



oneM2M Interworking Overview

SeungMyeong JEONG (sm.jeong@keti.re.kr)

Korea Electronics Technology Institute

2020.11.13

The project "International Digital Cooperation - ICT Standardisation" is funded by the European Union



- Since oneM2M Release 2, interworking has been emphasized
 - There are a lot of IoT technologies and products for oneM2M to reach out
- This tutorial provides how non-oneM2M technologies can be interworked with oneM2M systems so you can leverage your existing systems to have synergy with oneM2M

- Scope of oneM2M Interworking
- Basics of IPE
- Sample IPE Recipe



Scope of oneM2M Interworking

Types of Interworking

- There are many interworking specifications in oneM2M
 - E.g. Rel-3 defines LwM2M, OCF, 3GPP, OGSi interworking TSes
 - There are couple of interworking types so there has been defined the interworking framework
 - to provide the general guide instead of defining a number of interworking TSes which would be really similar each other

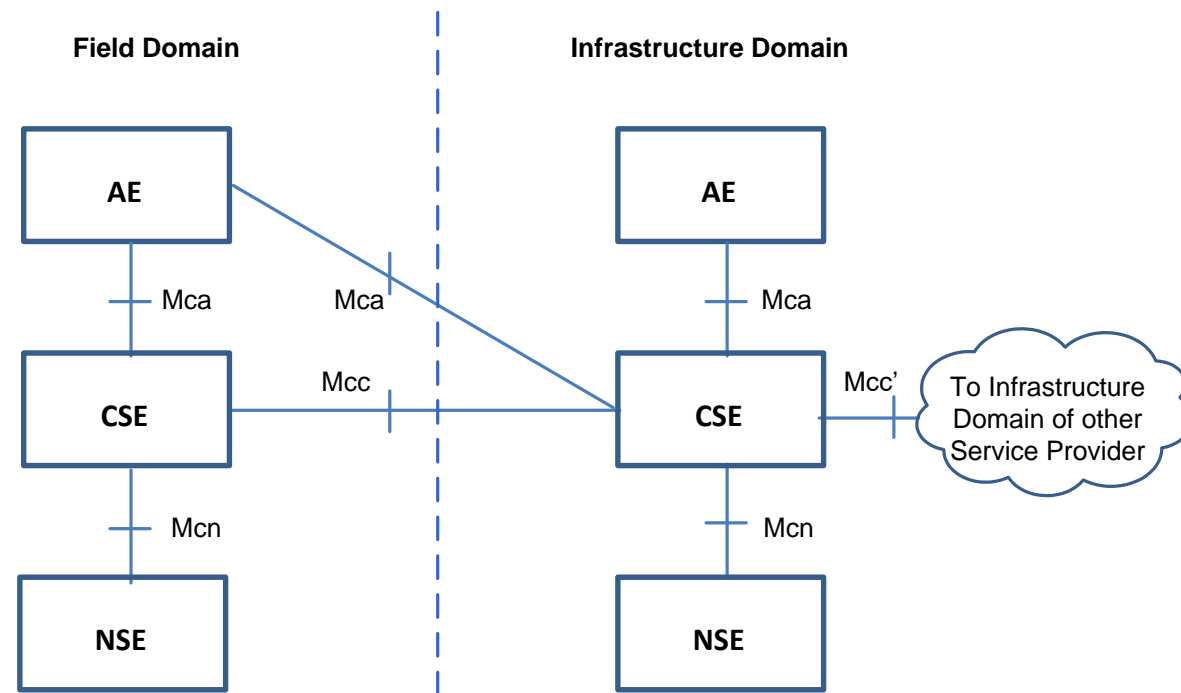
[TS-0014-LWM2M_Interworking-V3_1_1.DOC](#)
[TS-0016-Secure_Environment_Abstraction-V3_0_2.DOCX](#)
[TS-0017-Implementation_Conformance_Statements-V3_1_0.DOC](#)
[TS-0018-Test_Suite_Structure_and_Test_Purposes-V3_8_1.DOC](#)
[TS-0019-Abstract_Test_Suite_&_implementation_eXtra_Information_for_Test-V3_1_0.DOCX](#)
[TS-0020-WebSocket_Protocol_Binding-V3_0_1.DOCX](#)
[TS-0022-Field_Device_Configuration-V3_0_2.ZIP](#)
[TS-0023-Home_Appliances_Information_Model_and_Mapping-V3_7_3.DOC](#)
[TS-0024-OCF_Interworking-V3_2_2.DOCX](#)
[TS-0025-Product_profiles-V3_1_0.DOC](#)
[TS-0026-3GPP_interworking-V3_5_0.ZIP](#)
[TS-0030-Ontology_based_Interworking-V3_0_2.DOCX](#)
[TS-0031-Feature_Catalogue-V3_0_1.DOCX](#)
[TS-0032-MAF_and_MEF_Interface_Specification-V3_0_0.ZIP](#)
[TS-0033-Interworking_Framework-V3_0_0.DOC](#)
[TS-0034-Semantics_Support-V3_0_2.DOC](#)
[TS-0035-OSGi_interworking-V3_0_0.DOCX](#)

