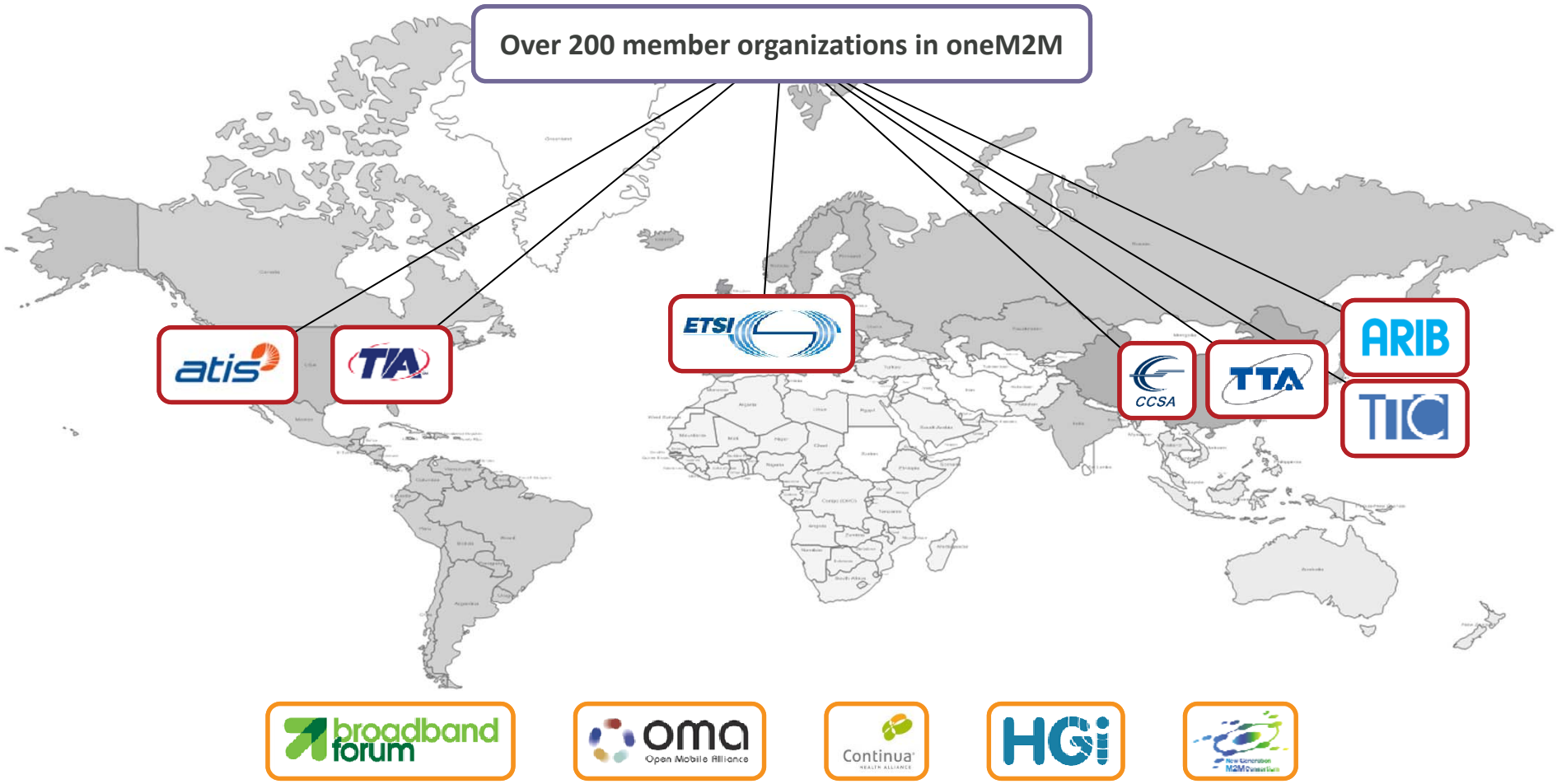
ONEM2M SERVICE LAYER PLATFORM – INITIAL RELEASE

Omar Elloumi (ALU)
oneM2M TP Chair (Elect)
omar.elloumi@alcatel-lucent.com
oneM2M www.oneM2M.org

