oneM2M

Open Standard Enables Interoperability in IOT

By Suhas V

under

Instructor: Deepak Gangadharan

Defining Interoperability

ISO definition:

- The capability to communicate, execute programs, or transfer data
- Requires the user to have little or no knowledge of unique characteristics

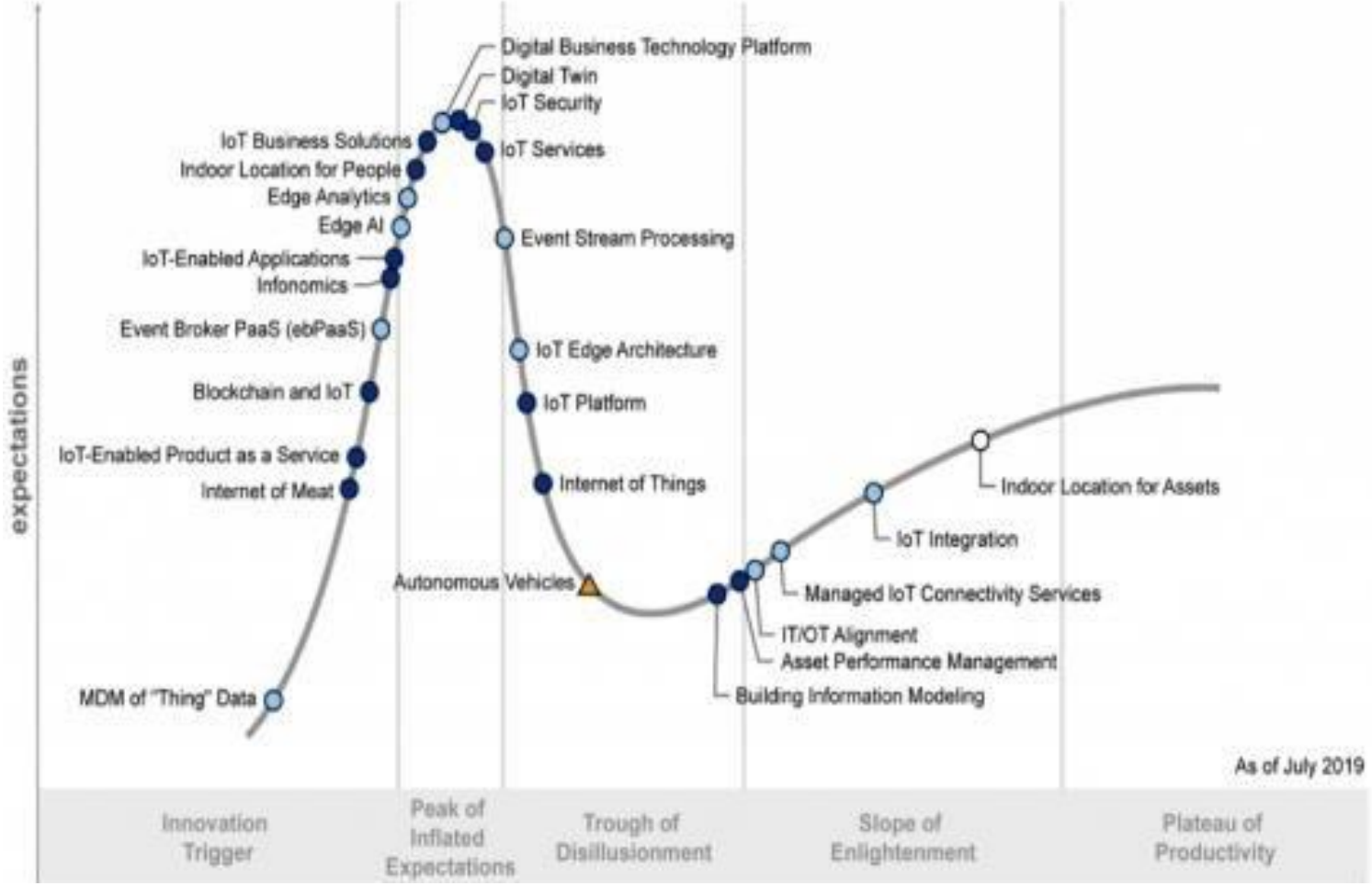
IEEE definition:

- The ability of two or more systems or components to exchange the information and use the information that has been exchanged

IoT interoperability:

- Interoperability in IoT is the compatibility of multiple devices to communicate with each other irrespective of deployed software and hardware.

