Introduction of OIC standard

January, 2016

Standard Working Group
Open Interconnect Consortium

Open Interconnect Consortium, Inc.
Table of Contents

• Internet of Things Standard Consideration
• Introduction of Open Interconnect Consortium
  - Overview
  - Core Framework
  - Smart Home Profile
  - Security
  - Remote Access
Definition of various Things

• By defining resources of things and its properties

  - **Resources** - properties
    - **BinarySwitch**
      - true(on), false(off)
    - **Dimming**
      - dimmingSetting (int)
      - step (int)
      - range [0-100]
    - **Brightness**
      - brightness (int)

  - (no Verbs) + Objects
    *Fixed set of verbs (CRUD operations) from transport layer will be used

• By defining functions/operations of things

  - **SetSwitch**
    - Power(in)
  - **SetDimmingLevel**
    - step (in), range (in)
  - **SetBrightness**
    - brightness (in)

  - (Verbs + Objects)
    - RPC model

  - **e.g., Light bulb**

  - Resource model in RESTful Architecture
    (e.g., W3C, CSEP, etc.)
Support of Constrained Things

• Less overhead/ Less Traffic
  • Minimize CPU Load, Memory impacts, Traffic and Bandwidth
    - Compact header
    - Binary protocol
    - Compressed encoding of payload

• Low Complexity
  - Simple Resource Model
    > Short URI (Late Binding w/ resource type defined)
    > Broad and Shallow Hierarchy

*RAM <10KB, Flash <100KB (RFC 7228)
Support of Multiple Verticals

- Legacy vertical services usually designed as silos ➔ No common way to communicate among them

- A common platform provides a foundation for vertical services to collaborate and interwork by providing common services and data models
Interoperability

- **Full interoperability** from the connectivity layer up to the service layer is the only way to truly guarantee a satisfactory UX.
- Interoperability at the Connectivity and/or Platform layer only provides partial interoperability which can ultimately lead to fragmentation.

[Diagram showing levels of interoperability: Connectivity, Platform, Service]
Interoperability & Certification

- Conformance test - Each device proves conformance to specifications
- Interoperability test - Each device proves interoperability with other devices

- Certification Scope

Open Source

- Mandatory (in spec, cert & committed in Open Source Project)
- Tested Optional Spec Features
- Optional Open Source Features

Specification

- Tested Optional Open Source Features
- Optional Spec Features
- Optional Open Source Features

Device under Test

Prerequisites: Dependency Certification (e.g. Connectivity)

Conformance Test

Interoperability Test

Certificate Issue & Logo Licensing

CERTIFIED
Licensing

• For IPR Policy: RAND-Z > RAND >> no IPR policy
• For Open Source: Apache 2.0 > ISC

• Due to the common nature of IoT connecting everything over the Internet, it’s most critical for manufacturers to avoid a licensing risk
  - Everything connected could be at potential risk

• Offering manufacturer-friendly Licensing and IPR Policy enables growth of market by attracting both start-ups and large enterprises; such an IPR policy must be clear and readily understandable ensuring that the terms are offered by all IP holders.
Introduction of
Open Interconnect Consortium
Introduction to OIC - Optimized for IoT

- RESTful Architecture
- Common Platform
- Certification Program
- CoAP for Constrained Devices
- Full Stack Interop. Test
OIC Key Concepts (1/2)

- **Free IPR License** (Code: Apache 2.0 & Spec: RAND-Z)
  - License covers both code, standards and related IPR
  - License applies to members and affiliates of members

- **Dedicated and optimized protocols for IoT** (e.g. CoAP)
  - Specific considerations for constrained devices
  - Fully compliant towards RESTful architecture
  - Built-in discovery and subscription mechanisms

- **Standards and Open Source to allow flexibility creating solutions**
  - Able to address all types of devices, form-factors, companies and markets with the widest possibility of options
  - Open Source is just one implementation to solve a problem
OIC Key Concepts (2/2)

• **Full stack definition for maximum interoperability**
  - Connectivity, Platform and Vertical Services defined
  - License applies to members and affiliates of members

• **Certification and Logo program**
  - Guarantees all devices work together
  - Consistent user awareness for interoperability
Sample of Current Members