The Partnership Project



Purpose & Deliverables

Purpose

To specify and promote an

M2M Common Service Layer

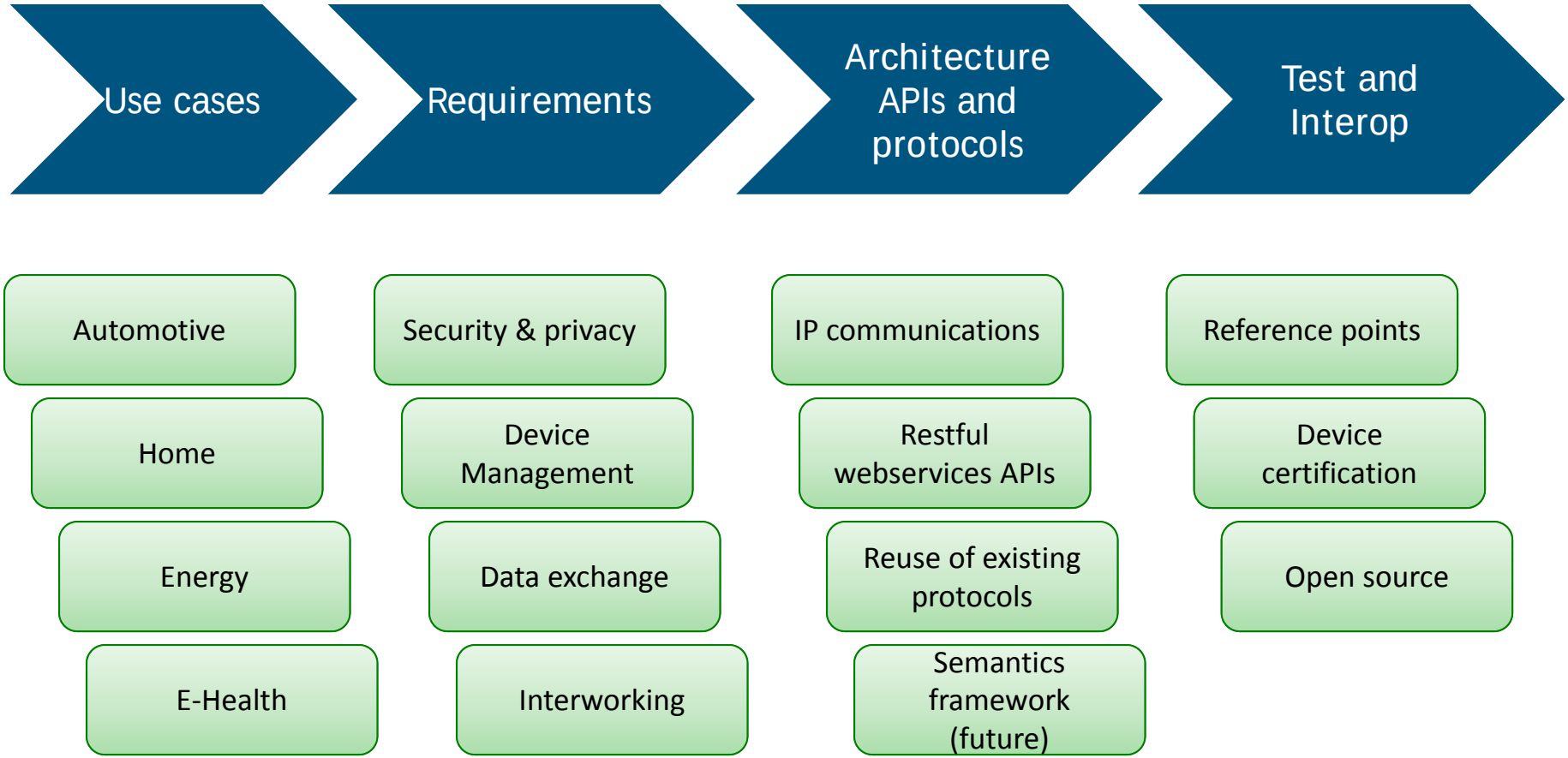
Deliverables

Technical Reports and Technical Specifications

M2M Common Service Layer in a nutshell

- It is a software layer
- It sits between M2M applications and communication HW/SW that provides data transport
- It normally rides on top of IP
- It provides functions that M2M applications across different industry segments commonly need. Those functions are exposed to Applications via IT-friendly APIs.
- It allows for distributed intelligence (device, gateway, cloud apps)

Standardization approach



Fresh news

SK Telecom Develops World's First Global Standard IoT Platform

SK Telecom announced on Dec. 7 that it completed developing the industry's first open-source oneM2M-based Internet of Things platform.

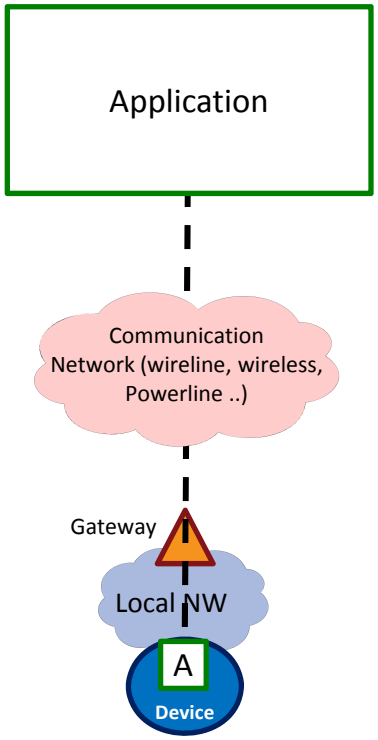
oneM2M is the largest standardization body for Machine-to-Machine (M2M) communications established in 2012. The organization includes related organizations, research centers, and enterprises from major countries around the world.

