



oneM2M Access Control

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



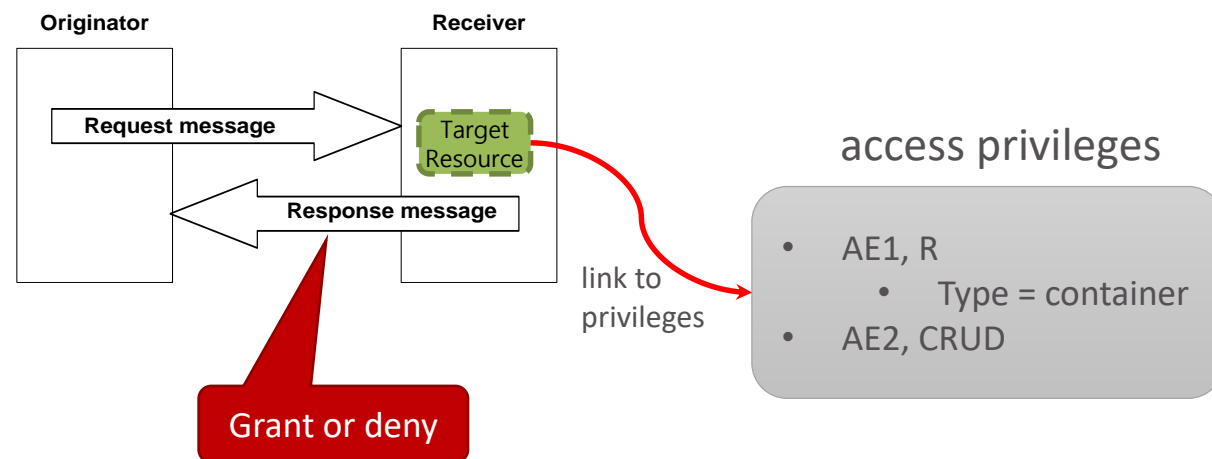
- Access control mechanisms in oneM2M
 - Access control concept
 - Different access control mechanisms
- Access control with <accessControlPolicy> resources
- Practices with <accessControlPolicy> resources



Access control mechanisms

Access control concept

- A oneM2M request is handled by the Receiver when the Originator is allowed to do so
 - Request is generally accessing a resource (C/R/U/D +DISCOVER)
 - Even discovery request primitive is formulated with RETRIEVE operation, different privilege is defined
 - Sometimes request sends a notification (N)
- To make an access decision, there shall be privileges given



Access control concept

- Decision making and enforcement on whether a CSE as a receiver allows an Originator's request to access (CRUDN) a resource on the CSE or not
- Three information is checked as mandatory
 - Who: Originator's Identification (**From** parameter or others)
 - What: Privilege(rights) to access the target resource (**To** parameter)
 - How: Operation (CRUD+N+DIS) to access a resource (**Operation** parameter)
- Basically a set of access privileges is pre-configured in the system and then used for decision making later when a CSE gets a request
 - Static vs. dynamic

Different access control mechanisms



- Access Control Policy
 - Basic mechanism supported in Rel-1 as well as the other releases
 - <accessControlPolicy> resource and accessControlPolicyIDs attribute
- Dynamic Authorization Consultation
 - <dynamicAuthorizationConsultation> resource and dynamicAuthorizationConsultationIDs attribute
- Role
 - <role> resource and Role IDs parameter
- Token
 - <token> resource and Token Request Indicator / Tokens / Token IDs parameters

Different access control mechanisms

- How to handle different mechanisms?
 - Clause 10.2.3 (Authorization) in TS-0001 provides the answer

