan/2 { "value": false }
	UPDATE /the/switches/1 { "value": false }</pre>
	Response:
	<pre>[
	{
	"href": "/his/light/1",
	"rep": {"value": false}
	},
	{
	"href": "/his/light/2",
	"rep": {"value": false}
	},
	{
	"href": "/my/fan/1",
	"rep": {"value": false}
	},
	{
	"href": "/my/fan/2",
	"rep": {"value": false}
	},
	{
	"href": "/the/switches/2",
	"rep":
	{
	}
	}
	]</pre>
	Since /a/room/1 does not have a "value" Property exposed by its Default OCF Interface, the UPDATE request will be silently ignored and it will not be included in the UPDATE response.
	Since the UPDATE request with the links list OCF Interface is not allowed, an empty payload for the "/the/switches/1" is included in the UPDATE response and an error code is sent.

Use of batch (UPDATE has POST semantics)	<pre>UPDATE
	/a/room/1?if=oic.if.b [
	{
	"href": "/the/light/1",
	"rep": {"value": false}
	}
	},
	{</pre>

```
      "href": "/the/light/2",
      "rep": {
        "value": true
      }
    },
{
     "href": "/a/room/1",
     "rep": {
       "x.org.example.colour": "red"
     }
}
]
```

This turns /the/light/1 off, turns /the/light/2 on, and sets the colour of /a/room/1 to "red".
The response will be same as response for GET /a/room/1?if=oic.if.b with the updated Property values as shown.

```
[
  {
    "href": "/a/room/1",
    "rep":{"x.org.example.colour":
    "red",
      "x.org.example.dimension": "15bx15wx10h"}
 }
 ,
  {
    "href": "/the/light/1",
    "rep": {"value": false}
 }
 ,
  {
    "href": "/the/light/1",
    "rep": {"value": true}
  }
]
```

Example use of additional query parameters to select items by matching Link Parameters.
Turn on light 1 based on the "ins" Link Parameters value of "11111"

```
UPDATE
/a/room/1?if=oic.if.b&ins=11111 [
{
     "href":
     "", "rep":
     {
       "value": false
     }
}
]
```

Similar to the earlier example, "href": "" applies the UPDATE request to all of the Resources in the Collection. Since the additional query parameter ins=11111 selects only links that have a matching "ins" value, only one link is selected. The payload is applied to the target Resource of that link, /the/light/1.
Retrieving the item using the same query parameter:

```
RETRIEVE /a/room/1?if=oic.if.b&ins=11111
```

Response payload:

```
[
  {
   "href": "/the/light/1",
   "rep": {"value": false}
  }
]
```

7.6.3.5 Actuator OCF Interface 执行器OCF接口

The actuator OCF Interface is the OCF Interface for viewing Resources that may be actuated i.e. changes some value within or the state of the entity abstracted by the Resource:
执行器OCF接口是用于查看可能被执行的资源的OCF接口，即改变资源抽象的实体内部或状态的某个值：

- The actuator OCF Interface name shall be "oic.if.a"
  执行器OCF接口名称为“oic.if.a”
- The actuator OCF Interface shall expose in the Resource Representation all mandatory Properties as defined by the applicable OpenAPI 2.0 schema; the actuator OCF Interface may also expose in the Resource Representation optional Properties as defined by the applicable OpenAPI 2.0 schema that are implemented by the target Device.
  执行器OCF接口应在资源表示中公开适用的OpenAPI 2.0模式定义的所有强制属性；执行器OCF接口还可以在资源表示中公开可选属性，这些可选属性由目标设备实现的适用OpenAPI 2.0模式定义。

For example, a "Heater" Resource (for illustration only):
例如，一个“加热器”资源（仅供参考）：

```
/a/act/heater
{
   "rt": ["x.com.acme.gas"],
   "if": ["oic.if.baseline", "oic.if.r", "oic.if.a", "oic.if.s"],
   "x.com.acme.settemp": 10,
   "x.com.acme.currenttemp" : 7
}
```

The actuator OCF Interface with respect to "Heater" Resource (for illustration only):
执行器OCF与“加热器”资源的接口（仅供参考）：

a. Retrieving values of an actuator. 检索执行器的值。

```
Request: RETRIEVE /a/act/heater?if="oic.if.a"
Response:Content
Payload:
{
  "x.com.acme.settemp": 10,
  "x.com.acme.currenttemp" : 7
}
```

b. Correct use of actuator OCF Interface. 执行器OCF接口的正确使用

```
Request: UPDATE /a/act/heater?if="oic.if.a"
  {
  "x.com.acme.settemp": 20
```

```
}
Response: Changed Payload:
{
"x.com.acme.settemp": 20
}
```

c. Incorrect use of actuator OCF Interface. 执行器OCF接口的错误使用。

```
Request: UPDATE /a/act/heater?if="oic.if.a"
{
"if": ["oic.if.s"]     this is visible through baseline OCF Interface
}
Response:Bad Request Payload:
{
}
```

- A RETRIEVE request using this OCF Interface shall return the Representation for this Resource subject to any query and filter parameters that may also exist.
  使用此OCF接口的检索请求应根据可能存在的任何查询和筛选参数返回此资源的表示形式。
- An UPDATE request using this OCF Interface shall provide a payload or body that contains the Properties that will be updated on the target Resource.
  使用此OCF接口的更新请求应提供包含将在目标资源上更新的属性的有效负载或主体。

7.6.3.6 Sensor OCF Interface 传感器OCF接口

The sensor OCF Interface is the OCF Interface for retrieving measured, sensed or capability specific information from a Resource that senses:
传感器OCF接口是用于从感知资源中检索测量、感知或能力特定信息的OCF接口：

- The sensor OCF Interface name shall be "oic.if.s".
  传感器OCF接口名称为“oic.if.s”。
- The sensor OCF Interface shall expose in the Resource Representation all mandatory Properties as defined by the applicable OpenAPI 2.0 schema; the sensor OCF Interface may also expose in the Resource Representation optional Properties as defined by the applicable OpenAPI 2.0 schema that are implemented by the target Device.
  传感器OCF接口应在资源表示中公开适用的OpenAPI 2.0模式定义的所有强制属性；传感器OCF接口还可以在由目标设备实现的适用OpenAPI 2.0模式定义的资源表示可选属性中公开。
- A RETRIEVE request using this OCF Interface shall return this representation for the Resource subject to any query and filter parameters that may also exist.
  使用此OCF接口的检索请求应返回资源主题的此表示形式，以满足可能存在的任何查询和筛选参数。

NOTE: The example here is with respect to retrieving values of a sensor
注：这里的示例是关于检索传感器的值的

```
Request: RETRIEVE /a/act/heater?if="oic.if.s"

Response: Content
Payload:
{
"x.com.acme.currenttemp": 7
}
```

Incorrect use of the sensor. 错误使用传感器

```
Request: UPDATE /a/act/heater?if="oic.if.s"   UPDATE is not allowed
{"x.com.acme.settemp": 20       this is possible through actuator OCF Interface
}
Response: Bad Request
Payload:
{
}
```

Another incorrect use of the sensor. 传感器的另一个错误使用。

```
Request: UPDATE /a/act/heater?if="oic.if.s"   UPDATE is not allowed
{
"x.com.acme.currenttemp": 15       this is not possible to be updated
}
Response: Bad Request
Payload:
{
}
```

7.6.3.7 Read-only OCF Interface 只读OCF接口

The read-only OCF Interface exposes only the Properties that may be read. This includesProperties that may be read-only, read-write but not Properties that are write-only or set-only. Theapplicable operations that can be applied to a Resource are only RETRIEVE and NOTIFY. Anattempt by a Client to apply a method other than RETRIEVE or NOTIFY to a Resource shall berejected with an error response code.
只读OCF接口只公开可读的属性。这包括只读、读写属性，但不包括只写或只集的属性。可以应用于资源的适用操作只有检索和通知。客户端尝试应用检索或通知资源以外的方法时，应使用错误响应代码予以拒绝。

The read-only OCF Interface with respect to "Heater" Resource (for illustration only):

```
Request: RETRIEVE /a/act/heater?if="oic.if.r"
Response: Content
Payload:
{
"x.com.acme.settemp": 10,
```

```
"x.com.acme.currenttemp" : 7
}
```

7.6.3.8 Read-write OCF Interface 读写OCF接口

The read-write OCF Interface is a generic OCF Interface to support reading and setting Properties in a Resource. The applicable methods that can be applied to a Resource are only RETRIEVE,NOTIFY, and UPDATE. For the RETRIEVE and NOTIFY operations, the behavior is the same as for the "oic.if.r" OCF Interface defined in 7.6.3.7. For the UPDATE operation, read-only Properties(i.e. Properties tagged with "read-Only=True" in the OpenAPI 2.0 definition) shall not be in theUPDATE payload. An attempt by a Client to apply a method other than RETRIEVE, NOTIFY, orUPDATE to a Resource shall be rejected with an error response code.
读写OCF接口是一个通用的OCF接口，支持读取和设置资源中的属性。可以应用于资源的适用方法只有检索、通知和更新。对于检索和通知操作，其行为与7.6.3.7中定义的"oic.if.r"　OCF接口的行为相同。对于更新操作，更新有效负载中不应包含只读属性（即在OpenAPI　2.0定义中标记为“只读=True”的属性）。客户端尝试应用除检索、通知或更新资源之外的方法时，应使用错误响应代码予以拒绝。

For example, a "Grinder" Resource (for illustration only):

```
/a/mygrinder
{
"rt": ["oic.r.grinder"],
"if": ["oic.if.rw", "oic.if.baseline"],
"coarseness": 10,
"remaining": 50
}
```

The read-write OCF Interface with respect to “Grinder" Resource (for illustration only):

a. Retrieving the value with read-write OCF Interface

```
 Request: RETRIEVE /a/mygrinder?if="oic.if.rw"
 Response: Content
 Payload:
{
  "coarseness": 10,
  "remaining": 50
}
```

b. Updating the value with read-write OCF Interface

```
Request: UPDATE /a/mygrinder?if="oic.if.rw"
{
  "coarseness": 20
}
  Response: Changed
```

```
  Payload:
{
  "coarseness": 20
}
```

7.6.3.9 Create OCF Interface 创建 OCF接口

7.6.3.9.1 Overview 综述

The create OCF Interface is used to create Resource instances in a Collection. An instance of aResource and the Link pointing to the Resource are created together, atomically, according to aClient-supplied representation. The create OCF Interface name is "oic.if.create". A Collection whichexposes the "oic.if.create" OCF Interface shall expose the "rts" Property (see clause 7.8.2.8) withall Resource Types that can be hosted with the Collection. If a Client attempts to create a ResourceType which is not supported by the Collection, the Server shall return an appropriate error statuscode, for example "Bad Request". Successful CREATE operations shall return a success code, i.e."Created". The IDD for all allowed Resource Types that may be created shall adhere toIntrospection for dynamic Resources (see clause 11.4).
创建OCF接口用于在集合中创建资源实例。资源的实例和指向资源的链接是根据客户端提供的表示一起自动创建的。创建OCF接口名是“oc.if.create”。一个暴露“oic.if”的集合。创建“OCF接口”时，应将“rts”属性（参见第7.8.2.8条）与可以托管在集合中的所有资源类型一起公开。如果客户端试图创建一个集合不支持的资源类型，服务器将返回一个适当的错误状态代码，例如“Bad Request”。成功的创建操作将返回一个成功代码，即“创建”。所有可创建的允许资源类型的IDD都应遵循对动态资源的内省（参见第11.4条）。

7.6.3.9.2 Data format for CREATE 创建数据格式

The data format for the create OCF Interface is similar to the data format for the batch OCF Interface. The create OCF Interface format consists of a set of Link Parameters and a "rep"Parameter which contains a representation for the created Resource.
创建OCF接口的数据格式类似于批量 OCF接口的数据格式。创建OCF接口格式由一组链接参数和一个“rep”参数组成，后者包含所创建资源的表示。

The representation supplied for the Link pointing to the newly created Resource shall contain at least the "rt" and "if" Link Parameters.
为指向新创建资源的链接提供的表示应该至少包含“rt”和“if”链接参数。

The Link Parameter "p" should be included in representations supplied for all created Resources.If the "Discoverable" bit is set, then the supplied Link representation shall be exposed in "/oic/res"of the Device on which the Resource is being created. The Link Parameters representation in the"/oic/res" Resource does not have to mirror the Link Parameters in the Collection of the created Resource (e.g., "ins" Parameter).
链接参数“p”应该包含在为所有创建的资源提供的表示中。如果设置了“可发现”位，则提供的链接表示形式

应在创建资源的设备的“/oic/res”中公开。“/oic/res”资源中的链接参数表示不必反映所创建资源集合中的链接参数（例如，“ins”参数）。

Creating a discoverable Resource is the only way to add a Link to "/oic/res".
创建可发现的资源是向“/oic/res”添加链接的惟一方法。

If the "p" Parameter is not included, the Server shall create the Resource using the default settings of not discoverable, and not observable.
如果不包含“p”参数，则服务器应使用“不可发现”和“不可观察”的默认设置创建资源。

The representation supplied for a created Resource in the value of the "rep" Parameter shall contain all mandatory Properties required by the Resource Type to be created excluding the Common Properties "rt" and "if" as they are already included in the create payload.
在“rep”参数的值中为创建的资源提供的表示应包含要创建的资源类型所需的所有强制属性，但公共属性“rt”和“if”除外，因为它们已经包含在创建有效负载中。

Note that the "rt" and "if" Property Values are created from the supplied Link Parameters of the Resource creation payload.
请注意，“rt”和“if”属性值是从资源创建有效负载提供的链接参数创建的。

If the supplied representation does not contain all of the required Properties and Link Parameters, the Server shall return an appropriate error status code, for example "Bad Request".
如果提供的表示不包含所有必需的属性和链接参数，服务器应返回适当的错误状态代码，例如“Bad Request”。

An example of the create OCF Interface payload is as illustrated:
创建OCF接口负载的一个例子如下所示：

```
{
  "rt": ["oic.r.temperature"],
  "if": ["oic.if.a","oic.if.baseline"],
  "p": {"bm":3},
  "rep": {
       "temperature": 20
   }
}
```

The representation returned when a Resource is successfully created shall contain the "href", "if", and "rt" Link Parameters and all other Link Parameters that were included in the CREATE operation. In addition, the "rep" Link Parameter shall include all Resource Properties as well as the "rt" and "if" Link Parameters supplied in the CREATE operation. The Server may include additional Link Parameters and Properties in the created Resource as required by the application-specific Resource Type. The Server shall assign an "ins" value to each created Link and shall include the "ins" Parameter in the representation of each created Link as illustrated in the Collection that the Link of the created Resource was created within:

成功创建资源时返回的表示应包含“href”、“if”和“rt”链接参数以及CREATE操作中包含的所有其他链接参数。此外，“rep”链接参数应包括创建操作中提供的所有资源属性以及“rt”和“if”链接参数。根据特定于应用程序的资源类型的要求，服务器可以在创建的资源中包含附加的链接参数和属性。服务器应为每个已创建的链接分配一个“ins”值，并应将“ins”参数包含在每个已创建链接的表示中，如集合中所示，所创建资源的链接是在：

```
{
    "href": "/3755f3ac",
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a","oic.if.baseline"],
    "ins": 39724818,
    "p": {"bm":3},
    "rep": {
        "rt": ["oic.r.temperature"],
        "if": ["oic.if.a","oic.if.baseline"],
        "temperature": 20
    }
}
```

The Link Parameters representation in the "/oic/res" Resource, if the created Resource is discoverable, may not mirror exactly all the Link Parameters added in the Collection; except it shall expose at a minimum the mandatory Properties of the Link (i.e., "rt", "if", and "href") of the created Resource.
“/oic/res”资源中的链接参数表示，如果可以发现创建的资源，则可能不能完全反映集合中添加的所有链接参数；但它至少应公开该链接的强制性属性（即、“rt”、“if”和“href”）。

7.6.3.9.3 Use with CREATE 创建的使用

The CREATE operation shall be sent to the URI of the Collection in which the Resource is to be created. The query string "?if=oic.if.create" shall be included in all CREATE operations.
创建操作应发送到要在其中创建资源的集合的URI。查询字符串"?if= oicif应包括在所有创建操作中。

The Server shall generate a URI for the created Resource and include the URI in the "href" Parameter of the created Link.
服务器将为创建的资源生成一个URI，并将URI包含在创建的链接的“href”参数中。

When a Server successfully completes a CREATE operation using the "oic.if.create" OCF Interface addressing a Collection, the Server shall automatically modify the ACL Resource to provide initial authorizations for accessing for the newly created Resource according to ISO/IEC 30118-2:2018.

An example performing a CREATE operation is as illustrated:
一个执行创建操作的例子如下所示：

```
CREATE /scenes/scene1?if=oic.if.create
{
```

```
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a","oic.if.baseline"],
    "p": {"bm":3},
    "rep":
       { "temperature": 20
       }
}
Response: Created
Payload:
{
    "href": "/3755f3ac",
    "ins": 39724818,
    "rt": ["oic.r.temperature"],
    "if": ["oic.if.a","oic.if.baseline"],
    "p": {"bm":3},
    "rep": {
       "rt": ["oic.r.temperature"],
       "if": ["oic.if.a","oic.if.baseline"],
       "temperature": 20
    }
}
```

7.6.3.9.4 Use with UPDATE and DELETE 更新和删除的使用

The UPDATE and DELETE operations are not allowed by the create OCF Interface. Attempts toperform UPDATE or DELETE operations using the create OCF Interface shall return an appropriateerror status code, for example "Method Not Allowed", unless the UPDATE and CREATE operationsmap to the same transport binding method (e.g., CoAP with the POST method). In that situationwhere the UPDATE and CREATE operations map to the same transport binding method, this shallbe processed as a CREATE operation according to clause 7.6.3.9.3.

创建 OCF接口不允许执行UPDATE和DELETE操作。尝试使用创建的OCF接口执行UPDATE或DELETE操作应返回适当的错误状态代码，例如“不允许的方法”，除非UPDATE和CREATE操作映射到相同的传输绑定方法（例如CoAP和POST方法）。在UPDATE和CREATE操作映射到相同的传输绑定方法的情况下，应根据7.6.3.9.3节将其作为CREATE操作处理。

7.7 Resource representation 资源表示

Resource representation captures the state of a Resource at a particular time. The Resource representation is exchanged in the request and response interactions with a Resource. A Resource representation may be used to retrieve or update the state of a Resource.

资源表示捕获特定时间内资源的状态。在与资源的请求和响应交互中交换资源表示。资源表示形式可用于检索或更新资源的状态。

The Resource representation shall not be manipulated by the data connectivity protocols and technologies (e.g., CoAP, UDP/IP or BLE).
资源表示不能被数据连接协议和技术操作（例如， CoAP, UDP/IP 或者BLE）。

7.8 Structure 结构

7.8.1 Introduction 引言

In many scenarios and contexts, the Resources may have either an implicit or explicit structure between them. This may be achieved through the use of Collection (7.8.3) and Atomic Measurement (7.8.4) Resources.
在许多场景和上下文中，资源之间可能具有隐式或显式结构。这可以通过使用收集（7.8.3）和原子测量（7.8.4）资源来实现。

7.8.2 Resource relationships (Links) 资源关系（链接）

7.8.2.1 Introduction 引言

Resource relationships are expressed as Links. A Link is a hyperlink, which defines a typed connection between two Resources. Hyperlinks, or web links, have the following components as defined in IETF RFC 8288:
资源关系表示为链接。链接是一个超链接，它定义了两个资源之间的类型化连接。超链接或web链接具有IETF RFC 8288中定义的以下组件：

- Link context (URI reference) as defined in 7.8.2.2
  在7.8.2.2中定义的链接上下文（URI引用）
- Link relation type as defined in 7.8.2.3
  在7.8.2.3中定义的链接关系类型
- Link target (URI reference) as defined in 7.8.2.4
  在7.8.2.4中定义的链接目标（URI引用）
- Link target attributes as defined in 7.8.2.5
  链接7.8.2.5中定义的目标属性

The Link context is the Resource with which the Link is associated. A Link is viewed as a statement of the form "(Link context) has a (Link relation type) to a Resource at (Link target), which has (Linktarget attributes)" as per IETF RFC 8288 clause 2.
链接上下文是与链接相关联的资源。根据IETF RFC 8288条款2，链接被视为“（链接上下文）具有（链接关系类型）到（链接目标）的资源的语句，该资源具有（链接目标属性）”。

To paraphrase, the Link target is related to the Link context according to the Link relation type.Additionally, the Link target attributes make semantic statements about the Link target, to identify the content type, physical location, etc.
换句话说，链接目标根据链接关系类型与链接上下文相关。此外，链接目标属性对链接目标做出语义声明，以识别内容类型、物理位置等。

Links conform to the definitions in IETF RFC 8288, with an example JSON serialization with associated Link Parameters as illustrated:
链接符合IETF RFC 8288中的定义，并提供了一个带有相关链接参数的JSON序列化示例，如下所示：

```
{
      "anchor": "/some/ocf/resource",              //Link context, optional
      "rel": ["hosts"],                            //Link relation Type, optional
      "href": "/some/other/ocf/resource",          //Link target, required
      "p": {"bm": 3},                             // Link target attributes, optional
      "if": ["oic.if.baseline"],                  // Link target attributes, required
      "rt": ["oic.r.sensor"]                     // Link target attributes, required
}
```

Additional items in the Link may be made mandatory based on the use of the Links in different contexts (e.g. in Collections, in discovery, in bridging etc.). The OpenAPI 2.0 file for the Linkpayload is detailed in Annex A.
根据链接在不同上下文中（如集合、发现、桥接等）的使用情况，可以强制添加链接中的附加项。链接有效负载的OpenAPI 2.0文件在附件A中有详细说明。

Another example of a Link is as illustrated:
另一个链接的例子如下：

```
{"href": "/switch", "rt": ["oic.r.switch.binary"], "if": ["oic.if.a",
"oic.if.baseline"], "p": {"bm": 3}, "rel": "item"}
```

7.8.2.2 Link context 链接内容

The Link context is defined in the Link using the "anchor" Parameter. If the Link doesn't contain an"anchor" Parameter, the Link context shall be the Resource from which the Link was retrieved.
链接上下文在链接中使用“锚”参数定义。如果链接不包含“锚”参数，则链接上下文应是检索链接的资源。

7.8.2.3 Link relation type 链接关系类型

The Link relation type conveys the semantics of the Link. The Link relation type is defined in the Link using the "rel" Parameter. If the Link doesn't contain a "rel" Parameter, the Link relation type shall be assumed to have the default value "hosts", which means that the Resource at the Linktarget is "hosted" by the Resource at the Link context. The set of Link relation types to be used to describe various

relationships between Resources are as listed:
链接关系类型传递链接的语义。链接关系类型是在链接中使用“rel”参数定义的。如果链接不包含“rel”参数，则应假定链接关系类型具有默认值“hosts”，即链接目标上的资源由链接上下文上的资源“承载”。用于描述资源之间各种关系的一组链接关系类型如下：

- "hosts"
    - The Link target points to a Resource that is hosted at the Link context. This Link relation type indicates that the Resource is allowed to be included in the batch representations of the Link target. This Link relation type is defined by IETF RFC 6690.
      链接目标指向位于链接上下文的资源。此链接关系类型指示允许将资源包含在链接目标的批处理表示中。该链接关系类型由IETF RFC 6690定义。
- "self"
    - The Link refers to the Link context, which allows a Link to describe the Resource at the Linq Context, which is to say that the Link can describe the Collection or Atomic Measurement Resource that the Link is retrieved from. The Link target points to the Link context, and the Link target attributes describe the Link context. This Link relation type is defined by IETF RFC 4287.
      链接指的是链接上下文，它允许一个链接描述链接上下文中的资源，也就是说，该链接可以描述该链接所检索到的集合或原子测量资源。链接目标指向链接上下文，链接目标属性描述链接上下文。这个链接关系类型由IETF RFC 4287定义。
- "item"
    - The Link target points to a Resource that is a member of the Collection or Atomic Measurement at the Link context, which might not specifically be hosted by the Collection Or Atomic Measurement Resource, and is allowed to be contained in batch representations of the Collection or Atomic Measurement. An example is using "rel": "item" to declare that the Properties of the Collection or Atomic Measurement Resource itself should be included in a batch representation of the Collection or Atomic Measurement. This Link relation type is defined by IETF RFC 6573.
      链接目标指向资源集合的成员或原子测量在上下文的联系,也可能没有特别举办的集合或原子测量资源,并允许包含在集合或原子的批表示测量。一个例子是使用“rel”:“item”来声明集合或原子测量资源本身的属性应该包含在集合或原子测量的批处理表示中。该链接关系类型由IETF RFC 6573定义。

All of these Link relation types are registered in the IANA Registry for Link relations types defined in IANA Link Relations. Other Link relation types may be included in Links, provided that they conform to the requirements in IETF RFC 8288. Other Link relation types may be defined for features contained in other specifications and may not be included in what is defined in this clause.The presence of Link relation types not defined in this document does not affect the processing of Link relation types defined in this document.
所有这些链接关系类型都在IANA注册中心注册IANA链接关系中定义的链接关系类型。其他链接关系类型可以包含在链接中，只要它们符合IETF RFC 8288中的要求。可以为其他规范中包含的功能定义其他链接关

系类型，也可以不包括在本条款中定义的功能中。本文档中未定义的链接关系类型的存在并不影响本文档中定义的链接关系类型的处理。

When there is more than one Link relation type value in a Link, all of the values apply to describe the relationship between the Link context and the Link target. A Link with multiple Link relation type values is equivalent to a set of Links having the same Link context and Link target, each having one of the Link relation values.
当一个链接中有多个链接关系类型值时，所有的值都用于描述链接上下文和链接目标之间的关系。具有多个链接关系类型值的链接等价于具有相同链接上下文和链接目标的一组链接，每个链接具有一个链接关系值。

7.8.2.4 Link target 链接目标

The Link target is a URI reference to a Resource using the "href" Parameter.
链接目标是一个使用“href”参数的资源的URI引用。

7.8.2.5 Parameters for Link target attributes 链接目标分布参数

7.8.2.5.1 Introduction 引言

Link target attributes are specialisations of Link Parameters. Table 10 lists all the Link target attributes defined in this document.
链接目标属性是链接参数的专门化。表10列出了在这个文档中定义的所有链接目标属性。

Table 10 - Link target attributes list
表10-链接目标分布列表

Parameter title 参数标题	Parametername 参数名称	Mandatory 强制性	Description 描述
Device ID 设备ID	"di"	No	Defined in clause 7.8.2.5.5 在第7.8.2.5.5条中定义
OCF Endpoint information OCF端点信息	"eps"	Yes	Defined in clause7.8.2.5.6 在第7.8.2.5.6条中定义
OCF Interface OCF接口	"if"	Yes	Defined in clause 7.6 在第7.6条中定义
Link instance 链接实例	"ins"	No	Defined in clause 7.8.2.5.2 在第7.8.2.5.2条中定义
Policy 策略	"p"	No	Defined in clause7.8.2.5.3 在第7.8.2.5.3条中定义
Resource Type 资源类型	"rt"	Yes	Defined in clause 7.4 在第 7.4条中定义
Media type 媒介类型	"type"	No	Defined in clause 7.8.2.5.4 在第7.8.2.5.4条中定义

Position description Semantic Tag 位置描述语义标签	"tag-pos-desc"	No	Defined in clause11.5.2.1.2 在第11.5.2.1.2条中定义
Relative position Semantic Tag 相对位置语义标签	"tag-pos-pos"	No	Defined in clause 11.5.2.1.3 在第11.5.2.1.3条中定义
Function description Semantic Tag 功能描述语义标签	"tag-func-desc"	No	Defined in clause11.5.2.2.2 在第11.5.2.2.2条中定义

Note: Other Link target attributes may to defined for features in other specifications and may not be included in this table.
注：其他链接的目标属性可能在其他规范中定义，也可能不包括在本表中。

7.8.2.5.2 "ins" or Link instance Parameter "ins"或链接实例参数

The "ins" Parameter identifies a particular Link instance in a list of Links. The "ins" Parameter may be used to modify or delete a specific Link in a list of Links. The value of the "ins" Parameter is setat instantiation of the Link by the OCF Device (Server) that is hosting the list of Links - once it has been set, the "ins" Parameter shall not be modified for as long as the Link is a member of that list.
“ins”参数标识链接列表中的特定链接实例。“ins”参数可用于修改或删除链接列表中的特定链接。“ins”参数的值是由承载链接列表的OCF设备（服务器）在链接实例化时设置的-一旦设置完毕，只要链接是该列表的成员，“ins”参数就不能修改。

7.8.2.5.3 "p" or policy Parameter "p"或策略参数

The policy Parameter defines various rules for correctly accessing a Resource referenced by a target URI. The policy rules are configured by a set of key-value pairs.
策略参数为正确访问目标URI引用的资源定义了各种规则。策略规则由一组键值对配置。

The policy Parameter "p" is defined by:
策略参数“p”定义为：

- "bm" key: The "bm" key corresponds to an integer value that is interpreted as an 8-bit bitmask.Each bit in the bitmask corresponds to a specific policy rule. The rules are specified for "bm" in Table 11:
  “bm”键:“bm”键对应一个整数值，该整数值被解释为一个8位的位掩码。位掩码中的每个位都对应一个特定的策略规则。有关“bm”的规则载于表11:

Table 11 - "bm" Property definition
表11 - "bm"属性定义

Bit Position 数位位置	Policy rule 策略规则	Comment 评论

Bit 0 (the LSB)	Discoverable 可观察	The discoverable rule defines whether the Link is to be included in the Resource discovery message via "/oic/res". If the Link is to be included in the Resource discovery message, then "p" shall include the "bm" key and set the discoverable bit to value 1. 可发现规则定义链接是否通过“/oic/res”包含在资源发现消息中。如果链接包含在资源发现消息中，则“p”应包含“bm”键，并将可发现的位设置为值1。 If the Link is NOT to be included in the Resource discovery message, then "p" shall either include the "bm" key and set the discoverable bit to value 0 or omit the "bm" key entirely. 如果该链接不包含在资源发现消息中，则“p”应该包括“bm”键并将可发现的位设置为0或完全省略“bm”键。
Bit 1 (2nd LSB)	Discoverable” 可观察	The Observable rule defines whether the Resource referenced by the target URI supports the NOTIFY operation. With the self-link, i.e. the Link with "rel" value of "self", "/oic/res" can have a Link with the target URI of "/oic/res" and indicate itself Observable. The "self" is defined by IETF RFC 4287 and registered in the IANA Registry for "rel" value defined at IANA Link Relations. 可观察规则定义目标URI引用的资源是否支持NOTIFY操作。通过自链接，即具有“self”的“rel”值的链接，“/oic/res”可以与“/oic/res”的目标URI链接，并表示自己是可观察的。“self”由IETF RFC 4287 定义，并在IANA注册表中注册 IANA Link Relations.中定义的“rel”值。 If the Resource supports the NOTIFY operation, then "p" shall include the "bm" key and set the Observable bit to value 1. 如果资源支持NOTIFY操作，则“p”应包含“bm”密钥，并将Observable位设置为值1。 If the Resource does NOT support the NOTIFY operation, then "p" shall either include the "bm" key and set the Observable bit to value 0 or omit the "bm" key entirely. 如果资源不支持NOTIFY操作，则“p”应该包括“bm”键并将可观察位设置为0，或者完全省略“bm”键。
Bits 2-7	--	Reserved for future use. All reserved bits in "bm" shall be set to value 0. 保留供将来使用。“bm”中的所有保留位都应设置为0。

NOTE If all the bits in "bm" are defined to value 0, then the "bm" key may be omitted entirely from "p" as an efficiency measure. However, if any bit is set to value 1, then "bm" shall be included in "p" and all the bits shall be defined appropriately.
注：如果“bm”中的所有位都被定义为值0，那么作为一种效率度量，“p”中可以完全省略“bm”键。但是，如果将任何一位设置为值1，则“p”中应包含“bm”，所有的位应适当定义。

- In a payload sent in response to a request that includes an OCF-Accept-Content-Format-Version option the "eps" Parameter shall provide the information for an encrypted connection.
  在响应包含OCF-Accept-Content-Format-Version选项的请求时发送的有效载荷中，“eps”参数应提供用于加密连接的信息。
- Note that access to the Resource is controlled by the ACL for the Resource. A successful encrypted connection does not ensure that the requested action will succeed. See ISO/IEC 30118-2:2018 clause 12 for more information.
  注：对资源的访问是由资源的ACL控制的。成功的加密连接不能确保请求的操作成功。更多信息参见ISO/IEC 30118-2:20 2018第12章节。

This shows the policy Parameter for a Resource that is discoverable but not Observable.
这列出了可发现但不可观察的资源的策略参数。

```
"p": {"bm": 1}
```

This shows a self-link, i.e. the "/oic/res" Link in itself that is discoverable and Observable.
这列出了一个自我链接，即“/oic/res”链接本身是可发现和可观察的。

```
{
    "href": "/oic/res",
    "rel": "self",
    "rt": ["oic.wk.res"],
    "if": ["oic.if.ll", "oic.if.baseline"],
    "p": {"bm": 3}
}
```

7.8.2.5.4 "type" or media type Parameter "type"或媒介类型参数

The "type" Parameter may be used to specify the various media types that are supported by a specific target Resource. The default type of "application/vnd.ocf+cbor" shall be used when the"type" element is omitted. Once a Client discovers this information for each Resource, it may use one of the available representations in the appropriate header field of the Request or Response.
“type”参数可用于指定特定目标资源支持的各种媒体类型。当“type”元素被省略时，应使用"application/vnd.ocf+cbor"默认类型。一旦客户端发现每个资源的此信息，它可能会在请求或响应的适当头字段中使用一个可用的表示。

7.8.2.5.5 "di" or Device ID Parameter "di"或设备ID 参数

The "di" Parameter specifies the Device ID of the Device that hosts the target Resource defined in the "href" Parameter.
“di”参数指定承载“href”参数中定义的目标资源的设备的设备ID。

The Device ID may be used to qualify a relative reference used in the "href" or to lookup OCF Endpoint information for the relative reference.
设备ID可用于限定“href”中使用的相对引用，或用于查找相对引用的OCF端点信息。

7.8.2.5.6 "eps"Parameter "eps"参数

The "eps" Parameter indicates the OCF Endpoint information of the target Resource.
“eps”参数表示目标资源的OCF端点信息。

"eps" shall have as its value an array of items and each item represents OCF Endpoint information with "ep" and "pri" as specified in 10.2. "ep" is mandatory but "pri" is optional.
“eps”的值为一个项目数组，每个项目用10.2中指定的“ep”和“pri”表示OCF端点信息。“ep”是强制性的，而“pri”是选择性的。

This is an example of "eps" with multiple OCF Endpoints.
这是一个具有多个OCF端点的“eps”示例。

```
"eps": [
    {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
    {"ep": "coaps://[fe80::b1d6]:1122"},
    {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
]
```

When "eps" is present in a link, the OCF Endpoint information in "eps" can be used to access the target Resource referred by the "href" Parameter.
当链接中包含“eps”时，可以使用“eps”中的OCF端点信息来访问由“href”参数引用的目标资源。

Note that the type of OCF Endpoint - Secure or Unsecure - that a Resource exposes merely determines the connection type(s) guaranteed to be available for sending requests to the Resource.For example, if a Resource only exposes a single CoAP "ep", it does not guarantee that the Resource cannot also be accessed via a Secure OCF Endpoint (e.g. via a CoAPS "ep" from another Resources "eps information). Nor does exposing a given type of OCF Endpoint ensure that access to the Resource will be granted using the "ep" information. Whether requests to the Resource are granted or denied by the Access Control layer is separate from the "eps" information, and is determined by the configuration of the /acl2 Resource (see ISO/IEC 30118-2:2018 clause 13.5.3 for details).
注：资源公开的OCF端点的类型（安全的或不安全的）仅仅决定了向资源发送请求时保证可用的连接类型。例如，如果一个资源只公开一个CoAP“ep”，它并不保证该资源不能通过一个安全的OCF端点访问（例如，通过来自另一个资源的“eps信息”的CoAPS“ep”）。公开给定类型的OCF端点也不能确保使用“ep”信息授予对资源的访问权。访问控制层是否授予或拒绝对资源的请求与“eps”信息是分开的，由/acl2资源的配置决定。(详见ISO/IEC 30118-2:20 08第13.5.3章节)

When present, max-age information (e.g. Max-Age option for CoAP defined in IETF RFC 7252)determines the maximum time "eps" values may be cached before they are considered stale.
当存在时，max-age信息（例如IETF RFC 7252中定义的CoAP的max-age选项）确定在“eps”值被认为失效之前缓存的最大时间。

7.8.2.6 Formatting 格式化

When formatting in JSON, the list of Links shall be an array.
在JSON格式中，链接列表应该是一个数组。

7.8.2.7 List of Links in a Collection 集合中的链接列表

A Resource that exposes one or more Properties that are defined to be an array of Links where each Link can be discretely accessed is a Collection. The Property Name "links" is recommended for such an array of Links.