Diamond
- ARRIS
- CableLabs
- Cisco
- GE Software
- Intel
- Samsung

Platinum
- Atmel
- AJOX
- CACI
- Dell
- Honeywell
- IBM
- MediaTek
- ZTE

Gold
- accenture
- Acer
- AEPONA
- Allion
- alticast
- Astoria Networks
- Asurion
- Asus
- Aventis Wireless
- Axstone
- b&O
- BigJungle
- Bontom
- Buffalo
- Cadence
- ESL
- Cryptosoft
- Connecting Your Things
- D+M Group
- DigiOn
- D-Link
- Oppo
- DSP Group
- DTT
- DSC
- ECS
- Encored
- ETRI
- FIH
-富士康
- FIM Mobile Limited
- Fine Point Technologies
- IB firstbuild
- fuse
- GCR
- GoPro
- GRL
- Granite River Labs
- GrowSafe
- HANSUNG
- Hisense
- Hitron
Sample of Current Members

Gold (continued)
Sample of Current Members

Non-profit

![Logos of Non-profit Members]

Liaisons

![Logos of Liaison Members]
OIC Structure

OIC

Board of Directors

Standard
Specification & Certification

Open Source

Membership

Technology Planning

Ecosystem

Marketing Communications

IoTivity
Open Source Project

Steering Group

Sponsored (funded) by OIC
Develops reference implementation of OIC standard

http://www.iotivity.org
OIC Specification Overview

Core Framework Specification
**Specification Structure**

**Infrastructure**
- Core Framework
- Security
- Remote Access
- Certification Test Plans and Test Cases

**Resource Model**
- Resource Specification (Domain agnostic)

**Per Vertical Domain**
- Device Specification
- Domain Specific Resource Specification
Core Framework Specification

Overview
Objectives

• **Core Framework Specification Scope**
  • Specifies the technical specification(s) comprising of the core architectural framework, messaging, interfaces and protocols based on approved use-case scenarios
  • Enables the development of vertical profiles (e.g. Smart Home) on top of the core
• Architect a core framework that is scalable from resource constrained devices to resource rich devices
• Evaluate technical specification(s) for maximum testability and interoperability
• Ensure alignment with OIC open source releases
OIC Roles

- **OIC Client**
  - i) Initiate a transaction (send a request) & ii) access an OIC Server to get a service

- **OIC Server**
  - i) host OIC Resource & ii) send a response & provide service
OIC Architecture

- OIC adopted RESTful Architecture
- Current OIC Architecture defines 2 logical roles that devices can take
  - OIC Server: A logical entity that exposes hosted resources
  - OIC Client: A logical entity that accesses resources on an OIC Server
Organization of an OIC Device

• OIC Device concept

Physical Device e.g. light bulb

OIC Device 1

/oic/p
/oic/res
/oic/d
/oic/mnt

OIC Device 2

/oic/p
/oic/res
/oic/d
/oic/prs

Resource URI: /oic/p
rt: oic.wk.p
if: oic.if.r
n: homePlatform
policy: bm:11
pi: at1908
mnmn: Samsung

Mandatory
Optional
**Device example: light device (oic.d.light)**

- **Example overview**
  - Smart light device with i) binary switch & ii) brightness resource

- **Device type: Light device (oic.d.light) [Defined by the domain]**

- **Associated resources**
  - Core resources: ① oic/res, ② oic/d
  - Device specific resources: ③ Binary switch (oic.r.switch.binary),
  - Other optional resources can be exposed, in this example ④ Brightness resource (oic.r.light.brightness)

### Example: Smart light device with 4 resources

<table>
<thead>
<tr>
<th>Device Title</th>
<th>Device Type</th>
<th>Associated Resource Type</th>
<th>M/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>oic.d.light</td>
<td>oic/res (oic.wk.core)</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>oic/d (oic.d.light)</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Binary switch (oic.r.switch.binary)</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brightness (oic.r.light.brightness)</td>
<td>O</td>
</tr>
</tbody>
</table>
Core Framework Specification

Key Features
OIC Spec Features - Core Framework Spec

1. **Discovery:** Common method for device discovery (IETF CoRE)
2. **Messaging:** Constrained device support as default (IETF CoAP) as well as protocol translation via intermediaries
3. **Common Resource Model:** Real world entities defined as data models (resources)
4. **CRUDN:** Simple Request/Response mechanism with Create, Retrieve, Update, Delete and Notify commands
5. **Device Management:** Network connection settings and remote monitoring/reset/reboot functions
6. **ID & Addressing:** OIC IDs and addressing for OIC entities (Devices, Clients, Servers, Resources)
7. **Security:** Basic security for network, access control based on resources, key management etc
**OIC Core Framework Basic Operation**

### Discovery
- Discover access policies, device info and resources on the devices

### Operation
- Get device information by retrieving resources
- Control devices by changing resources
- Observe change on the properties of resources

### Basic common services
- Device Monitoring
- Maintenance (e.g., reboot, factory reset, statistics collection, etc.)
Protocol Stack

