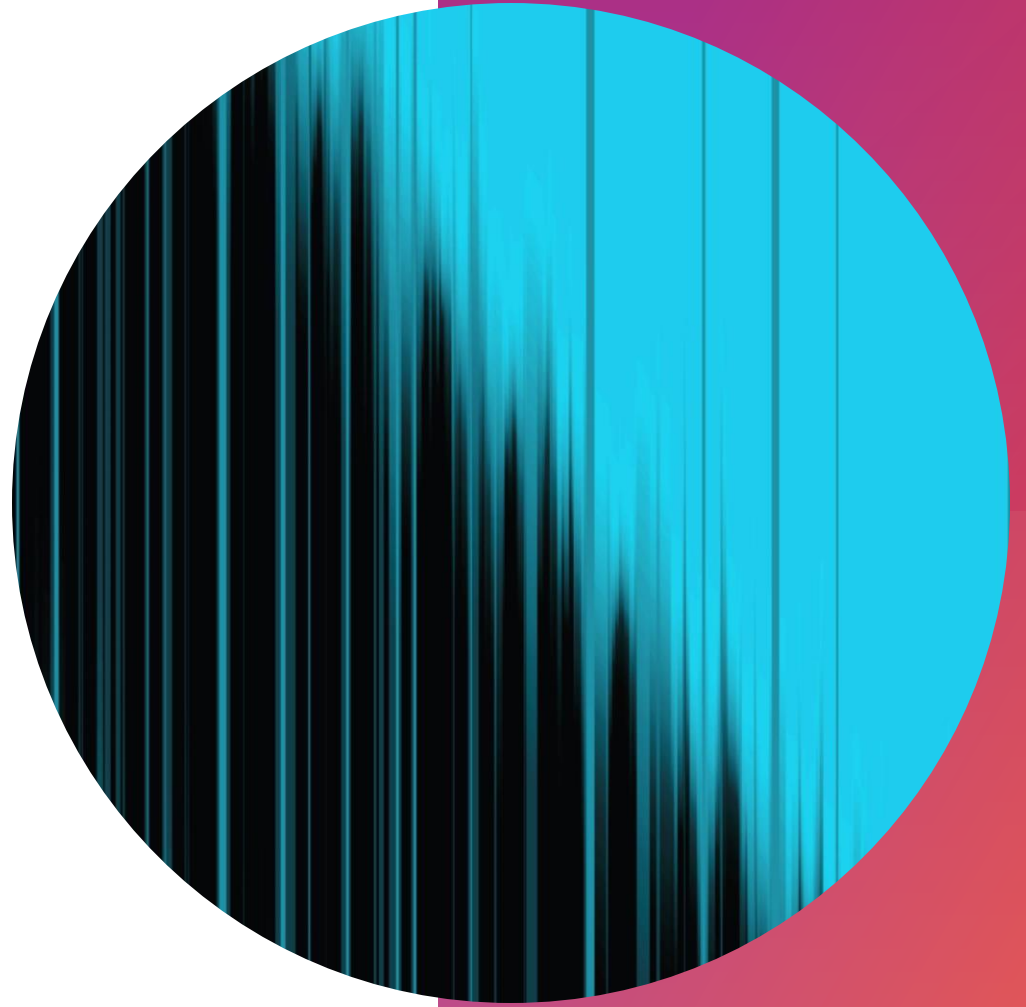


oneM2M

TinyIoT

ONEM2M TINYIOT SERVER

엄경호



이번 주 진행 상황

- I. 리소스 트리 변경 사항
- II. ACME 테스트 기반 보완

리소스 트리 변경 사항

기존 리소스 트리 컨셉 : 메모리 상에서 최소한의 정보를 담는 리소스 트리를 유지하여 리소스 트리 단에서 처리 가능한 로직들은 DB 접근을 하지 않고 처리 가능하게 만든다.

문제점

1. 생각보다 실제 리소스를 필요로 하는 동작이 많음
2. 리소스 트리 노드 변수의 양이 실제 리소스 변수들만큼 많아짐

노드에 최소한의 정보를 담고 있는 기존 형태 → 실제 리소스 자체를 가지고 있는 형태

리소스 트리 변경 사항

```
typedef struct RTNode {  
    struct RTNode *parent;  
    struct RTNode *child;  
    struct RTNode *sibling_left;  
    struct RTNode *sibling_right;  
  
    char *uri;  
    ResourceType ty;  
    void *obj;  
}RTNode;
```

type 값과 void *형 리소스를 가지고 있기 때문에

void *형 리소스를 리소스 구조체로 캐스팅하여 사용

당연하게도 대규모 오류 발생.. 대대적인 리팩토링이 필요했음

컴파일러가 잡아내지 못하는 로직 상 예외 상황들도 발생

ACME 테스트 기반 보완

Subscription 구현을 위해 기존 코드들을 현재 구조에 맞게 재작성 → CRUD는 가능한 상황

하지만 ACME 테스트로 ACP를 테스트 했더니 다양한 예외 상황 발생

고려 했어야 할 사항

1. acpi를 설정 시 실제 acp가 존재해야함
2. 1번 상황에서 실제 acp의 pvs의 권한을 따져야함
3. acpi는 update 시 유일한 attribute여야 함
4. acpi가 없다면 originator 외에는 acpi update가 불가능함 (admin도 불가능)
5. acpi가 존재한다면 originator도 예외가 없음

ACME 테스트 기반 보완

[ACME] - Test Results							
Test Suite	Count	Skipped	Errors	Times Exec Sleep Proc	Exec Time per Test Reque...	Proc Time per Test Request	Reque...
testACP	19	0	0	1.38... 0.00 0.2000	0.07... 0.0603	0.0105 0.0087	23
testAE	20	0	0	1.20... 0.00 0.1664	0.06... 0.0430	0.0083 0.0059	28
testCNT	19	0	0	1.65... 0.00 0.2365	0.08... 0.0718	0.0124 0.0103	23
testCNT_CIN	15	0	0	2.05... 0.00 0.2356	0.13... 0.0428	0.0157 0.0049	48
Totals	73	0	0	6.30... 0.00 0.8460	0.08... 0.0517	0.0116 0.0069	122

최대한 복구해본 상태..

testACP는 현재 스펙 상으로 약 7개 정도 테스트가

통과가 더 가능하지만 C로 구현하기에 많이 까다로워 보임

acpi를 제거할 때도 acp의 권한을 따져야 하는데,

acpi 에 ["ACP1"] -> ["ACP2"] 처럼 삭제와 생성이

동시에 일어난 경우는?