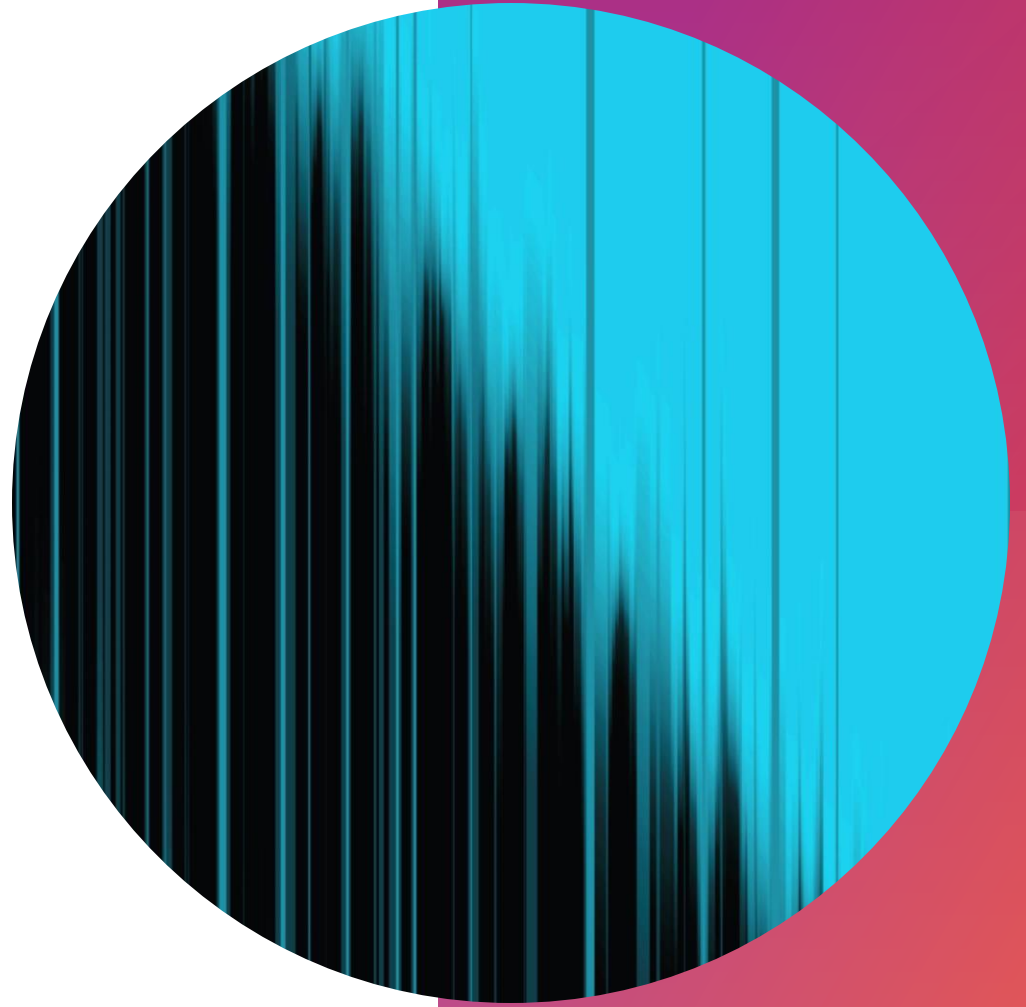


oneM2M

TinyIoT

ONEM2M TINYIOT SERVER

엄경호



## 이번 주 진행 상황

- I. ACME 테스트 기반 보완
- II. Action Triggering 조사

# testCNT.py

[ACME] - Test Results							
Test Suite	Count	Skipped	Errors	Times Exec   Sleep   Proc	Exec Time per Test   Request	Proc Time per Test   Request	Requests
testAE	16	0	0	1.7448   0.00   0.2614	0.1090   0.0759	0.0163   0.0114	23
Totals	16	0	0	1.7535   0.00   0.2701	0.1096   0.0762	0.0169   0.0117	23

지난 주 testAE.py  
25개 중 16개 성공

[ACME] - Test Results							
Test Suite	Count	Skipped	Errors	Times Exec   Sleep   Proc	Exec Time per Test   Request	Proc Time per Test   Request	Requests
testCNT	17	0	0	1.5187   0.00   0.2220	0.0893   0.0759	0.0131   0.0111	20
Totals	17	0	0	1.5277   0.00   0.2310	0.0899   0.0764	0.0136   0.0115	20