After successfully commercializing an M2M platform in 2008, SKT has been involved in the development of open-source platform Mobius from late 2011 as a national project, together with the Korea Electronics Technology Institute and Ntels. As oneM2M announced a candidate for an IoT/M2M standard in August of this year, Korea's largest mobile carrier implemented the standard with the Mobius, finishing the development of a commercialization-ready platform.

oneM2M Architecture approach

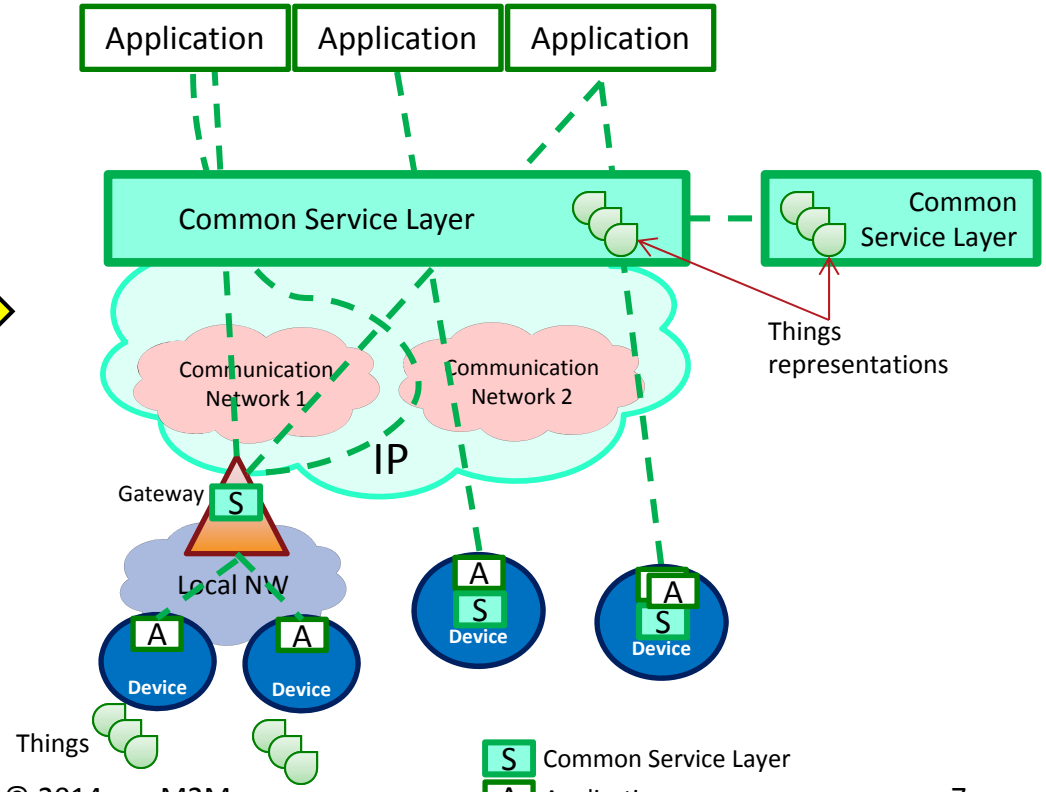
Pipe (vertical):

1 Application, 1 NW,
1 (or few) type of Device
Point to point communications



Horizontal (based on common Layer)

Applications share common service and network infrastructure
Multipoint communications