Gartner's Hype Cycle-Current IoT State



Need for Interoperability

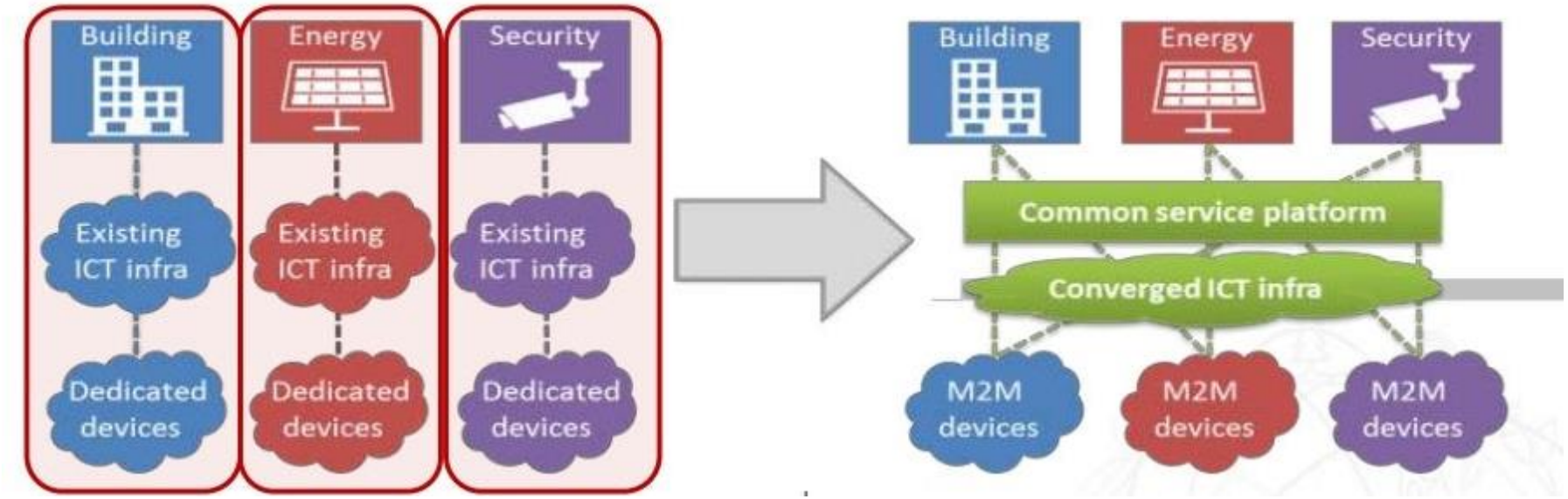
Nearly 40 percent of economic impact requires interoperability between IoT systems

Potential economic impact of IoT[1]

$11.1 trillion

Value potential requiring interoperability ($ trillion)		% of total value	Examples of how interoperability enhances value
Factories	1.3	36	Data from different types of equipment used to improve line efficiency
Cities	0.7	43	Video, cellphone data, and vehicle sensors to monitor traffic and optimize flow
Retail environments	0.7	57	Payment and item detection system linked for automatic checkout
Work sites	0.5	56	Linking worker and machinery location data to avoid accidents, exposure to chemicals
Vehicles	0.4	44	Equipment usage data for insurance underwriting, maintenance, pre-sales analytics
Agriculture	0.3	20	Multiple sensor systems used to improve farm management
Outside	0.3	29	Connected navigation between vehicles and between vehicles and GPS/traffic control
Home	0.1	17	Linking chore automation to security and energy system to time usage
Offices	0[2]	30	Data from different building systems and other buildings used to improve security

