oneM2M Rel-1 Functional Architecture (Part1)

2014. 09. 04

최성찬
(csc@keti.re.kr)

KETI 전자부품연구원
Korea Electronics Technology Institute

Contents

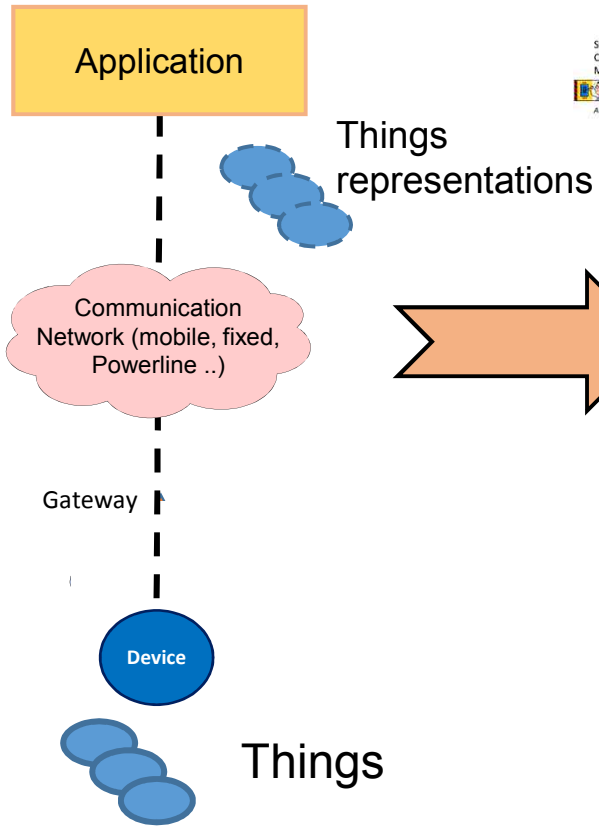
- Introduction to oneM2M Architecture
  - oneM2M architecture vision
  - oneM2M architecture benefits
- oneM2M Architecture Features
  - Architectural Model
  - Architecture Capability & Function
  - Identification and Addressing
  - Communication flow on reference points
- Summary

Intro - oneM2M architecture

oneM2M - inverting the pipes

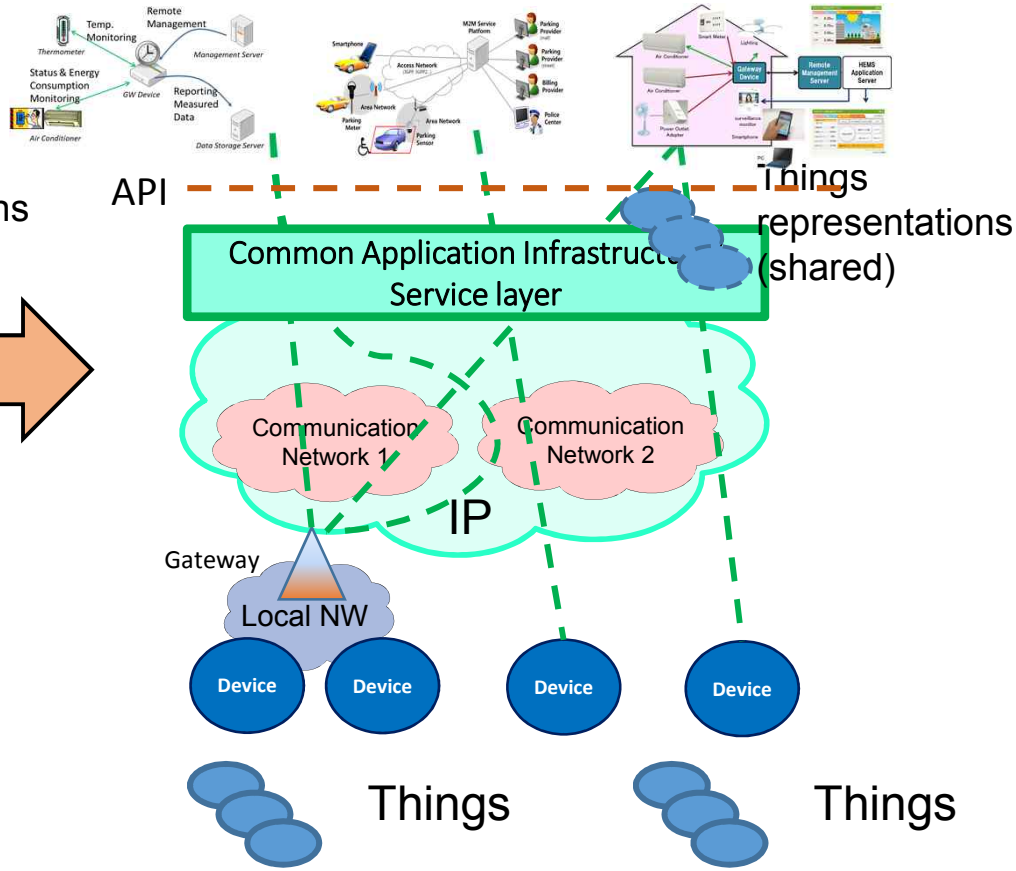
Pipe (vertical):

1 Application, 1 NW,
1 (or few) type of Device



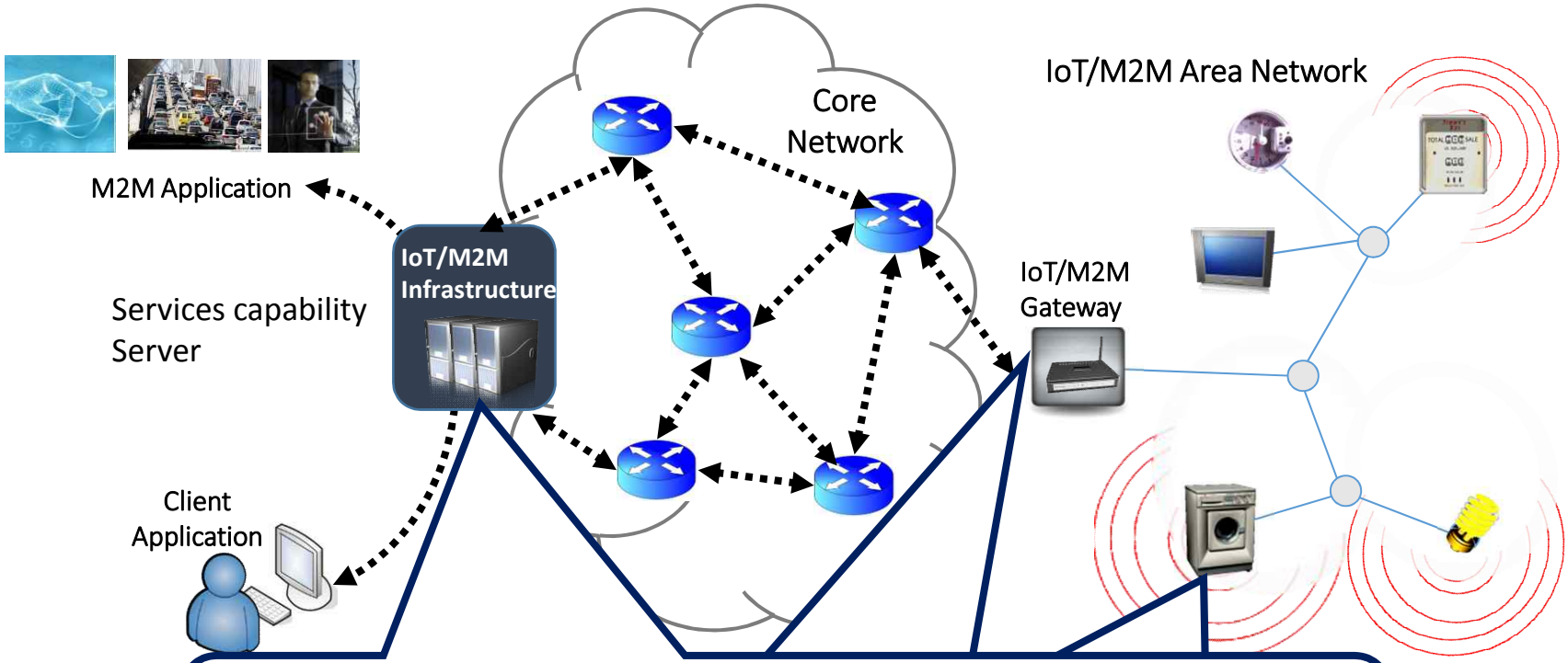
Horizontal (based on common Layer)

Applications share common infrastructure, environments and network elements



Source: oneM2M

oneM2M Standards on Service Layer



Standards for M2M Service Layer

- End-to-end enablement across servers, gateway, devices
- Standardized service interfaces

Source: oneM2M

Simple Definition of Service Layer

- M2M Service Layer
  - *Middleware* supporting secure end-to-end data/control exchange between M2M devices and customer applications by providing functions for remote provisioning & activation, security, connectivity, buffering, aggregation, device management, etc.

  - Software layer / "Platform"*: CSE=Common Services Entities*
  - Sits between M2M applications and communication HW/SW that provides data transport
  - Normally rides on top of IP

  - Provides functions that IoT/M2M applications across different industry segments commonly need => Horizontal layer

Source: oneM2M

Why oneM2M architecture

What are the General Benefits?