Common Service Functions

Why does it matter

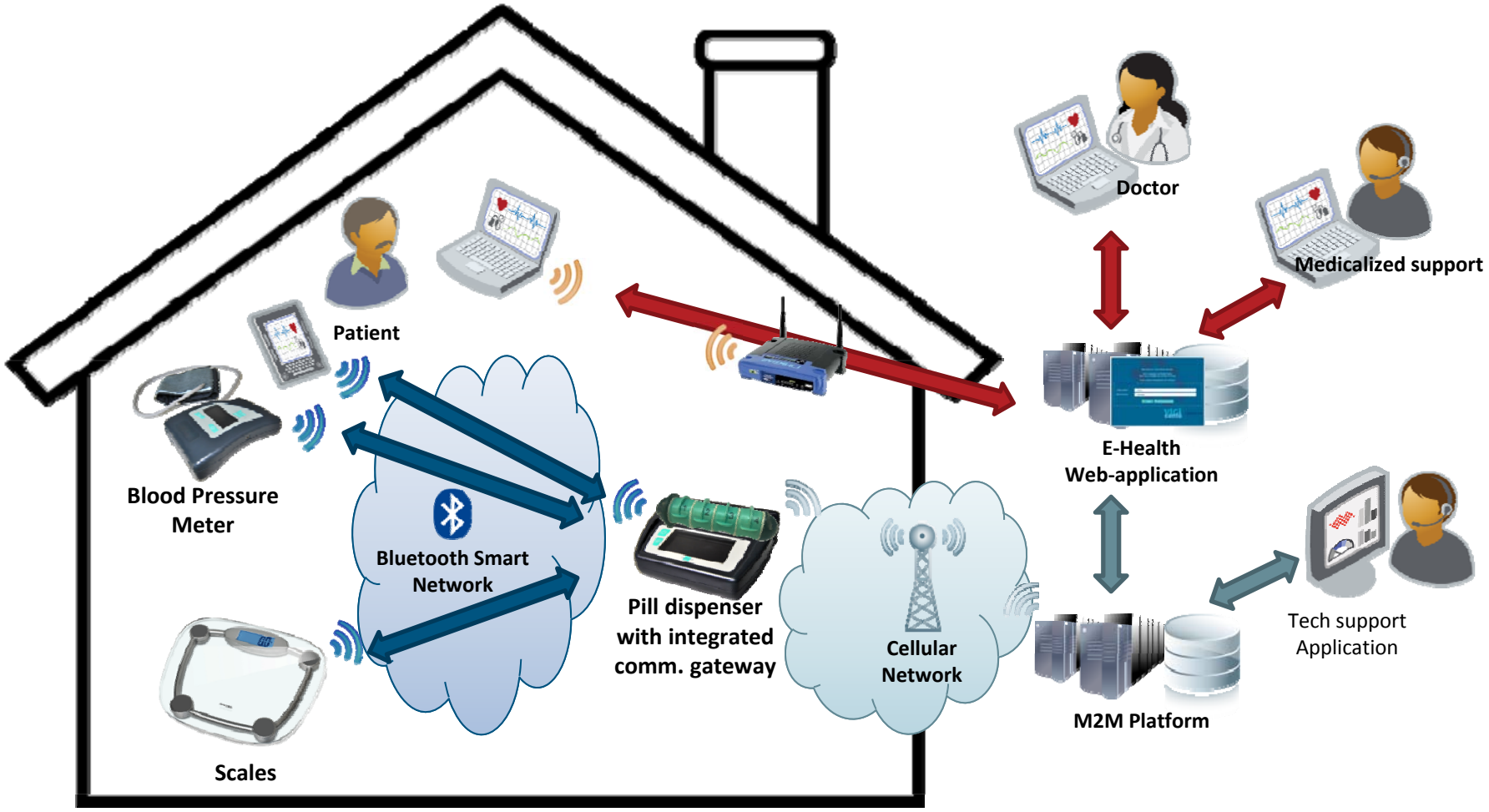
Combat fragmentation	• Healthy eco-system with economies of scale • More partnering choices and opportunities for M2M/IOT industry stakeholders
Lower CAPEX	• Standardized protocols / APIs -> simplifies application development/deployment • Cross-vertical standards -> same devices and back-ends in different industries
Lower OPEX	• Standard features to use networks more efficiently -> get better tariffs • Flexibility for verticals -> utilize best transport network meeting business needs
Time to Market	Reduced development, test and deployment lifecycles through focusing on core business (application logic)