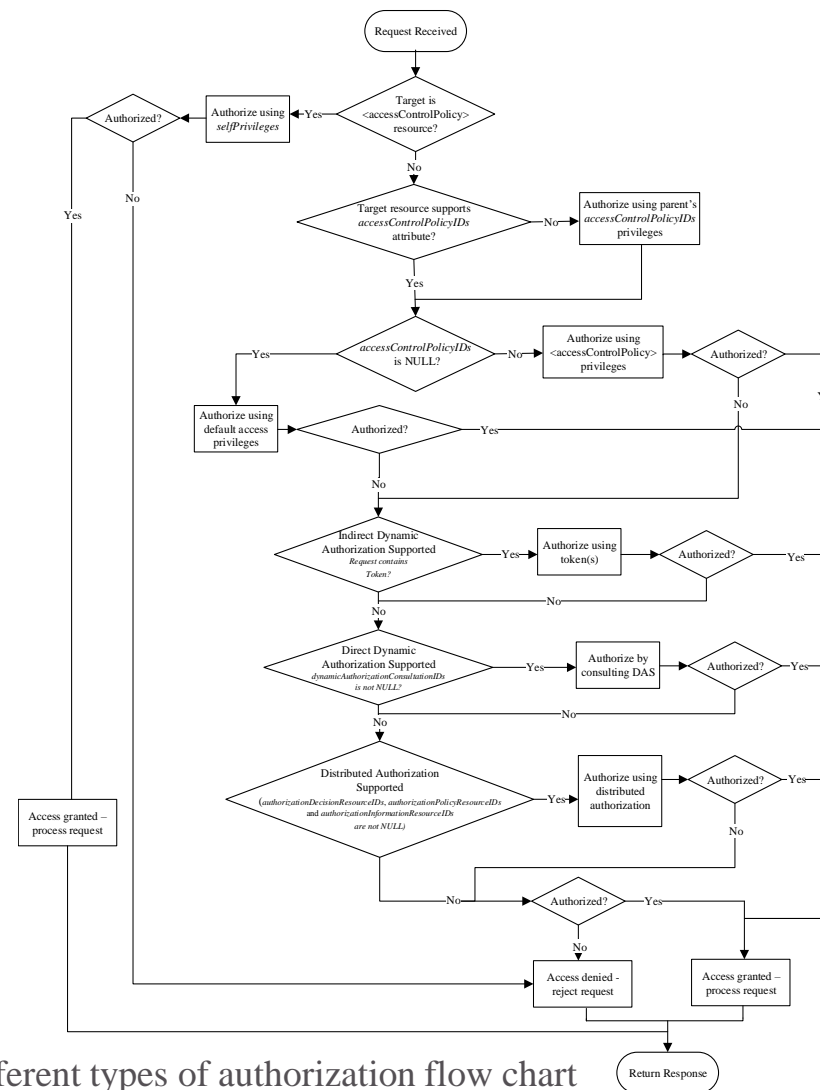


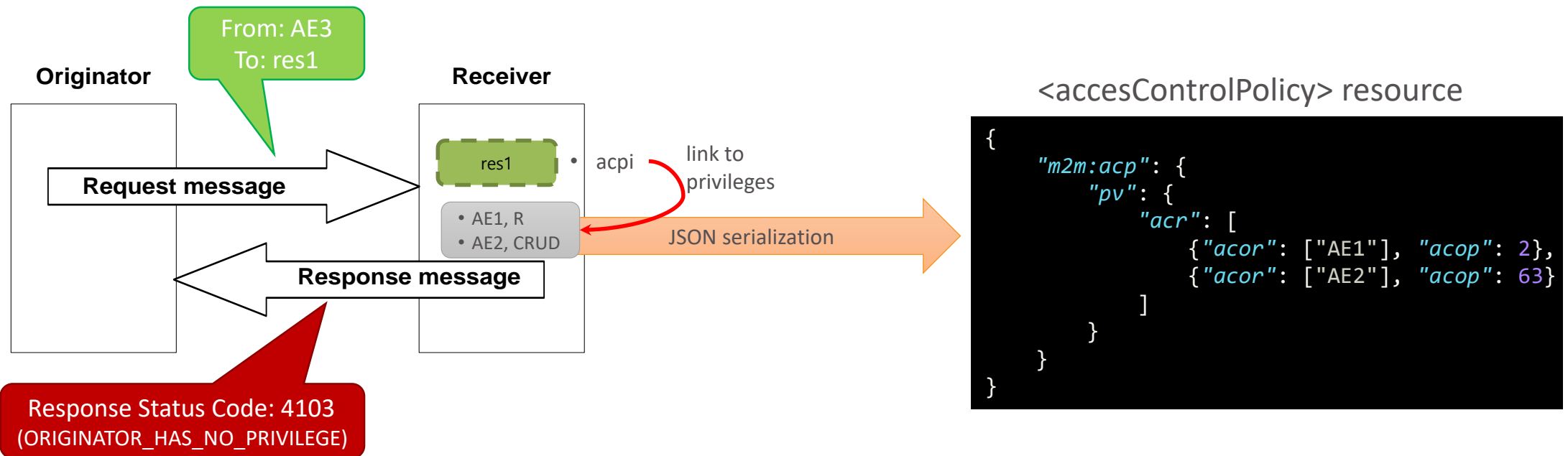
Figure 10.2.3.1-1: Different types of authorization flow chart



