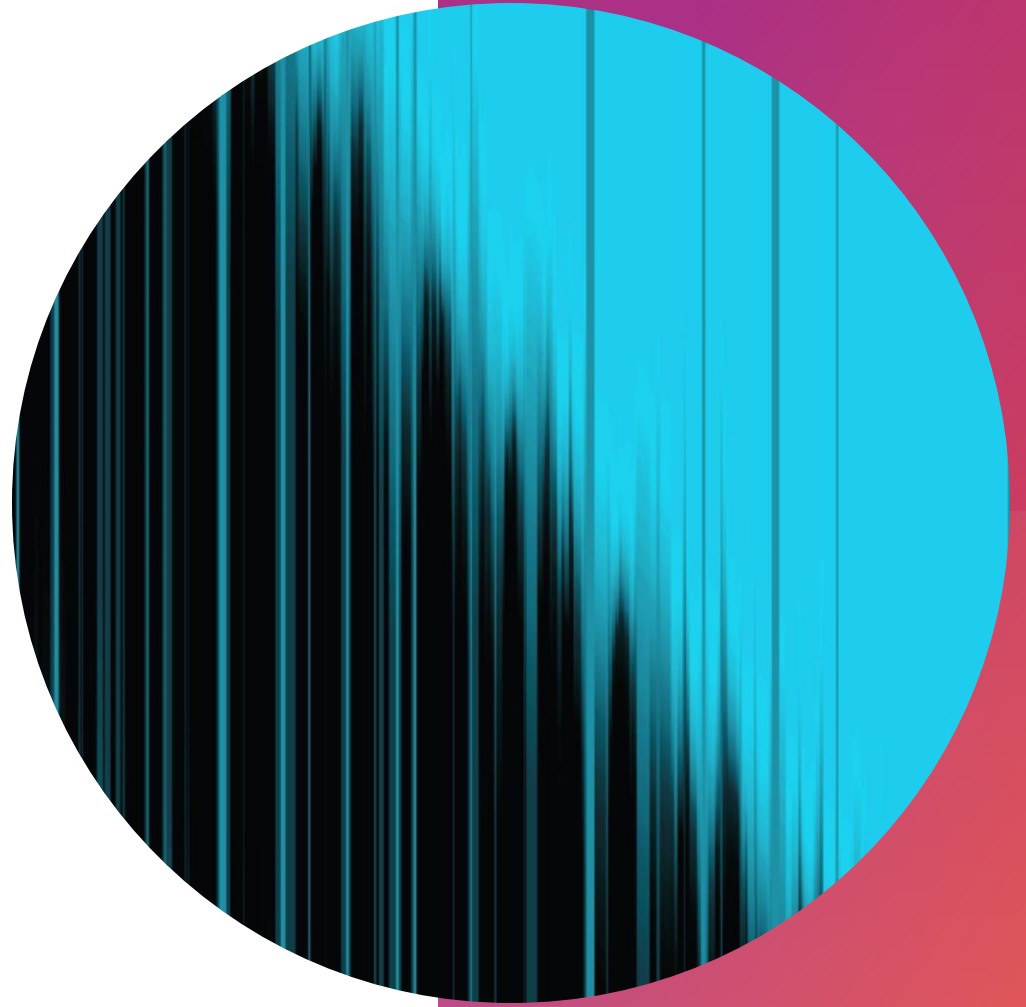


HTTP Daemon

ONEM2M TINY IOT PROJECT

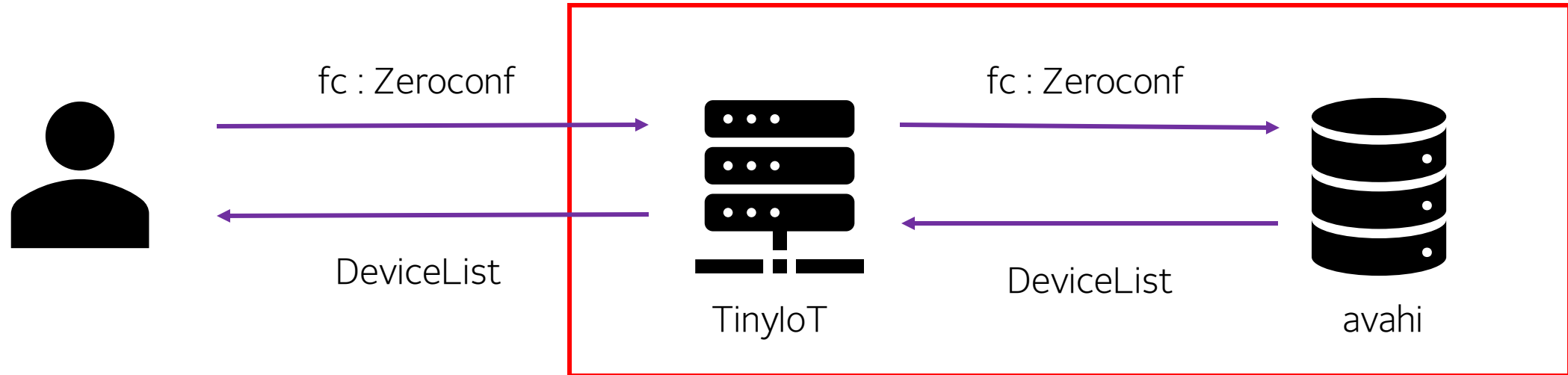
엄경호



이번주 진행 상황

- I. zero-conf 서버 스크립트 구현
- II. SubNode 삭제 -> Node 원복
- III. 테스트 대비를 위한 Mobius 모방