oneM2M is IoT ready

Technical Specifications

ftp://ftp.onem2m.org/Work Programme/

Example Scenario – E-Health

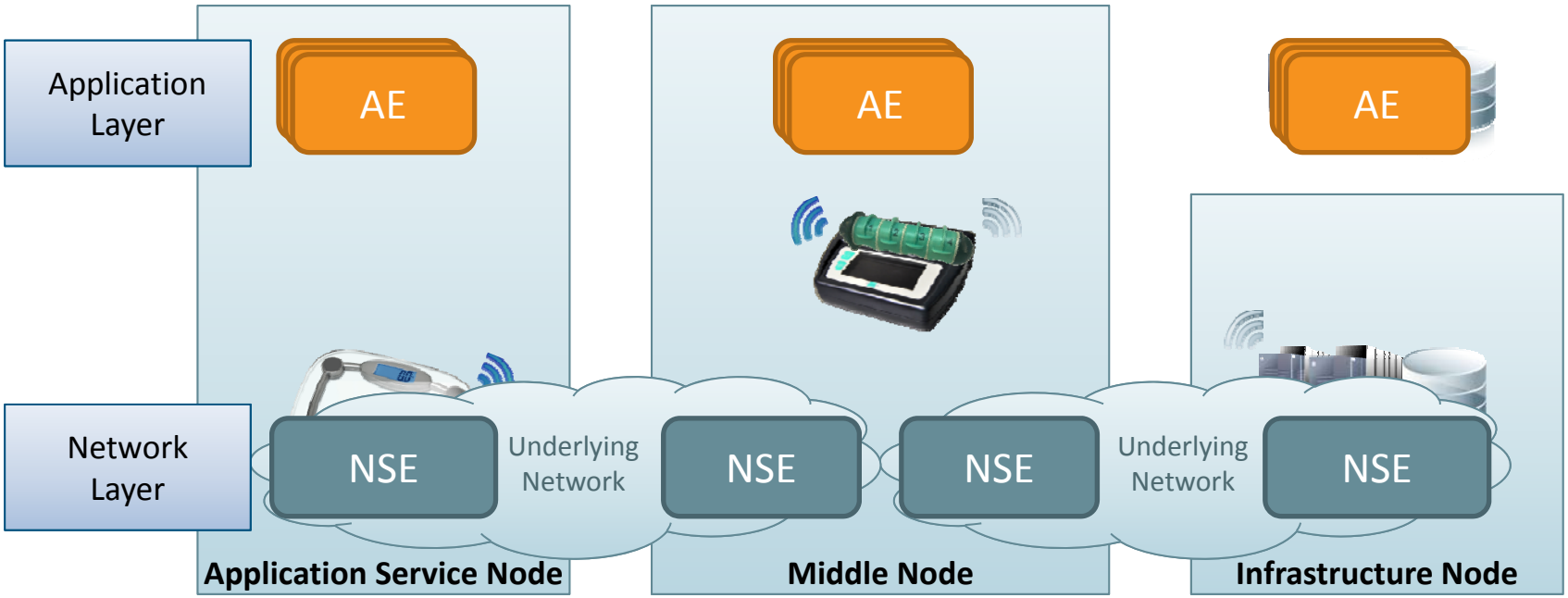


Design principles

- IP-based, but interworks with specific IP and non IP technologies in the M2M Area networks
- RESTful resource oriented APIs, resources are representations of devices, applications, things and related descriptions, etc.
- Distributed intelligence (device, gateway, edge, cloud)
- Reuse of existing device management frameworks
- Reuse of existing data exchange protocols
- Reuse of existing security
- Reuse of underlying network capabilities such as location, triggering, etc.
- Resource access control policies allows many to many communications framework
- Future proof – ready to add semantics support
- No mandated implementation (Database choice, intelligence location, etc.)

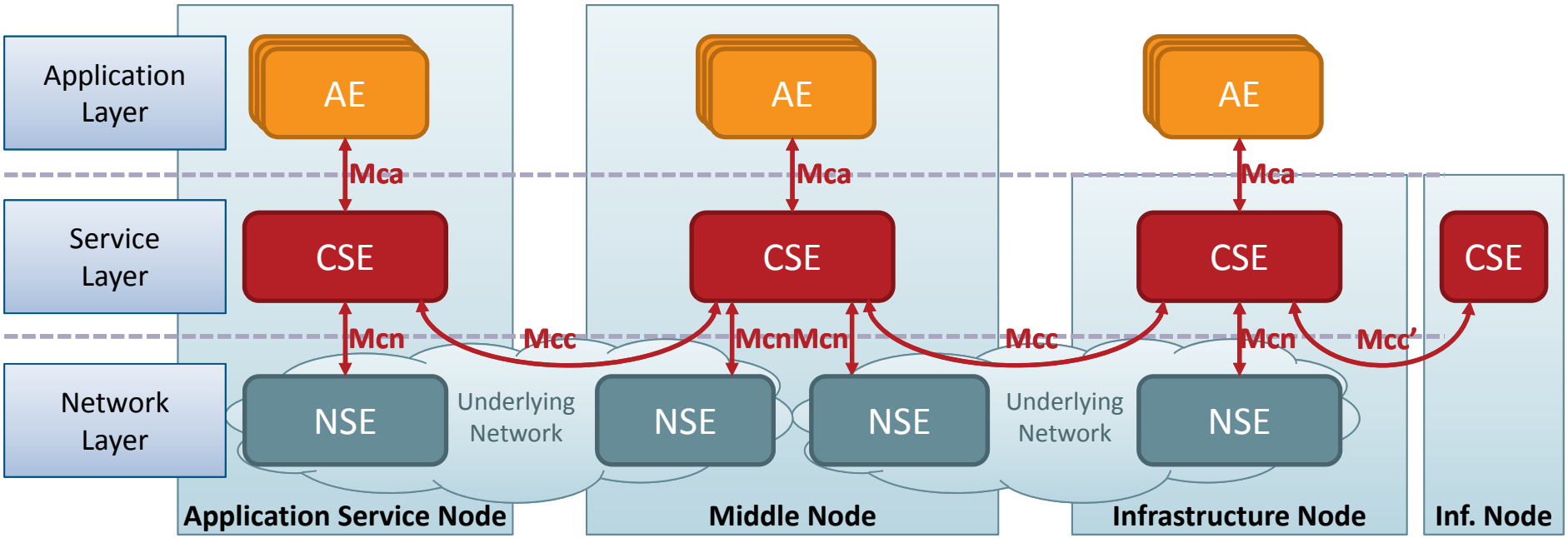
Architecture

Application Entity	Provides application logic for the end-to-end M2M solutions
Network Services Entity	Provides services to the CSEs besides the pure data transport
Node	Logical equivalent of a physical (or possibly virtualized, especially on the server side) device

