

# **Energy Framework**

Smart Home  
MIR Lab  
Hanyang University

# Index

---

## 1. Smart Home Energy Framework

### 1.1 Background

- 1.1.1 스마트그리드
- 1.1.2 수요반응 (OpenADR, SEP2.0, EMAP)
- 1.1.3 에너지 IoT
- 1.1.4 에너지 IoT 수요반응 프로토콜 (CoAP, MQTT)
- 1.1.5 Micro Grid (OpenFMB)
- 1.1.6 AMI**

### 1.2 Architecture

### 1.3 Block Diagram

### 1.4 Flowchart

### 1.5 참고문헌 논문 정리

## 2. Profile : OpenADR, EMAP, DLMS/COSEM

### 2.1 OpenADR

### 2.2 EMAP

### **2.3 DLMS/COSEM**

## 3. System : EMS, Server EMA, Client EMA, AMI, Device, DCU

### 3.1 EMS

### 3.2 Server EMA

### 3.3 Client EMA

### 3.4 Device

### **3.5 MDMS**

### 3.6 DCU

### 3.7 Smart Meter

## 4. Experiment

1. Experiment Scenario
2. Experiment Procedure

## 5. Appendix

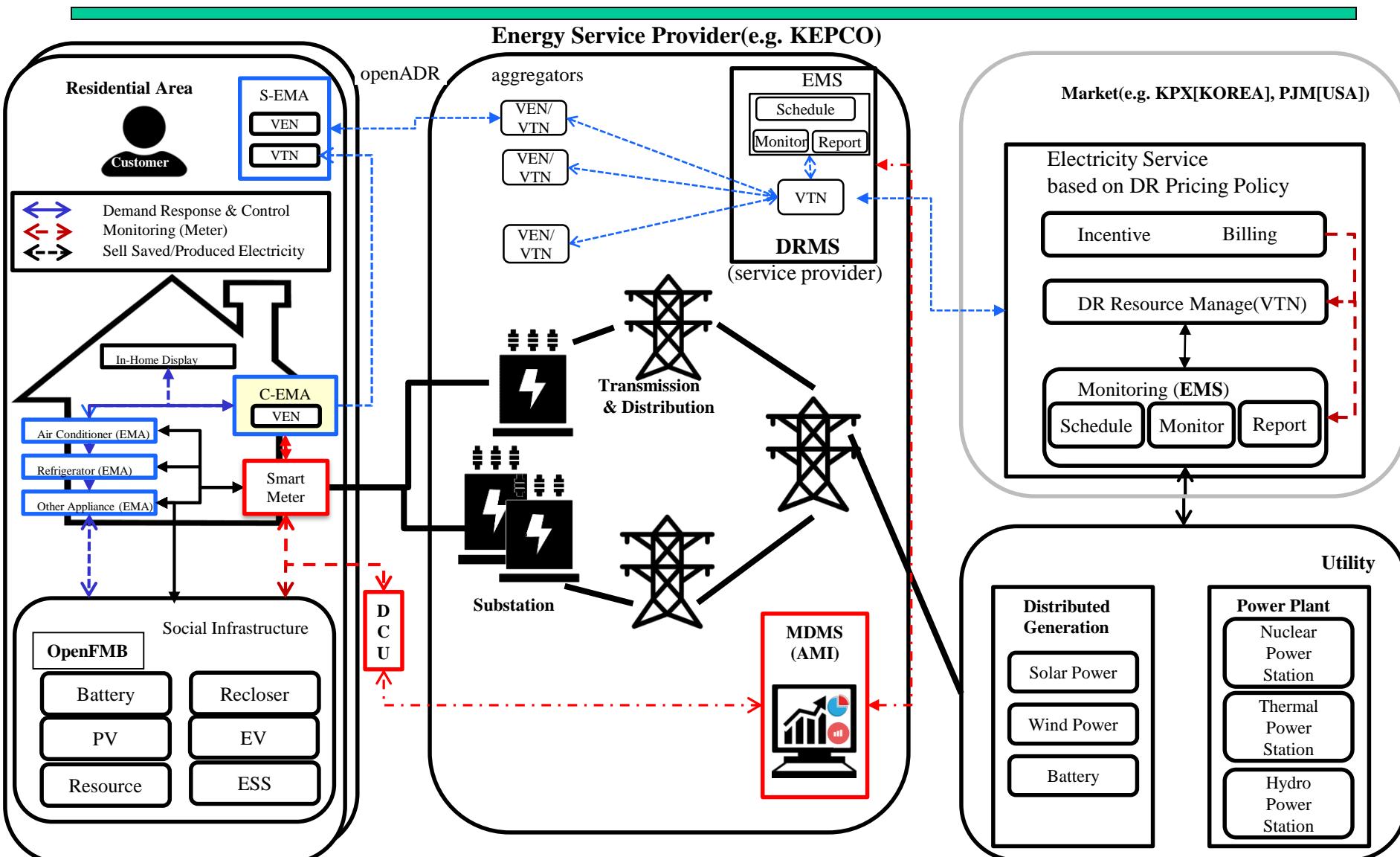
# 1. Smart Energy Home Framework

## 1.1 Background

- 1.1.1 스마트그리드
- 1.1.2 수요반응 (OpenADR, SEP2.0, EMAP)
- 1.1.3 에너지 IoT
- 1.1.4 에너지 IoT 수요반응 프로토콜 (CoAP, MQTT)
- 1.1.5 Micro Grid (OpenFMB)
- 1.1.6 AMI

# 1. Smart Home Energy Framework :

## 1.1.1 Background Smart Grid (1/2)



# 1. Smart Home Energy Framework :

## 1.1.1 Background 스마트그리드 (2/2)

- **Smart Grid**
  - 스마트그리드는 기존의 전력망(Grid)에 정보통신기술(Information and Communication Technologies; ICT)을 도입하여 융합하여 전력공급자와 소비자가 양방향으로 실시간 정보를 교환함으로써 에너지 효율을 극대화하는 첨단전력기술
- **수요자원 거래 시장**
  - 전력력시장 운영, 전력계통 운영, 실시간 계통 운영, 정부의 전력수급기본계획 수립 총괄 지원
- **발전사업자 및 수요관리 사업자**
  - 발전사업자 : 발전을 통해 전기를 생산하는 업체, 공장이다. 국내에는 한전자회사, 일반 신재생에너지발전사
  - 수요관리 사업자: 화력이나 원자력발전이 등과 같이 전기를 생산하는 방식이 아니라 사용하지 않고 아낀 전기를 판매하고 금전적인 보상을 받는 수요자원시장에 참여하는 사업자로 14년 11월 말 처음으로 전력시장 내 수요자원거래 시장이 개설 되었으며 현재 14개 사업체가 운영
- **분산전원**
  - 전력 소비가 있는 지역 근처에 분산·배치가 가능한 소규모의 발전 시설로, 비상용 예비전원과 같이 정전 시에만 이용 가능한 발전설비는 제외
  - Comparable with HTTP (CoAP-HTTP Proxy needed)
- **AMI**
  - 스마트미터에서 측정된 데이터를 원격 검침기를 통해 측정해 전력 사용 분석을 자동으로 진행하는 기술

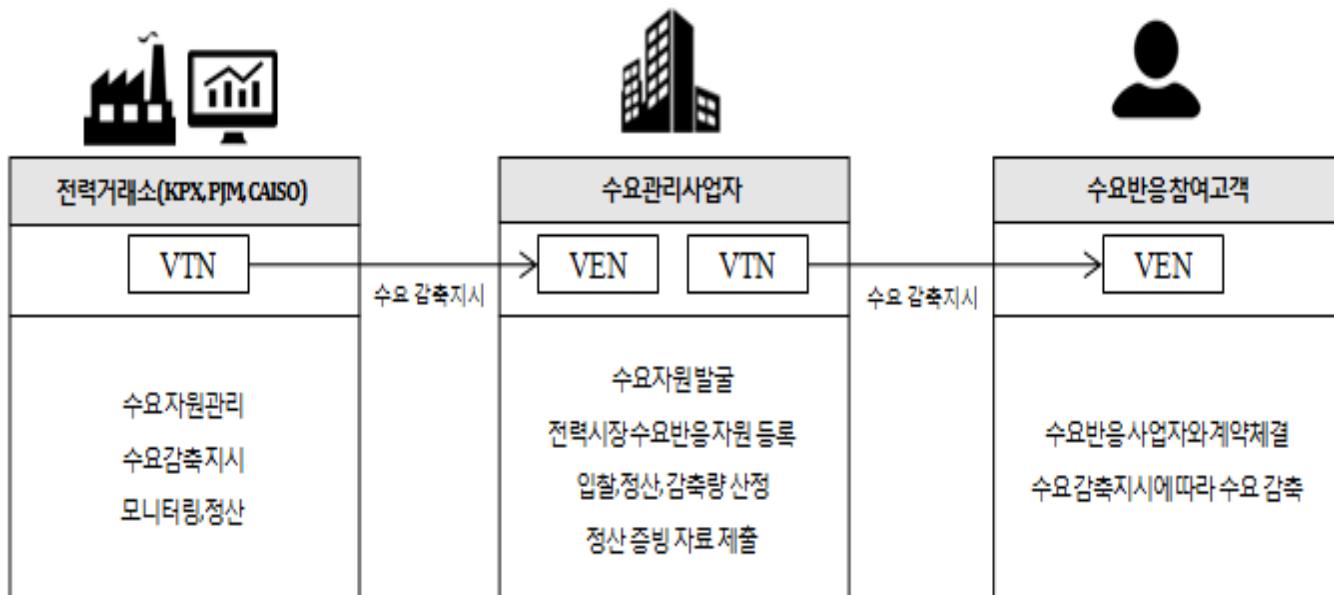
# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 (1/3)

### 수요반응 (DR - Demand Response)[\*]

-증가하는 전력 수요와 피크부하에 대응하고 안정적인 전력망을 구축하기 위한 방법

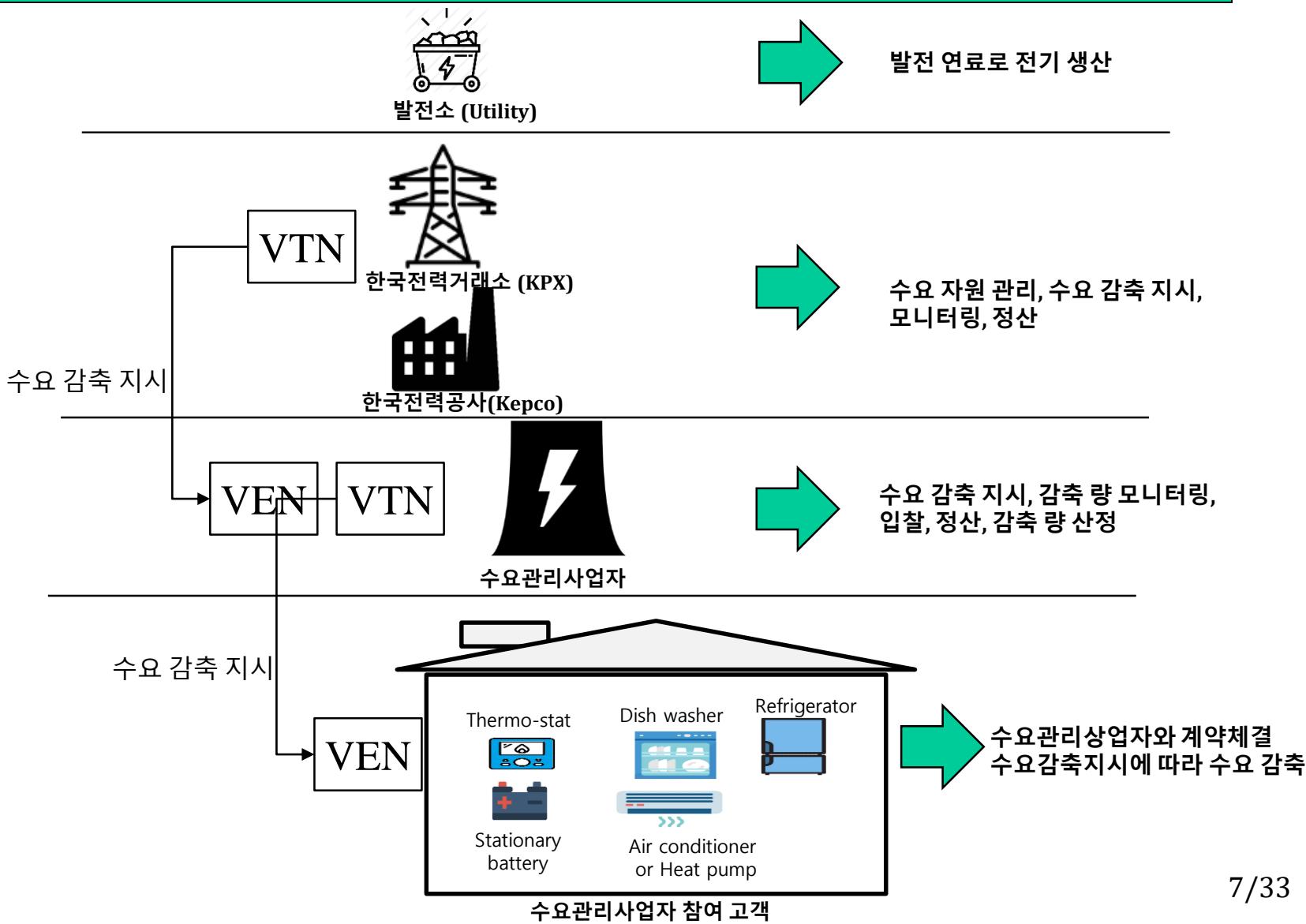
전력 소비자 측에서 전기 요금이나 공급자의 요청에 따라  
**전력 소비를 조절함으로써 전력계통의 안정성과 신뢰성을 확보하는 기술**



[\*] Paulson Institute, Demand Response, 2015

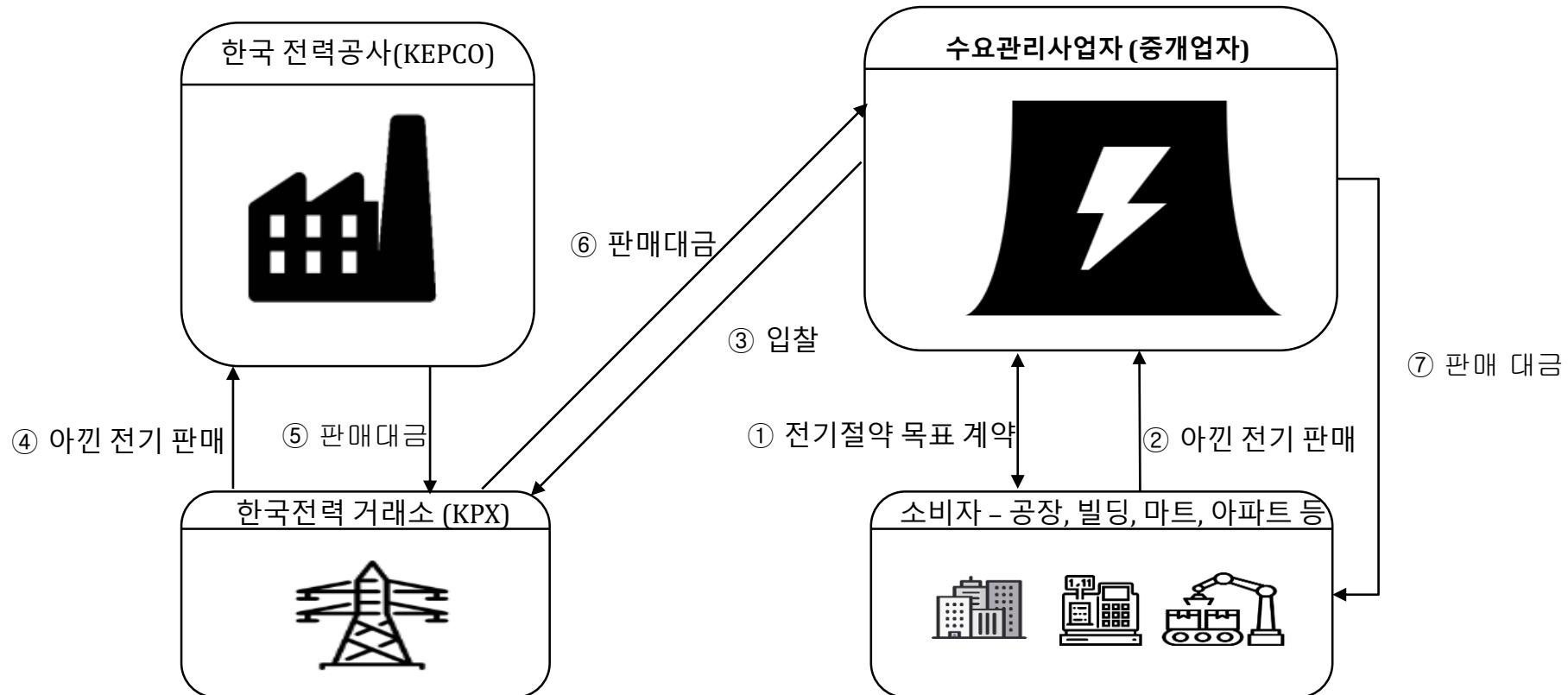
# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 (2/3)



# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 (3/3)



- 절약한 전기를 되팔려면 아이디알서비스 벽산파워 그리드파워 등 11개 수요관리사업자(증개업자)와 계약을 맺으면 된다. 기존 사용량보다 얼마만큼 적게 전기를 쓰겠다는 목표를 정해 계약하면 수요관리사업자가 실제 절약분을 한국전력거래소에 입찰한다.

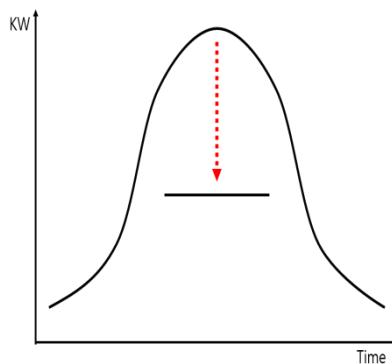
# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 : OpenADR (1/2)

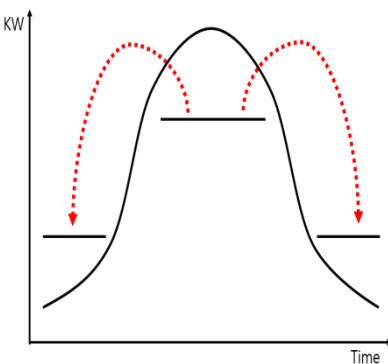
OpenADR(Open Automated Demand Response)[9]

지능형 DR에 적용되는 표준 통신 프로토콜이다.

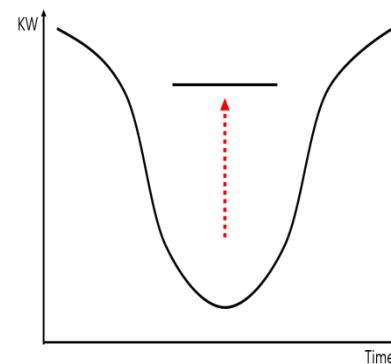
Utility Company와 ISO(International Organization for Standardization), Third part service provider가 소비자들의 최대 전기 사용량 차이를 완화시키고, 전기 공급과 수요의 균형을 맞추며 일시적으로 전기 사용량을 축소시키기 위해  
**소비자의 전력사용패턴 변화를 유도하는 기술이다.**



Hold Down Maximum Demand



Transfer Maximum Demand

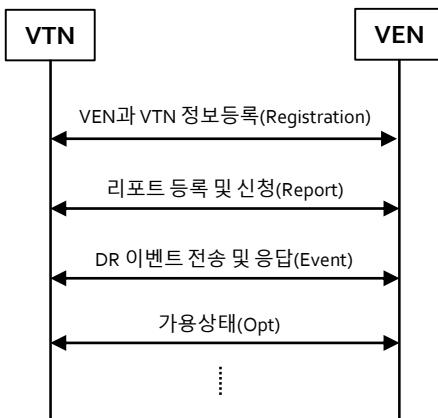


Increase of Base Load

# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 : OpenADR (2/2)

- OpenADR2.0b 동작 과정



- VEN과 VTN이 서로의 정보를 교환하여 연결을 수립하고, VEN/VTN 각각이 관리하고 있는 자원들의 정보를 등록 및 구독한다. 이후 수요자원의 사용 패턴 변화가 요구될 때, DR Event Signal을 보내어 VEN의 자원 사용 패턴 변화를 유도한다. VEN이 Event에 참여할 경우 VTN에게 참여여부를 알려준다. (Opt-in/out)

- OpenADR2.0b ( 4 Services )

- Registration
- Report
- Event
- Opt

- Registration

- VTN과 VEN이 서로 연결을 수립하는 과정이다. 서로의 정보를 교환하고, VTN에 VEN의 ID Name이 미리 등록되어 있을 경우 진행이 가능하다. VTN에서 VEN의 ID값을 발급해주고, 이 값을 이용하여 나머지 서비스들을 진행 할 수 있게 된다.

- Report

- VTN, VEN 모두 자신의 Report 자원을 서로에게 등록할 수 있으며, Report자원 중에서 원하는 것을 선택하여 구독신청 한다. 이 후 UpdateReport 메시지를 통하여 Report를 지속적으로 보내준다.

- Event

- 수요자원의 사용패턴 변화를 위한 Event 신호를 내려주는 기능을 한다. Event 신호에는 Event시간, 대상, 내용(level, price, power 등)에 대한 정보가 포함되어 수요반응을 만들어 낸다.

- Opt

- VEN의 가용상태를 알려주는 서비스이다. Event를 수행 할 때는 Opt-in, 불가능 할 때는 Opt-out메시지를 VTN에게 보내어 VEN의 가용상태를 알려준다.

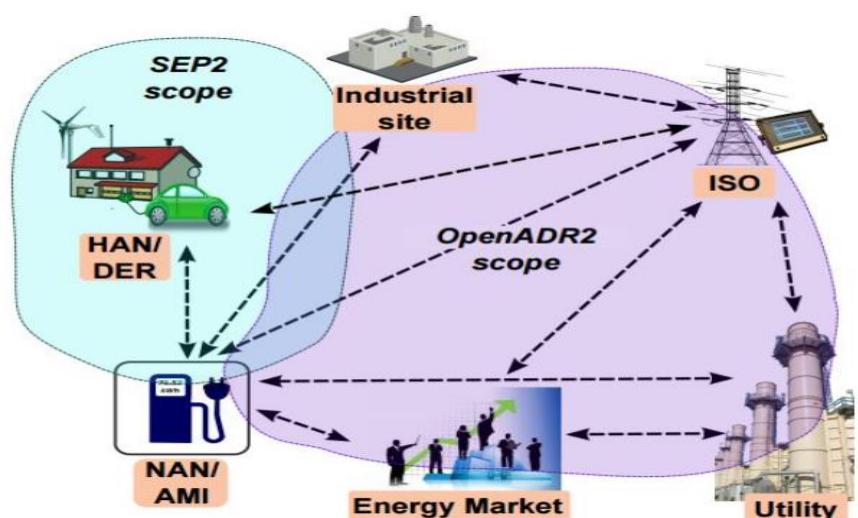
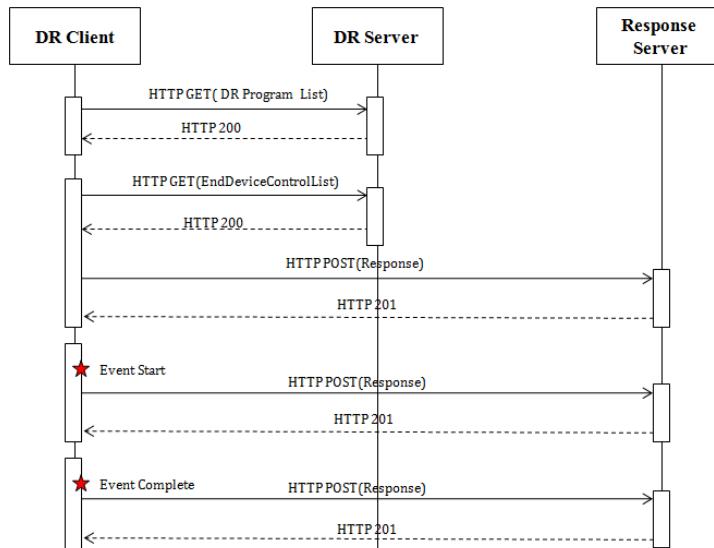
# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 : SEP 2.0 (1/2)

### SEP 2.0(Smart Energy Profile 2.0)

SEP 2.0 프로토콜은 주로 IP 상단 계층에 위치하는 프로토콜로, 리소스 중심의 URI (Uniform Resource Identifier) 기반 RESTful (Representational State Transfer) 웹 서비스를 사용한다. 기본적인 동작은 HTTP 함수를 이용하여 웹 어플리케이션에서 동작하며 XML 형식의 정해진 데이터를 전송한다.

DRLC (Demand Response Load Control), Metering, Pricing 등의 function set을 포함하고 있다. 각각의 function set마다 시퀀스 다이어그램 및 데이터 구조를 보여주고 있으며, 이를 기반으로 서버와 클라이언트 간의 메시지 교환이 이루어진다.



# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 : SEP 2.0 (OpenADR vs SEP 2.0) (2/2)

### OpenADR vs SEP 2.0(Smart Energy Profile 2.0)

구분	OpenADR2.0	SEP 2.0
Supported Services	Simple HTTP/XML	Restful HTTP
Direct Load Control	Limited	YES
Profile Structure	Tiered Profile	Modular Profile
Full Reporting Services	Yes	No
Non Repudiation	Optional	Not Specified
Boundary	Enables automated AutoDR to commercial, industrial and residential customers	Enables residential and light commercial DR
Communication	Communicates over the Internet using web services	Communicates over Automated Metering Infrastructure (AMI) or via a broadband gateway
Data size	Transmits larger data packets	Transmits small data packets
Service	Service provider (server) to customer energy system interface (client)	Ideally suited for use within a home or building

# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 : EMAP (1/4)

### 에너지 관리 에이전트 프로토콜(EMAP)의 필요성

(1) **국민 DR의 정책**으로 인해 수많은 집합건물의 수요관리 고객에 대한 **실시간 수요반응이 필요(빠른 수요반응 이벤트 )** 하며 효율적으로 에너지를 관리하기 위해 경량 수요반응 프로토콜이 필요

(2) 집합 건물 내부에 에너지관리에이전트가 있어 **개인 사용자가 상위 에너지관리에이전트와 통신을 통해 수요반응 및 스케줄링을 할 수 있음**

(이전 연구에서는 집단 건물의 총괄적 EMA로만 관리) → 단점 : 보안의 중요성과 EMA과 모든 기능을 수행을 못할 수도 있다.

(3) 에너지관리에이전트 프로토콜의 확장성을 높이고 각 가정의 독립적 사용과 프라이버시를 보장하기 위한 Implicit/Explicit 방법을 제시

(4) 집합 건물 내의 Client EMA는 Gateway나 IoT 디바이스에 모듈 형식으로 탑재됨

- Heavyweight인 HTTP/XML 방식의 수요반응 프로토콜은 제한이 있음(OpenADR2.0b, SEP2.0)

- 디바이스나 IoT기기, Gateway들에는 제한된 리소스 및 배터리 용량이 제한[10][11]

- IoT(CoAP, MQTT) 프로토콜이 lightweight, 저전력 기반[12]

(5) 제한적인 IoT환경[12]에서 데이터 트래픽을 줄이기 위해서 IoT 프로토콜의 메커니즘들이 필요

- 집합건물의 수많은 개인지역으로 확대, 효율적 수요반응 관리가 필요하게 됨(다양한 집합건물의 구조에서 유리)

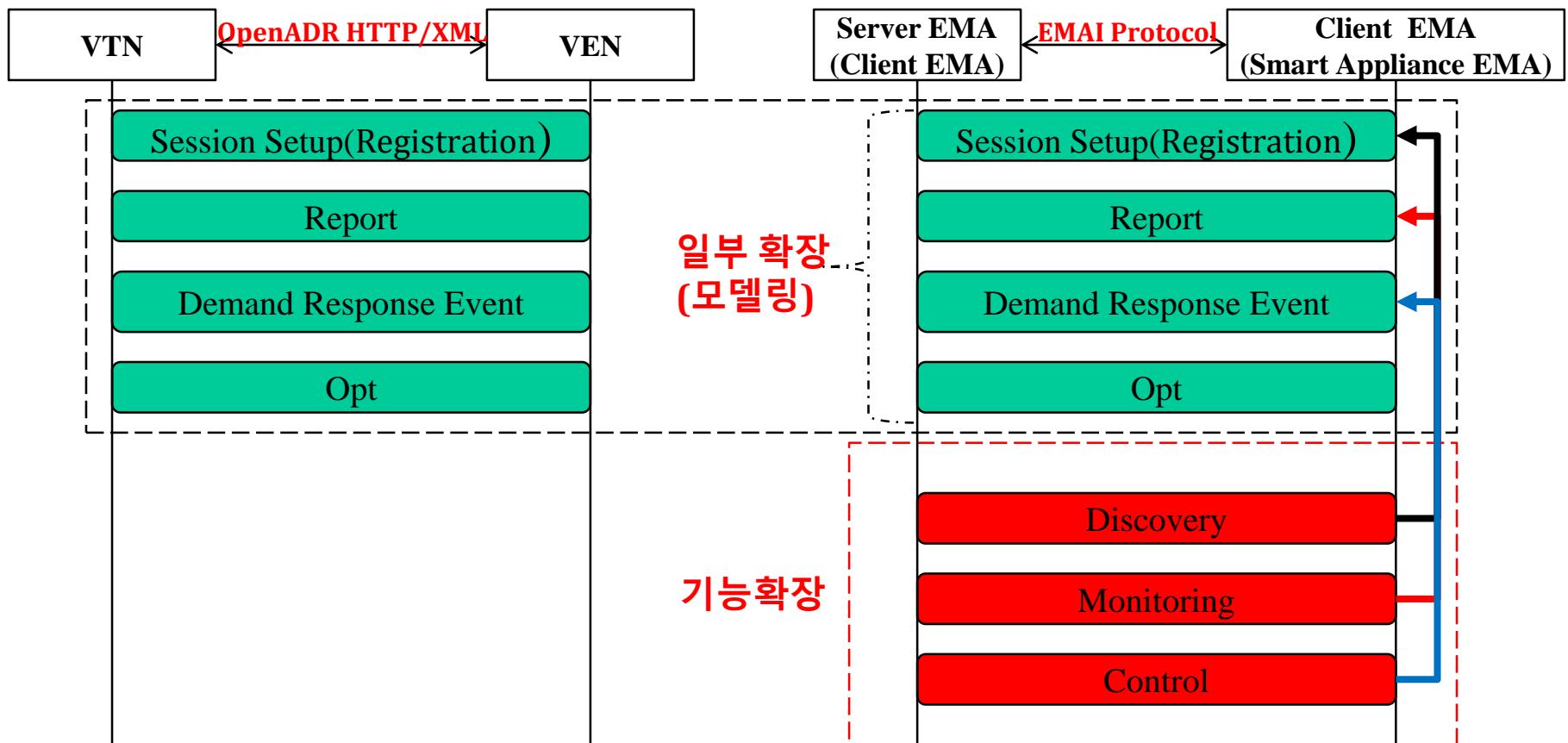
- 1. CoAP, MQTT 기반 lightweight, 2. CoAP Obs, 3. MQTT Broker 기반 Push 메커니즘, 4. MQTT 기반 멀티캐스트

# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 : EMAP

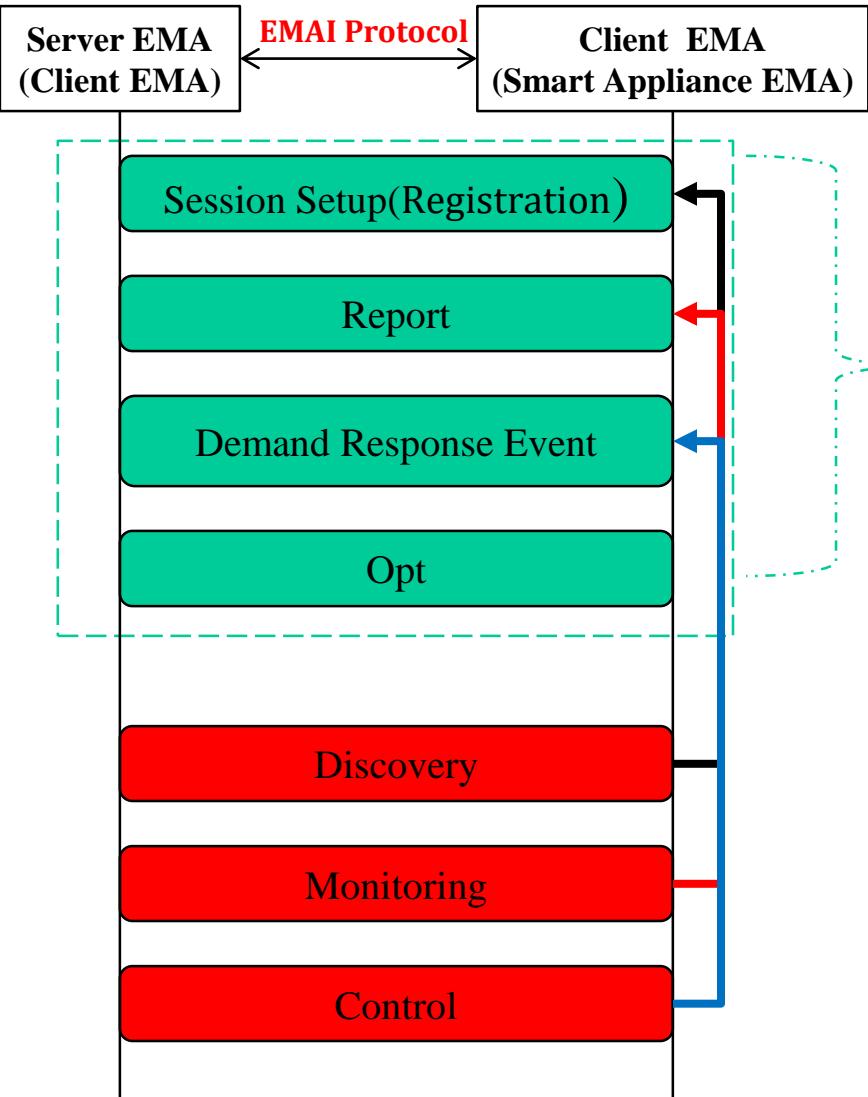
### EMAI Protocol

- EMA사이의 수요반응 통신 프로토콜이며 OpenADR 2.0b의 모델링을 따랐으며 일부 모델링을 확장했고 Discovery와 Monitoring, Control 부분의 통신 프로토콜이 확장.



# 1. Smart Home Energy Framework :

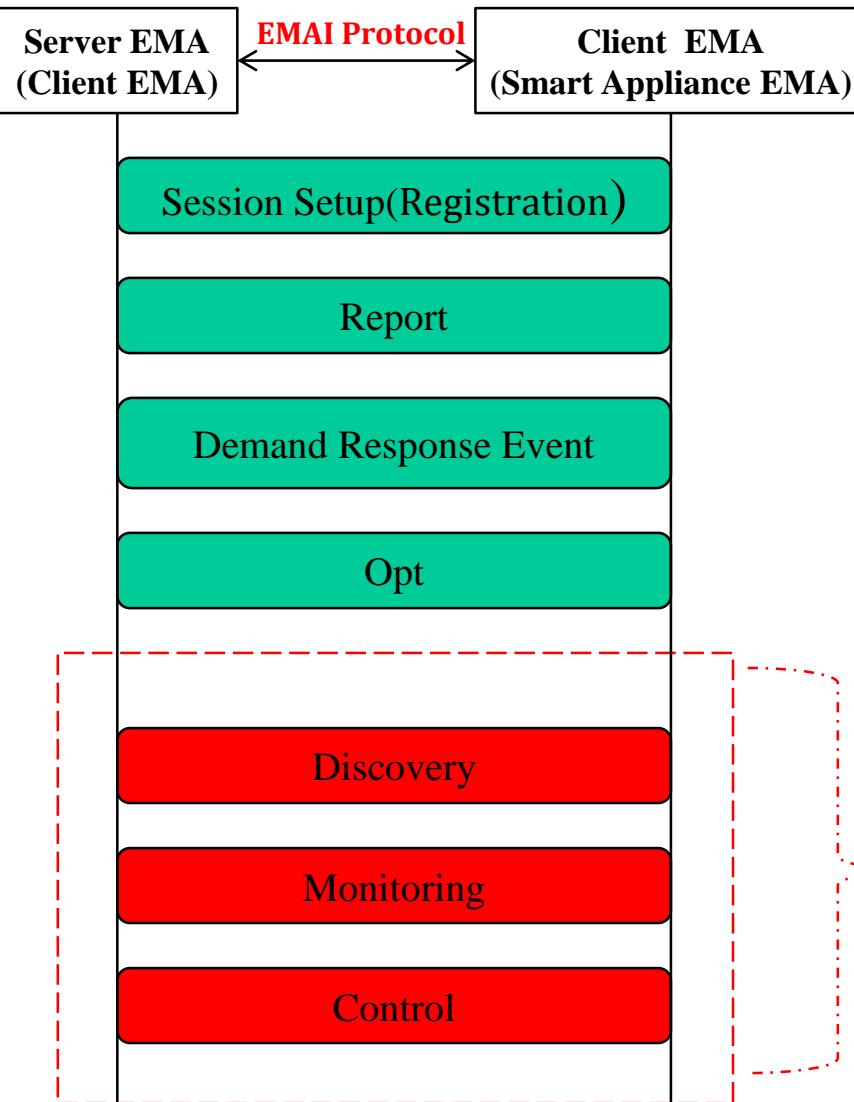
## 1.1.2 Background 수요반응 : EMAP (3/4)



- Session Setup(Registration)
  - 에너지관리에이전트 간 서로 연결을 수립하고 Report을 교환할 때 기존의 Report에 대한 에너지 가격 정보나 클라이언트 에너지관리에이전트의 에너지 관리 정보를 얻음.
- Report(updateReport)
  - 에너지관리에이전트 간 서로 연결을 수립 할 때 Report을 교환할 때 실시간 에너지에 대한 가격 정보나 클라이언트 에너지관리에이전트의 디바이스 정보 등을 얻음.
- Event(Demand Response Event)
  - 에너지관리에이전트 프로토콜은 다양한 이벤트 종류가 있다. 예로 가격기반 수요반응 이벤트에서 Initial Price, Incentive Price, Negotiation Price가 있다. 서버 에너지관리에이전트는 실시간적으로 가격에 대한 정보를 이벤트로 발생
- Opt
  - 에너지관리에이전트 프로토콜의 Opt는 클라이언트 에너지관리에이전트가 상위 서버 에너지관리에이전트에게 수요반응 이벤트의 가용상태 또는 수요반응 이벤트 프로그램 변경, 수요반응 스케줄링을 요청을 알려주는 서비스

# 1. Smart Home Energy Framework :

## 1.1.2 Background 수요반응 : EMAP (4/4)



### • Discovery(Connect)

- EMA Discovery 과정입니다. Connect 메시지는 Monitoring에서 Explicit/Abstract를 결정하는 단계입니다. 기존 HTTP/XML 방식의 OpenADR에는 Connect와 Disconnect 가 없습니다

### • Discovery(Disconnect)

- disconnect 하위 EMA가 연결을 끊을 땐 메시지를 절차에 의해서( CancelRegistration을 보내면) 상위 EMA쪽에서 확인하게 되며 session을 종료하게 되며 다른 Client EMA에게 session 종료를 알린다.