이번 주 testCNT.py  
22개 중 17개 성공

## testCNT.py

```
1  [
2      "m2m:cnt": {
3          "rn": "testCNT1",
4          "ty": 3,
5          "pi": "CAE",
6          "ri": "3-20230118T032820183",
7          "ct": "20230118T032820",
8          "lt": "20230118T032820",
9          "st": 0,
10         "et": "20250117T032820",
11         "cni": 0,
12         "cbs": 0,
13         "lbl": [
14             "key1",
15             "key2"
16         ]
17     }
18 ]
```

lbl 속성 추가 완료

기존 함수들은 partial update에 적합하지 않아서  
재작성함

lbl은 어떤 리소스를 특정 짓는 데에 도움을 주는  
키 역할로 주로 사용되는 속성임

# testCNT.py

```
#addTest(suite, TestCNT('test_updateCNT'))
addTest(suite, TestCNT('test_updateCNTTy'))
addTest(suite, TestCNT('test_updateCNTempty'))
addTest(suite, TestCNT('test_updateCNTpi'))
#addTest(suite, TestCNT('test_updateCNTUnknownAttribute'))
#addTest(suite, TestCNT('test_updateCNTWrongMNI'))
```



{"m2m:cnt": {"lbl": ["aTag"], "mni": 10, "mbs": 9999}}

{"m2m:cnt": {"unknown": "unknown"}}

	create	update	xs:nonNegativeInteger	No default
maxNrOfInstances	O	O	xs:nonNegativeInteger	No default
maxByteSize	O	O	xs:nonNegativeInteger	No default
maxInstanceAge	O	O	xs:nonNegativeInteger	No default
currentNrOfInstances	NP	NP	xs:nonNegativeInteger	No default (This is generated by the Hosting CSE and limited by the maxNrOfInstances)
currentByteSize	NP	NP	xs:nonNegativeInteger	No default (This is generated by the Hosting CSE and limited by the maxByteSize)

[TS-0004] CNT attributes

mni, mbs는 Optional attribute이기에 테스트 불가

NP 뜻?

# testCNT.py



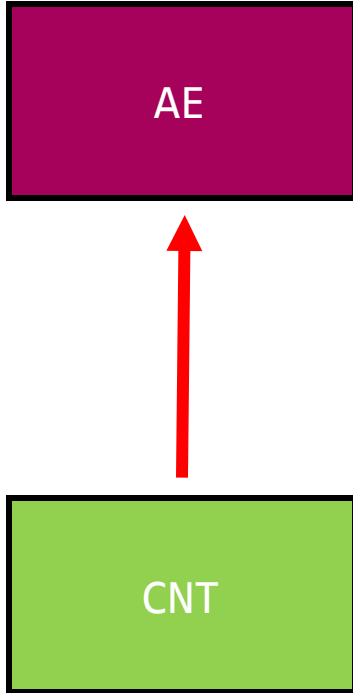
resourceType	1	WO	See clause 9.6.1.3.	NA
resourceID	1	RO	See clause 9.6.1.3.	NA
resourceName	1	WO	See clause 9.6.1.3.	NA
parentID	1	RO	See clause 9.6.1.3.	NA
expirationTime	1	RW	See clause 9.6.1.3.	MA
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3.	MA
labels	0..1 (L)	RW	See clause 9.6.1.3.	MA
creationTime	1	RO	See clause 9.6.1.3.	NA
lastModifiedTime	1	RO	See clause 9.6.1.3.	NA
stateTag	1	RO	See clause 9.6.1.3.	NA
announceTo	0..1 (L)	RW	See clause 9.6.1.3.	NA
announcedAttribute	0..1 (L)	RW	See clause 9.6.1.3.	NA
announceSyncType	0..1	RW	See clause 9.6.1.3.	MA
dynamicAuthorizationConsultationIDs	0..1 (L)	RW	See clause 9.6.1.3.	OA
creator	0..1	RO	See clause 9.6.1.3.	NA

[TS-0001] CNT attributes

```
{
  "m2m:cnt": {
    "ri": "cnt2710740919289509177",
    "pi": "COUEy3IxmPI",
    "rn": "cnt_q3bxMNSPqg",
    "ct": "20230117T204353,303479",
    "lt": "20230117T204353,303479",
    "et": "20280116T204353,303579",
    "ty": 3,
    "cni": 0,
    "cbs": 0,
    "st": 0
  }
}
```