Latest Draft Release 3 Technical Reports

[TR-0001-Use_Cases_Collection-V3_1_1.DOC](#)
[TR-0018-Industrial_Domain_Enablement-V2_5_1.ZIP](#)
[TR-0024-3GPP_Interworking-V3_0_0.DOCX](#)
[TR-0026-Vehicular_Domain_Enablement-V3_0_1.DOCX](#)
[TR-0033-Study_on_Enhanced_Semantic_Enablement-V3_0_0.DOC](#)
[TR-0036-Smart_City-V0_4_0.ZIP](#)
[TR-0047-Developer_Guide_of_3GPP_Interworking-V0_1_2.DOC](#)
[TR-0051-oneM2M_API_guide-V0_6_0.DOC](#)
[TR-0056-Summary_of_Differences_between_Rel-2A_&_Rel-3-V0_2_0.DOC](#)

Types of Interworking

- 3GPP interworking works on Mcn reference point
 - Refer TR-0047 for 3GPP interworking developer guide, which is not the scope of this tutorial
- The other interworking schemes are for Mca reference point
 - This means most of the oneM2M interworking specifications works with the APIs over the Mca reference point

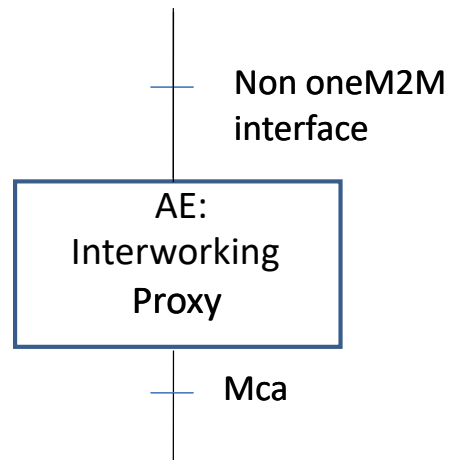


oneM2M Functional Architecture (TS-0001)