### • Monitoring

- 하위의 EMA의 정보를 모니터링 하는 단계입니다. Explicit/Abstract 방식의 Monitoring 방식이 있습니다.

### • Control

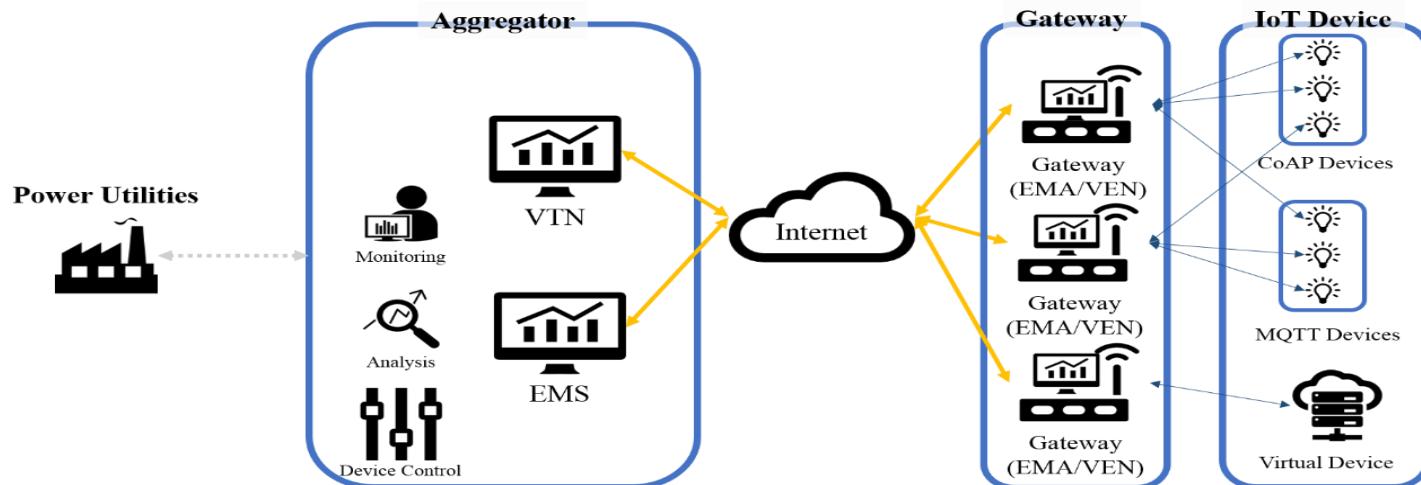
- EMA가 최하위 Device 컨트롤하기 위해서 Device를 총괄적으로 관리하는 Smart Appliances에게 Control 메시지를 줌

# 1. Smart Home Energy Framework :

## 1.1.3 Background 에너지 IoT

### • 에너지 IoT(EIoT)[1]

- 기술의 트랜드가 모든 사물이 네트워크를 통해 연결되는 초연결 사회로 진행됨에 따라 IoT를 이용한 다양한 서비스에 대한 연구와 서비스 구축을 위한 노력이 다방면에서 진행
- 스마트그리드는 전력 부문에 정보통신 기술을 이용하는 시스템이므로 사물인터넷이 그 중심이 될 것이라 기대가 모아지고 스마트 홈과 같이 가정 내의 에너지 기기를 관리하는 데에 사물인터넷을 접목하는 에너지 IoT가 등장

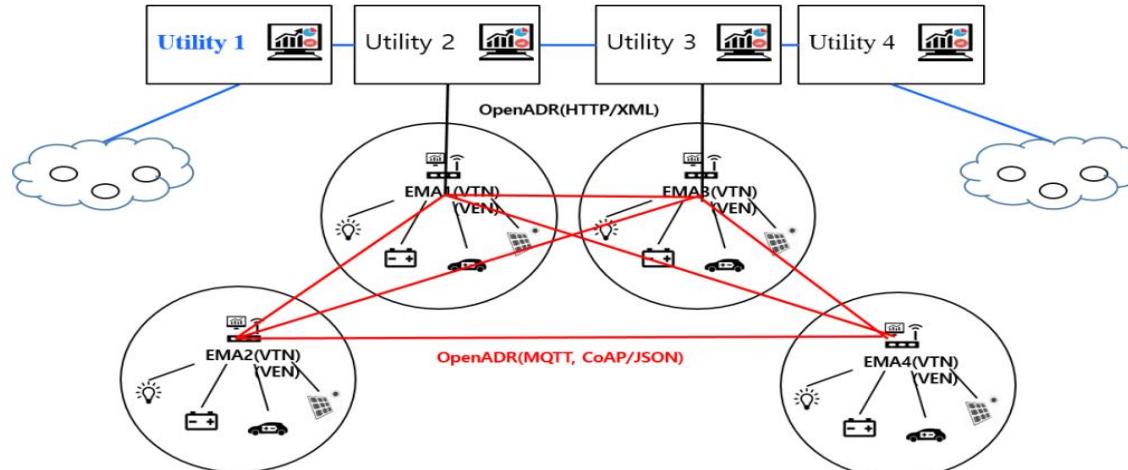


[1] 박현일, 김세영, 강성철, 박현진, 김일연, 최진식. (2017). 스마트 에너지 IoT 를 위한 CoAP 기반 Lightweight OpenADR2.0b 프로토콜의 구현 및 분석. 한국통신학회논문지, 42(4), 904-914

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 (1/2)

초경량 IoT(Internet of Things) 프로토콜은 MQTT(Message Queuing Telemetry Transport)[10]나 CoAP(Constrained Application Protocol)[11] 등이 있다. MQTT는 사물인터넷 망과 같은 대역폭이 제한된 통신 환경에 최적화하여 개발된 Push technology 기반의 초경량 메시지 전송프로토콜이다. CoAP도 대역폭이 제한된 통신 환경에 최적화하여 개발된 초경량 메시지 전송 프로토콜이다. CoAP와 MQTT로 수요반응 통신 프로토콜을 구현한 것이 초경량 수요반응 프로토콜이라고 한다.



[10] ISO/IEC 20922:2016, Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1

[11] Shelby, Zach, Klaus Hartke, and Carsten Bormann. "The constrained application protocol (CoAP)." (2014).

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 (2/2)

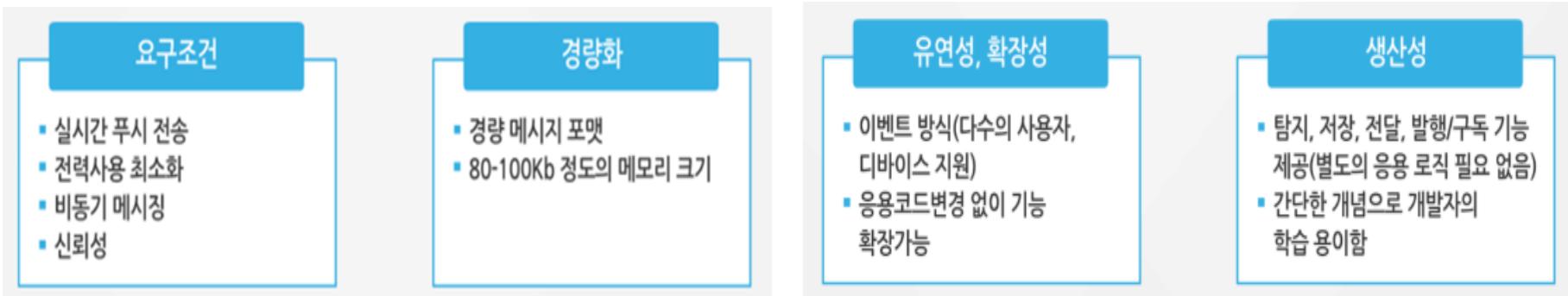
Protocol / Traits	Architecture Style	Intended or Actual Deployment	Relative position to other protocols	Data Model / Data Representation	Messaging	Security
CoAP	Client / server model P2P RESTful	IoT networks with low power constrained sensors such as smart metering	Above UDP (or DTLS/ UDP)	Plain text, XML, JSON, EXI, octet-stream Support for content negotiation	Request / Response	Largely depends on lower layers
MQTT	Client / Server model. Brokered style	Low bandwidth, high latency networks	MQTT runs above TCP (or TLS/TCP)	No formal data model	Publish-Subscribe	Authentication: Userid / password can be passed in a packet. SSL / TLS can be used
HTTP	RESTful	WWW	Above TCP (or TLS/TCP)	XML, JSON, etc. Support for Content negotiation	Request – Response	Largely depends on lower layers (SSL / TLS)
Web Sockets	Full duplex communication over TCP	Low latency, high performance web applications	Above TCP Uses HTTP for initial handshake	JSON, etc.	Full duplex communication same over TCP socket.	TLS
DDS	Data centric model (Virtual) Global Data space and broker-less Peer-to-Peer model	Several segments such as health care, factory, UAVs, Asset tracking, etc	Can run over UDP, TCP, shared memory and other transport types	DSSI defines a standard data format based on extension of Common Data Representation Named topics, user defined data types	Real-time Publish-Subscribe	TLS and some OMG specific security methods Vendor specific extensions also available
XMPP	Availability for Concurrent Transactions (ACT) style for carrying out asynchronous end-to-end exchange of structured data	Instant Messaging and Presence Applications, Jabber	Above TCP (or TLS/TCP)	XML, etc.	Publish-Subscribe	SASL, TLS, lower layer security

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : MQTT (1/7)

- MQTT의 특징

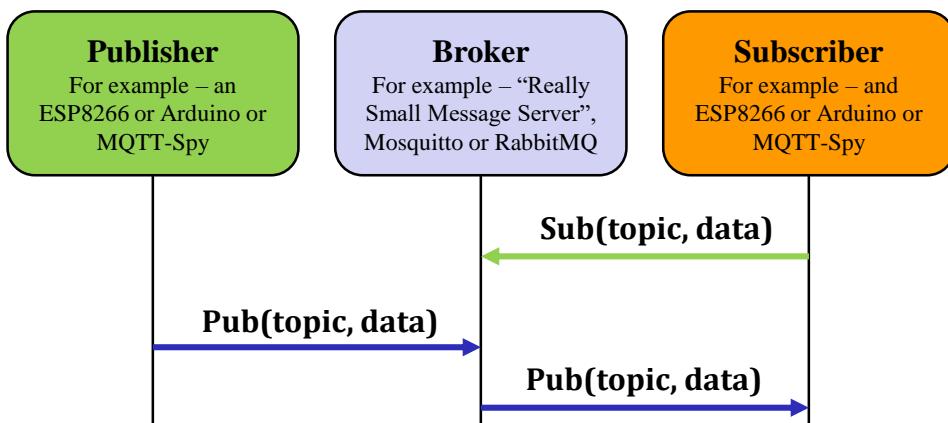
- 프로토콜이 차지하는 모든 면의 리소스 점유를 최소화
- 느리고 품질이 낮은 네트워크의 장애와 단절에 대비
- 클라이언트 애플리케이션 동작에 자원 활용이 극히 제한적임을 고려
- 다수의 클라이언트 연결에 적합한 Publish/Subscribe 네트워크
- 신뢰성 있는 메시징을 위한 QoS(Quality of Service) 옵션 제공.
- 개방형 표준 메시징 프로토콜을 지향



# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : MQTT (2/7)

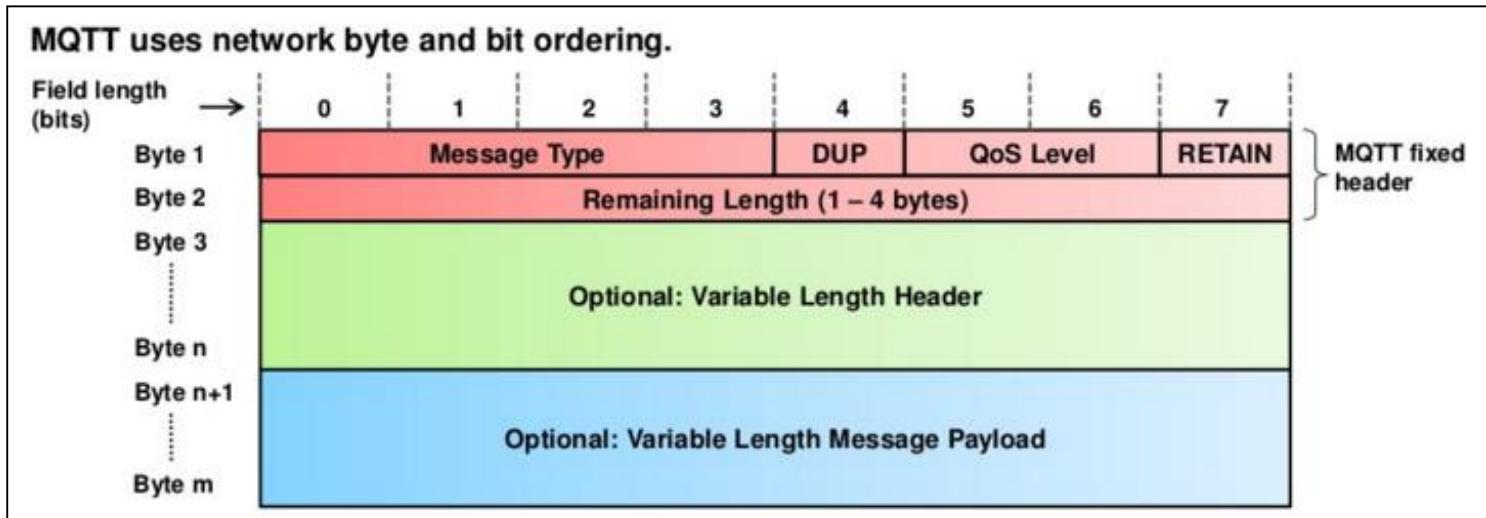
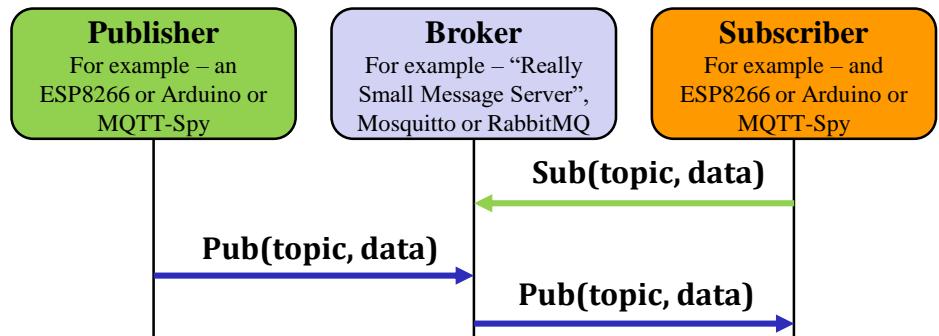
- MQTT(MQ Telemetry Transport)
  - Publish-Subscribe 기반의 경량 TCP/IP 프로토콜로써 통신 대역폭이 제한적인 사물인터넷에 적합하다.
  - Broker를 통하여 원하는 토픽의 메시지를 구독하고 발행한다.
  - IoT를 위해서 낮은 전력, 낮은 대역폭 환경에서도 사용할 수 있도록 설계됐다(저전력/소규모 디바이스를 위한 통신 프로토콜)



# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : MQTT (3/7)

- MQTT(Packet Structure)
  - Pub** : (Message Type = 3)
  - Sub** : (Message Type = 8)



- 기본적으로 2byte의 고정헤더로 이루어져 있고, 선택적으로 가변길이 헤더와 가변길이 메시지 Payload로 구성되어진다.

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : MQTT (4/7)

- MQTT(Header)

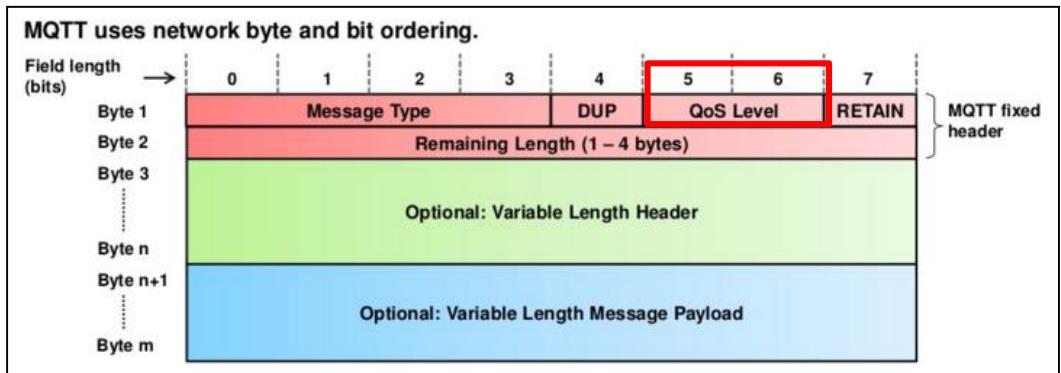
Message fixed header field	Description / Values
Message Type	0: Reserved 1: CONNECT 2: CONNACK <b>3: PUBLISH</b> 4: PUBACK 5: PUBREC 6: PUBREL 7: PUBCOMP 8: SUBSCRIBE 9: SUBACK 10: UNSUBSCRIBE 11: UNSUBACK 12: PINGREQ 13: PINGRESP 14: DISCONNECT 15: Reserved
DUP	QoS level 1~2에서 중복메시지를 보낼 때 사용
QoS Level	0: 메시지를 한번만 보냄. 유실 될 가능성 있음. 1: 메시지를 최소 1번 절달. 중복 전달 가능성 있음. 2: 메시지가 단 한 번, 정확하게 전달 <i>Further details see <a href="#">MQTT QoS</a>.</i>
RETAIN	Retain flag를 통하여 Broker에 메시지를 저장해 놓은 메시지를 사용할 수 있도록 구성 <i>Further details see <a href="#">RETAIN (keep last message)</a>.</i>
Remaining Length	Indicates the number of remaining bytes in the message, i.e. the length of the (optional) variable length header and (optional) payload. <i>Further details see <a href="#">Remaining length (RL)</a>.</i>

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : MQTT (5/7)

- QoS(Quality of Service):

통신서비스의 품질을 위해 네트워크에서 지연시간, 데이터 손실률 등을 보장하는 수준, 메시지 전달 보장이 증가되는 수준을 가리킨다.

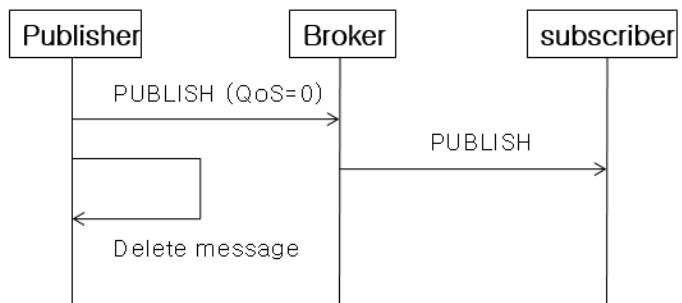


QoS Value	bit2	bit1	Description		
0	0	0	At most once	Fire and Forget	<=1
1	0	1	At least once	Acknowledged delivery	>=1
2	1	0	Exactly once	Assured delivery	=1

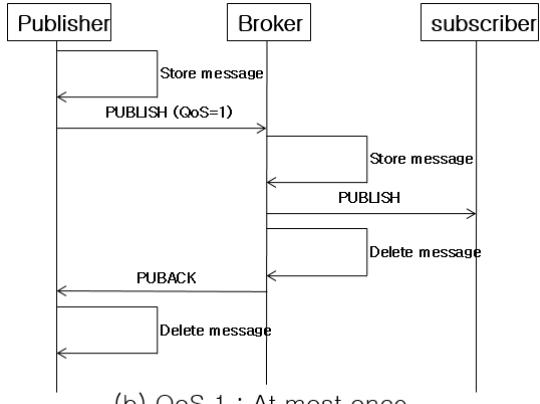
# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : MQTT (6/7)

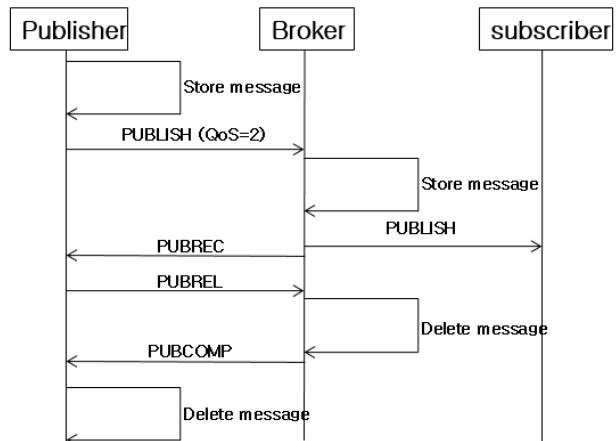
- MQTT(Header-QoS)



(a) QoS 0 : At most once (fire and forget)



(b) QoS 1 : At most once



(c) QoS 2 : Exactly once

### QoS(Quality of Service)

0:메시지는 한번만 전달하며, 전달 여부를 확인하지 않는다. **서버에서 응답을 요구하지 않으며, 재시도를 하지 않는다.**

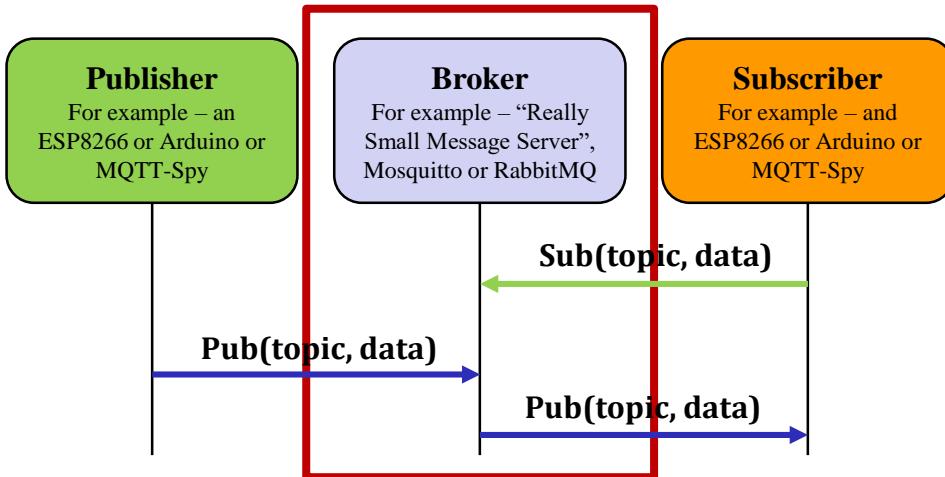
1:메시지는 반드시 한번 이상 전달된다. PUBACK 메시지에 의해 확인 응답이 가능하다. **지정된 시간 후에 수신 확인 메시지가 수신되지 않을 시 Client는 DUP 비트를 설정하여 다시 메시지를 보낸다.**

2:메시지는 한번만 전달된다. 메시지의 핸드셰이킹 과정을 추적한다. 높은 품질을 보장하지만 성능의 희생이 따른다(트래픽 증가).

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : MQTT (7/7)

- MQTT Broker

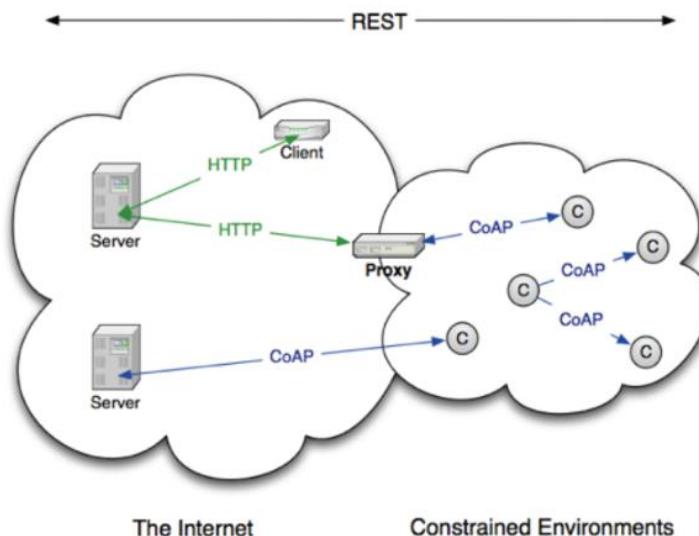


- MQTT Broker가 메시지 버스를 만들고 여기에 메시지를 흘려 보내면, 버스에 붙은 애플리케이션들이 메시지를 읽어가는 방식이다. 메시지 버스에는 다양한 주제의 메시지들이 흐를 수 있는데, 메시지를 구분하기 위해서 "Topic"을 이름으로 하는 메시지 채널을 만든다.
- 애플리케이션들은 Message Bus에 연결하고 관심있는 토픽(Topic)을 등록 해서 메시지를 구독(SUB)하거나 발행(PUB)한다.

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : CoAP (1/7)

- **Constrained Application Protocol (CoAP)**은 간단한 전자 기기들의 인터넷 통신을 지원하기 위해 만든 프로토콜이다. 특별히 저전력 센서, 스위치, 밸브 등의 기기를 표준적인 인터넷 환경에서 제어하기 위한 목적으로 만들어짐
- CoAP는 Wireless sensor network 노드들처럼 **제한된 자원의 인터넷 연결을 지원**
- CoAP는 **REST**와 같은 웹 애플리케이션 아키텍처의 적용이 수월하다. 또한 **UDP**을 사용하는데, IoT와 M2M 디바이스와 같은 환경에서 오버헤드를 줄일 수 있음



# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : CoAP (2/7)

- IoT의 중요성이 증가하면서 IETF는 경량 메시징을 위한 CoAP를 정의했음.
- IETF의 정의에 따르면, CoAP는 '제한된 노드와 제한된(즉, 저전력의 손실이 많은) 네트워크를 위한 것

주요 업체의 CoAP 구현 동향  
ARM



- ARM mbed과 사물인터넷 표준 솔루션
- 센서노드 인수, IETF CoAP을 포함한 국제 사물인터넷 표준화 활동에도 참여

(출처 : ARM, bloomberg.com)

주요 업체의 CoAP 구현 동향  
독일 브레멘(Bremen) 대학 TZI의 Carsten Bormann



[Carsten Bormann]

출처 : gogo6, gogo6.com/profile/CarstenBormann

- IETF CoAP 워킹그룹 총장으로써 표준화에 주도적인 활동
- CoAP 오픈 소스인 Libcoap 라이브러리를 개발
- 학계 연구소 차원에서 표준화, CoAP 및 사물인터넷 기술 보급

CoAP 구현 동향 및 주요 업체

주요 업체의 CoAP 구현 동향  
스위스 취리히 연방 공과대학의 ETH



[취리히 연방 공과대학]

- 수준 높은 CoAP 구현 소프트웨어 기술 보유

(출처 : de.wikipedia.org/wiki/ETH\_Z%C3%BCrich)

주요 업체의 CoAP 구현 동향  
벨기에의 iMinds



- 사물인터넷, CoAP 기술 구현에 참여
- ICT 솔루션 등 13개의 최신 트랜드 프로젝트 수행
- 임베디드 서비스를 타깃으로 하는 창업자를 위한 CoAP 프로젝트 구성

(출처 : iminds, iminds.be)

# 1. Smart Home Energy Framework :

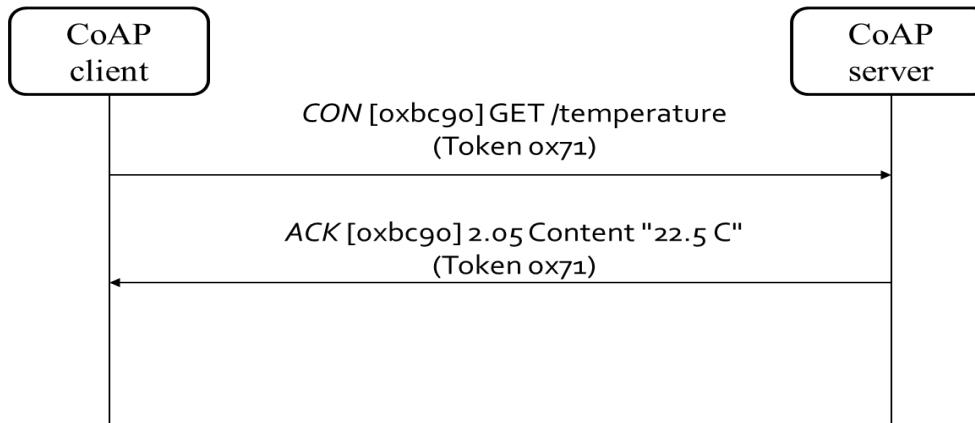
## 1.1.4 Background IoT 수요반응 프로토콜 : CoAP (3/7)

---

- CoAP는 클라이언트/서버 프로토콜이며, 일대일(1:1) ‘요청/보고’ 인터랙티브 모델을 제공한다. 또한 아직 IETF 표준화의 초기 단계에 있지만 멀티캐스트를 지원한다.
- 십여 년 전 개발된 프로토콜로부터 IoT의 요구에 맞추어 개조된 MQTT와 달리 CoAP는 IETF가 처음부터 제한된 환경에서 동작하는 제한된 기기의 경량 메시징의 IoT를 지원하기 위해 만들어졌다.
- CoAP는 간단한 프록시를 통해 HTTP와 RESTful 웹과 상호 운용되도록 설계되어 있어 본질적으로 인터넷에 적합하다

# 1. Smart Home Energy Framework :

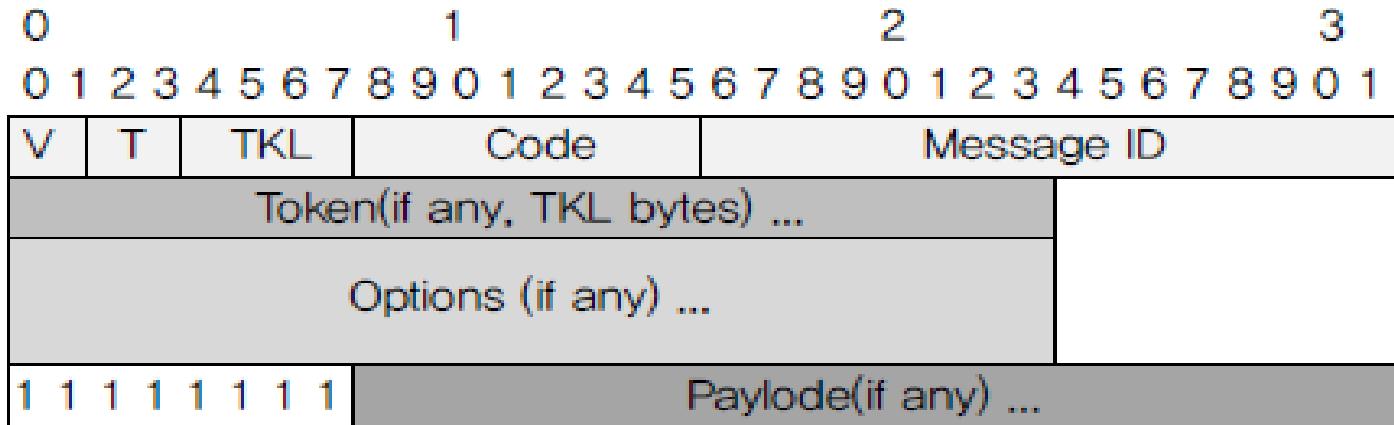
## 1.1.4 Background IoT 수요반응 프로토콜 : CoAP (4/7)



- 1) 임베디드 웹 전송 메커니즘 프로토콜이다. (coap://)
- 2) 비동기적(Asynchronous) 전송 방식을 따른다.
- 3) UDP 기반의 통신
- 4) GET, POST, PUT, DELETE 방식 지원(RESTful 방식)
- 5) URI 지원
- 6) 작고 단순화된 헤더를 사용한다. (최소 5byte 헤더)

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : CoAP (5/7)



**V**(0~1bit) :CoAP의 version, 현재 버전은 01이다.

**T**(2~3bit) : Type을 의미. 0은 확인형(confirmable), 1은 비확인형(non-confirmable), 2는 승인(ACK), 3은 리셋(RESET)을 의미. 보통은 0을 실어 보냄으로써 UDP이지만 신뢰성있는 송수신을 할 수 있도록 한다.

**TKL** : 토큰의 길이를 나타냄. 단위는 바이트이며, 0에서 8까지의 값을 사용. TKL이 0보다 크면 헤더 다음으로 토큰 값이 온다.

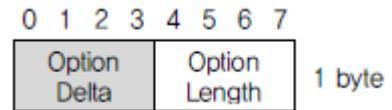
**Code** : 3bit는 class, 5bit는 detail을 의미. class의 종류는 요청(0), 성공적인 응답(2), 클라이언트 에러 응답(4), 서버 에러 응답(5)이 있고, detail은 GET(0.01), POST(0.02), PUT(0.03), DELETE(0.04)가 정의되어 있다. (ex. 4.04 is NOT FOUND)

**Message ID** : Message의 중복 확인을 위한 필드.

**Options** : 옵션 포맷은 오른쪽 그림과 같다.

**11111111(0xFF)** : 페이로드 마커로써 더 이상 옵션이 없는 상태를 나타낸다.

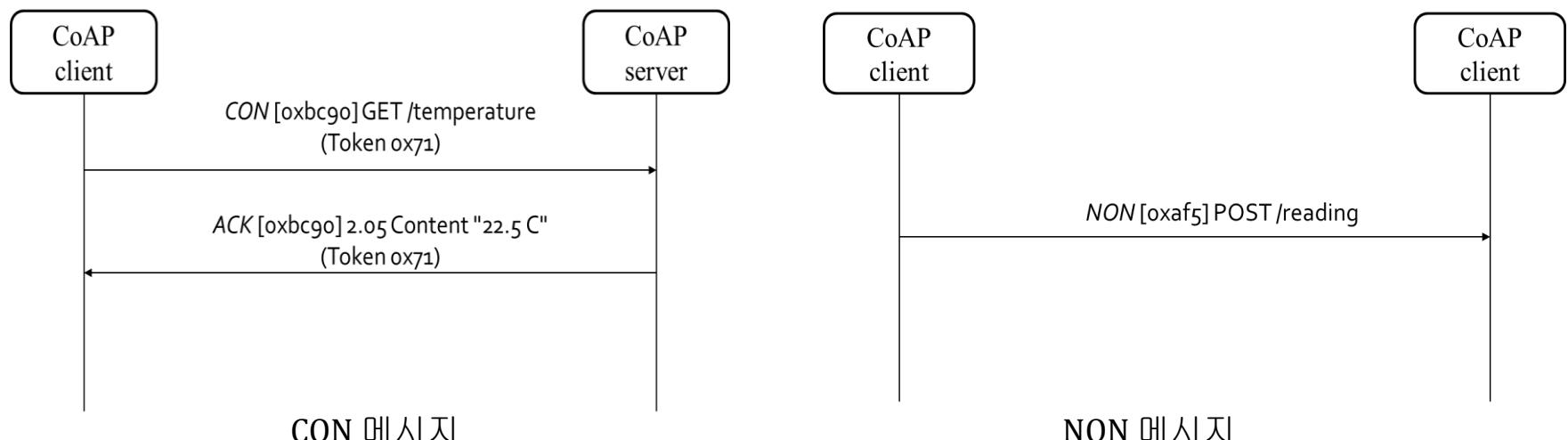
**Payload** : datagram의 끝까지 위치한다.



# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : CoAP (6/7)

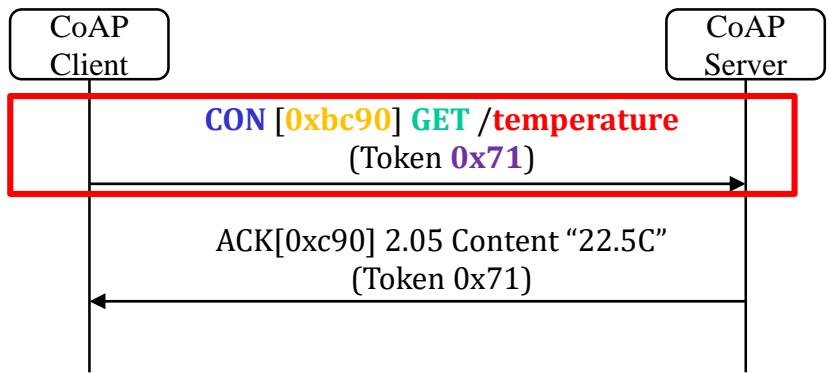
- CoAP는 기본적으로 UDP 프로토콜을 기반으로 하기 때문에 신뢰성이 적은 데이터 송수신을 할 것이라고 생각할 수 있음
- CoAP에서는 신뢰성 있는 송수신을 위해서 Type을 설정해서 보낼 수 있음.
- 아래 그림은 CoAP에서 제공되는 신뢰성 있는 전송과 신뢰성이 없는 전송의 예



# 1. Smart Home Energy Framework :

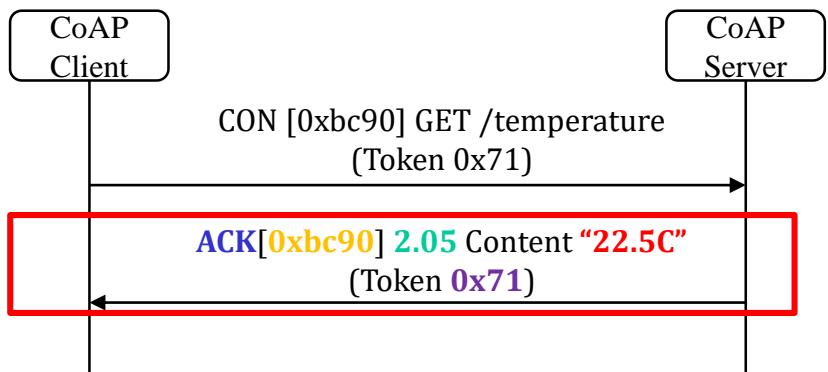
## 1.1.4 Background IoT 수요반응 프로토콜 : CoAP (7/7)

CoAP request message/response message는 아래 그림과 같이 인코딩 된다.