**Access control with
<accessControlPolicy> resources**

accessControlPolicy resource type

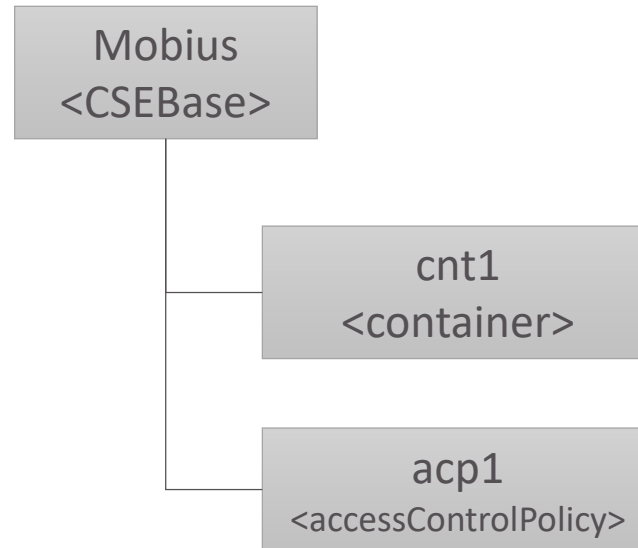
- <accessControlPolicy> resource
 - A resource (request target) links <accessControlPolicy> resource(s) in *accessControlPolicyIDs(acpi)* attribute
 - To make an access decision, the CSE looks into the linked <ACP> resource(s) which has access privileges given by the resource owners



accessControlPolicy resource type

- Example

- CAE1 retrieves “cnt1”
- CAE3 retrieves “cnt1”



```
{
  "m2m:cnt": {
    "rn": "cnt1",
    "acpi": ["Mobius/acp1"]
  }
}
```

```
{
  "m2m:acp": {
    "rn": "acp1",
    "pv": {
      "acr": [
        {"acor": ["CAE1"], "acop": 2},
        {"acor": ["CAE2"], "acop": 63}
      ]
    }
  }
}
```

<accessControlPolicy> resource

- Privileges(pv) have accessControlRules(acr)
 - Each accessControlRule consists of
 - accessControlOriginator (acor) and
 - accessControlOperator (acop)

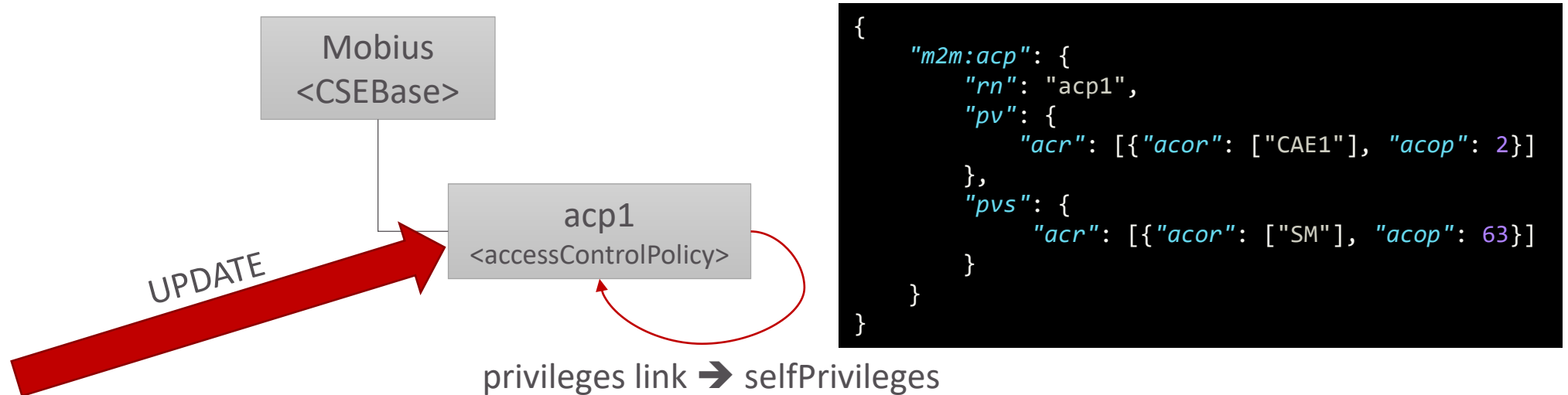
Table 6.3.4.2.29-1: Interpretation of accessControlOperations (TS-0004)