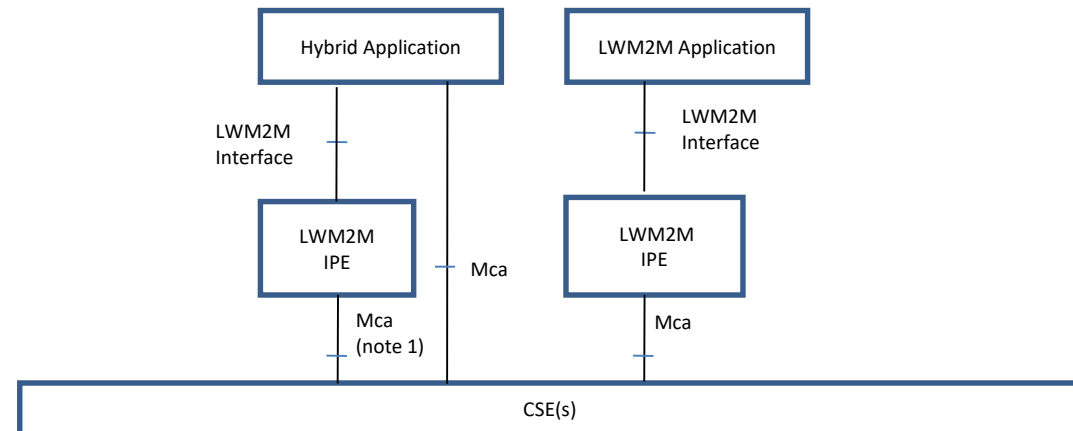
* NSE: Network Services Entity

Scope of Interworking

- The scope of interworking in this tutorial is Interworking Proxy application Entity (IPE) based ones
 - IPE is a bridge between non-oneM2M system to oneM2M platforms
 - IPE represents the non-oneM2M system as oneM2M resources
 - E.g. LwM2M, OCF, AllJoyn



Interworking Proxy (TS-0001)



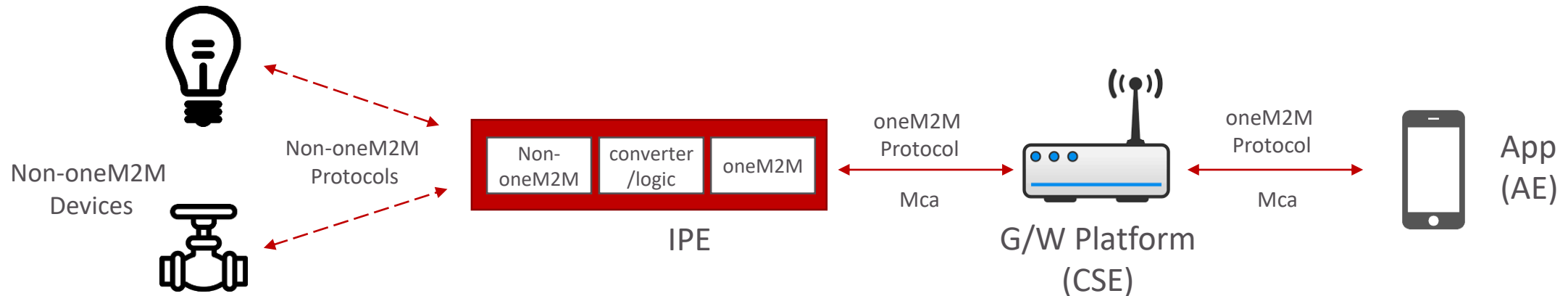
LwM2M Interworking Scenarios (TS-0014)