0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	0	1																			0xbc90
			GET = 1																		
			0x 71																		
				11	11																
																					"temperature"(11 B)

▲ Request



0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1	2	1																			0xbc90
			2.05=69																		
			0x 71																		
				11	11																
																					"22.5 C"(6B)

▲ Reply

# 1. Smart Home Energy Framework :

## 1.1.4 Background IoT 수요반응 프로토콜 : CoAP

### IoT 환경의 IoT 프로토콜 MQTT, CoAP 비교

#### MQTT

- MQTT는 **Publish/Subscribe**를 기반으로 중앙에 있는 브로커와 연결됨
- 메시지는 **브로커**로 모여서 Subscriber에게 메시지를 복사해서 전달
- 지속적인 연결**을 지원
- 실시간 데이터를 전달**하는데 적당함

#### CoAP

- 기본적으로 하나의 서버와 하나의 클라이언트가 참여하는 **1:1 프로토콜**
- 자원을 발견, 관찰 할 수 있기는 하지만 **이벤트 기반 데이터 통신에는 적당하지 않음**
- 분산환경에서의 상태정보에 적당한** 프로토콜

- TCP/IP 기반 **프로토콜**로 서버(브로커)와 지속적인 연결을 맺기 때문에 **NAT와 같은 환경에서 문제없이 작동함**

- UDP 패킷만을 사용**하기 때문에, NAT 환경에서 사용하기 힘듬
- NAT 환경에서 사용하려면 터널링 하거나 포트 포워딩(Port Forwarding)을 해야함

- 서비스 발견 기능이 없기 때문에 **DNS-SD, SSDP의 도움이 필요함**

- 서비스 발견 기능을 기본으로 지원**하기 때문에, 어떤 형식의 데이터를 사용해야 할지를 알고 있음

# 1. Smart Home Energy Framework :

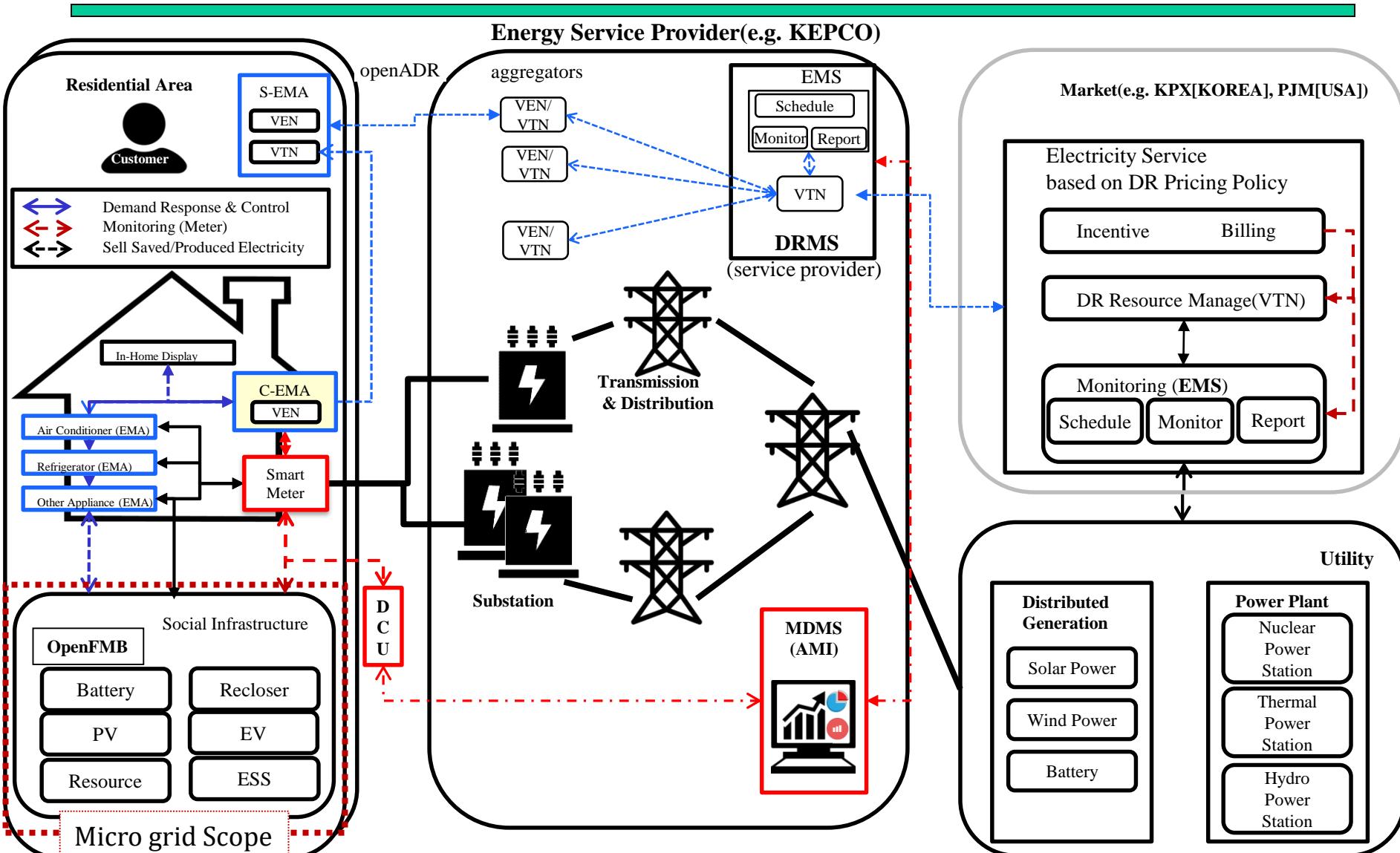
## 1.1.4 Background Microgrid (1/2)

---

- 마이크로그리드(Microgrid)는 기존의 광역적 전력시스템으로부터 독립된 분산전원을 중심으로 한 국소적인 전력공급시스템을 말하는 것으로, 기존 전력시스템과 상호보완적인 관계를 가지고 있다.
- 기존의 전력시스템은 발전소에서 생산된 전기를 소비자에게 전달하는 단방향 구성이었다. 하지만 기존의 소비자 중에서 직접 전기를 생산하여 공급자 역할까지 하는 프로슈머가 등장하였다. 기존의 전력시스템에서 이들은 자급자족만 하였고, 전체 계통망에는 기여하지 않았다. 생산되고 남은 전기는 대부분 버릴 수밖에 없어 효율성이 떨어졌다.
- 마이크로그리드는 이들이 생산하는 전기에너지를 활용하여 전체 네트워크의 에너지를 극대화 시키기 위한 기술로, 발전소에서만 전기를 생산하는 것이 아니라 양방향 송배전을 바탕으로 다수의 프로슈머가 전력망의 전력생산을 맡게 되는 것이다. 전원이 분산됨에 따라 안정적인 전기공급이 가능해지고, 재생가능 에너지의 효율적인 이용도 가능할 것으로 예상된다.

# 1. Smart Home Energy Framework :

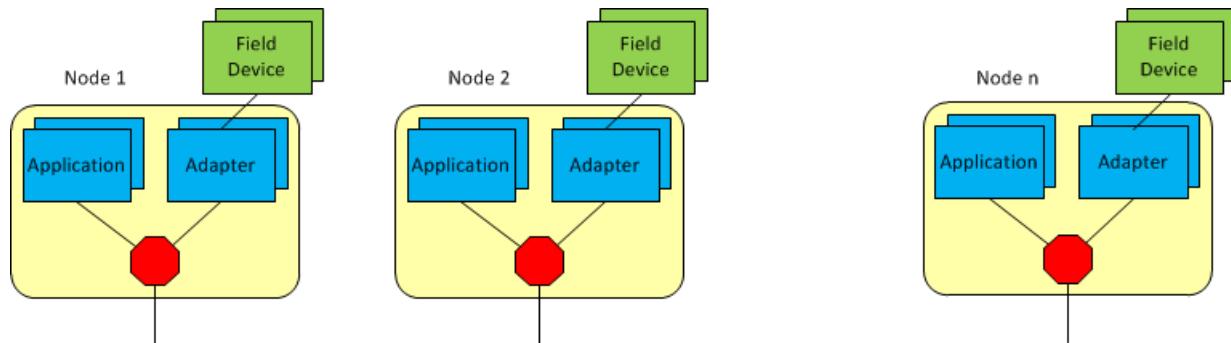
## 1.1.1 Background Microgrid (2/2)



# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (1/12)

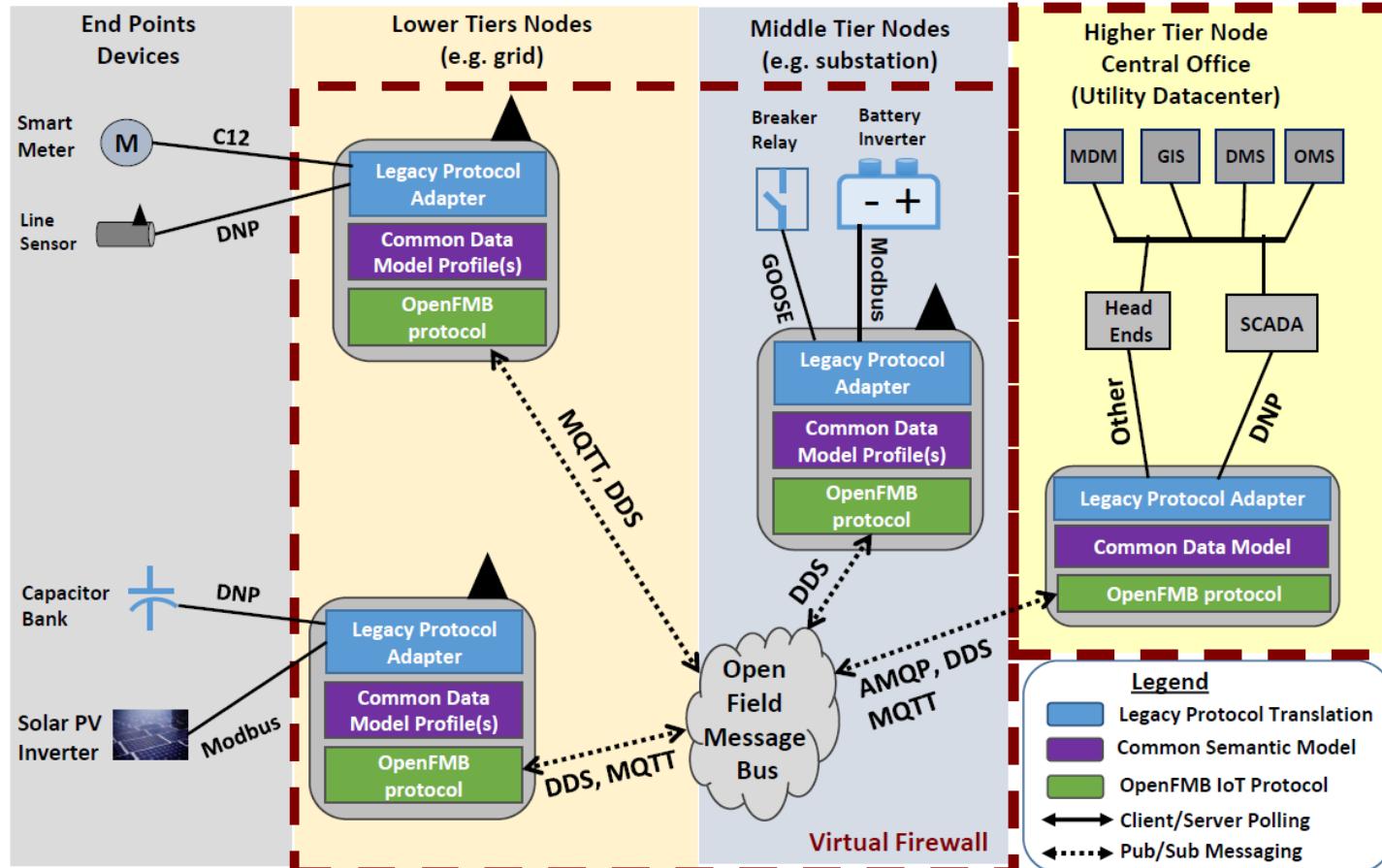
- OpenFMB는 그리드 말단의 **상호운용성**(interoperability)과 **분산정보**(distributed intelligence)를 위한 프레임워크이자 참고 아키텍쳐이다.
- 이것은 그리드edge에서 필드장치와 시스템의 peer-to-peer 메시징을 통해 상호 작용하는 분산 지능형 노드를 위한 프레임워크이다.
- 기존 기술과 새로운 기술 간의 **상호 운용성을** 향상시키기 위해 각 옵션에 대해 독립적으로 개발된 구현을 적어도 두 가지 사용하여 다양한 운영기능에 대한 여러 기술 옵션을 강조하였다.



# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (2/12)

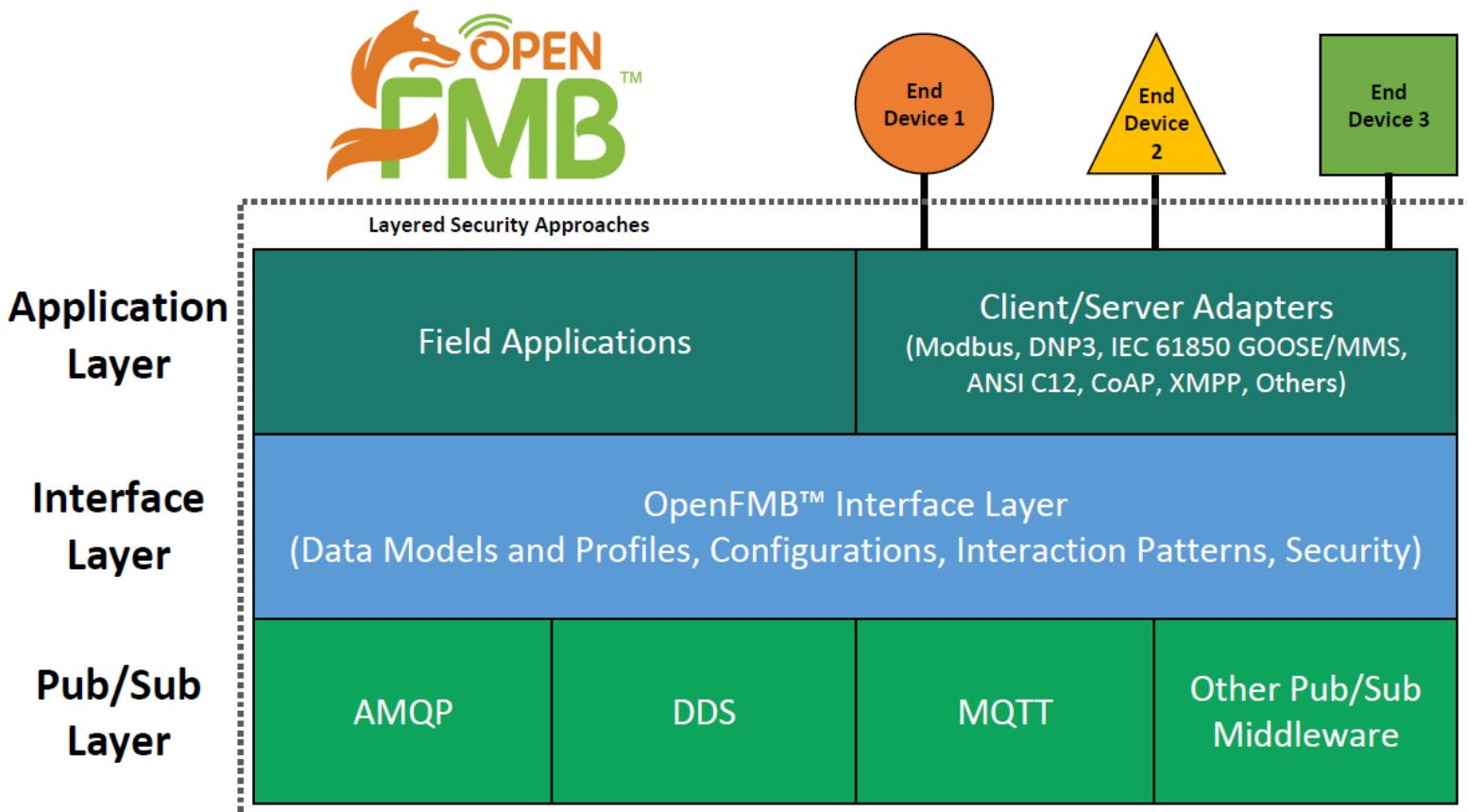
### OpenFMB™: Layered Architecture Framework



# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (3/12)

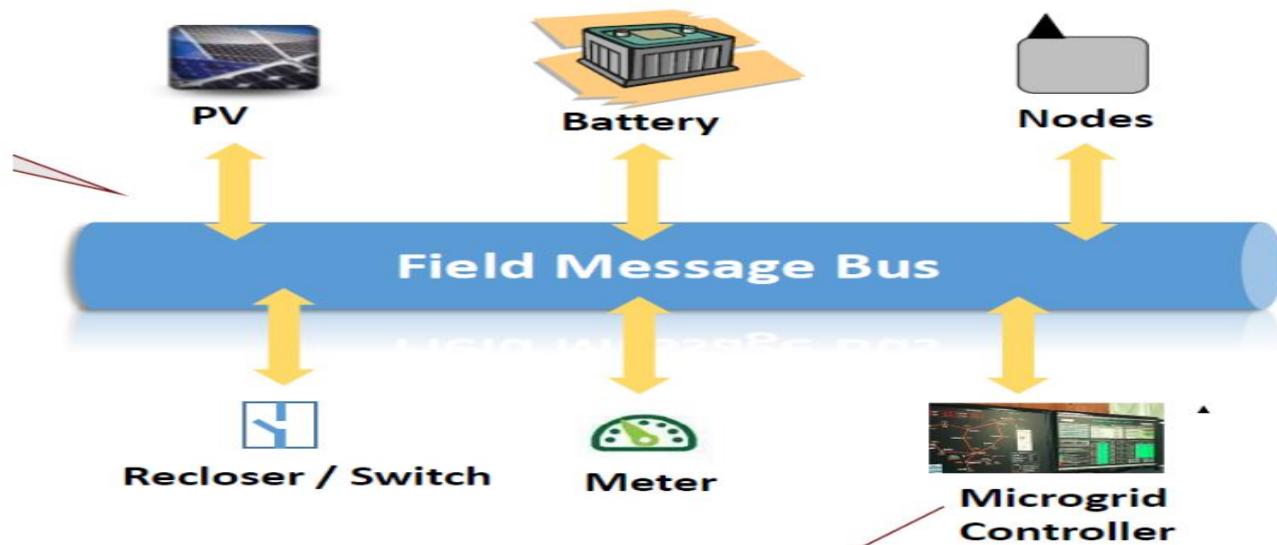
### OpenFMB™: Logical Reference Architecture



# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (4/12)

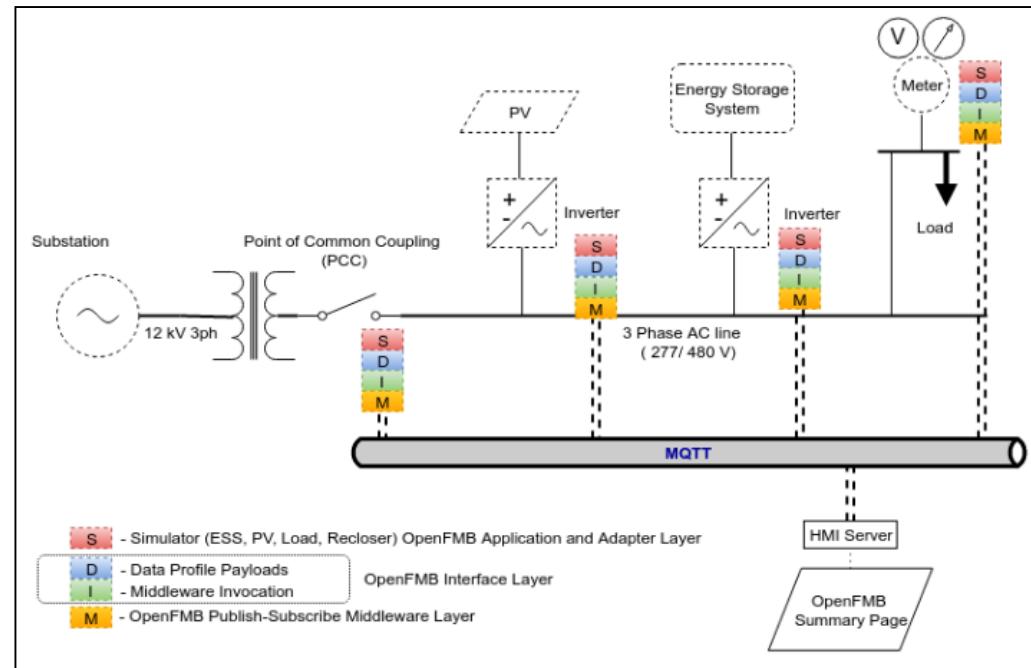
- OpenFMB architecture integrates subsystems and applications through a central, real-time publish-subscribe bus.



# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (5/12)

- Simulation Demo consist of following simulators



- PV simulator
- ESS simulator
- Load simulator
- PCC simulator
- PCC application to manage the resource
- The Visualization of the system (Web based UI)

# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (6/12)

- HMI Server

The screenshot shows the OpenFMB 2016 Microgrid UI running in a web browser on a Linux desktop. The interface includes a sidebar with various icons and a main content area with three main sections: Microgrid Summary, Device Profiles, and Resource details.

**Microgrid Summary:**

Status	Grid Connected
Resource (non-controllable load)	144.7 kW
Local Generation	-83.9 kW
Grid (export neg.)	60.8 kW

Solar	-83.9 kW
Battery	0.0 kW
Local Generation (total)	-83.9 kW

**Device Profiles (events & readings):**

Recloser: DEMO.MGRID.RECLOSER.1	
Is Closed	true
Control	Trip Close
Power Flow (export neg.)	60.8 kW
Voltage	277.1 V
Frequency	59.99 Hz

Battery: DEMO.MGRID.BATTERY.1	
Mode	Maintain Minimum Battery SoC
Power Setpoint	kW
SOC	50.0 %
Is Charging	true
Power (pos. charging)	0.0 kW
Voltage	276.9 V
Frequency	59.99 Hz

Solar: DEMO.MGRID.SOLAR.1	
Power	-83.9 kW

Resource: DEMO.MGRID.RESOURCE.1	
Power	144.7 kW

# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (7/12)

- Control - Trip

### Microgrid Summary

Status	Islanded
Resource (non-controllable load)	138.9 kW
Local Generation	-138.9 kW
Grid (export neg.)	0.0 kW

Solar	-83.9 kW
Battery	-55.0 kW
Local Generation (total)	-138.9 kW

### Device Profiles (events & readings)

Recloser: DEMO.MGRID.RECLOSER.1	
Is Closed	false
Control	<button>Trip</button> <button>Close</button>
Power Flow (export neg.)	0.0 kW
Voltage	276.9 V
Frequency	60.02 Hz

Battery: DEMO.MGRID.BATTERY.1	
Mode	Islanded
Power Setpoint	kW
SOC	49.8 %
Is Charging	false
Power (pos. charging)	-55.0 kW
Voltage	277.1 V
Frequency	60.01 Hz

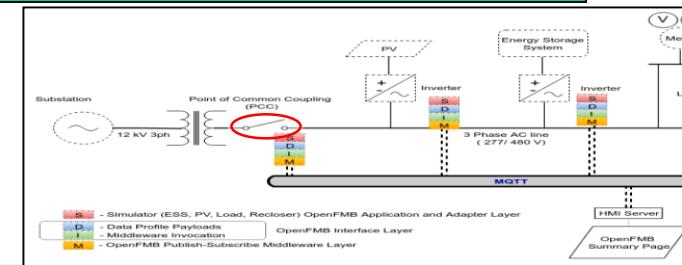
# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (8/12)

- Control - Close

### Microgrid Summary

Status	Grid Connected
Resource (non-controllable load)	139.1 kW
Local Generation	167.4 kW
Grid (export neg.)	306.5 kW



### Device Profiles (events & readings)

Recloser: DEMO.MGRID.RECLOSER.1	
Is Closed	true
Control	<button>Trip</button> <button>Close</button>
Power Flow (export neg.)	306.5 kW
Voltage	277.0 V
Frequency	60.03 Hz

Battery: DEMO.MGRID.BATTERY.1	
Mode	Maintain Minimum Battery SoC <input checked="" type="checkbox"/>
Power Setpoint	kW <input checked="" type="checkbox"/>
SOC	49.9 %
Is Charging	true
Power (pos. charging)	250.0 kW
Voltage	277.1 V
Frequency	60.01 Hz

# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (9/12)

- Battery(ES)
  - batterysim.properties

```
device.logicalDeviceID=DEMO.MGRID.BATTERY.1
device.mRID=DEMO.MGRID.BATTERY.1
device.name="Battery 250 kW"
device.description="Battery 250 kW"

topic.BatteryReadingProfile=openfmb/batterymodule/BatteryReadingProfile
topic.BatteryEventProfile=openfmb/batterymodule/BatteryEventProfile
topic.BatteryControlProfile=openfmb/batterymodule/BatteryControlProfile

battery.maxChargeRatekW=250
battery.maxDischargeRatekW=250
battery.energyRatingkWh=250
battery.energyMaxkWh=237.5
battery.energyMinKWh=37.5
battery.efficiencyRatio=0.8
battery.voltage=277
battery.hertz=60

config.intervalMs=2000
```

# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB(Field Message Bus) (10/12)

- Solar(PV)
  - solarsim.properties

```
device.logicalDeviceID=DEMO.MGRID.SOLAR.1
device.mRID=DEMO.MGRID.SOLAR.1
device.name=Solar
device.description=Solar

value.scale=0.01
value.offset=0.0
value.jitterChance=0.2
value.jitterPercent=0.05

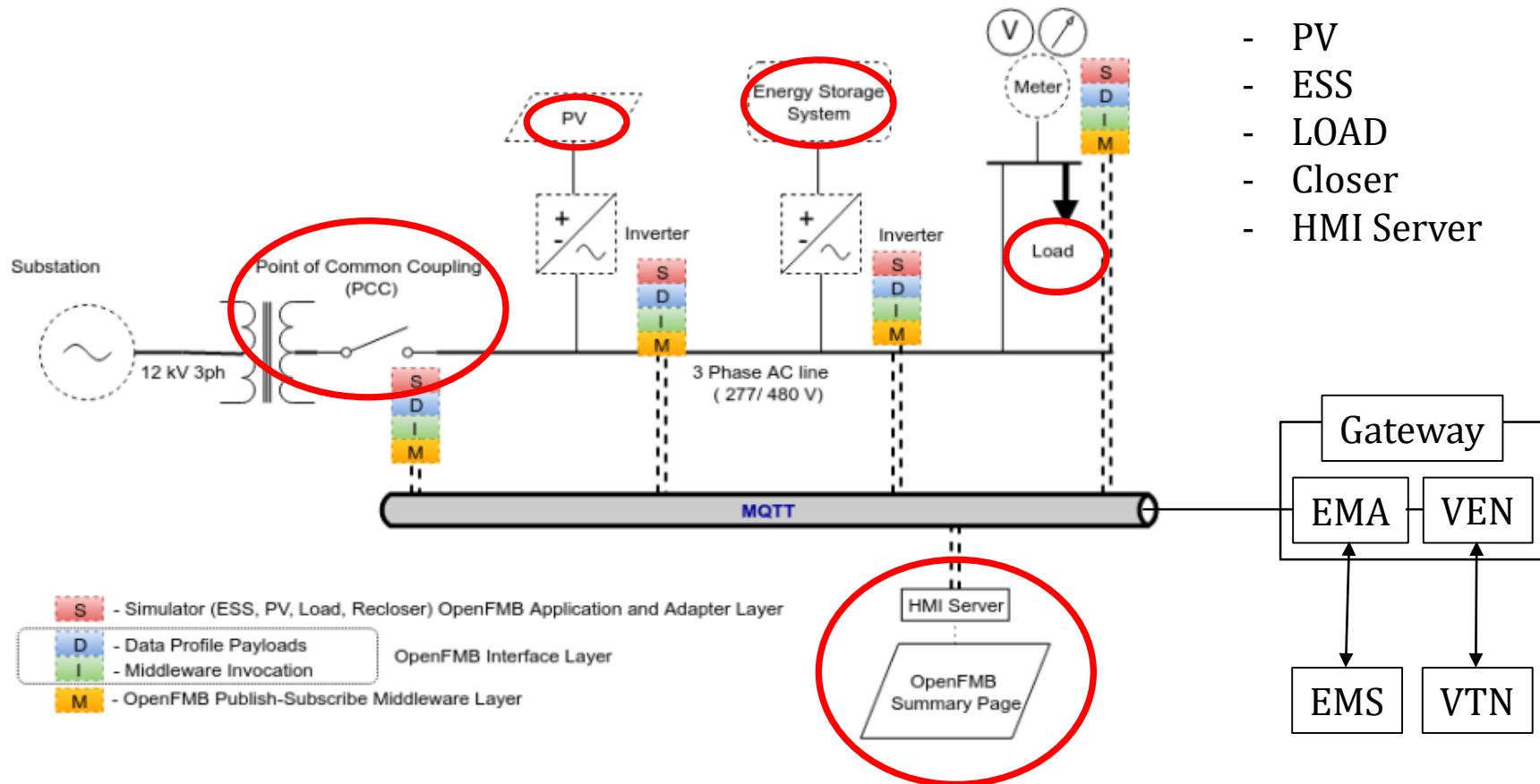
topic.SolarReadingProfile=openfmb/solarmodule/SolarReadingProfile
topic.SolarEventProfile=openfmb/solarmodule/SolarEventProfile

data.file=data/load.tsv

config.intervalMs=2000
```

# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB (Field Message Bus) (11/12)

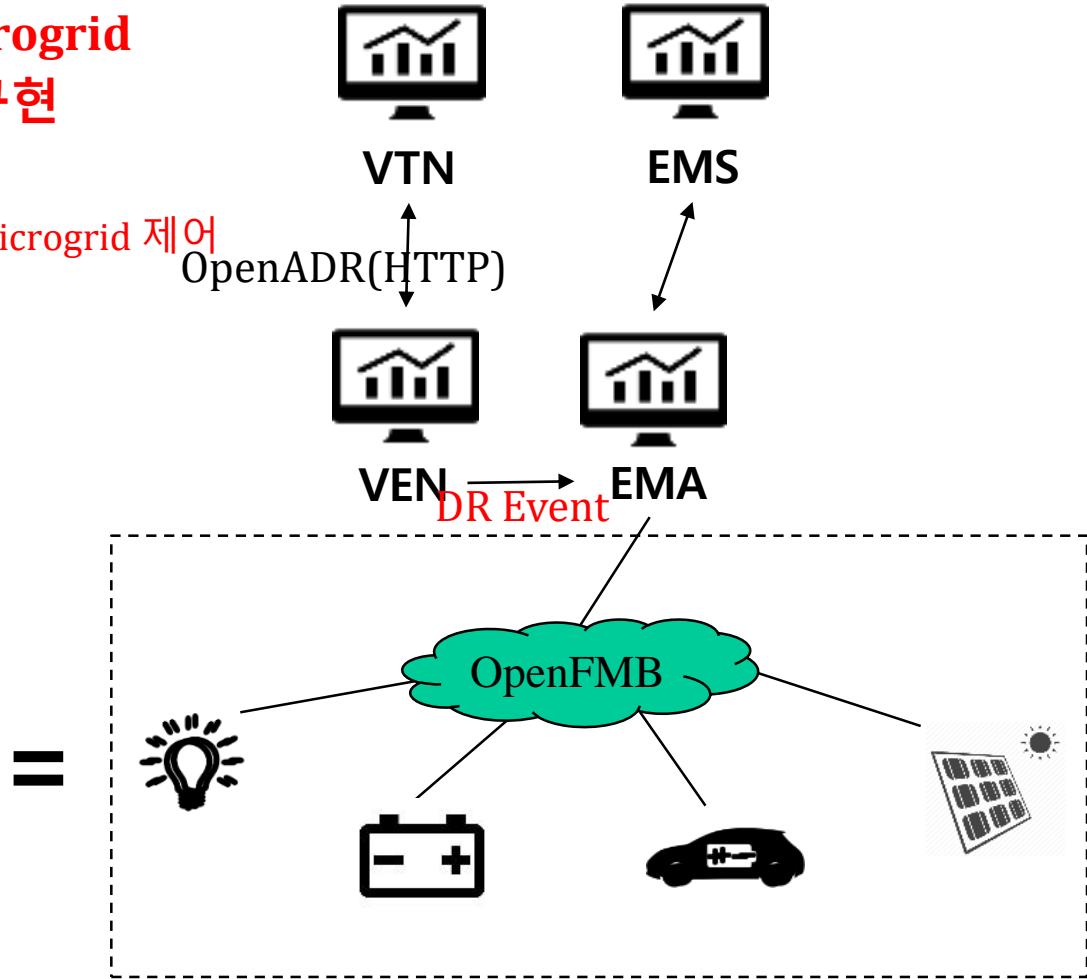


# 1. Smart Home Energy Framework :

## 1.1.1 Background Microgrid : OpenFMB (Field Message Bus) : Architecture (12/12)

- OpenFMB기반 Microgrid  
수요반응 시스템 구현

- DR이벤트에 따른 Microgrid 제어

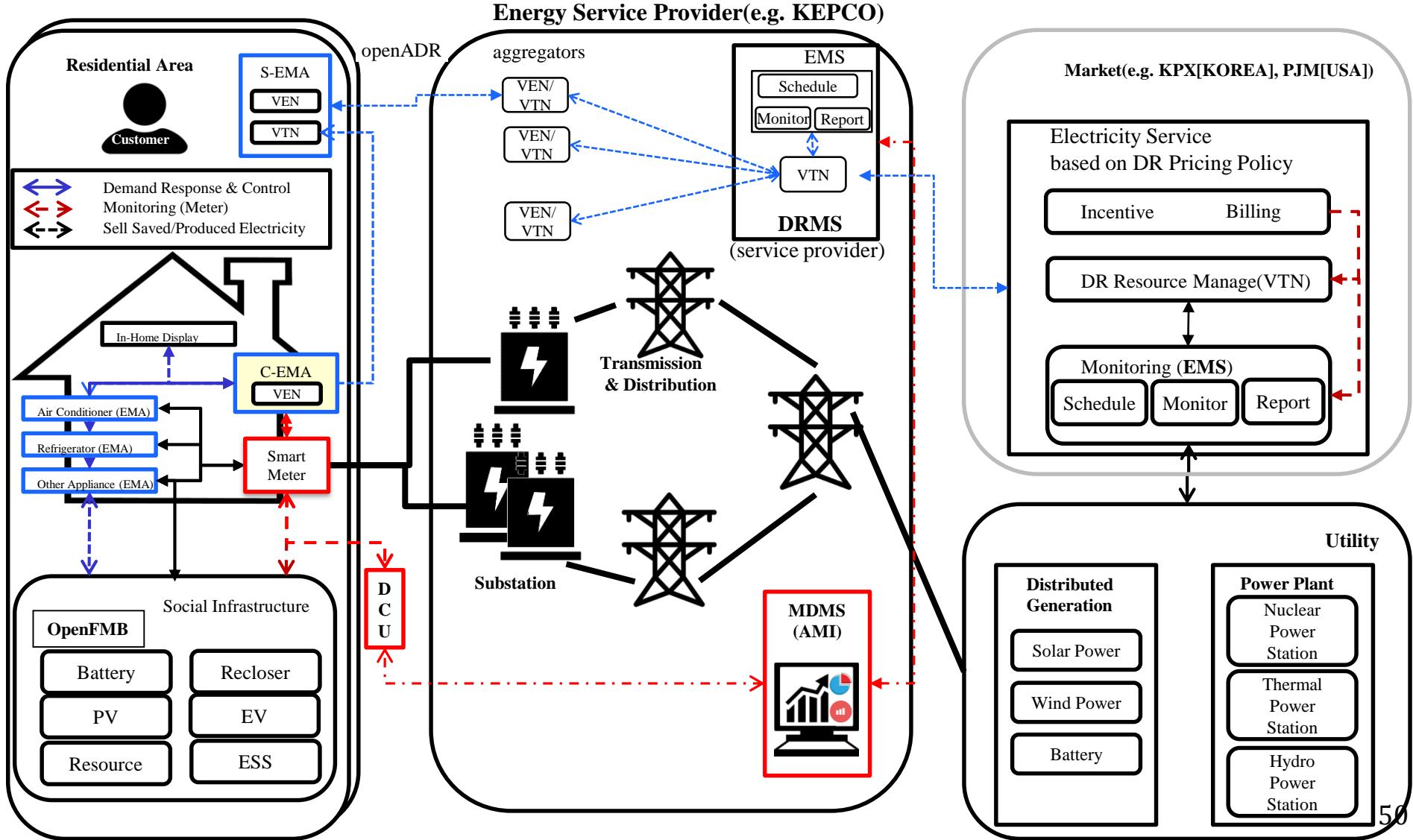


# **EnergyFramework**

## **1.2 Architecture**

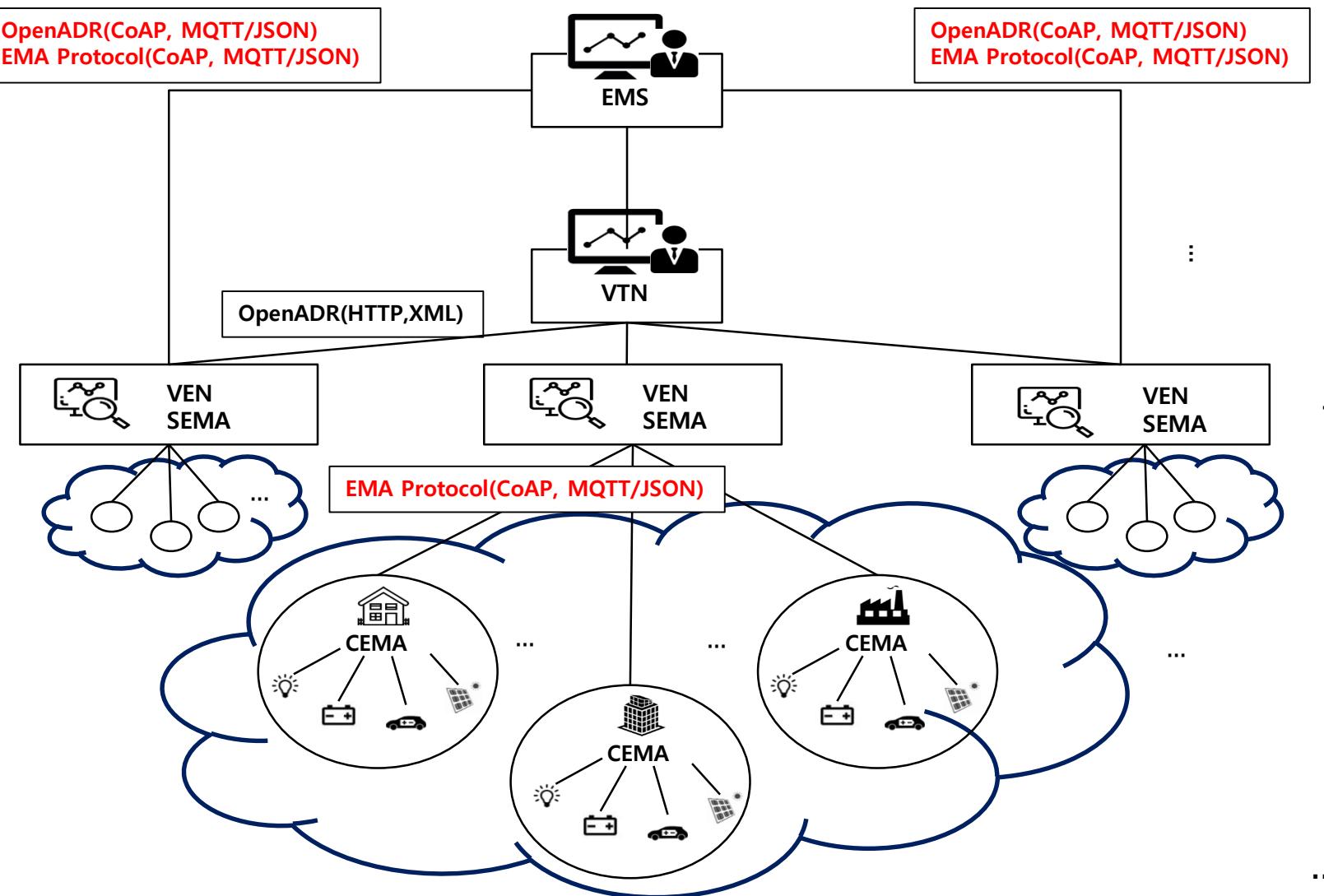
# 1. Smart Home Energy Framework :