IoT Cross Domain Interoperability



Highly fragmented market with limited vendor specific applications

Reinventing the wheel: same services developed again and again

Each technology contains its own functionalities without interoperability

End to end integration and common service functionalities

Interoperability at the level of communication and data

Seamless interaction between heterogeneous devices and applications

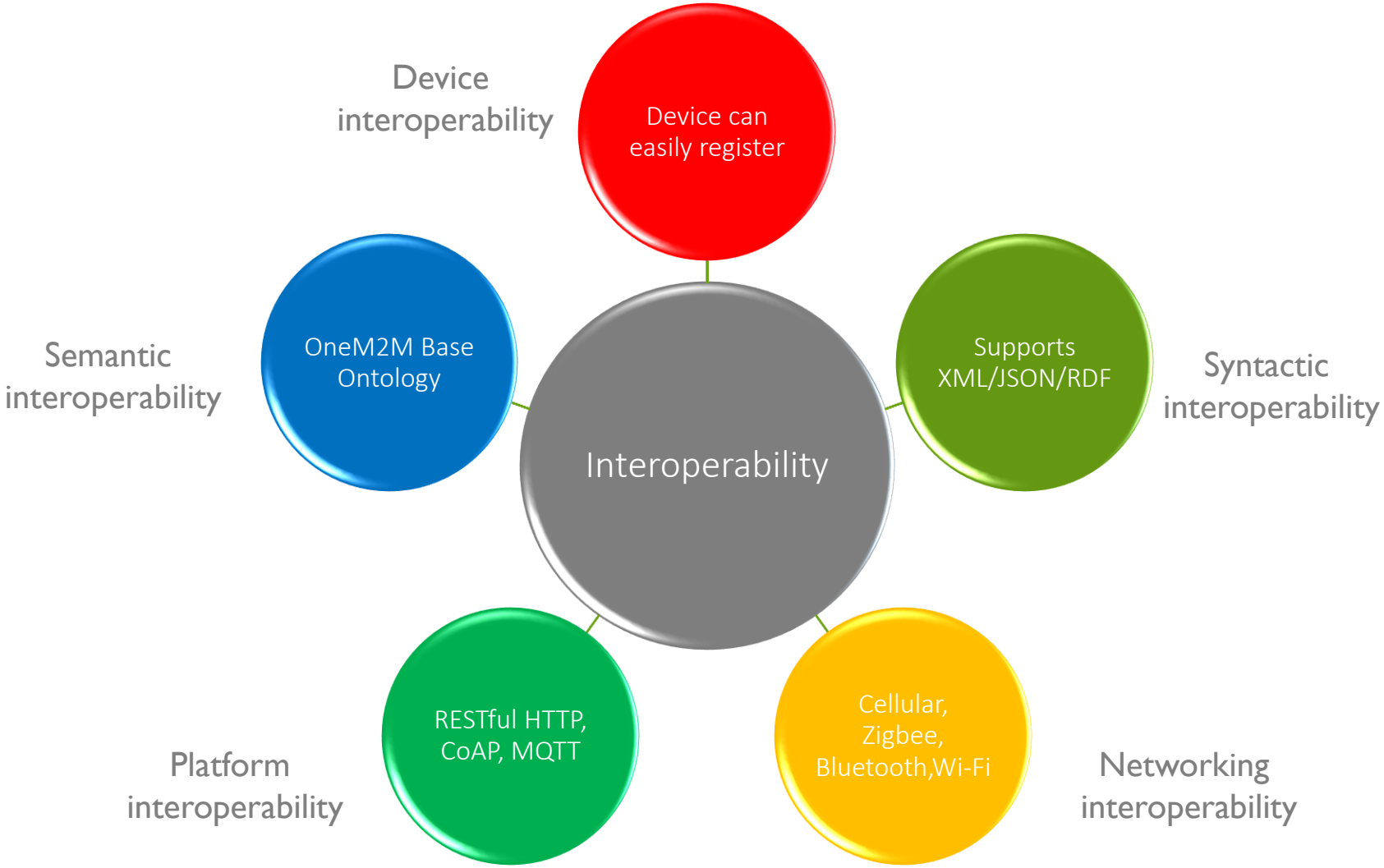
Components of Interoperability

Source: A Comprehensive Survey on Interoperability for IIoT: Taxonomy, Standards, and Future Directions