公开一个或多个属性的资源是一个集合，这些属性被定义为链接数组，其中每个链接都可以被离散地访问。对于这样一个链接数组，建议使用属性名“links”。

This is an example of a Resource with a list of Links.
这是一个带有链接列表的资源示例。

```
/Room1
{
    "rt": ["oic.wk.col"],
    "if": ["oic.if.ll", "oic.if.baseline" ],
    "color": "blue",
    "links":
    [
       {
          "href": "/switch",
          "rt": ["oic.r.switch.binary"],
          "if": [ "oic.if.a", "oic.if.baseline" ],
          "p": {"bm": 3}
       },
      {
          "href": "/brightness",
          "rt": ["oic.r.light.brightness"],
          "if": [ "oic.if.a", "oic.if.baseline" ],
          "p": {"bm": 3}
      }
   ]
}
```

7.8.2.8 Properties describing an array of Links 描述链接数组的属性

If a Resource Type that defines an array of Links (e.g. Collections, Atomic Measurements) has restrictions on the "rt" values that can be within the array of Links, the Resource Type will define the "rts" Property. The "rts" Property as defined in Table 12 will include all "rt" values allowed for all Links in the array. If the Resource Type does not define the "rts" Property or the "rts" Property is an empty array, then any "rt" value is permitted in the array of Links.
如果定义链接数组的资源类型（例如集合、原子测量）对链接数组中的“rt”值有限制，则资源类型将定义“rts”属性。表12中定义的“rts”属性将包括数组中所有链接允许的所有“rt”值。如果资源类型没有定义“rts”属性，或者“rts”属性是空数组，则在链接数组中允许任何“rt”值。

For all instances of a Resource Type that defines the "rts" Property, the "rt" Link Parameter in every Link in the array of Links shall be one of the "rt" values that is included in the "rts"Property.
对于定义“rts”属性的资源类型的所有实例，链接数组中的每个链接中的“rt”链接参数应是“rts”属性中包含的“rt”值之一。

Table 12 - Resource Types Property definition

表12 - 资源类型属性定义

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
ResourceTypes 资源类型	"rts"	"array"	Array of strings, conveying Resource Type IDs 字符串数组，传递资源类型id	N/A	R	No	An array of Resource Types that are supported within an array of Links exposed by a Resource. 资源公开的链接数组中支持的资源类型数组。

If a Resource Type that defines an array of Links has "rt" values which are required to be in the array, the Resource Type will define the "rts-m" Property, as defined in Table 13, which will contain all of the "rt" vaues that are required to be in the array of Links. If "rts-m" is defined, and "rts" isdefined and is not an empty array, then the "rt" values present in "rts-m" will be part of the values present in "rts". Moreover, if the "rts-m" Property is defined, it shall be mandated (i.e. included in the "required" field of a JSON definition) in the Resource definition and Introspection Device Data(see 11.4).

如果定义链接数组的资源类型有“rt”值，则需要在该数组中包含“rt”值，资源类型将定义表13中定义的“rt -m”属性，该属性将包含链接数组中所需的所有“rt”vaue。如果定义了“rts-m”，并且定义了“rts”而不是一个空数组，那么“rts-m”中的“rt”值将是“rts”中的值的一部分。此外，如果定义了“rts-m”属性，那么在资源定义和内省设备数据（见11.4）中应该强制使用该属性（即包含在JSON定义的“required”字段中）。

For all instances of a Resource Type that defines the "rts-m" Property, there shall be at least oneLink in the array of Links corresponding to each one of the "rt" values in the "rts-m" Property; forall such Links the "rt" Link Parameter shall contain at least one of the "rt" values in the "rts-m"Property.

对于定义“rt -m”属性的资源类型的所有实例，“rt -m”属性中的每个“rt”值对应的链接数组中应至少有一个链接；对于所有这些链接，“rt”链接参数应在“rt -m”属性中包含至少一个“rt”值。

Table 13 - Mandatory Resource Types Property definition

表13 - 强制性资源类型属性定义

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
MandatoryResource Types 强制资源类型	"rts-m"	"array"	Array of strings, conveying Resource Type IDs 字符串数组，传递资源类型id	N/A	R	No	An array of Resource Types that are mandatory to be exposed within an array of Links exposed by a Resource. 必须在资源公开的链接数组中公开的资源类型数组。

7.8.3 Collections 集合

7.8.3.1 Overview 综述

A Resource that contains one or more references (specified as Links) to other Resources is a Collection. These references may be related to each other or just be a list; the Collection provides a means to refer to this set of references with a single handle (i.e. the URI). A simple Resource is kept distinct from a Collection. Any Resource may be turned into a Collection by binding Resource references as Links. Collections may be used for creating, defining or specifying hierarchies,indexes, groups, and so on.
包含一个或多个指向其他资源的引用（指定为链接）的资源是一个集合。这些引用可能相互关联，也可能只是一个列表；集合提供了一种方法，通过一个句柄（即URI）来引用这组引用。简单的资源与集合是不同的。通过将资源引用绑定为链接，可以将任何资源转换为集合。集合可用于创建、定义或指定层次结构、索引、组等。

A Collection shall have at least one Resource Type and at least one OCF Interface bound at all times during its lifetime. During creation time of a Collection the Resource Type and OCF Interfaces are specified. The initial defined Resource Types and OCF Interfaces may be updated during its life time. These initial values may be overridden using mechanism used for overriding in the case of a Resource. Additional Resource Types and OCF Interfaces may be bound to the Collection at creation or later during the lifecycle of the Collection.
在一个集合的生命周期内，它应该具有至少一个资源类型和至少一个OCF接口。在集合的创建期间，将指定资源类型和OCF接口。初始定义的资源类型和OCF接口可以在其生命周期内更新。在集合的创建期间，将指定资源类型和OCF接口。初始定义的资源类型和OCF接口可以在其生命周期内更新。其他资源类型和OCF接口可能在创建时绑定到集合，或者在以后的集合生命周期中绑定到集合。

A Collection shall define a Property that is an array with zero or more Links. The target URIs in the Links may reference another Collection or another Resource. The referenced Collection or Resource may reside on the same Device as the Collection that includes that Link (called a local reference) or may reside on another Device (called a remote reference). The context URI of the Links in the array shall (implicitly) be the Collection that contains that Property. The (implicit)context URI may be overridden with explicit specification of the "anchor" Parameter in the Link Where the value of "anchor" is the new base of the Link.
集合应定义一个包含零个或多个链接的数组的属性。链接中的目标uri可以引用另一个集合或另一个资源。被引用的集合或资源可能与包含该链接的集合（称为本地引用）位于同一设备上，也可能位于另一个设备上（称为远程引用）。数组中链接的上下文URI应该(隐式地)是包含该属性的集合。（隐式）上下文URI可能被链接中“锚”参数的显式说明覆盖，其中“锚”的值是链接的新基。

A Resource may be referenced in more than one Collection, therefore, a unique parent-child relationship is not guaranteed. There is no pre-defined relationship between a Collection and the Resource referenced in the Collection, i.e., the application may use Collections to represent a relationship but none

is automatically implied or defined. The life cycles of the Collection and the referenced Resource are also independent of one another.
一个资源可能在多个集合中被引用，因此不能保证有唯一的主-从关系。集合和集合中引用的资源之间没有预定义的关系，即，应用程序可以使用集合来表示关系，但不会自动隐含或定义任何关系。集合和引用资源的生命周期也彼此独立。

In the following example a Property "links" represents the list of Links in a Collection. The "links"Property has, as its value, an array of items and each item is a Link.
在下面的示例中，属性“links”表示集合中的链接列表。“links”属性的值是一个项目数组，每个项目都是一个链接。

```
/my/house   This is IRI/URI of the Resource
{
    "rt": ["my.r.house"],   This and the next 3 lines are the Properties of the Resource.
    "color": "blue",
    "n": "myhouse",
    "links": [
        {  This and the next 4 lines are the Parameters of a Link
            "href": "/door",
            "rt": ["oic.r.door"],
            "if": ["oic.if.a", "oic.if.baseline"]
        },
        {
            "href": "/door/lock.status",
            "rt": ["oic.r.lock"],
            "if": ["oic.if.a", "oic.if.baseline"]
       },
       {
           "href": "/light",
           "rt": ["oic.r.light"],
           "if": ["oic.if.s", "oic.if.baseline"]
       },
       {
           "href": "/binarySwitch",
           "rt": ["oic.r.switch.binary"],
           "if": ["oic.if.a", "oic.if.baseline"]
        }
    ]
}
```

A Collection may be: 一个集合可以是：

- A pre-defined Collection where the Collection has been defined a priori and the Collection is static over its lifetime. Such Collections may be used to model, for example, an appliance that is

composed of other Devices or fixed set of Resources representing fixed functions.
预定义的集合，其中集合是预先定义的，并且在其生命周期内是静态的。这些集合可用于建模，例如，由其他设备或表示固定功能的固定资源集组成的设备。

- A Device local Collection where the Collection is used only on the Device that hosts the Collection. Such Collections may be used as a short-hand on a Client for referring to many Servers as one.
  设备本地集合，其中集合仅在承载该集合的设备上使用。这样的集合可以作为客户机的简写，将多个服务器引用为一个。
- A centralized Collection where the Collection is hosted on a Device but other Devices may access or update the Collection.
  集中式集合，其中集合托管在设备上，但其他设备可以访问或更新集合。
- A hosted Collection where the Collection is centralized but is managed by an authorized agentor party.
  托管的集合，其中集合是集中的，但由授权的代理或一方管理。

7.8.3.2 Collection Properties 集合属性

A Collection shall define a Property that is an array of Links (the Property Name "links" is recommended). In addition, other Properties may be defined for the Collection by the ResourceType. The mandatory and recommended Common Properties for a Collection are shown in Table 14.This list of Common Properties is in addition to those defined for Resources in 7.3.2.
集合应定义一个链接数组的属性（建议使用属性名“Links”）。此外，可以根据资源类型为集合定义其他属性。表14显示了集合的强制性和推荐的通用属性。除了7.3.2中为资源定义的属性之外，还提供了这个公共属性列表。

Table 14 - Common Properties for Collections (in addition to Common Properties definedin 7.3.2)
表14-集合的公共属性（除了7.3.2中定义的公共属性之外）

Property 属性	Description 描述	Property Name 属性名称	Value Type 值类型	Mandatory 强制性
Links **链接**	The array of Links in the Collection 集合中的链接数组	Per Resource Type definition 按资源类型定义	json Array of Links 数组的链接	Yes
Resource Types **资源类型**	The list of allowed Resource Types for Links in the Collection. 集合中链接的允许资源类型列表。 If this Property is not defined or is null string then any Resource Type is permitted如果这个属性没有定义或者是空字符串，那么任何资源类型都是允许的。	As defined in Table 12 如表12所示	As defined in Table 12 如表12所示	No

Mandatory Resource Types 强制资源类型	The list of Resource Types for Links that are mandatory in the Collection.	As defined in Table 13 如表13所示	As defined in Table 13 如表13所示	No

7.8.3.3 Default Resource Type 默认资源类型

A default Resource Type, "oic.wk.col", is available for Collections. This Resource Type shall be used only when another type has not been defined on the Collection or when no Resource Type has been specified at the creation of the Collection.
默认的资源类型“oic.wk.col”可用于收集。此资源类型应使用仅在未在集合上定义另一类型或在创建集合时未指定资源类型时使用。

The default Resource Type provides support for the Common Properties including an array of Links with the Property Name "links".
默认资源类型提供对公共属性的支持，包括属性名为“Links”的链接数组。

7.8.3.4 Default OCF Interface 默认OCF接口

All instances of a Collection shall support the links list ("oic.if.ll") OCF Interface in addition to the baseline ("oic.if.baseline") OCF Interface. An instance of a Collection may optionally support additional OCF Interfaces that are defined within this document. The Default OCF Interface for a Collection shall be links list ("oic.if.ll") unless otherwise specified by the Resource Type definition.
集合的所有实例除了支持基线（"oic.if.baseline"）OCF接口外，还应支持链接列表（"oic.if.ll"）OCF接口。集合的实例可以选择支持本文档中定义的其他OCF接口。除非资源类型定义另有规定，否则集合的默认OCF接口应该是links列表（“oic.if.ll”）。

7.8.4 Atomic Measurement 原子测量

7.8.4.1 Overview 综述

Certain use cases require that the Properties of multiple Resources are only accessible as a group and individual access to those Properties of each Resource by a Client is prohibited. The Atomic Measurement Resource Type is defined to meet this requirement. This is accomplished through the use of the Batch OCF Interface.
某些用例要求多个资源的属性只能作为一个组访问，并且禁止客户端单独访问每个资源的属性。原子测量资源类型是为了满足这个需求而定义的。这是通过使用批处理OCF接口实现的。

7.8.4.2 Atomic Measurement Properties 原子测量属性

An Atomic Measurement shall define a Property that is an array of Links (the Property Name "links"is recommended). In addition, other Properties may be defined for the Atomic Measurement by the

Resource Type. The mandatory and recommended Common Properties for an Atomic Measurement are shown in Table 15. This list of Common Properties is in addition to those defined for Resources in 7.3.2.
原子测量应该定义一个链接数组的属性（建议使用属性名“链接”）。此外，可以通过资源类型为原子度量定义其他属性。原子测量的强制性和推荐的通用属性如表15所示。除了7.3.2中为资源定义的属性之外，还提供了这个公共属性列表。

Table 15 - Common Properties for Atomic Measurement (in addition to Common Properties defined in 7.3.2)
表15 - 原子测量的一般性质（除了7.3.2中定义的公共属性之外）

Property 属性	Description 描述	Property Name 属性名称	Value Type 值类型	Mandatory 强制性
Links **链接**	The array of Links in the Atomic Measurement 原子测量中的链接数组	Per Resource Type definition 按资源类型定义	json Array of Links 数组的链接	Yes
Resource Types **资源类型**	The list of allowed Resource Types for Links in the Atomic Measurement 集合中链接的允许资源类型列表。 If this Property is not defined or is null string then any Resource Type is permitted如果这个属性没有定义或者是空字符串，那么任何资源类型都是允许的。	As defined in Table 12 如表12所示	As defined in Table 12 如表12所示	No
Mandatory Resource Types强制资源类型	The list of Resource Types for Links that are mandatory in the Atomic Measurement. 原子中强制链接的资源类型列表测量。	As defined in Table 13 如表13所示	As defined in Table 13 如表13所示	No

7.8.4.3 Normative behavior 规范行为

The normative behavior of an Atomic Measurement is as follows:
原子测量的规范行为如下所示：

- The behavior of the Batch OCF Interface ("oic.if.b") on the Atomic Measurement is defined as follows:
  批量OCF接口（“oic.if.b”）对原子测量的行为定义如下：
  - Only RETRIEVE and NOTIFY operations are supported, for Batch OCF Interface, on Atomic Measurement; the behavior of the RETRIEVE and NOTIFY operations shall be the same as specified in 7.6.3.4, with exceptions as provided for in 7.8.4.3.
    对于批量OCF接口，原子测量只支持检索和通知操作;检索和通知操作的行为应与7.6.3.4中规定的相同，但7.8.4.3中规定的除外。
  - The UPDATE operation is not allowed, for Batch OCF Interface, on Atomic Measurement; ifan UPDATE operation is received, it shall result in a method not allowed error code.
    对于批量OCF接口，原子测量不允许更新操作；一个更新操作被接收，它将导致一个方法不允许错误代码。

    - An error response shall not include any representation of a linked Resource (i.e. empty response for all linked Resources).
      错误响应不应包含任何链接资源的表示（即所有链接资源的空响应）。
- Any linked Resource within an Atomic Measurement (i.e. the target Resource of a Link in anAtomic Measurement) is subject to the following conditions:
  原子测量范围内的任何连接资源（即原子测量范围内连接的目标资源）均须符合下列条件:
    - Linked Resources within an Atomic Measurement and the Atomic Measurement itself shall exist on a single Server.
      原子测量内部的链接资源和原子测量本身应存在于单个服务器上。
    - CRUDN operations shall not be allowed on linked Resources and shall result in a forbidden error code.
      CRUDN操作不允许在链接的资源上进行，并且会导致一个禁止的错误代码。
    - Linked Resources shall not expose the "oic.if.ll" OCF Interface. Since CRUDN operations are not allowed on linked Resources, the "oic.if.ll" OCF Interface would never be accessible.
      链接的资源不应暴露"oic.if.ll" OCF接口。由于CRUDN操作不允许在链接的资源上执行，因此“"oic.if.ll" OCF 接口将永远无法访问。
- Links to linked Resources in an Atomic Measurement shall only be accessible through the"oic.if.ll" or the "oic.if.baseline" OCF Interfaces of an Atomic Measurement.
  原子测量中已链接资源的链接应只能通过原子测量的"oic.if.ll"或"oic.if.baseline" OCF 接口。
    - The linked Resources shall not be listed in "/oic/res".
      链接的资源不得列入“/oic/res”。
- A linked Resource in an Atomic Measurement shall have defined one of "oic.if.a", "oic.if.s","oic.if.r", or "oic.if.rw" as its Default OCF Interface.
  原子测量中的链接资源应定义"oic.if.a", "oic.if.s","oic.if.r", 或者"oic.if.rw"之一，作为其默认的OCF接口。
- Not all linked Resources in an Atomic Measurement are required to be Observable. If an Atomic Measurement is being Observed using the "oic.if.b" OCF Interface, notification responses shall not be generated when the linked Resources which are not marked Observable are updated or change state.
  在原子测量中，并非所有的连接资源都需要是可见的。如果一个原子使用“oic.if”进行测量。b”OCF接口，未标记为“可观察”的链接资源更新或改变状态时，不应生成通知响应。
- All linked Resources in an Atomic Measurement shall be included in every RETRIEVE and Observe response when using the"oic.if.b" OCF Interface.
  在使用"oic.if.b" OCF接口时，在每次检索和观察响应时应包括原子测量中的所有链接资源。
- An Atomic Measurement shall support the "oic.if.b" and the "oic.if.ll" OCF Interfaces.
  原子测量由"oic.if.b"和"oic.if.ll"“OCF的接口支撑。
- Filtering of linked Resources in an Atomic Measurement is not allowed. Query parameters that select one or more individual linked Resources in a request to an Atomic Measurement shall result in a "forbidden" error code.

不允许在原子测量中过滤链接资源。在对原子测量的请求中选择一个或多个单独链接资源的查询参数将导致“禁止”错误代码。

- If the "rel" Link Parameter is included in a Link contained in an Atomic Measurement, it shallhave either the "hosts" or the "item" value.
  如果“rel”链接参数包含在原子测量中包含的链接中，则它应该具有“hosts”或“item”值。
- The Default OCF Interface of an Atomic Measurement is "oic.if.b".
  原子测量的默认OCF接口是“oic.if.b”

7.8.4.4 Security considerations 安全注意事项

Access rights to an Atomic Measurement Resource Type is as specified in clause 12.2.7.2 (ACL considerations for batch request to the Atomic Measurement Resource Type) of ISO/IEC 30118¬2020 2:2018).
原子测量资源类型的访问权限如ISO/IEC 30118 - 2020 2:2018第12.2.7.2条（批量请求原子测量资源类型的ACL考虑因素）中规定。

7.8.4.5 Default Resource Type 默认资源类型

The Resource Type is defined as "oic.wk.atomicmeasurement" as defined in Table 16.
资源类型定义为“oic.wk.atomicmeasurement”。如表16所示。

Table 16 - Atomic Measurement Resource Type
表16-原子测量资源类型

Pre-defined URI 预定义URI	Resource Type Title 资源类型标题	Resource Type ID ("rt" value) 资源类型ID("rt"值)	OCF Interfaces OCF接口	Description 描述	Related Functional Interaction 关系功能联系	M/CR/O
none 无	Atomic Measurement 原子测量	"oic.wk.atomicmeasurement"	"oic.if.ll" "oic.if.baseline" "oic.if.b"	A specialisation of the Collection pattern to ensure atomic RETRIEVAL of its referred Resources 集合模式的专门化，以确保对其引用资源的原子检索	RETRIEVE,NOTIFY 检索，通知	O

The Properties for Atomic Measurement are as defined in Table 17.
原子测量的属性定义在表17中。

Table 17 - Properties for Atomic Measurement (in addition to Common Properties defined2027 in 7.3.2)

表17 - 原子测量属性（除了7.3.2中定义的公共属性2027之外）

Property 属性	Description 描述	Property name 属性名称	Value Type 值类型	Mandatory 强制性
Links 链接	The set of links that point to the linked Resources 指向链接资源的链接集	Per Resource Type definition 按资源类型定义	json Array of Links 数组的链接	Yes

7.9 Query Parameters 查询参数

7.9.1 Introduction 引言

Properties and Parameters (including those that are part of a Link) may be used in the query part of a URI (see 6.2.2) 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. Only ASCII strings are permitted in query filters, and NULL characters are disallowed in query filters. This means that only Property Values with ASCII characters may be matched in aquery filter.
属性和参数（包括链接的一部分）可以用于URI的查询部分（参见6.2.2），作为选择特定资源的标准之一。这是通过声明属性（即：<属性名>　=　<所需属性值>）作为查询的一个片段。查询过滤器中只允许ASCII字符串，查询过滤器中不允许空字符。这意味着在查询筛选器中只能匹配具有ASCII字符的属性值。

The Resource is selected when all the declared Properties or Link Parameters in the query match the corresponding Properties or Link Parameters in the target.
当查询中所有声明的属性或链接参数与目标中相应的属性或链接参数匹配时，将选择该资源。

7.9.2 Use of multiple parameters within a query 在查询中使用多个参数

When a query contains multiple separate query parameters these are delimited by an as described in 6.2.2.
当一个查询包含多个单独的查询参数时，这些参数由6.2.2中描述的an分隔。

A Client may apply multiple separate query parameters, for example "?ins=11111 &rt=oic.r.switch.binary". If such queries are supported by the Server this shall be accomplished by matching "all of" the different query parameter types ("rt", "ins", "if", etc)against the target of the query. In the example, this resolves to an instance of oic.r.switch.binary that also has an "ins" populated as "11111". There is no significance applied to the order of the query parameters.
客户端可以应用多个独立的查询参数，例如“?”ins = 11111&rt = oic.r.switch.binary”。如果服务器支持此类查

询，则应将“所有”不同的查询参数类型（“rt”、“ins”、“If”等）与查询的目标进行匹配。在本例中，这将解析为一个oic.r.switch实例。还有一个“ins”被填充为“11111”的二进制代码。查询参数的顺序没有意义。

A Client may select more than one Resource Type using repeated query parameters, for example"?rt=oic.r.switch.binary&rt=oic.r.ramptime". If such queries are supported by the Server this shall be accomplished by matching "any of" the repeated query parameters against the target of thequery. In the example, any instances of "oic.r.switch.binary" and/or "oic.r.ramptime" that may exist are selected.
客户端可以使用重复的查询参数选择多个资源类型，例如“?rt=oic.r.switch.binary&rt=oic.r.ramptime”。如果服务器支持此类查询，则应通过针对查询的目标匹配“任何”重复查询参数来实现。在本例中，"oic.r.switch.binary"和/或"oic.r.ramptime"可能存在的ramptime被选中。

A Client may combine both multiple repeated parameters and multiple separate parameters in a single query, for example "?if=oic.if.b&ins=11111 &rt=oic.r.switch.binary&rt=oic.r.ramptime". If such queries are supported by the Server this shall be accomplished by matching "any of" the repeated query parameters and then matching "all of" the different query parameter types. In the example any instances of "oic.r.switch.binary" and/or "oic.r.ramptime" that also have an "ins" of "11111" that may exist are selected in a batch response.
一个客户端可以在一个查询中将多个重复参数和多个独立参数组合起来，例如，"?if=oic.if.b&ins=11111 &rt=oic.r.switch.binary&rt=oic.r.ramptime".如果服务器支持此类查询，则应通过匹配“任何”重复查询参数，然后匹配“所有”不同的查询参数类型来完成。在本例中，是“oic.r.switch.binary”的任何实例。”和/或“oic.r。在批处理响应中选择“ramptime”，它也有可能存在“ins11111”。

NOTE The parameters within a query string are represented within the actual messaging protocol as defined in clause 11.5.
注 查询字符串中的参数是在第11.5章节中定义的实际消息传递协议中表示的。

7.9.3 Application to multi-value "rt" Resources 应用于多值“rt”资源

An "rt" query for a multi-value "rt" Resource with the Default OCF 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 a 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(s). 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.
默认OCF接口"oic.if.a", "oic.if.s","oic.if.r", "oic.if.rw"或者"oic.if.baseline"的多值“rt”资源的“rt”查询是通用“rt”查询的扩展。当服务器接收到带有“rt”查询的多值“rt”资源的检索请求时（如，GET/ResExample?rt=oic.r.foo），服务器应该只在查询值是目标资源的“rt”属性值的一个项时响应，并且应该只返回与查询值关联的属性。例如，收到针对GET/ResExample?rt=oic.r.switch.binary具有“rt”:["oic.r.switch.binary", "oic.r.light.brightness"]的资源，服务器只表现oic.r.switch.binary属性。

7.9.4 OCF Interface specific considerations for queries OCF接口查询的具体注意事项

7.9.4.1 OCF Interface selection OCF接口选择

When an OCF Interface is to be selected for a request, it shall be specified as a query parameter in the URI of the Resource in the request message. If no query parameter is specified, then the Default OCF Interface shall be used. If the selected OCF Interface is not one of the permitted OCF Interfaces on the Resource then selecting that OCF Interface is an error and the Server shall respond with an error response code.
当要为请求选择OCF接口时，应将其指定为请求消息中资源的URI中的查询参数。如果没有指定查询参数，则使用默认的OCF接口应使用。如果所选的OCF接口不是资源上允许的OCF接口之一，则选择该OCF接口是错误的，服务器应使用错误响应代码进行响应。

For example, the baseline OCF Interface may be 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".
例如，基线OCF接口可能选择添加"if=oic.if.baseline"到目标资源URI中的查询参数列表中。比如，"GET/oic/res?if=oic.if.baseline"。

7.9.4.2 Batch OCF Interface 批量OCF 接口

See 7.6.3.4 for details on the batch OCF Interface itself. Query parameters may be used with the batch OCF Interface in order to select particular Resources in a Collection for retrieval or update;these parameters are used to select items in the Collection by matching Link Parameter Values.
有关批量OCF接口本身的详细信息，请参见7.6.3.4。查询参数可与批量OCF接口一起使用，以选择集合中的特定资源进行检索或更新；这些参数用于通过匹配链接参数值来选择集合中的项。

When Link selection query parameters are used with RETRIEVE operations applied using the batch OCF Interface, only the Resources in the Collection with matching Link Parameters should be returned.
当链接选择查询参数与使用批量OCF接口应用的RETRIEVE操作一起使用时，应该只返回具有匹配链接参数的集合中的资源。

When Link selection query parameters are used with UPDATE operations applied using the batch OCF Interface, only the Resources having matching Link Parameters should be updated.
当链接选择查询参数与使用批量OCF接口应用的UPDATE操作一起使用时，应该只更新具有匹配链接参数的资源。

See 7.6.3.4.5 for examples of RETRIEVE and UPDATE operations that use Link selection queryparameters.
有关使用链接选择查询参数的RETRIEVE和UPDATE操作的示例，请参见7.6.3.4.5。

8 CRUDN

8.1 Overview 综述

CREATE, RETRIEVE, UPDATE, DELETE, and NOTIFY (CRUDN) are operations defined for manipulating Resources. These operations are performed by a Client on the Resources contained in n Server.
CREATE, RETRIEVE, UPDATE, DELETE, 以及NOTIFY（CRUDN）是为操作资源而定义的操作。这些操作由客户端执行以对n服务器包含的资源进行操作。

On reception of a valid CRUDN operation a Server hosting the Resource that is the target of the request shall generate a response depending on the OCF Interface included in the request; or based on the Default OCF Interface for the Resource Type if no OCF Interface is included.
在接收到有效的CRUDN操作时，承载请求目标资源的服务器应根据请求中包含的OCF接口生成响应；或者基于资源类型的默认OCF接口（如果请求中没有包含OCF接口）。

CRUDN operations utilize a set of parameters that are carried in the messages and are defined in Table 18. A Device shall use CBOR as the default payload (content) encoding scheme for Resource Representations included in CRUDN operations and operation responses; a Device may negotiate a different payload encoding scheme (e.g, see in 12.2.4 for CoAP messaging). Clauses 8.2 through 8.6 respectively specify the CRUDN operations and use of the parameters. The type definitions for these terms will be mapped in the clause 11.5 for each protocol.
CRUDN操作采用消息中携带的一组参数，这些参数在表18中定义。设备应使用CBOR作为CRUDN操作和操作响应中包含的资源表示的默认有效载荷（内容）编码方案；一个设备可以设配不同的有效载荷编码方案（例：参见12.2.4中的CoAP消息传递）。8.2章节至8.6章节分别规定了CRUDN操作和参数的使用。这些术语的类型定义将在11.5章节中映射到每个协议。

Table 18 - Parameters of CRUDN messages
表18 - CRUDN消息的参数

Applicability 适用性	Name 名称	Denotation 指示	Defination 定义
All messages 所有信息	*fr*	From 来自	The URI of the message originator. 消息发送者的URI。
	to	To 送至	The URI of the recipient of the message. 消息接收者的URI。
	ri	Request Identifier 请求标识符	The identifier that uniquely identifies the message in the originator and the recipient. 唯一标识发送者和接收者消息的标识符。
	cn	Content	Information specific to the operation.