Basics of IPE

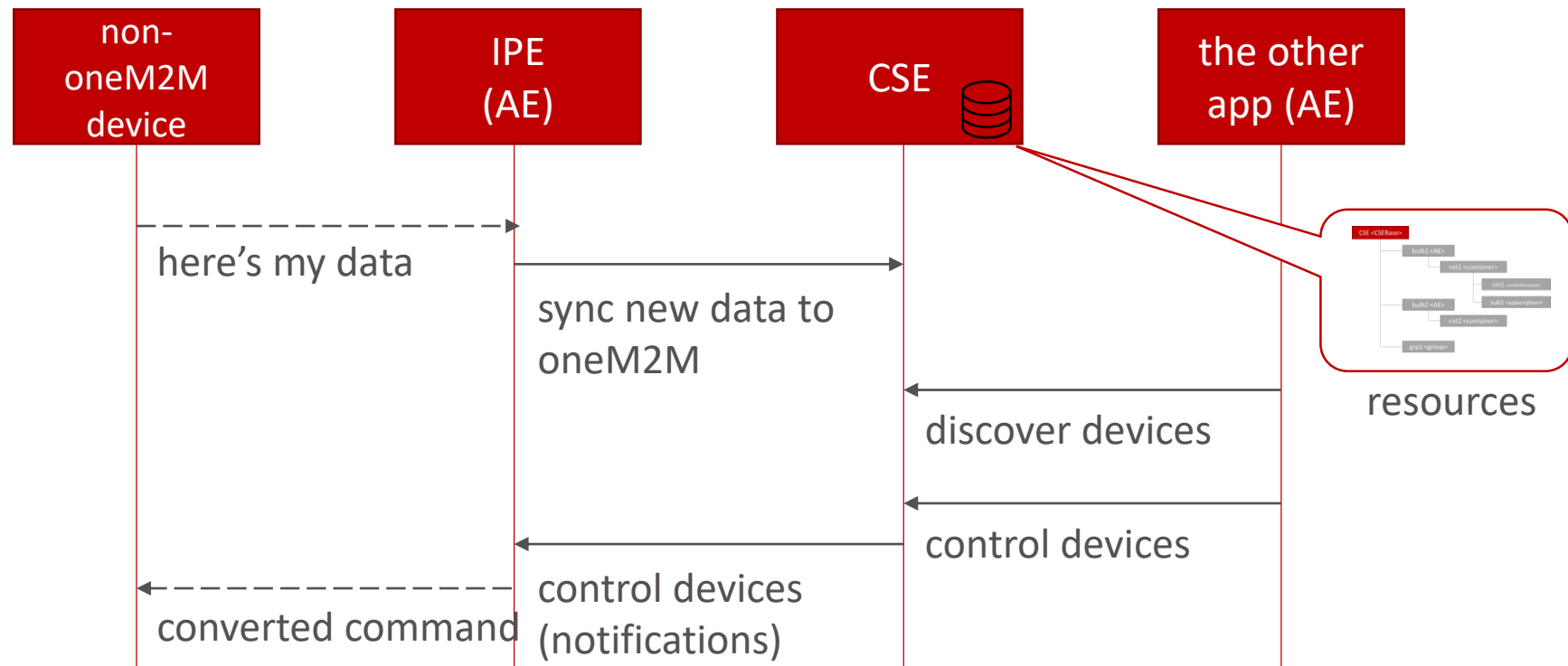
IPE as a Bridge

- An IPE supports oneM2M protocols as well as non-oneM2M's
 - converts different data models, and
 - also converts different protocols
 - the conversions works on both ways