## 1.2 Architecture (1/2)



# 1. Smart Home Energy Framework :

## 1.2 Architecture (2/2)

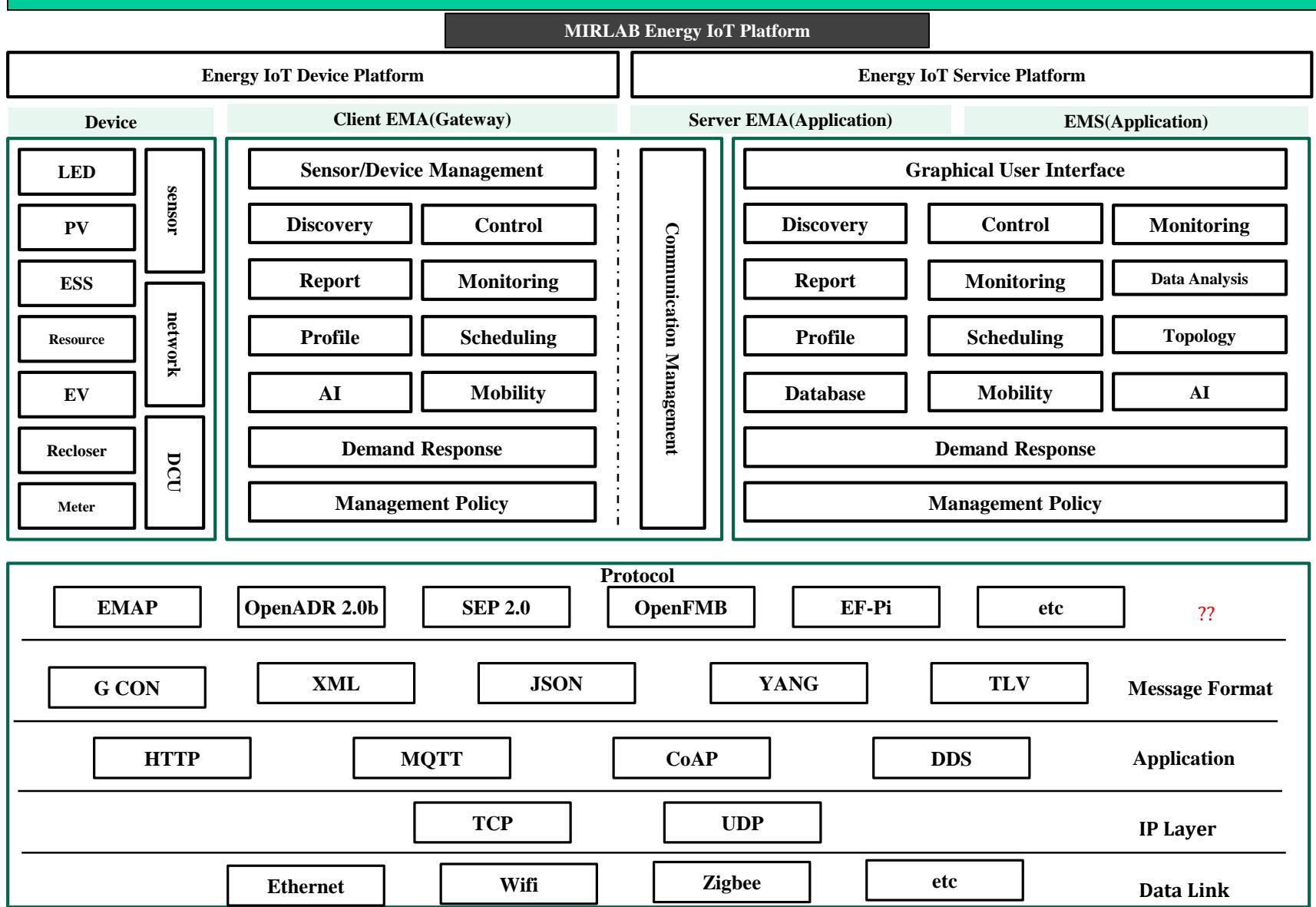


# **EnergyFramework**

## **1.3 Block Diagram**

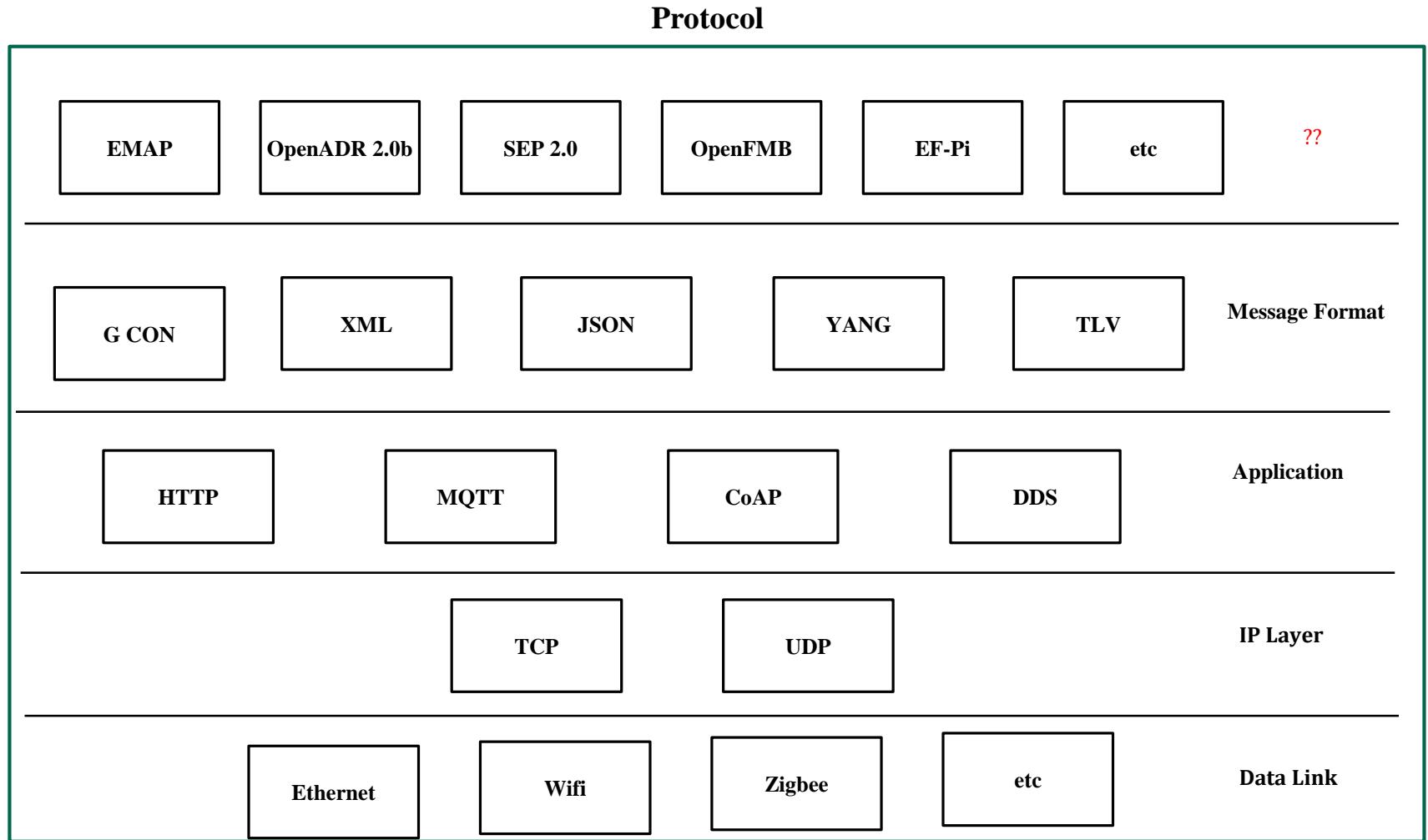
# 1. Smart Home Energy Framework :

## 1.3 Block Diagram : Overview (1/3)



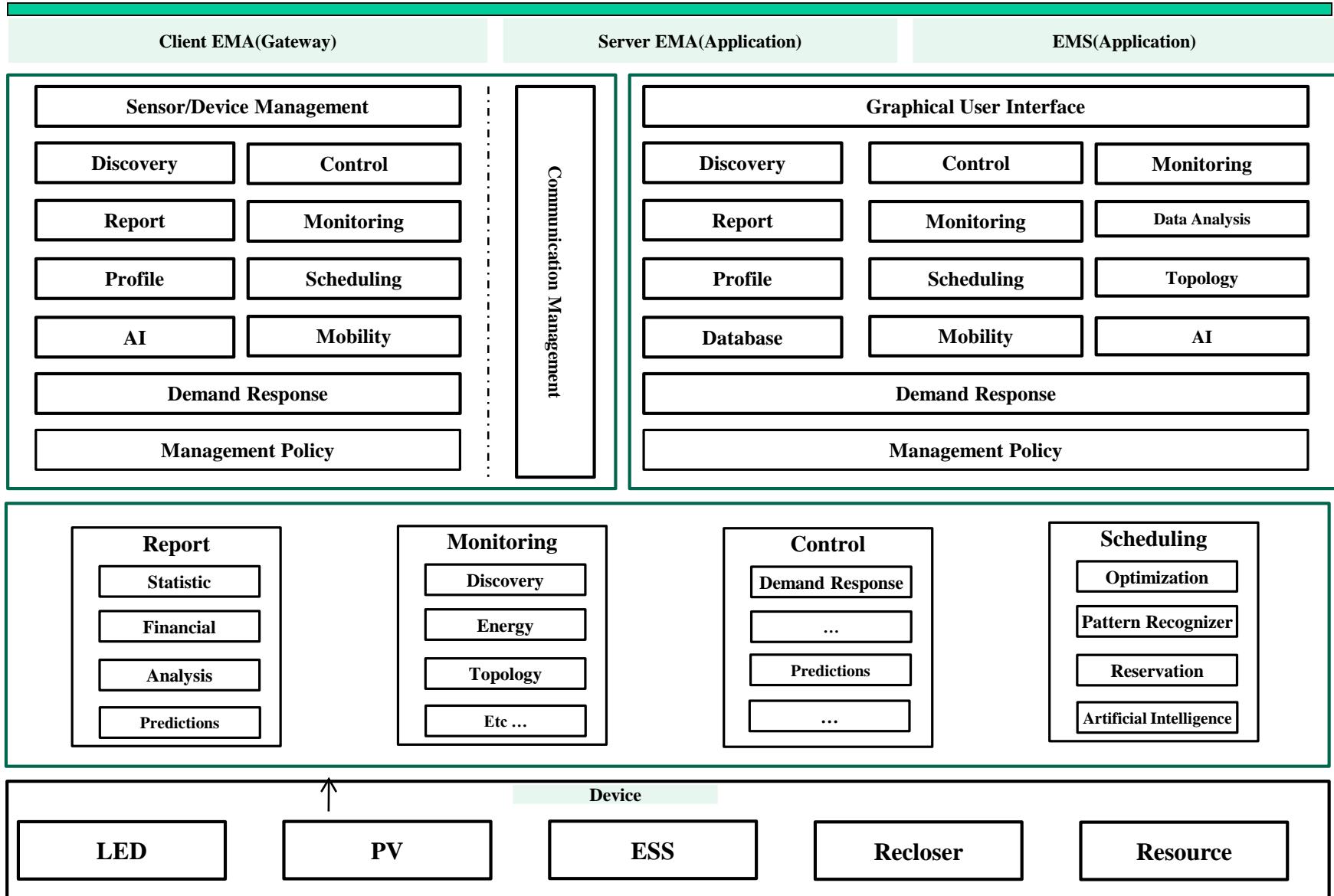
# 1. Smart Home Energy Framework :

## 1.3 Block Diagram : Protocol (2/3)



# 1. Smart Home Energy Framework :

## 1.3 Block Diagram : Function (3/3)



# **EnergyFramework**

## **1.4 참고문헌 논문 정리**

# Smart Home Energy Framework :

## 1.4 참고문헌 - 수요반응 및 스마트 그리드 기술에 관한 연구 (1/3)

제목	내용 및 범위	결론
스마트그리드에서 블랙아웃을 방지하기 위한 에너지 스케줄링 스키마에 의한 성능분석(델핀)	<p>블랙아웃 문제점과 화석 연료가 부족함  <b>→EMS를 통해서 에너지 수요 관리</b>          블랙아웃을 방지하기 위해 3가지 솔루션을 제시  <ul style="list-style-type: none"> <li>- Negotiation Method</li> <li>- ESS(Energy Storage System)</li> <li>- Shiftable Appliance</li> </ul> </p>	<p><b>ESS</b>를 사용하면 전력 수요가 사용 가능한 임계 값보다 높을 때 배터리에 저장된 전력 저장소를 사용할 수 있다.          수요가 높을 때 한 개 이상의 Shiftable Appliance이 가동되어야 할 때, 다른 <b>Shiftable Appliance</b>에 가용 전력을 할당 할 수 있다.(차단 값이용)  <b>협상 방법</b>을 구현할 때 각 게이트웨이 동적 임계 값은 사용자 전력 요구량과 다른 사용자의 전력 소비량에 따라 변경한다.</p>
OpenADR 기반의 실시간 VEN 스마트 게이트웨이 프로토콜 개발과 에너지 최적화 (장병욱)	<p>OpenWRT에 VEN과 EMA을 올려서          EMS, VTN과 <b>수요관리 및 디바이스 관리</b>  <b>에너지 최적화</b>를 하기 위해서 우선순위 게임이론을 활용함.          (DR서비스 환경이 단순히 전력사업자와 제공자에서 벗어나 사용자 에너지 만족도나 사용량을 관리 할 수 있는 통합서비스를 제공해주는 기기로 삼을 수 있다.)</p>	<p>스마트 게이트웨이를 활용하여 각각의 시간별 무작위 확률에 디바이스 요청이 이루어지도록 실험하였다. 결과는 우선 순위선택 보다 게임이론이 사용요구량에 맞춰 사용량이 조금 더 차별화할 수 있다는 결과를 얻음  <b>DR환경에서의 스마트 게이트웨이를 구현하고 이에 대한 에너지 스케줄링 환경을 만들어 실제 DR환경에서 제공하는 서비스 외에 사용자가 원하는 서비스 환경을 개선할 수 있었다.</b></p>
에너지 IoT 환경에서 효율적인 에너지 관리를 위한 분산형 스마트 홈 에너지관리시스템 구현 및 분석에 대한 연구(김세영)	<p>기존의 중앙집중형 에너지관리시스템은 <b>에너지 IoT 환경과 같이 시스템</b>이 소형화, 경량화 되는 환경에서 연결의 증가와 함께 서버의 과부화, 데이터 병목 현상과 같은 원인에 의해 서비스 지연을 초래하여 실시간성이 보장되지 않을 수 있음          -VTN과 EMS이 분리된 분산형 에너지시스템 제안(EMS, EMA 1:1, 1:N 구조 비교, CoAP/JSON 이용)</p>	<p>에너지 IoT 환경과 같이 초연결 사회로 진행되고 전력 시스템이 소형, 경량화 되고 있는 사회에서 <b>분산형의 에너지관리 시스템 구조는 데이터 분산을 통해 에너지 관리의 실시간성을 보장하여 보다 더 효율적인 에너지 관리</b>에 기여할 것으로 기대</p>
주거 공간 에너지 수요관리를 위한 MQTT기반의 수요반응 프로토콜 설계 및 구현(강성철)	<p><b>주거 공간에서의 수요반응 서비스</b>를 위한 점 대 다점 (Multicast) 방식의 경량 (Lightweight) 수요반응 프로토콜을 제안하고 IoT 메시징 프로토콜 중 하나인 <b>MQTT기반으로 수요반응 프로토콜 구현 및 비교</b></p>	<p>Multicast기반의 Lightweight 수요반응 프로토콜을 이용하면 <b>주거 공간의 다수의 수요자들과의 통신 및 소형 네트워크 기기를 갖는 에너지 IoT 환경에서 원활한 수요반응 자동화 서비스가 널리 활용될 것으로 기대된다.</b></p>
스마트 에너지 IoT를 위한 CoAP 기반 Lightweight OpenADR2.0b 프로토콜의 구현 및 분석 (한국통신학회)	<p><b>스마트 홈 에너지 IoT 환경</b>에서는 많은 수의 초경량 디바이스들이 연결되기 때문에 기존의 HTTP/XML 기반의 OpenADR 수요반응 프로토콜보다 경량의 메시지를 이용한 수요반응 프로토콜이 필요하다. <b>Smart Energy IoT 환경에서 수요반응 서비스를 제공하기 위한 경량의 CoAP/JSON 프로토콜에 기반 한 경량화된 OpenADR 프로토콜</b>을 제안하고 기존의 HTTP/XML 형식의 프로토콜과 성능을 비교 및 검증하였다.</p>	<p><b>에너지 IoT환경</b>의 소형 네트워크 기기에서 데이터 트래픽과 응답시간은 다양한 제어 서비스로 확장되어가고 있는 에너지 IoT서비스에서 중요한 요소가 된다. <b>경량 프로토콜과 데이터 포맷을 이용하면 소형 네트워크 기기를 갖는 에너지 IoT 환경에서 원활한 수요반응 자동화 서비스가 널리 활용될 것으로 기대된다.</b></p>

# Smart Home Energy Framework :

## 1.4 참고문헌 - 수요반응 및 스마트 그리드 기술에 관한 연구 (2/3)

제목	내용 및 범위	결론
Implementation Challenges of Automatic Demand Response for Households in Smart Grids <i>(주거공간까지 왜 DR을 확장하는가?)</i>	<p><b>OpenADR Benefits의 이점들 서술</b></p> <ul style="list-style-type: none"> <li>-Customer control</li> <li>-Supporting curtailment through aggregation</li> <li>-Diversified signaling</li> <li>-Supporting ancillary services</li> <li>-Supporting integration of renewable energy sources</li> </ul>	We discussed the benefits of ADR and the challenges related to <b>security, scalability and privacy</b> that hinder the wide acceptance of ADR system in the residential sector.
Information Mediator for Demand Response in Electrical Grids and Building <i>(수요반응을 확장하기 위해 어그리게이터에 페이로드 변환을 하는 논문(상호운용성))</i>	<p>상호 작용을 위한 시스템에 독립적 인 정보 모델 건물과의 정보 교환을 줄임</p> <p>상호 작용 우리는 온톨로지와 DR 상호 작용을 표현하고 가능성을 열어주는 표준 다른 정보 모델들을 연결하는 것 메시지를 해석하고 전달하는 <b>정보 중재자 시스템</b>을 제안</p> <p><b>OpenADR2.0b HTTP/XML → HTTP/Turtle</b></p>	<p><b>정보 중재자 아키텍처를 구현</b></p> <p>DR에 대한 표준 기반 시스템 독립 모델을 기반 DR 가격 결정 이벤트는 조정자와 중개자에게 전달</p> <p><b>OpenADR2.0b HTTP/XML → HTTP/Turtle</b></p>
VOLTTRON: An open-source software platform of the future <i>(VOLTTRON에서 OpenFMB를 이용)</i>	<p>VOLTTRON is a flexible, scalable, distributed agent based environment that seamlessly integrates data, devices, and systems</p> <p><b>Integration with other Platforms - OpenFMB</b></p> <p>Volttron 개발은 지금까지 건물 내부에 집중되어 있었지만, OpenFMB는 그리드에 중점을 두고 상호 작용 하기 위한 표준 인터페이스를 제공 (마이크로아이드 PV, ESS 관련 실험) Ex) Intelligent Load Control</p>	<p>유틸리티를 기반으로 한 응용 프로그램과 장치간에 <b>OpenFMB</b>을 통해 표준 기반 통신을 제공하여 그리드에 통합된 플랫폼을 제공</p> <p>Volttron 플랫폼은 장치에 대한 액세스를 제공함으로써 Volttron 개발은 지금까지 건물 내부에 집중되어 있었지만, openfmb는 그리드에 중점을 두고 상호 작용을 하기 위한 표준 인터페이스를 제공</p>
Advance Metering Infrastructure and DLMS/COSEM Standards for Smart Grid <i>(AMI, DLMS/COSEM 관련한 논문)</i>	<p><b>DLMS / COSEM 표준 개요 제공</b></p> <p>제안 된 커뮤니케이션 프로파일에 대한 자세한 논의 다양한 통신 기술을 다양한 용도로 사용</p> <p><b>AMI의 인터페이스가 제안과 특성 및 프로토콜 비교</b></p> <p>A. IEEE 802.11 Standards B. IEC 61334 C. ZigBee D. IEC 61850 E. Cellular Technologies</p>	<p><b>DLMS / COSEM 표준은 AMI 및 스마트 요구 사항에 가장 근접한 현재 솔루션</b></p> <p>그리드가 연구된다. 계량 장비 모델링 방법 및 OBIS 명명이 연구되었습니다. 제안 된 통신 프로파일 DLMS에서 제안 된 클라이언트 - 서버 아키텍처를 기반으로 표준이 검토됩니다. 다양한 통신 기술로 AMI의 다른 인터페이스에 사용할 수 있음</p>

# Smart Home Energy Framework :

## 1.4 참고문헌 - 수요반응 및 스마트 그리드 기술에 관한 연구 (3/3)

제목	내용 및 범위	결론
An Algorithm for Intelligent Home Energy Management and Demand Response Analysis <b>(DR, 에너지 최적화 논문)</b>	연구 결과는 <b>주거용 고객</b> 에게 가능한 부하 감소 수준에 대한 통찰력을 제공 할 것으로 기대되며 이는 <b>자동화 된 DR</b> 을 갖춘 주택 시장에서 DR 잠재력으로 해석 <b>가정용 전력 집약 형 기기 관리를 위한 HEM 알고리즘 개발</b>	제안 된 HEM 알고리즘은 총 가정 소비를 지정된 수요 한도 이하로 유지하기 위해 사전에 그리고 효과적으로 설비 운영을 제어하고 관리 할 수 있음
Home Energy Management System based on Power Line Communication <b>(HEMS을 통한 에너지관리, 수요자원을 관리)</b>	<b>Energy Management Requirement Functions :</b> 1)remote monitoring, 2)controlling 3)planning 4)repairing of operations 5)provide information on the status of installed devices and the network <b>Proposed HEMS has the following four functions :</b> auto-configuration: device discovery easy monitoring remote controlling smart planning	HEMS는 스마트 계량기를 모니터링하고 자동 구성, 원격 모니터링, 에너지 관리, 가능성 있는 제어를 제공하는 인터넷에서 원격으로 에너지 관련 기기를 제어 할 계획을 세웁니다. 최고 가 시간에 장치 제어를 전환하고 원활한 전력 수요 작업 부하를 발생시킵니다. HEMS는 자원 활용, 에너지 절약 및 비용 절감에 대한 명확한 이점을 사용자에게 제공 할 수 있습니다.
Application and Development of Zigbee Technology for Smart Grid Environment(SEP2.0) <b>(SEP2.0 관련 논문, 설명 참조할 예정)</b>	<b>Smart Grid and Zigbee SEP2.0</b> -Zigbee Smart Energy Profile 2.0 Features -Zigbee SEP 2.0 Applications -Electromagnetic compatibility (EMC)	스마트 그리드가 대중화되고 있으며 미래의 응용 분야는 단거리 <b>무선 EMC와 Zigbee SEP2.0 표준 기능과 결합 된 경우</b> <b>여러 네트워크 기술을 지원</b> <b>스마트 홈 기술을 위한 커뮤니케이션 기능.</b>
에너지 IoT 기반 집합 건물 환경에서 효율적인 수요반응을 위한 경량 에너지관리에이전트 프로토콜 설계 및 구현 논문 <b>(경량 에너지관리에이전트 프로토콜 설계 및 구현)</b> 박현일 선배 논문	경량 에너지관리에이전트 프로토콜 개발  기존 시스템 Scalability한계 => 서버 에너지관리에이전트의 Aggregator구현으로 데이터 트래픽 감소, 병복현상 감소  집합 건물 환경의 에너지관리에이전트의 모니터링 보안 문제점 => Implicit, Explicit 모니터링으로 문제 해결  기존 IoT 디바이스들에 Control지시 => 집단 건물 내부에 에너지관리에이전트가 있어 서버 에너지관리에이전트로 수요 반응 및 스케줄링 가능	집합건물 내부에도 대표적인 Aggregator(Server EMA)로 주거 공간인 EMA(Client EMA)들에게 수요 관리 (1)EMA를 통해 집합 건물 내에서도 지능화된 에너지관리인 수요반응을 함 (2)수요관리 고객에 대한 실시간 수요반응을 할 수 있음- Push 실현을 통해 증명 (3)집단 건물 내부에 에너지관리에이전트가 있어 개인 사용자가 상위 에너지관리에이전트와 통신을 통해 수요반응 및 스케줄링을 할 수 있음 (4)집합 건물 내의 EMA Protocol을 초경량 프로토콜인 CoAP, MQTT를 이용 (5)기존연구의 Explicit모니터링과 EMAP의 Implicit/Explicit 모니터링 비교를 통해 효율성 검증

# Energy Framework

## 2. Profile & Protocol : OpenADR, EMAP, **DLMS/COSEM**

2.1 OpenADR (HTTP/XML, MQTT, COAP)

*VTN ↔ VEN*

2.2 EMAP (MQTT, COAP/JSON)

*Server EMA ↔ Client EMA*

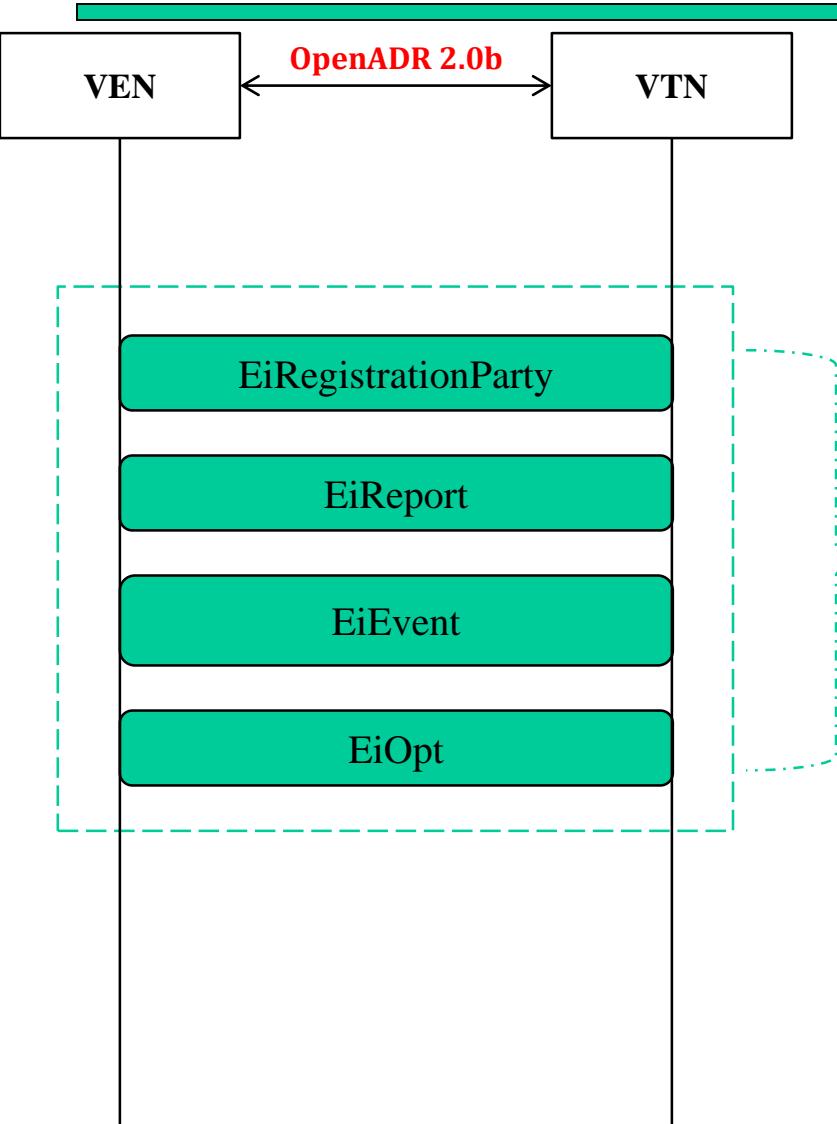
2.3 Device Control

*Client EMA ↔ Device*

2.4 **DLMS/COSEM**

# 2.1 Profile : OpenADR 2.0b

## 2.1 Services



- **EiRegistrationParty**

- VTN-VEN간 서로 연결을 수립하는 과정이다. 서로의 정보를 교환하고, VEN의 ID Name이 미리 등록되어 있을 경우 진행이 가능하다. VTN에서 VEN의 ID값을 발급해주고, 이 값을 이용하여 나머지 서비스들을 진행 할 수 있게 된다. 각 VTN-VEN을 등록하고 디바이스 정보 교환 과정이다.

- **EiReport**

- VEN 모두 자신의 Report 자원을 서로에게 등록할 수 있으며, Report자원 중에서 원하는 것을 선택하여 구독신청 한다. 이 후 Update Report 메시지를 통하여 Report를 지속적으로 보내준다. VEN 등록 완료 후 VTN에서 등록한 Resource 상태정보를 전송한다.

- **EiEvent**

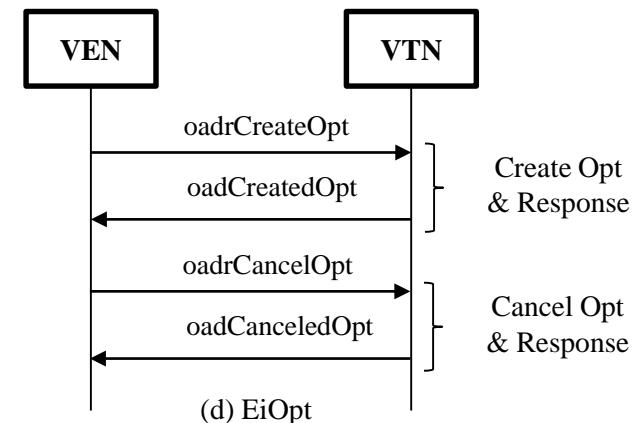
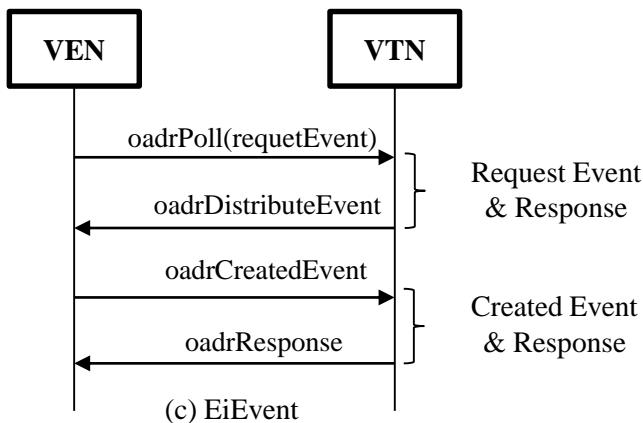
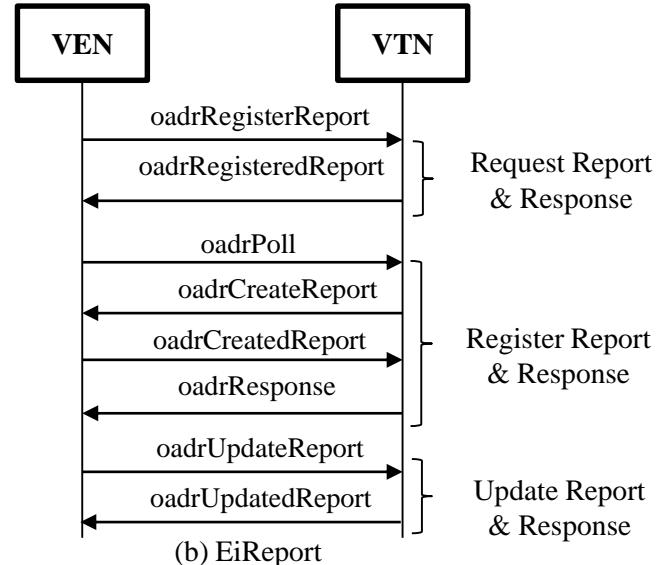
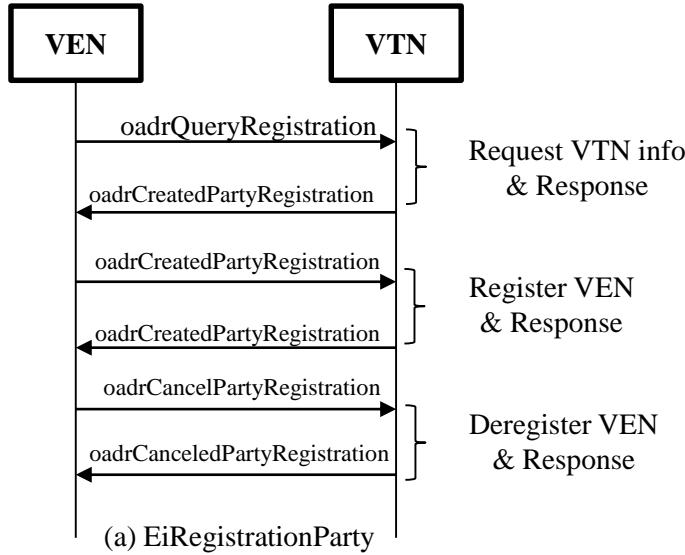
- 수요자원의 사용패턴 변화를 위한 Event 신호를 내려주는 기능을 한다. Event 신호에는 Event시간, 대상, 내용(level, price, power등)에 대한 정보가 포함되어 수요반응을 만들어 낸다.

- **EiOpt**

- VEN이 지정한 시간 대에 Event에 참여 여부를 예약 하는 서비스이다. Event를 수행 할 때는 Opt-in, 불가능 할 때는 Opt-out메시지를 VTN에게 보내어 VEN의 가용상태를 알려준다.