Application
Resource Model
Encoding (CBOR)
CoAP
DTLS
UDP
IPv6
L2 Connectivity (Wi-Fi)

Alternatives

<table>
<thead>
<tr>
<th>Encoding</th>
<th>JSON or XML/EXI can be negotiated</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Version</td>
<td>v6 (v4 supported for legacy devices)</td>
</tr>
</tbody>
</table>

Project BOIC Stack
End point Discovery (CoAP Discovery)

- OIC devices make use of CoAP Discovery (defined by IETF RFC 7252)
  - Resource Discovery (Possible to discovery resource being hosted by device directly)
  - Low processing overhead on each node
  - High traffic efficiency (in terms of amount of data sent/received for discovery)
**Encoding Schemes - JSON, XML/EXI, CBOR**

- OIC resource is represented as a sequence of bits by encoding schemes when to transfer it over the network.
- OIC supports several encoding schemes and it will be negotiated and accepted by OIC Server when OIC Client requests.
- OIC has mandated CBOR as the default encoding scheme.

<table>
<thead>
<tr>
<th></th>
<th>JSON</th>
<th>XML/EXI</th>
<th>CBOR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>- Lightweight, text-based, language-independent data interchange format</td>
<td>- Binary compression standard for XML</td>
<td>- Concise binary object representation based on JSON data model</td>
</tr>
<tr>
<td><strong>Standard</strong></td>
<td>IETF RFC 7159</td>
<td>W3C Efficient XML Interchange Format 1.0</td>
<td>IETF RFC 7049</td>
</tr>
<tr>
<td><strong>Content Type</strong></td>
<td>/application/json</td>
<td>/application/exi</td>
<td>/application/cbor</td>
</tr>
<tr>
<td><strong>OIC M/O</strong></td>
<td>Optional</td>
<td>Optional</td>
<td>Mandatory</td>
</tr>
</tbody>
</table>

*JSON: JavaScript Object Notation, EXI: Efficient XML Interchange, CBOR: Concise Binary Object Representation*
Collection Resources

• A container is used to model complex structures
• An OIC Resource that contains one or more references (specified as OIC Links) to other OIC Resources is an OIC Collection
• An OIC Link embraces and extends typed “web links” as specified in RFC 5988
• Offloads handling of discovery (response to multicast messages) to devices that are capable of doing so
• Key enabler for sleepy end nodes, enhances battery life.

Device B acts as Resource Directory for Device A and Device D; Device A and D do not respond to multicast query
Scenes/ Rules/ Scripts (1 of 3)

• Overview
  • Mechanisms for automating certain operations
  • Rules, Scripts and Scenes can be grouped and reused

• Scenes
  • A static entity that stores a set of defined resource property values for a collection of resources.
  • Provide a mechanism to store a setting over multiple OIC Resources that may be hosted by multiple separate OIC Servers.
  • Once set up, can be used by multiple OIC Clients to recall a setup
• Rules
  • A logical “if then” statement
  • Consists of a rule condition and a Rule Member (a script)
  • The rule condition is an evaluation criterion which can include evaluation of the value of a sensor on an OIC Server.
  • When the evaluation criterion is evaluated true then the Rule Members are set to a specific determined value.
  • A rule condition is evaluated when one of the observed resources in the rule condition changes.
• Scripts
  • A programmatic element that can be used to incorporate conditionals, delays, loops and other programmatic devices, including reading and writing scenes
  • Scripts can consist of a set of steps that are executed either upon meeting the conditions of a rule or as part of another script, in order to automate tasks
  • Scripts can also be used to set a scene to a specific value
  • A Script is realized as the set of Rule Members that are executed when a rule condition holds true

• Summary
  • Scenes are bundled user settings
  • Scripts are automated background tasks
  • Rules are conditional statements that execute scripts when the condition is true
Block Transfer with CoAP Messaging

- Basic CoAP messages work well for the small payloads we expect from light-weight, constrained IoT devices.
- It is envisioned whereby an application will need to transfer larger payloads.
- CoAP block wise transfer as defined in IETF draft-ietf-core-block-17 shall be used by all OIC Servers that receive a retrieve request for a content payload that would exceed the size of a CoAP datagram.
Messaging Protocol Negotiation

• Supported messaging protocols are conveyed in the property (mpro) on the /oic/res (resource discovery)

• Omitting this property defaults to the messaging protocol as specified in the vertical specification (e.g., CoAP for Smart Home)

• After discovery, an OIC Client can use any of the supported messaging protocols supported by the OIC Server
CoAP Serialization over TCP