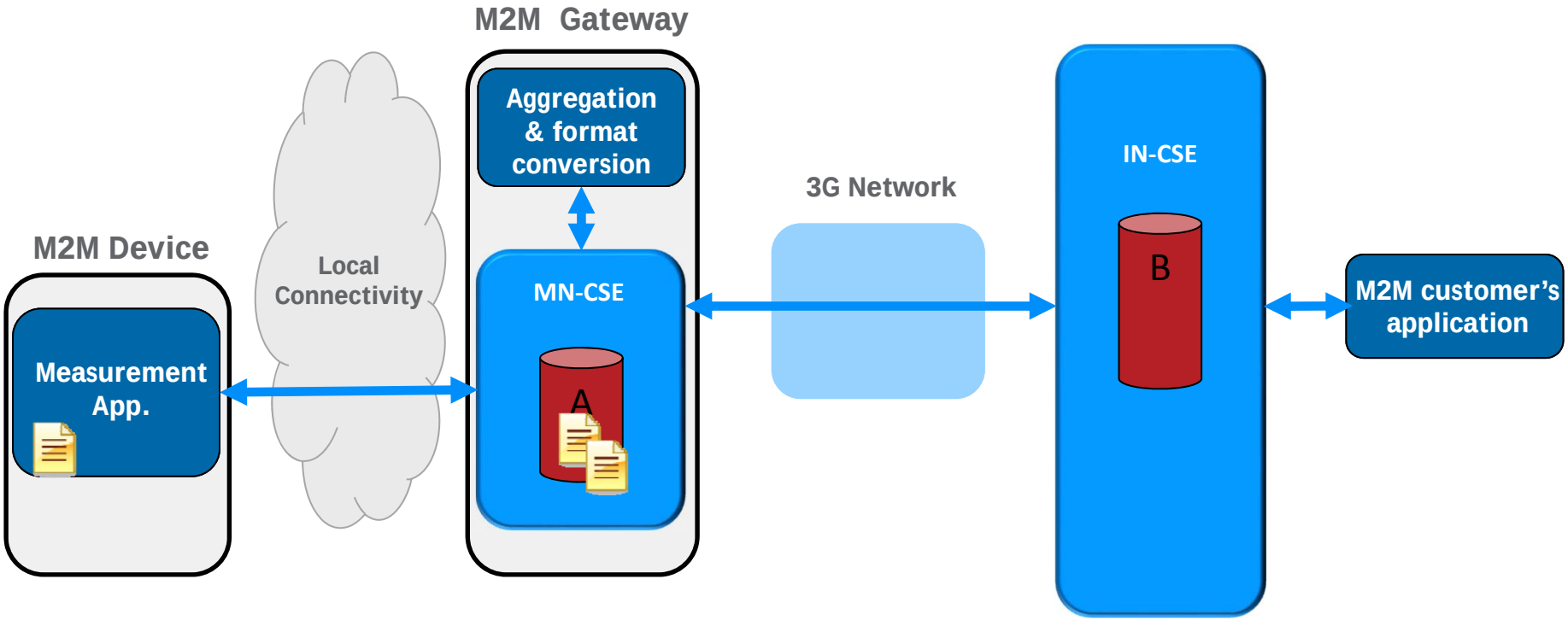


Architecture

Reference Point	One or more interfaces - Mca, Mcn, Mcc and Mcc' (between 2 service providers)
Common Services Entity	Provides the set of "service functions" that are common to the M2M environments
Application Entity	Provides application logic for the end-to-end M2M solutions
Network Services Entity	Provides services to the CSEs besides the pure data transport
Node	Logical equivalent of a physical (or possibly virtualized, especially on the server side) device



Concrete example



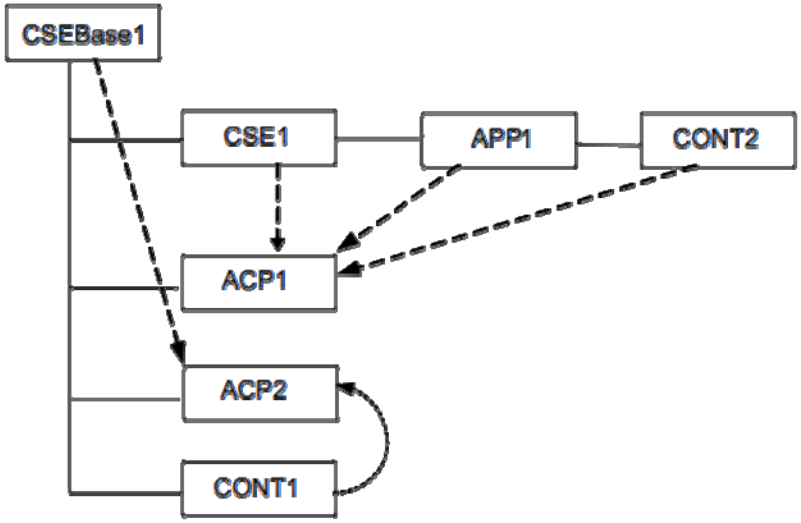
Starting assumptions:
- Bootstrapping / DM is done (provisioning of credentials/apps)
- MN-CSE and IN-CSE have logically connected (authentication, binding, encryption)
- Apps have authenticated to xCSE and access right were established