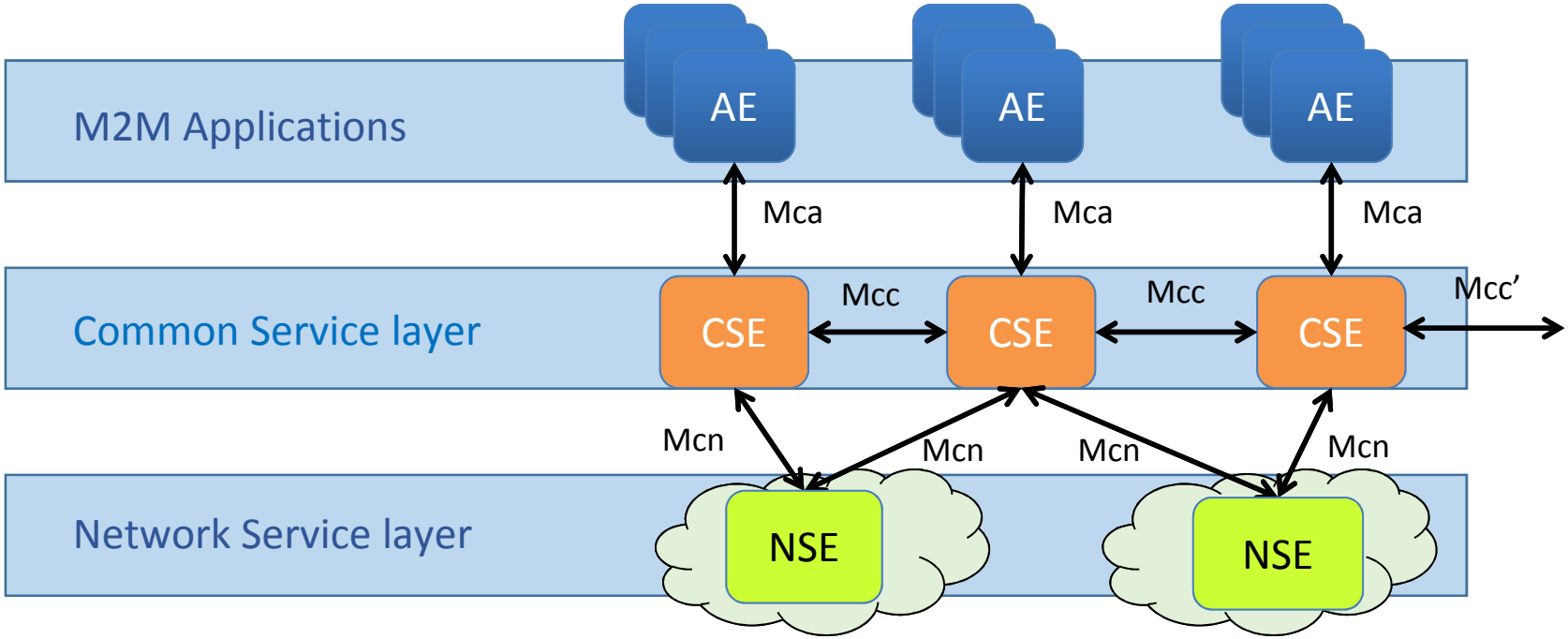
- Combat fragmentation => Healthy eco system with economies of scale
- Standards help lowering CAPEX for M2M Services:
  - Standardized protocols / APIs => simplifies application development
  - Cross-vertical standards => same devices and back-ends in different industries, etc.
- Standards help lowering OPEX for M2M Services:
  - Standard features to use networks more efficiently => get better tariffs
  - Flexibility for verticals => utilize best transport network meeting business needs, etc.
- Faster time to market due to reduced development/test/deployment time
- Address use cases and markets where cost was prohibitive so far
- Allow M2M users to focus on their core business and not worry about solvi ng M2M communication challenges on their own

Source: oneM2M

oneM2M Architecture Model

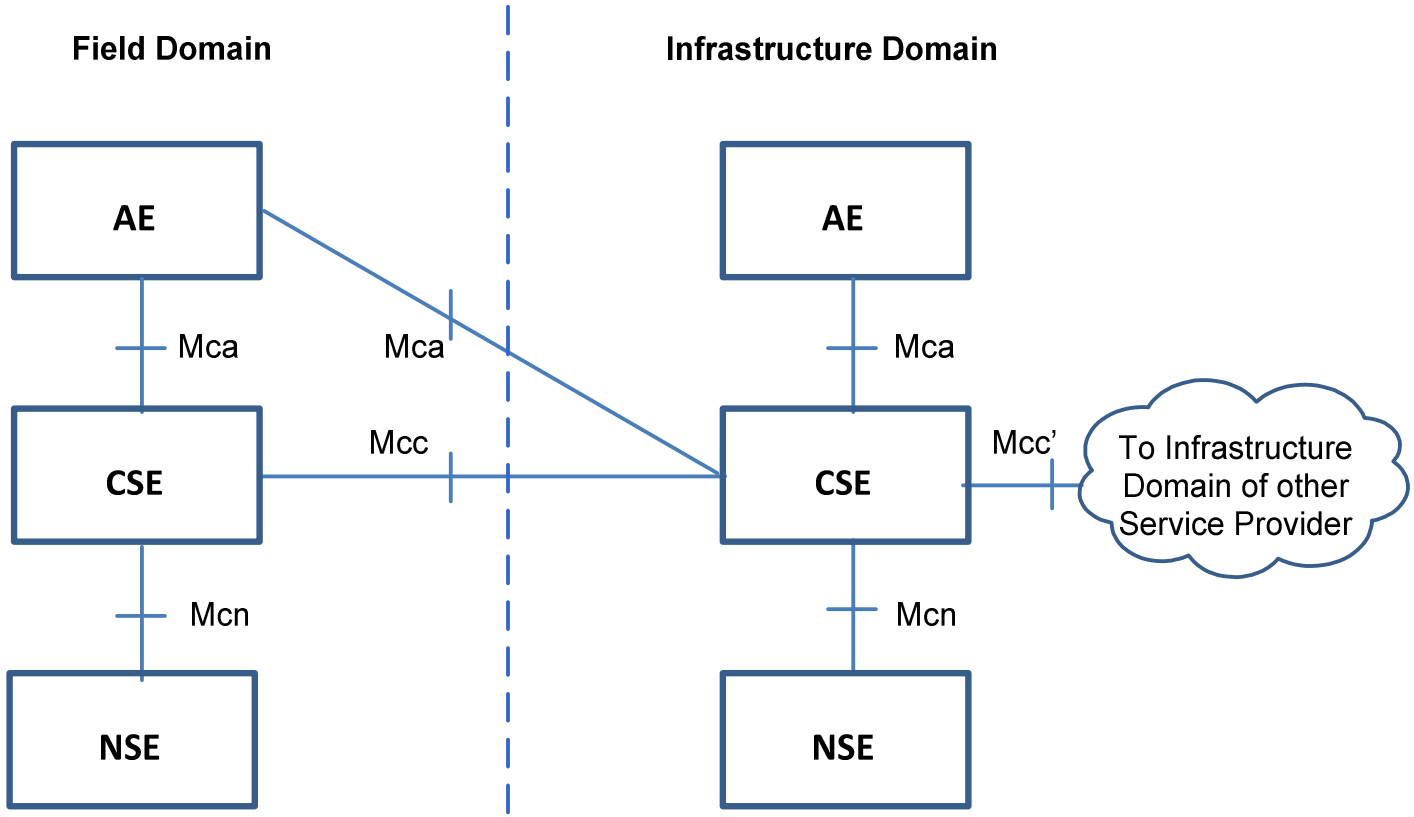
oneM2M Architecture Model

- General Concepts
  - oneM2M layered model which supports end-to-end M2M services
  - Application Layer, Common Service Layer, Network Service Layer



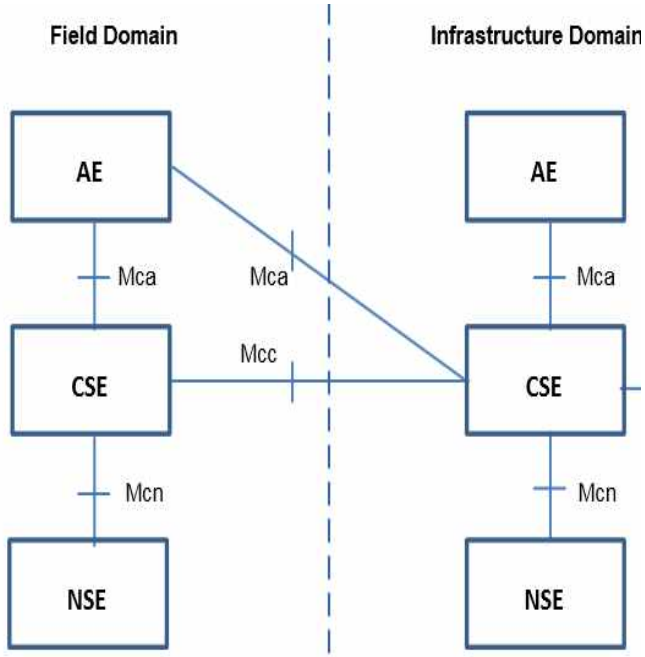
oneM2M Architecture Model

- Functional Architecture



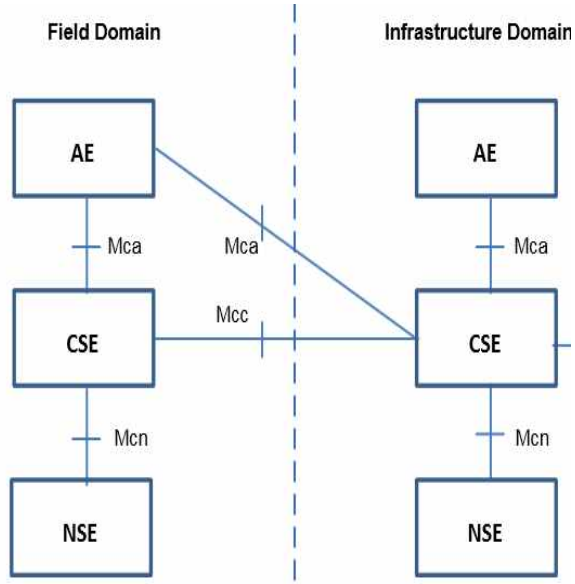
oneM2M Architecture Model

- Functional Entity
  - Application Entity (AE)
    - Application logic of the M2M solution
    - Identified with a unique AE-ID
    - e.g., fleet tracking application or power metering application instances
  - Common Services Entity (CSE)
    - Set of common service functions
    - Service functions are exposes to other entities through reference points
    - Identified with a unique CSE-ID
  - Network Services Entity (NSE)
    - Provide the underlying network service to the CSEs
    - e.g., device management, location services and device triggering services



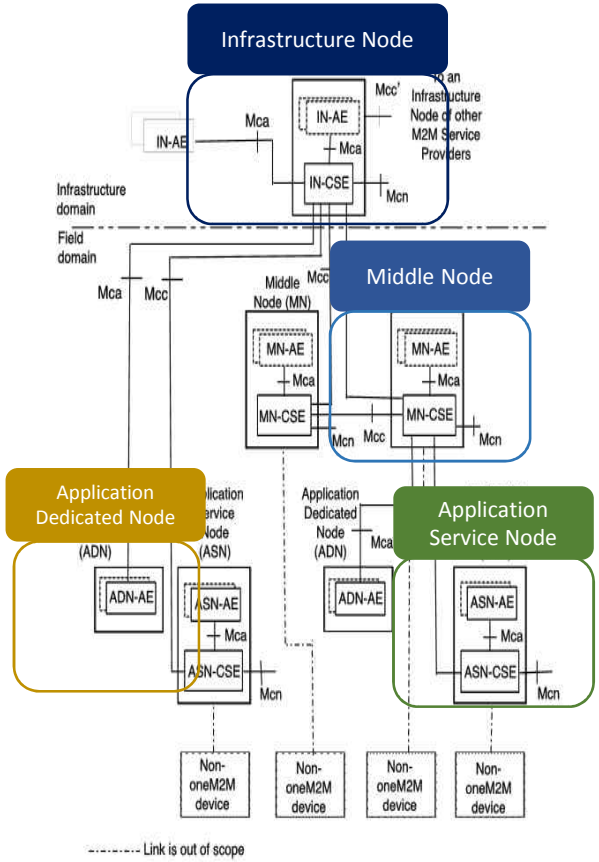
oneM2M Architecture Model

- Reference Points
  - Mca Reference Point
    - Communication flow point between AE and CSE
    - Enabling the AE to use the services supported by the CSE, and for the CSE to communicate with the AE
  - Mcc Reference Point
    - Communication flow point between two CSEs
    - Enabling a CSE to use the services supported by another CSE
  - Mcn Reference Point
    - Communication flow point between CSE and NSE
    - Enabling the CSE to use the supported services provided by the NSE
  - Mcc' Reference Point
    - Communication flow point between two CSEs in infrastructure nodes of different M2M SP domains
    - Enabling inter-M2M service provider communications