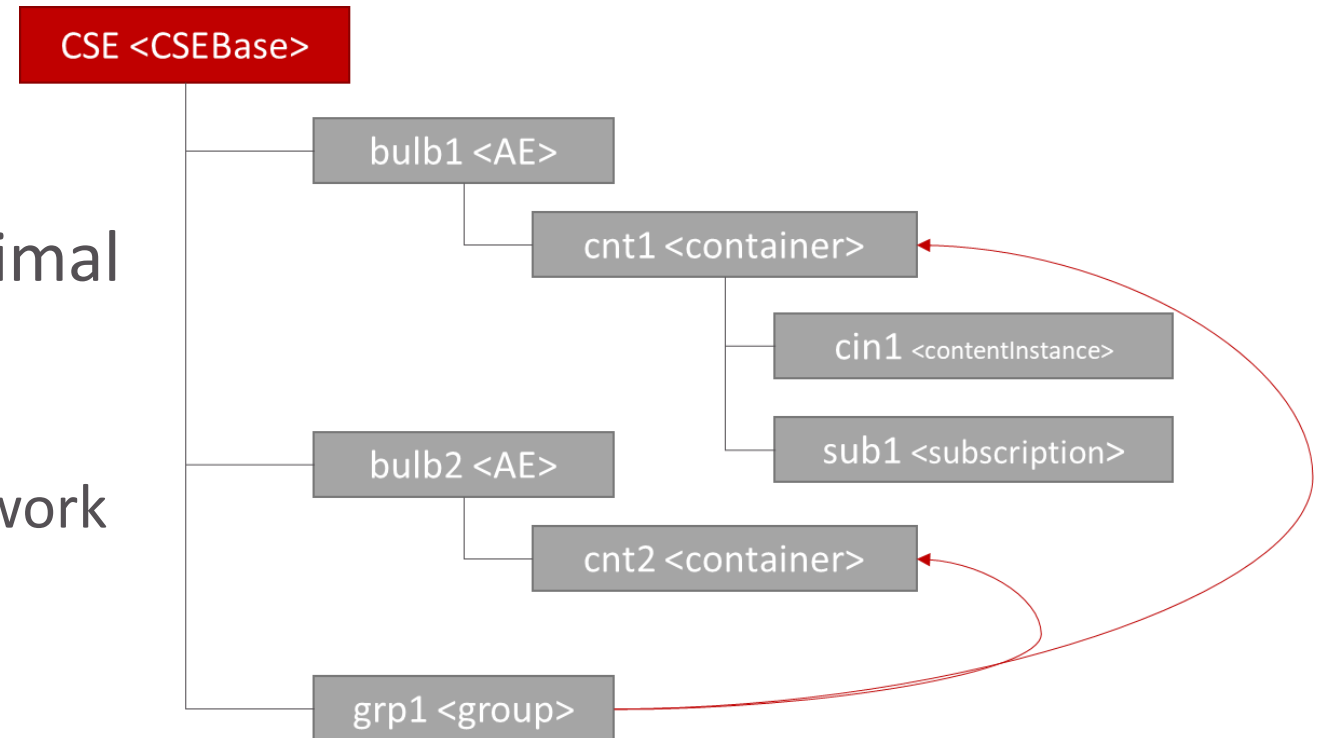
Designing an IPE

- To have a oneM2M interworking with something, the key is how to represent the non-oneM2M system with oneM2M resources
 - note that oneM2M provides RESTful APIs



Resource Tree

- Design a resource tree for an IPE
 - Resource types and their APIs are ingredients for your IPE recipe
 - E.g. container vs. flexContainer
- There's the guideline for IoT proximal network to be represented in oneM2M resource tree
 - Refer TS-0033 Interworking Framework



Designing an IPE

1. Registration
 - non-oneM2M Devices (NoDN)
 - IPE itself
2. Initial resource creation
 - Data Sharing Resources (DSR) for NoDN's data points and actuation points
 - e.g. <container>, <contentInstance>, <flexContainer> resources
3. Expose initial status
 - Get current status data of the non-oneM2M system and expose them as oneM2M resources
4. Subscribe to actuation points
5. Deal with data sync between the two worlds