[ACME] CNT 반환 값

testCNT.py



CNT Operation에 대한 Origin 검증은 AE-ID를 기반으로 두고 있음

ACME에서는 AE-ID = AE RI을 동일하게 두는 형태

리소스 트리 상 parent node를 따라가서 권한을 검증하는 식으로 구현

## testCNT.py

```
{"wrong": {"rn": "testCNT"}}
```

기존 JSON Parsing 함수에는 위와 같은 케이스를 고려하지 않았기 때문에  
cJSON을 활용하여 "m2m:ae", "m2m:cnt" 등의 키 값을 파싱하는 함수 구현

```
test_createCNTwithoutOriginator (testCNT.TestCNT)  
Create <CNT> under <CB> without an Originator -> Fail ... ok
```

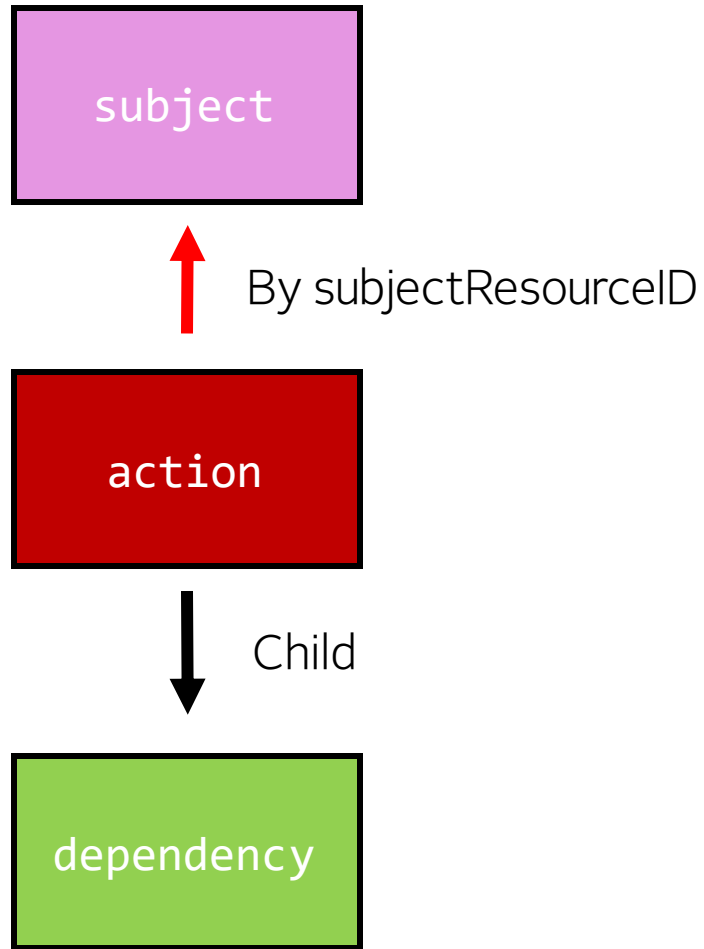
AE의 경우 Create 시 Originator가 없을 때를 고려했지만

CNT의 경우 Originator가 없으면 Create 실패가 되어야 함

Operation, Originator, Type 에 따라 권한에 대한 케이스를 나눔



# Action Triggering



만일 action object가 생성이 되면 CSE는 evalMode attribute를 확인한다.