- Provides the ability for CoAP to run over TCP in environments where TCP is already available and where UDP may be blocked.
- If TCP is used then reliability is provided by TCP rather than the inherent reliability mechanisms within CoAP (confirmable messages).
- Use the new protocol negotiation feature to convey support during resource discovery (/oic/res)
OIC Specification Overview

Smart Home Device and Resource Specification
Smart Home Device and Resource Specification

Way of Working
Defining OIC Components (on top of CORE)

OIC Servers
- Defined by device identifier: standardized name of the device
- List of mandatory OIC resources per device
- Note that OIC Clients are implicitly specified as “opposite” side of an OIC Server.
  - Currently OIC does not impose interaction sequences.
  - All Resources are allowed to talk to/from any OIC Client at any point in time

OIC Resource
- Defined by resource identifier: standardized name of the resource
- List of mandatory properties per resource
- List of allowed actions (read/readwrite/..) per resource
Vendor extensions

Vendor is allowed to:

• Create own defined (none OIC standardized) resources
• Create own defined (none OIC standardized) device types
• Extend existing devices with additional (not mandated) resources
  • With standardized resource types
  • With vendor defined resource types
Tooling

- SHTG defines all resource schemas using JSON, all resource APIs using RAML
- SHTG developed Python based tool chain that auto-generates specification text based on the RAML and JSON that is defined per resource.
- Capabilities provided by the tooling include:
  - Auto validation of the RAML against RAML syntax rules
  - Auto validation of the JSON schemas against JSON Draft-04 rules
  - Auto validation of all example JSON against the applicable JSON schemas

  High confidence level in the validity of the resource definitions
  Ability to simulate all resources
Specifications

• Specifications are split in 2 documents:
  • Device specification
  • Resource specification

The Device specification uses the resources defined in the resource specification
Device Specification

- Contains profiles of
  - Core specification
  - Security specification

- Contains list of smart home devices
- Each Smart home device definition contains:
  - Unique identifier (rt)
  - A list of mandatory resources

<table>
<thead>
<tr>
<th>OIC SmartHome Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor Smart Home Extensions</td>
</tr>
<tr>
<td>Vendor Core Resources Extensions</td>
</tr>
<tr>
<td>Smart Home Device specification</td>
</tr>
<tr>
<td>Smart Home Resources</td>
</tr>
<tr>
<td>Core Resources</td>
</tr>
<tr>
<td>Smart Home Core Profiles</td>
</tr>
</tbody>
</table>
Smart Home Device and Resource Specification

Key Features
Resource Specification

• List of reusable resources that are used in a Smart Home Device
  • Contains generic list of error codes
  • Uses core definitions

• Each Smart home resource definition contains:
  • unique identifier (rt)
  • Indication if the resource is a sensor or actuator
  • List supported methods
  • List per method the JSON schema for input and output

• Resources are specified in RESTful API Modelling Language (RAML)
**Smart Home Use Cases**

• Selected key enabling use cases to scope activity

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor Environment Control</td>
<td>1</td>
</tr>
<tr>
<td>Lighting control</td>
<td>1</td>
</tr>
<tr>
<td>Energy Saving Washer/Dryer</td>
<td>2</td>
</tr>
<tr>
<td>Energy Management</td>
<td>3</td>
</tr>
<tr>
<td>Remote Access for Device Control</td>
<td>4</td>
</tr>
<tr>
<td>Smart watch notify and control</td>
<td>6</td>
</tr>
<tr>
<td>Smart Video Environment</td>
<td>3</td>
</tr>
<tr>
<td>Smart Home Office</td>
<td>3</td>
</tr>
<tr>
<td>Smart Garage</td>
<td>3</td>
</tr>
<tr>
<td>Device Grouping and Control</td>
<td>3</td>
</tr>
<tr>
<td>Multi player gaming</td>
<td>7</td>
</tr>
<tr>
<td>Smart watch gaming on TV</td>
<td>7</td>
</tr>
<tr>
<td>Fire safety monitor and Notify</td>
<td>4</td>
</tr>
<tr>
<td>Keyless Entry</td>
<td>2</td>
</tr>
<tr>
<td>Home Security</td>
<td>2</td>
</tr>
<tr>
<td>Health Monitor and Notify</td>
<td>5</td>
</tr>
</tbody>
</table>

① Control proximal OIC Devices
② On board new Devices
③ Control remotely with an OIC Client
Indoor Environment Control

LAN Network (Home)  Home GW  WAN Network (Cloud)

Windows  A/C  Temperature  Humidity  Smart device  Smart device
Lighting Control

LAN Network (Home)

Home GW

WAN Network (Cloud)

Smartphone

Lighting

Smartphone

Lighting

Lighting

Lighting
Energy-saving washer/dryer

LAN Network (Home)

WAN Network (Cloud)

Home GW

Smart device

Washer

Dryer
Energy Management

Need to see my home energy use information
Whole-home energy use information
How much energy I consumed today?
Today, you consumed 20% less than usual
Remote Access Device Control

"Can I drop in your home, tonight?"