=> Very little effort to synch the different apps

Information Modelling

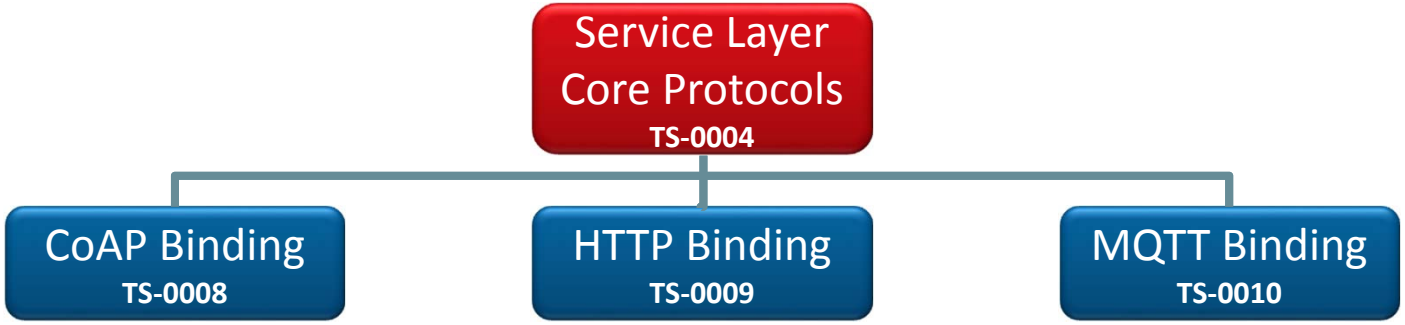
Resource-based information model

- Information is stored in the system as Resources
- A given Resource can be identified with a Uniform Resource Identifier
- A given Resource is of one of the defined Resource Types
- The Resource Type determines the semantics of the information in the Resource
- Resources can be Created, Read, Updated or Deleted to manipulate the information
- Resources are organized in a tree-like structure and connected by links



Communication Protocols

Reuse IP-based existing protocols



XML or JSON Content serialization

HTTP Example

REQUEST

```
GET http://provider.net/home/temperature HTTP/1.1
Host: provider.net
From: //provider.net/CSE-1234/WeatherApp42
X-M2M-RI: 56398096
Accept: application/onem2m-resource+json
```

RESPONSE

```
HTTP/1.1 200 OK
X-M2M-RI: 56398096
Content-Type: application/onem2m-resource+json
Content-Length: 107
{"typeOfContent":"application/json",
"encoding":1,
"content": "{'timestamp':1413405177000,'value':25.32}"
}
```

Security Challenges & Solutions

1. Large variety of scenarios
2. Any device in any deployment
3. A device cannot make "judgment calls" on privacy

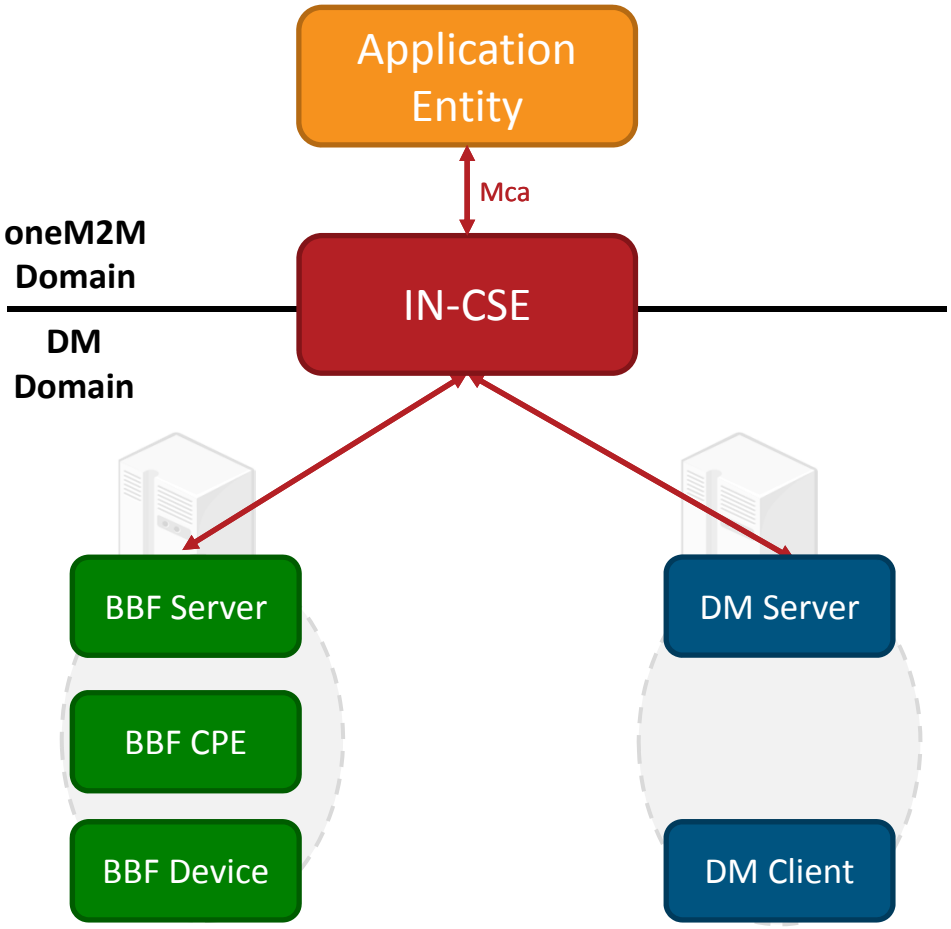
A. Secure communication
   various authentication options