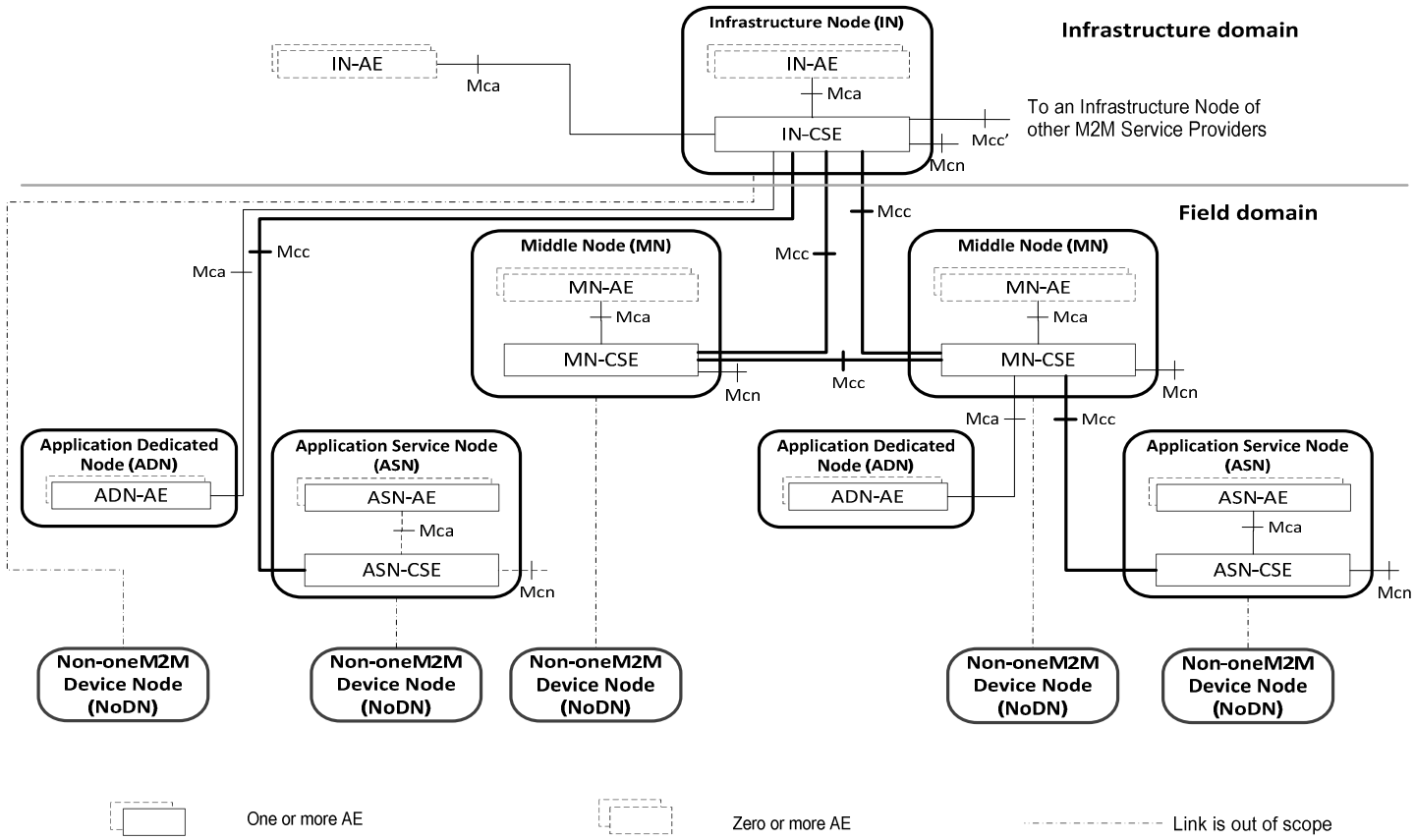
oneM2M Architecture Configuration

- oneM2M Node Types
  - Infrastructure Node (IN)
    - Contain one CSE and zero ore more AE
    - There is one logical Infrastructure Node in the Infrastructure Domain per oneM2M Service Provider of an M2M System
  - Middle Node (MN)
    - Contain one CSE and zero or more AE
    - There may be zero or more MNs in the Field Domain
  - Application Service Node (ASN)
    - Contain one CSE and at least one AE
    - There may be zero or more ASNs in the Field Domain
  - Application Dedicated Node (ADN)
    - Contain at least one AE and does not contain a CSE
    - There may be zero or more ADNs in the Field Domain