# 2. Profile :OpenADR 2.0b

## 2.1 Services Procedure



## 2. Smart Home Energy Framework :

### 2.1 OpenADR(HTTP/XML, MQTT, CoAP/JSON) Profile Spec (1/2)

---

- OpenADR 모델에서 필요한 서비스 4가지(OpenADR2.0b)
  - EiRegistration, EiEvent, EiReport, EiOpt
  - **EiRegistration**
    - oadrQueryRegistration/oadrCreatedPartyRegistration : VEN이 등록전에 등록에 필요한 정보를 VTN으로부터 얻는다. 응답메시지에는 VTN에 의해 지원되는 모든 프로파일과 전송 프로토콜 정보가 포함된다.
    - oadrCreatePartyRegistration/oadrCreatedPartyRegistration : VEN이 VTN의 정보 중에 등록할 내용을 입력하여 등록을 요청하고 VTN이 받아들인다.
    - oadrCancelPartyRegistration/oadrCanceledPartyRegistration : Source 측에서 등록을 취소하고 Target 측에서 받아들인다.
    - oadrRequestRegistration/oadrResponse : VTN의 등록 정보가 변경되는 경우에 해당 작업을 통해 VEN에게 재등록을 요구한다.
  - **EiEvent**
    - oadrRequestEvent : VEN이 DR이벤트를 요청
    - oadrDistributeEvent : VTN이 DR 이벤트를 보냄
    - oadrCreatedEvent : VEN이 이벤트에 참여하거나 참여하지 않음
    - oadrResponse : VTN이 응답을 보냄

## 2. Smart Home Energy Framework :

### 2.1 OpenADR(HTTP/XML, MQTT, CoAP/JSON) Profile Spec (2/2)

---

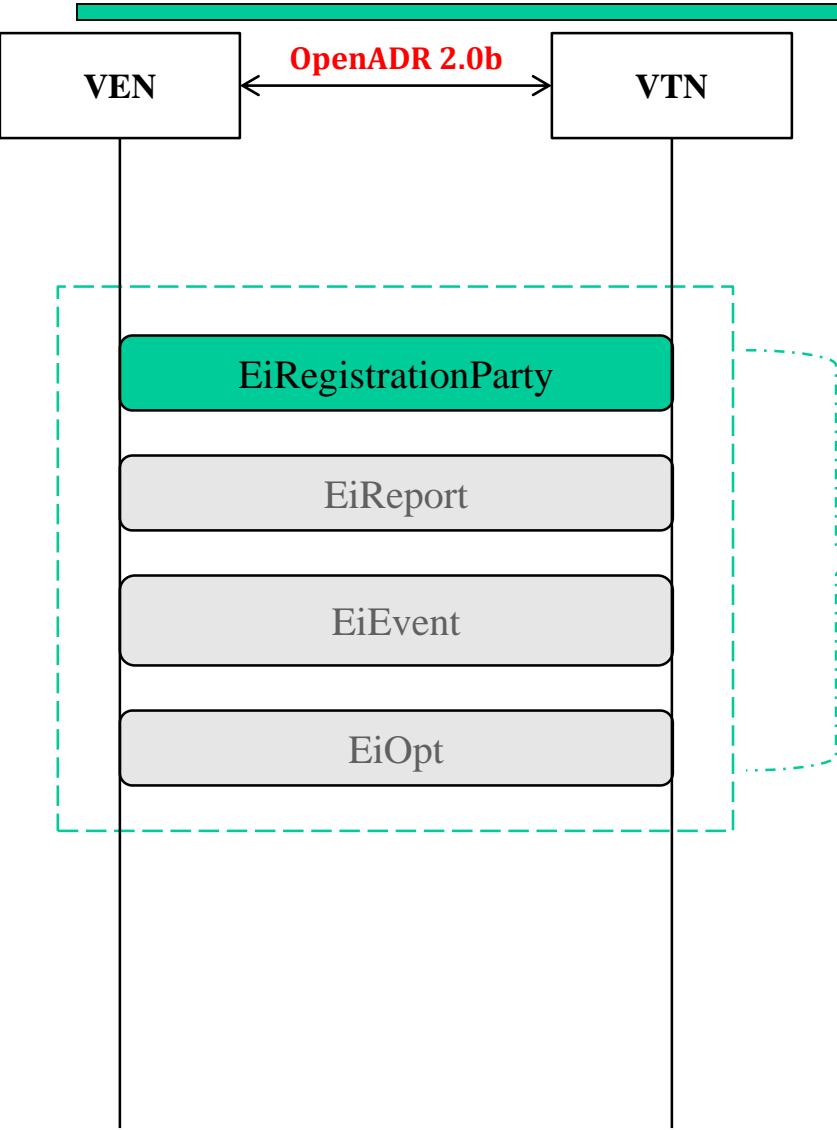
- OpenADR 모델에서 필요한 서비스 4가지(OpenADR2.0b)
  - EiReport
    - oadrRegisterReport/oadrRegisteredReport : 가능한 리포트를 알려주고 받아들임
    - oadrCreateReport/oadrCreatedReport : 구체적인 리포트를 요청하고 받아들임
    - oadrUpdateReport/oadrUpdatedReport : 요청한 리포트를 전달하고 받아들임
    - oadrCancelReport/oadrCanceledReport : 요청한 리포트를 취소하고 받아들임
  - EiOpt
    - oadrCreateOpt/oadrCreatedOpt : VEN이 opt 스케줄을 보내고 VTN이 응답
    - oadrCancelOpt/oadrCanceledOpt : VEN이 opt 스케줄을 취소하고 VTN이 응답

**OpenADR 2.0b**

**(1)EiRegistrationParty**

# 2. Profile : OpenADR 2.0b

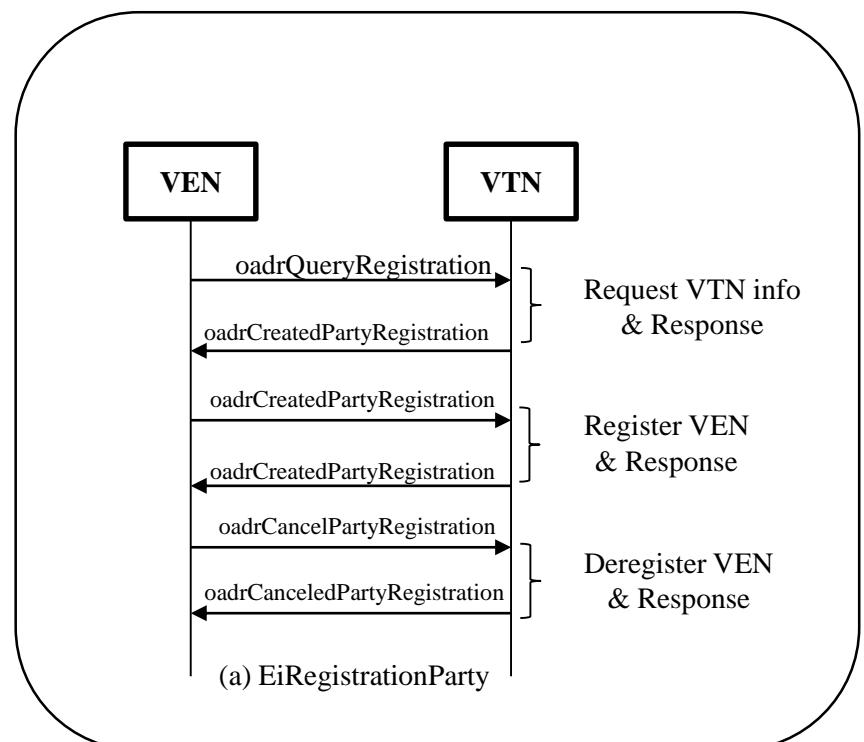
## 2.1 Services : EiRegistrationParty



- 

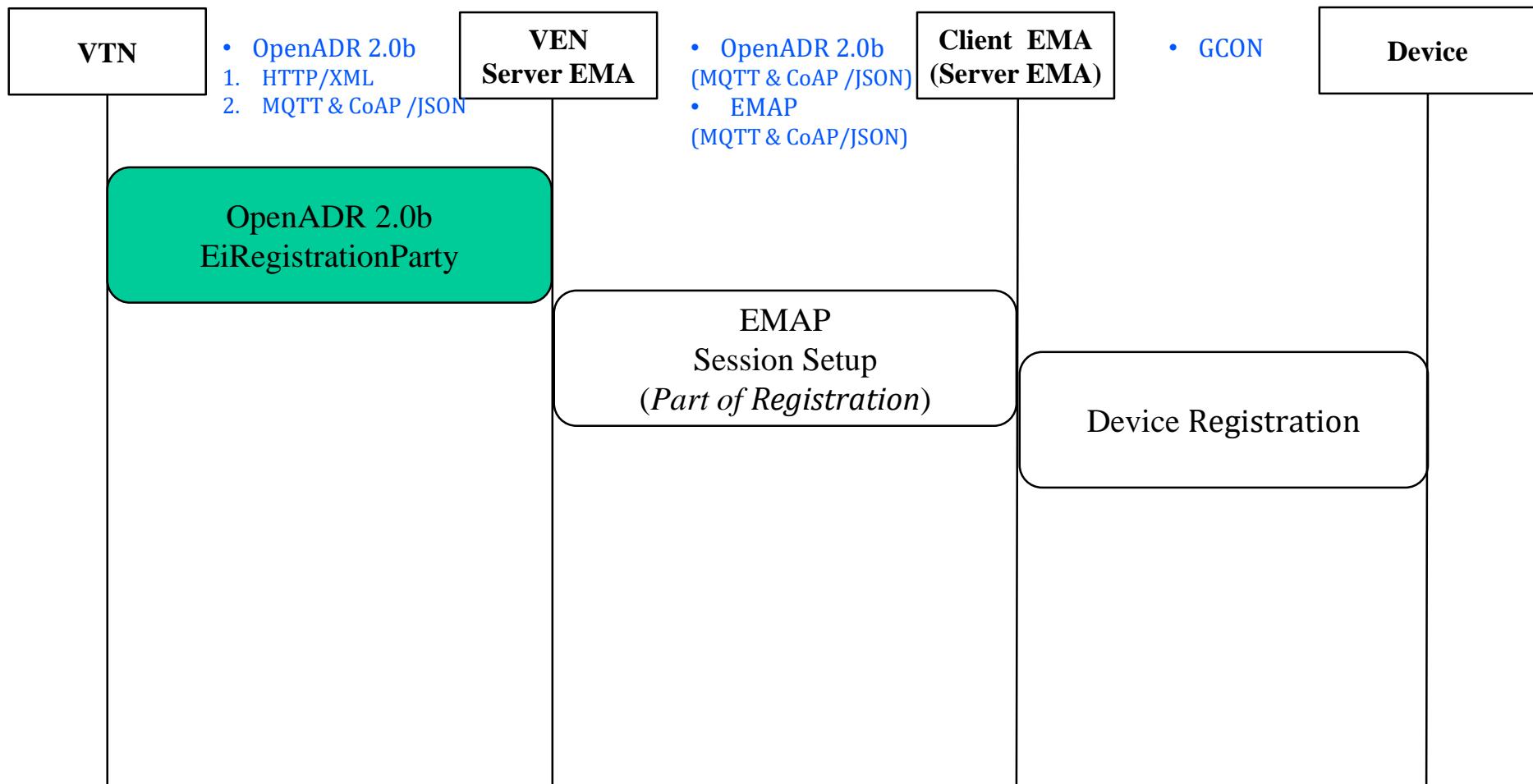
### EiRegistrationParty

- VTN-VEN간 서로 연결을 수립하는 과정이다. 서로의 정보를 교환하고, VEN의 ID Name이 미리 등록되어 있을 경우 진행이 가능하다. VTN에서 VEN의 ID값을 발급해주고, 이 값을 이용하여 나머지 서비스들을 진행 할 수 있게 된다. 각 VTN-VEN을 등록하고 디바이스 정보 교환 과정이다.



# 2. Profile : OpenADR 2.0b

## 2.1 Services : EiRegistrationParty



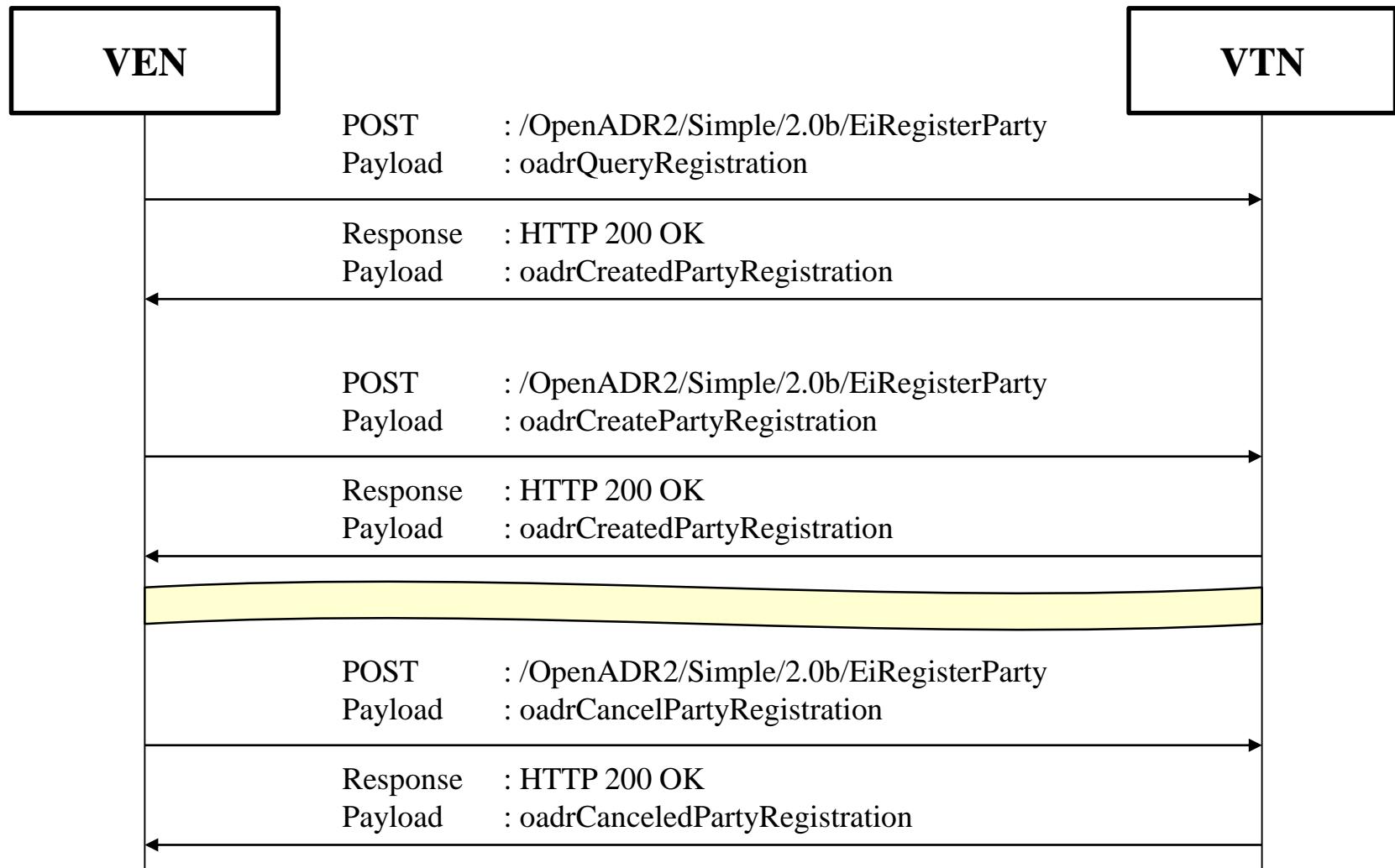
# **OpenADR 2.0b**

## **(1) EiRegistrationParty**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

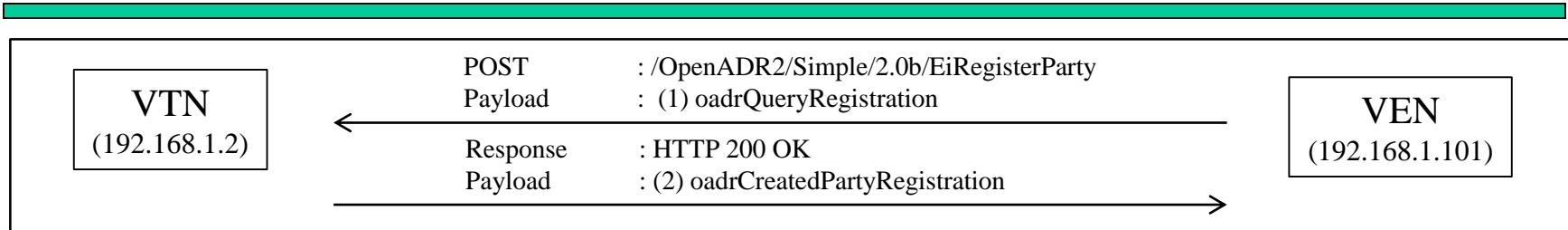
# 2. Profile : OpenADR 2.0b

## 2.1 Services : EiRegistrationParty ([HTTP / XML](#))



# 2. Profile : OpenADR 2.0b

## 2.1 Services : EiRegistrationParty ([HTTP / XML](#))



(1) oadrQueryRegistration

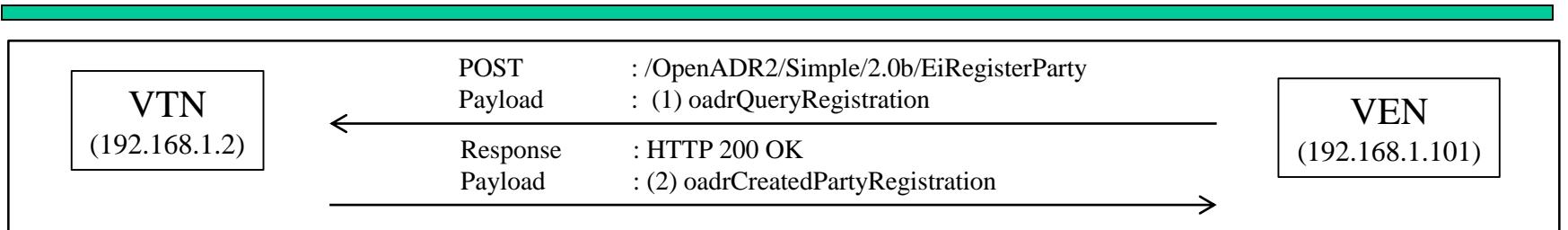
Key name	Comments
requestID	request identifier
service	message type
venID	requested VEN ID

(2) oadrCreatedPartyRegistration

Key name	Comments
venID	requested VEN ID
vtnID	responded VTN ID
responseCode	response code
responseDescription	description of response code
requestID	request identifier
duration	requested polling frequency
registrationID	registration identifier
oadrProfile	oadrProfileName
	oadrTransports
service	oadrTransportName
	message type

# 2. Profile : OpenADR 2.0b

## 2.1 Services : EiRegistrationParty (HTTP / XML)



(1) oadrQueryRegistration      192.168.1.101 192.168.1.2 POST /OpenADR2/Simple/2.0b/EiRegisterParty HTTP/1.1  
(2) oadrCreatedPartyRegistration      192.168.1.2 192.168.1.101 HTTP/1.1 200 OK

**oadrQueryRegistration XML{**  
    “requestID”: String  
}

```
> <?xml
  <p1:oadrPayload
    xmlns:p1="http://openadr.org/oadr-2.0b/2012/07">
    <p1:oadrSignedObject>
      <p1:oadrQueryRegistration
        xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110"
        p3:schemaVersion="2.0b"
        xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
        <p2:requestID>
          7714720BBB
        </p2:requestID>
      </p1:oadrQueryRegistration>
    </p1:oadrSignedObject>
  </p1:oadrPayload>
```

**oadrCreatedPartyRegistration XML{**  
    “venID”: String,  
    “vtnID”: String,  
    “requestID”: String,  
    “duration” : Integer,  
    “responseCode” : Integer,  
    “responseDescription”: String,  
    “registrationID” : String,  
    “**oadrProfile**” : Array

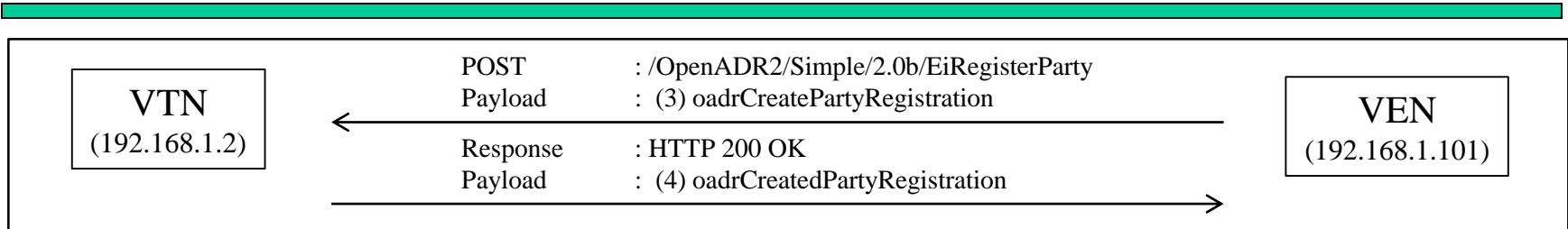
**oadrProfile** Array{  
    “**oadrTransports**”: Array,  
    “oadrProfileName”: String  
}

**oadrTransports** Array{  
    “oadrTransportName”: String,  
}

```
<ns1:oadrCreatedPartyRegistration
  ns2:schemaVersion="2.0b">
  <ns2:eiResponse>
    <ns2:responseCode>
      <ns2:responseDescription>
        <ns3:requestID>
        </ns3:requestID>
      </ns2:responseDescription>
    </ns2:eiResponse>
    <ns2:registrationID>
    <ns2:venID>
    <ns2:vtnID>
    <ns1:oadrProfiles>
      <ns1:oadrRequestedOadrPollFreq>
      </ns1:oadrRequestedOadrPollFreq>
      </ns1:oadrCreatedPartyRegistration>
    </ns1:oadrSignedObject>
```

## 2. Profile : OpenADR 2.0b

### 2.1 Services : EiRegistrationParty (HTTP / XML)

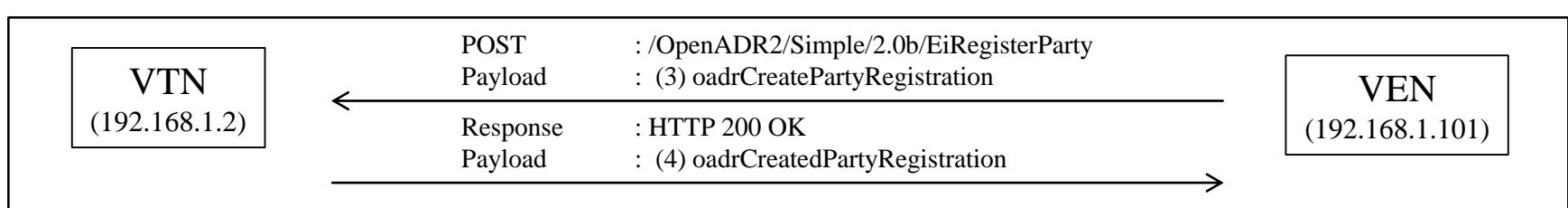


(3) oadrCreateRegistration

Key name	Comments
requestID	request identifier
oadrProfileName	profile name used by VEN
oadrTransportName	transport name used by VEN
oadrReportOnly	VEN type (report only or full functional)
oadrXmlSignature	xml 사용여부 true/false
oadrVenName	VEN name
oadrHttpPullMode	communication mode used by VEN (pull or push)
venID	requested VEN ID
service	message type

## 2. Profile : OpenADR 2.0b

### 2.1 Services : EiRegistrationParty (HTTP / XML)

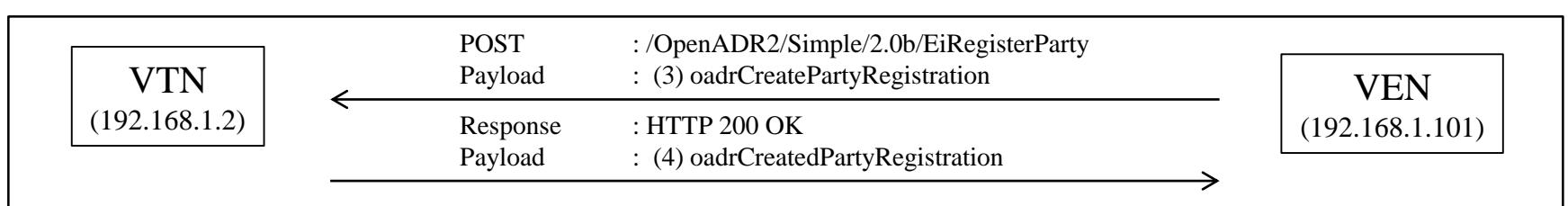


(4) oadrCreatedPartyRegistration

Key name	Comments
venID	requested VEN ID
vtnID	responded VTN ID
responseCode	response code
responseDescription	description of response code
requestID	request identifier
duration	requested polling frequency
registrationID	registration identifier
oadrProfile	oadrProfileName
	oadrTransports oadrTransportName
service	message type

## 2. Profile : OpenADR 2.0b

### 2.1 Services : EiRegistrationParty (HTTP / XML)



(3) oadrCreateRegistration      192.168.1.101 192.168.1.2 POST /OpenADR2/Simple/2.0b/EiRegisterParty HTTP/1.1  
(4) oadrCreatedPartyRegistration      192.168.1.2 192.168.1.101 HTTP/1.1 200 OK

#### oadrCreateRegistration XML{

```
"requestID": String,  
"oadrProfileName": String,  
"oadrTransportName": String,  
"oadrReportOnly": Boolean,  
"oadrXmlSignature": String,  
"oadrVenName": String,  
"oadrHttpPullModel": Boolean,  
}
```

```
> <p2:requestID>  
> <p1:oadrProfileName>  
> <p1:oadrTransportName>  
> <p1:oadrReportOnly>  
> <p1:oadrXmlSignature>  
> <p1:oadrVenName>  
> <p1:oadrHttpPullModel>  
</p1:oadrCreatePartyRegistration>
```

#### oadrCreatedPartyRegistration XML{

```
"venID": String,  
"vtnID": String,  
"requestID": String,  
"duration" : String,  
"responseCode" : Integer,  
"responseDescription": String,  
"registrationID": String,  
"oadrProfile" : Array  
}
```

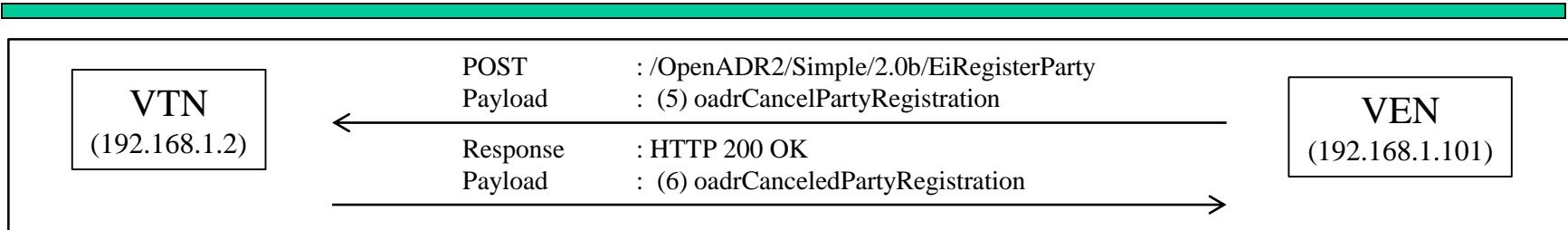
```
oadrProfile Array{  
    "oadrTransports": Array,  
    "oadrProfileName": String  
}
```

```
oadrTransports Array{  
    "oadrTransportName": String,  
}
```

```
<ns1:oadrCreatedPartyRegistration  
    ns2:schemaVersion="2.0b">  
    <ns2:eiResponse>  
        <ns2:responseCode>  
        <ns2:responseDescription>  
        <ns3:requestID>  
            </ns2:eiResponse>  
        <ns2:registrationID>  
        <ns2:venID>  
        <ns2:vtnID>  
        <ns1:oadrProfiles>  
        <ns1:oadrRequestedOadrPollFreq>  
            </ns1:oadrCreatedPartyRegistration>  
</ns1:oadrSignedObject>
```

## 2. Profile : OpenADR 2.0b

### 2.1 Services : EiRegistrationParty (HTTP / XML)



(5) `oadrCancelRegistration`

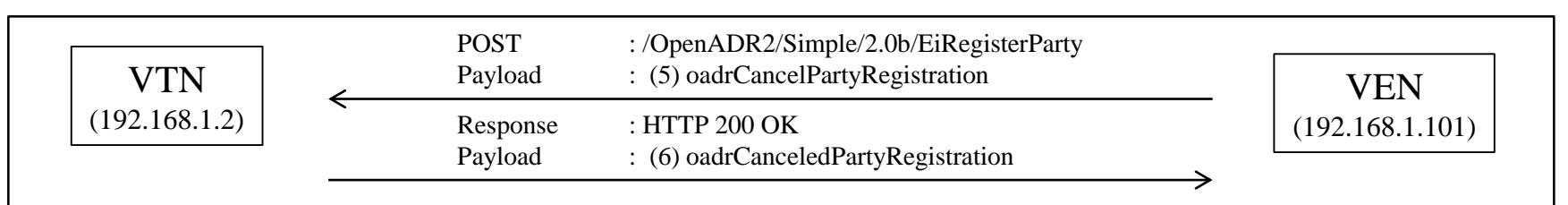
Key name	Comments
venID	requested VEN ID
requestID	request identifier
registrationID	registration identifier
service	message type

(6) `oadrCanceledPartyRegistration`

Key name	Comments
venID	requested VEN ID
requestID	request identifier
responseCode	response code
responseDescription	description of response code
registrationID	registration identifier
service	message type

## 2. Profile : OpenADR 2.0b

### 2.1 Services : EiRegistrationParty (HTTP / XML)



(5) oadrCancelRegistration      192.168.1.101 192.168.1.2 POST /OpenADR2/Simple/2.0b/EiRegisterParty HTTP/1.1  
(6) oadrCanceledPartyRegistration 192.168.1.2 192.168.1.101 HTTP/1.1 200 OK

#### oadrCancelRegistration XML{

```
"requestID": String,  
"registrationID": String,  
"venID": String  
}  
  
✓ <oadrSignedObject>  
  ✓ <oadrCancelPartyRegistration  
    d3p1:schemaVersion="2.0b"  
    xmlns:d3p1="http://docs.oasis-open.or  
  > <requestID  
  > <d3p1:registrationID>  
  > <d3p1:venID>  
    </oadrCancelPartyRegistration>  
  </oadrSignedObject>  
</oadrPayload>
```

#### oadrCanceledPartyRegistration XML{

```
"venID": String,  
"requestID": String,  
"responseCode": Integer,  
"responseDescription": String,  
"registrationID": String,  
}  
  
✓ <ns3:oadrSignedObject>  
  ✓ <ns3:oadrCanceledPartyRegistration  
    ns5:schemaVersion="2.0b">  
    ✓ <ns5:eiResponse>  
      > <ns5:responseCode>  
      > <ns5:responseDescription>  
      > <ns4:requestID>  
        </ns5:eiResponse>  
      > <ns5:registrationID>  
      > <ns5:venID>  
        </ns3:oadrCanceledPartyRegistration>  
    </ns3:oadrSignedObject>  
</ns3:oadrPayload>
```

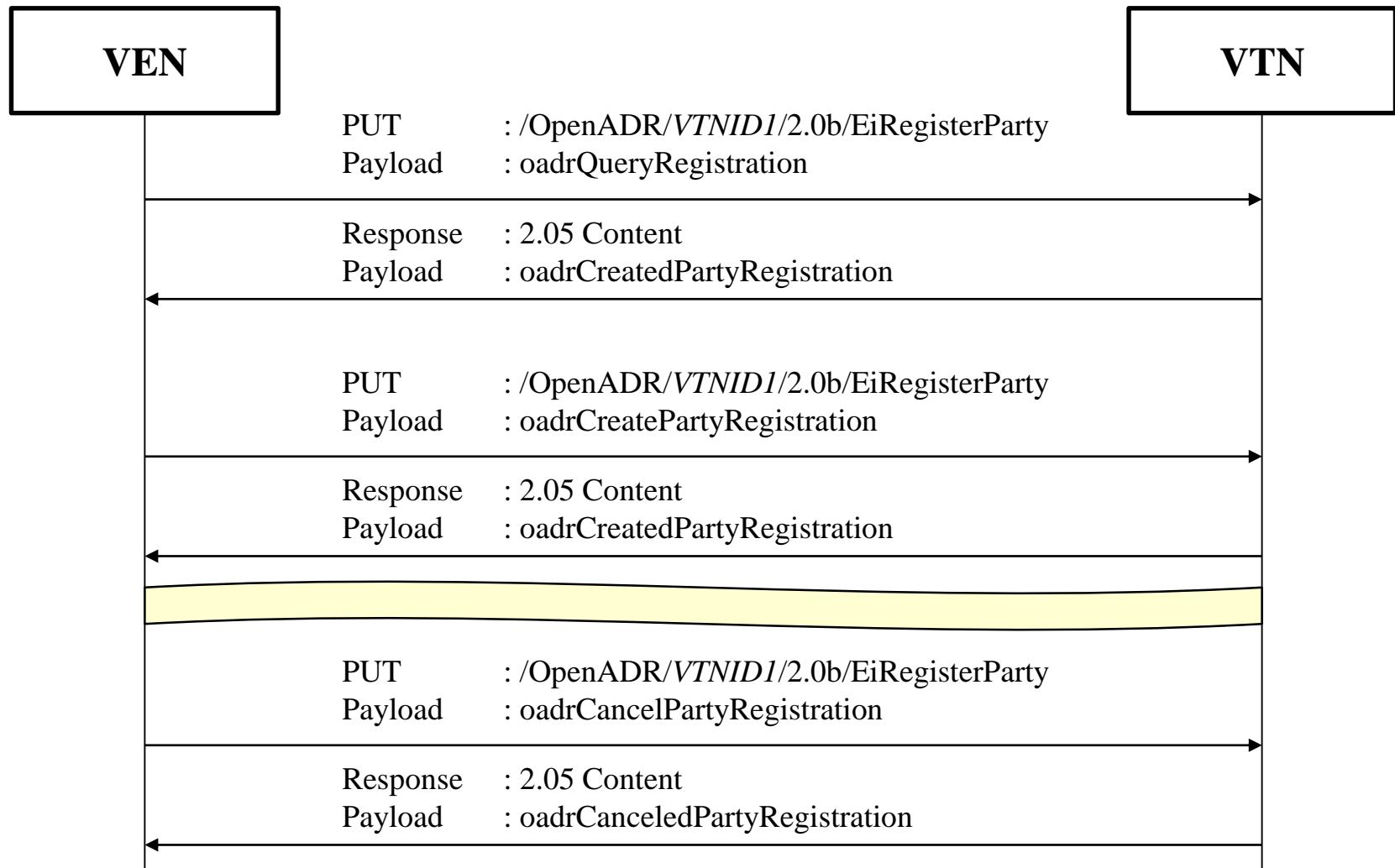
# **OpenADR 2.0b**

## **(1) EiRegistrationParty**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

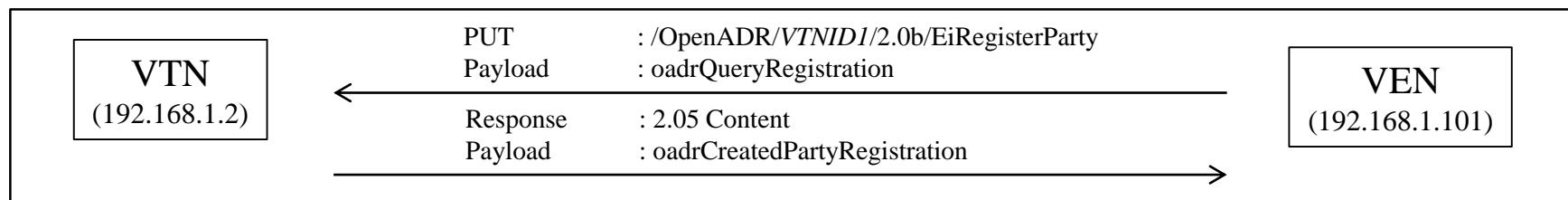
# 2. Profile : OpenADR 2.0b

## 2.1 Services : EiRegistrationParty (CoAP / JSON)



## 2. Profile : OpenADR 2.0b

### 2.1 Services : EiRegistrationParty (CoAP / JSON)



(1) QueryRegistration	CoAP	192.168.1.101	192.168.1.127	CON, MID:17505, PUT, /QueryRegistration
(2) CreatedPartyRegistration	CoAP	192.168.1.127	192.168.1.101	ACK, MID:17505, 2.05 Content (text/plain)

#### oadrQueryRegistration JSON{

```

  "requestID": String,
  "service" : String,
  "venID" : String
}
```

```

QueryRegistration{ "Service": "QueryRegistration", "GW": "gw\1",
"VENID": "VEN_MIR1", "Request ID": 1, "Version ": 2 }
```

#### oadrCreatedPartyRegistration JSON{

```

  "venID": String,
  "vtnID": String,
  "requestID": String,
  "duration" : Integer,
  "responseCode" : Integer,
  "responseDescription": String,
  "registrationID" : String,
  "oadrProfile" : Array,
  "service" : String
}
```

```

  oadrProfile Array{
    oadrTransports": Array,
    oadrProfileName": String
  }
```

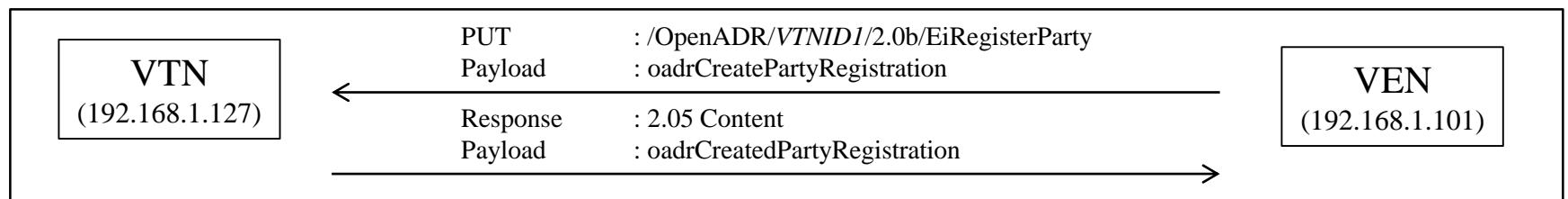
```

  oadrTransports Array{
    oadrTransportName": String,
  }
```

```
DR {"TransportName": "MIR_VTN", "RequestID": 1, "VENID": "VEN_MIR1", "RegistrationID": 1, "VTNID": "MIR_VTN", "Service": "CreatedPartyRegistration", "Duration": 2000}
```

## 2. Profile : OpenADR 2.0b

### 2.1 Services : EiRegistrationParty (CoAP / JSON)



- (3) CreateRegistration  
 (4) CreatedPartyRegistration

CoAP	192.168.1.101	192.168.1.127	CON, MID:17760, PUT, /CreatePartyRegistration
CoAP	192.168.1.127	192.168.1.101	ACK, MID:17760, 2.05 Content (text/plain)

#### oadrCreateRegistration JSON{

```

  "requestID": String,
  "oadrProfileName": String,
  "oadrTransportName": String,
  "oadrReportOnly": Boolean,
  "oadrXmlSignature": String,
  "oadrVenName": String,
  "oadrHttpPullMode": Boolean,
  "service" : String,
  "venID" : String
}
  
```

```

  CreatePartyRegistration{ "Service": "CreatePartyRegistrationRequest",
  "GW": "gw \v1", "RequestID": 1, "Version": 2, "TransportName": "CoAP", "ReportOnly": 0,
  "oadrVenName": "VEN_MIR1", "LastPol1PushGet": 3, "oadrProfileName": 2, "oadrXmlSignature": 0,
  "oadrTransportAddress": "192.168.1. 2" }
  
```

#### oadrCreatedPartyRegistration JSON{

```

  "venID": String,
  "vtID": String,
  "requestID": String,
  "duration" : String,
  "responseCode" : Integer,
  "responseDescription": String,
  "registrationID": String,
  "oadrProfile" : Array,
  "service" : String
}
  
```

```

  oadrTransports Array{
    "oadrTransportName": String,
  }
  
```

```

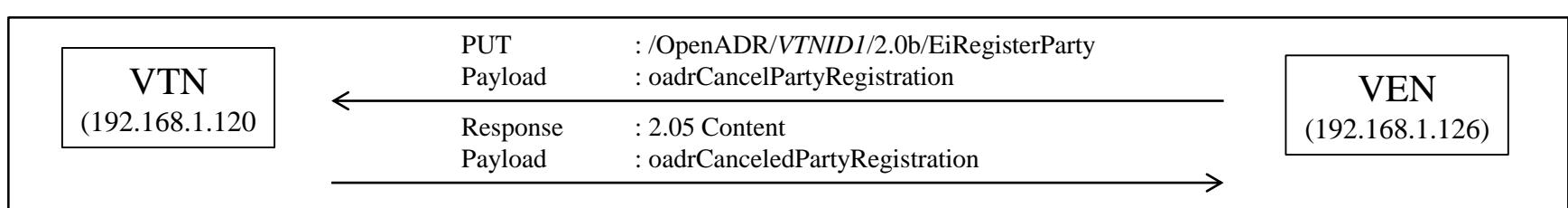
  oadrProfile Array{
    "oadrTransports": Array,
    "oadrProfileName": String
  }
  
```

```

  DR {"Response":200,"TransportName":"MIR_VTN","RequestID":1,"Version":2,"VENID":"VEN_MIR1","VTNID":"MIR_VTN","Service":"CreatedPartyRegistration2","Duration":2000}
  