"Absolutely..."

'1 believe you, robot'

[ready for girls' visit : fresh and clean]
Keyless Entry

LAN Network (Home)

WAN Network (Cloud)

Home GW

Door locks

Door lock

Smartphone

Smartphone
Health Monitor & Notify

LAN Network (Home) → Home GW → Smartphone

WAN Network (Cloud)
## Smart Home Device Type

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Minimum Resource Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Conditioner</td>
<td>Binary Switch, Temperature</td>
</tr>
<tr>
<td>Air Purifier</td>
<td>Binary Switch</td>
</tr>
<tr>
<td>Blind</td>
<td>Open Level</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>Binary Switch, Mode</td>
</tr>
<tr>
<td>Door</td>
<td>Open Level</td>
</tr>
<tr>
<td>Clothes Dryer</td>
<td>Binary Switch, Mode</td>
</tr>
<tr>
<td>Clothes Washer</td>
<td>Binary Switch, Mode</td>
</tr>
<tr>
<td>Fan</td>
<td>Binary Switch</td>
</tr>
<tr>
<td>Garage Door</td>
<td>Door</td>
</tr>
<tr>
<td>Light</td>
<td>Binary Switch</td>
</tr>
<tr>
<td>Oven</td>
<td>Binary Switch, Temperature (2)</td>
</tr>
<tr>
<td>Printer</td>
<td>Binary Switch, Operational State</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Minimum Resource Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refrigerator</td>
<td>Binary Switch, Refrigeration, Temperature (2)</td>
</tr>
<tr>
<td>Robot Cleaner</td>
<td>Binary Switch, Mode</td>
</tr>
<tr>
<td>Smart Plug</td>
<td>Binary Switch</td>
</tr>
<tr>
<td>Switch</td>
<td>Binary Switch</td>
</tr>
<tr>
<td>Thermostat</td>
<td>Temperature (2)</td>
</tr>
<tr>
<td>Camera</td>
<td>Media</td>
</tr>
<tr>
<td>Generic Sensor</td>
<td>Sensor</td>
</tr>
<tr>
<td>Receiver</td>
<td>Binary Switch, Audio, Media Source List (2)</td>
</tr>
<tr>
<td>Scanner</td>
<td>Binary Switch, Operational State, Automatic Document Feeder</td>
</tr>
<tr>
<td>Security Panel</td>
<td>Mode</td>
</tr>
<tr>
<td>Television</td>
<td>Binary Switch, Audio, Media Source List</td>
</tr>
<tr>
<td>Water Valve</td>
<td>Open Level</td>
</tr>
</tbody>
</table>

Exposure of an OIC Device Type is Mandatory. If an OIC Server hosts an OIC known device then it shall follow all normative requirements in the Device Specification applicable to that Device.
## Defined Resource Types (1/2)

<table>
<thead>
<tr>
<th>Resource Types</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Flow</td>
<td>Indoor Environment Control</td>
</tr>
<tr>
<td>Air Flow Control</td>
<td></td>
</tr>
<tr>
<td>Battery</td>
<td>Device Control</td>
</tr>
<tr>
<td>Binary switch</td>
<td>Device Control</td>
</tr>
<tr>
<td>Brightness</td>
<td></td>
</tr>
<tr>
<td>Colour Chroma</td>
<td></td>
</tr>
<tr>
<td>Colour RGB</td>
<td></td>
</tr>
<tr>
<td>Dimming</td>
<td></td>
</tr>
<tr>
<td>Door</td>
<td>Indoor Environment Control</td>
</tr>
<tr>
<td>Energy Consumption</td>
<td></td>
</tr>
<tr>
<td>Energy Usage</td>
<td>Energy Management</td>
</tr>
<tr>
<td>Humidity</td>
<td>Indoor Environment Control</td>
</tr>
<tr>
<td>Icemaker</td>
<td>Device Control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource Types</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lock</td>
<td>Keyless Entry</td>
</tr>
<tr>
<td>Lock Code</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>Device Control</td>
</tr>
<tr>
<td>Open Level</td>
<td></td>
</tr>
<tr>
<td>Operational State</td>
<td></td>
</tr>
<tr>
<td>Ramp Time</td>
<td>Lighting Control</td>
</tr>
<tr>
<td>Refrigeration</td>
<td>Device Control</td>
</tr>
<tr>
<td>Temperature</td>
<td>Indoor Environment Control</td>
</tr>
<tr>
<td>Time Period</td>
<td>Device Control</td>
</tr>
</tbody>
</table>
### Defined Resource Types (2/2)