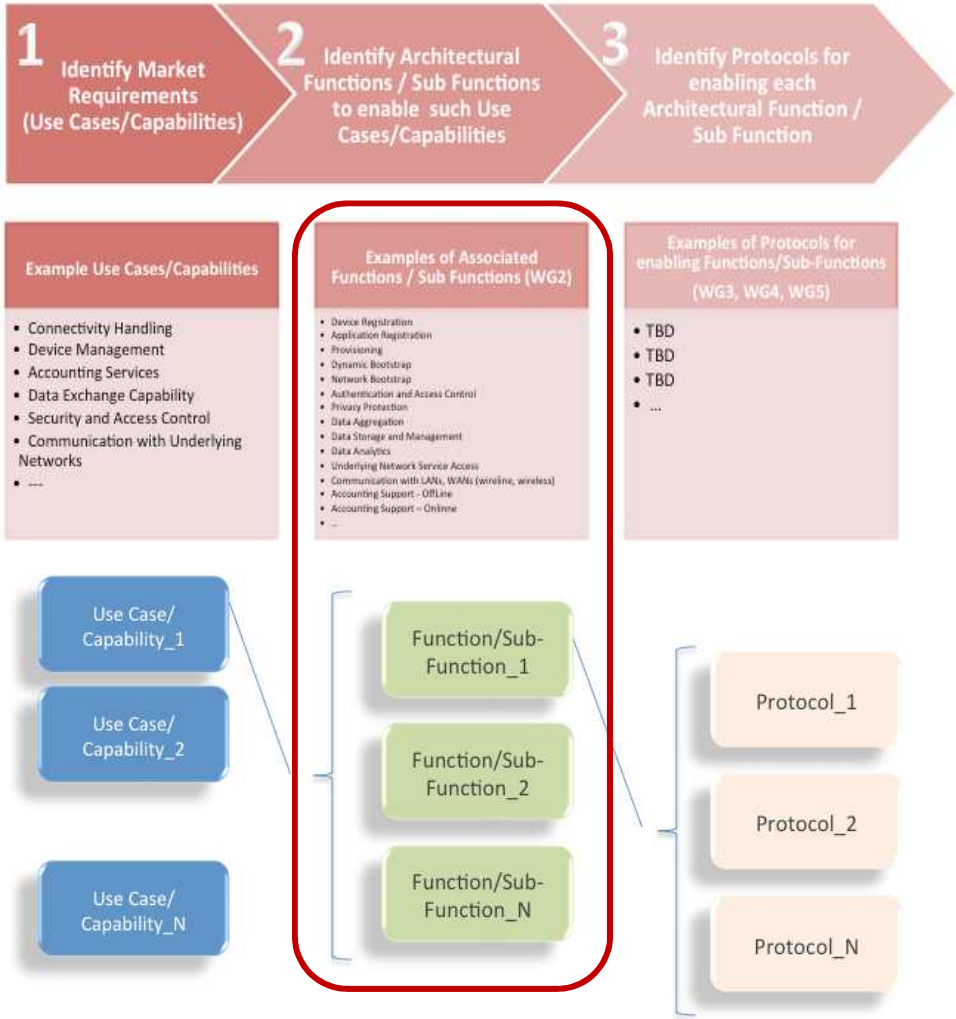
oneM2M Architecture Configuration

- Relationships among oneM2M Entities



Architecture capabilities & functions

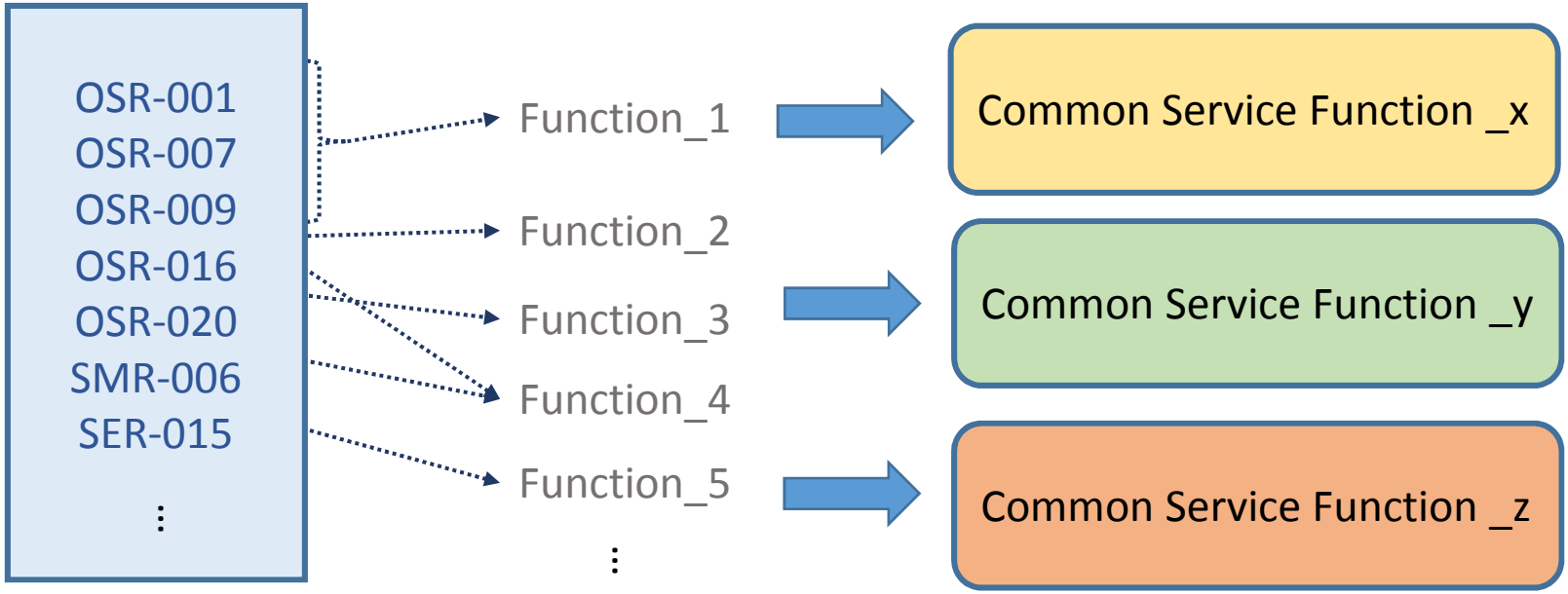
oneM2M Architectural Function



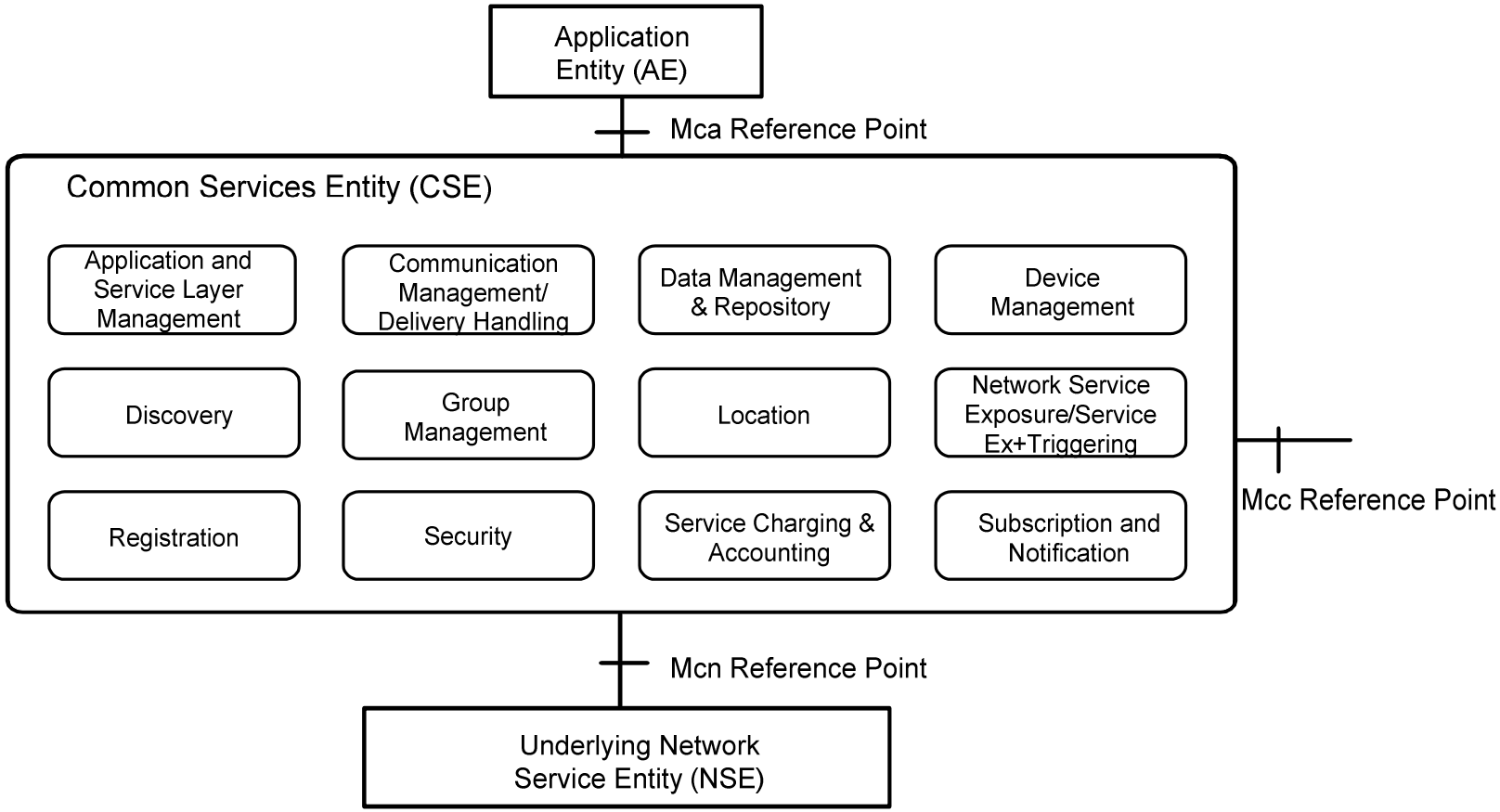
- Stage 1
  - Defines the use cases and associated requirements for the use cases/capabilities
- Stage 2
  - Define Functions/Sub-Functions for enabling the associated requirements and capabilities
  - Define resource structure and information-flows
- Stage 3
  - Defines the Protocols necessary for enabling Functions/Sub-Functions

Common Services Functions

- Each Common Service Function include Sub-Functions consists of 1 to N numbers of functions
- Each CSF can be one realization of a oneM2M architecture service capability or in order to realize one capability, some CSF can be cooperated

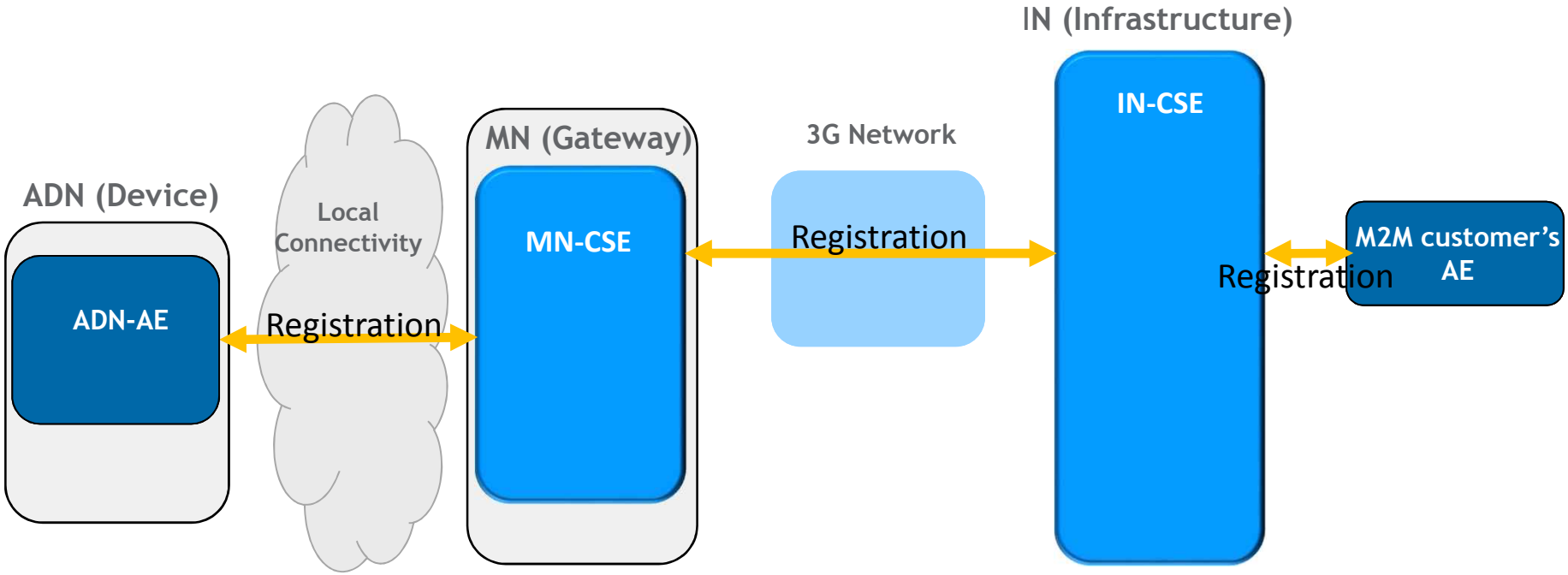


Common Services Functions



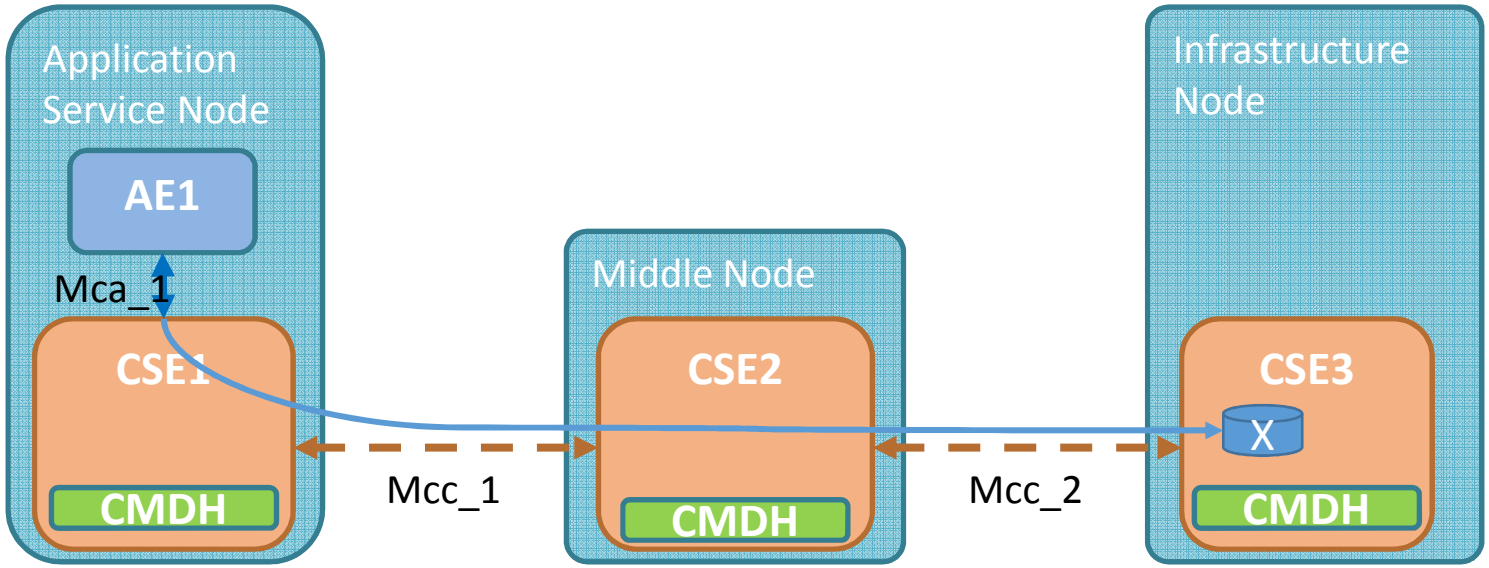
Common Services Functions

- Registration (REG)
  - The REG CSF is responsible for handling AE or another CSE to register with a CSE in order to allow the registered entities to use the services offered by the registered-with CSE
  - AE/CSE should register to enable communication and data access between other CSEs