```

## 2. Profile : OpenADR 2.0b

### 2.1 Services : EiRegistrationParty (CoAP / JSON)



- (5) CancelRegistration
- (6) CanceledPartyRegistration

```

CancelRegistration JSON{
  "requestID": String,
  "registrationID": String,
  "venID": String,
  "service" : String
}
  
```

```

CanceledPartyRegistration JSON{
  "venID": String,
  "requestID": String,
  "responseCode" : Integer,
  "responseDescription": String,
  "registrationID": String,
  "service" : String
}
  
```

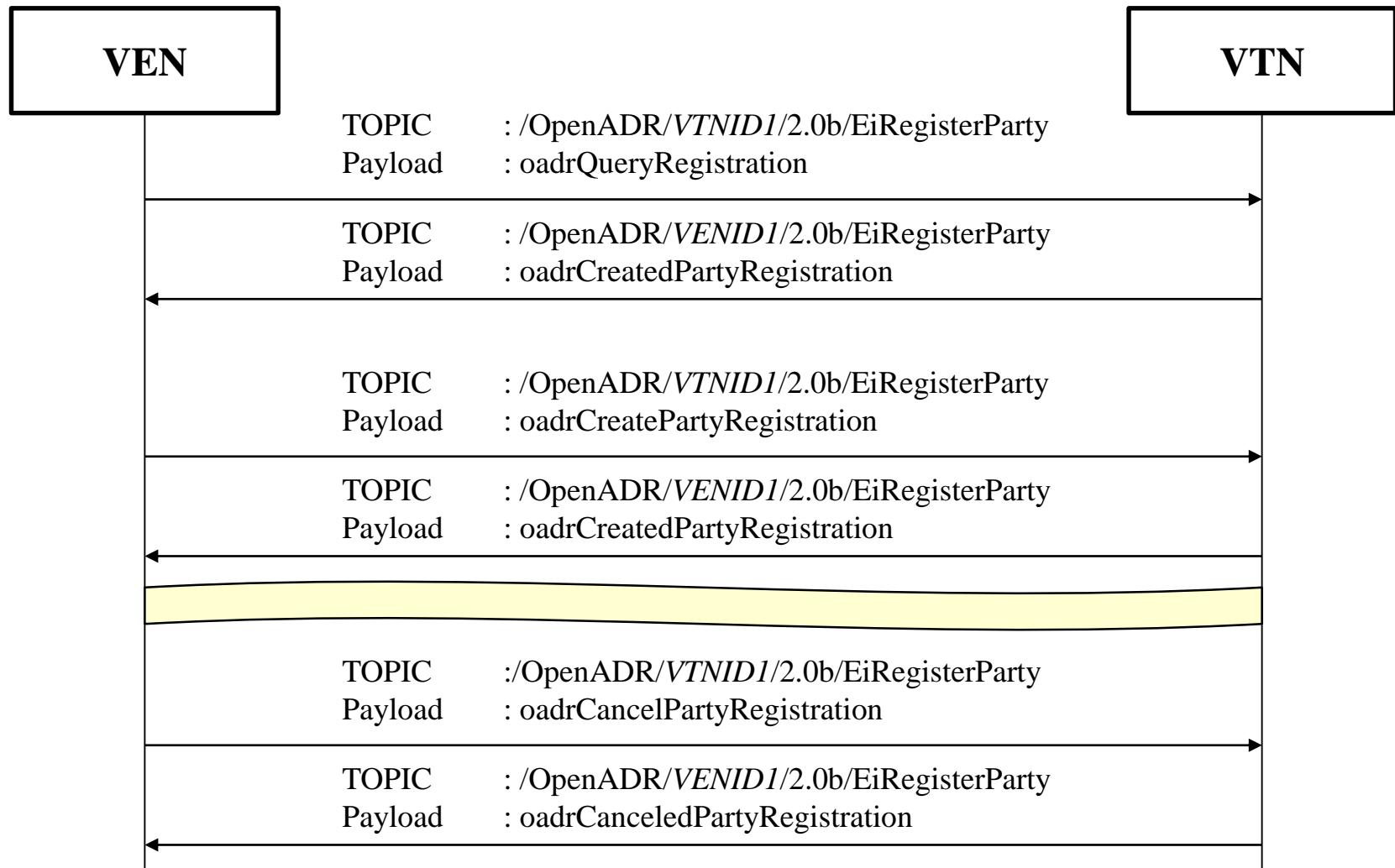
# **OpenADR 2.0b**

## **(1) EiRegistrationParty**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

# 2. Profile : OpenADR 2.0b

## 2.1 Services : EiRegistrationParty (MQTT / JSON)

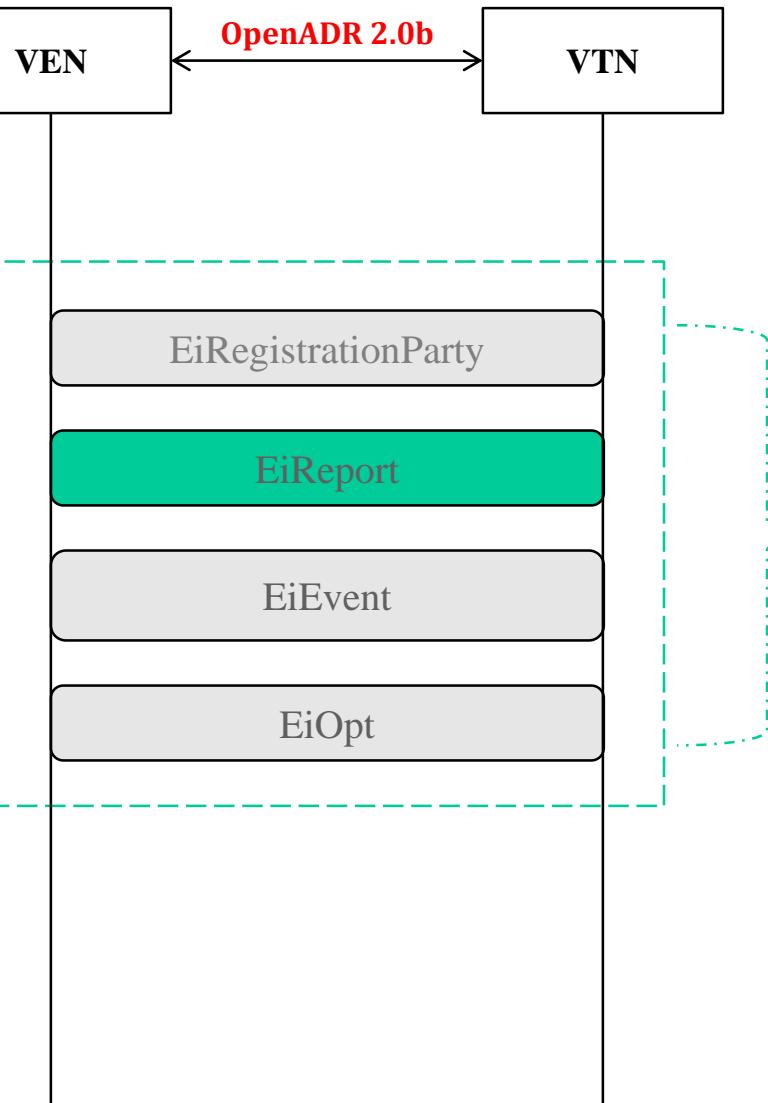


**OpenADR 2.0b**

**(2) EiReport**

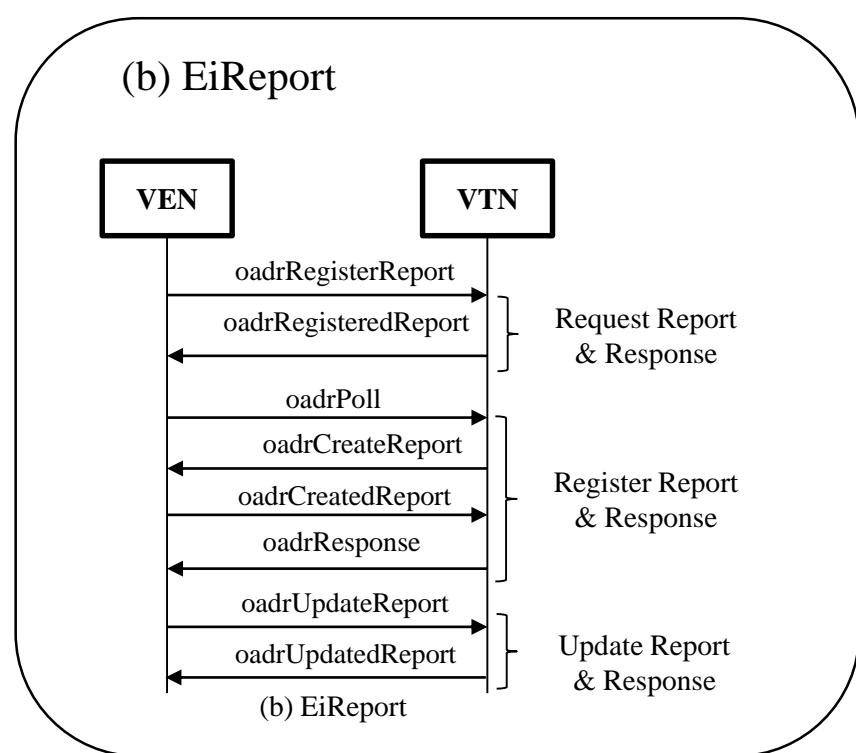
# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport



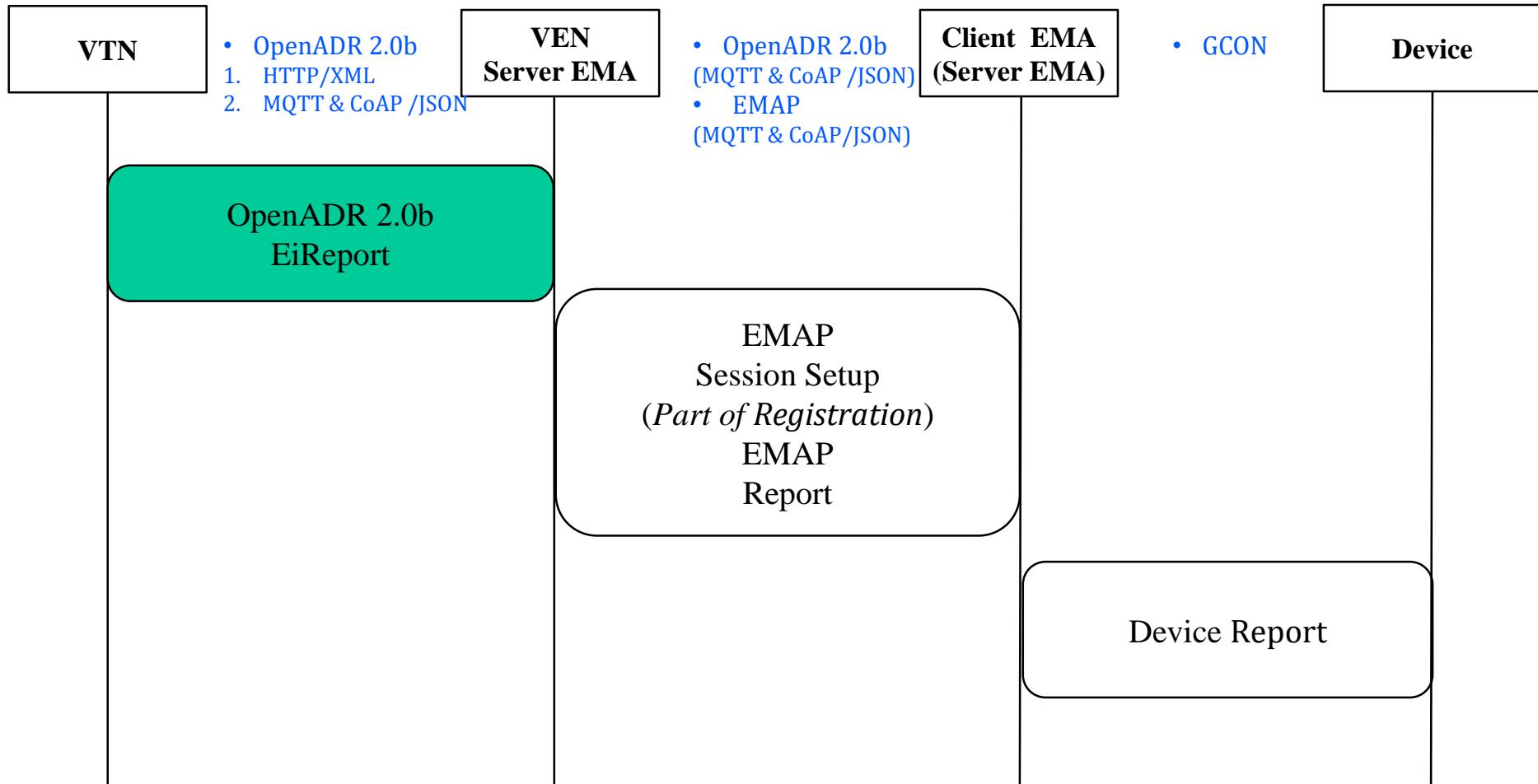
- EiReport

- VEN 모두 자신의 Report 자원을 서로에게 등록할 수 있으며, Report자원 중에서 원하는 것을 선택하여 구독신청 한다. 이후 Update Report 메시지를 통하여 Report를 지속적으로 보내준다. VEN 등록 완료 후 VTN에서 등록한 Resource 상태정보를 전송한다.



# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport



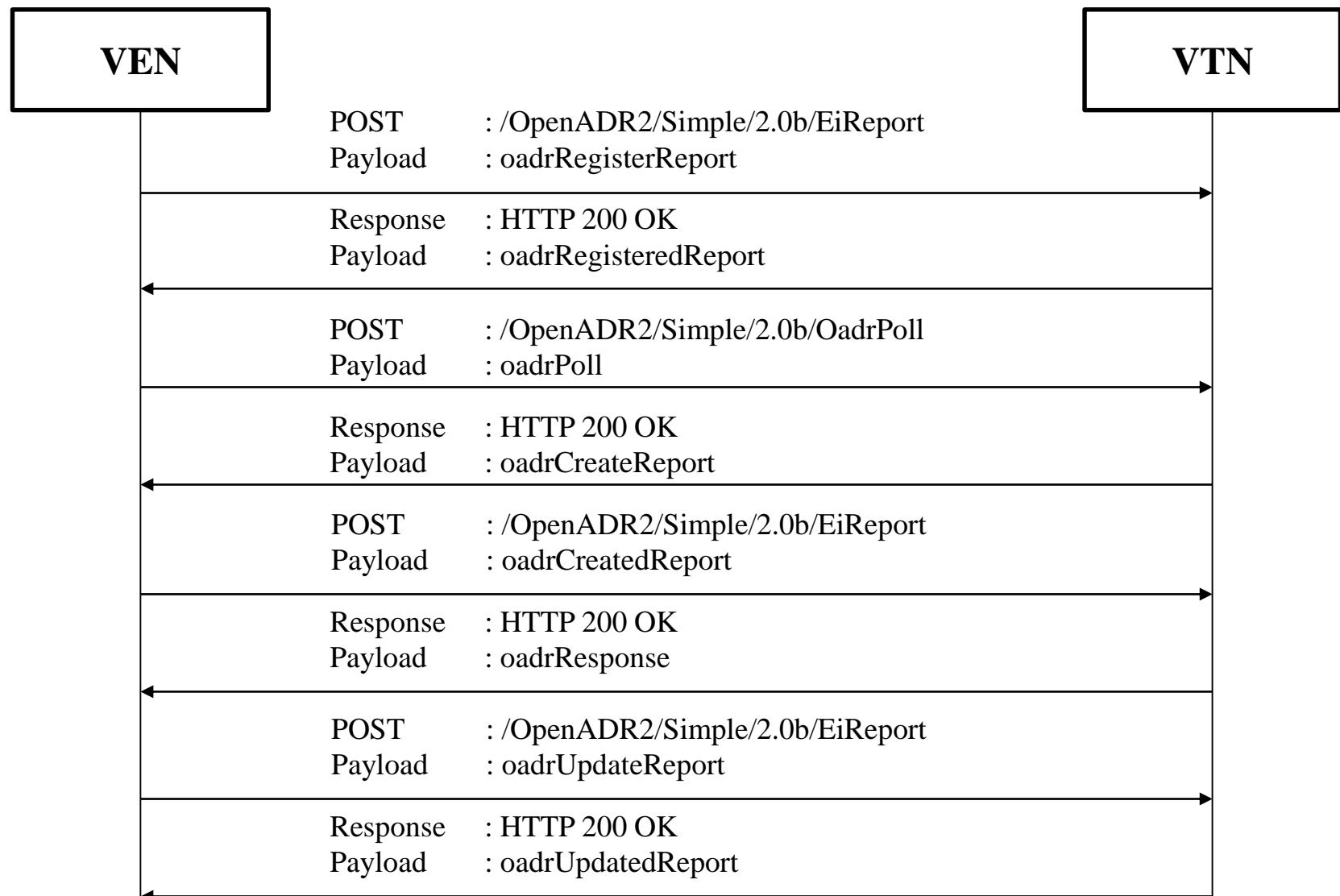
# **OpenADR 2.0b**

## **(2) Report**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

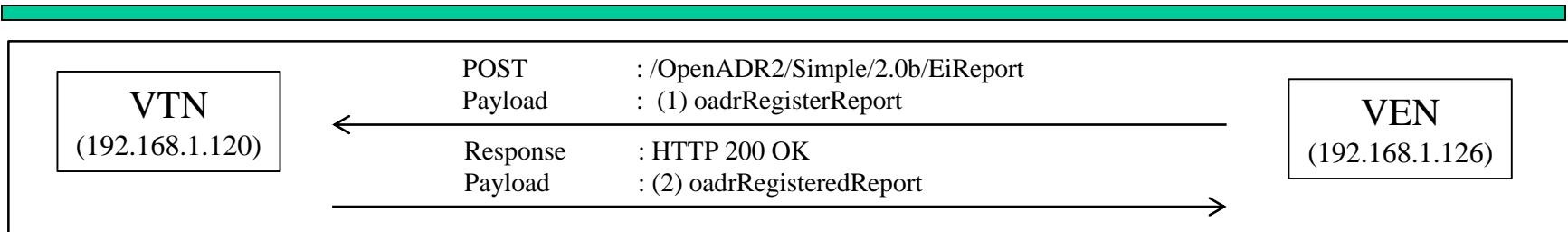
# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport ([HTTP / XML](#))



# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (HTTP / XML)

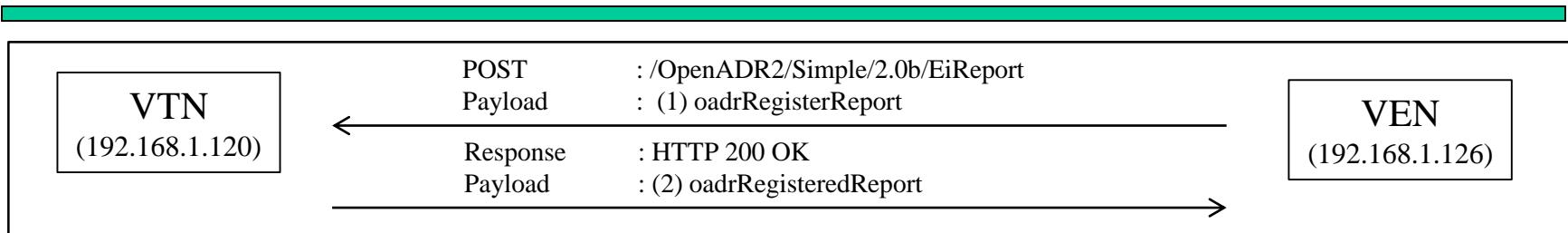


(1) oadrRegisterReport

Key name		Comments
venID		requested VEN ID
requestID		request identifier
oadrReport	duration	report duration
	reportRequestID	report request identifier
	reportSpecifierID	report specific id (created from ven)
	reportName	report name
	createdDateTime	created time of this report
	rID	
	resourceID	resource identifier
	reportType	report type
	itemUnits	unit of item that report
	siScaleCode	
reportDescription	marketContext	refer marketContext address
	oadrMinPeriod	Energy usage minimum period
	oadrMaxPeriod	Energy usage maximum period
	oadrOnChange	
	itemDescription	type of item that report
	powerAttributes	hertz pulse frequency of power
		voltage voltage of power
		ac Is this AC power? (True or False)
service		message type

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport ([HTTP / XML](#))

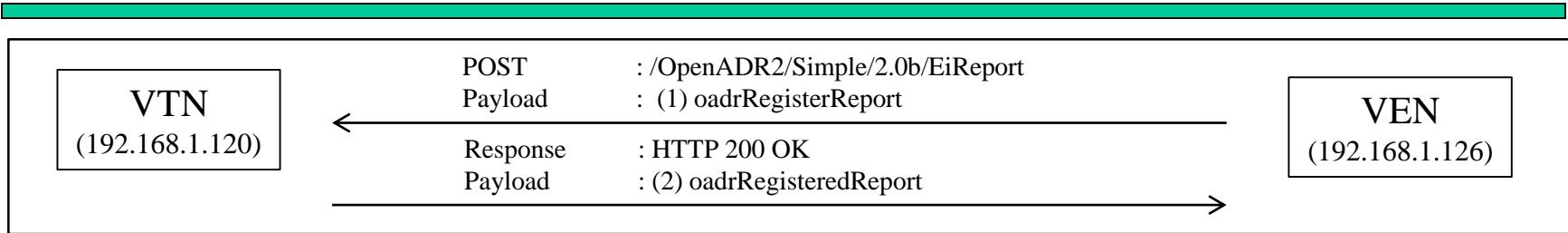


(2) oadrRegisteredReport

Key name	Comments
venID	requested VEN ID
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	message type

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (HTTP / XML)



### oadrRegisterReport XML {

```
"requestID": String,  
"oadrReport" : Array,  
"venID" : String  
}
```

### oadrReportDescription Array {

```
"rID" : String,  
"resourceID" : String,  
"reportType" : String,  
"itemUnits" : String,  
"siScaleCode" : String,  
"marketContxt" : String,  
"oadrMinPeriod" : String,  
"oadrMaxPeriod" : String,  
"oadrOnChange" : Boolean,  
"itemDescription" : String.  
"powerAttributes" : Array  
}
```

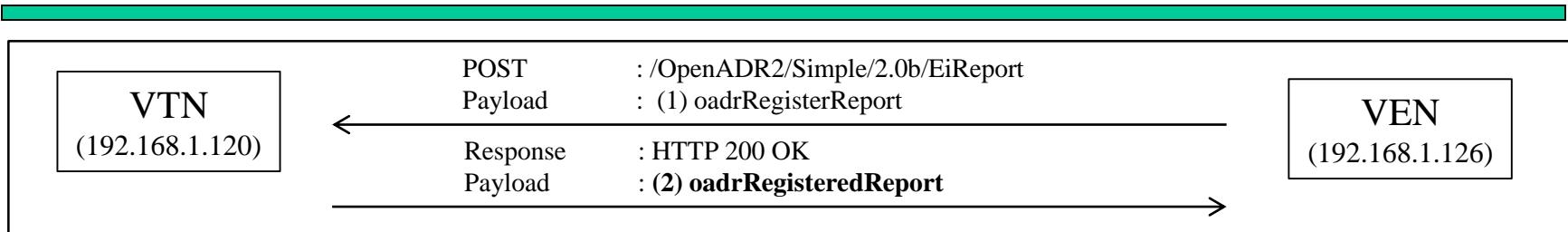
```
oadrReport Object{  
    "duration" : String,  
    "reportRequestID" : Integer,  
    "reportSpecifierID" : String,  
    "reportName" : String,  
    "createdDateTime" : Date,  
    "oadrReportDescription" : Array,
```

```
powerAttributes Array{  
    "hertz" : Integer,  
    "voltage" : Integer,  
    "ac" : Boolean  
}
```

```
<p2:requestID>  
C59517182C  
</p2:requestID>  
<p1:oadrReport  
xmlns:p3="urn:ietf:params:xml:ns:icalendar-2.0"  
xmlns:p4="http://docs.oasis-open.org/ns/energyinterop/201110">  
<p3:duration>  
<p1:oadrReportDescription  
<p1:oadrReportDescription  
<p4:reportRequestID>  
<p4:reportSpecifierID>  
<p4:reportName>  
<p4:createdDateTime>  
</p1:oadrReport>  
<p1:oadrReport  
<p3:venID>  
</p1:oadrRegisterReport>
```

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (HTTP / XML)



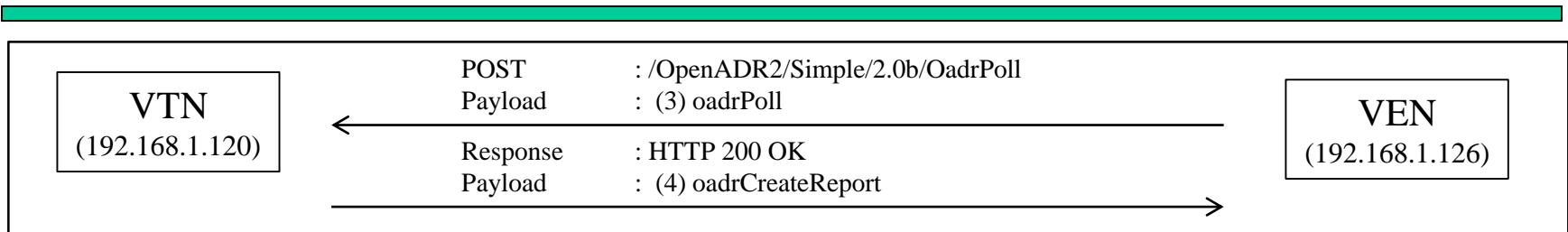
(1) oadrRegisterReport	HTTP/XML	192.168.1.126	192.168.1.120	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
(2) oadrRegisteredReport	HTTP/XML	192.168.1.120	192.168.1.126	HTTP/1.1 200 OK

```
oadrRegisteredReport XML{
    "responseCode": Integer,
    "responseDescription": String,
    "requestID": String,
    "venID": String
}
```

```
<ns3:oadrRegisteredReport
    ns5:schemaVersion="2.0b">
    <ns5:eiResponse>
        <ns5:responseCode>
        <ns5:responseDescription>
        <ns4:requestID>
            </ns5:eiResponse>
    </ns3:oadrRegisteredReport>
```

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport ([HTTP / XML](#))



(3) oadrPoll

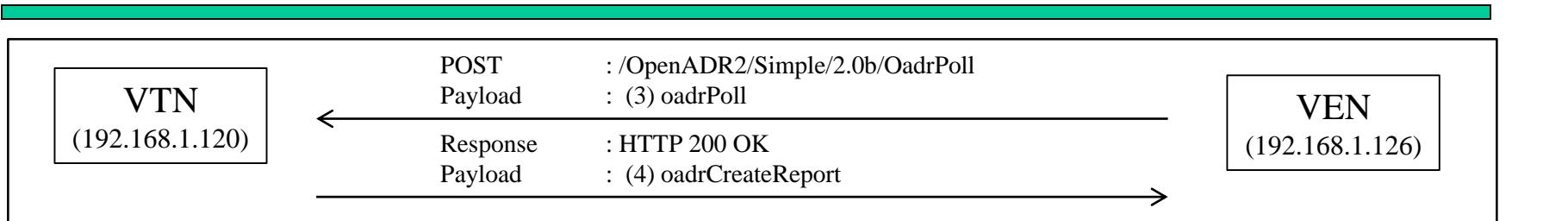
Key name	Comments
venID	requested VEN ID
service	message type

(4) oadrCreateReport

Key name	Comments
venID	requested VEN ID
requestID	request identifier
reportRequestID	report request identifier
reportSpecifierID	report specific id(create from ven)
duration	report duration
dtstart	report start time
rID	
readingType	reading type xml or json
service	message type

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (HTTP / XML)



```
oadrPoll XML{
    "venID" : String
}
```

```
<oadrPoll
    d3p1:schemaVersion="2.0b"
    xmlns:d3p1="http://docs.oasis-open.org/ns/energyinterop/201110">
    <d3p1:venID>
        72f8bbdcc0e87e6417de
    </d3p1:venID>
</oadrPoll>
```

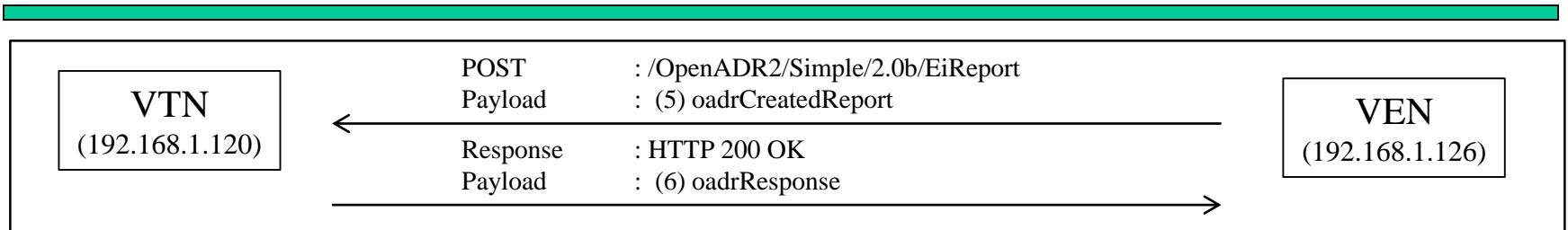
```
oadrCreateReport XML{
```

```
    "requestID" : String,
    "reportRequestID" : String,
    "reportSpecifierID" : String,
    "duration" : String,
    "dtstart" : Date,
    "rID" : String,
    "readingType" : String
}

<ns4:oadrCreateReport
    ns3:schemaVersion="2.0b">
    <ns2:requestID>
    <ns4:oadrReportRequest>
        <ns3:reportRequestID>
            10f8df64623771318528
        </ns3:reportRequestID>
        <ns3:reportSpecifier>
            <ns3:reportSpecifierID>
                789ed6cd4e_telemetry_usage
            </ns3:reportSpecifierID>
            <ns6:granularity>
                <ns6:duration>
                    PT2S
                </ns6:duration>
                </ns6:granularity>
            <ns3:reportBackDuration>
            <ns3:reportInterval>
            <ns3:specifierPayload>
```

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport ([HTTP / XML](#))



(5) `oadrCreatedReport`

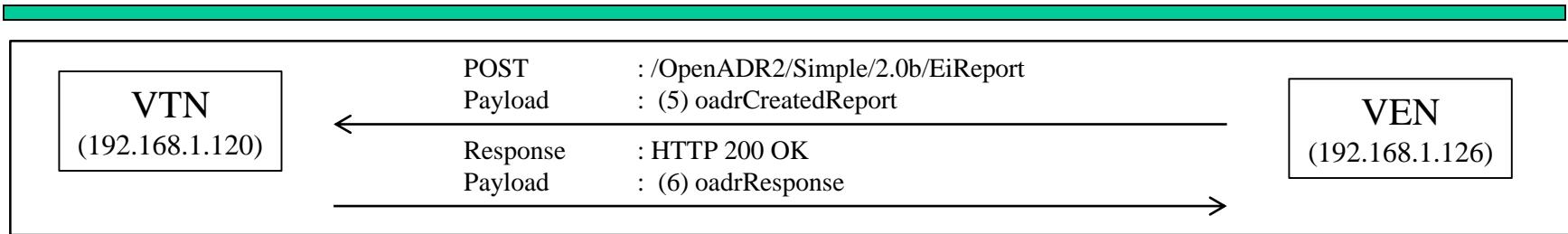
Key name	Comments
venID	requested VEN ID
requestID	request identifier
responseCode	response code
reportRequestID	report request identifier
service	message type

(6) `oadrResponse`

Key name	Comments
venID	requested VEN ID
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	message type

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (HTTP / XML)



(5) oadrCreatedReport	HTTP/XML	192.168.1.126	192.168.1.120	POST /OpenADR2/Simple/2.0b/EiReport	HTTP/1.1
(6) oadrResponse	HTTP/XML	192.168.1.120	192.168.1.126	HTTP/1.1	200 OK

```
oadrCreatedReport XML{
    "responseCode" : Integer,
    "requestID" : String,
    "reportRequestID" : String,
    "venID" : String
}
```

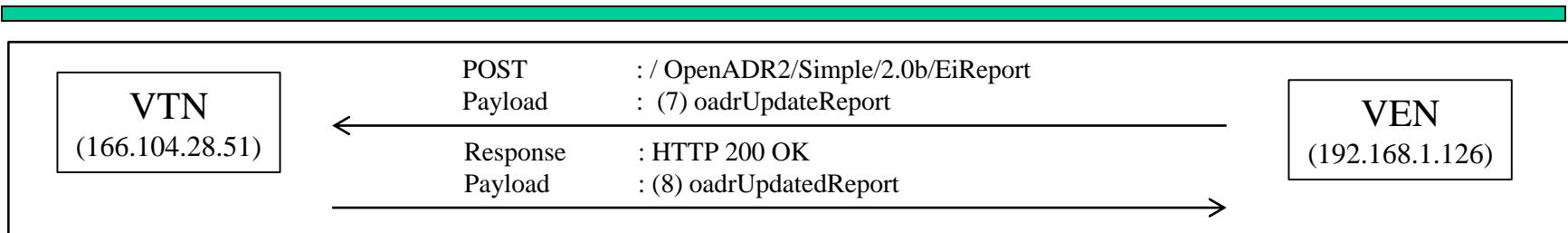
```
    ▲ <oadrCreatedReport
        d3p1:schemaVersion="2.0b"
        xmlns:d3p1="http://docs.oasis-open.org/ns/energyinterop/201110">
    ▲ <d3p1:eiResponse>
        ▲ <d3p1:responseCode>
            200
        </d3p1:responseCode>
    ▲ <requestID
        xmlns="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
        0ad9bb7fc8c5224c38d
    </requestID>
    </d3p1:eiResponse>
    ▲ <oadrPendingReports>
        ▲ <d3p1:reportRequestID>
            10f8df64623771318528
        </d3p1:reportRequestID>
    </oadrPendingReports>
    ▶ <d3p1:venID>
        </oadrCreatedReport>
```

```
oadrResponse XML{
    "responseCode" : Integer,
    "responseDescription" : String,
    "requestID" : String,
    "venID" : String
}
```

```
    ▲ <ns4:oadrResponse
        ns3:schemaVersion="2.0b">
    ▲ <ns3:eiResponse>
        ▷ <ns3:responseCode>
        ▷ <ns3:responseDescription>
        ▷ <ns2:requestID>
        </ns3:eiResponse>
    ▲ <ns3:venID>
        72f8bbdcc0e87e6417de
    </ns3:venID>
```

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (HTTP / XML)

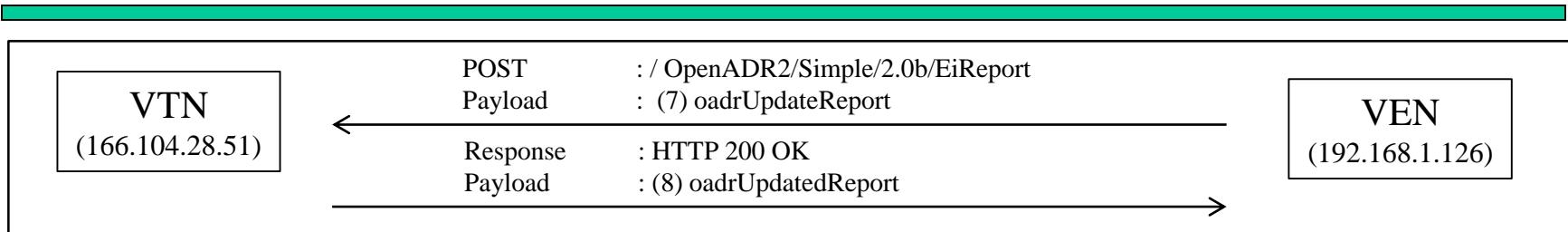


(7) oadrUpdateReport

	<b>Key name</b>	<b>Comments</b>
venID		requested VEN ID
requestID		request identifier
oadrReport	duration	report duration
	reportRequestID	report request identifier
	reportSpecifierID	report specific id (created from ven)
	reportName	report name
	createdDateTime	created time of this report
	reportDescription	rID
		resourceID
		reportType
		itemUnits
		siScaleCode
		marketContext
		oadrMinPeriod
		oadrMaxPeriod
		oadrOnChange
		itemDescription
service	powerAttributes	hertz
		voltage
		ac
		Is this AC power? (True or False)
		message type

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport ([HTTP / XML](#))

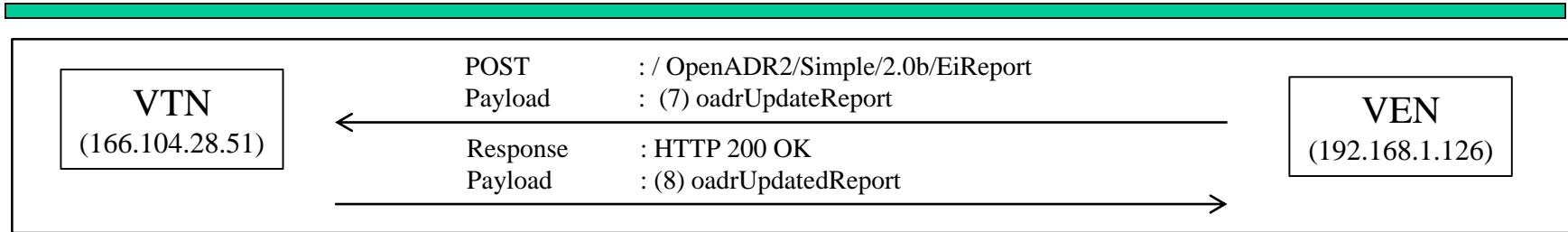


(8) oadrUpdatedReport

Key name	Comments
venID	requested VEN ID
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	message type

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (HTTP / XML)



(7) oadrUpdateReport  
 (8) oadrUpdatedReport

	HTTP/XML	192.168.1.126	166.104.28.51	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
HTTP/XML	166.104.28.51	192.168.1.126	HTTP/1.1 200 OK	

**oadrUpdateReport XML{**  
 “requestID”: String,  
**“oadrReport” : Array,**  
 “venID” : String  
`}`

**oadrReportDescription Array{**  
 “rID” : String,  
 “resourceID” : String,  
 “reportType” : String,  
 “itemUnits” : String,  
 “siScaleCode” : String,  
 “marketContxt” : String,  
 “oadrMinPeriod” : String,  
 “oadrMaxPeriod” : String,  
 “oadrOnChange” : Boolean,  
 “itemDescription” : String,  
**“powerAttributes” : Array**  
`}`