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>TV, Home Entertainment</td>
</tr>
<tr>
<td>Auto Focus</td>
<td>IP Camera</td>
</tr>
<tr>
<td>Auto White Balance</td>
<td>IP Camera</td>
</tr>
<tr>
<td>Automatic Document Feeder</td>
<td>Scanner Support</td>
</tr>
<tr>
<td>Button</td>
<td>Device Control</td>
</tr>
<tr>
<td>Colour Saturation</td>
<td>IP Camera</td>
</tr>
<tr>
<td>DRLC</td>
<td>Smart Energy</td>
</tr>
<tr>
<td>Energy Overload</td>
<td>Smart Energy</td>
</tr>
<tr>
<td>Media</td>
<td>IP Camera</td>
</tr>
<tr>
<td>Media Source List</td>
<td>TV, Home Entertainment</td>
</tr>
<tr>
<td>Movement (Linear)</td>
<td>Robot Cleaner</td>
</tr>
<tr>
<td>Night Mode</td>
<td>IP Camera</td>
</tr>
<tr>
<td>PTZ</td>
<td>IP Camera</td>
</tr>
<tr>
<td>Signal Strength</td>
<td>Proximity</td>
</tr>
</tbody>
</table>

### Sensor Support Resources

<table>
<thead>
<tr>
<th>Sensor Resource Type</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration</td>
<td>Extended Sensor Set</td>
</tr>
<tr>
<td>Activity Count</td>
<td></td>
</tr>
<tr>
<td>Atmospheric Pressure</td>
<td></td>
</tr>
<tr>
<td>Carbon Dioxide</td>
<td></td>
</tr>
<tr>
<td>Carbon Monoxide</td>
<td></td>
</tr>
<tr>
<td>Contact</td>
<td></td>
</tr>
<tr>
<td>Glass Break</td>
<td></td>
</tr>
<tr>
<td>Heart Rate Zone</td>
<td></td>
</tr>
<tr>
<td>Illuminance</td>
<td></td>
</tr>
<tr>
<td>Magnetic Field Direction</td>
<td></td>
</tr>
<tr>
<td>Presence</td>
<td></td>
</tr>
<tr>
<td>Radiation (UV)</td>
<td></td>
</tr>
<tr>
<td>Sleep</td>
<td></td>
</tr>
<tr>
<td>Smoke</td>
<td></td>
</tr>
<tr>
<td>Smoke</td>
<td></td>
</tr>
<tr>
<td>Three Axis</td>
<td></td>
</tr>
<tr>
<td>Touch</td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td></td>
</tr>
</tbody>
</table>

Resource Types are Conditionally Mandatory. If an OIC Server hosts an OIC known resource then it shall follow all normative requirements in the Resource Specification applicable to that Resource.
OIC Bridge - Background & technical need

• There are many different IoT standards out there
• There are many different vendor solutions out there
Hence it would be good for OIC if OIC could use these devices and create a (vendor defined) bridge to these non-OIC devices.

Goal:
• To represent non OIC devices by means of a bridge as an OIC server on the network.

Conceptual:
• Bridge establishes an OIC standardized north bridge so that all OIC clients can use the bridged devices.
• The south bridge will be vendor/implementation specific: it uses the protocol defined by the bridged device.

(for example: it needs to realize Philips Hue APIs if a Hue light is bridged)
An OIC smart home bridging device is a device that represents one or more other non-OIC devices as OIC Smart Home Devices on the network. The represented devices themselves are out of the scope of OIC. The bridging (that is, how the bridge communicates with the non-OIC devices) is implementation and vendor specific. The only difference between a ‘regular’ OIC Device and a bridged device is that the latter is encapsulated in an OIC Smart Home Bridge Device. An OIC Smart Home Bridge Device shall be indicated on the network with an “rt” of “oic.d.bridge”. When such a device is discovered the exposed resources on the OIC Smart Home Bridge Device describe the devices that are being bridged.
**Bridge Device example: bridge (oic.d.bridge)**