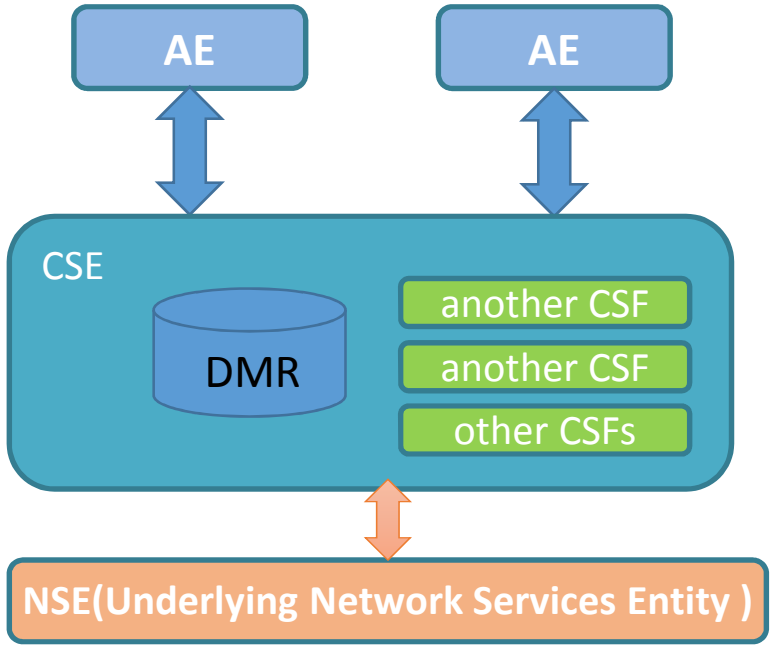
Common Services Functions

- Communication Management and Delivery Handling (CMDH)
  - Providing communication with other CSE, AEs, and NSEs
  - Deciding what time to use which communication connection for delivering messages and to buffer request so that they can be forwarded at a later time



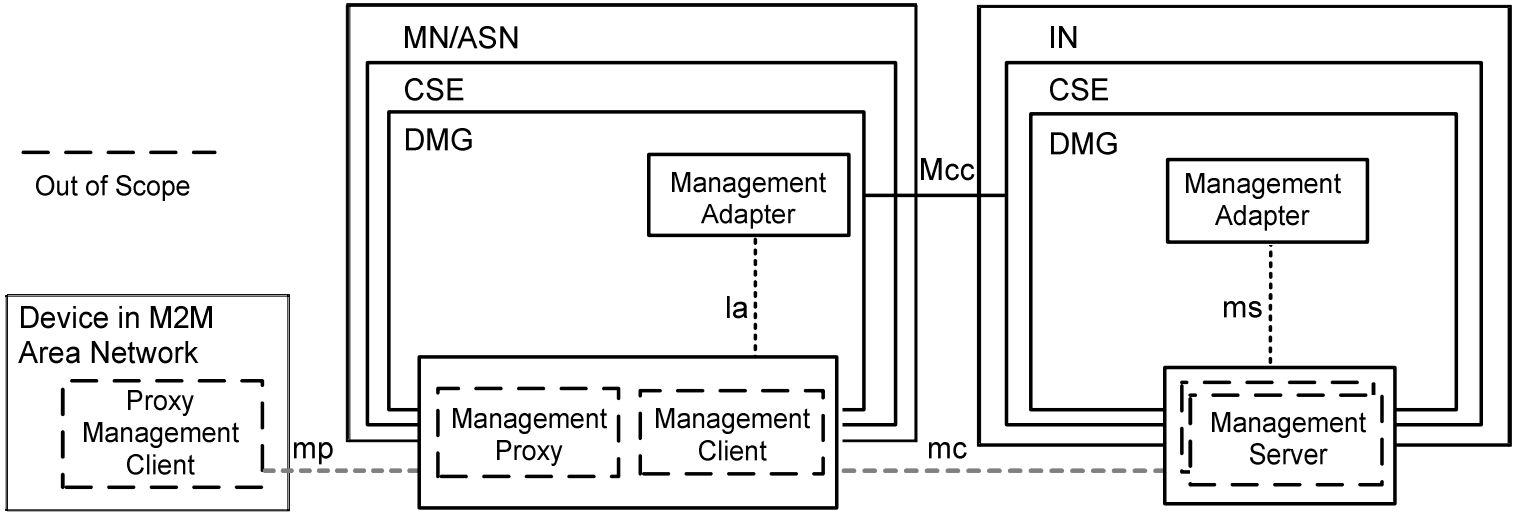
Common Services Functions

- Data Management and Repository (DMR)
  - Collecting data for aggregating large amounts of data, converting this data into a specified format, and storing it for analytics and semantic processing
  - DMR CSF functionalities:
    - Data Repository
    - Grand access to the data
    - Organizing the data
    - Aggregating the data
    - Perform data analytics
    - Semantic data management



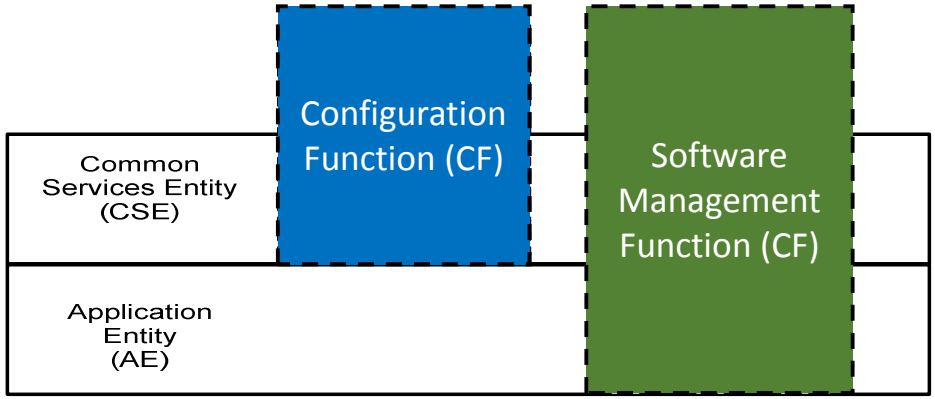
Common Services Functions

- Device Management (DMG)
  - Provide management of device capabilities on MNs, ASNs and ADNs, as well as devices that reside within an M2M Area Network
  - In order to manage the CSE and device capabilities of the MNs, ASNs and ADNs, the DMG can utilize existing device management technologies (e.g. BBF TR-069, OMA-DM, and LWM2M)



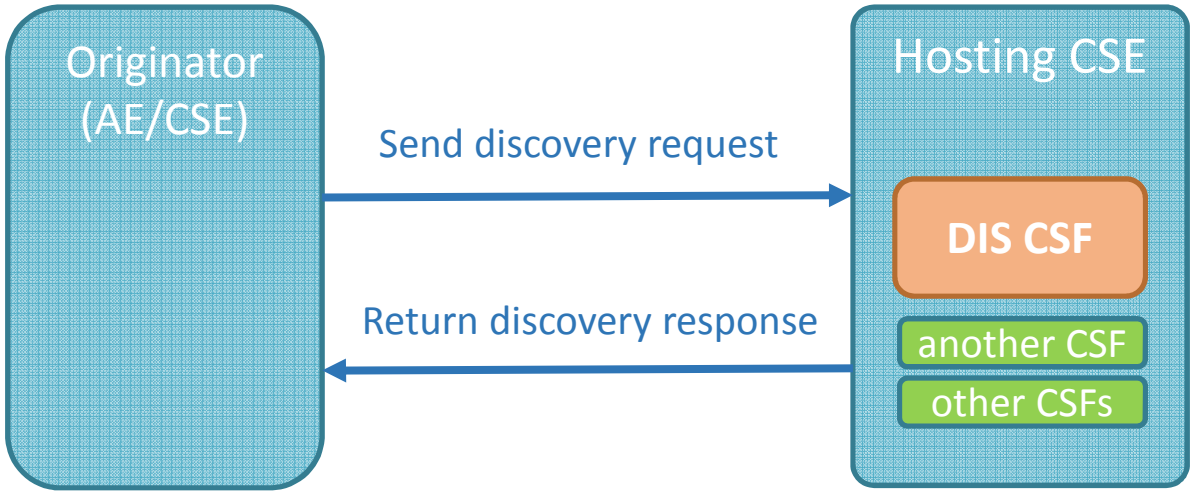
Common Services Functions

- Application and Service Layer Management (ASM)
  - Provides management of the AEs and CSEs of Application, Device, Middle, and Infrastructure nodes
  - This includes capabilities to configure, troubleshoot and upgrade the functions of the CSE, as well as to upgrade the AEs
  - Configuration Function (CF)
    - Enables the configuration of the capabilities and features CSE provided
  - Software Management Function (SMF)
    - Provides lifecycle management for software components and associated artifacts (e.g. configuration files) for different entities such as CSE and AE



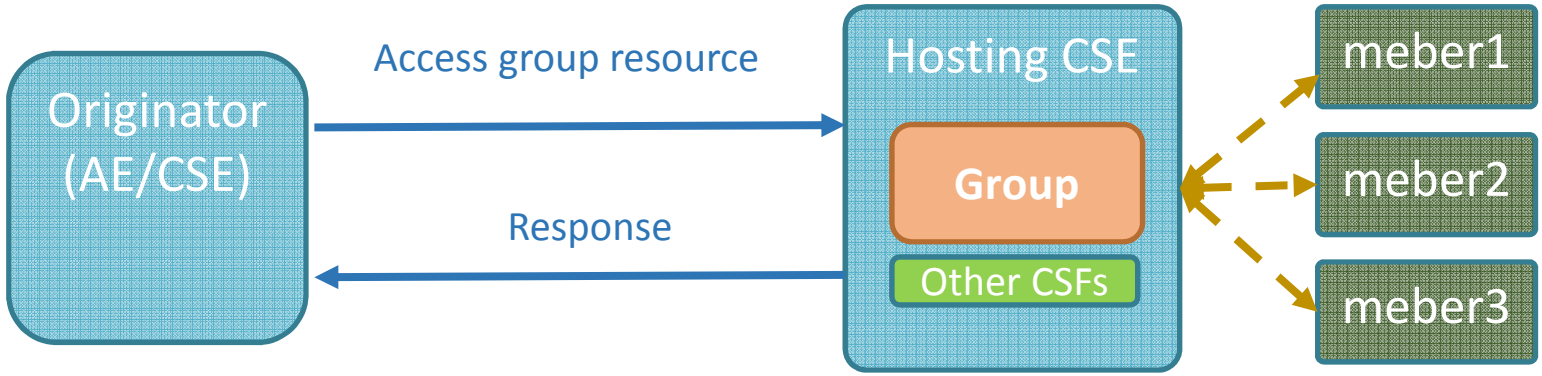
Common Services Functions

- Discovery (DIS)
  - Searching information about applications and services as contained in attributes and resources
  - DIS CSF uses the Originator provided *filter criteria* which describe the rules for resource discovery e.g., creation time, matching string, upper limit, and ordering parameter for the searching result