oneM2M Standard for Interoperable IoT

- A global initiative to develop IoT standards to enable interoperable, secure, and simple-to-deploy services for the IoT ecosystem.
- Allow any IoT application to discover and interact with any IoT device.
- IoT solutions can interoperate across different technologies
- Reduce fragmentation, increase reusability and improve the cost base through economies of scale

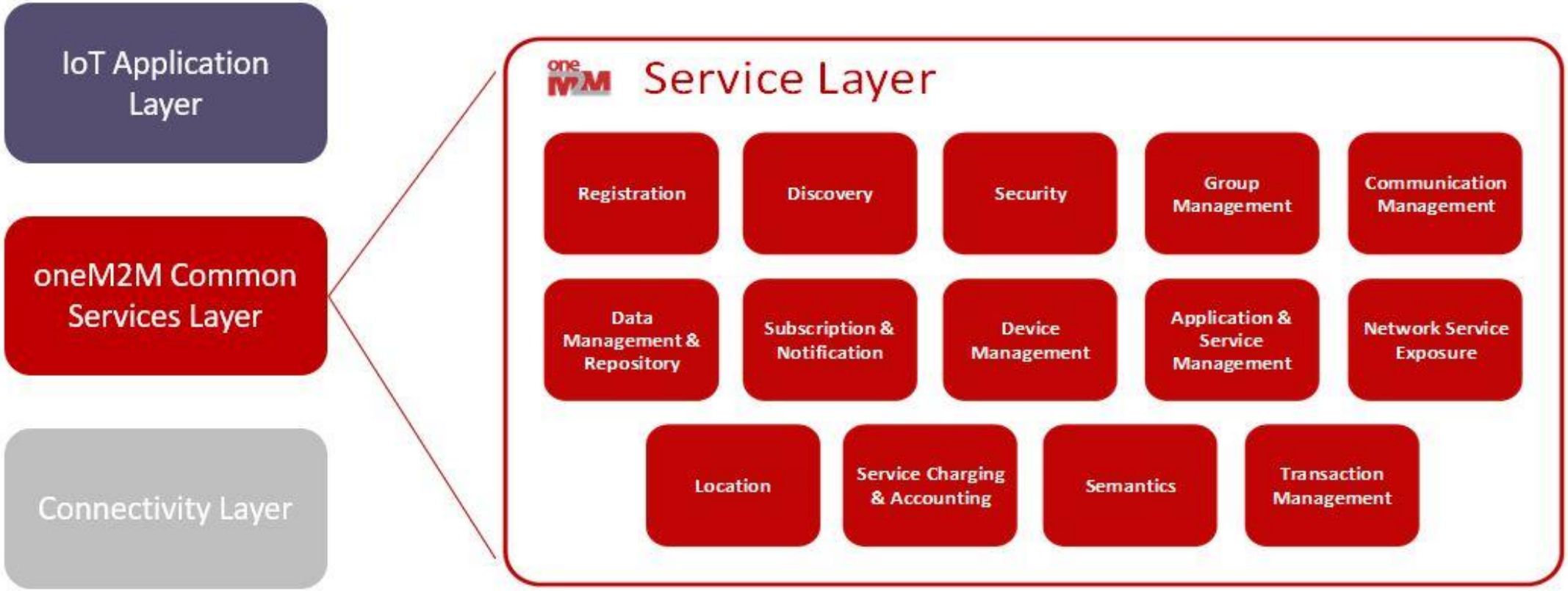
oneM2M and Interoperability



Frequently heard terms in oneM2M Standard

1. Common Service Functions
2. Resources