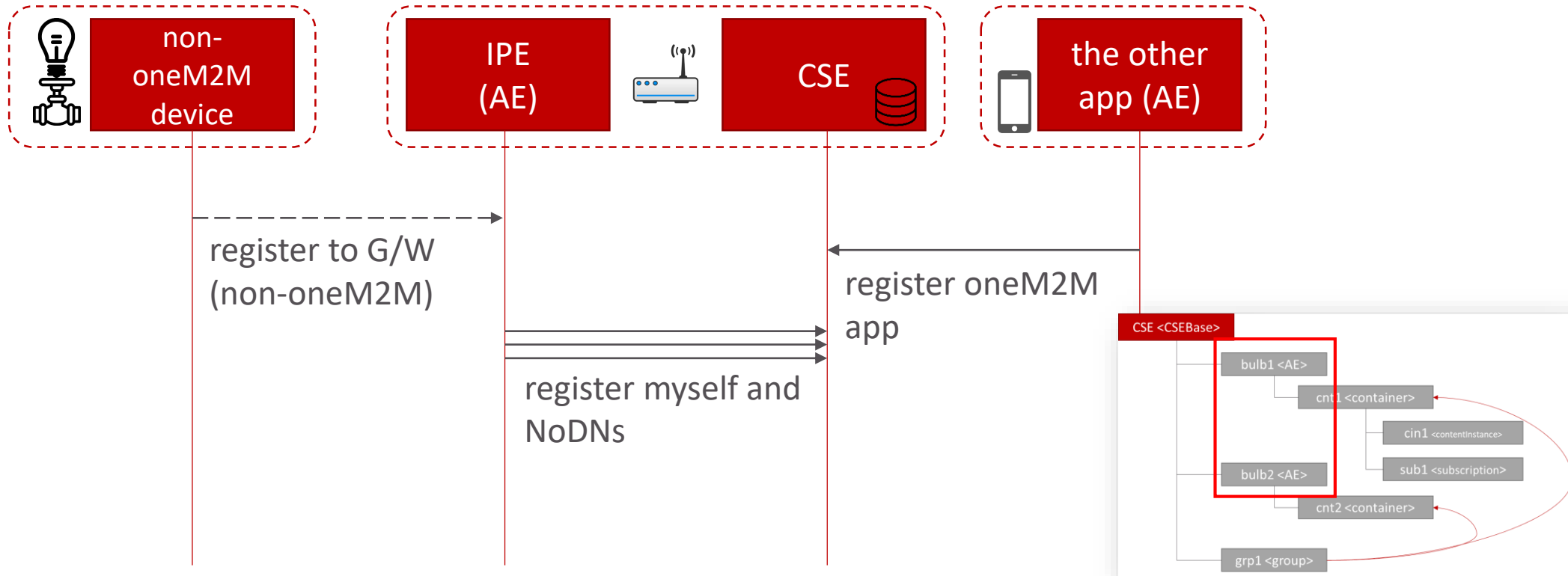
NoDN: Non-oneM2M Device Node



Sample IPE Recipe

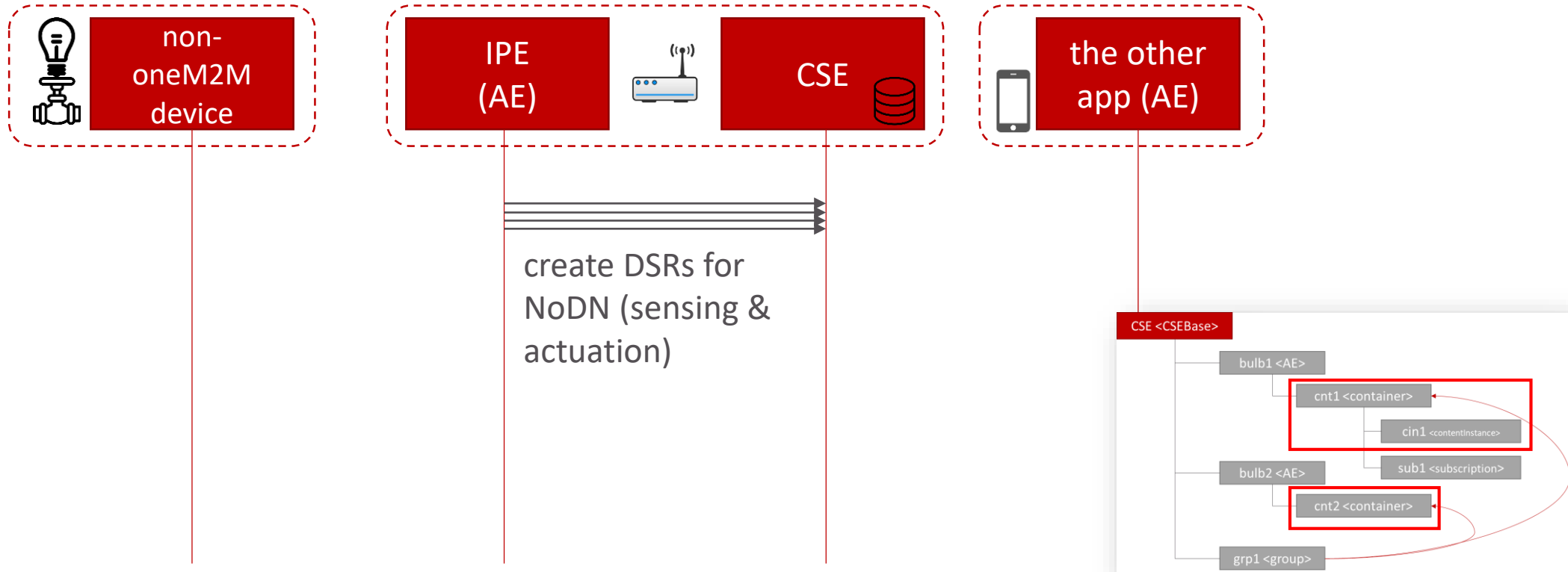
Registration

- IPE gets NoDNs information and register them as <AE> resources
 - fetching NoDN information is not oneM2M scope but target protocol dependent