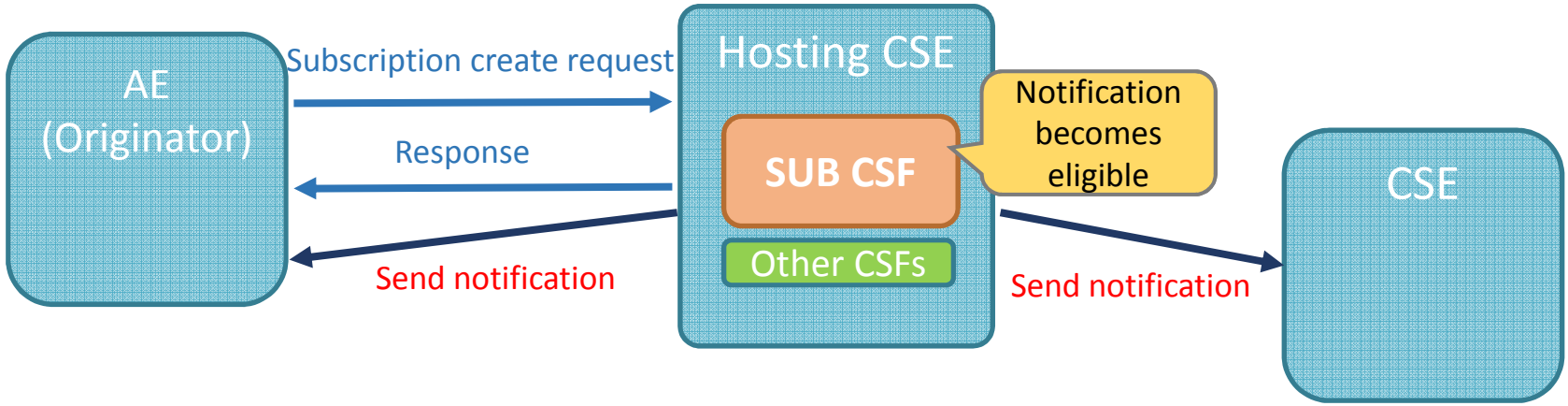
Common Services Functions

- Group Management (GMG)
  - The Group Management (GMG) CSF is responsible for handling group related requests.
  - The request is sent to manage a group and its membership as well as for fan-out common operations supported by the group.
  - Handles the requests to retrieve the information (e.g. URI, metadata, etc.) of a group and its associated members and Forwards requests to all members in the group



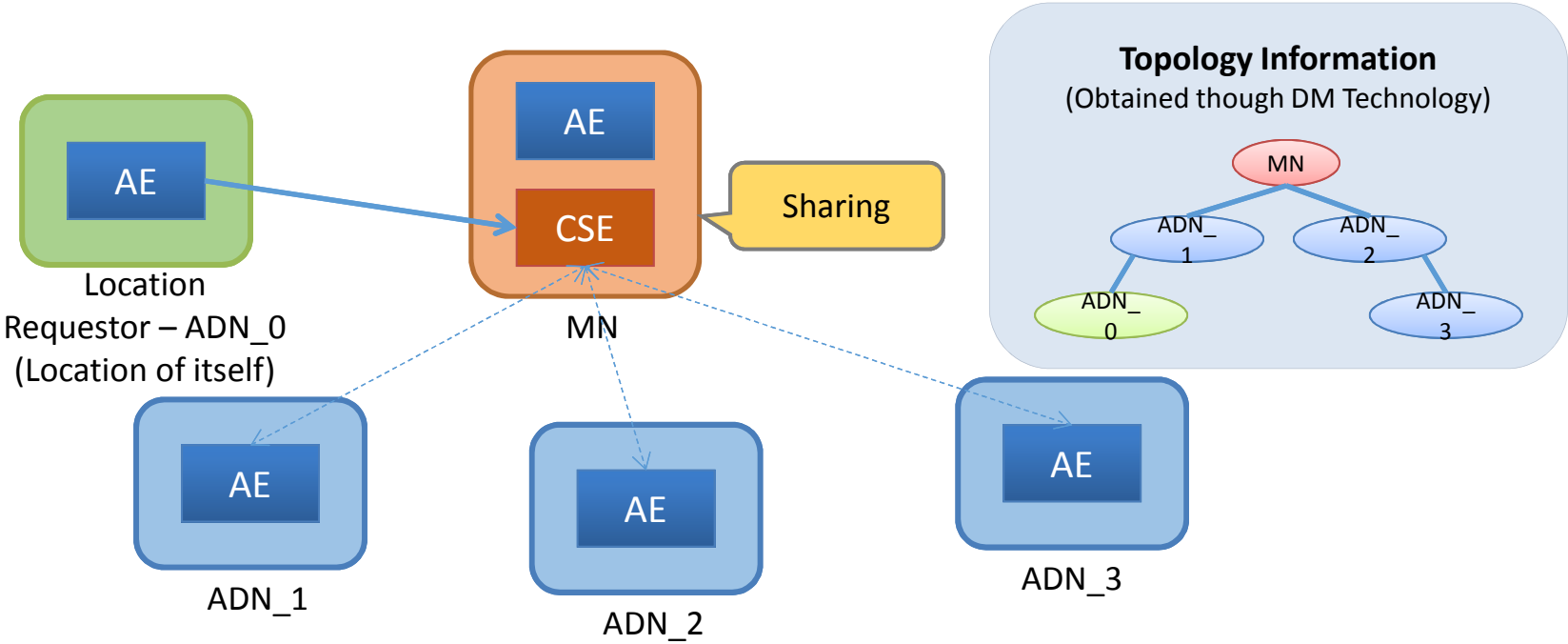
Common Services Functions

- Subscription and Notification (SUB)
  - Provide notifications pertaining to a subscription that tracks changes on a resource (e.g., deletion of a resource)
  - The scope of a resource subscription includes tracking changes of attribute(s) and direct child resource(s) of the subscribed-to resource
  - Each subscription may include notification policies that specify which, when, and how notifications are sent. These notification policies may work in conjunction with CMDH policies



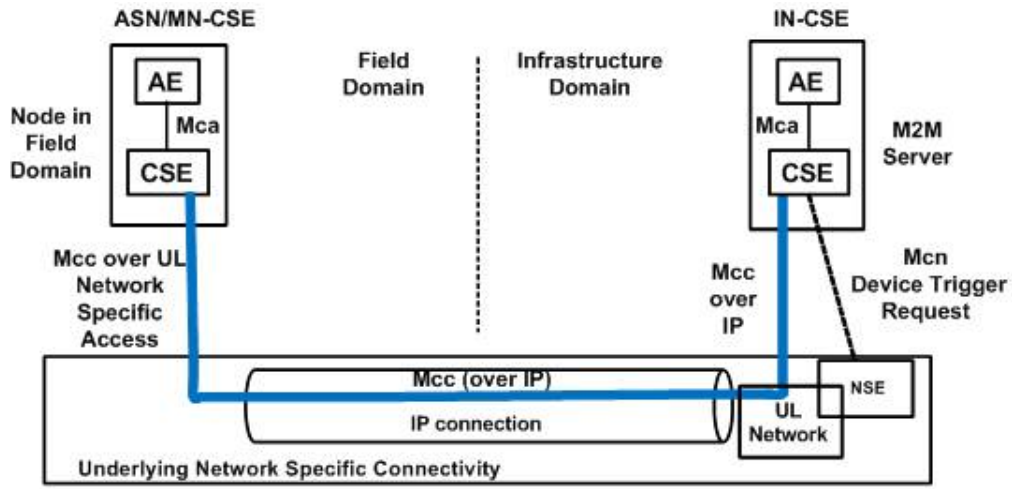
Common Services Functions

- Location (LOC)
  - Allowing AEs to obtain geographical location information of Nodes (e.g., ASN, MN) for location-based services
  - LOC CSF interacts with any of the following
    - Network-based method
    - Device-based method
    - Sharing-based method



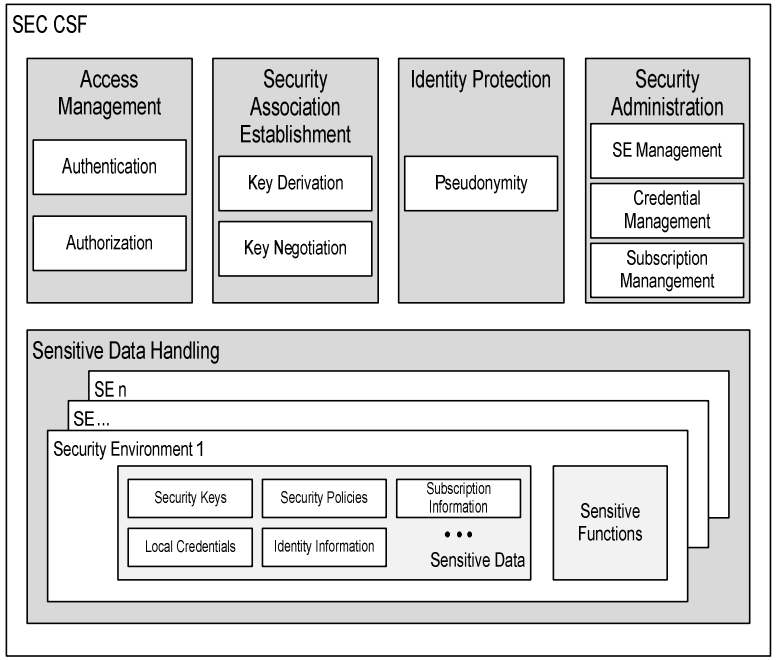
Common Services Functions

- Network Service Exposure, Service Execution and Triggering (NSSE)
  - Manage communications with the underlying network for accessing network service functions over the Mcn reference point
  - NSSE CSF supported functions
    - Device triggering mechanism
    - Shields the network interface difference among underlying networks
    - Provide information for other CSFs that need to use Underlying network service



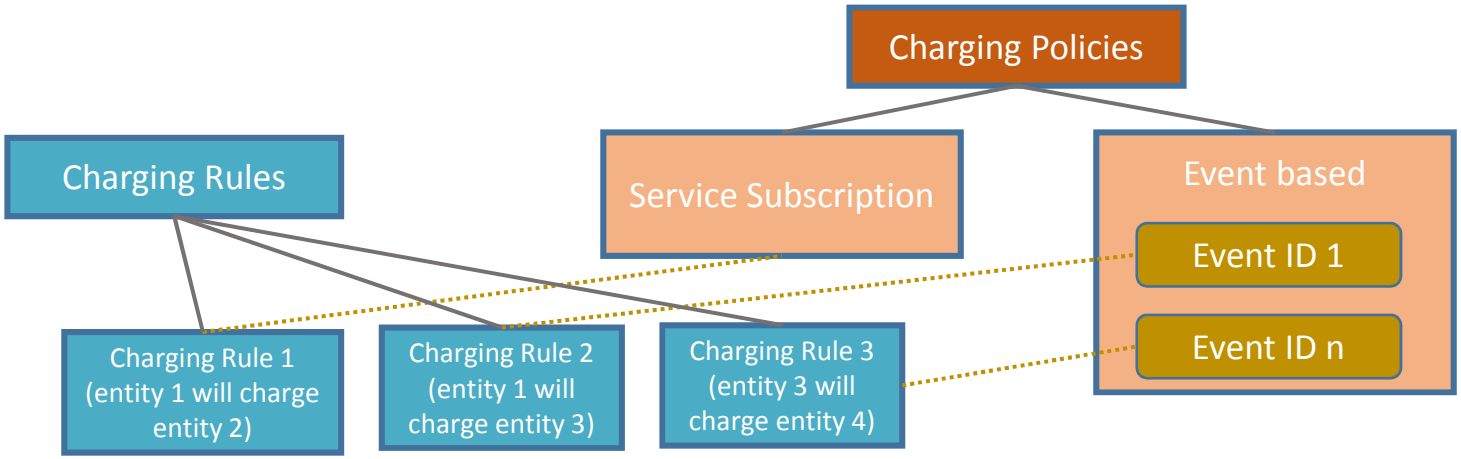
Common Services Functions

- Security (SEC)
  - Identification and Authentication
    - Identification: checking if the identity of the request originator provided for authentication is valid
    - Authentication: validating if the identity supplied in the identification step is associated with a trustworthy credential
  - Security Association Establishment
    - Establishment of a security context between communicating entities to provide confidentiality (encryption) and integrity
  - Authorization (Access Control)
    - Authorizing services and data access to authenticated entities
  - Identity protection
  - Sensitive data handling
  - Security administration (related to device management)