B. Remote provisioning
   various authentication options

C. Access Control Policy
   express wide variety of rules

Interworking – OMA & BBF

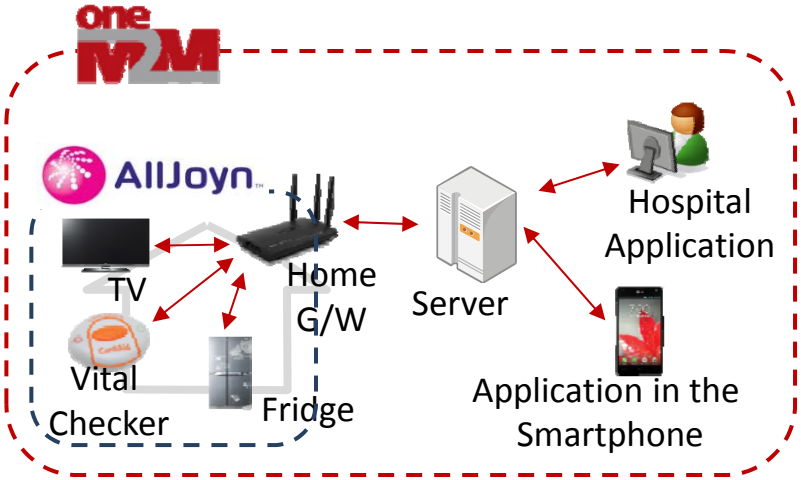
Reuse existing Device Management technologies



OMA DM 1.3 | OMA DM 2.0 | OMA LWM2M

BBF TR-069

Interworking – AllJoyn



	AllJoyn	oneM2M
Network Architecture	Peer-to-Peer in LAN	Server-to-Client in WAN
API Style	RPC(RMI) API	Resource-based API
Discovery Style	Proactive Discovery	Passive Discovery

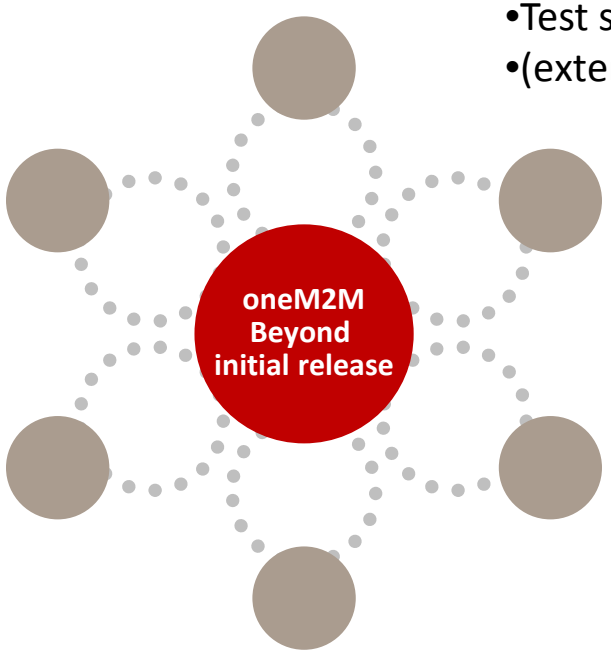
Candidate features for oneM2M next steps

Testing and interoperability

- Test specifications
- (external) certification

Wide-scale deployment enchancements:

- Home Domain Enablement
- API versioning
- Plug and play scenarios
- Any lessons learnt from prototypes and deployment



Security

- E2e security
- Group authentication
- Role based security

Interworking

- AllJoyn
- Enhance 3GPP interworking

Application enablements

- App APIs
- App development guide and SDK
- Service profiling

Big data enablement

- Semantics support and use cases
- Ontology, query, reasoning

Thank You!

Q&A