		内容	特定于操作的信息。
Requests 请求	*Op*	Operation 操作	Specific operation requested to be performed by the Server. 服务器请求执行的特定操作。
	obs	Observe 观察	Indicator for an Observe request. 用于观察请求的指示器。
Responses 响应	*rs*	Response Code 响应代码	Indicator of the result of the request; whether it was accepted and what the conclusion of the operation was. The values of the response code for CRUDN operations shall conform to those as defined in clause 5.9 and 12.1.2 in IETF RFC 7252. 查询结果的指示器；是否被接收以及操作的结果是什么。CRUDN操作的响应代码的值应符合IETF RFC 7252中5.9和12.1.2章节中定义的值。
	obs	Observe 观察	Indicator for an Observe response. 用于观察响应的指示器。

8.2 CREATE 创建

8.2.1 Overview 综述

The CREATE operation is used to request the creation of new Resources on the Server. The CREATE operation is initiated by the Client and consists of three steps, as depicted in Figure 5.
CREATE操作用于在服务器上请求创建新资源。CREATE操作由客户端发起，包含三个步骤，如图5所示。

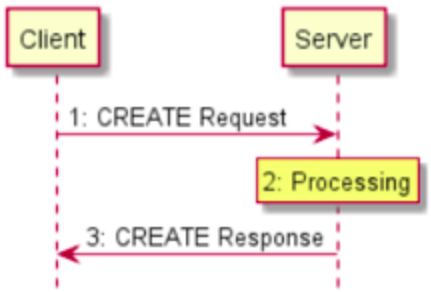

Figure 5 - CREATE operation
图5 - CREATE操作

8.2.2 CREATE request CREATE请求

The CREATE request message is transmitted by the Client to the Server to create a new Resource by the Server. The CREATE request message will carry the following parameters:
CREATE请求消息由客户端传输到服务器，并由服务器创建新资源。CREATE请求消息将携带以下参数:

- fr: Unique identifier of the Client
  fr：客户端的唯一标识符
- to: URI of the target Resource responsible for creation of the new Resource.
  to：负责创建新资源的目标资源的URI。
- ri: Identifier of the CREATE request.
  ri：CREATE请求的标识符。
- cn: Information of the Resource to be created by the Server.
  cn：服务器要创建的资源信息。
    - cn will include the URI and Resource Type Property of the Resource to be created.
      cn将包含要创建的资源的URI和资源类型属性。
    - cn may include additional Properties of the Resource to be created.
      cn可以包含要创建的资源的附加属性。
- op: CREATE
  op: CREATE

8.2.3 Processing by the Server 服务器处理

Following the receipt of a CREATE request, the Server may validate if the Client has the appropriate rights for creating the requested Resource. If the validation is successful, the Server creates the requested Resource. The Server caches the value of ri parameter in the CREATE request for inclusion in the CREATE response message.
接收到CREATE请求之后，服务器可能会验证客户端是否具有创建该请求资源的适当权限。如果验证成功，服务器将创建请求的资源。服务器缓存CREATE请求中ri参数的值，以便将其包含在CREATE响应消息中。

8.2.4 CREATE response CREATE响应

The Server shall transmit a CREATE response message in response to a CREATE request message from a Client. The CREATE response message will include the following parameters:
服务器应发送CREATE响应消息以响应来自客户端的CREATE请求消息。CREATE响应消息将包括以下参数：

- fr: Unique identifier of the Server
  fr：服务器的唯一标识符
- to: Unique identifier of the Client
  to：客户端的唯一标识符
- ri: Identifier included in the CREATE request
  ri：CREATE请求中包含的标识符
- cn: Information of the Resource as created by the Server.
  cn：由服务器创建的资源信息。

    - cn will include the URI of the created Resource.
      cn将包含所创建资源的URI。
    - cn will include the Resource representation of the created Resource.
      cn将包括所创建资源的资源表示。
- rs: The result of the CREATE operation.
  rs：CREATE操作的结果。

8.3 RETRIEVE 检索

8.3.1 Overview 综述

The RETRIEVE operation is used to request the current state or representation of a Resource. The RETRIEVE operation is initiated by the Client and consists of three steps, as depicted in Figure 6.
RETRIEVE操作用于请求资源的当前状态或表示形式。RETRIEVE操作由客户端发起，包含三个步骤，如图6所示。

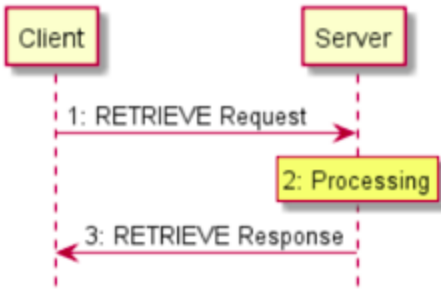

Figure 6 - RETRIEVE operation
图6 - RETRIEVE操作

8.3.2 RETRIEVE request RETRIEVE请求

RETRIEVE request message is transmitted by the Client to the Server to request the representation of a Resource from a Server. The RETRIEVE request message will carry the following parameters:
RETRIEVE请求消息由客户端发送到服务器，以便从服务器请求资源的表示形式。RETRIEVE请求消息将携带以下参数：

- fr: Unique identifier of the Client.
  fr:客户端的唯一标识符。
- to: URI of the Resource the Client is targeting.
  to:客户端目标资源的URI。
- ri: Identifier of the RETRIEVE request.
  ri:RETRIEVE请求的标识符。

- op: RETRIEVE.
  op:RETRIEVE。

8.3.3 Processing by the Server 服务器处理

Following the receipt of a RETRIEVE request, the Server may validate if the Client has the appropriate rights for retrieving the requested data and the Properties are readable. The Server caches the value of ri parameter in the RETRIEVE request for use in the response
在接收到RETRIEVE请求之后，服务器可能会验证客户端是否具有检索所请求数据的适当权限，并且该数据属性是可读的。服务器缓存RETRIEVE请求中ri参数的值，以便在响应中使用。

8.3.4 RETRIEVE response RETRIEVE响应

The Server shall transmit a RETRIEVE response message in response to a RETRIEVE request message from a Client. The RETRIEVE response message will include the following parameters:
服务器应发送RETRIEVE响应消息以响应来自客户端的RETRIEVE请求消息。RETRIEVE响应消息将包括以下参数：

- fr: Unique identifier of the Server.
  fr:服务器的唯一标识符。
- to: Unique identifier of the Client.
  to:客户端的唯一标识符。
- ri: Identifier included in the RETRIEVE request.
  ri: 包含在RETRIEVE请求中的标识符。
- cn: Information of the Resource as requested by the Client.
  cn:客户端请求的资源信息。
    - cn should include the URI of the Resource targeted in the RETRIEVE request.
      cn应该在RETRIEVE请求中包含目标资源的URI。
- rs: The result of the RETRIEVE operation.
  rs: RETRIEVE操作的结果。

8.4 UPDATE 更新

8.4.1 Overview 综述

The UPDATE operation is either a Partial UPDATE or a complete replacement of the information in a Resource in conjunction with the OCF Interface that is also applied to the operation. The UPDATE operation is initiated by the Client and consists of three steps, as depicted in Figure 7.
UPDATE操作是将资源中的信息与也应用于操作的OCF接口一起进行部分UPDATE或完全替换。UPDATE操作由客户端发起，包含三个步骤，如图7所示。

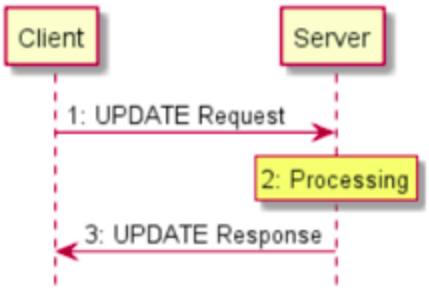

Figure 7 - UPDATE operation
图7 - UPDATE操作

8.4.2 UPDATE request UPDATE请求

The UPDATE request message is transmitted by the Client to the Server to request the update of information of a Resource on the Server. The UPDATE request message will carry the following parameters:
客户端将UPDATE请求消息发送到服务器，以请求更新服务器上资源的信息。UPDATE请求消息将包含以下参数：

- fr: Unique identifier of the Client.
  fr:客户端的唯一标识符。
- to: URI of the Resource targeted for the information update.
  to:用于信息更新的目标资源的URI。
- ri: Identifier of the UPDATE request.
  ri: UPDATE请求的标识符。
- op: UPDATE.
  op:UPDATE。
- cn: Information, including Properties, of the Resource to be updated at the target Resource.
  cn:信息，包括要在目标资源上更新的资源属性。

8.4.3 Processing by the Server 服务器处理

8.4.3.1 Overview 综述

Following the receipt of an UPDATE request, the Server may validate if the Client has the appropriate rights for updating the requested data. If the validation is successful the Server updates the target Resource information according to the information carried in cn parameter of the UPDATE request message. The Server caches the value of ri parameter in the UPDATE request for use in the response.
在接收到UPDATE请求之后，服务器可能会验证客户端是否具有更新所请求数据的适当权限。如果验证成

功，服务器将根据UPDATE请求消息的cn参数中包含的信息更新目标资源信息。服务器缓存UPDATE请求中ri参数的值，以便在响应中使用。

An UPDATE request that includes Properties that are read-only shall be rejected by the Server with an rs indicating a bad request.
包含只读属性的UPDATE请求将被服务器拒绝，并用rs表示为坏请求。

An UPDATE request shall be applied only to the Properties in the target Resource visible via the applied OCF Interface that support the operation. An UPDATE of non-existent Properties is ignored.
UPDATE请求只能通过支持操作的应用OCF接口应用到目标资源中可见的属性。不存在属性的UPDATE将被忽略。

An UPDATE request shall be applied to the Properties in the target Resource even if those Property Values are the same as the values currently exposed by the target Resource.
UPDATE请求应应用于目标资源中的属性，即使这些属性值与目标资源当前公开的值相同。

8.4.3.2 Resource monitoring by the Server 服务器对资源的监视

The Server shall monitor the state the Resource identified in the Observe request from the Client.Anytime there is a change in the state of the Observed Resource or an UPDATE operation applied to the Resource, the Server sends another RETRIEVE response with the Observe indication. The Mechanism does not allow the Client to specify any bounds or limits which trigger a notification,the decision is left entirely to the Server.
服务器应监视来自客户端的观察请求中标识的资源状态。每当被观察资源的状态发生变化，或者对资源应用了UPDATE操作，服务器就会发送另一个带有观察指示的RETRIEVE响应。该机制不允许客户端指定触发通知的任何界限或限制，完全由服务器来决定。

8.4.3.3 Additional RETRIEVE responses with Observe indication 观察指示的附加RETRIEVE响应

The Server shall transmit updated RETRIEVE response messages following Observed changes in the state of the Resources requested by the Client. The RETRIEVE response message shall include the parameters listed in 11.3.2.4.
在客户端请求的资源状态发生变化后，服务器应发送更新的RETRIEVE响应消息。RETRIEVE响应消息应包含11.3.2.4中列出的参数。

8.4.4 UPDATE response UPDATE响应

The UPDATE response message will include the following parameters:
UPDATE响应消息将包括以下参数：

- fr: Unique identifier of the Server.
  fr:服务器的唯一标识符。

- to: Unique identifier of the Client.
  to:客户端的唯一标识符。
- ri: Identifier included in the UPDATE request.
  ri:UPDATE请求中包含的标识符。
- rs: The result of the UPDATE request.
  rs: UPDATE请求的结果。

The UPDATE response message may also include the following parameters:
UPDATE响应消息还可以包括以下参数：

- cn: The Resource representation following processing of the UPDATE request.
  cn: UPDATE请求处理后的资源表示。

8.5 DELETE 删除

8.5.1 Overview 综述

The DELETE operation is used to request the removal of a Resource. The DELETE operation is initiated by the Client and consists of three steps, as depicted in Figure 8.
DELETE操作用于请求删除资源。 DELETE操作由客户端发起，包含三个步骤，如图8所示。

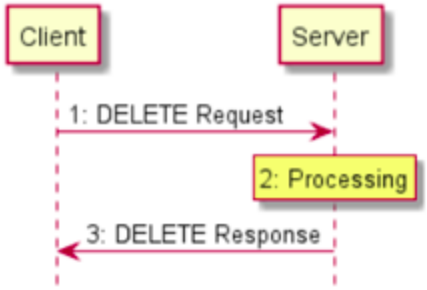

Figure 8 - DELETE operation
图8 - DELETE操作

8.5.2 DELETE request DELETE请求

DELETE request message is transmitted by the Client to the Server to delete a Resource on the Server. The DELETE request message will carry the following parameters:
DELETE请求消息由客户端发送到服务器，以删除服务器上的资源。DELETE请求消息将包含以下参数：

- fr: Unique identifier of the Client.
  fr:客户端的唯一标识符。

- to: URI of the target Resource which is the target of deletion.
  to:要删除的目标资源的URI。
- ri: Identifier of the DELETE request.
  ri:DELETE请求的标识符。
- op: DELETE.
  op:DELETE。

8.5.3 Processing by the Server 服务器处理

Following the receipt of a DELETE request, the Server may validate if the Client has the appropriate rights for deleting the identified Resource, and whether the identified Resource exists. If the validation is successful, the Server removes the requested Resource and deletes all the associated information. The Server caches the value of ri parameter in the DELETE request for use in the response.
在接收到DELETE请求之后，服务器可能会验证客户端是否具有删除已标识的资源的适当权限，以及标识的资源是否存在。如果验证成功，服务器将删除请求的资源和所有相关信息。服务器缓存DELETE请求中的ri参数的值，以便在响应中使用。

8.5.4 DELETE response DELETE 响应

The Server shall transmit a DELETE response message in response to a DELETE request message from a Client. The DELETE response message will include the following parameters:
服务器应发送一个DELETE响应消息来响应来自客户端的一个DELETE请求消息。DELETE响应消息将包括以下参数：

- fr: Unique identifier of the Server.
  fr:服务器的唯一标识符。
- to: Unique identifier of the Client.
  to:客户端的唯一标识符。
- ri: Identifier included in the DELETE request.
  ri:DELETE请求中包含的标识符。
- rs: The result of the DELETE operation.
  rs:DELETE操作的结果。

8.6 NOTIFY 通知

8.6.1 Overview 综述

The NOTIFY operation is used to request asynchronous notification of state changes. Complete description of the NOTIFY operation is provided in 11.3. The NOTIFY operation uses the NOTIFICATION response message which is defined here.

NOTIFY操作用于请求状态更改的异步通知。NOTIFY操作的完整描述在11.3章节中提供。NOTIFY操作使用这里定义的NOTIFICATION响应消息。

8.6.2 NOTIFICATION response NOTIFICATION响应

The NOTIFICATION response message is sent by a Server to notify the URLs identified by the Client of a state change. The NOTIFICATION response message carries the following parameters:
NOTIFICATION响应消息由服务器发送，以通知客户端标识的状态更改的url。NOTIFICATION响应消息包含以下参数：

- fr: Unique identifier of the Server.
  fr:服务器的唯一标识符。
- to: URI of the Resource target of the NOTIFICATION message.
  to:NOTIFICATION消息的资源目标的URI。
- ri: Identifier included in the CREATE request.
  ri:CREATE请求中包含的标识符。
- op: NOTIFY.
  op: NOTIFY。
- cn: The updated state of the Resource.
  cn: 资源的更新状态。

9 Network and connectivity 网络和连接

9.1 Introduction 引言

The Internet of Things is comprised of a wide range of applications which sense and actuate the physical world with a broad spectrum of device and network capabilities: from battery powered nodes transmitting 100 bytes per day and able to last 10 years on a coin cell battery, to mains powered nodes able to maintain Megabit video streams. It is estimated that many 10s of billions of IoT devices will be deployed over the coming years.
物联网由众多的应用程序组成，这些应用程序通过众多的设备和网络功能来感知和驱动物理世界：从纽扣电池支持使用10年（每天传输100字节）的电池供电节点到可维持超大视频流的电源供电节点。据估计，未来几年将部署数以十亿计的物联网设备。

It is desirable that the connectivity options be adapted to the IP layer. To that end, IETF has completed considerable work to adapt Bluetooth®, Wi-Fi, 802.15.4, LPWAN, etc. to IPv6. These Adaptations, plus the larger address space and improved address management capabilities, makeIPv6 the clear choice for the OCF network layer technology.
连接选项最好适配于IP层。为此，IETF已经完成了大量工作，以使 Bluetooth®, Wi-Fi, 802.15.4, LPWAN等等适配 IPv6。这些调整，加上更大的地址空间和改进的地址管理功能，使IPv6成为OCF网络层技术的明确选择。

9.2 Architecture 架构

While the aging IPv4 centric network has evolved to support complex topologies, its deployment was primarily provisioned by a single Internet Service Provider (ISP) as a single network. More Complex network topologies, often seen in residential home, are mostly introduced through the acquisition of additional home network devices, which rely on technologies like private Network Address Translation (NAT). These technologies require expert assistance to set up correctly and should be avoided in a home network as they most often result in breakage of constructs like routing, naming and discovery services.
尽管老化的IPv4中心网络已经发展到可以支持复杂的拓扑结构，但它的部署主要由单个互联网服务提供商（ISP）作为单个网络提供。更复杂的网络拓扑结构，通常出现在住宅中，主要是通过购买额外的家庭网络设备引入的，这些设备依赖于私有网络地址转换（NAT）等技术。这些技术需要专家的帮助才能正确设置，在家庭网络中应该避开这些技术，因为它们通常会导致路由、命名和发现服务等结构的破坏。

The multi-segment ecosystem OCF addresses will not only cause a proliferation of new devices and associated routers, but also new services introducing additional edge routers. All these new requirements require advance architectural constructs to address complex network topologies like the one shown in Figure 9.
多段生态系统OCF地址不仅会导致新设备和相关路由器的激增，而且还会带来引入更多边缘路由器的新服务。所有这些新需求都需要高级的体系结构构造来处理复杂的网络拓扑，如图9所示。

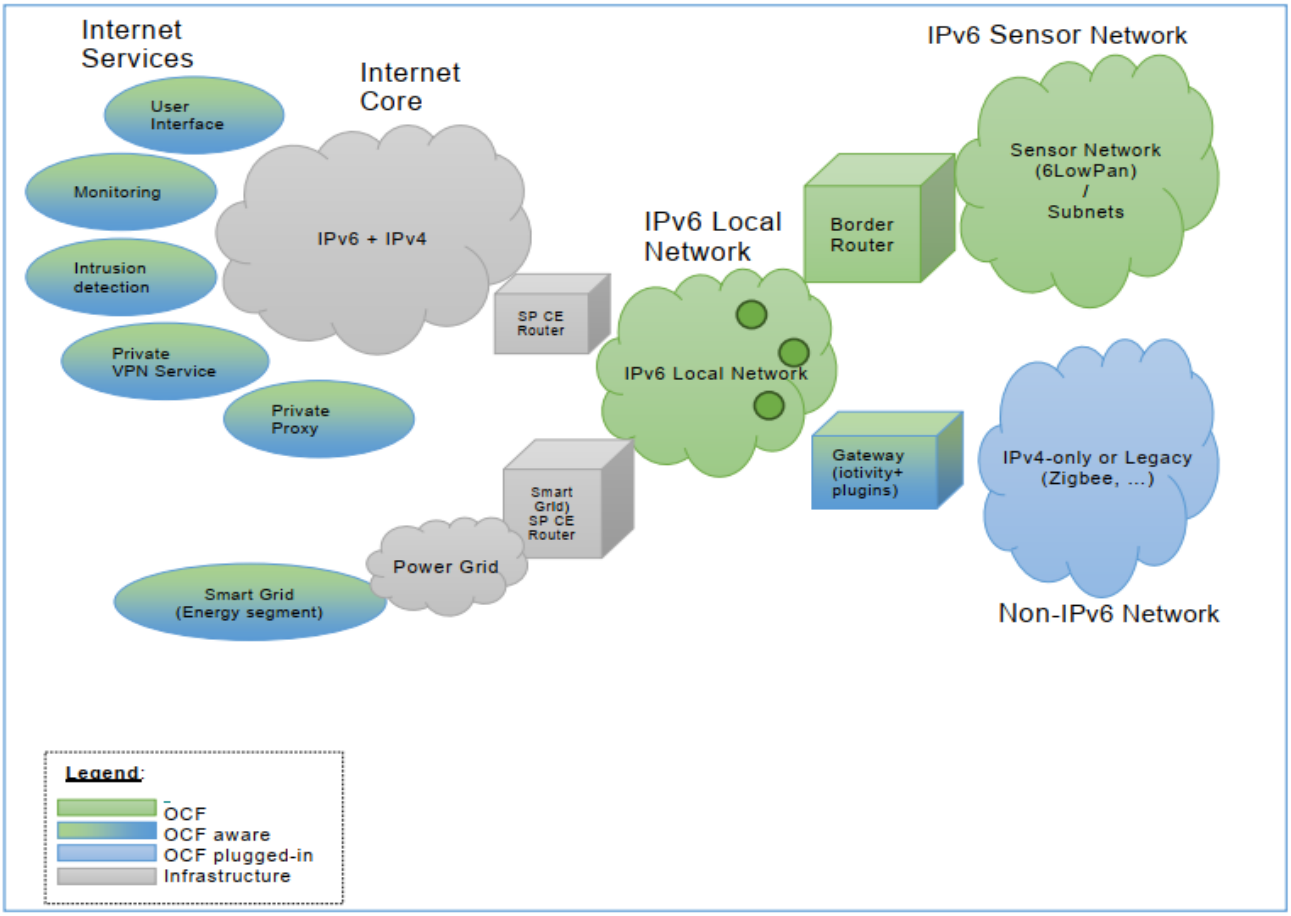

Figure 9 - High Level Network & Connectivity Architecture
图9-高级网络和连接架构

In terms of IETF RFC 6434, IPv6 nodes assume either a router or host role. Nodes may further implement various specializations of those roles:
就IETF RFC 6434而言，IPv6节点承担路由器或主机角色。节点可以进一步实现这些角色的各种特有功能：

- A Router may implement Customer Edge Router capabilities as defined in IETF RFC 7084.
  路由器可以实现IETF RFC 7084中定义的客户端路由器功能。
- Nodes limited in processing power, memory, non-volatile storage or transmission capacity requires special IP adaptation layers (6LoWPAN) and/or dedicated routing protocols (RPL).Examples include devices transmitting over low power physical layer like IEEE 802.14.5, ITUG9959, Bluetooth Low Energy, DECT Ultra Low Energy, and Near Field Communication (NFC).
  处理能力、内存、非易失性存储或传输能力有限的节点需要特殊的IP适配层（6LoWPAN）和/或专

用路由协议（RPL）。例如，通过低功耗物理层传输的设备，如IEEE 802.14.5、ITU G9959、蓝牙低功耗、DECT超低功耗和近场通信（NFC）。

- A node may translate and route messaging between IPv6 and non-IPv6 networks.
  节点可以转换和路由的IPv6和非IPv6网络之间的消息。

9.3 IPv6 network layer requirements IPv6网络层的需求

9.3.1 Introduction 引言

Projections indicate that many 10s of billions of new IoT endpoints and related services will be brought online in the next few years. These endpoint's capabilities will span from battery powered nodes with limited compute, storage, and bandwidth to more richly resourced devices operating over Ethernet and WiFi links.
预测显示，在未来几年内，将有数十亿个新的IoT端点和相关服务上线。这些端点的功能将从计算、存储和带宽有限的电池供电节点扩展到通过以太网和WiFi链接操作的资源更丰富的设备。

Internet Protocol version 4 (IPv4), deployed some 30 years ago, has matured to support a wide variety of applications such as Web browsing, email, voice, video, and critical system monitoring and control. However, the capabilities of IPv4 are at the point of exhaustion, not the least of which is that available address space has been consumed.
互联网协议版本4（IPv4）部署于大约30年前，现已成熟，可以支持各种应用程序，如Web浏览、电子邮件、语音、视频和关键系统监视和控制。然而，IPv4的功能临近耗尽，至少可用地址空间已经耗尽。

The IETF long ago saw the need for a successor to IPv4, thus the development of IPv6.OCF recommends IPv6 at the network layer. Amongst the reasons for IPv6 recommendations are:
IETF很久以前就意识到需要IPv4的后继产品，从而发展了IPv6。OCF建议在网络层使用IPv6。推荐IPv6的原因包括：

- Larger address space. Side-effect: greatly reduce the need for NATs.
  更大的地址空间。副作用:大大减少了对NAT的需求。
- More flexible addressing architecture. Multiple addresses and types per interface: Link-local,ULA, GUA, variously scoped Multicast addresses, etc. Better ability to support multi-homed networks, better re-numbering capability, etc.
  更灵活的寻址架构。每个接口有多个地址和类型:本地链接、ULA、GUA、不同作用域的多播地址等。更好的支持多家庭网络的能力，更好的重新编码能力等等。
- More capable auto configuration capabilities: DHCPv6, SLAAC, Router Discovery, etc.
  更强大的自动配置功能:DHCPv6、SLAAC、路由器发现等。
- Technologies enabling IP connectivity on constrained nodes are based upon IPv6.
  在受限制的节点上实现IP连接的技术基于IPv6。
- All major consumer operating systems (IoS, Android, Windows, Linux) are already IPv6 enabled.
  所有主要的消费者操作系统(IoS, Android, Windows, Linux)已经启用了IPv6。

- Major Service Providers around the globe are deploying IPv6.
  全球主要的服务提供商正在部署IPv6。

9.3.2 IPv6 node requirements IPv6节点需求

9.3.2.1 Introduction 引言

In order to ensure network layer services interoperability from node to node, mandating a common network layer across all nodes is vital. The protocol should enable the network to be: secure,manageable, and scalable and to include constrained and self-organizing meshed nodes. OCFmandates IPv6 as the common network layer protocol to ensure interoperability across all Devices.More capable Devices may also include additional protocols creating multiple-stack Devices. The Remainder of this clause will focus on interoperability requirements for IPv6 hosts, IPv6 constrained hosts and IPv6 routers. The various protocol translation permutations included in multi-stack gateway devices may be addresses in subsequent addendums of this document.
为了确保网络层服务在节点之间的互操作性，在所有节点之间指定一个公共网络层是至关重要的。协议应该使网络能够：安全、可管理和可伸缩，并包括受约束和自组织的网状节点。OCF应用IPv6作为公共网络层协议，以确保所有设备之间的互操作性。功能更强的设备还可能包括创建多堆栈设备的附加协议。这一条款的其余部分将集中在IPv6主机的互操作性需求，IPv6受限主机和IPv6路由器。多堆栈网关设备中包含的各种协议转换排列可能是本文档后续附录中的地址。

9.3.2.2 IP Layer IP层

An IPv6 node shall support IPv6 and it shall conform to the requirements as specified in IETF RFC 6434.
IPv6节点应支持IPv6，并应符合IETF RFC 6434中指定的要求。

10 OCF Endpoint OCF 端点

10.1 OCF Endpoint definition OCF端点定义

The specific definition of an OCF Endpoint depends on the Transport Protocol Suite being used. For the example of CoAP over UDP over IPv6, the OCF Endpoint is identified by an IPv6 address and UDP port number.
OCF端点的特定定义取决于所使用的传输协议套件。例如，基于IPv6的UDP上的CoAP, OCF端点由IPv6地址和UDP端口号标识。

Each Device shall associate with at least one OCF Endpoint with which it can exchange request and response messages. When a message is sent to an OCF Endpoint, it shall be delivered to the Device which is associated with the OCF Endpoint. When a request message is delivered to an OCF Endpoint, path component is enough to locate the target Resource.
每个设备应与至少一个OCF端点关联，以便与之交换请求和响应消息。当消息被发送到OCF端点时，它将被发送到与OCF端点相关联的设备。当将请求消息传递到OCF端点时，路径组件足以定位目标资源。

A Device can be associated with multiple OCF Endpoints. For example, n Device can have several IP addresses or port numbers or support both CoAP and HTTP transfer protocol. Different Resources in Device may be accessed with the same OCF Endpoint or need different ones. Some Resources may use one OCF Endpoint and others a different one. It depends on an implementation.
一个设备可以与多个OCF端点关联。例如，n设备可以有多个IP地址或端口号，或者同时支持CoAP和HTTP传输协议。设备中的不同资源可以使用相同的OCF端点访问，也可以使用不同的端点访问。一些资源可能使用一个OCF端点，而另一些则使用另一个端点。它取决于实现。

On the other hand, an OCF Endpoint can be shared among multiple Devices, only when there is a way to clearly designate the target Resource with request URI. For example, when multiple CoAP servers use uniquely different URI paths for all their hosted Resources, and the CoAP implementation demultiplexes by path, they can share the same CoAP OCF Endpoint. However,this is not possible in this version of the document, because a pre-determined URI (e.g. "/oic/d") is mandatory for some mandatory Resources (e.g. "oic.wk.d").
另一方面，OCF端点可以在多个设备之间共享，只有在能够使用请求URI明确指定目标资源的情况下才可以共享。例如，当多个CoAP服务器为其所有托管资源使用唯一不同的URI路径时，CoAP实现通过路径进行多路解复用时，它们可以共享相同的CoAP OCF端点。但是，在这个版本的文档中，这是不可能的，因为一个预先确定的URI（例：“/oic/d”）是一些强制性资源（例：“oic.wk.d”）。

10.2 OCF Endpoint information OCF 端点信息

10.2.1 Introduction 信息

OCF Endpoint is represented by OCF Endpoint information which consists of two items of key-value pair, "ep" and "pri".
OCF端点由OCF端点信息表示，OCF端点信息由键值对“ep”和“pri”两项组成。

10.2.2 "ep"

"ep" represents Transport Protocol Suite and OCF Endpoint Locator specified as follows:
"ep"表示指定的传输协议套件和OCF端点定位器，如下所示：

- Transport Protocol Suite - a combination of protocols (e.g. CoAP + UDP + IPv6) with which request and response messages can be exchanged for RESTful transaction (i.e. CRUDN). A Transport Protocol Suite shall be indicated by a URI scheme name. All scheme name supported by this documentare IANA registered, these are listed in Table 19. A vendor may also make use of a non-IANA registered scheme name for their own use (e.g."com.example.foo"), this shall follow the syntax for such scheme names defined by IETF RFC 7595. The behavior of a vendor-defined scheme name is undefined by this document. All OCF defined Resource Types when exposing OCF Endpoint Information in an"eps" (see 10.2.4) shall include at least one "ep" with a Transport Protocol Suite as defined in Table 19.

传输协议套件-协议的组合(如CoAP + UDP + IPv6)，其中的请求和响应消息可以交换RESTful事务(即CRUDN)。传输协议套件应由URI方案名称表示。此文档支持的所有方案名称都是注册的IANA，它们列在表19中。供应商亦可使用非iana注册计划名称作自用（例如:"com.example.foo"）这应该遵循IETF RFC 7595定义的这种方案名称的语法。此文件未定义供应商定义的方案名称的行为。在"eps"（见10.2.4）中公开OCF端点信息时，所有OCF定义的资源类型都应包括至少一个带有表19中定义的传输协议集的"ep"。

- OCF Endpoint Locator - an address (e.g. IPv6 address + Port number) or an indirect identifier(e.g., DNS name) resolvable to an IP address, through which a message can be sent to theOCF Endpoint and in turn associated Device. The OCF Endpoint Locator for "coap" and "coaps"shall be specified as "IP address: port number". The OCF Endpoint Locator for "coap+tcp" or"coaps+tcp" shall be specified as "IP address: port number" or "DNS name: port number" or"DNS name" such that the DNS name shall be resolved to a valid IP address for the targetResource with a name resolution service (i.e., DNS). For the 3rd case, when the port number is omitted, the default port "5683" (and "5684") shall be assumed for "coap+tcp" (and for"coaps+tcp") scheme respectively as defined in IETF RFC 8323.Temporary addresses should not be used because OCF Endpoint Locators are for the purpose of accepting incoming sessions, whereas temporary addresses are for initiating outgoing sessions (IETF RFC 4941).Moreover, its inclusion in "/oic/res" can cause a privacy concern (IETF RFC 7721).
  OCF端点定位器-一个地址（如IPv6地址+端口号）或一个可解析为IP地址的间接标识符(如DNS名称)，通过它可以将消息发送到OCF端点，进而关联设备。"coap"和"coaps"的OCF端点定位符应指定为"IP地址:端口号"。"coap+tcp"或"coaps+tcp"的OCF端点定位符应指定为"IP地址:端口号"或"DNS名称:端口号"或"DNS名称"，以便通过名称解析服务（即,DNS）。对于第三种情况，当端口号被省略时，IETF RFC 8323中定义的"coap+tcp"（和"coaps+tcp"）方案分别假定默认端口"5683"（和"5684"）。不应该使用临时地址，因为OCF端点定位器用于接收传入的会话，而临时地址用于初始化传出的会话（IETF RFC 4941）。此外，它包含在"/oic/res"中可能会引起隐私问题（IETF RFC 7721）。

"ep" shall have as its value a URI (as specified in IETF RFC 3986) with the scheme component indicating Transport Protocol Suite and the authority component indicating the OCF EndpointLocator.
"ep"的值应该是一个URI（在IETF RFC 3986中指定），其中scheme组件表示传输协议套件，authority组件表示OCF端点定位器。

An "ep" example for "coap" and "coaps" is as illustrated:
"coap"和"coaps"的"ep"示例如下：

```
"ep": "coap://[fe80::b1d6]:1111"
```

An "ep" example for "coap+tcp" and "coaps+tcp" is as illustrated:
"coap+tcp"和"coaps+tcp"的"ep"示例如下：

```
"ep": "coap+tcp://[2001:db8:a::123]:2222"
"ep": "coap+tcp://foo.bar.com:2222"
"ep": "coap+tcp://foo.bar.com"
```

The current list of "ep" with corresponding Transport Protocol Suite is shown in Table 19:
当前的“ep”及其对应的传输协议套件列表如表19所示:

Table 19 - "ep" value for Transport Protocol Suite
表19 - 传输协议套件的“ep”值

Transport Protocol Suite **传输协议套件**	**scheme** **组合**	**OCF Endpoint Locator** **OCF端点定位器**	**"ep" Value example** **"ep"值示例**
coap+udp+ip	"coap"	IP address + port number IP地址+端口号	"coap://[fe80::b1d6]:1111"
coaps + udp + ip	"coaps"	IP address + port number IP地址+端口号	"coaps://[fe80::b1d6]:1122"
coap + tcp + ip	"coap+tcp"	IP address + port number IP地址+端口号 DNS name: port number DNS名称:端口号 DNS name DNS名称	"coap+tcp://[2001:db8:a::123]:2222" "coap+tcp://foo.bar.com:2222" "coap+tcp://foo.bar.com"
coaps + tcp + ip	"coaps+tcp"	IP address + port number IP地址+端口号 DNS name: port number DNS名称:端口号 DNS name DNS名称	"coaps+tcp://[2001:db8:a::123]:2233" "coaps+tcp://[2001:db8:a::123]:2233" "coaps+tcp://foo.bar.com:2233"

10.2.3 "pri"

When there are multiple OCF Endpoints, "pri" indicates the priority among them.
当有多个OCF端点时，“pri”表示其中的优先级。

"pri" shall be represented as a positive integer (e.g. "pri": 1) and the lower the value, the higher the priority.
“pri”应表示为正整数（例如:“pri”:1）并且值越低，优先级越高。

The default "pri" value is 1, i.e. when "pri" is not present, it shall be equivalent to "pri": 1.
默认的“pri”值为1，即当“pri”不存在时，等同于“pri”:1。

10.2.4 OCF Endpoint information in "eps"Parameter 参数"eps"中的OCF端点信息

To carry OCF Endpoint information, a new Link Parameter "eps" is defined in 7.8.2.5.6. "eps" has an array of items as its value and each item represents OCF Endpoint information with two key-value pairs, "ep" and "pri", of which "ep" is mandatory and "pri" is optional.OCF Endpoint Information in an "eps"

Parameter is valid for the target Resource of the Link, i.e.,the Resource referred by "href" Parameter. OCF Endpoint information in an "eps" Parameter may be used to access other Resources on the Device, but such access is not guaranteed.
为了携带OCF端点信息，在7.8.2.5.6中定义了一个新的链接参数“eps”。“eps”的值是一个项目数组，每个项目用两个键值对表示OCF端点信息，“ep”和“pri”，其中“ep”是强制的，“pri”是可选的。“eps”参数中的OCF端点信息对于链接的目标资源是有效的，即，由“href”参数引用的资源。“eps”参数中的OCF端点信息可被用于访问设备上的其他资源，但这种访问不能被保证。

A Client may resolve the "ep" value to an IP address for the target Resource, i.e., the address to access the Device which hosts the target Resource. A valid (transfer protocol) URI for the target Resource can be constructed with the scheme, host and port components from the "ep" value andthe "path" component from the "href" value.
客户端可以将“ep”值解析为目标资源的IP地址，即，用于访问承载目标资源的设备的地址。目标资源的有效（传输协议）URI可以由scheme、来自“ep”值的主机和端口组件以及来自“href”值的“path”组件构造。

Links with an"eps":
带有"eps"链接：

```
{
   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9 ",
   "href": "/myLightSwitch",
   "rt": ["oic.r.switch.binary"],
   "if": ["oic.if.a", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [
      {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
       {"ep": "coaps://[fe80::b1d6]:1122"}
   ]
}
{
   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
   "href": "/myTemperature",
   "rt": ["oic.r.temperature"],
   "if": ["oic.if.a", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [
      {"ep": "coap+tcp://foo.bar.com", "pri": 2},
      {"ep": "coaps+tcp://foo.bar.com:1122"}
]
}
```

In the previous example, "anchor" represents the hosting Device, "href", target Resource and "eps" the two OCF Endpoints for the target Resource. The (fully-qualified) URIs for the target Resource are as illustrated:

在前面的例子中，“anchor”表示主机设备，“href”表示目标资源，“eps”表示目标资源的两个OCF端点。目标资源的（完全限定的）uri如下图所示：

```
coap://[fe80::b1d6]:1111/myLightSwitch
coaps://[fe80::b1d6]:1122/myLightSwitch
coap+tcp://foo.bar.com:5683/myTemperature
coaps+tcp://foo.bar.com:1122/myTemperature
```

If the target Resource of a Link requires a secure connection (e.g. CoAPS), "eps" Parameter shall be used to indicate the necessary information (e.g. port number) in OCF 1.0 payload. For optional backward compatibility with OIC 1.1, the "sec" and "port" shall only be used in OIC 1.1 payload.
如果一个链接的目标资源需要一个安全的连接（如CoAPS），“eps”参数应用来指示OCF 1.0有效载荷中必要的信息（如端口号）。对于可选的向后兼容OIC 1.1，“sec”和“port”只能用于OIC 1.1有效载荷。

10.3 OCF Endpoint discovery OCF端点发现

10.3.1 Introduction 引言

OCF Endpoint discovery is defined as the process for a Client to acquire the OCF Endpoint information for Device or Resource.
OCF端点发现定义为客户端获取设备或资源的OCF端点信息的过程。

10.3.2 Implicit discovery 隐式发现

If a Device is the source of a CoAP message (e.g. "/oic/res"response), the source IP address and port number may be combined to form the OCF Endpoint Locator for the Device. Along with a "coap" scheme and default "pri" value, OCF Endpoint information for the Device may be constructed.
如果一个设备是CoAP消息的源头（例如，"/oic/res"响应），源IP地址和端口号可以组合起来形成设备的OCF端点定位信息。除了“coap”模式和默认的“pri”值之外，还可以构造设备的OCF端点信息。

In other words, a "/oic/res" response message with CoAP may implicitly carry the OCF Endpoint information of the responding Device and in turn all the hosted Resources, which may be accessed with the same transfer protocol of CoAP. In the absence of an "eps" Parameter, a Client shall be able to utilize implicit discovery to access the target Resource.
换句话说，带有CoAP的“/oic/res”响应消息可以隐式地携带响应设备的OCF端点信息，并依次携带所有托管资源，这些资源可以通过相同的CoAP传输协议访问。在没有“eps”参数的情况下，客户端应该能够通过隐式发现来访问目标资源。

10.3.3 Explicit discovery with "/oic/res" response 带有“/oic/res”响应的显式发现

OCF Endpoint information may be explicitly indicated with the "eps" Parameter of the Links in "/oic/res".
OCF端点信息可以用“/oic/res”中链接的“eps”参数明确表示。

As in 10.3.2, an "/oic/res" response may implicitly indicate the OCF Endpoint information for some Resources hosted by the responding Device. However implicit discovery, i.e., inference of OCF Endpoint information from CoAP response message, may not work for some Resources on the same Device. For example, some Resources may allow only secure access via CoAPS which requires the "eps" Parameter to indicate the port number. Moreover "/oic/res" may expose a target Resource which belongs to another Device.
与10.3.2中一样，“/oic/res”响应可以隐式地指示由响应设备承载的某些资源的OCF端点信息。然而隐式发现，即，从CoAP响应消息推断OCF端点信息可能不适用于同一设备上的某些资源。例如，一些资源可能只允许通过CoAPS进行安全访问，而CoAPS需要“eps”参数来指示端口号。此外，“/oic/res”可能会公开属于另一个设备的目标资源。

When the OCF Endpoint for a target Resource of a Link cannot be implicitly inferred, the "eps" Parameter shall be included to provide explicit OCF Endpoint information with which a Client can access the target Resource. In the presence of the "eps" Parameter, a Client shall be able to utilize it to access the target Resource. For "coap" and "coaps", a Client may use the IP address in the"ep" value in the "eps" Parameter to access the target Resource. For "coap+tcp" and "coaps+tcp",a Client may use the IP address in the "eps" Parameter or resolve the DNS name in the "eps"Parameter to acquire a valid IP address for the target Resource. If "eps" Parameter omits the port number, then the default port "5683" (and "5684") shall be assumed for "coap+tcp" (and"coaps+tcp") scheme as defined in IETF RFC 8323.To access the target Resource of a Link, a Client may use the "eps" Parameter in the Link, if it is present and fall back on implicit discovery ifnot.
当无法隐式推断链接的目标资源的OCF端点时，应包含“eps”参数，以提供客户端可以访问目标资源的显式OCF端点信息。在“eps”参数存在的情况下，客户端应该能够利用它来访问目标资源。对于“coap”和“coaps”，客户端可以使用“eps”参数中的“ep”值中的IP地址来访问目标资源。对于“coap+tcp”和“coaps+tcp”，客户端可以使用“eps”参数中的IP地址或解析“eps”参数中的DNS名称来获取目标资源的有效IP地址。如果“eps”参数省略了端口号，则IETF RFC 8323中定义的“coap+tcp”（和“coaps+tcp”）方案的默认端口为“5683”（和“5684”）。要访问链接的目标资源，客户端可以使用链接中的“eps”参数（如果存在），如果不存在则使用隐式发现。

This is an example of an "/oic/res" response from a Device having the "eps" Parameter in Links.
这是一个来自具有链接中的“eps”参数的设备的“/oic/res”响应的示例。


```
[
   {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/res",
      "rel": "self",
      "rt": ["oic.wk.res"],
      "if": ["oic.if.ll", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
```

```
            {"ep": "coap://[2001:db8:a::b1d4]:55555"},
            {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
   },
   {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/d",
      "rt": ["oic.wk.d"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
         {"ep": "coap://[2001:db8:a::b1d4]:55555"},
         {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
   },
   {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/p",
      "rt": ["oic.wk.p"],
      "if": ["oic.if.r", "oic.if.baseline"],
      "p": {"bm": 3},
      "eps": [
          {"ep": "coap://[2001:db8:a::b1d4]:55555"},
          {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
   },
   {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/sec/doxm",
      "rt": ["oic.r.doxm"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
         {"ep": "coap://[2001:db8:a::b1d4]:55555"},
         {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
      ]
   },
   {
      "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
      "href": "/oic/sec/pstat",
      "rt": ["oic.r.pstat"],
      "if": ["oic.if.baseline"],
      "p": {"bm": 1},
      "eps": [
         {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
```

```
    ]
},
{
   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
   "href": "/oic/sec/cred",
   "rt": ["oic.r.cred"],
   "if": ["oic.if.baseline"],
   "p": {"bm": 1},
   "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
   ]
},
{
   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
   "href": "/oic/sec/acl2",
   "rt": ["oic.r.acl2"],
   "if": ["oic.if.baseline"],
   "p": {"bm": 1},
   "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
   ]
},
{
   "anchor": "ocf://e61c3e6b-9c54-4b81-8ce5-f9039c1d04d9",
   "href": "/myIntrospection",
   "rt": ["oic.wk.introspection"],
   "if": ["oic.if.r", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:11111"}
   ]
},
{
   "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989",
   "href": "/myLight",
   "rt": ["oic.r.switch.binary"],
   "if": ["oic.if.a", "oic.if.baseline"],
   "p": {"bm": 3},
   "eps": [
      {"ep": "coaps://[2001:db8:a::b1d4]:22222"}
   ]
}
]
```

The exact format of the "/oic/res" response and a way for a Client to acquire a "/oic/res" responsemessage is specified in Annex A and 11.2.4 respectively.
“/oic/res”响应的确切格式和客户端获取“/oic/res”响应消息的方式分别在附件a和11.2.4中规定。

11 Functional interactions 功能联系

11.1 Introduction 引言

The functional interactions between a Client and a Server are described in 11.1 through 11.4respectively. The functional interactions use CRUDN messages (clause 8) and include Discovery,Notification, and Device management. These functions require support of core defined Resources as defined in Table 20.
客户端和服务器之间的功能交互分别在11.1到11.4中描述。功能交互使用CRUDN消息（第8章），包括发现、通知和设备管理。这些功能需要支持表20中定义的核心定义资源。

Table 20 - List of Core Resources
表20 - 核心资源列表

Pre-defined URI 预定义URI	Resource Name 资源	Resource Type 资源类型	Related Functional Interaction 相关功能的相互作用	Mandatory 强制性
"/oic/res"	Default 默认	"oic.wk.res"	Discovery发现	Yes
"/oic/p"	Platform 平台	"oic.wk.p"	Discovery 发现	Yes
"/oic/d"	Device服务	"oic.wk.d"	Discovery 发现	Yes
Implementationdefined 实施定义	Introspection 内省	"oic.wk.introspection"	Introspection 内省	Yes

11.2 Resource discovery 资源发现

11.2.1 Introduction 引言

Discovery is a function which enables OCF Endpoint discovery as well as Resource based discovery. OCF Endpoint discovery is described in detail in clause 10. This clause mainly describes the Resource based discovery.
发现功能支持OCF端点发现和基于资源的发现。OCF端点发现在第10章中详细描述。这个章节主要描述基于资源的发现。

11.2.2 Resource based discovery: mechanisms 资源基础发现：机制

11.2.2.1 Overview 综述

As part of discovery, a Client may find appropriate information about other OCF peers. This Information could be instances of Resources, Resource Types or any other information represented in the Resource model that an OCF peer would want another OCF peer to discover.
作为发现的一部分，客户端可以找到关于其他OCF对等点的适当信息。此信息可以是资源实例、资源类型或资源模型中表示的任何其他信息，OCF对等点希望另一个OCF对等点发现这些信息。

At the minimum, Resource based discovery uses the following:
基于资源的发现至少使用以下方法：

- A Resource to enable discovery shall be defined. The representation of that Resource shall contain the information that can be discovered.
  应定义允许发现的资源。该资源的表示应包含可被发现的信息。
- The Resource to enable discovery shall be specified and commonly known a-priori. A Device for hosting the Resource to enable discovery shall be identified.
  可进行发现的资源应预先指定并为一般所知。应确定托管资源以支持发现的设备。
- A mechanism and process to publish the information that needs to be discovered with the Resource to enable discovery.
  发布需要与资源一起被发现的信息的机制和过程，以支持发现。
- A mechanism and process to access and obtain the information from the Resource to enable discovery. A query may be used in the request to limit the returned information.
  一种机制和过程，以访问和获取资源中的信息，从而使发现成为可能。请求中可以使用查询来限制返回的信息。
- A scope for the publication.
  出版范围。
- A scope for the access.
  访问范围。
- A policy for visibility of the information.
  信息可见性的政策。

Depending on the choice of the base aspects, the Framework defines three Resource based discovery mechanisms:
根据基本方面的选择，框架定义了三种基于资源的发现机制:

- Direct discovery, where the Resources are published locally at the Device hosting the Resources and are discovered through peer inquiry.
  直接发现，其中资源被发布在本地的设备托管资源，并通过对等查询发现。

- Indirect discovery, where Resources are published at a third party assisting with the discovery and peers publish and perform discovery against the Resource to enable discovery on the assisting 3rd party.
  间接发现，即在第三方发布资源以协助发现，而对等方发布并对资源执行发现，以便在第三方上启用发现。
- Advertisement discovery, where the Resource to enable discovery is hosted local to the initiator of the discovery inquiry but remote to the Devices that are publishing discovery information.
  广告发现，其中用于启用发现的资源驻留在本地的发现查询发起者处，但远程驻留在发布发现信息的设备处。

A Device shall support direct discovery.
设备应支持直接发现。

11.2.2.2 Direct discovery 直接发现

In direct discovery,
在直接发现中，

- The Device that is providing the information shall host the Resource to enable discovery.
  提供信息的设备应承载资源以支持发现。
- The Device publishes the information available for discovery with the local Resource to enable discovery (i.e. local scope).
  设备使用本地资源发布可用于发现的信息，以启用发现（即本地范围）。
- Clients interested in discovering information about this Device shall issue RETRIEVE requests directly to the Resource. The request may be made as a unicast or multicast. The request maybe generic or may be qualified or limited by using appropriate queries in the request.
  有兴趣发现此设备信息的客户端应直接向资源发出检索请求。请求可以是单播的，也可以是多播的。请求可以是通用的，也可以通过在请求中使用适当的查询进行限定或限制。
- The Server Device that receives the request shall send a response with the discovered information directly back to the requesting Client Device.
  接收请求的服务器设备应将发现的信息直接发送回请求的客户端设备。
- The information that is included in the request is determined by the policies set for the Resource To be discovered locally on the responding Device.
  请求中包含的信息由为在响应设备上本地发现的资源设置的策略决定。

11.2.3 Resource based discovery: Finding information 基于资源的发现：查找信息

The discovery process (Figure 10) is initiated as a RETRIEVE request to the Resource to enable discovery. The request may be sent to a single Device (as in a Unicast) or to multiple Devices (as in Multicast). The specific mechanisms used to do Unicast or Multicast are determined by the support in the data connectivity layer. The response to the request has the information to be discovered based on the policies for that information. The policies can determine which information is shared, when and to which

requesting agent. The information that can be discovered can be Resources, types, configuration and many other standards or custom aspects depending on the request to appropriate Resource and the form of request. Optionally the requester may narrow the information to be returned in the request using query parameters in the URI query.
发现过程（图10）作为对资源的检索请求初始化，以启用发现。请求可以发送到单个设备（如单播），也可以发送到多个设备（如多播）。用于单播或多播的特定机制由数据连接层中的支持决定。请求的响应具有根据该信息的策略发现的信息。策略可以确定共享哪些信息、何时共享以及向哪个请求代理发出请求。可以发现的信息可以是资源、类型、配置和许多其他标准或自定义方面，这取决于对适当资源的请求和请求的形式。请求者可以选择使用URI查询中的查询参数来缩小请求中要返回的信息的范围。

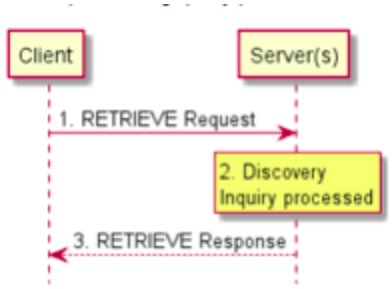

Figure 10 - Resource based discovery: Finding information
图10-基于资源的发现：发现信息

Discovery Resources 发现资源

The following Core Resources shall be implemented on all Devices to support discovery:
，在所有设备上必须实现以下核心资源以支持发现:

- "/oic/res" for discovery of Resources.
  “/oic/res”用于发现资源。
- "/oic/p" for discovery of Platform.
  “/oic/p”用于发现平台。
- "/oic/d" for discovery of Device information.
  “/oic/d”用于发现设备信息。

Devices shall expose each of "/oic/res", "/oic/d", and "/oic/p" via an unsecured OCF Endpoint. Further details for these mandatory Core Resources are described in Table 21.
设备应通过不安全的OCF端点公开“/oic/res”、“/oic/d”和“/oic/p”。这些强制性核心资源的详细情况见表21。

Platform Resource平台资源

The OCF recognizes that more than one instance of Device may be hosted on a single Platform. Clients need a way to discover and access the information on the Platform. The Core Resource, "/oic/p" exposes Platform specific Properties. All instances of Device on the same Platform shall have the same values of

any Properties exposed (i.e. a Device may choose to expose optional Properties within "/oic/p" but when exposed the value of that Property should be the same as the value of that Property on all other Devices on that Platform).
OCF认识到可以在一个平台上托管多个设备实例。客户端需要一种方法来发现和访问平台上的信息。核心资源“/oic/p”公开了平台的特定属性。同一平台上的所有设备实例应具有相同的公开属性值（例如，一个设备可以选择在“/oic/p”中公开可选属性，但当公开时，该属性的值应该与该平台上所有其他设备上该属性的值相同）。

Device Resource 设备资源

The Device Resource shall have the pre-defined URI "/oic/d", the Device Resource shall expose the Properties pertaining to a Device as defined in Table 24. The Device Resource shall have a default Resource Type that helps in bootstrapping the interactions with the Device (the default type is described in Table 21).The Device Resource may have one or more Resource Type(s) that are specific to the Device in addition to the default Resource Type or if present overriding the default Resource Type. The base Resource Type "oic.wk.d" defines the Properties that shall be exposed by all Devices. The Device specific Resource Type(s) exposed are dependent on the class of Device (e.g. air conditioner, smoke alarm, etc. Since all the Resource Types of "/oic/d" are not known a priori, the Resource Type(s) of "/oic/d" are determined by discovery through the Core Resource "/oic/res".
设备资源应具有预定义的URI“/oic/d”，设备资源应公开表24中定义的设备的属性。设备资源应该有一个默认的资源类型，帮助引导与设备的交互（默认类型在表21中描述）。除了默认资源类型外，设备资源可以有一个或多个特定于设备的资源类型，如果存在则覆盖默认资源类型。基本资源类型“oic.wk.d”定义了所有设备应公开的属性。公开的设备特定资源类型依赖于设备的类别（例如:空调、烟雾报警器等）。由于“/oic/d”的所有资源类型都不是先验已知的，因此“/oic/d”的资源类型是通过核心资源“/oic/res”的发现来确定的。

Table 21 - Mandatory discovery Core Resources
表21-强制性发现核心资源

Pre-defined URI 预定义的URI	**Resource Type Title 资源类型标题**	**Resource TypeID ("rt" value) 资源类型ID ("rt" value)**	**OCF Interfaces OCF 接口**	**Description 描述**	**Related Functional Interaction 相关功能的相互作用**
"/oic/res"	Default 默认	"oic.wk.res"	"oic.if.ll"	The Resource through which the corresponding Server is discovered and introspected for available Resources. 通过该资源发现相应的服务器并对可用资源进行内省。 "/oic/res" shall expose the Resources that are discoverable on a Device. When a Server receives a RETRIEVE request targeting "/oic/res" (e.g., "GET /oic/res"), it shall respond with the links list of all the Discoverable Resources of itself. The "/oic/d" and "/oic/p" are Discoverable Resources, hence	Discovery 发现

				their links are included in "/oic/res" response. The Properties exposed by "/oic/res" are listed in Table 22. “/oic/res”将公开设备上可发现的资源。当服务器接收到针对“/oic/res”的检索请求时(例如，“GET /oic/res”)，它将反应所有可发现链接列表的资源。“/oic/d”和“/oic/p”是可以发现的资源，因此它们的链接包括在“/oic/res”响应中。“/oic/res”公布的属性列于表22。	
"/oic/p"	Platform 平台	"oic.wk.p"	"oic.if.r"	The Discoverable Resource through which Platform specific information is discovered. 用于发现平台特定信息的可发现资源。 The Properties exposed by "/oic/p" are listed in Table 25 表25列出了“/oic/p”所揭示的特性。	Discovery 发现
"/oic/d"	Device 设备	"oic.wk.d" and/or one or more Device Specific Resource Type ID(s) "oic.wk.d"和/或一个或多个设备特定资源类型ID(s)	"oic.if.r"	The discoverable via "/oic/res" Resource which exposes Properties specific to the Device instance. 通过“/oic/res”资源公开设备实例的特定属性。 The Properties exposed by "/oic/d" are listed in Table 24. 表24列出了“/oic/d”所揭示的特性。	Discovery 发现

Table 22 defines "oic.wk.res" Resource Type.

表22定义了“oic.wk.res”资源类型。

Table 22 - "oic.wk.res" Resource Type definition

表22-"oic.wk.res"资源类型定义

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
Name 名称	"n"	string	N/A	N/A	R	No 否	Human-friendly name defined by the vendor 由供应商定义人性化的名称
Links 链接	"links"	array	See 7.8.2	N/A	R	Yes 是	The array of Links describes the URI, supported Resource Types and OCF Interfaces, and access policy. 链接数组描述URI、支持的资源类型和OCF接口，以及访问策略。

A Device shall support CoAP based discovery as the baseline discovery mechanism (see 11.2.5).
设备应支持基于CoAP的发现作为基线发现机制（见11.2.5）。

The "/oic/res" shall list all Resources that are indicated as discoverable (see 11.2). Also the following architecture Resource Types shall be listed:
“/oic/res”应列出所有指示为可发现的资源（见11.2）。还应列出以下架构资源类型:

- Introspection Resource indicated with an "rt" value of "oic.wk.introspection".
  内省资源，其“rt”值为“oic.wk.introspection”。
- "/oic/p" indicated with an "rt" value of "oic.wk.p".
  “/oic/p”表示“oic.wk.p”的“rt”值。
- "/oic/d" indicated with an "rt" value of "oic.wk.d"
  “/oic/d”表示“oic.wk.d”的“rt”值
- "/oic/sec/doxm" indicated with an "rt" value of "oic.r.doxm" as defined in ISO/IEC 30118-2:2018.
  "oic.r.doxm"用“rt”值表示"/oic/sec/doxm"，定义见ISO/IEC 30118-2:2018。
- "/oic/sec/pstat" indicated with an "rt" value of "oic.r.pstat" as defined in ISO/IEC 30118-2:2018.
  "oic.r.pstat"用“rt”值表示"/oic/sec/pstat"，定义见 ISO/IEC 30118-2:2018。
- "/oic/sec/acl2" indicated with an "rt" value of "oic.r.acl2" as defined in ISO/IEC 30118-2:2018.
  "oic.r.acl2"用“rt”值表示"/oic/sec/acl2，定义见ISO/IEC 30118-2:2018。
- "/oic/sec/cred" indicated with an "rt" value of "oic.r.cred" as defined in ISO/IEC 30118-2:2018.
  "oic.r.cred"用“rt”值表示"/oic/sec/cred"，定义见ISO/IEC 30118-2:2018。

Conditionally required: 条件要求：

- "/oic/res" with an "rt" value of "oic.wk.res" as self-reference, on the condition that "oic/res" hasto signal that it is Observable by a Client.
  作为参考，"oic.wk.res"用“rt”值为"/oic/res"，前提是“oic/res”必须表明它被客户端观察到。
- if the Device supports batch retrieval of "/oic/res" then "oic.if.b" shall be included in the "if" Property of "/oic/res".
- if the Device supports batch retrieval there shall be a self-reference that includes an "if" Link Parameter containing "oic.if.b"; the self-reference shall expose a secure OCF Endpoint.

The Introspection Resource is only applicable for Devices that host Vertical Resource Types (e.g."oic.r.switch.binary") or vendor-defined Resource Types. Devices that only host Resources Required to onboard the Device as a Client do not have to implement the Introspection Resource.
内省资源仅适用于承载垂直资源类型的设备（例："oic.r.switch.binary"）或者供应商定义的资源类型。只作为客户端承载设备所需的资源的设备不必实现内省资源。

Table 23 provides an OCF registry for protocol schemes.
表23提供了协议方案的OCF注册表。

Table 23 - Protocol scheme registry
表23 - 协议方案注册表

SI NumberSI码	Protocol 协议
1	"coap"
2	"coaps"
3	"http"
4	"https"
5	"coap+tcp"
6	"coaps+tcp"

NOTE The discovery of an OCF Endpoint used by a specific protocol is out of scope. The mechanism used by a Client to form requests in a different messaging protocol other than discovery is out of scope.
注: 发现特定协议使用的OCF端点超出了范围。客户端用于在除发现之外的其他消息传递协议中形成请求的机制超出了范围。

The following applies to the use of "/oic/d":
“/oic/d”的使用情况如下:

- A vertical may choose to extend the list of Properties defined by the Resource Type "oic.wk.d".In that case, the vertical shall assign a new Device Type specific Resource Type ID. The Mandatory Properties defined in Table 24 shall always be present.
  垂直可以选择扩展资源类型“oic.wk.d”定义的属性列表。在这种情况下，垂线应该指定一个新的设备类型特定的资源类型ID。表24中定义的强制属性必须始终存在。
- A Device may choose to expose a separate, Discoverable Resource with its Resource Type ID set to a Device Type. In this case the Resource is equivalent to an instance of "oic.wk.d" and adheres to the definition thereof. As such the Resource shall at a minimum expose the mandatory Properties of "oic.wk.d". In the case where the Resource tagged in this manner isdefined to be an instance of a Collection in accordance with 7.8.3 then the Resources that are part of that Collection shall at a minimum include the Resource Types mandated for the DeviceType.
  设备可以选择公开一个单独的、可发现的资源，其资源类型ID设置为设备类型。在这种情况下，资源相当于“oic.wk.d”的一个实例并遵循其定义。因此，该资源应至少公开“oic.wk.d”的强制属性。如果按照7.8.3将以这种方式标记的资源定义为集合的实例，则作为该集合一部分的资源至少应包括为该设备类型规定的资源类型。

Table 24 "oic.wk.d" Resource Type definition defines the base Resource Type for the "/oic/d"Resource.
表24 "oic.wk.d"资源类型定义“/oic/d”资源的基本资源类型。

Table 24 - "oic.wk.d"Resource Type definition

表24 -"oic.wk.d"资源类型定义

Property title 属性标题	Property Name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
(Device) Name 设备名称	"n"	"string:	N/A	N/A	R	Yes是	Human friendly name defined by the vendor. In the presence of "n" Property of "/oic/con", both have the same Property Value. When "n" Property Value of "/oic/con" is modified, it shall be reflected to "n" Property Value of "/oic/d". 由供应商定义的人性化名称。当存在“/oic/con”的“n”属性时，两者具有相同的属性值。当修改“/oic/con”的“n”属性值时，应反映为“/oic/d”的“n”属性值。
Spec Version	"icv"	"string:	N/A	N/A	R	Yes是	The specification version of this document that a Device is implemented to. The syntax shall be "ocf.<major>.<minor>.<sub-version>" where <major>, <minor, and <sub-version> are the major, minor and sub-version numbers of this document respectively. The specification version number (i.e., <major>.<minor>.<sub-version>) shall be obtained from the title page of this document (e.g. "2.0.5"). An example of the string value for this Property is "ocf.2.0.5". 设备实现到的本文档的规范版本。语法应该是“ocf.<major>.<minor>.”<sub-version>"其中<major>， <minor，<sub-version>分别是本文档的major, minor和sub-version>。版本如<major>.<minor>.<sub-version>）应该从本文件的扉页获得。此属性的字符串值的一个示例是“ocf.2.0.5”。
Device ID 设备ID	"di"	"uuid"	N/A	N/A	R	Yes是	Unique identifier for Device. This value shall be the same value (i.e. mirror) as the doxm.deviceuuid Property as defined in ISO/IEC 30118-2:2018.Handling privacy-sensitivity for the "di" Property, refer to clause 13.16 in ISO/IEC 30118-2:2018. 设备的唯一标识符。此值应与ISO/IEC 30118-2:2018.中定义的doxm.deviceuuid属性相同(即镜像)。对于“di”属性的隐私敏感性处理，请参阅ISO/IEC 30118-2:2018.中的第13.16条。
Data Model Version 数据模型版本	"dmv"	"csv"	N/A	N/A	R	Yes是	Spec version of the Resource specification to which this Device data model is implemented; if implemented against a Vertical specific Device specification(s), then

							the Spec version of the vertical specification this Device model is implemented to. 实现该设备数据模型的资源规范的规范版本；如果反对垂直特定的设备规范实施，则将会使用规格版的垂直规格本设备模型。 The syntax is a comma separated list of <res>.<major>.<minor>.<sub-version> or<vertical>.<major>.<minor>.<sub-version>. <res>is the string "ocf.res" and <vertical> is the name of the vertical defined in the Vertical specific Resource specification. 语法经常用一系列的逗号分开， <res>.<major>.<minor>.<sub-version> or<vertical>.<major>.<minor>.<sub-version>. <res>是字符串“ocf”和<vertical>是在垂直特定资源规范中定义的垂直的名称。 The <major>, <minor>, and <sub-version>are the major, minor and sub-version numbers of the specification respectively. One entry in the csvstring shall be the applicable version of the Resource Type Specification for the Device (e.g "ocf.res.1.0.0"). If applicable, additional entry(-ies) in the csv shall be the vertical(s) being realized (e.g. "ocf.sh.1.0.0"). <major>, <minor>, and <sub-version> 分别为规范的主版本号、次版本号和子版本号。csv字符串中的一个条目应该是设备资源类型规范的适用版本。(e例如，"ocf.res.1.0.0")。如适用，csv中的附加项应为实现的垂直（如，"ocf.sh.1.0.0"）。 This value may be extended by the vendor. The syntax for extending this value, as a comma separated entry, by the vendor shall be by addingx.<Domain_Name>.<vendor_string>.For example "ocf.res.1.0.0, ocf.sh.1.0.0, x.com.example.string", The order of the values in the comma separated string can be in any order (i.e. no prescribed order). This Property shall not exceed 256 octets. 此值可由供应商扩展。扩展这个值的语法，作为逗号分隔的条目，由供应商添加 x.<Domain_Name>.<vendor_string>。例如，"ocf.res.1.0.0, ocf.sh.1.0.0, x.com.example.string"。逗号分隔字符串中值的顺序可以是任意顺序（如，没有规定的顺序）。此属性不应超过256字节。
Permanent Immutable	"piid"	"uuid"	N/A	N/A	R	Yes是	A unique and immutable Device identifier. A Client can detect that a

ID 永久不变的ID							single Device supports multiple communication protocols if it discovers that the Device uses a single Permanent Immutable ID value for all the protocols it supports. Handling privacy- sensitivity for the "piid" Property, refer to clause 13.16 in ISO/IEC 30118-2:2018. 唯一且不可变的设备标识符。客户端可以检测到一个设备支持多个通信协议，如果它发现该设备对它支持的所有协议使用一个永久不变的ID值。处理“piid”属性的隐私敏感性，请参阅ISO/IEC 30118-2:20 08中的第13.16条。
Localized Descriptions 本地化的描述	"ld"	"array"	N/A	N/A	R	No否	Detailed description of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing an IETF RFC 5646 language tag) and a "value" field containing the Device description in the indicated language. 设备的一种或多种语言的详细描述。这个属性是一个对象数组，其中每个对象都有一个“language”字段(包含一个IETF RFC 5646语言标记)和一个“value”字段(包含指定语言中的设备描述)。
Software Version 软件版本	"sv"	"string”	N/A	N/A	R	No否	Version of the Device software. 设备软件版本。
Manufacturer Name 制造商名称	"dmn"	"array"	N/A	N/A	R	No否	Name of manufacturer of the Device, in one or more languages. This Property is an array of objects where each object has a "language" field (containing anIETF RFC 5646 language tag) and a "value" field containing the manufacturer name in the indicated language. 设备制造商的名称，以一种或多种语言表示。这个属性是一个对象数组，其中每个对象都有一个“language”字段(包含一个IETF RFC 5646语言标记)和一个“value”字段(包含指定语言中的制造商名称)。
Model Number 型号	"dmno"	"string”	enum	N/A	R	No否	Model number as designated by manufacturer. 型号由制造商指定。
Ecosystem Name 生态系统名称	"econame"	“string”	N/A	N/A	R	No否	This is the name of ecosystem that a Bridged Device belongs to. If a Device has "oic.d.virtual" as one of Resource Type values ("rt") the Device shall contain this Property, otherwise this Property shall not be included. 这是连接设备所属的生态系统的名称。如果一个设备有“oic.d”。作为资源类型值(“rt”)之一，设备应包含此属

							性，否则不包括此属性。 This Property has enumeration values: ["BLE", "oneM2M", "UPlus", "Zigbee", "Z-Wave"]. 该属性有列举值：["BLE", "oneM2M", "UPlus", "Zigbee", "Z-Wave"].
Version of Ecosystem 生态系统的版本	"ecoversion"	“string”	N/A	N/A	R	No否	This is the version of ecosystem that a Bridged Device belongs to. If a Device has "oic.d.virtual" as one of its Resource Type values ("rt") 这是一个连接设备所属的生态系统版本。如果一个设备有“oic.d”作为它的资源类型值之一(“rt”)。 the Device should contain this Property, otherwise this Property shall not be included. 设备应包含此属性，否则不包含此属性。

Table 25 defines "oic.wk.p" Resource Type.