Common Services Functions

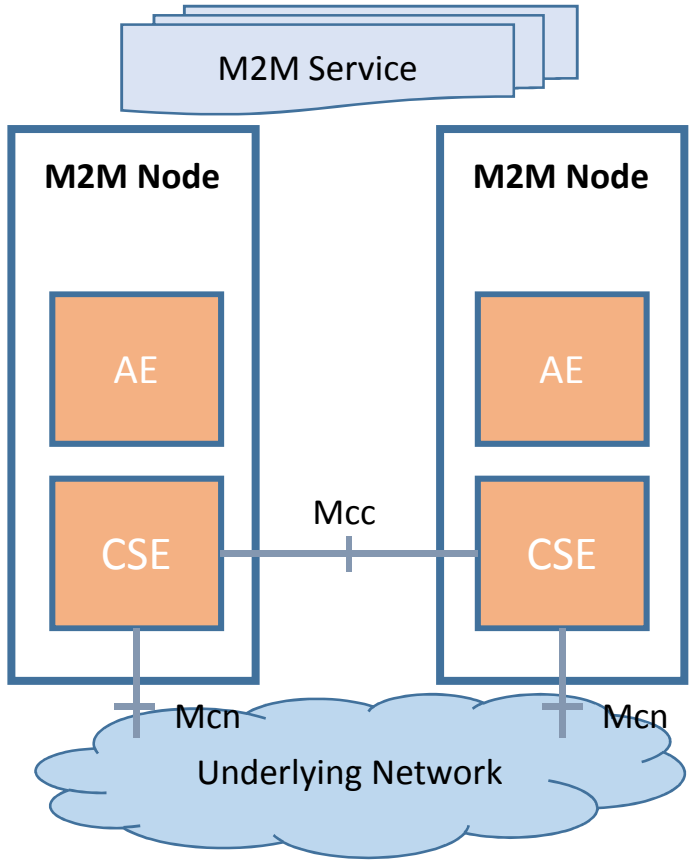
- Service Charging and Accounting (SCA)
  - The SCA CSF provides charging functions for the Service Layer
  - Manages service layer charging policies and configuration capturing service layer chargeable events, generating charging records and charging information
  - Provides re-using charging function in underlying networks (e.g. 3GPP charging)
  - Support multiple charging model
    - Service subscription based: service provider charges service subscribers, e.g., monthly fee
    - Event based: Charging is based on chargeable events



Identification & Message Flow

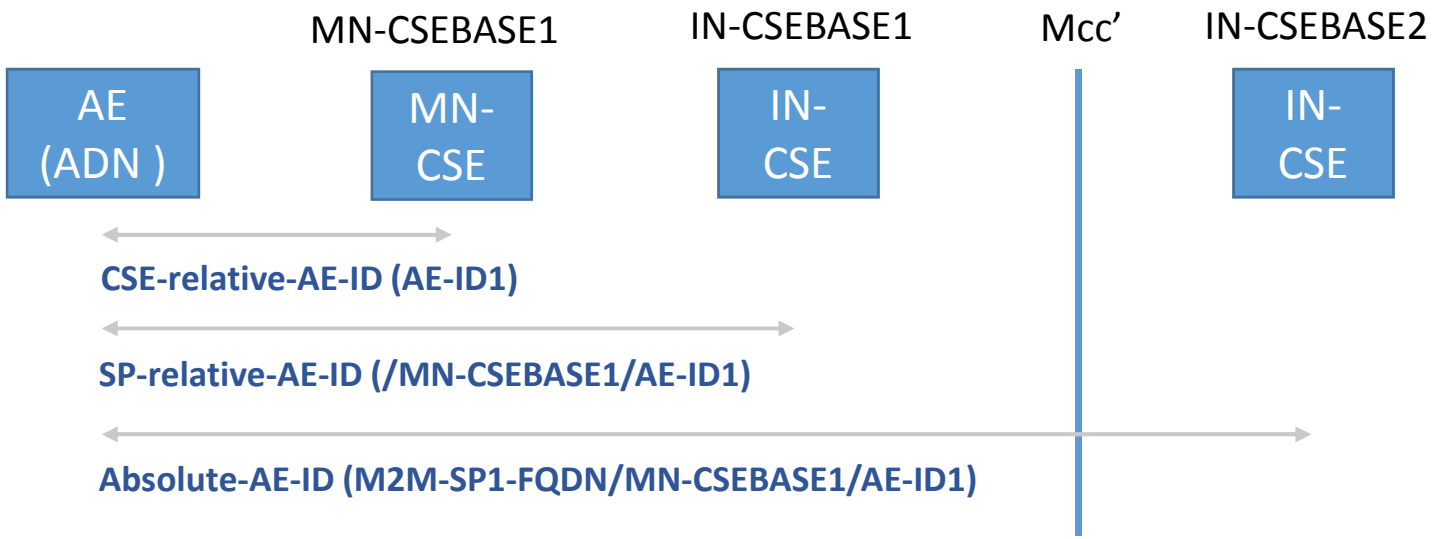
M2M Identifiers

Entity ID	Service ID	Network ID
Application Entity ID (AE-ID)	M2M Service Subscription ID (M2M-Sub-ID)	M2M Request ID (M2M-Request-ID)
CSE ID (CSE-ID)	M2M Service ID (M2M-Serv-ID)	M2M External ID (M2M-Ext-ID)
M2M Service Provider ID (M2M-SP-ID)		Trigger Recipient ID (Trigger-Recipient-ID)
M2M Node ID (M2M-Node-ID)		



M2M Identifiers

- An M2M identifier is a sequence of characters used to refer to a an entity (such as CSE, AE, …), a resource or an object (such as an M2M Service Provider or an M2M Node) defined in oneM2M
- An M2M identifier can be either:
  - Absolute M2M identifier, in which case it is globally unique and shall be a Unified Resource Identifier (URI) as specified
  - Relative M2M identifier, in which case it shall be understood in a particular context



M2M Identifiers

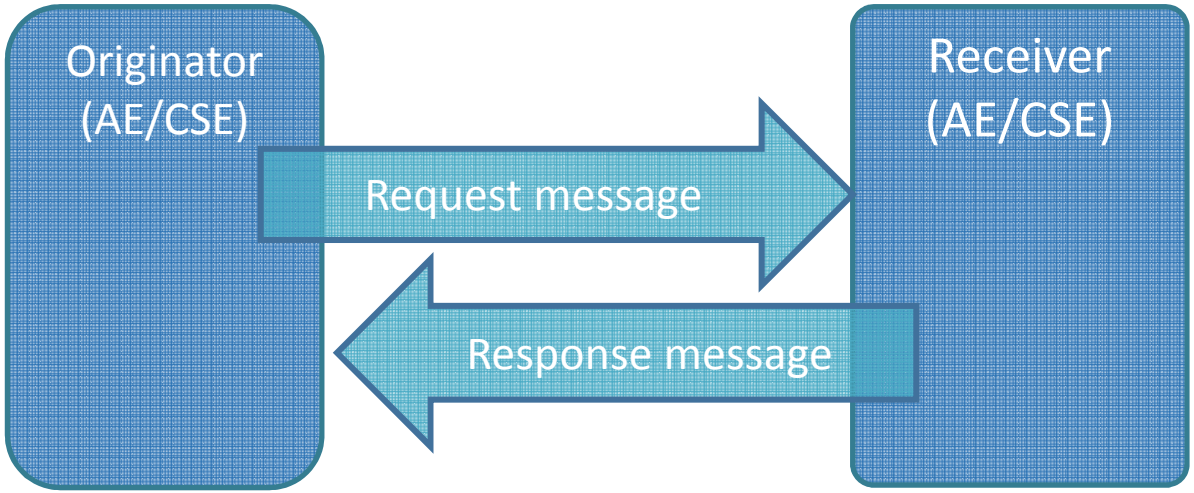
- M2M Service Provider Identifier (M2M-SP-ID)
  - An M2M Service Provider shall be uniquely identified by the M2M Service Provider Identifier (M2M-SP-ID). This is a static value assigned to the Service Provider
- Application Entity Identifier (AE-ID)
  - An Application Entity Identifier (AE-ID) uniquely identifies an AE
  - According to context of AE-ID used, it is sufficient to be unique within the scope
- Application Identifier (App-ID)
  - This is equivalent to the application name and is not guaranteed to be globally unique on its own
- CSE Identifier (CSE-ID)
  - A CSE shall be identified by a globally unique identifier, the CSE-ID, when instantiated within an M2M Node in the M2M System
  - The CSE-ID is globally unique when used internally within a specific M2M SP domain and it is extended to become globally unique outside service provider domain

M2M Identifiers

- M2M Node Identifier (M2M-Node-ID)
  - An M2M Node, hosting a CSE and/or Application(s) shall be identified by a globally unique identifier, the M2M-Node-ID
  - The M2M System shall allow the M2M Service Provider to set the CSE-ID and the M2M-Node-ID to the same value
- M2M Request Identifier (M2M-Request-ID)
  - The M2M-Request-ID tracks a Request initiated by an AE over the Mca reference point, and by a CSE over the Mcc reference point, if applicable, end to end. It is also included in the Response to the Request over the Mca or Mcc reference points
- M2M External Identifier (M2M-Ext-ID)
  - The M2M-Ext-ID is used by an M2M Service Provider (M2M SP) when services targeted to a CSE, identified by a CSE-ID, are requested from the Underlying Network
  - The M2M External Identifier allows the Underlying Network to identify the M2M Device (e.g., ASN, MN) associated with the CSE ID

Description and Flows of Reference Points

- General Communication Flow on Mca and Mcc
  - Entities communicate with each other via pairs of requests and response
  - A request-message triggers a response message
  - Originator: AE or CSE
  - Receivers: CSE (in most cases), AE (notification)
  - The message applies to communications such as:
    - between an AE and a CSE (Mca reference point); and
    - among CSEs (Mcc reference point).