**OIC bridge device**
- `baseURI: 100.0.0.1:5683`
- `oic/res`
- `oic/d`

**OIC light device**
- `baseURI: 100.0.0.1:5683/0`
- `oic/res`
- `oic/d (oic.d.light)`
- Binary switch

**OIC fan device**
- `baseURI: 100.0.0.1:5683/1`
- `oic/res`
- `oic/d (oic.d.fan)`
- Binary switch
Bridging relationship with oic/res

```
/oic/res

[ {
  "di": "bridge_device_id",
  "links": [
    { "href": "/oic/d",
      "rt": "oic.d.bridge",
      "if": "oic.if.r",
      "rel": "hosts"}],
   { "di": "light_device_id",
     "links": [
      { "href": "0/oic/d",
        "rt": "oic.d.light",
        "if": "oic.if.r",
        "rel": "contains external"},
      { "href": "1/myLightSwitch",
        "rt": "oic.r.switch.binary",
        "if": "oic.if.a",
        "rel": "contains external"}]}
],
```

```
/oic/d

{ "n": "myRoomBridgeDevice",
  "rt": "oic.d.bridge",
  "if": "oic.if.r",
  "di": "bridge_device_id",
  "icv": "oic.1.5",
}

```

```
/oic/d

{ "n": "myRoomLightDevice",
  "rt": "oic.d.light",
  "if": "oic.if.a",
  "di": "light_device_id",
  "icv": "oic.1.5",
}

```

```
/oic/d

{ "n": "myRoomFanDevice",
  "rt": "oic.d.fan",
  "if": "oic.if.r",
  "di": "fan_device_id",
  "icv": "oic.1.5",
}

```

```
/oic/d

{ "n": "myRoomFanSwitch",
  "rt": "oic.r.switch.binary",
  "if": "oic.if.a",
  "di": "fan_device_id",
  "icv": "oic.1.5",
}

```

65
```
OIC Security Summary

• OIC key management supports end-to-end device protection
• Resource layer ACLs allow intended interactions while preventing unintended interactions
• Secure device ownership helps prevent attacks when devices are added to the network
To Cross a Boundary We Must Define the Endpoint

An OIC device is the endpoint

...more specifically it is the OIC resource layer

OIC resources define how device capabilities are exposed to other OIC devices

Resources are accessed securely through a secure channel such as DTLS

- End-to-end message encryption, integrity and replay protection

OIC does not define endpoint hardening techniques

- Resource layer hardening is implied
Secure Resource Manager (SRM)

OIC Device

<table>
<thead>
<tr>
<th>OIC Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Introspection (RI) layer</td>
</tr>
<tr>
<td>Secure Resource Manager (SRM) Layer</td>
</tr>
<tr>
<td>Resource Manager (RM)</td>
</tr>
<tr>
<td>Policy Engine (PE)</td>
</tr>
<tr>
<td>Persistent Storage Interface (PSI)</td>
</tr>
<tr>
<td>Connectivity Abstraction (CA) layer</td>
</tr>
</tbody>
</table>

Secure Virtual Resource database

- SRM Duties
  - Manage secure endpoint resources (Creds, ACLs, Device ID, Config status)
  - Enforce resource access and endpoint protection
  - Device ownership
  - Security provisioning
  - SVRD storage protection
Ownership Transfer and Bootstrapping

- Devices typically ship from a manufacturer in an “un-owned” state.
- The user does some magic to affect taking ownership of the device, using an Onboarding Tool (OBT).
  Take over responsibility of the device and relieve manufacturer of any liability due any actions the device may take under user’s ownership.
- Ownership transfer creates a relationship between an OIC device and an OBT.
  The relationship is defined through establishment of an Ownership Credential and a set of ownership-complete states.
Ownership Transfer and Bootstrapping

- Security Spec Defines Several Ownership Transfer Methods (OTM):
  - Just-Works, DECAP, Random-PIN, Manufacturer Certificates
  - Also allows Vendor Specific Method
- All OTMs are optional for an OIC device to implement, but it is mandatory to support at least one among Just-Works, DECAP, Random-PIN or Manufacturer Certificates.
  - (We will need to be able to test all for certification ultimately)
  - Might change in the future spec
- OTMs differ in:
  - How a device establishes trust
  - How the physical owner’s “intent” is proved
  - What cipher suites are used
- OTMs should bring the device to a well defined state
Secured vs. Un-secured

- OIC Servers support a secured and un-secured interface.
- Generally speaking, the un-secured interface is for discovery only. All other services should be visible on the secured interface only.
  - The un-secured interface has no message protection and no access control enforcement
  - Publicly visible unique IDs (device, platform, etc.) may present a privacy problem
- Discoverable resources are resources that can be delivered as part of a discovery request (secured interface or not)
  - At the time of creating, a resource is defined as “discoverable” or not.
Message Integrity and Confidentiality

• DTLS only for now.
• The devices communicating need to have useable credentials to talk to each other. If they are missing, the devices could contact the CMS to get them.
• All secured communication is encrypted and signed.
Access Control

- Resources on the secured interface (that should be almost everything) are **only** accessible if there is a proper entry in the Access Control List
  - No ACL, No Service
- An ACL says "X can do Y on resource Z"
  - X can be a deviceId, a role, or a group (in the future)
  - Y can be any combination of CRUDN
- If no ACL is present, and the device has an AMS configured, it can ask the AMS what authorization X has on Z

1/13/2016
Access Control: example

/oic/sec/acl

- **Subject**: device/group or role
- **Resource(s)**: one or more URN
- **Permission**: bitmap of CRUDN
- **Period(s)**: validity periods
- **Recurrence(s)**: recurrence rule(s)
- **Rowner**: the service that owns this acl