3. Resource Tree

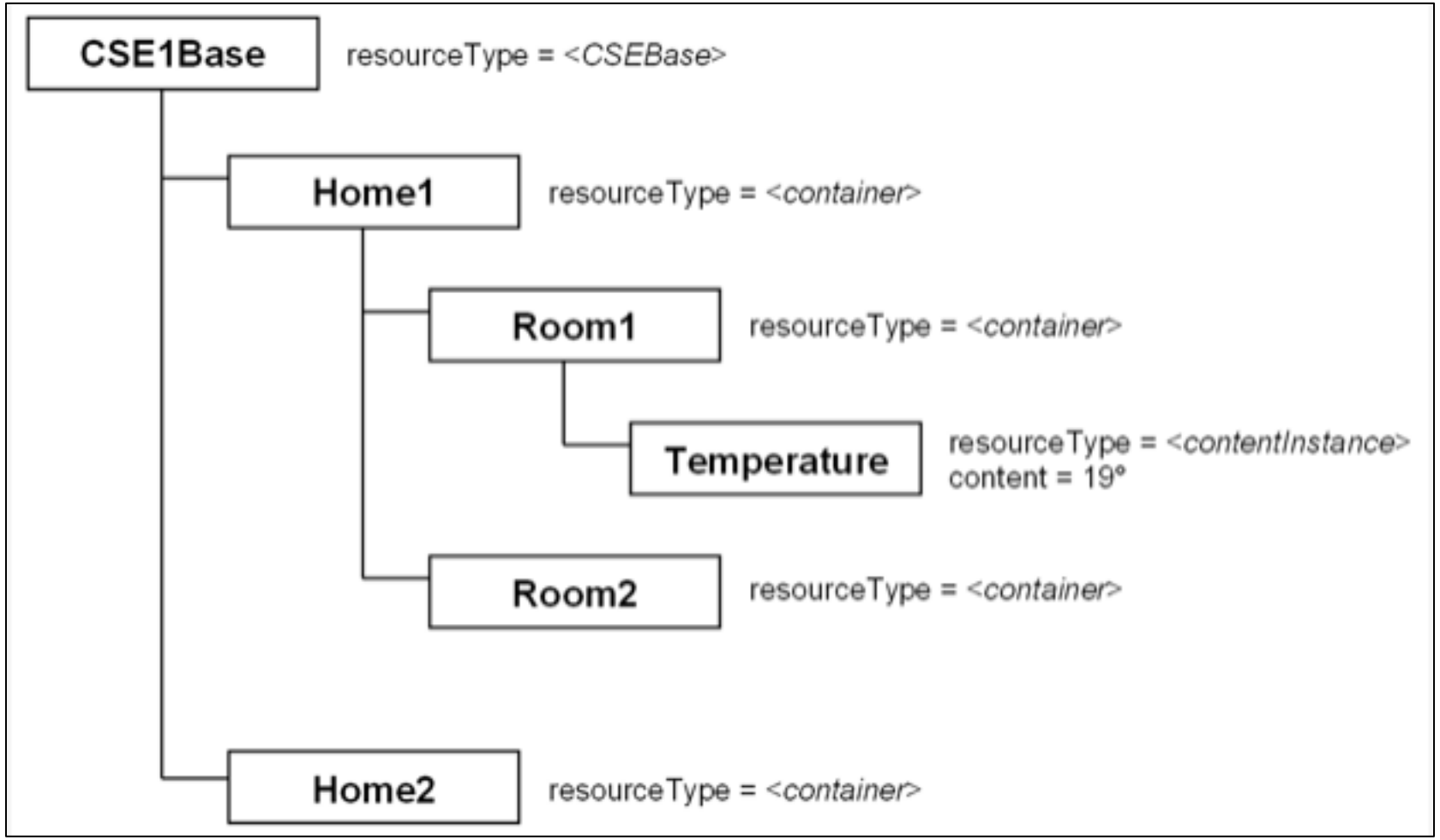
Common Service Functions



Resources

1. Access Control Policy – ACP
2. Application Entity - AE
3. Container - CNT
4. Content Instance – CIN
5. CSE Base – CB
6. Group Resource – GRP
7. Subscription Resource - SUB

Resource Tree



OM2M

Building an Interoperable IoT System

Eclipse-OM2M

- An open-source implementation of oneM2M standards by LAAS-CNRS
- A horizontal M2M service platform
- A horizontal Service Common Entity (CSE) that can be deployed in an M2M server, a gateway, or a device.