Description and Flows of Reference Points

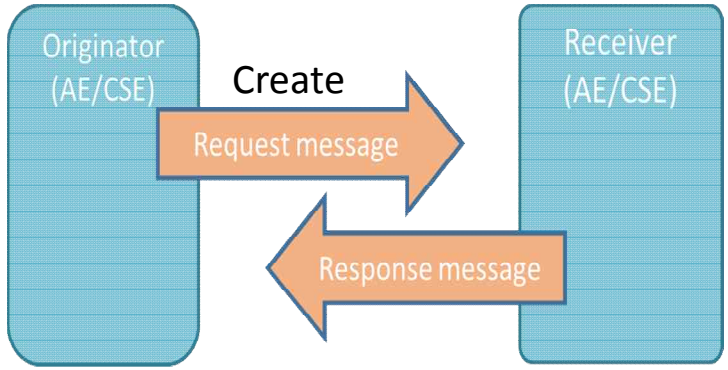
- Summary of Request Message Parameters
  - Mandatory Parameters:
    - ***op***: operation to be executed (create, retrieve, update, delete, notify)
    - ***to***: address of the target resource
    - ***from***: originator ID
    - ***ri***: request identifier
    - ***ty***: the type of the resource to be created (in Create case)
    - ***cn***: resource content to be transferred
  - Optional Parameters:
    - ***nm***: optional name of the resource to be created
    - ***ot***: optional originating timestamp of what the message was built
    - ***rt***: optional response message type
    - ***ec***: event category (related with CMDH CSF), scheduling of connection
    - ***da***: delivery aggregation on/off
    - ***fc***: optional filter criteria

Description and Flows of Reference Points

- Summary of Response Message Parameters
  - Mandatory Parameters:
    - ***rs***: response code (successful, unsuccessful, acknowledgement)
    - ***ri***: request identifier, ***ri*** in the Response shall match the ***ri*** in the corresponding Request
  - Conditional Parameters:
    - ***cn***: resource content
    - If ***rs*** is successful then:
      - **Create**: ***cn*** is the address and/or the content of the created resource
      - **Update**: ***cn*** is the content replaced in an existing resource
      - **Delete:** Optionally, ***cn*** is the content actually deleted
      - **Retrieve:** ***cn*** is the retrieved resource content or aggregated contents of discovered resources
    - If ***rs*** is unsuccessful then:
      - ***cn*** parameter provide more error information

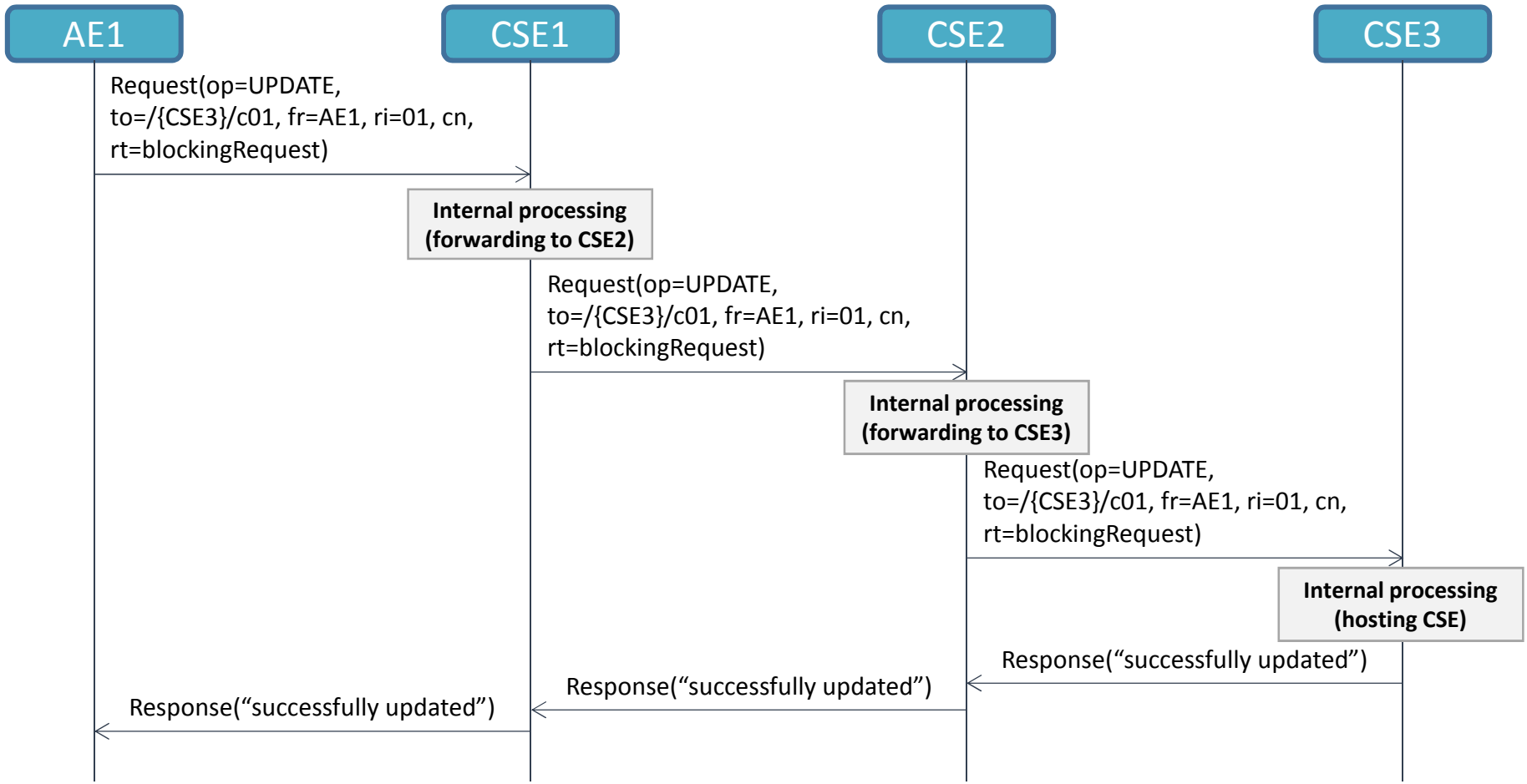
Description and Flows of Reference Points

- Example: Create Case
  - Request
    - ***op***: C (Create)
    - ***to***: address of the target resource
    - ***fr***: Originator ID
    - ***ri***: Request Identifier
    - ***ty***: resource type to be created
    - ***cn***: mandatory attributes for the resource
    - ***nm***: resource name
  - Response
    - ***rs***: result (status codes, etc.)
    - ***ri***: Request Identifier
    - ***cn***: address and content of created resource (if successful)



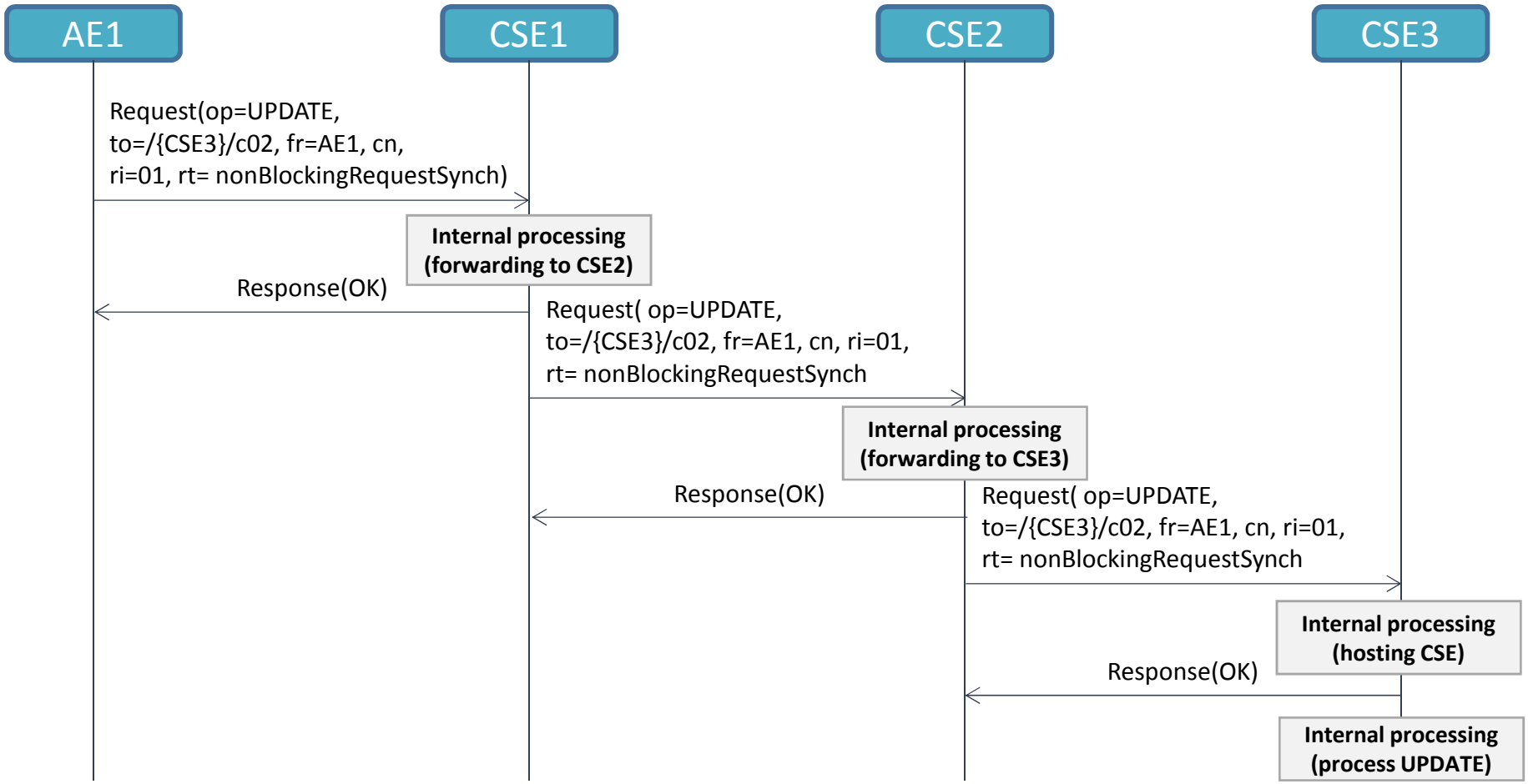
Description and Flows of Reference Points

- Accessing Resources in CSEs – Blocking Requests



Description and Flows of Reference Points

- Accessing Resources in CSEs – nonBlocking Requests



Summary

- oneM2M architecture model consists of 3 functional entities
  - Application Entity (AE)
  - Common Service Entity (CSE)
  - Network Service Entity (NSE)
- Interface between Entities defined as *Mca, Mcc, Mcn, Mcc'*
- According to the configuration with AEs and CSEs in the M2M environments, each node can be defined as *ADN, ASN, MN, IN*
- Common service entity defined 12 CSFs (*common service function*) and each common service function has been developed by *resource types* respectively
- Each resource can be accessed by CRUD operation (*information flow*)
  - RESTful interface

Thank you!