**powerAttributes Array{**  
 “hertz” : Integer,  
 “voltage” : Integer,  
 “ac” : Boolean  
`}`

**oadrReport Object{**  
 “duration” : String,  
 “reportRequestID” : Integer,  
 “reportSpecifierID” : String,  
 “reportName” : String,  
 “createdDateTime” : Date,  
**“reportDescription” : Array,**  
`}`

- <ns5:oadrUpdatedReport  
 ns1:schemaVersion="2.0b">
- <ns1:eiResponse>
  - ▷ <ns1:responseCode>
  - ▷ <ns1:responseDescription>
  - ▷ <ns1:requestID>
- <ns1:venID>
- </ns5:oadrUpdatedReport>

**oadrUpdatedReport XML{**  
 “reponseCode” : Integer,  
 “responseDescription” : String,  
 “requestID” : String,  
 “venID” : String  
`}`

```

<p1:oadrUpdateReport
  xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110"
  p3:schemaVersion="2.0b"
  xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payloads"
  >
  <p2:requestID>
  <p1:oadrReport
    xmlns:p3="urn:ietf:params:xml:ns:icalendar-2.0:stream"
    xmlns:p4="http://docs.oasis-open.org/ns/energyinterop/201110">
    <p3:intervals
      xmlns:p4="http://docs.oasis-open.org/ns/energyinterop/201110">
      <p4:interval
        </p3:intervals>
      <p4:reportRequestID>
      <p4:reportSpecifierID>
      <p4:reportName>
      <p4:createdDateTime>
        </p1:oadrReport>
      <p3:venID>
        </p1:oadrUpdateReport>
    </p3:intervals>
  </p1:oadrReport>
</p2:requestID>

```

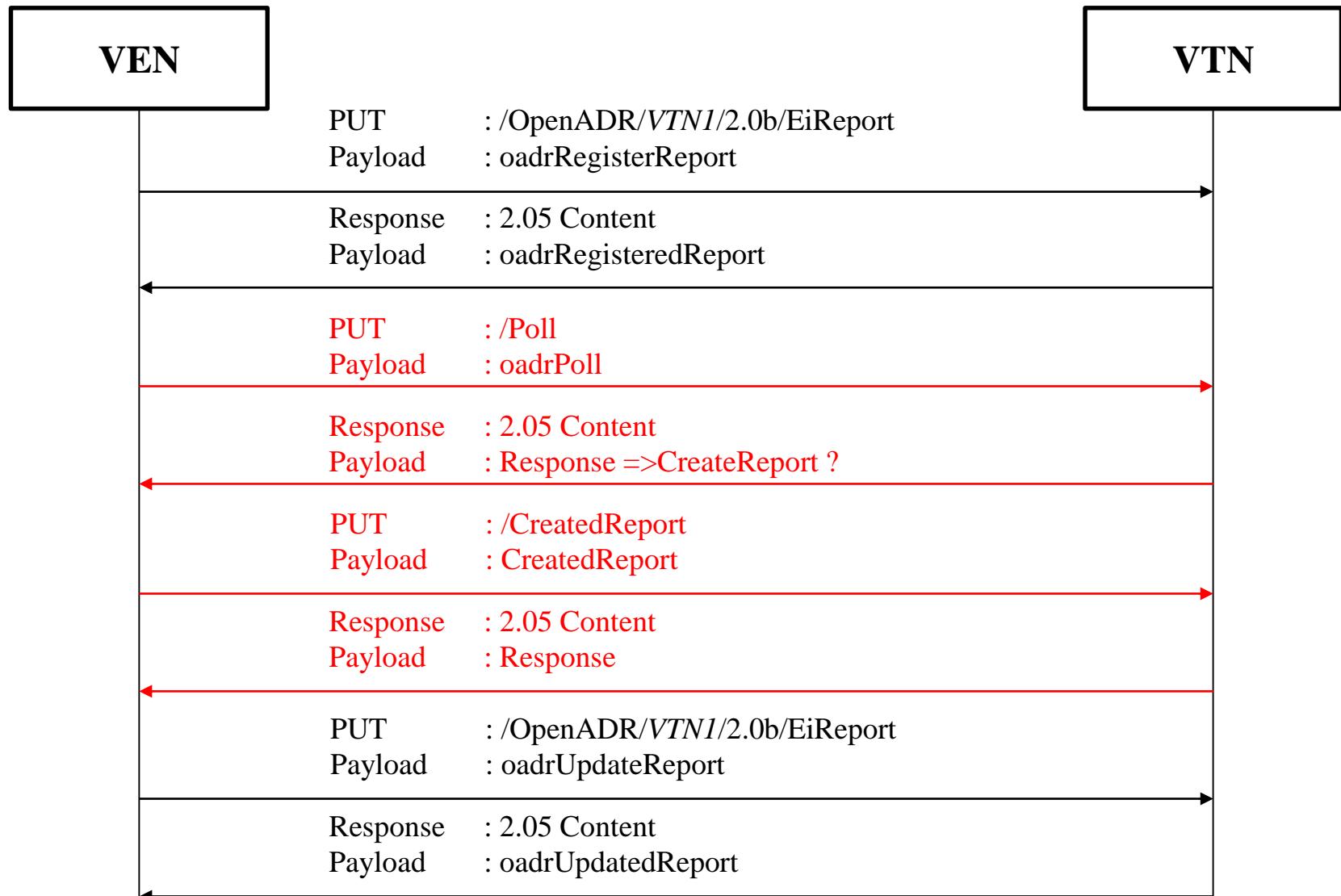
# **OpenADR 2.0b**

## **(2) EiReport**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

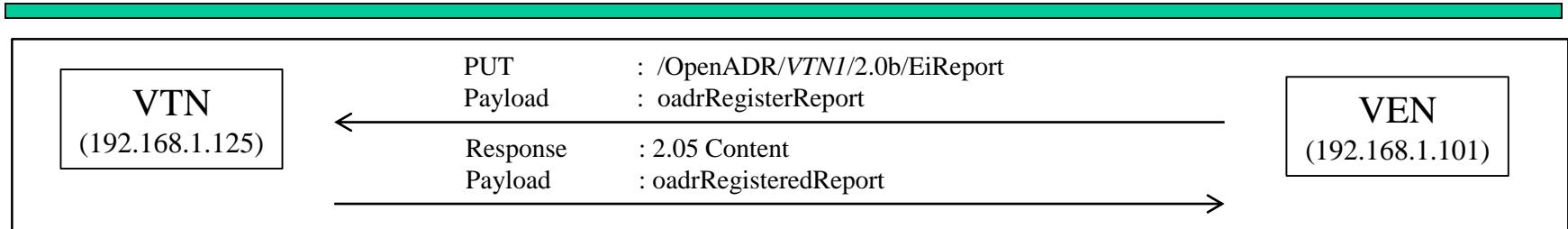
## 2. Profile : OpenADR 2.0b

### 2.2 Services : EiReport (CoAP / JSON)



# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (CoAP / JSON)



(1) RegisterReport	CoAP	192.168.1.101	192.168.1.125	CON, MID:46948, PUT, /RegisterReport
(2) RegisteredReport	CoAP	192.168.1.125	192.168.1.101	ACK, MID:46948, 2.05 Content (text/plain)

```
oadrRegisterReport JSON{
  "requestID": String,
  "oadrReport" : Array,
  "venID" : String,
  "service" : String
}
```

```
oadrReport Object{
  "duration" : String,
  "reportRequestID" : Integer,
  "reportSpecifierID" : String,
  "reportName" : String,
  "createdDateTime" : Date,
  "reportDescription" : Array,
}
```

```
powerAttributes Array{
  "hertz" : Integer,
  "voltage" : Integer,
  "ac" : Boolean
}
```

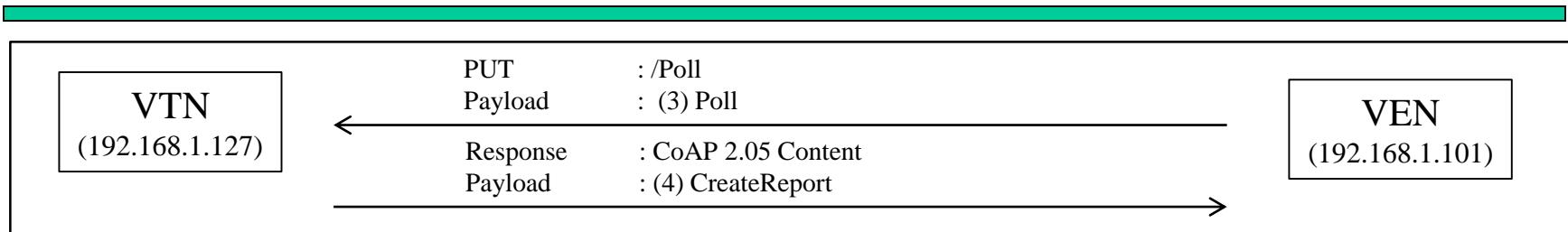
```
oadrRegisteredReport JSON{
  "responseCode": Integer,
  "responseDescription": String,
  "requestID": String,
  "service" : String
}
```

```
oadrReportDescription Array{
  "rID" : String,
  "resourceID" : String,
  "reportType" : String,
  "itemUnits" : String,
  "siScaleCode" : String,
  "marketContxt" : String,
  "oadrMinPeriod" : String,
  "oadrMaxPeriod" : String,
  "oadrOnChange" : Boolean,
  "itemDescription" : String,
  "powerAttributes" : Array
}
```

## 2. Profile : OpenADR 2.0b

Not implement yet

### 2.2 Services : EiReport (CoAP / JSON)



(3) Poll

(4) CreateReport

```
oadrPoll JSON{
    "venID": String,
    "service" : String
}
```

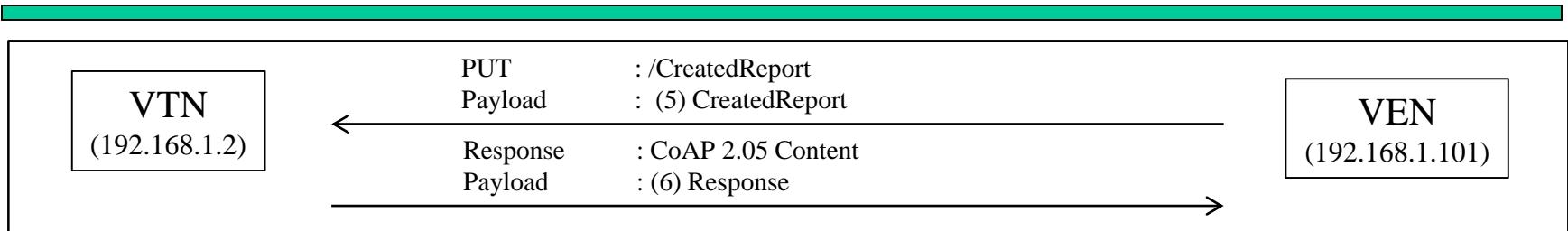
```
Poll{ "Service": "Poll", "GW": "gw\1", "VENID": "VEN_MIR1", "Re questID":  
1, "Ve rsion": 2 }
```

```
oadrCreateReport JSON{
    "requestID" : String,
    "reportRequestID" : String,
    "reportSpecifierID" : String,
    "duration" : String,
    "dtstart" : Date,
    "rID" : String,
    "readingType" : String,
    "service" : String
}
```

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (CoAP / JSON)

Not implement yet



(5) CreatedReport

(6) Response

**CreatedReport JSON{**

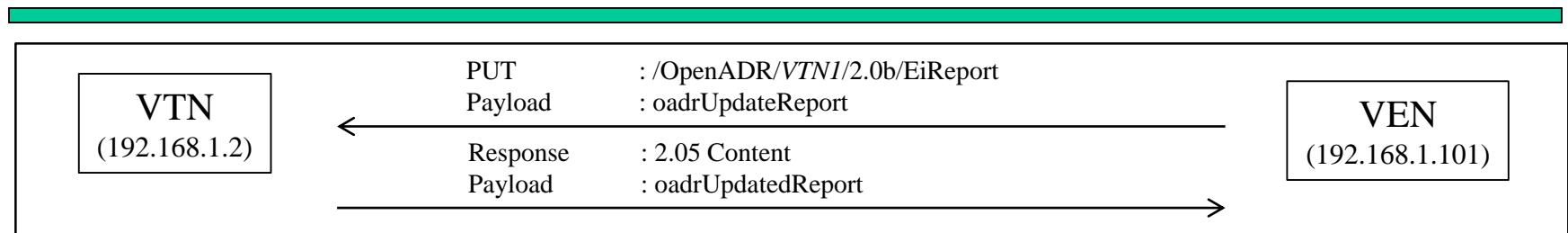
```
    "responseCode" : Integer,  
    "requestID" : String,  
    "reportRequestID" : String,  
    "venID" : String,  
    "service" : String  
}
```

**Response JSON{**

```
    "responseCode" : Integer,  
    "responseDescription" : String,  
    "requestID" : String,  
    "venID" : String,  
    "service" : String  
}
```

# 2. Profile : OpenADR 2.0b

## 2.2 Services : EiReport (CoAP / JSON)



(5) UpdateReport  
 (6) UpdatedReport

CoAP	192.168.1.101	192.168.1.127	CON, MID:25996, PUT, /UpdateReport
CoAP	192.168.1.127	192.168.1.101	ACK, MID:25996, 2.05 Content (text/plain)

**oadrUpdateReport JSON{**

```

    "requestID": String,
    "oadrReport" : Array,
    "venID" : String,
    "service" : String
  }
  
```

**oadrReportDescription Array{**

```

    "rID" : String,
    "resourceID" : String,
    "reportType" : String,
    "itemUnits" : String,
    "siScaleCode" : String,
    "marketContxt" : String,
    "oadrMinPeriod" : String,
    "oadrMaxPeriod" : String,
    "oadrOnChange" : Boolean,
    "itemDescription" : String,
    "powerAttributes" : Array
  }
  
```

**powerAttributes Array{**

```

    "hertz" : Integer,
    "voltage" : Integer,
    "ac" : Boolean
  }
  
```

**oadrReport Object{**

```

    "duration" : String,
    "reportRequestID" : Integer,
    "reportSpecifierID" : String,
    "reportName" : String,
    "createdDateTime" : Date,
    "reportDescription" : Array,
  }
  
```

**oadrUpdatedReport JSON{**

```

    "reponseCode" : Integer,
    "responseDescription" : String,
    "requestID" : String,
    "venID" : String,
    "service" : String
  }
  
```

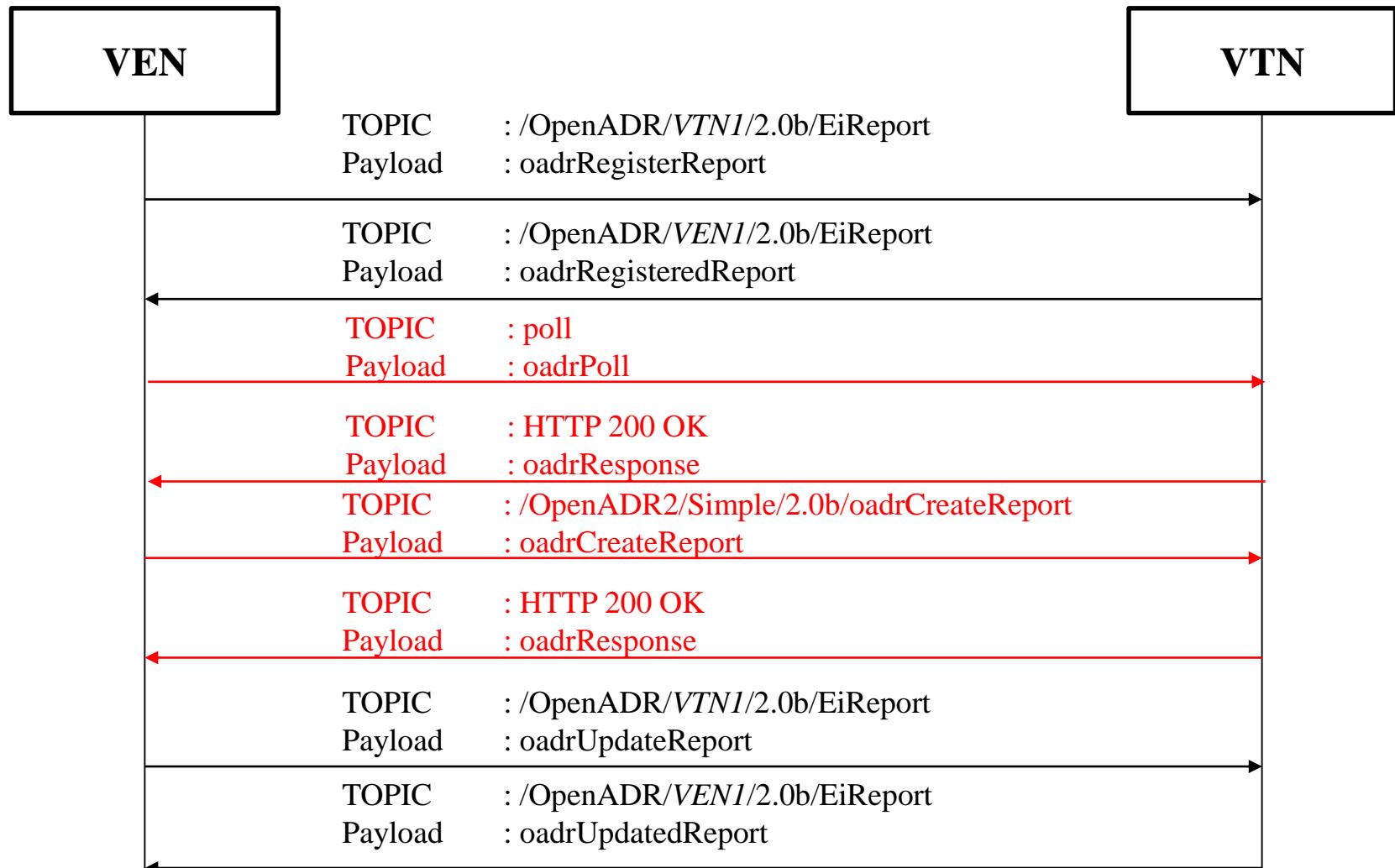
# **OpenADR 2.0b**

## **(2) EiReport**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

## 2. Profile : OpenADR 2.0b

### 2.2 Services : EiReport (MQTT / JSON)

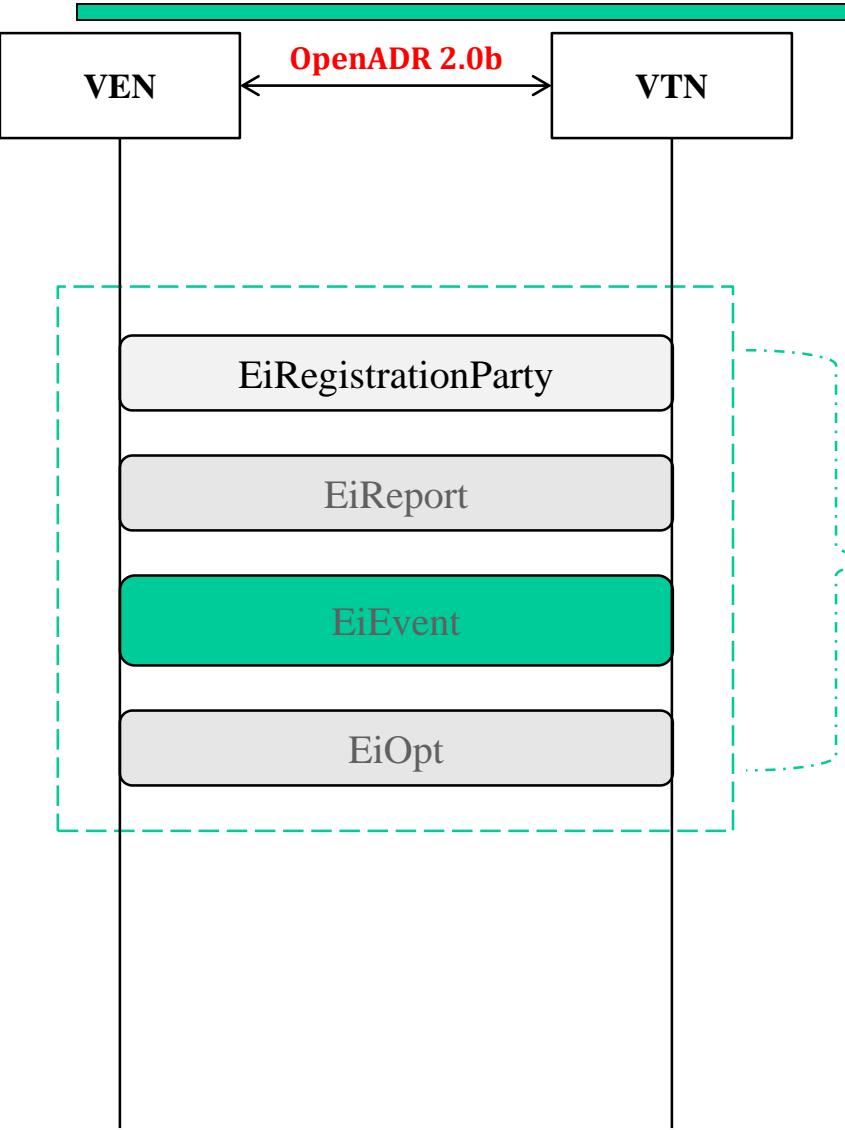


# **OpenADR 2.0b**

## **(3) EiEvent**

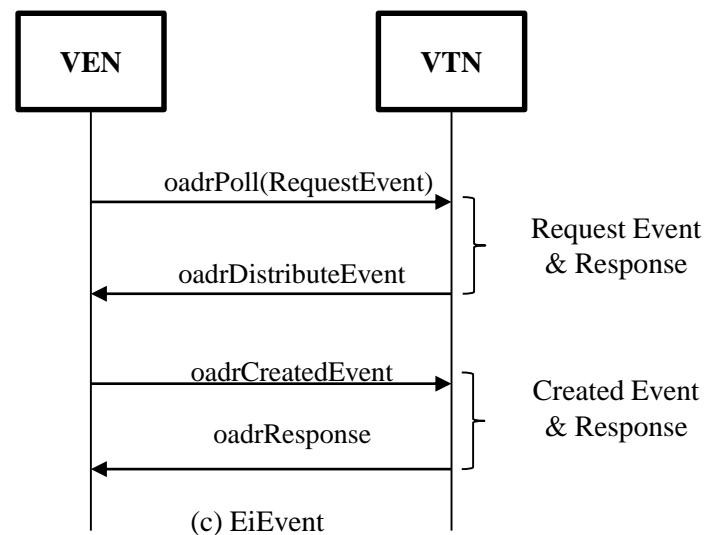
# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent



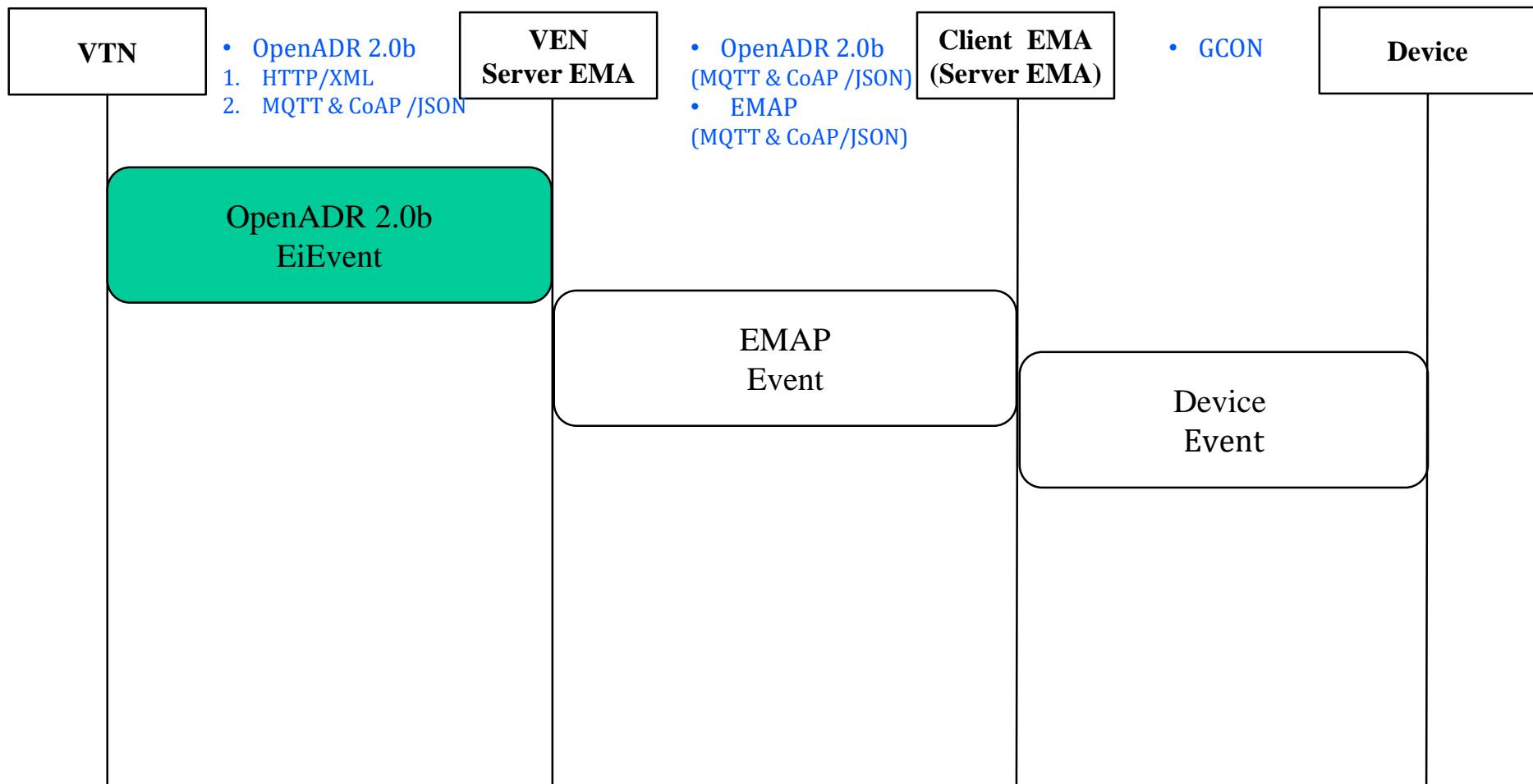
- **EiEvent**

- 수요자원의 사용패턴 변화를 위한 Event 신호를 내려주는 기능을 한다. Event 신호에는 Event시간, 대상, 내용(level, price, power등)에 대한 정보가 포함되어 수요반응을 만들어 낸다.



# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent



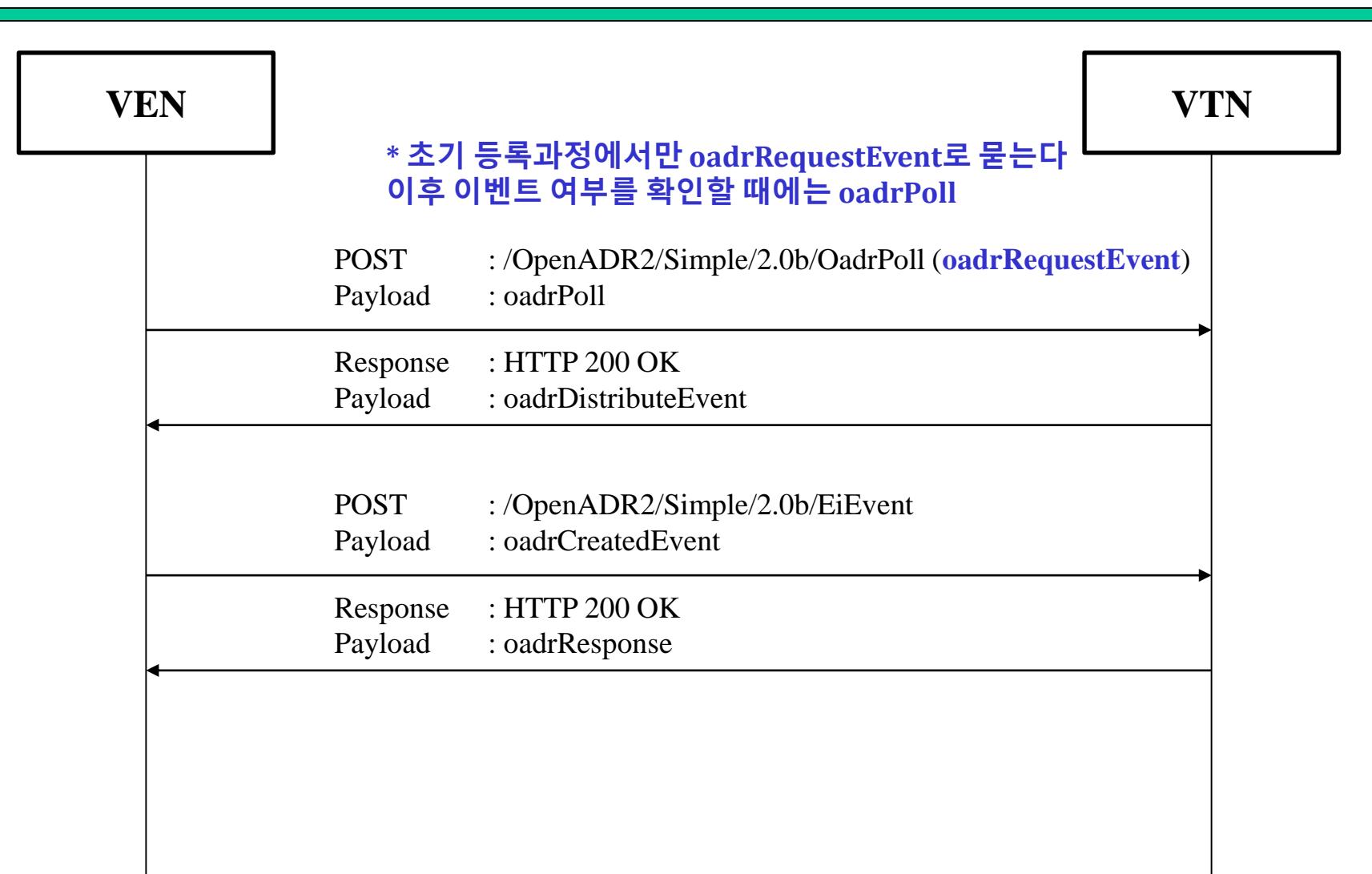
# **OpenADR 2.0b**

## **(3) EiEvent**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

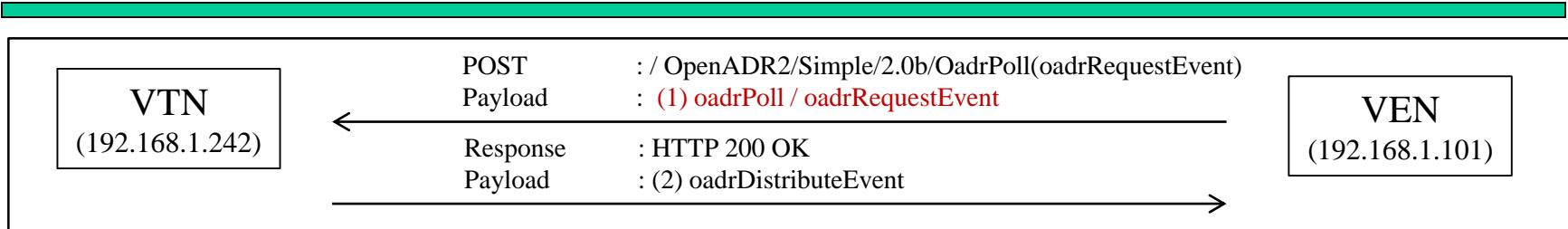
## 2. Profile : OpenADR 2.0b

### 2.3 Services : EiEvent ([HTTP / XML](#))



## 2. Profile : OpenADR 2.0b

### 2.3 Services : EiEvent ([HTTP / XML](#))



(1) oadrPoll

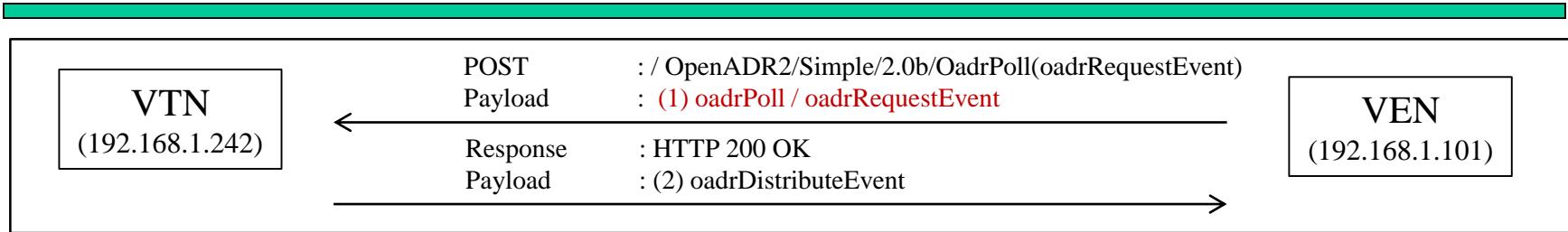
Key name	Comments
venID	requested VEN ID
service	message type

oadrRequestEvent

Key name	Comments
venID	requested VEN ID
requestID	request identifier
service	message type

# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent ([HTTP / XML](#))

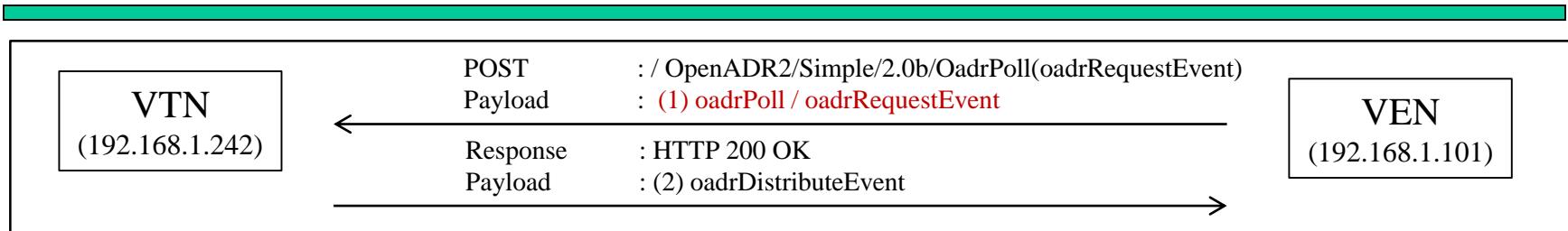


### (2) oadrDistributeEvent

Key name		Comments	
venID		requested VEN ID	
response	requestID	duration	event signal interval duration
	responseCode	uid	event user id
	responseDescription	value	event value
event	eventID	signalName	event signal name
	eventSignals	signalType	event signal type (bi direct, level)
		signalID	event signal ID
		currentValue	current usage value
		modificationNumber	modification Number(count)
	modificationReason	modificationReason	modification reason(event reason)
		priority	priority
		eiMarketContext	market address(market reference)
	createdDateTime	event create date & time	
	eventStatus	event status	
	testEvent	if event test or not	
	vtnComment		
	dtstart	event start time	
	duration	event duration	
	properties		
	components		
	venID	ven ID	
	tolerance	tolerance duration	
	notification	notification duration	
	rampUp	ramp up duration	
	recovery		
oadrResponseRequired		response mandatory or not	
service		message type	

# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent (HTTP / XML)



(1) oadrPoll	HTTP/XML	192.168.1.101	192.168.1.242	POST /OpenADR2/Simple/2.0b/OadrPoll	HTTP/1.1
(2) oadrDistributeEvent	HTTP/XML	192.168.1.242	192.168.1.101	HTTP/1.1 200	OK

```
oadrPoll XML{
    "venID" : String
}
    ↳ <p1:oadrPoll
        xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110"
        p2:schemaVersion="2.0b">
        ↳ <p2:venID>
            cc0853370e25e04d58e3
        </p2:venID>
    </p1:oadrPoll>
```

```
oadrRequestEvent XML{
    "venID" : String,
    "requestID": String,
}
    ↳ <oadrSignedObject>
        ↳ <oadrRequestEvent
            d3p1:schemaVersion="2.0b"
            xmlns:d3p1="http://docs.oasis-open.org/ns/energyinterop/201110">
                ↳ <eiRequestEvent
                    xmlns="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
                        ↳ <requestID>
                            459abd7343
                        </requestID>
                        ↳ <requestID>
                            459abd7343
                        </requestID>
                    </eiRequestEvent>
                </oadrRequestEvent>
            </oadrSignedObject>
```

\* 초기 등록과정에서만 oadrRequestEvent를 룰는다  
이후 이벤트 여부를 확인할 때에는 oadrPoll

Session Setup

Distribute Event (request Event)

vtnCommnet = "SessionSetup"

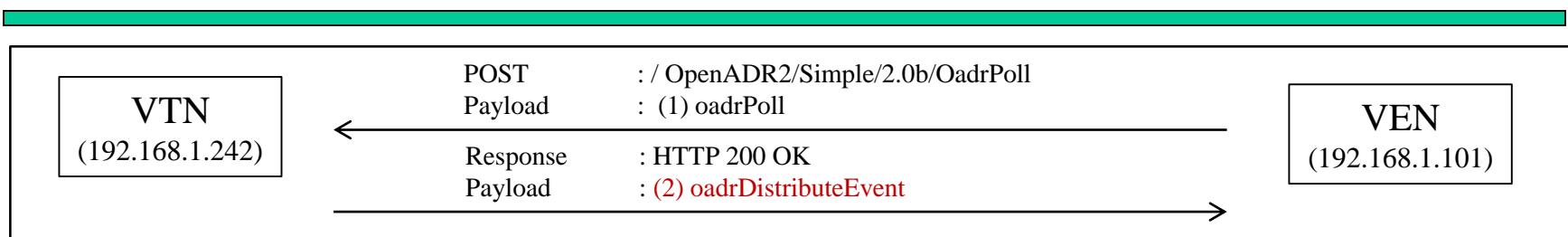
Poll & Push

Distribute Event

vtnComment = "Event"

# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent ([HTTP / XML](#))



(1) oadrPoll	HTTP/XML	192.168.1.101	192.168.1.242	POST /OpenADR2/Simple/2.0b/OadrPoll	HTTP/1.1
(2) oadrDistributeEvent	HTTP/XML	192.168.1.242	192.168.1.101	HTTP/1.1	200 OK

```
oadrDistributeEvent XML{
  "requestID": String,
  "vtnID": String,
  "response": Array,
  "event": Array,
  "oadrResponseRequired": String
}
```

```
  ▾ <ns1:oadrDistributeEvent
    ns5:schemaVersion="2.0b">
      ▷ <ns5:eiResponse>
      ▷ <ns6:requestID>
      ▷ <ns5:vtnID>
      ▷ <ns1:oadrEvent>
    </ns1:oadrDistributeEvent>
```

```
event Array{
  "eventID": String,
  "eventSignals": Array,
  "modificationNumber": Integer,
  "modificationReason": String,
  "priority": Integer,
  "eiMarketContext": String,
  "createdDataTime": Date,
  "eventStatus": String,
  "testEvent": Boolean,
  "vtnComment": String,
  "dtstart": Date,
  "duration": String,
  "properties": String,
  "components": String,
  "venID": String,
  "tolerance": String,
  "notification": String,
  "rampUp": String,
  "recovery": String
}
```