Setting up OM2M

1. Setup up Java- jdk 1.8.XXX version (Mandatory)
   a) Open command-prompt/terminal
   b) Enter java-version
   c) The version should be jdk-1.8.XXX (E.g.: open jdk1.8.251)

2. Download OM2M from https://wiki.eclipse.org/OM2M/Download
3. Extract the contents

Note:

Installing Java:

a) Windows:
   i. https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html#license-lightbox
   ii. Set JAVA_HOME - https://javatutorial.net/set-java-home-windows-10

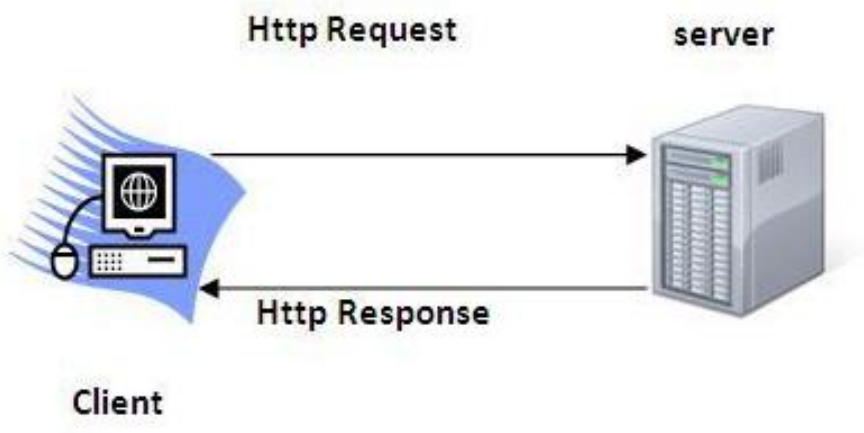
a) Ubuntu:
   i. https://computingforgeeks.com/how-to-install-java-8-on-ubuntu/
   ii. sudo update-alternatives --config java
   iii. (https://www.xmodulo.com/change-default-java-version-linux.html)

How do we use OM2M

1. (Platform Interoperability)Protocol: HTTP/HTTPS based data exchange
2. (Syntactic Interoperability) Format: JSON data format
3. (Device Interoperability)Device: PC/Laptop/ESP
4. (Network Interoperability): Wi-Fi

HTTP Architecture

- It is the protocol that allows web servers and browsers to exchange data over the web.
- Uses Client Server Architecture



HTTP Requests

HTTP Request	Description
GET	Asks to get the resource at the requested URL.
POST	Asks the server to accept the body info attached. It is like GET request with extra info sent with the request.
DELETE	Says to delete the resource at the requested URL.
TRACE	Asks for the loopback of the request message, for testing or troubleshooting.
PUT	Says to put the enclosed info (the body) at the requested URL.
HEAD	Asks for only the header part of whatever a GET would return. Just like GET but with no body.
OPTIONS	Asks for a list of the HTTP methods to which the thing at the request URL can respond

HTTP Response Codes

1. 200- OK
2. 201-Created
3. 400 Bad Request
4. 404-Not Found
5. 403-Forbidden
6. 500-Internal Server Error

source: https://en.wikipedia.org/wiki/List_of_HTTP_status_codes

JSON Format

- JavaScript Object Notation (JSON) is **a standard text-based format for representing structured data**

```
{
   "text":"i have excess cabin altitude what is procedure",
   "intent":"get_document",
   "entities":[
      {
         "start":7,
         "end":28,
         "value":"excess cab alt",
         "entity":"system"
      }
   ]
},
```

Restful API and REST Client

- A RESTful API is **an architectural style for an application program interface (API) that uses HTTP requests to access and use data**
- REST Client is **a method or a tool to invoke a REST service API**