表25 定义"oic.wk.p"资源类型

Table 25 - "oic.wk.p" Resource Type definition

表25 - "oic.wk.p"资源类型定义

Property title 属性标题	**Property Name 属性名称**	**Value type 值类型**	**Value rule 值规则**	**Unit 单元**	**Access mode 访问模式**	**Mandatory 强制性**	**Description 描述**
Platform ID 平台ID	"pi"	"uuid"	N/A	N/A	R	Yes	Unique identifier for the physical Platform (uUiD); this shall be a UUID in accordance with IETF RFC 4122. It is recommended that the UUID be created using the random generation scheme (version 4 UUID) specific in the RFC. Handling privacy- sensitivity for the "pi" Property, refer to clause 13.16 in ISO/IEC 301182:2018. 物理平台唯一标识符(uUiD)；根据IETF RFC 4122，这应该是一个UUID。建议使用RFC中特定的随机生成方案(版本4 UUID)创建UUID。处理“pi”属性的隐私敏感性，参考ISO/IEC 301182:2018.第13.16条。
ManufacturerName 制造商名称	"mnmn"	"string"	N/A	N/A	R	Yes	Name of manufacturer. 制造商名称。
Manufacturer Details Link 制造商细节链接	"mnml"	"uri"	N/A	N/A	R	No	Reference to manufacturer, represented as a URI. 对制造商的引用，用URI表示。

ModelNumber **型号**	"mnmo"	"string"	N/A	N/A	R	No	Model number as designated by manufacturer. 型号由制造商指定。
Date of Manufacture **出厂日期**	"mndt"	"date"	N/A	N/A	R	No	Manufacturing date of Platform. 平台制造日期。
Serial Number 序列号	"mnsel"	"string"	N/A	N/A	R	No	Serial number of the Platform, may be unique for each Platform of the same model number. 该平台的序列号，对于每个相同型号的平台可能是唯一的。
Platform Version **平台版本**	"mnpv"	"string"	N/A	N/A	R	No	Version of Platform - string (defined by manufacturer). 平台字符串的版本(由制造商定义)。
OS Version **操作系统版本**	"mnos"	"string"	N/A	N/A	R	No	Version of Platform resident OS - string (defined by manufacturer). 平台常驻操作系统的版本-字符串(由制造商定义)。
HardwareVersion **软件版本**	"mnhw"	"string"	N/A	N/A	R	No	Version of Platform hardware. 平台硬件版本。
Firmwareversion 固件版本	"mnfv"	"string"	N/A	N/A	R	No	Version of Platform firmware. 固件平台的的版本。
Support link **支撑链接**	"mnsl"		N/A	N/A	R	No	URI that points to support information from manufacturer. 指向支持来自制造商的信息的URI。
SystemTime **系统时间**	"st"		N/A	N/A	R	No	Reference time for the Platform. 平台参考时间。
Vendor ID **厂商ID**	"vid"		N/A	N/A	R	No	Vendor defined string for the Platform. The string is freeform and upto the vendor on what text to populate it. 供应商为平台定义了字符串。该字符串是自由格式的，由供应商决定填充什么文本。
NetworkConnectivityType 网络连接类型	"mnnct"	"string"	N/A	N/A	R	No	An array of integer where each integer indicates the network connectivity type based on IANAIfType value as defined by IANA ifType-MIB Definitions, e.g., [71, 259] which represents Wi-Fi and Zigbee. 一个整数数组，其中每个整数表示基于IANAIfType值的网络连接类型，定义见 IANA ifType-MIB Definitions, 如，[71, 259]表示 Wi-Fi 和Zigbee。

11.2.4 Resource discovery using "/oic/res" 使用"/oic/res"发现资源

Discovery using "/oic/res" is the default discovery mechanism that shall be supported by all Devices as follows:
使用“/oic/res”进行发现是所有设备都应支持的默认发现机制，如下：

- Every Device updates its local "/oic/res" with the Resources that are discoverable (see 7.3.2.2).Every time a new Resource is instantiated on the Device and if that Resource is discoverable by a remote Device then that Resource is published with the "/oic/res" Resource that is local to the Device (as the instantiated Resource).
  每个设备使用可发现的资源更新其本地“/oic/res”（见7.3.2.2）。每当在设备上实例化一个新资源时，如果该资源可以被远程设备发现，那么该资源将与设备本地的“/oic/res”资源一起发布（作为实例化的资源）。
- A Device wanting to discover Resources or Resource Types on one or more remote Devices Makes a RETRIEVE request to the "/oic/res" on the remote Devices. This request may be sent multicast (default) or unicast if only a specific host is to be probed. The RETRIEVE request may optionally be restricted using appropriate clauses in the query portion of the request. Queries May select based on Resource Types, OCF Interfaces, or Properties.
  希望在一个或多个远程设备上发现资源或资源类型的设备向远程设备上的“/oic/res”发出检索请求。此请求可以发送多播（默认）或单播，如果只是要探测特定的主机。可以在请求的查询部分使用适当的条款限制检索请求。查询可以基于资源类型、OCF接口或属性进行选择。
- The query applies to the representation of the Resources. "/oic/res" is the only Resource whose representation has "rt". So "/oic/res" is the only Resource that can be used for Multicast Discovery at the transport protocol layer.
  查询应用于资源的表示。“/oic/res”是其表示具有“rt”的唯一资源。因此，“/oic/res”是传输协议层中唯一可用于多播发现的资源。
- The Device receiving the RETRIEVE request responds with a list of Resources, the ResourceType of each of the Resources and the OCF Interfaces that each Resource supports.Additionally, information on the policies active on the Resource can also be sent. The policy supported includes Observability and discoverability.
  接收检索请求的设备以资源列表、每个资源的资源类型和每个资源支持的OCF接口进行响应。此外，还可以发送关于资源上活动的策略的信息。所支持的策略包括可观察性和可发现性。
- The receiving Device may do a deeper discovery based on the Resources returned in the request to "/oic/res".
  接收设备可以根据请求“/oic/res”中返回的资源进行更深层次的发现。

The information that is returned on discovery against "/oic/res" is at the minimum:
根据“/oic/res”的发现而返回的信息最少为：

- The URI (relative or fully qualified URL) of the Resource.
  资源的URI（相对的或完全限定的URL）。

- The Resource Type(3) of each Resource. More than one Resource Type may be returned if the Resource enables more than one type. To access Resources of multiple types, the specified Resource Type that is targeted shall be specified in the request.
  每个资源的资源类型(3)。如果资源启用了多个类型，则可能返回多个资源类型。要访问多种类型的资源，需要在请求中指定特定的资源类型。
- The OCF Interfaces supported by that Resource. Multiple OCF Interfaces may be returned. To Access a specific OCF Interface that OCF Interface shall be specified in the request. If the OCF Interface is not specified, then the Default OCF Interface is assumed.
  该资源支持的OCF接口。可以返回多个OCF接口。要访问特定的OCF接口，应在请求中指定该OCF接口。如果没有指定OCF接口，则假定使用默认的OCF接口。

For Clients that do include the OCF-Accept-Content-Format-Version option, an "/oic/res" response includes an array of Links to conform to IETF RFC 6690. Each Link shall use an "eps" Parameterto provide the information for an encrypted connection and carry "anchor" of the value OCF URIwhere the authority component of <deviceID> indicates the Device hosting the target Resource.
对于包含OCF-Accept-Content-Format-Version选项的客户端，“/oic/res”响应包含一个符合IETF RFC 6690的链接数组。每个链接应使用一个“eps”参数来提供加密连接的信息，并携带OCF URI值的“锚”，其中<deviceID>的权威组件指示承载目标资源的设备。

The OpenAPI 2.0 file for discovery using "/oic/res" is described in Annex A. Also refer to clause 10(OCF Endpoint discovery) for details of Multicast discovery using "/oic/res" on a CoAP transport.
使用“/oic/res”进行发现的OpenAPI 2.0文件在附件a中进行了描述。关于在CoAP传输上使用“/oic/res”进行多播发现的详细信息，请参阅第10章（OCF端点发现）。

An example Device might return the following to Clients that request with the Content Format of"application/vnd.ocf+cbor" in Accept Option:
一个示例设备可能会将以下内容返回给请求内容格式为在接受选项上"application/vnd.ocf+cbor"：

```
[
    {
        "href": "/oic/res",
        "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989/oic/res",
        "rel": "self",
        "rt": ["oic.wk.res"],
        "if": ["oic.if.ll", "oic.if.baseline"],
        "p": {"bm": 3},
        "eps": [
                 {"ep": "coap://[fe80::b1d6]:44444"}
               ]
    },
    {
        "href": "/oic/p",
        "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989,
```

```
        "rt": ["oic.wk.p"],
        "if": ["oic.if.r", "oic.if.baseline"],
        "p": {"bm": 3},
        "eps": [
                  {"ep": "coap://[fe80::b1d6]:44444"},
                  {"ep": "coaps://[fe80::b1d6]:11111"}
                ]
    },
    {
        "href": "/oic/d",
        "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989,
        "rt": ["oic.wk.d"],
        "if": ["oic.if.r", "oic.if.baseline"],
        "p": {"bm": 3},
        "eps": [
                   {"ep": "coap://[fe80::b1d6]:44444"},
                   {"ep": "coaps://[fe80::b1d6]:11111"}
                  ]
    },
    {
        "href": "/myLightSwitch",
        "anchor": "ocf://dc70373c-1e8d-4fb3-962e-017eaa863989,
        "rt": ["oic.r.switch.binary"],
        "if": ["oic.if.a", "oic.if.baseline"],
        "p": {"bm": 3},
        "eps": [
                   {"ep": "coap://[fe80::b1d6]:44444"},
                   {"ep": "coaps://[fe80::b1d6]:11111"}
                  ]
    }
]
```

11.2.5 Multicast discovery using "/oic/res" 使用“/oic/res”多播发现

Generic requirements for use of CoAP multicast are provided in clause 12.2.9. Devices shall support use of CoAP multicast to allow retrieving the "/oic/res" Resource from an unsecured OCF Endpoint on the Device. Clients may support use of CoAP multicast to retrieve the "/oic/res" Resource from other Devices. The CoAP multicast retrieval of "/oic/res" supports filtering Links based on the "rt" Property in the Links:
使用CoAP多播的一般要求在第12.2.9章节中规定。设备应支持使用CoAP多播，以允许从设备上不安全的OCF端点检索“/oic/res”资源。客户端可以支持使用CoAP多播从其他设备中检索“/oic/res”资源。CoAP多播检索“/oic/res”支持根据链接中的“rt”属性过滤链接：

- If the discovery request is intended for a specific Resource Type including as part of a multivalue Resource Type, the query parameter "rt" shall be included in the request (see 6.2.2) with its value

set to the desired Resource Type. Only Devices hosting the Resource Type shall respond to the discovery request.
如果发现请求针对特定的资源类型，包括多值资源类型的一部分，则查询参数“rt”应包括在请求中（见6.2.2），其值应设置为所需的资源类型。只有承载资源类型的设备才能响应发现请求。

- When the "rt" query parameter is omitted, all Devices shall respond to the discovery request.
当省略“rt”查询参数时，所有设备都应响应发现请求。

11.3 Notification 通知

11.3.1 Overview 综述

A Server shall support NOTIFY operation to enable a Client to request and be notified of desired states of one or more Resources in an asynchronous manner. 11.3.2 specifies the Observe mechanism in which updates are delivered to the requester.
服务器应支持NOTIFY操作，使客户端能够以异步方式请求并得到一个或多个资源的所需状态的通知。11.3.2指定向请求者交付更新的观察机制。

11.3.2 Observe 观察

11.3.2.1 Overview 综述

In the Observe mechanism the Client utilizes the RETRIEVE operation to require the Server forupdates in case of Resource state changes. The Observe mechanism consists of five steps which are depicted in Figure 11.
在观察机制中，客户端利用RETRIEVE操作要求服务器在资源状态发生更改时进行更新。观察机制由5个步骤组成，如图11所示。

NOTE the Observe mechanism can only be used for a resource with a Property of Observable (see 7.3.2.2).
注：观察机制只能用于具备可观察属性的的资源（见7.3.2.2）。

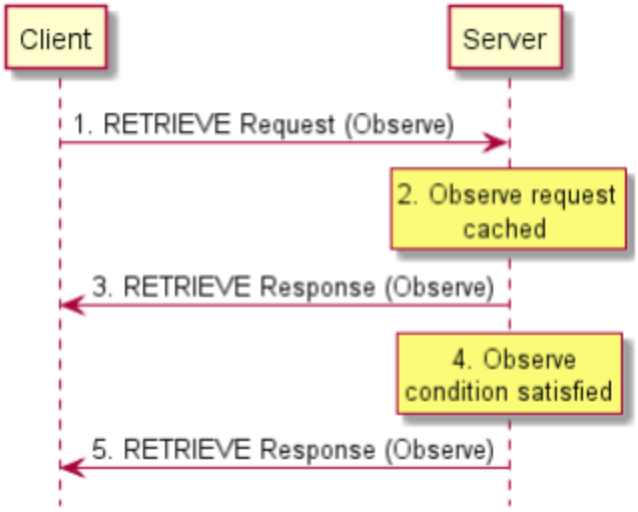

Figure 11 - Observe Mechanism
图11 - 观察机制

11.3.2.2 RETRIEVE request with Observe indication 带观察指示的RETRIEVE请求

The Client transmits a RETRIEVE request message to the Server to request updates for the Resource on the Server if there is a state change. The RETRIEVE request message carries the following parameters:
客户端向服务器发送RETRIEVE请求消息，请求对服务器上资源状态发生更改的资源进行更新。。检索请求消息包含以下参数：

- fr: Unique identifier of the Client.
  fr:客户端的唯一标识符。
- to: Resource that the Client is requesting to Observe.
  to:客户机请求观察的资源。
- ri: Identifier of the RETRIEVE operation.
  ri:RETRIEVE操作的标识符。
- op: RETRIEVE.
  op:RETRIEVE
- obs: Indication for Observe operation.
  obs:观察操作指示。

11.3.2.3 Processing by the Server 服务器处理

Following the receipt of the RETRIEVE request, the Server may validate if the Client has the appropriate rights for the requested operation and the Properties are readable and Observable. If the validation is successful, the Server caches the information related to the Observe request. The Server caches the value of the ri parameter from the RETRIEVE request for use in the initial response and future responses in case of a change of state.

在接收到RETRIEVE请求之后，服务器可能会验证客户端是否具有请求的操作的适当权限，以及属性是否可读和可观察。如果验证成功，服务器将缓存与观察请求相关的信息。服务器从RETRIEVE请求中缓存ri参数的值，以便在状态发生更改时用于初始响应和后续响应。

11.3.2.4 RETRIEVE response with Observe indication 带观察指示的RETRIEVE响应

The Server shall transmit a RETRIEVE response message in response to a RETRIEVE request message from a Client. If validation succeeded, the response includes an Observe indication. If not, the Observe indication is omitted from the response which signals to the requesting Client that registration for notification was not allowed.
服务器应发送RETRIEVE响应消息以响应来自客户端的RETRIEVE请求消息。如果验证成功，则响应包括一个观察指示。如果验证失败，则在响应中省略了观察指示，该指示向请求客户端发出信号，表示不允许注册通知。

The RETRIEVE response message shall include the following parameters:
RETRIEVE响应消息应包括以下参数:

- fr: Unique identifier of the Server.
  fr:服务器的唯一标识符。
- to: Unique identifier of the Client.
  to:客户端的唯一标识符。
- ri: Identifier included in the RETRIEVE operation.
  ri:RETRIEVE操作中包含的标识符。
- cn: Information Resource representation as requested by the Client.
  cn:客户端请求的信息资源表示。
- rs: The result of the RETRIEVE operation.
  rs: RETRIEVE操作的结果。
- obs: Indication that the response is made to an Observe operation.
  obs: 表示响应是针对一个观察操作的

11.3.2.5 Resource monitoring by the Server 服务器对资源的监视

The Server shall monitor the state the Resource identified in the Observe request from the Client.Anytime there is a change in the state of the Observed Resource, the Server sends another RETRIEVE response with the Observe indication. The mechanism does not allow the client to specify any bounds or limits which trigger a notification, the decision is left entirely to the server.
服务器应监视来自客户端的观察请求中标识的资源的状态。每当观察到的资源的状态发生变化时，服务器就发送另一个带有观察指示的检索响应。该机制不允许客户端指定触发通知的任何界限或限制，完全由服务器来决定。

11.3.2.6 Additional RETRIEVE responses with Observe indication 附加检索响应与观察指示

The Server shall transmit updated RETRIEVE response messages following Observed changes in the state of the Resources indicated by the Client. The RETRIEVE response message shall include the parameters listed in 11.3.2.4.
在客户端指示的资源状态发生观察到的更改之后，服务器应发送更新的检索响应消息。RETRIEVE响应消息应包含11.3.2.4中列出的参数。

11.3.2.7 Cancelling Observe 取消观察

The Client can explicitly cancel Observe by sending a RETRIEVE request without the Observeindication field to the same Resource on the Server which it was Observing. For certain protocol mappings, the Client may also be able to cancel an Observe by ceasing to respond to the RETRIEVE responses.
客户端可以明确地取消资源观察，通过向服务器正在观察的该资源发送一个不带观察指示字段的RETRIEVE请求。对于某些协议映射，客户端也可以通过停止响应RETRIEVE响应来取消观察。

11.4 Introspection 自省

11.4.1 Overview 综述

Introspection is a mechanism to announce the capabilities of Resources hosted on the Device.
自省是一种公开设备上托管的资源的功能的机制。

The intended usage of the Introspection Device Data (IDD) is to enable dynamic Clients e.g. Clients that can use the IDD) to generate dynamically a UI or dynamically create translations of the hosted Resources to another eco-system. Other usages of Introspection is that the information can be used to generate Client code. The IDD is designed to augment the existing data already on the wire. This means that existing mechanisms need to be used to get a full overview of what is implemented in the Device. For example, the IDD does not convey information about Observability,since that is already conveyed with the "p" Property on the Links in "/oic/res" (see 7.8.2.5.3).
内省设备数据（IDD）的用途是使动态客户端（例如可以使用IDD的客户端）能够动态地生成UI或动态地创建托管资源到另一个生态系统的转换。自省的其他用途是，可以使用这些信息来生成客户端代码。IDD旨在增强现有的线上数据。这意味着需要使用现有的机制来全面了解设备中实现的内容。例如，IDD没有传达关于观察性的信息，因为观察性已经用“/oic/res”链接上的“p”属性传达了（见7.8.2.5.3）。

The IDD is recommended to be conveyed as static data. Meaning that the data does not change during the uptime of a Device. However, when the IDD is not static, the Introspection Resource shall be Observable and the url Property Value of "oic.wk.introspection" Resource shall change to indicate that the IDD is changed.
建议将IDD作为静态数据进行传输。这意味着数据在设备正常运行期间不会改变。但是，当IDD不是静态的

时候，自省资源应该是可观察的，并且“oic.wk.introspection”资源的url属性值应该改变，表示IDD已经改变。

The IDD describes the Resources that make up the Device. For the complete list of included Resources see Table 20. The IDD is described as a OpenAPI 2.0 in JSON format file. The text in the following bulleted list contains OpenAPI 2.0 terms, such as paths, methods etc. The OpenAPI 2.0 file shall contain the description of the Resources:
IDD描述组成设备的资源。有关所包含资源的完整列表，请参见表20。IDD是一个JSON格式的OpenAPI 2.0文件。以下项目符号列表中的文本包含OpenAPI 2.0术语，如路径、方法等。OpenAPI 2.0文件应包含资源描述：

- The IDD will use the HTTP syntax, e.g., define the CRUDN operation as HTTP methods and use the HTTP status codes.
  IDD将使用HTTP语法，例如，将CRUDN操作定义为HTTP方法，并使用HTTP状态码。
- The IDD does not have to define all the status codes that indicate an error situation.
  IDD不需要定义所有指示错误情况的状态码。
- The IDD does not have to define a schema when the status code indicates that there is no payload (see HTTP status code 204 as an example).
  当状态码指示没有有效载荷时，IDD不必定义模式（参见HTTP状态码204作为示例）。
- The paths (URLs) of the Resources in the IDD shall be without the OCF Endpoint description,e.g. it shall not be a fully-qualified URL but only the relative path from the OCF Endpoint, akathe "href". The relative path may include a query parameter (e.g. "?if=oic.if.ll"), in such cases the text following (and including) the delimiter shall be removed before equating to the "href"that is conveyed by "/oic/res".
  IDD中资源的路径（URL）应该没有OCF端点描述，例如，它不应该是一个完全限定的URL，而应该是来自OCF端点的相对路径，即“href”。相对路径可以包含一个查询参数（例如“?if=oic.if.ll”），在这种情况下，分隔符后面（包括）的文本应在等于由“/oic/res”传递的“href”之前移除。
- The following Resources shall be excluded in the IDD:
  IDD应不包括下列资源：
  - Resource with Resource Type: "oic.wk.res" unless 3rd party defined or optional Propertiesare implemented.
    资源类型:“oic.wk.res除非实现了第三方定义的或可选的属性。
  - Resource with Resource Type:"oic.wk.introspection".
    资源与资源类型:"oic.wk.introspection".。
  - Resources explicitly identified within other specifications working in conjuction with this document (e.g. Resources that handle Wi-Fi Easy Setup, see [2]).
    在与本文件一起使用的其他规范中明确标识的资源（例如，处理Wi-Fi简易设置的资源，参见[2]）
- The following Resources shall be included in the IDD when optional or 3rd party defined Properties are implemented:
  在实现可选或第三方定义的属性时，应在IDD中包含以下资源:

    - Resources with type:"oic.wk.p" and "oic.wk.d"(e.g. discovery related Resources).
      资源类型："oic.wk.p"和"oic.wk.d"（如，发现相关资源）
    - Security Virtual Resources from ISO/IEC 30118-2:2018.
      ISO/IEC 30118-2:2018安全虚拟资源。
- When the Device does not expose instances of Vertical Resource Types, and does not have any 3rd party defined Resources (see 7.8.4.4), and does not need to include Resources in the IDD due to other clauses in this clause, then the IDD shall be an empty OpenAPI 2.0 file. An Example of an empty OpenAPI 2.0 file can be found in found in Annex B.2.
  当设备没有公开垂直资源类型的实例，也没有任何第三方定义的资源（见7.8.4.4），且由于本款其他条款不需要在IDD中包含资源时，则IDD为空OpenAPI 2.0文件。一个空的OpenAPI 2.0文件的例子可以在附件B.2中找到。
- All other Resources that are individually addressable by a Client (i.e. the "href" can be resolved and at least one operation is supported with a success path response) shall be listed in the IDD.
  所有其他可由客户端单独寻址的资源(即“href”可以被解析，并且至少有一个操作支持成功路径响应)应在IDD中列出。
- Per Resource the IDD shall include:
  每个资源的IDD应包括：
    - All implemented methods
      所有实施的方法
        - For an OCF defined Resource Type, only the methods that are listed in the OpenAPI 2.0definition are allowed to exist in the IDD. For an OCF defined Resource Type, methods not listed in the OpenAPI 2.0 definition shall not exist in the IDD. The supported methods contained in the IDD shall comply with the listed OCF Interfaces. For example, if the POST method is listed in the IDD, then an OCF Interface that allows UPDATE will be listed in the IDD.
          对于OCF定义的资源类型，只有OpenAPI 2.0定义中列出的方法才允许在IDD中存在。对于定义的OCF资源类型，OpenAPI 2.0定义中未列出的方法在IDD中不存在。IDD中支持的方法应符合所列OCF接口。例如，如果POST方法在IDD中列出，那么允许更新的OCF接口将在IDD中列出。
    - Per supported method:
      按支持方法:
        - Implemented query parameters per method.
          实现每个方法的查询参数。
            - This includes the supported OCF Interfaces ("if") as enum values.
              这包括作为enum值支持的OCF接口（“if”）。
        - Schemas of the payload for the request and response bodies of the method.
          方法的请求和响应主体的有效载荷模式。
        - Where the schema provides the representation of a batch request or response ("oic.if.b")the schema shall contain the representations for all Resource Types that may be included within the batch representation. The representations shall be provided within the IDD itself.

当模式提供批处理请求或响应的表示形式（"oic.if.b"）时，模式应包含批处理表示形式中可能包含的所有资源类型的表示形式。申述必须在IDD内部提供。

- The schema data shall be conveyed by the OpenAPI 2.0 schema.
  模式数据应由OpenAPI 2.0模式传递。
- The OpenAPI 2.0 schema object shall comply with:
  OpenAPI 2.0架构对象应符合:
    - The schemas shall be fully resolved, e.g. no references shall exist outside the OpenAPI 2.0 file.
      架构应该被完全解析，例如OpenAPI 2.0文件之外不应该存在任何引用。
    - The schemas shall list which OCF Interfaces are supported on the method.
      架构应列出该方法支持哪些OCF接口。
    - The schemas shall list if a Property is optional or required.
      如果属性是可选的或必需的，模式应该列出。
    - The schemas shall include all Property validation keywords. Where an enum isdefined the enum shall contain the values supported by the Device. When vendor defined extensions exist to the enum (defined in accordance to 7.8.4.4) these shall be included in the enum.
      架构应包含所有属性验证关键字。在定义枚举的地方，枚举应该包含设备所支持的值。当供应商定义的扩展存在于enum（根据7.8.4.4定义）时，这些扩展应包括在enum中。
    - The schemas shall indicate if an Property is read only or read-write.
      模式应指示属性是只读还是读写。
        - By means of the readOnly schema tag belonging to the Property.
          通过属于属性的只读模式标记。
        - Default value of readOnly is false as defined by OpenAPI 2.0.OpenAPI 2.0定义的readOnly的默认值为false。
    - The default value of the "rt" Property shall be used to indicate the supported Resource Types.
      “rt”属性应使用的默认值，表示支持的资源类型。
    - oneOf and anyOf constructs are allowed to be used as part of a OpenAPI 2.0 schema object. The OpenAPI 2.0 schema with oneOf and anyOf constructs can be found in Annex B.1.
      任何一个结构都可以作为OpenAPI 2.0模式对象的一部分。OpenAPI 2.0模式有一个和任意结构，可以在附件B.1中找到。

- For Atomic Measurements (see clause 7.8.4), the following apply:
  对于原子测量（见第7.8.4条），适用如下:
    - The "rts" Property Value in the IDD shall include only the Resource Types the instance contains and not the theoretical maximal set allowed by the schema definition.
      IDD中的“rts”属性值应该只包含实例所包含的资源类型，而不包括模式定义所允许的理论最大集。

- The Resources that are part of an Atomic Measurement, excluding the Atomic Measurement Resource itself, shall not be added to their own individual path in the IDD, as they are not individually addressable; however, the schemas for the composed Resource Types shall be provided in the IDD as part of the batch response definition along with the "href" for theResource.
  属于原子测量的资源（不包括原子测量资源本身），不得加入国际直拨电话的个别路径，因为它们并非个别可寻址；但是，组合资源类型的模式应该在IDD中与资源的“href”一起作为批处理响应定义的一部分提供。

Dynamic Resources (e.g. Resources that can be created on a request by a Client) shall have a URL definition which contains a URL identifier (e.g. using the {} syntax). A URL with {} identifies that the Resource definition applies to the whole group of Resources that may be created. The Actual path may contain the Collection node that links to the Resource.
动态资源（例：可以由客户端请求创建的资源）应该有一个包含URL标识符的URL定义（例：使用{}语法）。带有{}的URL标识资源定义适用于可能创建的整个资源组。实际路径可能包含链接到资源的集合节点。

Example of a URL with identifiers:
带有标识符的URL示例:

```
/SceneListResURI/{SceneCollectionResURI}/{SceneMemberResURI}:
```

When different Resource Types are allowed to be created in a Collection, then the different schemas for the CREATE method shall define all possible Resource Types that may be created.The schema construct oneOf allows the definition of a schema with selectable Resources. The oneOf construct allows the integration of all schemas and that only one existing subschema shall be used to indicate the definition of the Resource that may be created.
当允许在集合中创建不同的资源类型时，那么CREATE方法的不同模式应该定义所有可能创建的资源类型。模式结构之一允许定义具有可选资源的模式。oneOf结构允许集成所有模式，并且只允许一个现有的子模式应使用来表示可能创建的资源的定义。

Example usage of oneOf JSON schema construct is shown in Figure 12:
一个JSON模式结构的示例用法如图12所示:

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

Figure 12 – Example usage of oneOf JSON schema
图12-一个JSON架构案例使用

A Client using the IDD of a Device should check the version of the supported IDD of the Device.The OpenAPI 2.0 version is indicated in each file with the tag "swagger". Example of the 2.0supported version of the tag is: "swagger": "2.0". Later versions of this document may reference newer versions of the OpenAPI specification, for example 3.0.
使用设备的IDD的客户端应该检查设备支持的IDD的版本。OpenAPI 2.0版本在每个文件中都用标记“swagger”表示。标签的2.0支持版本的例子是:“swagger”:“2.0”。本文档的后续版本可能会引用OpenAPI规范的新版本，例如3.0。

A Device shall support one Resource with a Resource Type of "oic.wk.introspection" as defined in Table 26. The Resource with a Resource Type of "oic.wk.introspection" shall be included in the Resource "/oic/res".An empty IDD file, e.g. no URLs are exposed, shall still have the mandatory OpenAPI 2.0 fields.See OpenAPI specification. An example of an empty OpenAPI 2.0 file can be found in found in Annex B.2.
一个设备应该支持一个资源，其资源类型为表26中定义的“oic.wk.introspection”。资源类型为“oic.wk.introspection”的资源应包含在资源“/oic/res”中。一个空的IDD文件，例如没有url暴露，仍然应该有强制的OpenAPI 2.0字段。看到OpenAPI规范。一个空的OpenAPI 2.0文件的例子可以在附件B.2中找到。

An empty IDD file, e.g. no URLs are exposed, shall still have the mandatory OpenAPI 2.0 fields. See OpenAPI specification. An example of an empty OpenAPI 2.0 file can be found in found in Annex B.2.

Table 26 - Introspection Resource
表26-自省资源

Pre-defined URI 预定义URI	Resource Type Title 资源类型的标题	Resource Type ID ("rt" value) 资源类型 ID ("rt" value)	OCF Interfaces OCF接口	Description 描述	Related Functional Interaction 相关功能联系

None 无	Introspection 自省	"oic.wk.introspection"	"oic.if.r"	The Resource that announces the URL of the Introspection file. 声明自省文件URL的资源。	Introspection 自省

Table 27 defines "oic.wk.introspection" Resource Type.
表27 定义"oic.wk.introspection"资源类型

Table 27 - "oic.wk.introspection"Resource Type definition
表27 -"oic.wk.introspection"资源类型定义

Property title 属性标题	Property name 属性名称	Value type 值类型	Value rule 值规则	Unit 单元	Access mode 访问模式	Mandatory 强制性	Description 描述
urlInfo	"urlInfo"	"array"	N/A	N/A	R	Yes	array of objects 对象组合
url	"url"	"string"	"uri"	N/A	R	Yes	URL to the hosted payload 到承载的有效负载的URL
Protocol 协议	"protocol"	"string"	"enum"	N/A	R	Yes	Protocol definition to retrieve the Introspection Device Data from the url. 从url检索自检设备数据的协议定义。
content-type 内容-类型	"content-type"	"string"	"enum"	N/A	R	Yes	content type of the url. url的内容类型。
Version 版本	"version"	"integer"	"enum"	N/A	R	Yes	Version of the Introspection protocol, indicates which rules are applied on the Introspection Device Data regarding the content of the OpenAPI 2.0 file. Current value is 1. 内省协议的版本，指示哪些规则应用于有关OpenAPI 2.0文件内容的内省设备数据。 当前值为1。

11.4.2 Usage of Introspection 自省的使用

The Introspection Device Data is retrieved in the following steps and as depicted in Figure 13:
自检设备数据通过以下步骤获取，如图13所示：

- Check if the Introspection Resource is supported and retrieve the URL of the Resource.
  检查是否支持自省资源并检索资源的URL。

- Retrieve the contents of the Introspection Resource
  检索自省资源的内容
- Download the Introspection Device Data from the URL specified the Introspection Resource.
  从指定自省资源的URL下载自省设备数据。
- Usage of the Introspection Device Data by the Client
  客户端自检设备数据的使用。

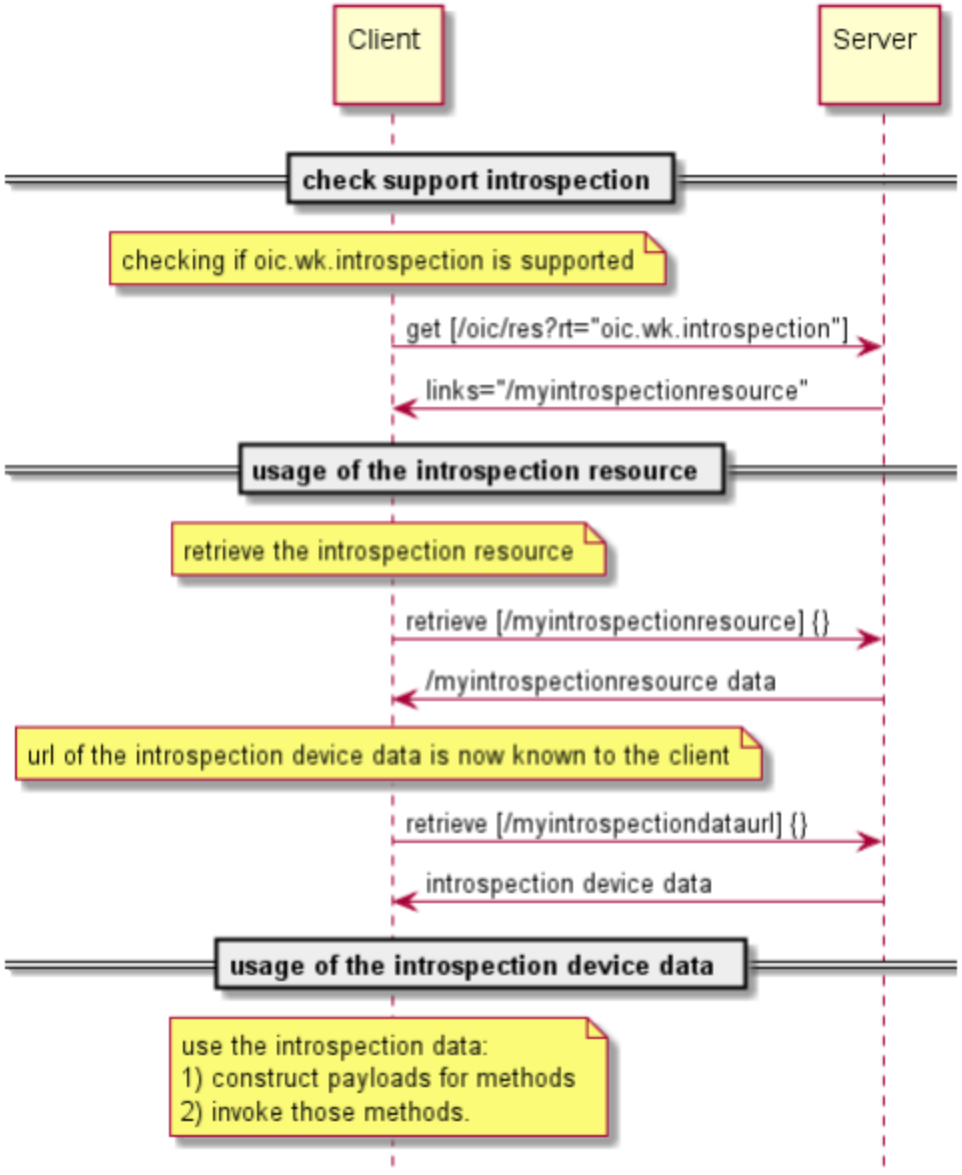

Figure 13 - Interactions to check Introspection support and download the IntrospectionDevice Data.