off -> no monitoring

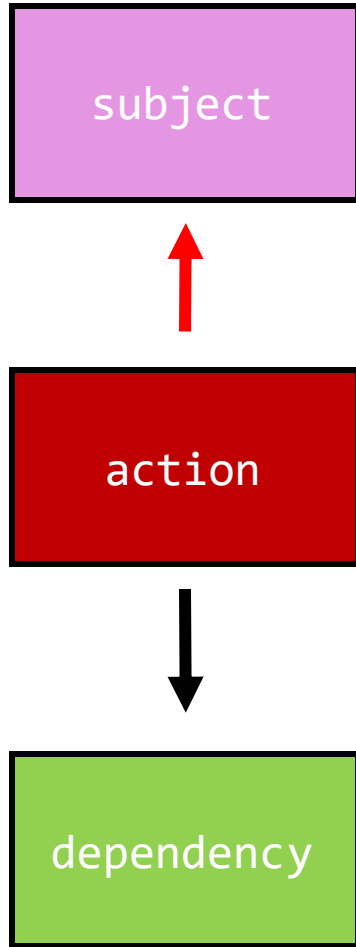
off 외에는 모니터링 실행

once -> 이벤트 한번만 실행

periodic -> 기간 내에 이벤트 한번만 실행

continuos -> evalControlParam attribute 값 만큼 이벤트 실행

# Action Triggering



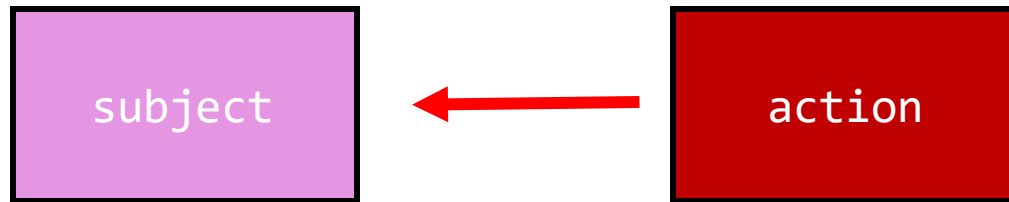
dependency는 좀 더 복잡한 event를 다루기 위해 생성 가능

dependency가 정의하는 조건에 도달했는지 여부를 확인 가능하고,

action triggering 조건에 만족했는지 확인 가능하고

action 리소스에서 어떤 dependency를 우선적으로 고려할지 설정 가능하다.

## Action Triggering 예시

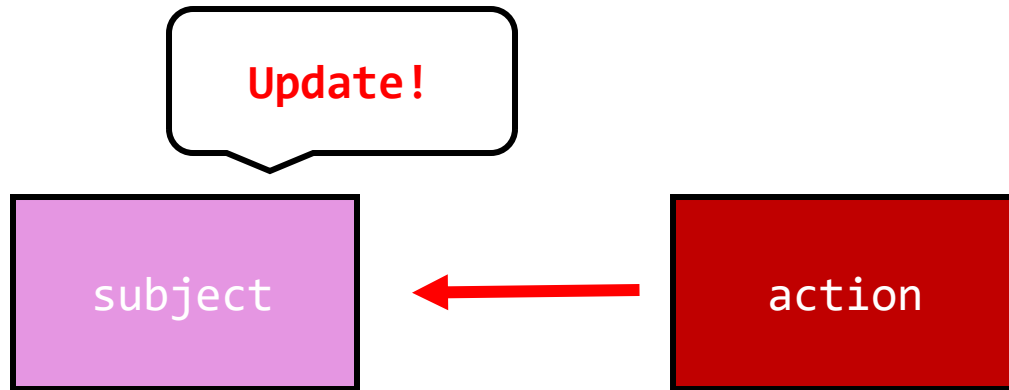


By subjectResourceID

Step 1

리소스 트리 어딘가에 action 오브젝트를 생성한다.