```
{
  "Subject": "switch1",
  "Resource": "/light",
  "Permission": "00000100", <update>
  "Period": "",
  "Recurrence": "",
  "Rowner": "oic.sec.ams"
}
```
Resource Access Example

- Access is blocked if no ACL match is found
- Device1 request to get /oic/d is **accepted** due to ACL Read permission
- Device2 request to update /oic/light/1 is **denied** due to time-of-day policy
- An intermediary (Device4) may also enforce ACLs
Credential Management

- OIC devices can support the use of both symmetrical and asymmetrical credentials for establishing secure communication
  - Symmetric Key is mandatory
  - Local PKI mechanism is supported (Keys are issued in home domain and used only within that domain.)
- Missing credentials could be procured from a CMS
- Credentials may have an expiration period
  - Expired credentials can be refreshed
Credential Management: example

/oic/sec/cred

- **CredID**: Local short ID
- **SubjectID**: device or group
- **RoleID(s)**: roles this credential allows a subject to assert
- **CredType**: sym/asym/cert/…
- **PublicData, PrivateData, OptionalData**
- **Period**: Expiration period of credential
- **Credential Refresh Method**:
- **Rowner**: service that can modify this resource

```json
{
    "CredID": "1",
    "SubjectID": "device1",
    "RoleID": "",
    "CredType": "1", <symmetric pair-wise>
    "PublicData": "",
    "PrivateData": "ABCDEFGHIJKLMNP",
    "Period": "P1W",
    "Recurrence": "",
    "Rowner": "oic.sec.ams"
}
```
OIC Specification Overview

Remote Access
Remote Access ("RA") in OIC (implementation plan)

- Remote Access endpoint Devices:
  - Remote Access Endpoints ("RAE"):
    - OIC Servers also capable of XMPP, optionally capable of ICE-client
  - Remote Access Proxies ("RA-Proxy"):
    - Superset of RAE – Capable of ‘representing’ “RA-constrained devices”
      - “RA-Constrained”: Devices incapable of natively supporting RA tech

- Cloud Components:
  - XMPP Server(s)
The OIC RA Model

Non-OIC (RA-Constrained) device
RA-Constrained OIC Device
"RAE"
"RA-Proxy"
CoAP
XMPP-native

Realm I

A
B
C
D
E

K
L
M
N
P

Realm II

F
G
H
Q
R
S

XMPP Server 1
XMPP Server 2

1/13/2016
Remote Access

- **Server Components:**
  - Device Management Server: Device/Capability Registration and Authorization
  - Signaling Server: Delivering candidate address to recipient, discovery, presence, low BW data, SDP control

- **Client Components:** RA Endpoint (RAE) & RA-Proxy
  - XMPP Client
RA as defined in Spec 1.0

• Format for bare-J IDs (owner) and full-J IDs for RAEs
  • Includes J ID-Resource overloading for:
    • OIC Spec version
    • Device-type
    • UUID

• Mapping from Core/Smart-Home Resources to full-J ID format
  • Allows for Presence, Remote Discovery, XMPP-Roster-based access

• Communication of CRUDN messages between the OIC clients and OIC servers that are in the same roster
RA-Roadmap - Post Spec 1.0 priorities

• Defining RA-Proxy functionality
  • Leverage XMPP PubSub (**XEP-0060**)
  • Extending full-JID overloading model & XMPP Presence
    • Adding RA-Proxy Device-type – avoid gratuitous remote device queries

• “App notes” for temporary remote access via XMPP Multi-User Chat (**MUC – XEP-0045**),
  • Family members, neighbors, etc.

• Adding Jingle (**XEP-0166**) for media signaling
Thank you!

• Access the OIC specifications
  http://openinterconnect.org/developer-resources/specs/

• Contact OIC at admin@openinterconnect.org