 - e.g. Zigbee device registration



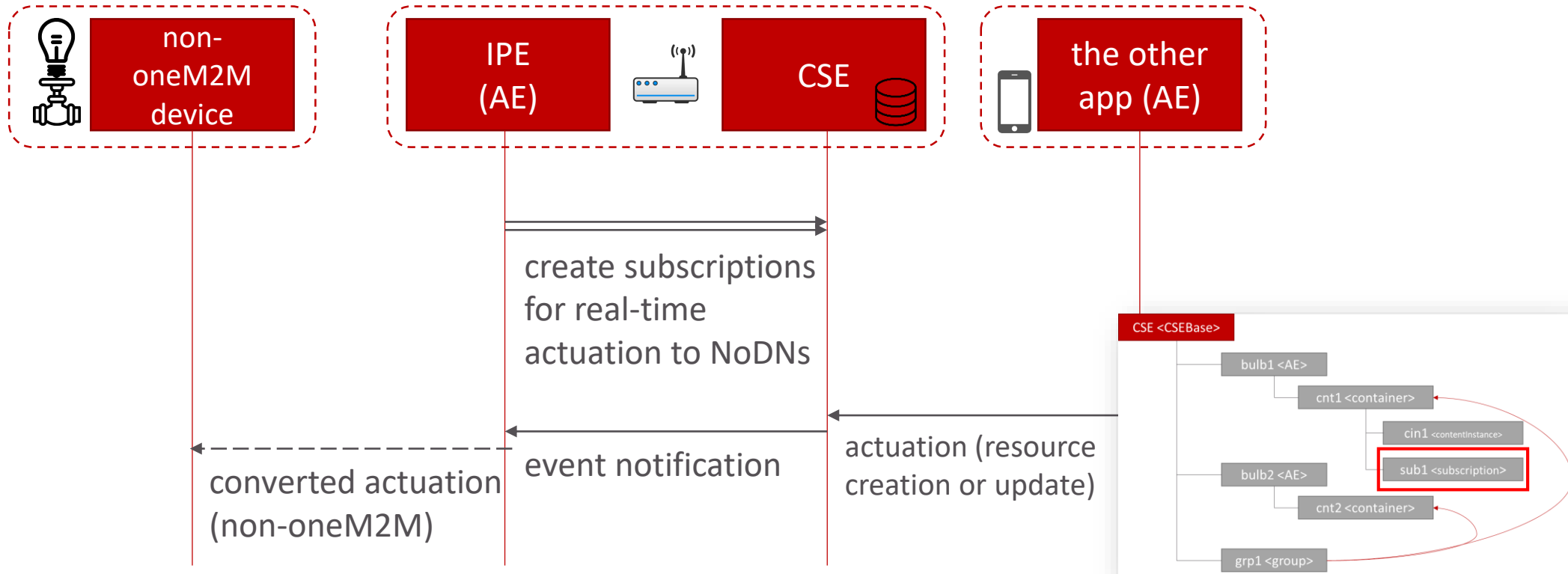
Initialization

- Create DSRs to expose NoDNs
 - Preferred DSR type is your choice (e.g. container vs. flexContainer)
 - Provide latest status of NoDNs to the DSRs