POSTMAN

To store data in OM2M we need

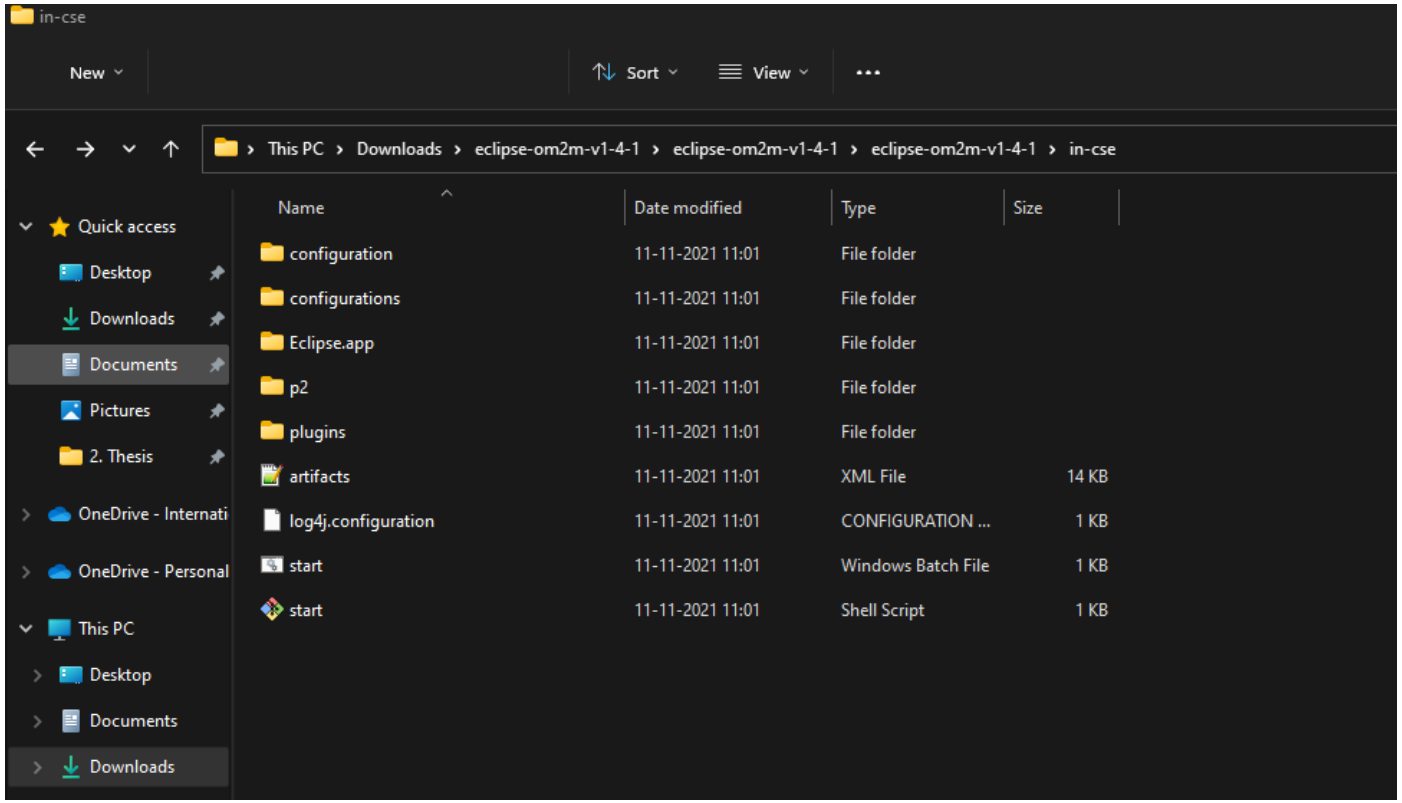
Using HTTP-Restful Client and Json format we need to create