图13 - 检查自省支持和下载自省设备数据的交互。

11.5 Semantic Tags 语义标签

11.5.1 Introduction 引言

Semantic Tags are meta-information associated with a specific Resource instance that are represented as both Link Parameters and Resource Properties that provide a mechanism where by the Resource be annotated with additional contextual metadata that helps describe the Resource.
语义标签是与特定资源实例相关联的元信息，它以链接参数和资源属性的形式表示，提供了一种机制，通过这种机制，可以使用附加的上下文元数据对资源进行注释，从而帮助描述资源。

When a Semantic Tag is defined for a Resource, it shall be present as a Link Parameter in all Links That are present that target the Resource, including Links in "/oic/res" if the Resource is a Discoverable Resource. The Semantic Tag is further treated as a Common Property associated with the Resource and so shall be returned as part of the "baseline" response for the Resource ifa Semantic Tag has been populated.
当为资源定义一个语义标签时，它应该以链接参数的形式出现在所有针对该资源的链接中，包括“/oic/res”中的链接（如果该资源是可发现的资源）。语义标签进一步被视为与资源相关的公共属性，因此，如果填充了语义标签，则应作为资源的“基线”响应的一部分返回。

11.5.2 Semantic Tag definitions 语义标签定义

11.5.2.1 Relative and descriptive position Semantic Tags 相对和描述位置语义标签

11.5.2.1.1 Introduction 引言

Consider where there may be multiple instances of the same Resource Type exposed by a Device;or a case where there may be potentially ambiguity with regard to the physical attribute that a Resource is representing. In such a case the ability to annotate the Links to the Resource with information pertaining to the relative position of the Resource within the Physical Device becomes useful.
考虑设备可能公开同一资源类型的多个实例；或者在资源所表示的物理属性方面可能存在潜在的模糊性。在这种情况下，使用与物理设备中资源的相对位置相关的信息来注释到资源的链接的能力变得非常有用。

11.5.2.1.2 "tag-pos-desc"or position description Semantic Tag "tag-pos-desc"或者位置描述语义标签

The "tag-pos-desc" Semantic Tag as defined in Table 28 describes the position of the Resource as a descriptive position. If the tag is not exposed it conveys the same meaning as if the tag is exposed with a value of "unknown". The value for the "tag-pos-desc" Semantic Tag if exposed, shall be a string containing a value from the enumeration detailed in Annex C. The population of the Semantic Tag is defined by the Device vendor and shall not be mutable by a Client.
表28中定义的“Tag -pos-desc”语义标记将资源的位置描述为描述性位置。如果标签没有被公开，它所传达

的含义与标签被公共的值“unknown”相同。“Tag -pos-desc”语义标签的值如果公开，应该是一个包含附录c中详细说明的枚举值的字符串。语义标签的填充由设备供应商定义，客户端不能改变。

Table 28 - "tag-pos-desc" Semantic Tag definition
表28 - "tag-pos-desc"语义标记定义

Link Parameter name 链接参数名称	Type 类型	Contents 内容	Value example 数值实例
"tag-pos-desc"	enum	See Annex C	"tag-pos-desc": "topleft"

11.5.2.1.3 "tag-pos-rel" or relative position Semantic Tag "tag-pos-rel"或相对位置语义标签

The "tag-pos-rel" Semantic Tag describes the position of the Resource as a relative position in 3Dspace against a known point defined by the Device vendor. The known point is defined using [x,y,z]form as [0.0,0.0,0.0]. The position itself is then represented by the x-, y-, and z- plane relativeposition from this known point using a bounded box of size +1.0/-1.0 in each plane.
“Tag -pos-rel”语义标签将资源的位置描述为3D空间中相对于设备供应商定义的已知点的相对位置。使用[x,y,z]形式将已知点定义为[0.0,0.0,0.0]。然后在每个平面中使用大小为+1.0/-1.0的有界方框，用x-、y-和z-面相对于已知点的相对位置来表示位置本身。

Figure 14 illustrates the definition of "tag-pos-rel".
表14 举例说明"tag-pos-rel"的定义。

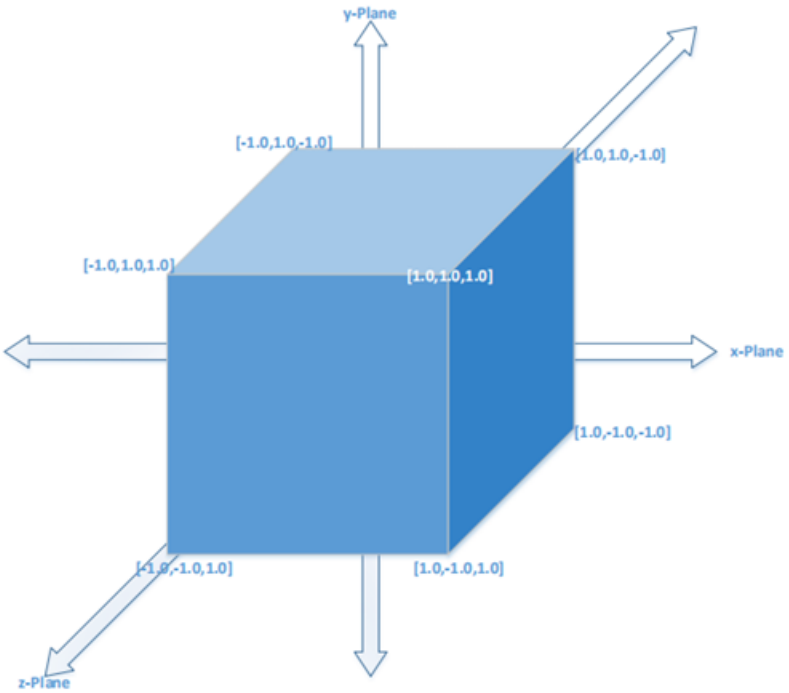

Figure 14 - "tag-pos-rel"definition
图14-tag-pos-rel"定义

The "tag-pos-rel" Semantic Tag value is defined by the Device vendor and shall not be mutable by a Client. This is detailed in Table 29.
“Tag -pos-rel”语义标签值由设备供应商定义，客户端不能更改。这在表29中有详细说明。

Table 29 - "tag-pos-rel" Semantic Tag definition
表29-"tag-pos-rel"语义标签定义

Link Parameter name 链接参数名称	Type 类型	Contents 内容	Value example 数值示例
"tag-pos-rel"	array	Three element array of numbers defining the position relative to a known [0,0,0] point within the context of an abstract box [-1,-1,-1],[1,1,1]. 三个元素的数字数组，定义了抽象框[-1，-1，-1]，[1,1,1]上下文中相对于已知[0,0,0]点的位置。	"tag-pos-rel": [0.5,0.5,0.5]

11.5.2.2 Functional behavior Semantic Tags 功能表现语义标签

11.5.2.2.1 Introduction 引言

Consider, for example, the case of a Device that supports two target temperatures simultaneously for different modes of operation, for example a temperature for heating and a separate temperature for cooling.
例如，考虑一个设备的情况，它同时支持两种不同操作模式的目标温度，例如加热温度和单独的冷却温度。

There is then an ambiguity with respect to the target mode of the specific temperature Resource;it isn't explicit which instance of temperature is associated with which Device function. In such a case the ability to annotate the Links to the Resource with information pertaining to the function of the Resource within the Physical Device becomes useful.
然后，就特定温度资源的目标模式而言，就会产生歧义;没有明确说明哪个温度实例与哪个设备功能相关。在这种情况下，使用与物理设备中资源的功能相关的信息来注释到资源的链接的能力变得非常有用。

11.5.2.2.2 "tag-func-desc" or function description Semantic Tag "tag-func-desc"或功能描述语义标签

The "tag-func-desc" Semantic Tag describes the function of the Resource, if exposed it shall be populated with a value from the currently supported set of standardized enumeration values defined by the Device ecosystem specifications. If the tag is not exposed it conveys the same meaning as if the tag is exposed with a value of "unknown". The value for the "tag-func-desc" Semantic Tag, if exposed, is defined by the

Device vendor and shall not be mutable by a Client.
“Tag - function -desc”语义标签描述了资源的功能，如果公开了该资源，则应该使用设备生态系统规范定义的当前支持的标准化枚举值集中的值填充该资源。如果标签没有被暴露，它所传达的含义与标签被暴露的值“unknown”相同。“Tag - function -desc”语义标签的值，如果暴露，由设备供应商定义，客户端不能改变。

This "tag-func-desc" Semantic Tag is detailed in Table 30.
这个“Tag - function -desc”语义标签详见表30。

Table 30 - "tag-func-desc"Semantic Tag definition
表30-"tag-func-desc"语义标签定义

Link Parameter name 链接参数名称	Type 类型	Contents 内容	Value example 数值示例
"tag-func-rel"	enum	Defined by Device ecosystem 由设备生态系统定义	"tag-func-desc": "cool"

12 Messaging 信息收发

12.1 Introduction 引言

This clause specifies the protocol messaging mapping to the CRUDN messaging operations (clause 8) for each messaging protocol specified (e.g., CoAP.). Mapping to additional protocols is expected in later version of this document. All the Property information from the Resource model shall be carried within the message payload. This payload shall be generated in the Resource model layer and shall be encapsulated in the data connectivity layer. The message header shall only be used to describe the message payload (e.g., verb, mime-type, message payload format), in addition to the mandatory header fields defined in a messaging protocol (e.g., CoAP) specification. If the message header does not support this, then this information shall also be carried in the message payload. Resource model information shall not be included in the message header structure unless the message header field is mandatory in the messaging protocol specification.
本章节为每个指定的消息传递协议（例如，CoAP）中指定的协议消息映射到CRUDN消息传递操作（第8章）。希望本文件的后续更新版本能映射别的协议。来自资源模型的所有属性信息都应该包含在消息有效载荷中。此有效载荷应在资源模型层生成，并封装在数据连接层中。除了消息协议（如CoAP）规范中定义的强制头字段外，消息头部只能用于描述消息有效负载（例如，verb、mime类型、消息有效载荷格式）。如果消息头不支持此操作，则此信息也应包含在消息有效载荷中。除非消息头字段在消息协议规范中是必需的，否则资源模型信息不应包含在消息头结构中。

When a Resource is specified with a RESTful description language like OpenAPI 2.0 then the HTTPsyntax definitions are used in the description (e.g., HTTP syntax for the CRUDN operations, status codes, etc). The HTTP syntax will be mapped to the actual used web transfer protocol (e.g., CoAP).

当使用诸如OpenAPI 2.0之类的RESTful描述语言指定资源时，将在描述中使用HTTP语法定义（例如，CRUDN操作的HTTP语法、状态代码等）。HTTP语法将映射到实际使用的web传输协议（例如，CoAP）。

12.2 Mapping of CRUDN to CoAP CRUDN到CoAP的映射

12.2.1 Overview 综述

A Device implementing CoAP shall conform to IETF RFC 7252 for the methods specified in clause12.2.3.
A Device implementing CoAP shall conform to IETF RFC 7641 to implement the CoAP Observe option.
Support for CoAP block transfer when the payload is larger than the MTU is defined in 12.2.8.
实现CoAP的设备应符合IETF RFC 7252中第12.2.3章节中规定的方法。实现CoAP的设备应符合IETF RFC 7641来实现CoAP观察选项。当有效载荷大于MTU时，在12.2.8中定义了对CoAP块传输的支持。

12.2.2 URIs URIs

An OCF: URI is mapped to a coap: URI by replacing the scheme name "ocf" with "coap" if unsecure or "coaps" if secure before sending over the network by the requestor. Similarly on the receiverside, the scheme name is replaced with "ocf".
OCF: URI映射到coap: URI，如果不安全，则使用“coap”替换方案名称“OCF”，如果安全，则在请求者通过网络发送之前使用“coaps”。类似地，在接收端，方案名称被替换为“ocf”。

Any query string that is present within the URI is encoded as one or more URI-Query Options as defined in IETF RFC 7252 clause 6.4.
URI中出现的任何查询字符串都被编码为IETF RFC 7252条款6.4中定义的一个或多个URI查询选项。

12.2.3 CoAP method with request and response 带有请求和响应的CoAP方法

12.2.3.1 Overview 综述

Every request has a CoAP method that realizes the request. The primary methods and their meanings are shown in Table 31, which provides the mapping of GET/POST/DELETE methods to CREATE, RETRIEVE, UPDATE, and DELETE operations. The associated text provides the generic behaviors when using these methods, however Resource OCF Interfaces may modify these generic semantics. The HTTP codes in the RESTful descriptions will be translated as described in IETF RFC 8075 clause 7 Response Code Mapping. CoAP methods not listed in Table 31 are not supported.
每个请求都有一个CoAP方法来实现请求。主要方法及其含义如表31所示，其中提供了用于创建、检索、更新和删除操作的GET/POST/DELETE方法的映射。关联的文本在使用这些方法时提供了通用行为，但是资源OCF接口可能会修改这些通用语义。RESTful描述中的HTTP代码将按照IETF RFC 8075第7章中的响应代码映射进行翻译。不支持表31中未列出的CoAP方法。

Table 31 - CoAP request and response

表31-CoAP请求和响应

Method for CRUDN **CRUDN 方法**	**(mandatory) Request data** **（强制）请求数据**	**(mandatory) Response data** **（强制）响应数据**
GET for RETRIEVE 得到检索	● Method code: GET (0.01). 方法编码：GET (0.01). ● Request URI: an existing URI for the Resource to be retrieved 请求URI：要检索的资源的现有URI	● Response code: success (2.xx) or error (4.xx or 5.xx). 请求编码：成功(2.xx)或错误(4.xx或5. xx) ● Payload: Resource representation of the target Resource (when successful). 有效载荷：当成功时资源表现为目标资源。
POST for CREATE 创建发布	● Method code: POST (0.02). 方法编码：POST (0.02) ● Request URI: an existing URI for the Resource responsible for the creation. 请求URI：负责创建资源的现有URI。 ● Payload: Resource presentation of the Resource to be created. 有效载荷：要创建的资源的资源表示。	● Response code: success (2.xx) or error (4.xx or 5.xx). 请求编码：成功(2.xx)或错误(4.xx或5. xx) ● Payload: the URI of the newly created Resource (when successful). 有效载荷：等成功时会显示新创建资源的URI。
POST for UPDATE 发布更新	● Method code: POST (0.02). 方法编码：POST (0.02) ● Request URI: an existing URI for the Resource to be updated. 请求URI：负责创建资源的现有URI。 ● Payload: representation of the Resource to be updated. 有效载荷：表示要更新的资源。	● Response Code: success (2.xx) or error (4.xx or 5.xx). 响应编码：成功(2.xx)或错误(4)xx或5. xx)。
DELETE for DELETE 删除	● Method code: DELETE (0.04). 方法编码：DELETE (0.04). ● Request URI: an existing URI for the Resource to be deleted. 请求URI：要删除的资源的现有URI。	● Response code: success (2.xx) or error (4.xx or 5.xx). 响应编码：成功(2.xx)或错误(4.xx或5. xx)。

12.2.3.2 CREATE with POST 创建与发布

POST shall be used only in situations where the request URI is valid, that is it is the URI of an existing Resource on the Server that is processing the request. If no such Resource is present, the Server shall respond with an error response code of 4.xx. The use of POST for CREATE shall use an existing request URI which identifies the Resource on the Server responsible for creation. TheURI of the created Resource is determined by the Server and provided to the Client in the response.

应仅在请求URI有效的情况下使用POST，即处理请求的是服务器上现有资源的URI。如果没有这样的资源

，服务器将以错误响应代码4.xx进行响应。使用POST创建应使用标识服务器上负责创建的资源的现有请求URI。创建的资源的URI由服务器决定，并在响应中提供给客户端。

A Client shall include the representation of the new Resource in the request payload. The new resource representation in the payload shall have all the necessary Properties to create a valid Resource instance, i.e. the created Resource should be able to properly respond to the valid Request with mandatory OCF Interface (e.g., "GET with ?if=oic.if.baseline").
客户端应在请求有效载荷中包含新资源的表示。有效载荷中的新资源表示应该具有创建有效资源实例所需的所有属性，即创建的资源应该能够通过强制OCF接口正确响应有效请求。（例："GET with ?if=oic.if.baseline"）

Upon receiving the POST request, the Server shall either:
在收到POST请求后，服务器应:

- Create the new Resource with a new URI, respond with the new URI for the newly created Resource and a success response code (2.xx); or
  使用新URI创建新资源，并使用新创建资源的新URI和成功响应代码（2.xx）进行响应；或
- respond with an error response code (4.xx or 5.xx).
  响应一个错误响应代码（4.xx或5. xx）。

12.2.3.3 RETRIEVE with GET 检索与获得

GET shall be used for the RETRIEVE operation. The GET method retrieves the representation of the target Resource identified by the request URI.
RETRIEVE操作应使用GET方法。GET方法检索由请求URI标识的目标资源的表示形式。

Upon receiving the GET request, the Server shall either:
服务器收到GET请求后，应:

- Send back the response with the representation of the target Resource with a success response code (2.xx); or
  返回携带成功响应代码（2.xx）的目标资源表示的响应;或respond with an error response code (4.xx or 5.xx) or ignore it (e.g. non-applicable multicastGET).
  响应一个错误响应代码（4.xx或5.xx）或忽略它（如不适用的多播GET请求）。

GET is a safe method and is idempotent.
GET是一种安全的等冥方法。

12.2.3.4 UPDATE with POST 更新与发布

POST shall be used only in situations where the request URI is valid, that is it is the URI of anexisting Resource on the Server that is processing the request. If no such Resource is present, theServer shall respond with an error response code of 4.xx. A client shall use POST to UPDATEProperty values of an existing Resource.

仅在请求URI有效的情况下发使用POST，即处理请求的是服务器上现有资源的URI。如果没有这样的资源，服务器将以错误响应代码4.xx进行响应。客户端应使用POST方式更新现有资源的属性值。

Upon receiving the request, the Server shall either:
在收到请求后，服务器应:

- Apply the request to the Resource identified by the request URI in accordance with the appliedOCF Interface (i.e. POST for non-existent Properties is ignored) and send back a response with a success response code (2.xx); or
  根据应用的OCF接口将请求应用于请求URI标识的资源（即忽略不存在属性的POST请求），并返回一个携带成功响应代码（2.xx）的响应；或
- respond with an error response code (4.xx or 5.xx). Note that if the representation in the payloadis incompatible with the target Resource for POST using the applied OCF Interface (i.e. theoverwrite semantic cannot be honored because of read-only Property in the payload), then the error response code 4.xx shall be returned.
  响应一个错误响应代码（4.xx or 5.xx）。请注意，如果负载中的表示与使用应用的OCF接口的POST的目标资源不兼容（即由于负载中的只读属性，不能使用覆盖语义），则错误响应代码4.xx将被返回。

12.2.3.5 DELETE with DELETE 删除操作

DELETE shall be used for DELETE operation. The DELETE method requests that the Resourceidentified by the request URI be deleted.
DELETE操作应使用DELETE请求。DELETE方法请求删除请求URI标识的资源。

Upon receiving the DELETE request, the Server shall either:
在收到删除请求后，服务器应:

- Delete the target Resource and send back a response with a success response code (2.xx); or
  删除目标资源，返回一个携带成功响应代码（2.xx）的响应;或
- respond with an error response code (4.xx or 5.xx).
  响应一个错误响应代码（4.xx或5.xx）。

DELETE is unsafe but idempotent (unless URIs are recycled for new instances).
删除是不安全的，但具有等冥性（除非将uri回收用于新实例）。

12.2.4 Content-Format negotiation 内容格式协商

The Framework mandates support of CBOR, however it allows for negotiation of the payload body if more than one Content-Format (e.g. CBOR and JSON) is supported by an implementation. In this case the Accept Option defined in clause 5.10.4 of IETF RFC 7252 shall be used to indicate which Content-Format (e.g. JSON) is requested by the Client.
框架要求支持CBOR，但是如果实现支持一种以上的Content-Format（例如CBOR和JSON），则允许协商

有效负载主体。在这种情况下，应使用IETF RFC 7252中5.10.4条定义的Accept选项来指明客户端请求所用的内容格式（例如JSON）。

The Content-Formats supported are shown in Table 32.
表32列出了支持的内容格式。

Table 32 - OCF Content-Formats
表32- OCF 内容格式

Media Type 媒介类型	ID
"application/vnd.ocf+cbor"	10000

Clients shall include a Content-Format Option in every message that contains a payload. Servers Shall include a Content-Format Option for all success (2.xx) responses with a payload body. PerIETF RFC 7252 clause 5.5.1, Servers shall include a Content-Format Option for all error (4.xx or5.xx) responses with a payload body unless they include a Diagnostic Payload; error responses with a Diagnostic Payload do not include a Content-Format Option. The Content-Format Option Shall use the ID column numeric value from Table 32. An OCF vertical may mandate a specific Content-Format Option.
客户端应在包含有效负载的每个消息中包含Content-Format选项。服务器应包含包含有效负载主体的所有成功（2.xx）响应的Content-Format选项。根据IETF RFC 7252第5.5.1条，服务器应包括所有错误的Content-Format 选项（4.xx或5.xx）与有效载荷体响应，除非包括诊断有效载荷；带有诊断负载的错误响应不包括Content-Format选项。Content-Format选项应该使用表32中的ID列数值。OCF垂直可能要求使用特定的内容格式选项。

Clients shall also include an Accept Option in every request message. The Accept Option shall indicate the required Content-Format as defined in Table 32 for response messages. The Servershall return the required Content-Format if available. If the required Content-Format cannot bereturned, then the Server shall respond with an appropriate error message.
客户端还应该在每个请求消息中包含一个Accept选项。Accept选项应为响应消息指明表32中定义的所需的Content-Format。服务器应返回所需的内容格式(如果可用)。如果无法返回所需的内容格式，则服务器应返回适当的错误消息。

12.2.5 OCF-Content-Format-Version information OCF-Content-Format-Version 信息

Servers and Clients shall include the OCF-Content-Format-Version Option in both request and response messages with a payload. Clients shall include the OCF-Accept-Content-Format-Version Option in request messages. The OCF-Content-Format-Version Option and OCF-Accept-Content-Format-Version Option are specified as Option Numbers in the CoAP header as shown in Table 33.
服务器和客户端应在请求和响应消息的有效负载中带有OCF-Content-Format-Version选项。客户端应在请求消息中包含OCF-Accept-Content-Format-Version选项。OCF-Content-Format-Version选项和OCF-Accept-Content-Format-Version选项被指定为CoAP头部中的选项号，如表33所示。

Table 33 - OCF-Content-Format-Version and OCF-Accept-Content-Format-Version OptionNumbers
表33 - OCF-Content-Format-Version 和OCF-Accept-Content-Format-Version选项号

CoAP Option Number CoAP 选项号	Name 名称	Format 格式	Length 长度(bytes)
2049	OCF-Accept-Content-Format-Version	unit	2
2053	OCF-Content-Format-Version	unit	2

The value of both the OCF-Accept-Content-Format-Version Option and the OCF-Content-Format-Version Option is a two-byte unsigned integer that is used to define the major, minor and subversions. The major and minor versions are represented by 5 bits and the sub version is represented by 6 bits as shown in Table 34.

OCF-Accept-Content-Format-Version 选项 和 OCF-Content-Format-Version 选项的数值是一个双字节无符号整数，用于定义主版本、次版本和子版本。主版本和次版本用5位表示，子版本用6位表示，如表34所示。

Table 34 - OCF-Accept-Content-Format-Version and OCF-Content-Format-VersionRepresentation
表34 - OCF-Accept-Content-Format-Version 和 OCF-Content-Format-Version表示

	Major Version 主版本					Minor Version 主版本					Sub Version 主版本					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Table 35 illustrates several examples:

表35列举几个例子

Table 35 - Examples of OCF-Content-Format-Version and OCF-Accept-Content-Format-Version Representation
表35 - OCF-Accept-Content-Format-Version 和 OCF-Content-Format-Version表示示例

OCF version OCF版本	Binary representation 二进制表示法	Integer value 整数值
"1.0.0"	"0000 1000 0000 0000"	2048
"1.0.0"	"0000 1000 0100 0000"	2112

The OCF-Accept-Content-Format-Version Option and OCF-Content-Format-Version Option for this version of the document shall be "1.0.0" (i.e. 0b0000 1000 0000 0000").

本文件此版本的OCF-Accept-Content-Format-Version选项和OCF-Content-Format-Version选项为“1.0.0”（如，"0b0000 1000 0000 0000"）

12.2.6 Content-Format policy Content-Format政策

All Devices shall support the current Content-Format Option, "application/vnd.ocf+cbor", and OCF-Content-Format-Version "1.0.0".
所有设备应支持当前Content-Format选项，"application/vnd.ocf+cbor"和OCF- Content-Format-Version "1.0.0"。

For backward compatibility with previous OCF-Content-Format-Version Options:
为了向后兼容先前的OCF-Content-Format-Version选项：

- All Client Devices shall support OCF-Content-Format-Version Option set to "1.0.0" and higher.
  所有客户端设备应支持OCF-Content-Format-Version选项设置为“1.0.0”或更高。
- All Client Devices shall support OCF-Accept-Content-Format-Version Option set to "1.0.0" and higher.
  所有客户端设备应支持OCF-Accept-Content-Format-Version选项设置为“1.0.0”或更高。
- A Client shall send a discovery request message with its Accept Option set to "application/vnd.ocf+cbor", and its OCF-Accept-Content-Format-Version Option matching its highest supported version.
  客户端应发送一个发现请求消息，其接受选项设置为"application/vnd.ocf+cbor"，其OCF-Accept-Content-Format-Version应匹配最高支持的版本。
- A Server shall respond to a Client's discovery request that is higher than its OCF-Content-Format-Version by responding with its Content-Format Option set to "application/vnd.ocf+cbor", and OCF-Content-Format-Version matching its highest supported version. The response representation shall be encoded with the OCF-Content-Format-Version matching the Server's highest supported version.
  服务器应通过将其内容格式选项设置为“application/vnd”来响应高于其OCF-Content-Format-Version的客户端发现请求。以及匹配其最高支持版本的ocf - content - format - version。响应表示应该使用与服务器支持的最高版本相匹配的OCF-Content-Format-Version进行编码。
- A Server may support previous Content-Formats and OCF-Content-Format-Versions to support backward compatibility with previous versions.
  服务器可以支持以前的内容格式和OCF-Content-Format-Versions来支持与以前版本的向后兼容性。
- For a Server that supports multiple OCF-Content-Format-Version Options, the Server should attempt to respond with an OCF-Content-Format-Version that matches the OCF-Accept-Content-Format-Version of the request.
  对于支持多个OCF-Content-Format-Version选项的服务器，服务器应该尝试使用与请求的OCF-Accept- Content-Format-Version匹配的OCF-Content-Format-Version进行响应。

To maintain compatibility between Devices implemented to different versions of this document, Devices should follow the policy as described in Figure 15.
为了保持不同版本的设备之间的兼容性，设备应该遵循图15中描述的策略。

The OCF Clients in Figure 15 support sending Content-Format Option set to "application/vnd.ocf+cbor", Accept Option set to "application/vnd.ocf+cbor", OCF-Content-Format- Version Option set to "1.0.0", and OCF-Accept-Content-Format-Version Option set to "1.0.0" (representing OCF 1.0 and later Clients). The OCF Servers in Figure 15 support sending Content- Format Option set to "application/vnd.ocf+cbor" and OCF-Content-Format-Version Option set to"1.0.0"(representing OCF 1.0 and later Servers).
图15中的OCF客户端支持将内容格式选项设置为“application/vnd.ocf+cbor”，接受选项设置为"application/vnd.ocf+cbor"，OCF-Content-Format- Version选项设置为"1.0.0"，OCF-Accept-Content-Format-Version选项设置为“1.0.0”（表示OCF 1.0及以后的客户端）。图15中的OCF服务器支持将内容格式选项设置为"application/vnd.ocf+cbor"以及OCF-Content-Format-Version Option设置为"1.0.0"（表示OCF 1.0及以后的服务器）。

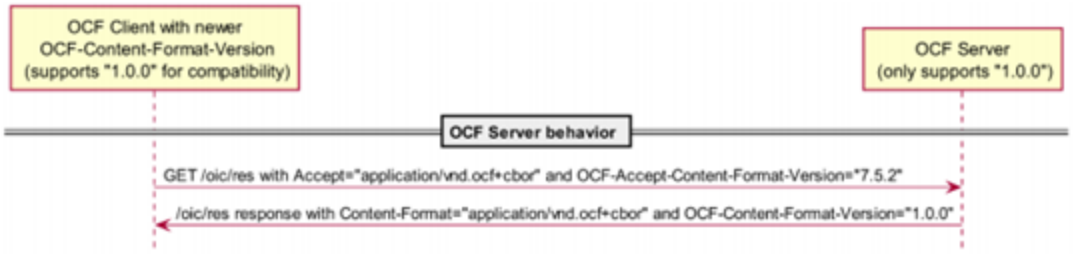

Figure 15 - Content-Format Policy for backward compatible OCF Clients negotiating lower OCF Content-Format-Version
图15-后兼容OCF客户端协商的Content-Format 策略低于OCF Content-Format-Version

12.2.7 CRUDN to CoAP response codes CRUDN 对CoAP响应编码

The mapping of CRUDN operations response codes to CoAP response codes are identical to the response codes defined in IETF RFC 7252.
CRUDN操作响应代码到CoAP响应代码的映射与IETF RFC 7252中定义的响应代码相同。

12.2.8 CoAP block transfer CoAP块传输

Basic CoAP messages work well for the small payloads typical of light-weight, constrained IoT devices. However scenarios can be envisioned in which an application needs to transfer large payloads.
基本CoAP消息适用于典型的轻型、受限IoT设备的小型有效载荷。但是，可以设想应用程序需要传输更大的有效载荷的场景。

CoAP block-wise transfer as defined in IETF RFC 7959 shall be used by all Servers which generate a content payload that would exceed the size of a CoAP datagram as the result of handling any defined CRUDN operation.
IETF RFC 7959应使用中定义的CoAP块传输，由所有服务器生成的内容载荷将超过处理任何已定义的CRUDN操作的CoAP数据报的大小。

Similarly, CoAP block-wise transfer as defined in IETF RFC 7959 shall be supported by all Clients.The use of block-wise transfer is applied to both the reception of payloads as well as transmission of payloads that would exceed the size of a CoAP datagram.
类似地，所有客户端都应该支持IETF RFC 7959中定义的CoAP块传输。块传输的使用既适用于有效载荷的接收，也适用于超过CoAP数据报大小的有效载荷的传输。

All blocks that are sent using this mechanism for a single instance of a transfer shall all have the same reliability setting (i.e. all confirmable or all non-confirmable).
使用该机制为单个传输实例发送的所有块应该具有相同的可靠性设置（即所有可验证的或所有不可验证的）。

A Client may support both the block1 (as descriptive) and block2 (as control) options as describedby IETF RFC 7959. A Server may support both the block1 (as control) and block2 (as descriptive)options as described by IETF RFC 7959.
客户端可以支持IETF RFC 7959所描述的block1（作为描述性的）和block2（作为控制的）选项。服务器可以支持IETF RFC 7959所描述的block1（作为控制）和block2（作为描述性）选项。

12.2.9 Generic requirements for CoAP multicast CoAP多播的一般要求

A Client may use CoAP multicast to retrieve a target Resource with a fixed local path from multiple other Devices. This clause provides generic requirements for this mechanism.
客户端可以使用CoAP多播从多个其他设备检索具有固定本地路径的目标资源。本章节提供了此机制的一般要求。

- Devices shall join the All OCF Nodes multicast groups (as defined in [IANA IPv6 MulticastAddress Space Registry]) with scopes 2, 3, and 5 (i.e., ff02::158, ff03::158 and ff05::158) and shall listen on the port 5683. For compliance to IETF RFC 7252 a Device may additionally join the All CoAP Nodes multicast groups.
  设备应加入所有OCF节点组播定义在（[IANA IPv6多播地址空间注册表]）范围2,3和5(即。，ff02::158, ff03::158和ff05::158)，并监听端口5683。为了符合IETF RFC 7252，一个设备可以另外加入所有CoAP节点的多播组。
- Clients intending to discover Resources shall join the multicast groups as defined in the first bullet.
  打算发现资源的客户端应加入第一个项目符号中定义的多播组。

- Clients shall send multicast requests to the All OCF Nodes multicast group address with scope2 (Mff02::158M) at port "5683". The requested URI shall be the fixed local path of the targetResource optionally followed by query parameters. For compliance to IETF RFC 7252 a Clientmay additionally send to the All CoAP Nodes multicast groups.
  客户端在“5683”端口向范围为2 （Mff02::158M）的所有OCF节点组播地址发送组播请求。请求的URI应该是目标资源的固定本地路径（可选），后跟查询参数。为了符合IETF RFC 7252的要求，客户端还可以向所有CoAP节点发送多播组。
- To discover Devices on a low-rate wireless personal area network (LR-WPAN) [see IETF RFC 7346], Clients should send additional discovery requests (GET request) to the AHOCF Nodes multicast group address with REALM_LOCAL scope 3 ("ff03::158") at port "5683".The set of replying Devices then can be used to distinguish if the Device is SITE_LOCAL orREALM_LOCAL to the Client discovering the Devices. Such request shall use the IPv6 hop limitwith a value of 255. If the Client sends discovery requests to AH OCF Nodes, then forcompliance to IETF RFC 7252 a Client may additionally send to the AH CoAP Nodes multicastgroups with the same REALM_LOCAL scope with the IPv6 hop limit value of 255.
  发现低速率无线个人区域网络（LR-WPAN）上的设备（参见IETF RFC 7346），客户端应该在端口“5683”向带有REALM_LOCAL scope 3（“ff03::158”）的AH OCF节点组播地址发送额外的发现请求（GET请求）。然后，可以使用应答设备集来区分该设备对于发现设备的客户机是SITE_LOCAL还是REALM_LOCAL。这样的请求将使用值255的IPv6跃点限制。如果客户端向AH OCF节点发送发现请求，那么为了遵从IETF RFC 7252，客户端可以额外向AH CoAP节点多播组发送REALM_LOCAL作用域相同的IPv6跳限值255。
- Clients should send discovery requests (GET request) to the AH OCF Nodes multicast groupaddress with SITE_LOCAL scope 5 ("ff05::158") at port "5683". Such request shall use the IPv6hop limit with a value of 255. If the Client sends discovery requests to AH OCF Nodes, then forcompliance to IETF RFC 7252 a Client may additionally send to the AH CoAP Nodes multicastgroups with the same SITE_LOCAL scope with the IPv6 hop limit value of 255.
  客户端应将发现请求（GET请求）发送到端口为“5683”的、SITE_LOCAL scope 5（“ff05::158”）的AH OCF节点组播地址。这样的请求将使用值255的IPv6跃点限制。如果客户端向AH OCF节点发送发现请求，那么为了遵从IETF RFC 7252，客户端可以额外向AH CoAP节点发送具有相同SITE_LOCAL作用域的多播组，IPv6跳限值为255。
- The multicast request shall be permitted by matching the request to an ACE which permitsunauthenticated access to the target Resource as described in ISO/IEC 30118-2:2018.
  多播请求应与允许对目标资源进行未经身份验证访问的ACE匹配，如ISO/IEC 30118-2:20 08所述。
- Handling of multicast requests shall be as described in clause 8 of IETF RFC 7252 and clause4.1 in IETF RFC 6690.
  多播请求的处理应按照IETF RFC 7252第8条和IETF RFC 6690第4.1条的规定进行。
- Devices which receive the request shall respond, subject to query parameter processingspecific to the requested Resource.
  接收请求的设备应根据特定于被请求资源的查询参数处理进行响应。

12.3 Mapping of CRUDN to CoAP serialization over TCP 基于TCP的CRUDN到CoAP序列化的映射

12.3.1 Overview 综述

In environments where TCP is already available, CoAP can take advantage of it to provide reliability.Also in some environments UDP traffic is blocked, so deployments may use TCP. For example,consider a cloud application acting as a Client and the Server is located at the user's home. A Server which already support CoAP as a messaging protocol could easily support CoAP serialization over TCP rather than utilizing another messaging protocol. A Device implementing CoAP Serialization over TCP shall conform to IETF RFC 8323.
在TCP已经可用的环境中，CoAP可以利用它来提供可靠性。而且在一些环境中UDP通信被阻塞，因此部署可能使用TCP。例如，考虑一个充当客户端的云应用程序，服务器位于用户家中。已经支持CoAP作为消息传递协议的服务器可以很容易地通过TCP支持CoAP序列化，而不需要使用其他消息传递协议。通过TCP实现CoAP序列化的设备应符合IETF RFC 8323。