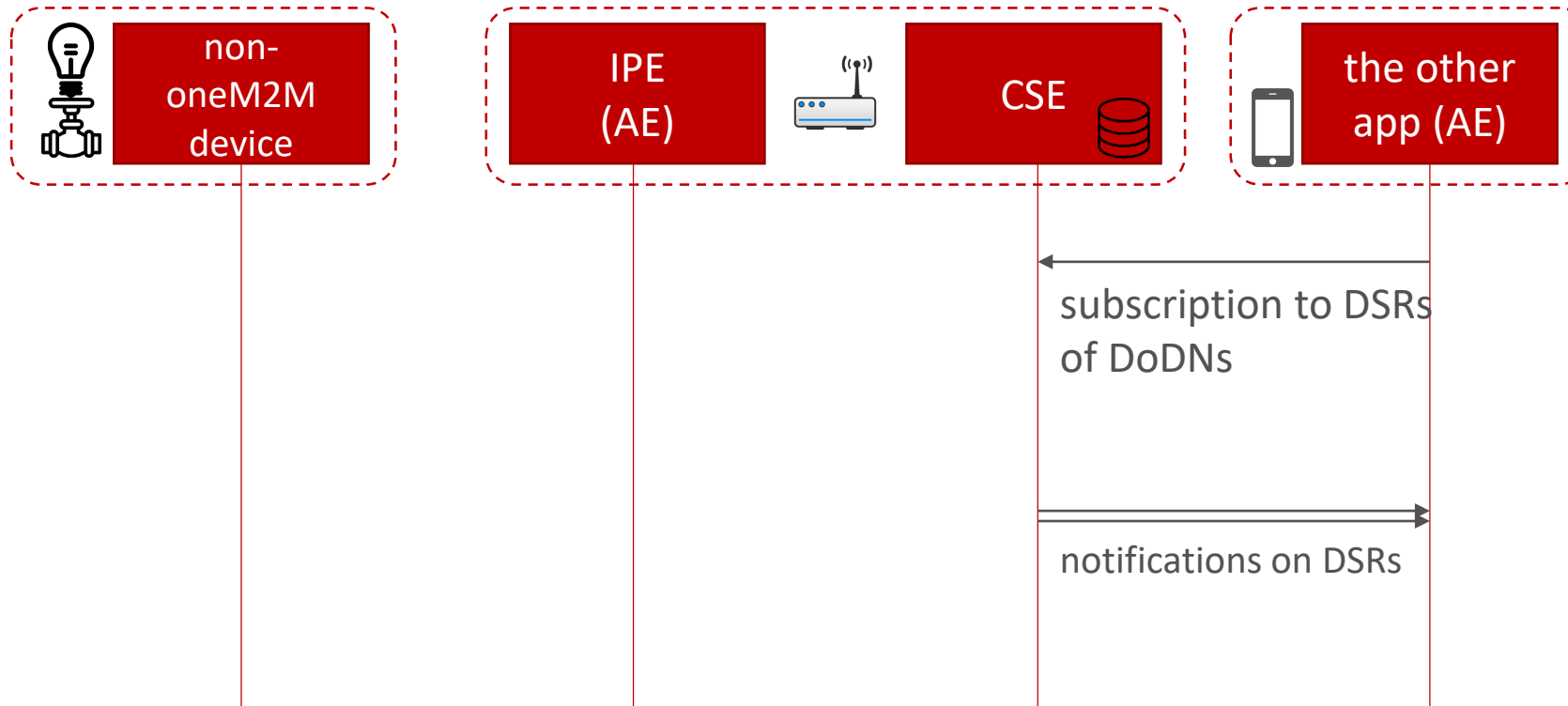


Event Subscription

- Create subscriptions for event-driven interworking
 - When there's a resource creation or an update, IPE gets notified due to subscription
 - Push vs. pull



- Get up-to-date status data of NoDNs and sync to the corresponding resources
 - Depending on the target protocol there can be their subscription/notification feature
 - Push vs. pull



Group Fan-out

- Group fan-out can be leveraged by other oneM2M apps
 - E.g. initial data loading of all devices