Value	Interpretation	Note
1	CREATE	
2	RETRIEVE	
4	UPDATE	
8	DELETE	
16	NOTIFY	
32	DISCOVERY	
NOTE: Combinations of these values are specified by adding them together. For example the value 5 is interpreted as "CREATE and UPDATE".		

```
{
  "m2m:acp": {
    "rn": "acp1",
    "pv": {
      "acr": [
        {"acor": ["CAE1"], "acop": 2},
        {"acor": ["CAE2"], "acop": 63}
      ]
    }
  }
}
```

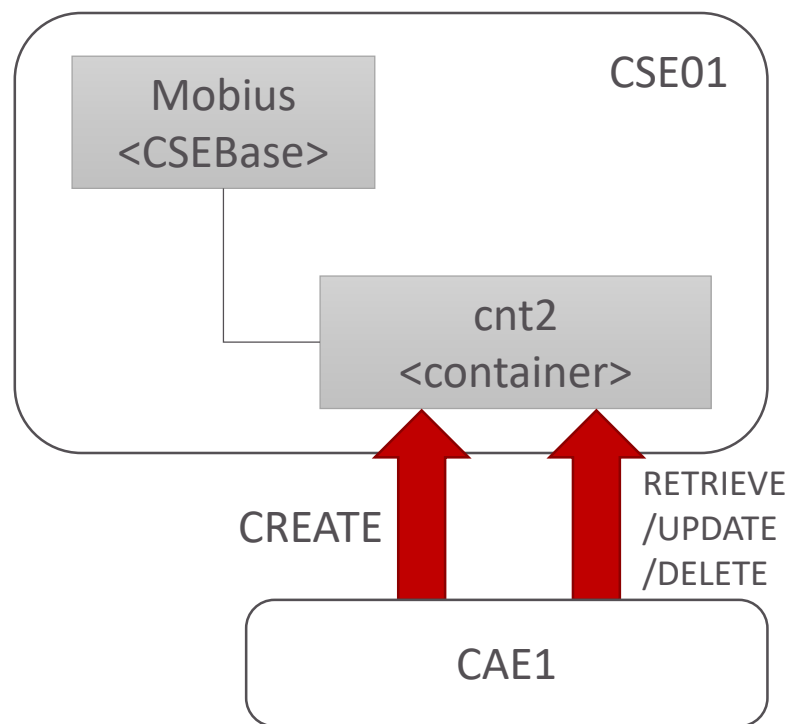
accessControlPolicyIDs attribute

- Not all resource types have(define) *acpi* attribute
- *e.g. contentInstance* resource type
 - apply parent, which is <container> resource, access privileges
- *i.e. accessControlPolicy* resource type
 - use *selfPrivileges(pvs)* of the <accessControlPolicy> resource



accessControlPolicyIDs attribute

- Some resource instances don't have *acpi* attribute even the type defines it since it's optional
- The creator of the resource has all privileges



```
{  
  "m2m:cnt": {  
    "rn": "cnt2",  
    "lbl": ["hello"]  
  }  
}
```



Practices with <accessControlPolicy> resources

Add access control to Day 1

- Not anyone can control my buzzer
 - Set access privileges to the buzzer control resource
 - Confirm whether my home IoT application can access to the resource