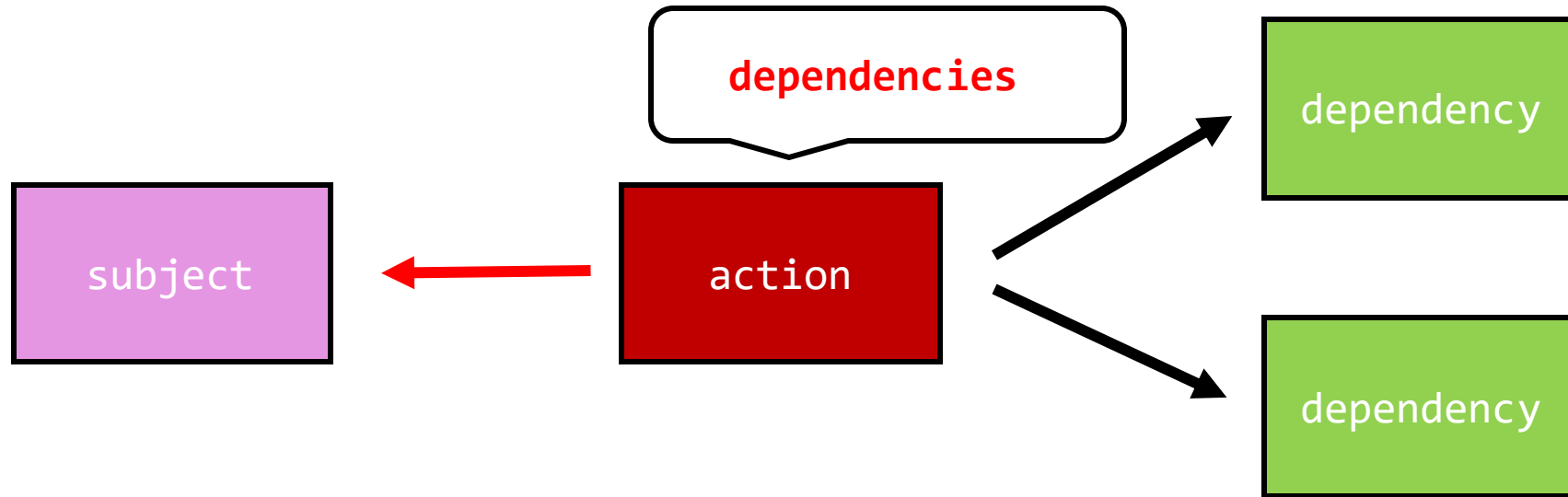
## Action Triggering 예시



Step 2

subject를 update한다.

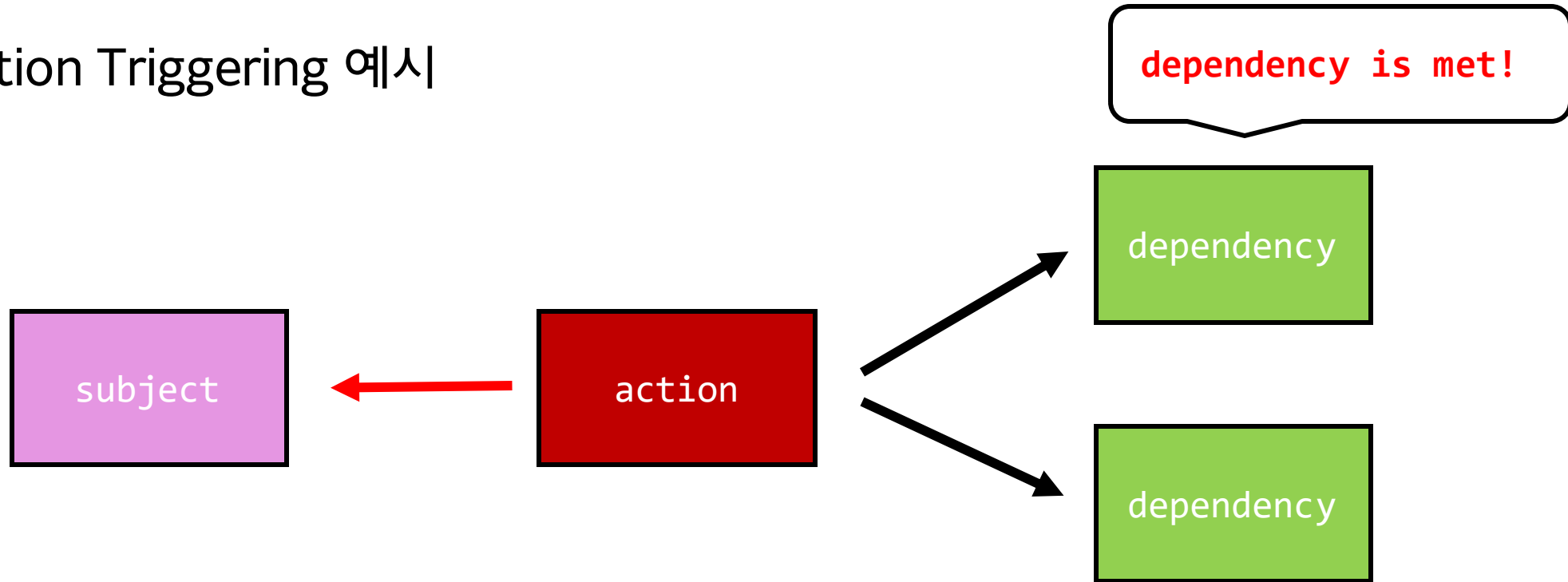
## Action Triggering 예시



Step 4

만일 dependency가 존재한다면 dependencies attribute의 우선 순위에 따라 dependency를 고려한다.

## Action Triggering 예시



Step 5

dependency 조건 만족에 따라 action triggering 여부 결정