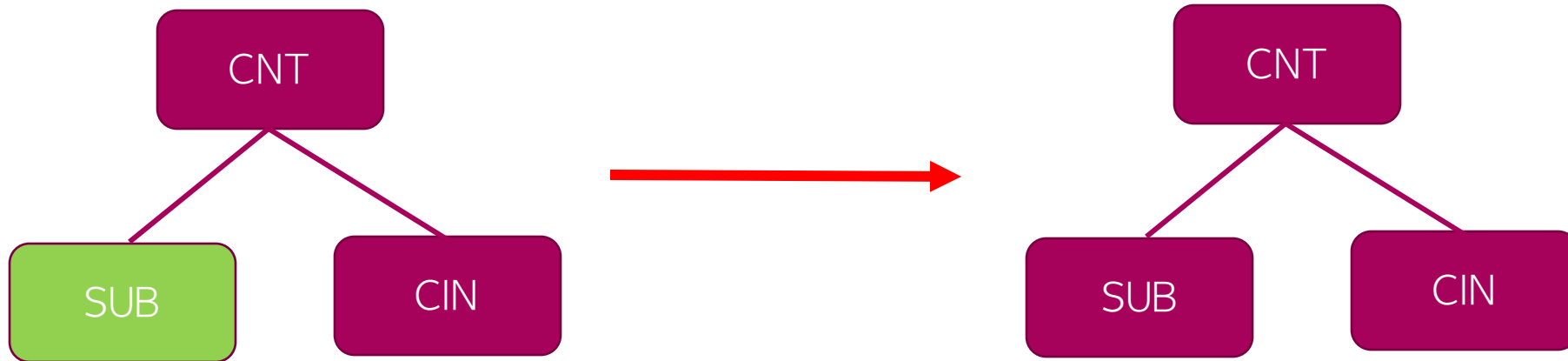
zero-conf 서버 스크립트 구현



fc 필드 값이 Zeroconf인 Get 요청이 CSE에 들어오면 avahi 서버에 DeviceList 를 요청하여 그 리스트를 클라이언트에게 다시 반환

현재는 아직 PlainText 형태지만 JSON으로 가공하는 방향을 제시해봄

SubNode 삭제 및 Node 원복



Sub 오브젝트 노드에만 필요한 변수들 (sur, net 등)이 존재했기에 SubNode를 별도로 정의 했었음
필요 이상의 변수들이 기존 노드에 포함 <<<<<<< 기존 리소스 트리 함수의 재사용

테스트 대비를 위한 Mobius 모방

```
"m2m:dbg": "invalid object"
```

간단한 에러 메시지도 json 형태로 응답

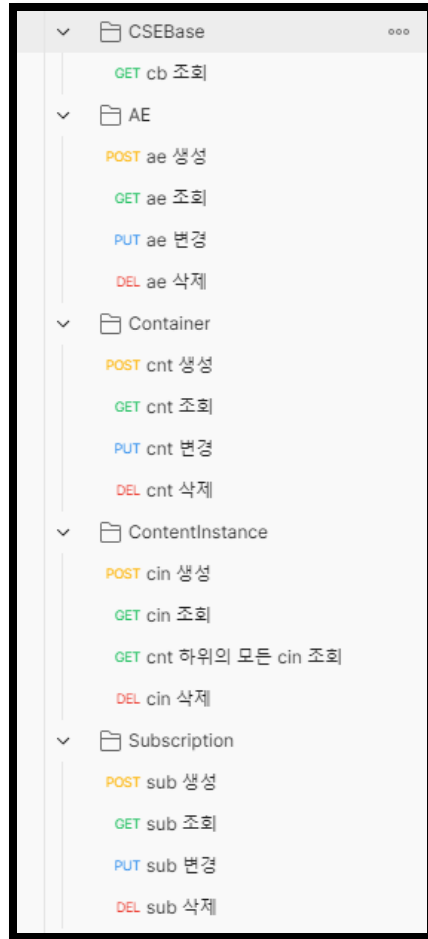
Access-Control-Allow-Origin	①	*
Access-Control-Allow-Methods	①	GET, PUT, POST, DELETE, OPTIONS
Access-Control-Allow-Headers	①	Origin, X-Requested-With, Content-Type, X-
Access-Control-Expose-Headers	①	Origin, X-Requested-With, Content-Type, X-
x-m2m-ri	①	12345
x-m2m-rvi	①	2a
Accept	①	application/json
Content-Type	①	application/json; charset=utf-8

응답 헤더의 필드와 해당 값들

```
"m2m:ae": {  
  "pi": "5-20220909124059366654",  
  "ri": "S0origin",  
  "ty": 2,  
  "ct": "20220909T124427",  
  "rn": "justin",  
  "lt": "20220909T130430",  
  "et": "20240909T124427",  
}
```

et는 ct기준 2년 update 시에도 갱신 x
ri는 생성 시간 + 랜덤 값으로 고유성 확보
특이하게 AE의 ri는 X-M2M-Origin 필드 값
해당 값으로 접근 제어를 함

테스트 대비를 위한 Mobius 모방



Mobius를 로컬 설치 후 해당 Mobius가 잘 동작하는지 확인하기 위한

postman collection이 존재함

현재 TinyIoT 서버에 테스트를 해보았을 때 Update 제외하고 모두 잘 동작함

※ Update가 동작하지 않는 이유

Mobius -> 업데이트 하고자 하는 필드 값만 페이로드에 포함함

TinyIoT -> Create시 필요한 필드 값들을 모두 페이로드에 포함함

테스트 대비를 위한 Mobius 모방