Application Entity

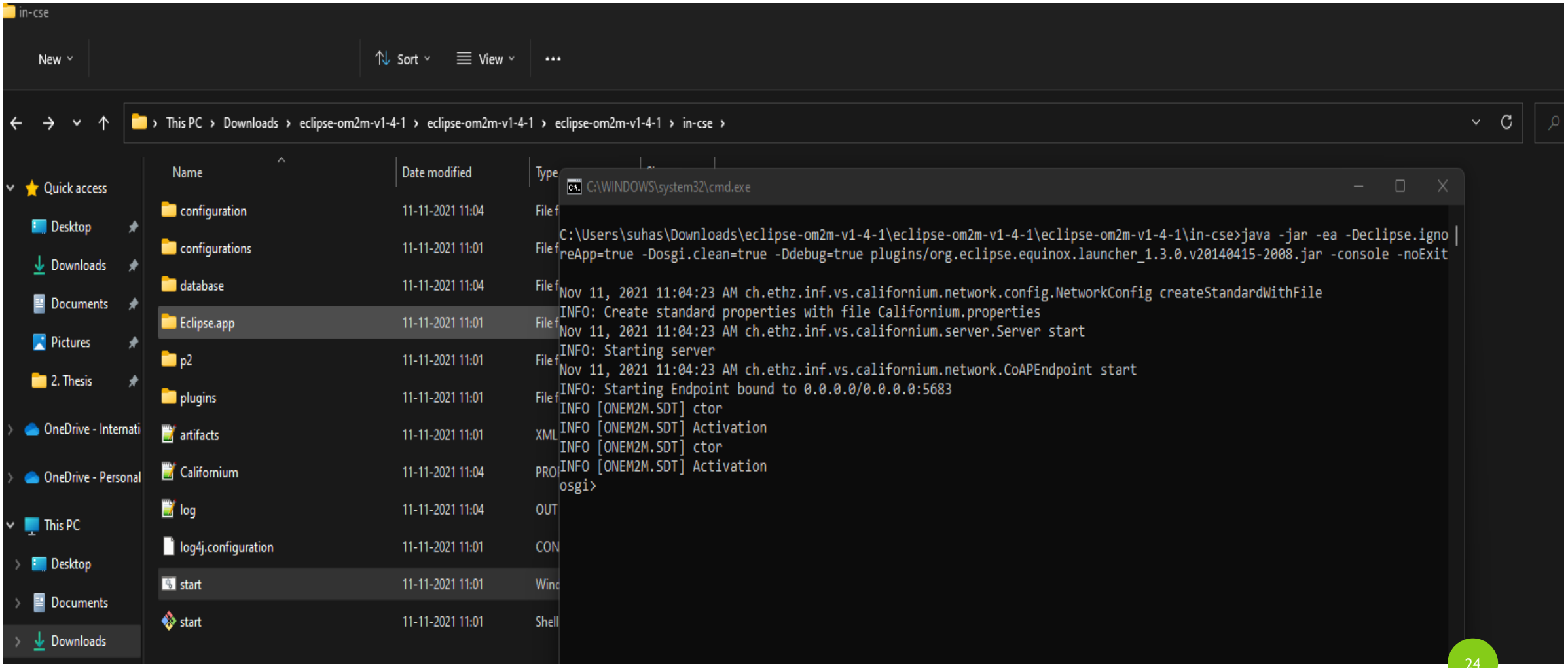
- Container for each Device
  - Container for Describing what the Data Sends
    - A data Packet describing the device's data
  - Container for Storing Data
    - The actual data

Launching OM2M

- Start the OM2M platform by executing the "start.bat" script on Windows or "start.sh" on Linux and Mac OS.



Successful OM2M Launch



OM2M Home Page

- Open http://127.0.0.1:8080/webpage in browser.
- Type admin as username and password

OM2M CSE Resource Tree

http://127.0.0.1:8080/~/in-cse

- in-name
  - acp_admin
  - SDT_Home_Monitoring_Application_ACP
  - ACP_Device_Admin_1636608866302
  - SDT_Home_Monitoring_Application
  - SDT_IPE

Thank You