12.3.2 URIs URIs

When UDP is blocked, Clients are dependent on pre-configured details of the Device to determine if the Device supports CoAP serialization over TCP. When UDP is not-blocked, a Device which supports CoAP serialization over TCP shall populate the "eps" Parameter in the "/oic/res" response,as defined in 10.2, with the URI scheme(s) as defined in clause 8.1 or 8.2 of IETF RFC 8323. For The "coaps+tcp" URI scheme, as defined in clause 8.2 of IETF RFC 8323, IETF RFC 7301 shall be used. In addition, the URIs used for CoAP serialization over TCP shall conform to 12.2.2 by substituting the scheme names with the scheme names defined in clauses 8.1 and 8.2 ofIETF RFCrespectively.
当UDP被阻塞时，客户端依赖于预先配置的设备细节来确定设备是否支持通过TCP进行CoAP序列化。当UDP没有被阻塞时，一个支持通过TCP进行CoAP序列化的设备应该使用IETF RFC 8323的8.1或8.2条款中定义的URI模式填充10.2中定义的"/oic/res"响应中的"eps"参数。IETF RFC 8323的第8.2章节和IETF RFC 7301中定义的"coaps+tcp"URI方案应该被使用。此外，用于TCP上CoAP序列化的uri应符合12.2.2章节，用IETF RFC第8.1和8.2条中定义的方案名代替方案名。

12.3.3 CoAP method with request and response 带有请求和响应的CoAP方法

The CoAP methods used for CoAP serialization over TCP shall conform to 12.2.3.
用于TCP上CoAP序列化的CoAP方法应符合12.2.3。

12.3.4 Content-Format negotiation Content-Format协议

The Content Format negotiation used for CoAP serialization over TCP shall conform to 12.2.4.
用于TCP上CoAP序列化的内容格式协议应符合12.2.4章节。

12.3.5 OCF-Content-Format-Version information OCF-Content-Format-Version信息

The OCF Content Format Version information used for CoAP serialization over TCP shall conform to 12.2.5.
用于TCP上CoAP序列化的OCF内容格式版本信息应符合12.2.5章节。

12.3.6 Content-Format policy Content-Format 政策

The Content Format policy used for CoAP serialization over TCP shall conform to 12.2.6.
用于TCP上CoAP序列化的内容格式政策应符合12.2.6章节。

12.3.7 CRUDN to CoAP response codes 从CRUDN到CoAP响应代码

The CRUDN to CoAP response codes for CoAP serialization over TCP shall conform to 12.2.7.
用于TCP上CoAP序列化的CRUDN到CoAP响应代码应符合12.2.7章节。

12.3.8 CoAP block transfer CoAP块传输

The CoAP block transfer for CoAP serialization over TCP shall conform to clause 6 of IETF RFC 8323.
用于在TCP上进行CoAP序列化的CoAP块传输应符合 IETF RFC 8323的第6章。

12.3.9 Keep alive (connection health) 保持生机（连接健康）

The Device that initiated the CoAP over TCP connection shall send a Ping message as described in clause 5.4 in IETF RFC 8323. The Device to which the connection was made may send a Ping Message. The recipient of any Ping message shall send a Pong message as described in clause5.4 in IETF RFC 8323.
如IETF RFC 8323中的第5.4条中所述，通过TCP连接发起CoAP的设备将发送一条Ping消息。连接到的设备可能会发送Ping消息。任何Ping消息的接收方应按照IETF RFC 8323中的第5.4条的规定发送一条Pong消息。

Both sides of an established CoAP over TCP connection may send subsequent Ping (and corresponding Pong) messages.
通过TCP连接建立的CoAP的两端可以发送后续的Ping（和相应的Pong）消息。

12.4 Payload Encoding in CBOR CBOR中的有效载荷编码

OCF implementations shall perform the conversion to CBOR from JSON defined schemas and toJSON from CBOR in accordance with IETF RFC 7049 clause 4 unless otherwise specified in thisclause.
除非IETF RFC 7049中第4章有其他规定，OCF的实现应该按照该条款执行从JSON定义的模式转换到CBOR和从CBOR到JSON的转换，。

Properties defined as a JSON integer shall be encoded in CBOR as an integer (CBOR major types0 and 1). Properties defined as a JSON number shall be encoded as an integer, single- or double-precision floating point (CBOR major type 7, sub-types 26 and 27); the choice is implementationdependent. Half-precision floating point (CBOR major 7, sub-type 25) shall not be used. Integernumbers shall be within the closed interval [-2A53, 2A53]. Properties defined as a JSON numbershould be encoded as integers whenever possible; if this is not possible Properties defined as aJSON number should use single-precision if the loss of precision does not affect the quality ofservice, otherwise the Property shall use double-precision.
定义为JSON整数的属性应在CBOR中编码为整数（CBOR主要类型0和1）。定义为JSON数字的属性应编码为整数、单精度或双精度浮点数（CBOR主要类型7、子类型26和27）；选择取决于实现。半精度浮点数（CBOR major 7, sub-type 25）不能使用。整数应在封闭区间内[-2A53, 2A53]。定义为JSON数字的属性应尽可能编码为整数;如果这是不可能的，那么定义为JSON数字的属性应该使用单精度（如果精度损失不影响服务质量），否则属性应该使用双精度。

On receipt of a CBOR payload, an implementation shall be able to interpret CBOR integer valuesin any position. If a Property defined as a JSON integer is received encoded other than as aninteger, the implementation may reject this encoding using a final response as appropriate for theunderlying transport (e.g. 4.00 for CoAP) and thus optimise for the integer case. If a Property isdefined as a JSON number an implementation shall accept integers, single- and double-precisionfloating point.
在接收到CBOR有效载荷时，实现应能够解释任何位置的CBOR整数值。如果一个定义为JSON整数的属性被编码后接收，而不是被编码为整数，那么实现可能会使用适合底层传输的最终响应来拒绝这种编码（例如，CoAP的4.00），从而优化整数的情况。如果属性定义为JSON数字，则实现应接受整数、单精度浮点数和双精度浮点数。

13 Security 安全

The details for handling security and privacy are specified in ISO/IEC 30118-2:2018.
处理安全和隐私的细节在ISO/IEC 30118-2:20 08中有详细说明。

Annex A (normative)Resource Type definitions

附录A(规范性附录)

资源类型定义

A.1 List of Resource Type definitions 资源类型定义列表

All the clauses in Annex A describe the Resource Types with a RESTful API definition language. The Resource Type definitions presented in Annex A are formatted for readability, and so may appear to have extra line breaks. Table A.1 contains the list of defined Core Common Resources in this document.
附录A中的所有章节都使用RESTful API定义语言描述资源类型。附录A中提供的资源类型定义的格式是为了便于阅读，因此可能会出现额外的换行符。表A.1列出了本文件中定义的核心公共资源。

Table A.1 - Alphabetized list of Core Resources
表A.1 - 按字母顺序排列的核心资源列表

Friendly Name (informative) 易记名称（资料性）	Resource Type (rt) 资源类型（rt）	Clause 章节
Atomic Measurement 原子测量	"oic.wk.atomicmeasurement"	A.2
Collections集合	"oic.wk.col"	A.3
Device 设备	"oic.wk.d"	A.4
Discoverable Resource 可发现资源	"oic.wk.res"	A.7
Introspection 自省	"oic.wk.introspection"	A.5
Platform 平台	"oic.wk.p"	A.6

A.2 Atomic Measurement links list representation 原子测量链表表示

A.2.1 Introduction 引言

The oic.if.baseline OCF Interface exposes a representation of the links and the CommonProperties of the Atomic Measurement Resource.
oic.if.baseline OCF 接口展现链接的表示形式和原子测量资源的公共属性。

A.2.2 Example URI URI示例

/AtomicMeasurementResURI

A.2.3 Resource type 资源类型

The Resource Type is defined as: "oic.wk.atomicmeasurement".
资源类型定义为："oic.wk.atomicmeasurement"。

A.2.4 OpenAPI 2.0 definition OpenAPI 2.0定义

```
{
          "swagger": "2.0",
          "info": {
                   "title": "Atomic Measurement links list representation",
                   "version": "2019-03-04",
                   "license": {
        "name": "OCF Data Model License",
        "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
        "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights
reserved."
   },
   "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
      },
      "schemes": ["http"],
      "consumes": ["application/json"],
      "produces": ["application/json"],
      "paths": {
              "/AtomicMeasurementResURI?if=oic.if.ll":
                     { "get": {
                             "description": "The oic.if.ll OCF Interface exposes a representation
of the Links",
                             "parameters": [
       {
           "$ref": "#/parameters/interface-all"
       }
     ],
                             "responses": {
                                     "200": {
                                             "description": "",
                                             "x-example": [{
                                                     "href": "/temperature",
                                                     "rt": ["oic.r.temperature"],
                                                     "if": ["oic.if.s",
                                                     "oic.if.baseline"]
                                             },
                                             {
                                                     "href": "/bodylocation",
                                                     "rt": ["oic.r.body.location.temperature"],
"if": ["oic.if.s", "oic.if.baseline"]
                                             },
                                             {
                                                     "href": "/timestamp",
                                                     "rt": ["oic.r.time.stamp"],
```

```
                                                "if": ["oic.if.s", "oic.if.baseline"]
                                        }],
                                        "schema": {
                                                "$ref": "#/definitions/links"
                                        }
                                }
                        }
                }
        },
        "/AtomicMeasurementResURI?if=oic.if.b":
                { "get": {
                        "description": "The oic.if.b OCF Interface returns data items
                        retrieved from Resources pointed to by the Links.\n",
                   "parameters": [
   {
      "$ref": "#/parameters/interface-all"
   }
 ],
                   "responses": {
                           "200": {
                                   "description": "Normal response, no errors, all
Properties are returned correctly\n",
                                   "x-example": [{
                                            "href": "/temperature",
                                            "rep": {
                                                   "temperature": 38,
                                                   "units": "C",
                                                   "range": [25,  45]
                                            }
                                   },
                                   {
                                            "href": "/bodylocation",
                                            "rep": {
                                                   "bloc": "ear"
                                            }
                                   },
                                   {
                                            "href": "/timestamp",
                                            "rep": {
                                                   "timestamp": "2007-04-05T14:30+09:00"
                                            }
                                   }],
                                   "schema": {
                                          "$ref": "#/definitions/batch-retrieve"
                                   }
                           }
                    }
                }
     },
     "/AtomicMeasurementResURI?if=oic.if.baseline": {

                   "get": {
```

```
                        "description": "The oic.if.baseline OCF Interface exposes a
representation of the links and\nthe Common Properties of the Atomic Measurement Resource.\n",
                        "parameters": [
  {
     "$ref": "#/parameters/interface-all"
  }
],
                        "responses": {
                                "200": {
                                        "description": "",
                                        "x-example": {
                                               "rt": ["oic.wk.atomicmeasurement"],
                                               "if": ["oic.if.b", "oic.if.ll",
"oic.if.baseline"],
                                               "rts": ["oic.r.temperature",
"oic.r.body.location.temperature", "oic.r.time.stamp"],
                                               "rts-m": ["oic.r.temperature",
"oic.r.body.location.temperature", "oic.r.time.stamp"],
                                               "links": [{
                                                     "href": "/temperature", "rt":
                                                     ["oic.r.temperature"],
                                                     "if": ["oic.if.s", "oic.if.baseline"]
                                              },
                                              {
                                                     "href": "/bodylocation", "rt":
["oic.r.body.location.temperature"],
                                                     "if": ["oic.if.s", "oic.if.baseline"]
                                              },
                                              {
                                                     "href": "/timestamp",
                                                     "rt": ["oic.r.time.stamp"],
                                                     "if": ["oic.if.s", "oic.if.baseline"]
                                             }]
                                       },
                                       "schema": {
                                              "$ref": "#/definitions/baseline"
                                       }
                                }
                        }
                }
        },
        "parameters": {
                "interface-all": {
                        "in": "query",
                        "name": "if",
                        "type": "string",
                        "enum": ["oic.if.b", "oic.if.ll",   "oic.if.baseline"]
                }
        },
```

```
        "definitions": {
                "links": {
                "type": "array",
                "items": {
                         "$ref": "#/definitions/oic.oic-link"
                       }
              },
              "batch-retrieve": {
                     "title": "Collection Batch Retrieve Format (auto merged)",
                     "minItems": 1,
                     "items": {
                            "additionalProperties": true,
                            "properties": {
                                    "href": {
                                         "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
                                  },
                                  "rep": {
                                     "oneOf": [{
                                              "description": "The response payload
from a single Resource",
                                              "type": "object"
                                    },
                                    {
                                              "description": " The response payload
from a Collection (batch) Resource",
                                              "items": {
                                                    "properties": {
                                                             "anchor": {
                                         "$ref":

"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
                                   },
                                    "di": {
                                       "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
                                   },
                                   "eps": {

                                      "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
                                  },
                                   "href": {
                                     "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
                                  },
                                                      "if": {
                                                           "description": "The OCF
```

```
Interface set supported by this Resource",
                                                        "items": {
                                                                "enum": [
                                                                "oic.if.baseline",
                                                                "oic.if.ll",
                                                                "oic.if.b",
                                                                "oic.if.rw",
                                                                "oic.if.r",
                                                                "oic.if.a",
                                                                "oic.if.s"],
                                                                "type":
                                                                "string"
                                                         },
                                                         "minItems": 1,
                                                         "uniqueItems": true,
                                                         "type": "array"
                                                },
                                                "ins": {
                                    "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
                                }, "p":
                                {
                                     "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
                                },
                                "rel": {
                                   "description": "The relation of the target URI referenced by
the Link to the context URI",
                                  "oneOf": [
                                     {
                                        "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
                                     },
                                     {
                                     "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_string"
                                     }
                                   ]
                                 },
                                                        "rt": {
                                                                "description":
"Resource Type of the Resource",
                                                                "items": {
                                                                       "maxLength":
64,
                                                                       "type":
```

```
"string"
                                                                    },
                                                                "minItems": 1,
                                                                "uniqueItems": true,
                                                                "type": "array"
                                                            },
                                                            "title": {
                                        "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
                                    },
                                    "type": {
                                        "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
                                    }
                                                    },
                                                    "required": [
                                                      "href",
                                                      "rt",
                                                      "if"
                                                    ],
                                                    "type": "object"
                                                },
                                                "type": "array"
                                    }]
                            }
                        },
                        "required": [
                           "href",
                           "rep"
                        ],
                        "type": "object"
                    },
                     "type": "array"
               },
               "baseline": {
                       "properties": {
                            "links": {
                                 "description": "A set of simple or individual Links.",
                                 "items": {
                                       "$ref": "#/definitions/oic.oic-link"
                                   },
                                   "type": "array"
                             },
                             "n": { "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"},
```

```
            "id": { "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"},
                    "rt": {
                "description": "Resource Type of this Resource", "items": {
                "enum": ["oic.wk.atomicmeasurement"],
                "type": "string",
                "maxLength": 64
             },
             "minItems": 1,
             "readOnly": true,
             "uniqueItems": true,
             "type": "array"
          },
                                  "rts": {
                                         "description": "An array of Resource Types that are
supported within an array of Links exposed by the Resource",
                                         "items": {
                                                 "maxLength": 64,
                                                 "type": "string"
                                         },
                                         "minItems": 1,
                                         "readOnly": true,
                                         "uniqueItems": true,
                                         "type": "array"
                                  },
                                  "rts-m": {
                                        "description": "An array of Resource Types that are
mandatory to be exposed within an array of Links exposed by the Resource",
                                       "items": {
                                              "maxLength": 64,
                                              "type": "string"
                                       },
                                      "minItems": 1,
                                      "readOnly": true,
                                      "uniqueItems": true,
                                      "type": "array"
                                },
                                "if": {
                                            "description": "The OCF Interface set supported by
this Resource",
                                            "items": {
                                                    "enum": ["oic.if.b", "oic.if.ll",
"oic.if.baseline"],
                                                    "type": "string"
                                            },
                                            "minItems": 3,
               "readOnly": true,
               "uniqueItems": true,
               "type": "array"
```

```
                                        }
                                },
                                "type": "object",
                                "required": [
          "rt",
          "if",
          "links"
        ]
                    },
                     "oic.oic-link": {
                            "properties": {
                                    "anchor": {
             "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
         },
          "di": {
               "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
        },
         "eps": {
           "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
        },
         "href": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
        },
                                 "if": {
                                    "description": "The OCF Interface set supported by this Resource
                                    "items": {
                                         "enum":
                                         [ "oic.if.baseline",
                                         "oic.if.ll",
                                         "oic.if.b",
                                         "oic.if.rw",
                                         "oic.if.r",
                                         "oic.if.a",
                                         "oic.if.s"],
                                         "type": "string"
                                },
                                "minItems": 1,
                                "uniqueItems": true,
                                "type": "array"
                        },
                        "ins": {
            "$ref":
```

```
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
         }, "p":
         {
             "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
         },
         "rel": {
            "description": "The relation of the target URI referenced by the Link to the context
URI",
            "oneOf": [
               {
                 "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
               },
               {
                   "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_string"
               }
            ]
       },
                            "rt": {
                                    "description": "Resource Type of the Resource",
                                    "items": {
                                           "maxLength": 64,
                                           "type": "string"
                                    },
                                    "minItems": 1,
                                    "uniqueItems": true,
                                    "type": "array"
                            },
                            "title": {
            "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
          },
          "type": {
             "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
         }
                        },
                        "required": [
                                "href",
                         "rt",
                         "if"
                        ],
```

```
                        "type": "object"
                }
        }
}
```

A.2.5 Property definition 属性定义

Table A.2 defines the Properties that are part of the "oic.wk.atomicmeasurement" Resource Type.

表A.2定义了“oic.wk atomicmeasurement“资源类型的一部分属性。

Table A.2 - The Property definitions of the Resource with type "rt"="oic.wk.atomicmeasurement".
表A.2 - 具有"rt"="oic.wk.atomicmeasurement" 资源的属性定义.

Property Name 属性名称	Value Type 数值类型	Mandatory 强制	Access mode 访问模式	Description 描述
href	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
rep	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
links	arrays: see schema 数组:参见模式	Yes 是	Read Write 可读写	A set of simple or individual Links. 一组简单或独立的链接。
n	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
id	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
rt	arrays: see schema 数组:参见模式	Yes 是	Read Only 仅可读	Resource Type of this Resource 此资源的资源类型
rts	arrays: see schema 数组:参见模式	No 否	Read Only 仅可读	An array of Resource Types that are supported within an array of Links exposed by the Resource 资源公开的链接数组中支持的资源类型数组
rts-m	arrays: see schema 数组:参见模式	No 否	Read Only 仅可读	An array of Resource Types that are mandatory to be exposed within an array of Links exposed by the Resource 在资源公开的链接数组中必需公开的资源类型数组
if	arrays: see schema 数组:参见模式	Yes 是	Read Only 仅可读	The OCF Interface set supported by this Resource 该资源支持OCF 接口集
anchor	multiple types: see schema	No 否	Read Write 可读写	

	多种类型:参见模式			
di	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
eps	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
href	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
if	arrays: see schema 数组:参见模式	Yes 是	Read Write 可读写	The OCF Interface set supported by this Resource 该资源支持OCF 接口集
ins	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
p	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
rel	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	The relation of the target URI referenced by theLink to the context URI 链接引用的目标URI 与上下文URI的关系
rt	arrays: see schema 数组:参见模式	Yes 是	Read Write 可读写	Resource Type of the Resource 资源的资源类型
title	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
type	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	

A.2.6 CRUDN behavior CRUDN行为

Table A.3 defines the CRUDN operations that are supported on the "oic.wk.atomicmeasurement" Resource Type.
表A.3定义了"oic.wk.atomicmeasurement"资源类型支持的CRUDN操作。

Table A.3 - The CRUDN operations of the Resource with type "rt"="oic.wk.atomicmeasurement".
表A.3 - CRUDN操作中类型为“rt”="oic.wk.atomicmeasurement"的资源。

Create 创建	Read 可读	Update 更新	Delete 删除	Notify通知

	get 得到			observe 观察

A.3 Collection 集合

A.3.1 Introduction 引言

Collection Resource Type contains Properties and Links.The oic.if.baseline OCF Interface exposes a representation of the Links and the Properties of the Collection Resource itself
集合资源类型包含属性和链接。oic.if.baseline OCF 接口公开链接的表示形式和集合资源本身的属性

A.3.2 Example URI URI示例

```
/CollectionResURI
```

A.3.3 Resource type 资源类型

The Resource Type is defined as: "oic.wk.col".
资源类型定义为："oic.wk.col"。

A.3.4 OpenAPI 2.0 definition OpenAPI 2.0定义

```
{
   "swagger": "2.0",
   "info": {
     "title": "Collection",
     "version": "2019-03-04",
     "license": {
       "name": "OCF Data Model License",
       "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights
reserved."
     },
     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
   },
   "schemes":
      [ "http"
   ],
   "consumes":
      [ "application/json"
   ],
   "produces":
      [ "application/json"
   ],
   "paths":
      { "/CollectionResURI?if=oic.if.ll" :
      {
```

```
         "get": {
            "description": "Collection Resource Type contains Properties and Links.\nThe
oic.if.ll OCF Interface exposes a representation of the Links\n",
        "parameters": [
            {
                "$ref": "#/parameters/interface-all"
            }
         ],
         "responses": {
            "200": {
              "description" : "",
              "x-example": [
                {
                   "href": "/switch",
                   "rt":    ["oic.r.switch.binary"],
                   "if":    ["oic.if.a", "oic.if.baseline"],
                  "eps": [
                     {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                     {"ep": "coaps://[fe80::b1d6]:1122"},
                     {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                  ]
              },
              {
                  "href": "/airFlow",
                  "rt":       ["oic.r.airflow"],
                  "if":       ["oic.if.a", "oic.if.baseline"],
                  "eps": [
                    {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                    {"ep": "coaps://[fe80::b1d6]:1122"},
                    {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                 ]
             }
         ],
         "schema": {
             "$ref": "#/definitions/slinks"
          }
        }
      }
    }
 },
"/CollectionResURI?if=oic.if.baseline" :
{ "get": {
    "description": "Collection Resource Type contains Properties and Links.\nThe oic.if.baseline
OCF Interface exposes a representation of\nthe Links and the Properties of the Collection
Resource itself\n",
    "parameters": [
      {
         "$ref": "#/parameters/interface-all"
      }
    ],
```

```
    "responses": {
       "200": {
          "description" : "",
          "x-example": {
              "rt": ["oic.wk.col"],
              "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
              "rts": [ "oic.r.switch.binary", "oic.r.airflow" ],
              "rts-m": [ "oic.r.switch.binary" ],
              "links": [
                 {
                    "href": "/switch",
                    "rt":     ["oic.r.switch.binary"],
                    "if":     ["oic.if.a", "oic.if.baseline"],
                    "eps": [
                       {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                       {"ep": "coaps://[fe80::b1d6]:1122"},
                       {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
                 ]
              },
              {
                 "href": "/airFlow",
                 "rt":["oic.r.airflow"],
                 "if":["oic.if.a", "oic.if.baseline"],
                 "eps": [
                    {"ep": "coap://[fe80::b1d6]:1111", "pri": 2},
                    {"ep": "coaps://[fe80::b1d6]:1122"},
                    {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
                 ]
               }
            ]
         },
         "schema": {
            "$ref": "#/definitions/sbaseline"
         }
      }
    }
},
"post": {
    "description": "Update on Baseline OCF Interface\n",
    "parameters": [
       {
           "$ref": "#/parameters/interface-update"
       },
       {
           "name": "body",
           "in": "body",
           "required": true,
           "schema": {
              "$ref": "#/definitions/sbaseline-update"
           }
```

```
        }
    ],
    "responses": {
       "200": {
         "description" : "",
         "schema": {
            "$ref": "#/definitions/sbaseline"
          }
        }
      }
   }
},
"/CollectionResURI?if=oic.if.b" :
    { "get": {
     "description": "Collection Resource Type contains Properties and Links.\nThe oic.if.b OCF
Interfacce exposes a composite representation of the\nResources pointed to by the Links\n",
     "parameters": [
       {
          "$ref": "#/parameters/interface-all"
       }
     ],
     "responses": {
         "200": {
            "description" : "All targets returned OK status",
            "x-example": [
               {
                   "href": "/switch",
                   "rep": {
                      "value": true
                   }
               },
               {
               "href": "/airFlow",
               "rep": {
                 "direction": "floor",
                 "speed":     3
               }
            }
         ],
         "schema": {
             "$ref": "#/definitions/sbatch-retrieve"
        }
     },
     "404": {
        "description" : "One or more targets did not return an OK status, return a representation
containing returned Properties from the targets that returned OK",
        "x-example": [
            {
                "href": "/switch",
                "rep": {
```

```
                    "value": true
                }
             }
          ],
          "schema": {
             "$ref": "#/definitions/sbatch-retrieve"
          }
       }
    }
},
"post": {
    "description": "Update on Batch OCF Interface\n",
    "parameters": [
      {
           "$ref": "#/parameters/interface-update"
      },
      {
         "name": "body",
         "in": "body",
         "required": true,
         "schema": {
             "$ref": "#/definitions/sbatch-update"
        },
        "x-example": [
           {
              "href": "/switch",
              "rep": {
                 "value": true
              }
           },
           {
              "href": "/airFlow",
              "rep": {
                 "direction": "floor",
                 "speed": 3
              }
           }
        ]
     }
  ],
  "responses": {
     "200": {
        "description" : "All targets returned OK status, return a representation of the current
state of all targets",
        "x-example": [
           {
              "href": "/switch",
              "rep": {
                 "value": true
              }
```

```
            },
            {
               "href": "/airFlow",
               "rep": {
                  "direction": "demist",
                  "speed": 5
               }
            }
         ],
          "schema": {
             "$ref": "#/definitions/sbatch-retrieve"
          }
        },
      "403": {
         "description" : "One or more targets did not return OK status; return a retrieve
representation of the current state of all targets in the batch",
         "x-example": [
            {
               "href": "/switch",
               "rep": {
                   "value": true
               }
            },
            {
                "href": "/airFlow",
                "rep": {
                    "direction": "floor",
                    "speed": 3
                 }
              }
           ],
           "schema": {
               "$ref": "#/definitions/sbatch-retrieve"
            }
          }
        }
      }
    }
  }
},
"parameters": {
    "interface-all" : {
    "in" : "query",
    "name" : "if",
    "type" : "string",
    "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
   },
    "interface-update" : {
    "in" : "query",
    "name" : "if",
    "type" : "string",
```

```
    "enum" : ["oic.if.b", "oic.if.baseline"]
  }
},
"definitions": {
   "sbaseline" : {
     "properties": {
        "links" : {
           "description": "A set of simple or individual Links.",
           "items": {
              "$ref": "#/definitions/oic.oic-link"
           },
           "type": "array"
        },
       "n": {
          "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
       },
       "id": {
          "$ref" :
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
       },
                "rt": {
            "$ref": "#/definitions/oic.core.rt-col"
       },
                "rts": {
            "$ref": "#/definitions/oic.core.rt"
       },
                "rts-m": {
            "$ref": "#/definitions/oic.core.rt"
       },
                "if": {
                   "description": "The OCF Interfaces supported by this Resource",
                   "items": {
                         "enum": [
                     "oic.if.ll",
                     "oic.if.baseline",
                                      "oic.if.b"
                  ],
                      "type": "string",
                    "maxLength": 64
                   },
                   "minItems": 2,
                   "uniqueItems": true,
                   "readOnly": true,
                   "type":"array"
                   }
          },
          "additionalProperties": true,
```

```
          "type" : "object",
          "required": [
             "rt",
             "if",
             "links"
          ]
       },
       "sbaseline-update": {
            "additionalProperties": true
       },
                 "oic.core.rt-col": {
             "description": "Resource Type of the Resource",
                               "items": {
                                   "enum": ["oic.wk.col"],
                                   "type": "string",
             "maxLength": 64
                              },
                              "minItems": 1,
        "uniqueItems": true,
                             "readOnly": true,
                             "type": "array"
               },
     "oic.core.rt": {
           "description": "Resource Type or set of Resource Types",
                                "items": {
                                       "type": "string",
             "maxLength": 64
                               },
                                "minItems": 1,
            "uniqueItems": true,
                                "readOnly": true,
                                "type": "array"
                    },
         "sbatch-retrieve" : {
              "minItems" : 1,
              "items" : {
                 "additionalProperties": true,
                 "properties": {
                    "href": {
                       "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
                    },
                    "rep": {
                       "oneOf": [
                          {
                              "description": "The response payload from a single Resource",
                              "type": "object"
                          },
                          {
```

```
                            "description": " The response payload from a Collection (batch)
Resource",
                            "items": {
                                "$ref": "#/definitions/oic.oic-link"
                            },
                            "type": "array"
                        }
                    ]
                }
            },
             "required": [
                "href",
                "rep"
             ],
             "type": "object"
          },
           "type" : "array"
        },
         "sbatch-update" : {
              "title" : "Collection Batch Update Format",
              "minItems" : 1,
              "items" :  {
                   "$ref": "#/definitions/sbatch-update.item"
              },
              "type" :  "array"
           },
            "sbatch-update.item" : {
               "additionalProperties": true,
               "description": "Array of Resource representations to apply to the batch Collection,
using href to indicate which Resource(s) in the batch to update. If the href Property is empty,
effectively making the URI reference to the Collection itself, the representation is to be
applied to all Resources in the batch",
              "properties": {
                 "href": {
                     "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
              },
              "rep": {
                 "oneOf": [
                     {
                        "description": "The payload for a single Resource",
                        "type": "object"
                     },
                     {
                         "description": " The payload for a Collection (batch) Resource",
                         "items": {
                             "$ref": "#/definitions/oic.oic-link"
                         },
                         "type": "array"
```

```
                }
              ]
           }
     },
      "required": [
        "href",
        "rep"
     ],
      "type": "object"
  },
   "slinks"  :  {
  "type" : "array",
  "items" : {
      "$ref": "#/definitions/oic.oic-link"
    }
},
"oic.oic-link": {
    "properties": {
       "if": {
           "description": "The OCF Interfaces supported by the Linked target",
           "items": {
           "enum": [
              "oic.if.baseline",
              "oic.if.ll",
              "oic.if.b",
              "oic.if.rw",
              "oic.if.r",
              "oic.if.a",
              "oic.if.s",
              ],
              "type" : "array"
              "maxLength": 64
           },
           "minItems": 1,
           "uniqueItems": true,
           "readOnly": true,
           "type": "array"
        },
        "rt": {
            "$ref": "#/definitions/oic.core.rt"
        },
        "anchor": {
           "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
       },
       "di": {
         "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
```

```
},
      "eps": {
        "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
     },
     "href": {
        "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
     },
      "ins": {
         "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
     },
     "p": {
        "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
    },
     "rel": {
        "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
    },
    "title": {
       "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
    },
    "type": {
       "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"
    },
    "tag-pos-desc": {
       "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/tag-pos-desc"
    },
    "tag-pos-rel": {
       "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/tag-pos-rel"
    },
     "tag-func-desc": {
       "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/tag-func-desc"
```

```
      }
    },
     "required": [
       "href",
       "rt",
       "if"
     ],
     "type": "object"
     }
  }
}
```

A.3.5 Property definition 属性定义

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

表A.4 定义部分属于"oic.wk.col"资源类型的属性。

Table A.4 - The Property definitions of the Resource with type "rt" = "oic.wk.col".

表A.4 - 类型为"rt" = "oic.wk.col"的资源的属性定义。

Property Name 属性名称	Value Type 数值类型	Mandatory 强制	Access mode 访问模式	Description 描述
links	arrays: see schema 数组:参见模式	Yes 是	Read Write 可读写	A set of simple or individual Links. 一组简单或独立的链接。
n	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
id	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
rt	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
rts	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
rts-m	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
if	arrays: see schema 数组:参见模式	Yes 是	Read Only 仅可读	The OCF Interfaces supported by this Resource 该资源支持OCF 接口
href	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
rep	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
href	multiple types: see	Yes 是	Read Write	

	schema 多种类型:参见模式		可读写	
rep	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
if	arrays: see schema 数组:参见模式	Yes 是	Read Only 仅可读	The OCF Interfaces supported by the Linked target 被链接目标支持的OCF 接口
rt	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
anchor	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
di	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
eps	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
href	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
ins	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
p	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
rel	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
title	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
type	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
tag-pos-desc	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
tag-pos-rel	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
tag-func-desc	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	

A.3.6 CRUDN behavior CRUDN行为

Table A.5 defines the CRUDN operations that are supported on the "oic.wk.col" Resource Type.
表A.5 定义了支持"oic.wk.col"资源类型的CRUDN操作。

Table A.5 - The CRUDN operations of the Resource with type "rt" = "oic.wk.col".
表A.5 - 具有类型的资源的CRUDN操作"rt" = "oic.wk.col"。

Create 创建	Read 可读	Update 更新	Delete 删除	Notify通知
	get 得到	post 发布		observe 观察

A.4 Device 设备

A.4.1 Introduction 引言

Known Resource that is hosted by every Server.Allows for logical Device specific information to be discovered.
由每个服务器托管的已知资源。允许发现逻辑设备特定的信息。

A.4.2 Well-known URI 知名的URI

/oic/d

A.4.3 Resource type 资源类型

The Resource Type is defined as: "oic.wk.d".
资源类型定义为："oic.wk.d"。

A.4.4 OpenAPI 2.0 definition OpenAPI 2.0 定义

```
{
        "swagger": "2.0",
        "info": {
            "title": "Device",
            "version": "2019-03-13",
            "license": {
                "name": "OCF Data Model License",
        "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
               "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights
reserved."
         },
         "termsOfService":
```

```
         "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
      },
       "schemes": [
          "http"
      ],
      "consumes": [
          "application/json"
       ],
       "produces": [
           "application/json"
       ],
       "paths": {
          "/oic/d" : {
               "get": {
              "description": "Known Resource that is hosted by every Server.\nAllows for logical
Device specific information to be discovered.\n",
                  "parameters": [
                  {
                     "$ref": "#/parameters/interface"
                  }
                ],
                "responses": {
                    "200": {
                       "description": "",
                       "x-example":
                          {
                             "n":    "Device 1",
                             "rt":["oic.wk.d"],
                             "di":"54919CA5-4101-4AE4-595B-353C51AA983C",
                             "icv":  "ocf.2.0.2",
                             "dmv":  "ocf.res.1.0.0, ocf.sh.1.0.0",
                             "piid": "6F0AAC04-2BB0-468D-B57C-16570A26AE48"
                          },
                          "schema": {
                               "$ref": "#/definitions/Device"
                           }
                      }
                  }
          }
      }
 },
  "parameters": {
  "interface" : {
  "in": "query",
  "name": "if",
  "type": "string",
  "enum": ["oic.if.r", "oic.if.baseline"]
  }
},
"definitions": {
```

```
  "Device": {
  "properties": {
  "rt":  {
    "description": "Resource Type of the Resource",
    "items": {
      "type": "string",
      "maxLength": 64
   },
   "minItems": 1,
   "readOnly": true,
   "uniqueItems": true,
   "type": "array"
},
"ld":  {
  "description": "Localized Descriptions.",
  "items": {
     "properties": {
     "language": {
       "allOf": [
         {
"$ref"           :           "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/language-tag"
         },
         {
           "description": "An RFC 5646 language tag.",
           "readOnly": true
         }
      ]
  },
       "value": {
       "description": "Device description in the indicated language.",
       "maxLength": 64,
      "readOnly": true,
       "type": "string"
   }
},
"type": "object"
},
"minItems": 1,
"readOnly": true,
"type": "array"
},
"piid": {
   "allOf": [
      {
        "$ref"       :       "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/uuid"
      },
      {
              "description":  "Protocol  independent  unique  identifier  for  the  Device  that  is
```

```
immutable.",
         "readOnly": true
      }
   ]
},
"di": {
   "allOf": [
      {
        "$ref"       :       "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/uuid"
     },
     {
        "description": "Unique identifier for the Device",
        "readOnly": true
     }
  ]
},
"dmno":  {
  "description": "Model number as designated by manufacturer.",
  "maxLength": 64,
  "readOnly": true,
  "type": "string"
},
"sv":  {
   "description": "Software version.",
   "maxLength": 64,
   "readOnly": true,
   "type": "string"
},
"dmn":  {
   "description": "Manufacturer Name.",
   "items": {
     "properties": {
       "language": {
          "allOf": [
            {
       "$ref"       :       "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
schema.json#/definitions/language-tag"
            },
            {
                "description": "An RFC 5646 language tag.",
                "readOnly": true
            }
         ]
    },
   "value": {
   "description": "Manufacturer name in the indicated language.",
   "maxLength": 64,
   "readOnly": true,
```

```
   "type": "string"
  }
},
"type": "object"
},
"minItems": 1,
"readOnly": true,
"type": "array"
},
"icv": {
   "description": "The version of the Device",
   "maxLength": 64,
   "readOnly": true,
   "type": "string"
},
"dmv":  {
    "description": "Specification versions of the Resource and Device Specifications to which this
device data model is implemented",
      "maxLength": 256,
      "readOnly": true,
      "type": "string"
   },
   "n":  {
"$ref"  :  "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
   },
    "id": {
"$ref"  :  "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
   },
   "if":  {
      "description": "The OCF Interfacces supported by this Resource",
      "items": {
         "enum": [
         "oic.if.r",
         "oic.if.baseline"
     ],
     "type": "string",
     "maxLength": 64
   },
    "minItems": 2,
    "uniqueItems": true,
    "readOnly": true,
    "type": "array"
},
"econame" : {
   "description": "Ecosystem Name of the Bridged Device which is exposed by this VOD.",
   "type": "string",
   "enum": ["BLE", "oneM2M", "UPlus", "Zigbee", "Z-Wave"],
```

```
   "readOnly": true
},
"ecoversion" : {
    "description": "Version of ecosystem that a Bridged Device belongs to. Typical version string
format is like n.n (e.g. 5.0).",
     "type": "string",
     "maxLength": 64,
     "readOnly": true
  }
},
"type": "object",
"required": ["n", "di", "icv", "dmv", "piid"]
  }
 }
}
```

A.4.5 Property definition 属性定义

Table A.6 defines the Properties that are part of the "oic.wk.d" Resource Type.
表 A.6 定义部分属于"oic.wk.d"资源类型的属性。

Table A.6 - The Property definitions of the Resource with type "rt" = "oic.wk.d".
表A.6 - 类型为"rt" = "oic.wk.d"的资源的属性定义。

Property Name 属性名称	Value Type 数值类型	Mandatory 强制	Access mode 访问模式	Description 描述
rt	array: see schema 数组：参见模式	No 否	Read Only 仅可读	Resource Type of the Resource 资源的资源类型
1d	array: see schema 数组：参见模式	No 否	Read Only 仅可读	LocalizedDescriptions. 本地化描述
piid	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
di	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
dmno	string	No 否	Read Only 仅可读	Model number as designated by manufacturer. 型号由制造商指定。
sv	string	No 否	Read Only 仅可读	Software version. 软件版本
dmn	array: see schema 数组：参见模式	No 否	Read Only 仅可读	Manufacturer Name. 厂商名称
icv	string	Yes 是	Read Only 仅可读	The version of the Device 设备的版本
dmv	string	Yes 是	Read Only 仅可读	Specification versions of the Resource and Device Specifications to which this device data

				model is implemented 资源和设备的规范版本 实现此设备数据模型的规范
n	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
id	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
if	array: see schema 数组：参见模式	No 否	Read Only 仅可读	The OCF Interfaces supported by this Resource 该资源支持OCF接口
econame	string	No 否	Read Only 仅可读	Ecosystem Name of the Bridged Device which is exposed by this VOD. 被VOD公开的桥接设备的生态系统名称
ecoversion	string	No 否	Read Only 仅可读	Version of ecosystem that a Bridged Device belongs to. Typical version string format is like n.n (e.g. 5.0).一个桥接设备所属的生态系统的版本。典型的版本字符串格式是n.n(例如5.0)。

A.4.6 CRUDN behavior CRUDN行为

Table A.7 defines the CRUDN operations that are supported on the "oic.wk.d" Resource Type.
表A.7 定义支持在"oic.wk.d"资源类型的CRUDN操作

Table A.7 - The CRUDN operations of the Resource with type "rt" = "oic.wk.d".
表A.7 -类型为"rt" = "oic.wk.d"的资源的CRUDN操作。

Create 创建	Read 可读	Update 更新	Delete 删除	Notify通知
	get 得到			observe 观察

A.5 Introspection Resource 自省资源

A.5.1 Introduction 引言

This Resource provides the means to get the Introspection Device Data (IDD) specifying all the OCF Endpoints of the Device.
该资源提供了获取设备中的所有OCF端点的自省设备数据(IDD)的方法。

The url hosted by this Resource is either a local or an external url.
此资源承载的url可以是本地url，也可以是外部url。

A.5.2 Well-known URI 知名的URI

/IntrospectionResURI

A.5.3 Resource type 资源类型

The Resource Type is defined as: "oic.wk.introspection".
资源类型标题定义为："oic.wk.introspection".

A.5.4 OpenAPI 2.0 definition OpenAPI 2.0 定义


```
{
         "swagger": "2.0",
         "info": {
             "title": "Introspection Resource",
             "version": "2019-03-04",
             "license": {
                  "name": "OCF Data Model License",
                  "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
                  "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All
rights reserved."
      },
      "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
},
  "schemes": [
        "http"
],
"consumes": [
       "application/json"
],
"produces": [
       "application/json"
],
"paths": {
      "/IntrospectionResURI": {
           "get": {
                "description": "This Resource provides the means to get the Introspection Device
Data (IDD) specifying all the OCF Endpoints of the Device.\nThe url hosted by this Resource is
either a local or an external url.\n",
            "parameters": [
             {
                    "$ref": "#/parameters/interface"
             }
       ],
      "responses": {
           "200": {
              "description": "",
              "x-example": {
                 "rt": ["oic.wk.introspection"],
```

```
                  "urlInfo": [
                        {
                              "content-type": "application/cbor",
                              "protocol": "coap",
                              "url": "coap://[fe80::1]:1234/IntrospectionExampleURI"
                        }
                  ]
            },
             "schema": {
                  "$ref": "#/definitions/oic.wk.introspectionInfo"
                     }
                }
             }
         }
    }
},
"parameters": {
       "interface": {
             "in": "query",
            "name": "if",
            "type": "string",
            "enum": ["oic.if.r", "oic.if.baseline"]
      }
},
"definitions": {
        "oic.wk.introspectionInfo": {
               "properties": {
                    "rt":  {
                         "description": "Resource Type of the Resource",
                        "items": {
                              "enum": ["oic.wk.introspection"],
                              "type": "string",
                              "maxLength": 64
                     },
                     "minItems": 1,
                     "readOnly": true,
                     "uniqueItems": true,
                     "type": "array"
               },
                "n":  {
                    "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
             },
             "urlInfo":  {
                    "description": "Information on the location of the Introspection Device Data
(IDD).",
                    "items": {
                      "properties": {
                          "content-type": {
```

```
                                "default": "application/cbor",
                                "description": "content-type of the Introspection Device Data",
                                "enum": [
                                    "application/json",
                                    "application/cbor"
                                ],
                                "type": "string"
                            },
                            "protocol": {
                                   "description": "Identifier for the protocol to be used to obtain
the Introspection Device Data",
                                   "enum": [
                                          "coap",
                                          "coaps",
                                          "http",
                                          "https",
                                          "coap+tcp",
                                          "coaps+tcp"
                                    ],
                                    "type": "string"
                            },
                             "url": {
                                    "description": "The URL of the Introspection Device Data.",
                                    "format": "uri",
                                    "type": "string"
                             },
                             "version": {
                                    "default": 1,
                                    "description": "The version of the Introspection Device Data that
can be downloaded",
                                    "enum": [
                                         1
                                    ],
                                    "type": "integer"
                             }
                         },
                          "required": [
                               "url",
                               "protocol"
                          ],
                          "type": "object"
                    },
                     "minItems": 1,
                     "readOnly": true,
                     "type": "array"
              },
               "id":  {
"$ref": "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
       },
```

```
      "if": {
             "description": "The OCF Interfaces supported by this Resource",
            "items": {
                 "enum": [
                         "oic.if.r",
                         "oic.if.baseline"
                  ],
                  "type": "string",
                  "maxLength": 64
           },
            "minItems": 2,
            "readOnly": true,
            "uniqueItems": true,
            "type": "array"
         }
    },
     "type" : "object",
     "required": ["urlInfo"]
      }
    }
}
```

A.5.5 Property definition 属性定义

Table A.8 defines the Properties that are part of the "oic.wk.introspection" Resource Type.
表A.8 定义了部分"oic.wk.introspection"资源类型的属性。

Table A.8 - The Property definitions of the Resource with type"rt"="oic.wk.introspection".
表A.8 - 类型为"rt"="oic.wk.introspection"的资源的属性定义

Property Name 属性名称	Value Type 数值类型	Mandatory 强制	Access mode 访问模式	Description 描述
rt	array: see schema 数组:参见模式	No 否	Read Only 仅可读	Resource Type of the Resource 资源的资源类型
n	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
urlInfo	array: see schema 数组:参见模式	Yes 是	Read Only 仅可读	Information on the location of the Introspection Device Data (IDD). 有关自检设备数据(IDD)位置的信息。
id	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
if	array: see schema 数组:参见模式	No 否	Read Only 仅可读	The OCF Interfaces supported by this Resource 该资源支持OCF接口

A.5.6 CRUDN behavior CRUDN行为

Table A.9 defines the CRUDN operations that are supported on the "oic.wk.introspection" Resource Type.
表A.9定义了"oic.wk.introspection"资源类型支持的CRUDN操作。

Table A.9 - The CRUDN operations of the Resource with type "rt" = "oic.wk.introspection".
表A.9 - 类型为"rt" = "oic.wk.introspection"的资源的CRUDN操作。

Create 创建	Read 可读	Update 更新	Delete 删除	Notify通知
	get 得到			observe 观察

A.6 Platform 平台

A.6.1 Introduction 引言

Known Resource that is defines the Platform on which an Server is hosted. Allows for Platform specific information to be discovered.
已知资源定义服务器所在的平台。允许发现平台特定的信息。

A.6.2 Well-known URI 知名的URI

/oic/p

A.6.3 Resource type 资源类型

The Resource Type is defined as: "oic.wk.p".
资源类型定义为："oic.wk.p

A.6.4 OpenAPI 2.0 definition

```
{
          "swagger": "2.0",
         "info": {
              "title": "Platform",
              "version": "2019-03-04",
              "license": {
                  "name": "OCF Data Model License",
                  "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba
/LI CENSE.md",
                "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights
reserved."
         },
          "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
   },
    "schemes": ["http"],
    "consumes": ["application/json"],
    "produces": ["application/json"],
    "paths": {
         "/oic/p" : {
              "get": {
                   "description": "Known Resource that is defines the Platform on which an Server
is hosted.\nAllows for Platform specific information to be discovered.\n",

                 "parameters": [
                        {"$ref": "#/parameters/interface"}
                ],
                "responses": {
                       "200": {
                             "description" : "",
                             "x-example": {
                                 "pi": "54919CA5-4101-4AE4-595B-353C51AA983C",
                                 "rt": ["oic.wk.p"],
                                 "mnmn": "Acme, Inc"
                           },
                            "schema": { "$ref": "#/definitions/Platform" }
                      }
                }
           }
      }
   },
    "parameters": {
             "interface" : {
                  "in" : "query",
                  "name" : "if",
                  "type" : "string",
                "enum" : ["oic.if.r", "oic.if.baseline"]
```

```
            }
    },
    "definitions": {
          "Platform" : {
          "properties": {
               "rt" : {
                    "description": "Resource Type of the Resource",
                    "items": {
                       "enum":["oic.wk.p"],
                       "type": "string",
                       "maxLength": 64
               },
               "minItems": 1,
               "uniqueItems": true,
               "readOnly": true,
               "type": "array"
         },
          "pi" : {
                                                                                                    "pattern":
"^[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}$",
              "type": "string",
              "description": "Platform Identifier",
              "readOnly": true
    },
     "mnfv" : {
         "description": "Manufacturer's firmware version",
         "maxLength": 64,
         "readOnly": true,
         "type": "string"
    },
    "vid" : {
             "description": "Manufacturer's defined information for the Platform. The content is
freeform, with population rules up to the manufacturer",
         "maxLength": 64,
         "readOnly": true,
         "type": "string"
     },
      "mnmn" : {
                "description": "Manufacturer name",
                "maxLength": 64,
                "readOnly": true,
                "type": "string"
},
    "mnmo" : {
              "description": "Model number as designated by the manufacturer",
              "maxLength": 64,
              "readOnly": true,
              "type": "string"

},
    "mnhw" : {
```

```
            "description": "Platform Hardware Version",
            "maxLength": 64,
            "readOnly": true,
            "type": "string"
},
     "mnos" : {
                "description": "Platform Resident OS Version",
                "maxLength": 64,
                "readOnly": true,
                "type": "string"
},
       "mndt" : {
            "pattern": "^([0-9]{4})-(1[0-2]|0[1-9])-(3[0-1]|2[0-9]|1[0-9]|0[1-9])$",
            "type": "string",
            "description": "Manufacturing Date.",
            "readOnly": true
},
    "id" : {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
},
     "mnsl" : {
            "description": "Manufacturer's Support Information URL",
            "format": "uri",
            "maxLength": 256,
            "readOnly": true,
            "type": "string"
  },
    "mnpv" : {
        "description": "Platform Version",
        "maxLength": 64,
        "readOnly": true,
        "type": "string"
},
    "st" : {
         "description": "The date-time format pattern according to IETF RFC 3339.",
         "format": "date-time",
         "readOnly": true,
         "type": "string"
},
     "n" : {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
},
      "mnml" : {
            "description": "Manufacturer's URL",
            "format": "uri",
```

```
            "maxLength": 256,
            "readOnly": true,
            "type": "string"
     },
      "mnsel" : {
             "description": "Serial number as designated by the manufacturer",
             "maxLength": 64,
             "readOnly": true, "type": "string"
     },
     "if" : {
         "description": "The OCF Interfaces supported by this Resource",
          "items": {
               "enum": [
                     "oic.if.r",
                     "oic.if.baseline"
           ],
          "type": "string",
          "maxLength": 64
   },
       "minItems": 2,
       "readOnly": true,
       "uniqueItems": true,
       "type": "array"
  },
    "mnnct" : {
                 "description": "An array of integers and each integer indicates the network connectivity type based on IANAIfType value as defined by: https://www.iana.org/assignments/ianaiftype- mib/ianaiftype-mib, e.g., [71, 259] which represents Wi-Fi and Zigbee.",
               "items": {
                     "type": "integer",
                     "minimum": 1,
                      "description": "The network connectivity type based on IANAIfType value as defined by: https://www.iana.org/assignments/ianaiftype-mib/ianaiftype-mib."
           },
            "minItems": 1,
            "readOnly": true,
            "type": "array"
      }
},
 "type" : "object",
 "required": ["pi", "mnmn"]
     }
  }
}
```

A.6.5 Property definition 属性定义

Table A.10 defines the Properties that are part of the "oic.wk.p" Resource Type.
表A.10定义了"oic.wk.p"的一部分资源类型的属性。

Table A.10 - The Property definitions of the Resource with type "rt" = "oic.wk.p".
表A.10-类型为"rt" = "oic.wk.p"的资源的属性定义。

Property Name 属性名称	Value Type 数值类型	Mandatory 强制	Access mode 访问模式	Description 描述
rt	array: see schema 数组:参见模式	No 否	Read Only 仅可读	Resource Type of the Resource 资源的资源类型
pi	string	Yes 是	Read Only 仅可读	Platform Identifier 平台标识符
mnfv	string	No 否	Read Only 仅可读	Manufacturer's firmware version 制造商的固件版本。
vid	string	No 否	Read Only 仅可读	Manufacturer's defined information for the Platform. The content is freeform, with population rules up to the manufacturer 制造商为平台定义的信息。内容是自由形式的，由制造商决定规则。
mnmn	string	Yes 是	Read Only 仅可读	Manufacturer name 制造商名称
mnmo	string	No 否	Read Only 仅可读	Model number as designated by the manufacturer 由制造商指定的型号
mnhw	string	No 否	Read Only 仅可读	Platform Hardware Version 平台硬件版本
mnos	string	No 否	Read Only 仅可读	Platform Resident OS Version 平台驻留操作系统版本
mndt	string	No 否	Read Only 仅可读	Manufacturing Date. 制造日期
id	array: see schema 数组:参见模式	No 否	Read Write 读写	
mnsl	string	No 否	Read Only 仅可读	Manufacturer's Support Information URL 制造商的支持信息URL
mnpv	string	No 否	Read Only 仅可读	Platform Version 平台版本
st	string	No 否	Read Only 仅可读	The date-time format pattern according to IETF RFC 3339. 符合IETF RFC 3339规定的日期-时间格式模式。
n	array: see schema 数组:参见模式	No 否	Read Write 读写	

mnml	string	No 否	Read Only 仅可读	Manufacturer's URL 制造商的URL
mnsel	string	No 否	Read Only 仅可读	Serial number as designated by the manufacturer 制造商指定的序列号
if	array: see schema 数组:参见模式	No 否	Read Only 仅可读	The OCF Interfaces supported by this Resource 该资源支持OCF接口
mnnct	array: see schema 数组:参见模式	No 否	Read Only 仅可读	An array of integers and each integer indicates the network connectivity type based on IANAIfType value as defined by: https://www.iana.org/assignments/ianaiftype- mib/ianaiftype-mib, e.g., [71, 259] which represents Wi-Fi and Zigbee. 一个整数数组，每个整数表示基于IANAIfType值的网络连接类型，定义参照: https://www.iana.org/assignments/ianaiftype- mib/ianaiftype-mib, 例如， [71, 259] 代表Wi-Fi和 Zigbee.

A.6.6 CRUDN behavior CRUDN行为

Table A.11 defines the CRUDN operations that are supported on the "oic.wk.p" Resource Type.
表A.11定义了支持在"oic.wk.p"资源类型的CRUDN操作。

Table A.11 - The CRUDN operations of the Resource with type "rt" = "oic.wk.p".
表A.11 - 类型为"rt" = "oic.wk.p"的资源的CRUDN操作。

Create 创建	Read 可读	Update 更新	Delete 删除	Notify通知
	get 得到			observe 观察

A.7 Discoverable Resources 可发现资源

A.7.1 Introduction 引言

Baseline representation of /oic/res; list of discoverable Resources
/oic/res的基线表示；可发现资源列表

A.7.2 Well-known URI 知名的URI

/oic/res

A.7.3 Resource type 资源类型

The Resource Type is defined as: "oic.wk.res".
资源类型定义为："oic.wk.res"。

A.7.4 OpenAPI 2.0 definition

```
{
     "swagger": "2.0",
     "info": {
        "title": "Discoverable Resources",
        "version": "2019-04-22",
        "license": {
            "name": "OCF Data Model License",
            "url": "https://openconnectivityfoundation.github.io/core/LICENSE.md",
               "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights
reserved."
          },
          "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
        },
        "schemes": [
           "http"
        ],
          "consumes": [
             "application/json"
        ],
        "produces": [
            "application/json"
        ],
         "paths": {
            "/oic/res?if=oic.if.ll": {
               "get": {
                     "description": "Links list representation of /oic/res; list of discoverable
Resources\n",
                  "parameters": [
                      {
                         "$ref": "#/parameters/interface-all"
                      }
               ],
                "responses": {
                   "200": {
                      "description" : "",
                      "x-example": [
                         {
                            "href": "/oic/res",
                            "rt":     ["oic.wk.res"],
                            "if":     ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
                            "rel":  ["self"],
                            "p":      {"bm": 3},
```

```
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1122"}   ]
              },
              {
                  "href": "/humidity",
                  "rt":["oic.r.humidity"],
                  "if":["oic.if.s", "oic.if.baseline"],
                  "p":    {"bm": 3},
                  "eps": [
                     {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
                     {"ep": "coaps://[fe80::b1d6]:1122"},
                     {"ep": "coaps+tcp://[2001:db8:a::123]:2222", "pri": 3}
                  ]
               },
               {
                  "href": "/temperature",
                  "rt":["oic.r.temperature"],
                  "if":["oic.if.s", "oic.if.baseline"],
                  "p":    {"bm": 3},
                  "eps": [
                      {"ep": "coaps://[[2001:db8:a::123]:2222"}
                    ]
                 }
            ],
             "schema": {
                "$ref": "#/definitions/slinklist"
             }
          }
       }
    }
},
 "/oic/res?if=oic.if.b" : {
 "get": {
   "description": "Batch representation of /oic/res; list of discoverable Resources\n",
   "parameters": [
      {"$ref": "#/parameters/interface-all"}
   ],
   "responses": {
     "200": {
        "description" : "",
        "x-example": [
          {
             "href": "/humidity",
             "rep":{
                "rt":     ["oic.r.humidity"],
                "humidity": 40,
                "desiredHumidity": 40
          }
       },
   {
```

```
          "href": "/temperature",
          "rep":{
          "rt":["oic.r.temperature"],
          "temperature": 20.0,
          "units": "C"
        }
     }
  ],
   "schema": { "$ref": "#/definitions/sbatch" }
     }
   }
 }
},
"/oic/res?if=oic.if.baseline": {
   "get": {
   "description": "Baseline representation of /oic/res; list of discoverable Resources\n",
   "parameters": [
 {
     "$ref": "#/parameters/interface-all"
  }
],
 "responses": {
    "200": {
      "description": "",
      "x-example": [
        {
            "rt": ["oic.wk.res"],
            "if": ["oic.if.ll", "oic.if.b", "oic.if.baseline"],
            "links": [
               {
                  "href": "/humidity",
                  "rt":["oic.r.humidity"],
                  "if":["oic.if.s", "oic.if.baseline"],
                  "p":{"bm": 3},
                  "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2},
                      {"ep": "coaps://[fe80::b1d6]:1122"},
                      {"ep": "coap+tcp://[2001:db8:a::123]:2222", "pri": 3}
                   ]

                },
                {
                   "href": "/temperature",
                   "rt":["oic.r.temperature"],
                   "if":["oic.if.s", "oic.if.baseline"],
                   "p":       {"bm": 3},
                   "eps": [
                      {"ep": "coaps://[[2001:db8:a::123]:2222"}
                    ]
            }
         ]
```

```
      }
   ],
    "schema": {
        "$ref": "#/definitions/sbaseline"
       }
     }
   }
  }
 }
},

 "parameters": {
   "interface-all": {
       "in": "query",
       "name": "if",
       "type": "string",
       "enum": ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
     }
  },

   "definitions": {
      "oic.oic-link": {
          "type": "object",
          "properties": {
          "anchor": {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/anchor"
           },
           "di": {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/di"
            },
             "eps": {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/eps"
            },
             "href": {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
            },
             "if": {
                 "description": "The OCF Interfaces supported by the Linked Resource",
                 "items": {
                    "enum": [ "oic.if.baseline",
                    "oic.if.ll",
```

```
                    "oic.if.b",
                    "oic.if.rw",
                    "oic.if.r",
                    "oic.if.a",
                    "oic.if.s"
                ],
                 "type": "string",
                 "maxLength": 64
              },
              "minItems": 1,
              "uniqueItems": true,
              "type": "array"
            },
            "ins": {
                                                                                    "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/ins"
              },
              "p": {
                                                                                    "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/p"
              },
              "rel": {
                    "description": "The relation of the target URI referenced by the Link to the
context URI",
                "oneOf": [
                  {
                                                                                    "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_array"
                  },
                  {
                                                                                    "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/rel_string"
                  }
                ]
            },
            "rt": {
                "description": "Resource Type of the Linked Resource",
                "items": {
                "maxLength": 64,
                "type": "string"
            },
               "minItems": 1,
               "uniqueItems": true,
               "type": "array"
          },
          "title": {
```

```
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/title"
          },

          "type": {

                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/type"

         },

          "tag-pos-desc": {
              "$ref":

"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/tag-pos-desc"

         },

           "tag-pos-rel": {
               "$ref":

"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/tag-pos-rel"

         },

             "tag-func-desc": {
                  "$ref":

"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/tag-func-desc"
          }
      },
      "required": [
         "href",
         "rt",
         "if"
       ]
    },

      "slinklist": {
            "type" : "array",
            "readOnly": true,
            "items": {
               "$ref": "#/definitions/oic.oic-link"
         }
    },
     "sbaseline": {
            "type": "array",
            "minItems": 1,
```

```
            "maxItems": 1,
            "items": {
                "type": "object",
                "properties": {
                    "n": {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/n"
                    },
                    "id": {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
schema.json#/definitions/id"
                },
                 "rt": {
                    "description": "Resource Type of this Resource",
                    "items": {
                    "enum": ["oic.wk.res"],
                    "type": "string",
                    "maxLength": 64
                },
                 "minItems": 1,
                 "readOnly": true,
                 "uniqueItems": true,
                 "type": "array"
            },
             "if": {
               "description": "The OCF Interfaces supported by this Resource",
               "items": {
                    "enum": [
                       "oic.if.ll",
                       "oic.if.b",
                       "oic.if.baseline"
                    ],

                     "type": "string",
                     "maxLength": 64
                 },

                  "minItems": 2,
                  "readOnly": true,
                  "uniqueItems": true,
                  "type": "array"
              },
             "links":  {
                "type": "array",
                "items": {
                      "$ref": "#/definitions/oic.oic-link"
              }
        }
    },
```

```
     "required": [
        "rt",
        "if",
        "links"
     ]
   }
},
"sbatch"  :  {
   "type" : "array",
   "minItems":1,
   "items"  :  {
   "type": "object",
   "additionalProperties": true,
   "properties": {
       "href": {
                                                                                "$ref":
"https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
schema.json#/definitions/href"
      },
       "rep": {
          "oneOf": [
              {
               "description": "The response payload from a single Resource",
               "type": "object"
            },
            {
                "description": " The response payload from a Collection (batch) Resource",
                "items": {
                     "$ref": "#/definitions/oic.oic-link"
             },
             "type": "array"
          }
      ]
   }
},
"required": [
"href",
"rep"
    ]
   }
  }
 }
}
```

A.7.5 Property definition 属性定义

Table A.12 defines the Properties that are part of the "None" Resource Type.

表A.12定义了属于“None”资源类型的属性。

Table A.12 - The Property definitions of the Resource with type "rt" = "None".

表A.12 - 类型为"rt" = "None"的资源的属性定义。

Property Name 属性名称	Value Type 数值类型	Mandatory 强制	Access mode 访问模式	Description 描述
anchor	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
di	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
eps	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
href	multiple types: see schema 多种类型:参见模式	Yes 是	Read Write 可读写	
if	arrays: see schema 数组:参见模式	Yes 是	Read Write 可读写	The OCF Interface set supported by this Resource 该资源支持OCF接口集
ins	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
p	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
rel	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	The relation of the target URI referenced by theLink to the context URI 由链接引用的目标URI与上下文URI的关系
rt	arrays: see schema 数组:参见模式	Yes 是	Read Write 可读写	Resource Type of the Resource 资源的资源类型
title	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
type	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
tag-pos-desc	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
tag-pos-rel	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
tag-func-desc	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	

n	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
id	multiple types: see schema 多种类型:参见模式	No 否	Read Write 可读写	
rt	arrays: see schema 数组:参见模式	Yes 是	Read Only 仅可读	Resource Type of the Resource 资源的资源类型
if	arrays: see schema 数组:参见模式	Yes 是	Read Only 仅可读	The OCF Interface set supported by this Resource 该资源支持OCF接口集
links	arrays: see schema 数组:参见模式	Yes 是	Read Write 可读写	

A.7.6 CRUDN behavior CRUDN行为

Table A.13 defines the CRUDN operations that are supported on the "None" Resource Type.

表A.13定义了"None"资源类型支持的CRUDN操作。

Table A.13 - The CRUDN operations of the Resource with type "rt" = "None".

表A.13 - 类型为"rt" = "None"的资源类型的CRUDN操作。

Create 创建	**Read 可读**	**Update 更新**	**Delete 删除**	**Notify通知**
	get 得到			observe 观察

Annex B(informative)OpenAPI 2.0 Schema Extension

附件B(资料性附录)

OpenAPI 2.0模式扩展

B.1 OpenAPI 2.0 Schema Reference OpenAPI 2.0模式参考

OpenAPI 2.0 does not support allOf and anyOf JSON schema valiation constructs; this document has extended the underlying OpenAPI 2.0 schema to enable these, all OpenAPI 2.0 files are valid against the extended schema. Reference the following location for a copy of the extended schema:
OpenAPI2.0不支持allOf和anyOf JSON模式的valiation结构。为了支持这些结构，本文档扩展了底层的OpenAPI 2.0模式，所有的OpenAPI 2.0文件对扩展模式都是有效的。扩展模式的副本请参考以下链接：
https://github.com/openconnectivityfoundation/OCFswagger2.0-schema

B.2 OpenAPI 2.0 Introspection empty file OpenAPI 2.0自省空文件

Reference the following location for a copy of an empty OpenAPI 2.0 file:
一个空的OpenAPI 2.0文件的副本请参考以下链接：

https://github.com/openconnectivityfoundation/DeviceBuilder/blob/master/introspection-examples/introspection-empty.txt

Annex C(normative)Semantic Tag enumeration support

附录C(规范性附录)

语义标签列举的支持

C.1 Introduction 引言

This Annex defines the enumerations that are applicable to defined Semantic Tags.
本附件定义了适用于定义的语义标签的列举。

C.2 "tag-pos-desc" supported enumeration "tag-pos-desc"支持的列举

Figure C.1 defines the enumeration from which a value populated within an instance of the "tag-pos-desc" Semantic Tag is taken.
图C.1定义了列举，从该列举中获取“tag- pos-desc”语义标记实例中填充的值。

```
"pos-descriptions": {
"enum": ["unknown","top","bottom","left","right","centre","topleft","bottomleft","centreleft"
,"centreright","bottomright","topright","topcentre","bottomcentre"]
}
```

Figure C.1 - Enumeration for "tag-pos-desc" Semantic Tag

图C.1 -列举"tag-pos-desc"语义标签

Figure C.2 provides an illustrative representation of the definition of the values that can be represented within an instance of "tag-pos-desc".
图C.2提供了可以在“tag-pos-desc”实例中表示的值的定义的说明性表示。

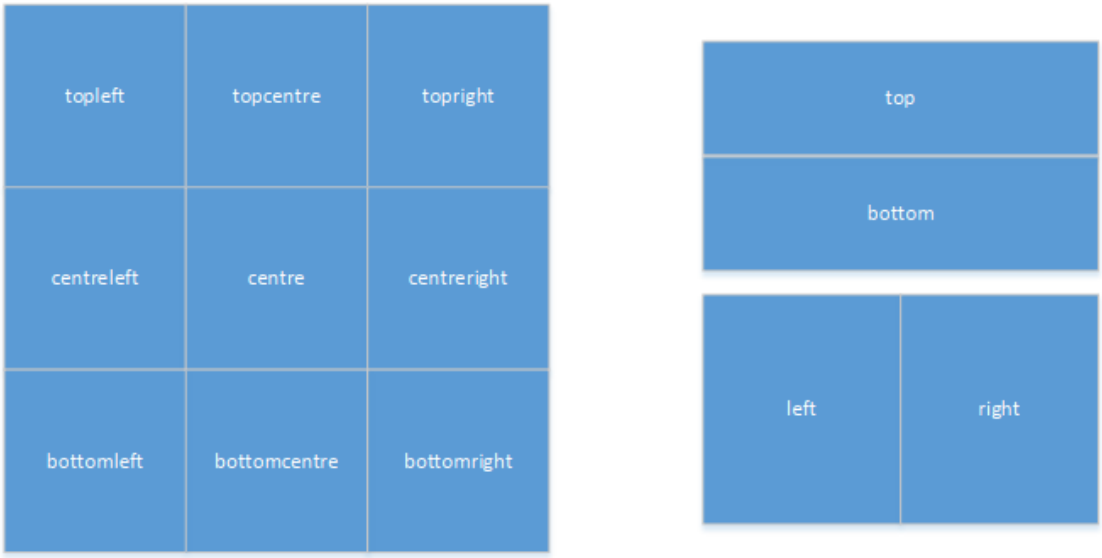

Figure C.2 - Definition of "tag-pos-desc" Semantic Tag values

图C.2 -“Tag -pos-desc”语义标签值的定义

Bibliography

参考文献

[1] OCF Core - Optional, Information technology - Open Connectivity Foundation (OCF) Specification - Part X: Core - Optional specification Latest version available at:https://openconnectivity.org/specs/OCF_Core_Optional_Specification.pdf

[1]OCF核心——可选的,信息技术——开放互联基金会(OCF)规范——第X部分:核心-可选规范，最新版本可访问 https://openconnectivity.org/specs/OCF_Core_Optional_Specification.pdf

[2] OCF Easy Wi-Fi Setup, Information technology - Open Connectivity Foundation (OCF) Specification - Part 7: Wi-Fi Easy Setup specificationLatest version available at:https://openconnectivity.org/specs/OCF_Wi- Fi_Easy_Setup_Specification.pdf

[2]OCF简单的无线设置,信息技术-开放互联基金会(OCF)规范第7部分:无线网基本设置规范，最新版本可访问 https://openconnectivity.org/specs/OCF_Wi- Fi_Easy_Setup_Specification.pdf