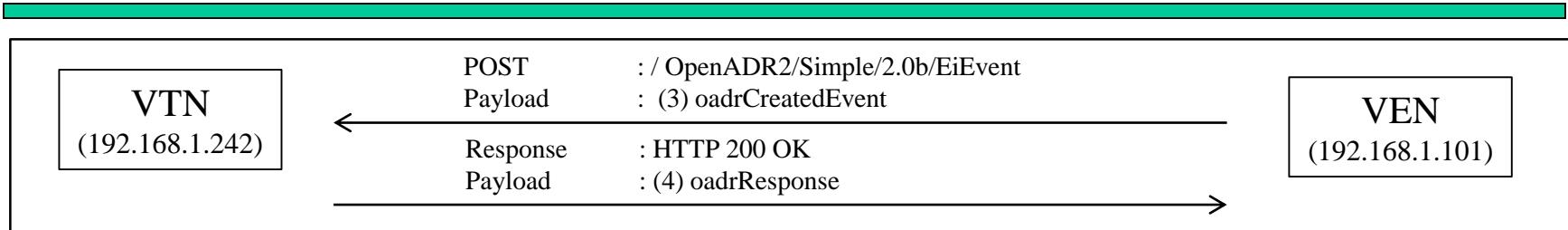
```
eventSignals Array{
  "intervals": Array,
  "signalName": String,
  "signalType": String,
  "signalID": String,
  "currentValue": Double
}
```

```
intervals Array{
  "duration": String,
  "uid": Integer,
  "value": Double
}
```

```
response Array{
  "requestID": String,
  "responseCode": Integer,
  "responseDescription": String
}
```

# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent ([HTTP / XML](#))



(3) oadrCreatedEvent

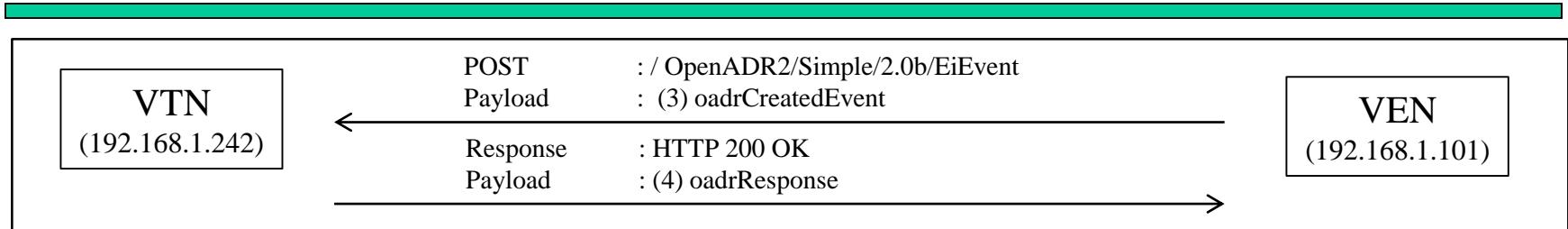
Key name	Comments
vtnID	responded VTN ID
venID	requested VEN ID
requestID	request identifier
responseCode	response code
responseDescription	description of response code
eventID	Event identifier
modificationNumber	modification number(count)
optType	if participate event or not
service	message type

(4) oadrResponse

Key name	Comments
venID	requested VEN ID
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	message type

# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent ([HTTP / XML](#))



(3) oadrCreatedEvent	HTTP/XML	192.168.1.101	192.168.1.242	POST /OpenADR2/Simple/2.0b/EiEvent	HTTP/1.1
(4) oadrResponse	HTTP/XML	192.168.1.242	192.168.1.101	HTTP/1.1 200 OK	

```
oadrCreatedEvent XML{
    "responseCode": Integer,
    "responseDescription": String,
    "requestID": String,
    "vtnID": String,
    "eventID": String,
    "modificationNumber": Integer,
    "optType": String,
    "venID": String
}
```

```
    ▲ <p1:oadrCreatedEvent
        xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110"
        p3:schemaVersion="2.0b"
        xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
        ▲ <p2:eiCreatedEvent
            xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110">
            ▲ <p3:eiResponse>
                ▷ <p3:responseCode>
                ▷ <p3:responseDescription>
                <p2:requestID/>
                ▷ <p3:eiResponse>
            ▷ <p3:eventResponses>
            ▲ <p3:venID>
                8461ba1b0cd8acac41a7
                </p3:venID>
            </p3:venID>
        </p2:eiCreatedEvent>
```

```
oadrResponse XML{
    "venID": String,
    "requestID": Integer,
    "responseCode": Integer,
    "responseDescription": String
}
```

```
    ▲ <ns1:oadrResponse
        ns5:schemaVersion="2.0b">
        ▲ <ns5:eiResponse>
            ▷ <ns5:responseCode>
            ▷ <ns5:responseDescription>
            ▷ <ns6:requestID>
            </ns5:eiResponse>
        ▷ <ns5:venID>
    </ns1:oadrResponse>
```

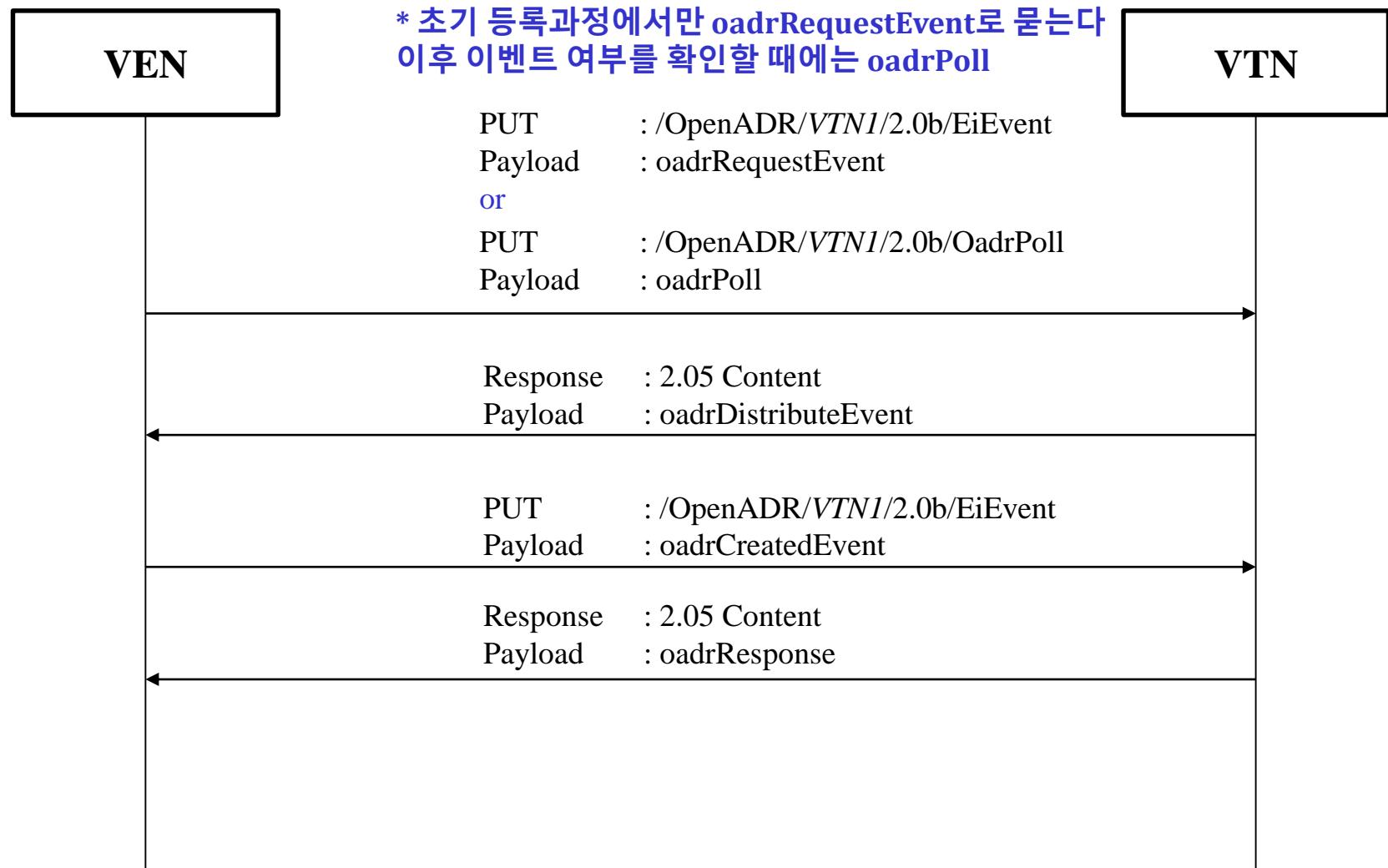
# **OpenADR 2.0b**

## **(3) EiEvent**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent (**CoAP / JSON**)



Session Setup

Distribute Event (request Event)

vtnCommnet = "SessionSetup"

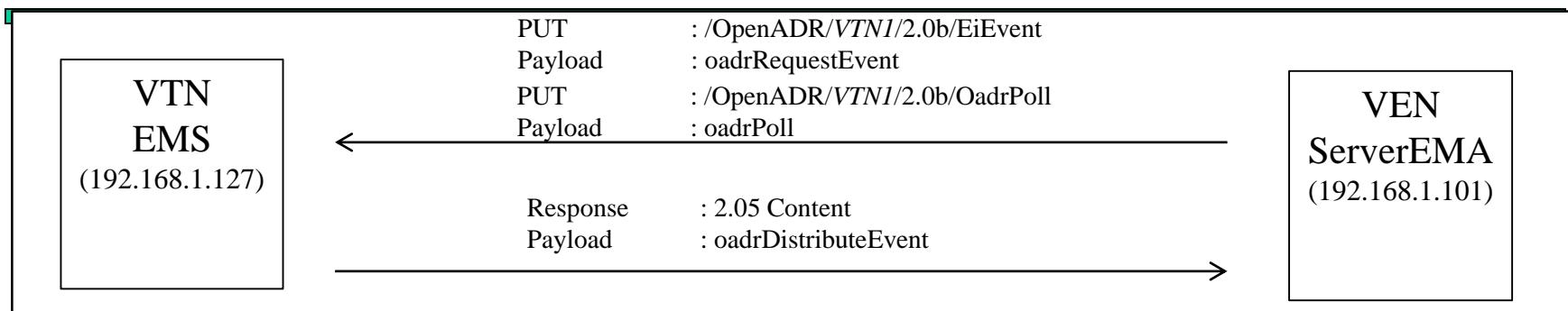
Poll & Push

Distribute Event

vtnComment = "Event"

## 2. Profile : OpenADR 2.0b

### 2.3 Services : EiEvent (CoAP / JSON)



(1) Poll

CoAP 192.168.1.101 192.168.1.127 CON, MID:35224, PUT, /Poll

(2) DistributeEvent

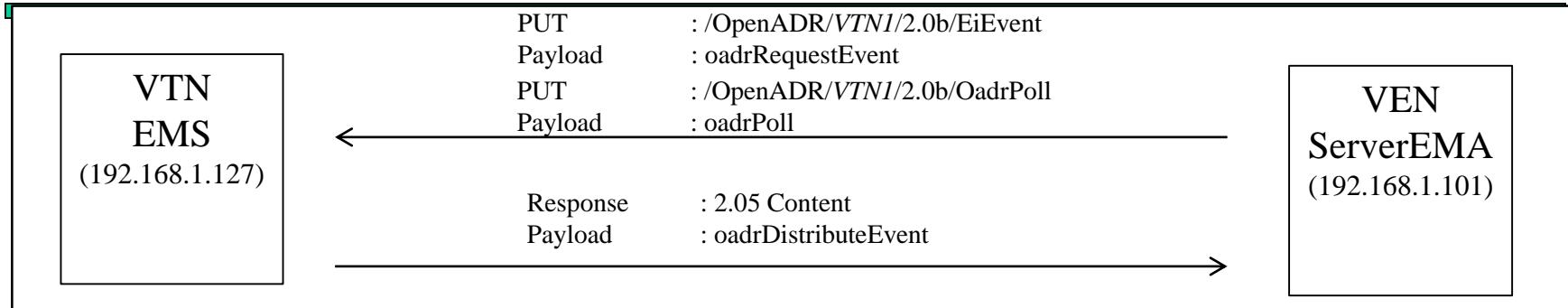
CoAP 192.168.1.127 192.168.1.101 ACK, MID:35224, 2.05 Content (text/plain)

```
oadrPoll JSON{
  "venID": String,
  "service" : String
}
```

Topic: EMS/oadrPoll/Poll/1  
Message: { "GW": "gw\1", "VENID": "VEN\_MIR1", "RequestID": 1, "Version": 2 }

# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent (CoAP / JSON)



(1) Poll

CoAP 192.168.1.101 192.168.1.127 CON, MID:35224, PUT, /Poll

(2) DistributeEvent

CoAP 192.168.1.127 192.168.1.101 ACK, MID:35224, 2.05 Content (text/plain)

```
oadrDistributeEvent JSON{
  "requestID": String,
  "vtnID": String,
  "response": Array,
  "event": Array,
  "oadrResponseRequired": String,
  "service": String
}
```

```
  {
    "Response": "MIR ",
    "EndTime": "194 1",
    "RequestID": 1,
    "StartTime": "19 40",
    "Service": "DistributeEvent",
    "EndID": "201702 09",
    "Response": 1,
    "Value": 1,
    "OptType": "optIn",
    "StartYMD": "2017020 9",
    "EventID": 1,
    "ModificationNumber": 0,
    "TargetVEN": "MIR_VEN1"
```

```
event Array{
  "eventID": String,
  "eventSignals": Array,
  "modificationNumber": Integer,
  "modificationReason": String,
  "priority": Integer,
  "eiMarketContext": String,
  "createdDataTime": Date,
  "eventStatus": String,
  "testEvent": Boolean,
  "vtnComment": String,
  "dtstart": Date,
  "duration": String,
  "properties": String,
  "components": String,
  "venID": String,
  "tolerance": String,
  "notification": String,
  "rampUp": String,
  "recovery": String
}
```

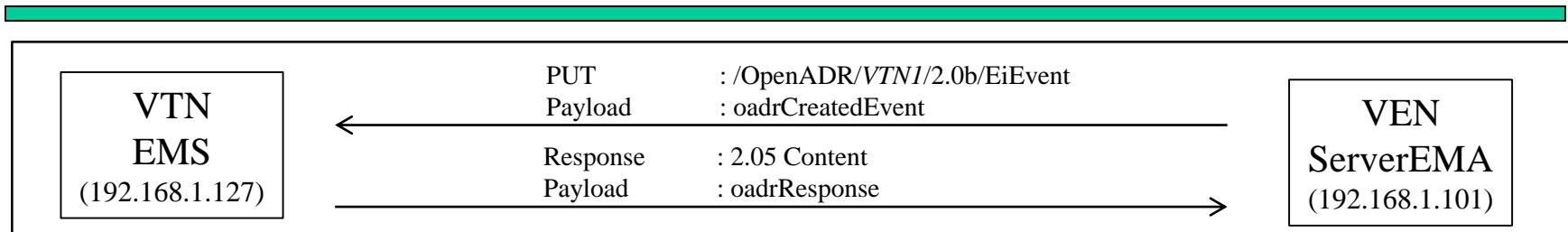
```
eventSignals Array{
  "intervals": Array,
  "signalName": String,
  "signalType": String,
  "signalID": String,
  "currentValue": Double
}
```

```
intervals Array{
  "duration": String,
  "uid": Integer,
  "value": Double
}
```

```
response Array{
  "requestID": String,
  "responseCode": Integer,
  "responseDescription": String
}
```

## 2. Profile : OpenADR 2.0b

### 2.3 Services : EiEvent (CoAP / JSON)



(3) CreatedEvent  
(4) Response

CoAP	192.168.1.101	192.168.1.127	CON, MID:864, PUT, /createdEvent
CoAP	192.168.1.127	192.168.1.101	ACK, MID:864, 2.05 Content (text/plain)

```
oadrCreatedEvent JSON{
  "responseCode": Integer,
  "responseDescription": String,
  "requestID": String,
  "vtnID": String,
  "eventID": String,
  "modificationNumber": Integer,
  "optType": String,
  "venID": String,
  "service": String
}
```

```
createdEvent{ "Service": "Create dEvent", "VENID" : "VEN_MIR01", "Version": 1,
  "Response": 1, "RequestID": 1, "Value": 1 }
```

```
oadrResponse JSON{
  "venID": String,
  "requestID": Integer,
  "responseCode": Integer,
  "responseDescription": String,
  "service": String
}
```

```
{"Response":200,"RequestID":1,"VENID":"VEN_MIR01","Service":"Response"}
```

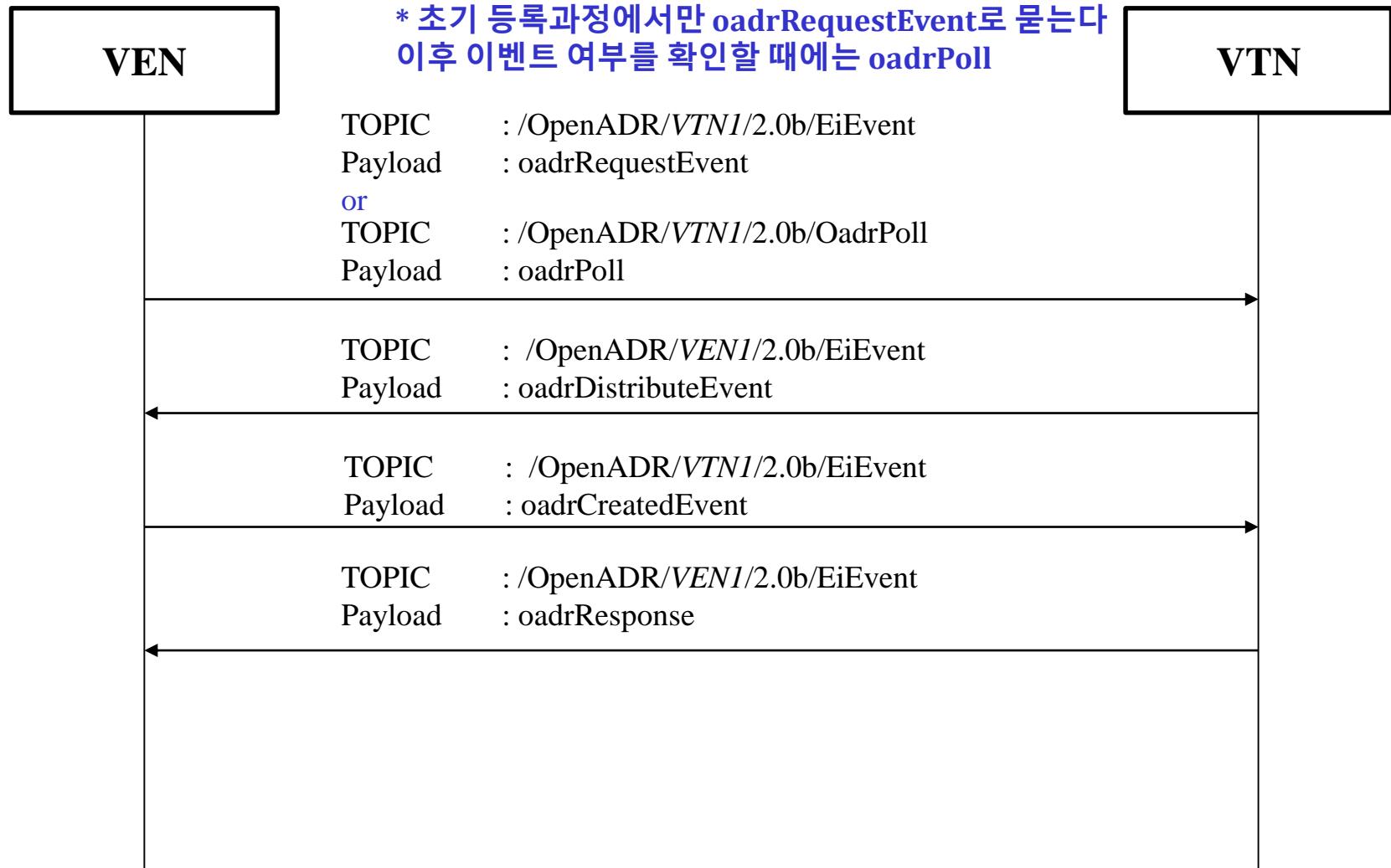
# **OpenADR 2.0b**

## **(3)EiEvent**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

## 2. Profile : OpenADR 2.0b

### 2.3 Services : EiEvent (MQTT / JSON)

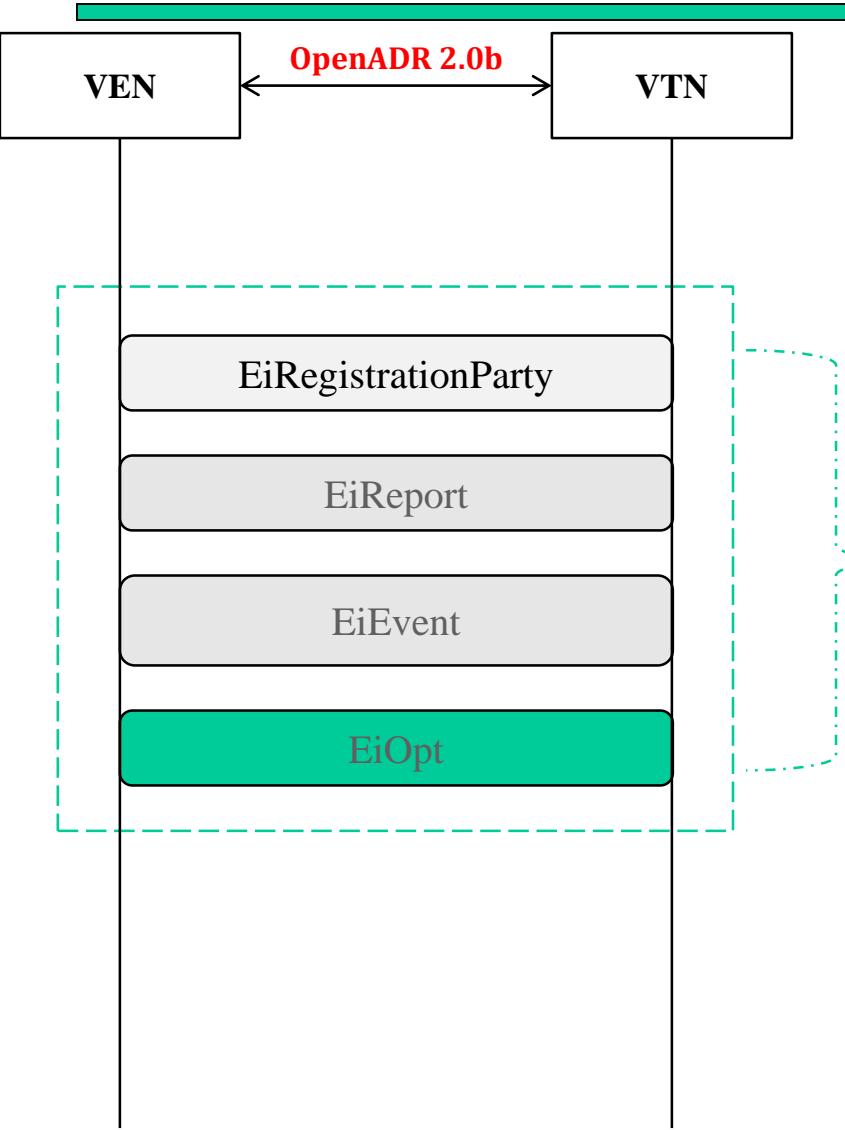


# **OpenADR 2.0b**

## **(4) EiOpt**

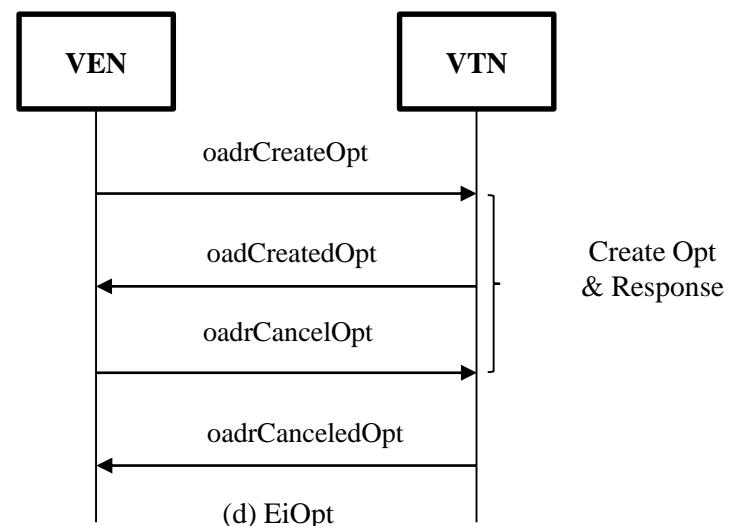
# 2. Profile : OpenADR 2.0b

## 2.4 Services : EiOpt



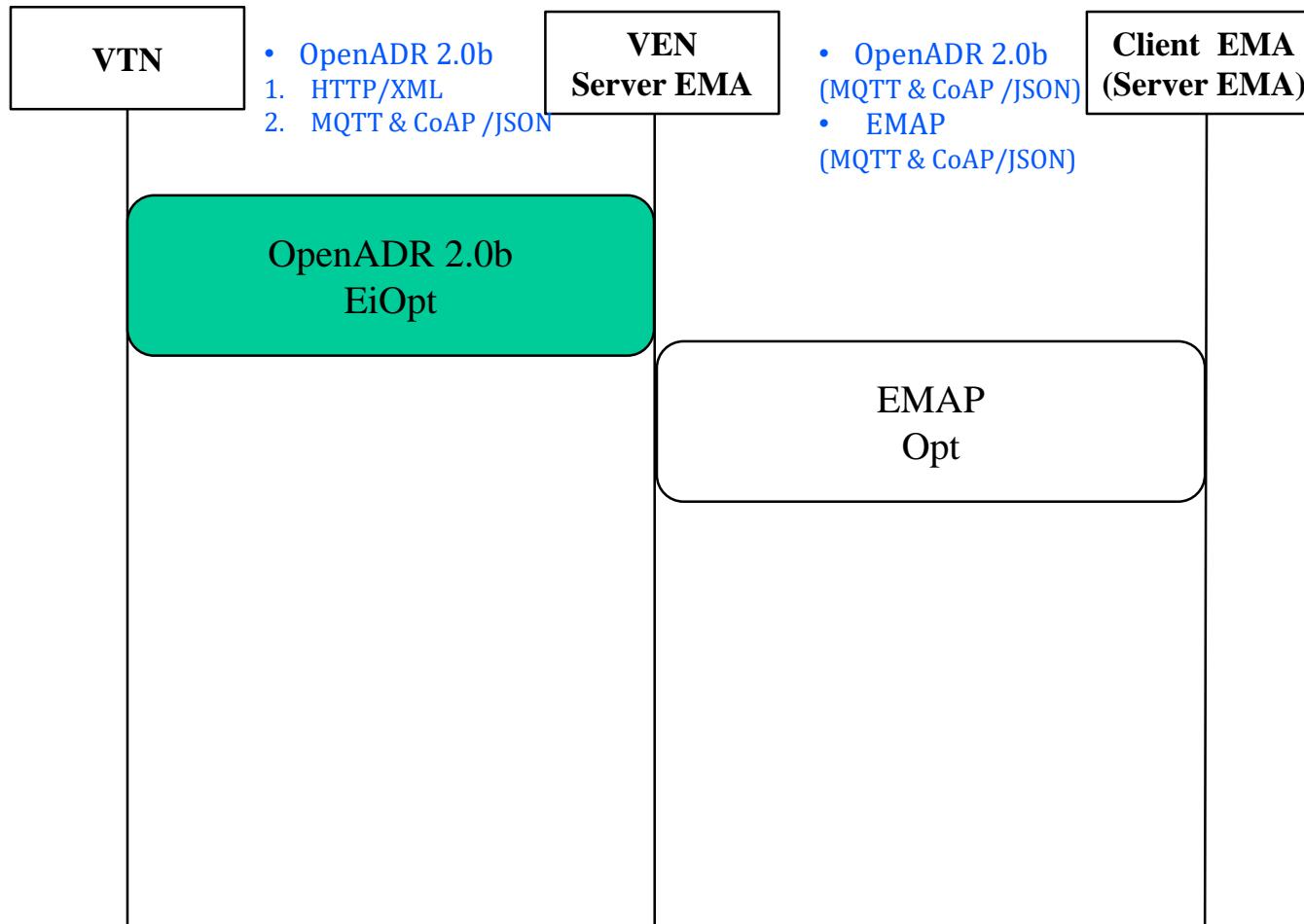
- EiOpt

- VEN이 지정한 시간 대에 Event에 참여 여부를 예약하는 서비스이다. Event를 수행 할 때는 Opt-in, 불가능 할 때는 Opt-out메시지를 VTN에게 보내어 VEN의 가용상태를 알려준다.



# 2. Profile : OpenADR 2.0b

## 2.4 Services : EiOpt



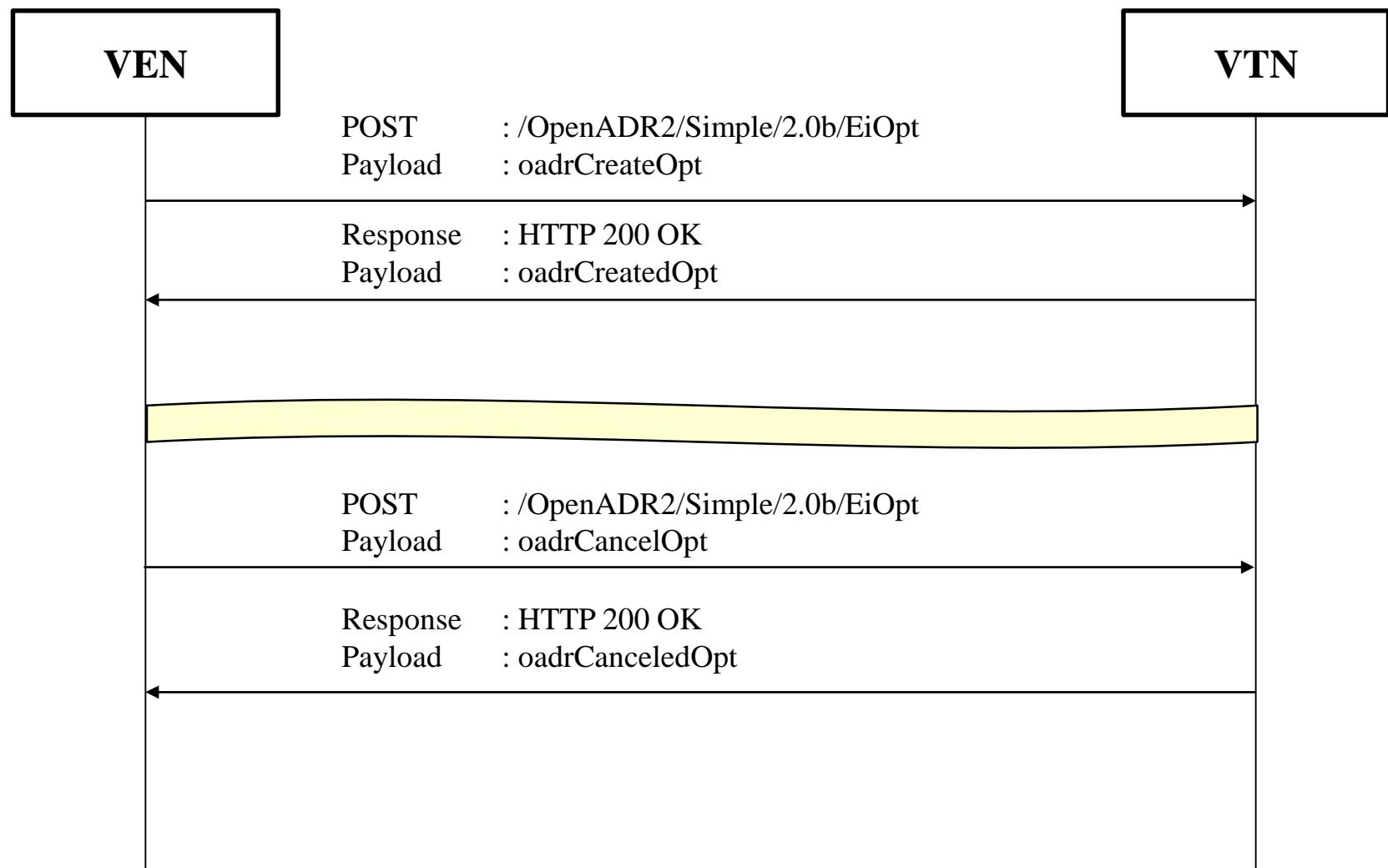
# **OpenADR 2.0b**

## **(4) EiOpt**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

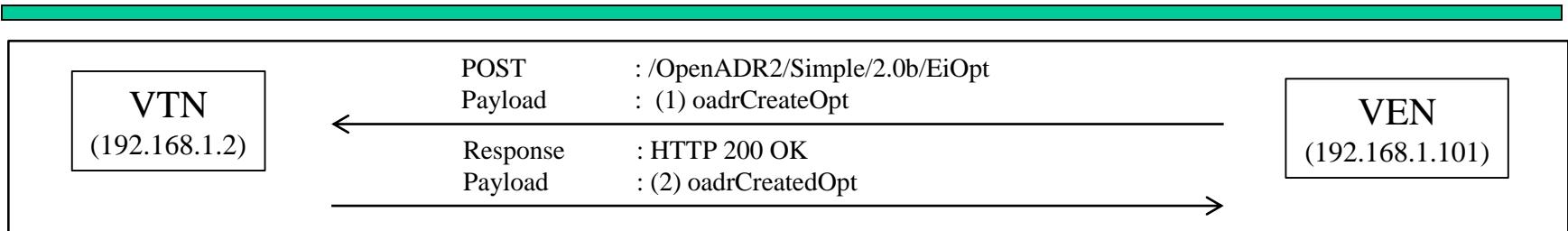
## 2. Profile : OpenADR 2.0b

### 2.4 Services : EiOpt ([HTTP / XML](#))



## 2. Profile : OpenADR 2.0b

### 2.4 Services : EiOpt (HTTP / XML)



(1) oadrCreateOpt

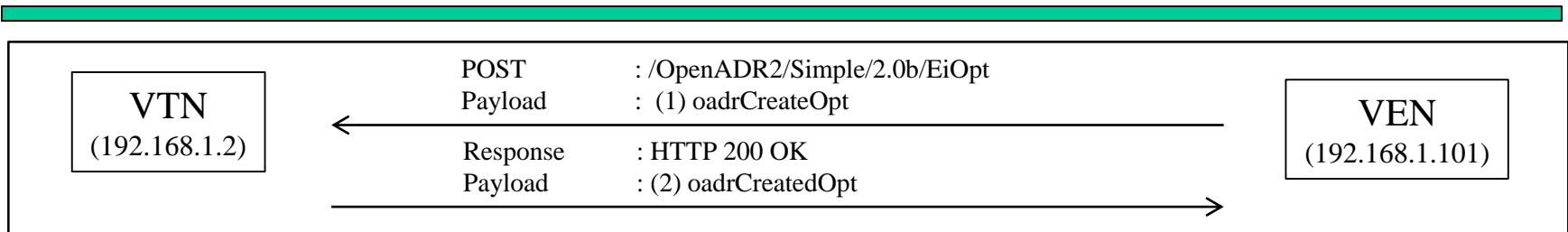
Key name	Comments	
venID	requested VEN ID	
optID	opt identifier	
optType	type of opt	
optReason	opt reason(e.g. emergency)	
marketContext	refer market address	
available	dtstart	opt start time
	duration	opt duration
createdDateTime	created time of this message	
requestID	request identifier	
service	message type	

(2) oadrCreatedOpt

Key name	Comments
requestID	request identifier
responseCode	response code
responseDescription	description of response code
optID	opt identifier
service	message type

# 2. Profile : OpenADR 2.0b

## 2.4 Services : EiOpt (HTTP / XML)



(1) oadrCreateOpt  
 (2) oadrCreatedOpt

192.168.1.101	192.168.1.2	POST /OpenADR2/Simple/2.0b/EiOpt	HTTP/1.1
192.168.1.2	192.168.1.101	HTTP/1.1	200 OK

```

oadrCreateOpt XML{
    "optID": String,
    "optType" : String,
    "optReason" : String,
    "venID" : String,
    "marketContext" : String,
    "available" : Array,
    "createdDateTime" : Date,
    "requestID" : String
}
    ↳ <oadrCreateOpt
        ↳ d3p1:schemaVersion="2.0b"
            ↳ xmlns:d3p1="http://docs.oasis-open.org/ns/energyinterop/201110"
                ↳ <d3p1:optID>
                ↳ <d3p1:optType>
                ↳ <d3p1:optReason>
                ↳ <d3p1:venID>
                ↳ <availability>
                ↳ <d3p1:createdDateTime>
                ↳ <requestID>
                    ↳ </requestID>
                ↳ </oadrCreateOpt>
  
```

```

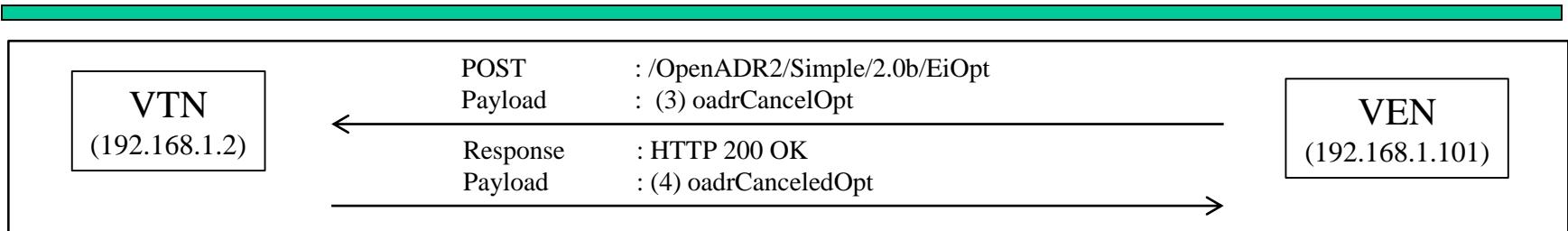
available Array{
    "dtstart" : Date,
    "duration" : String,
}
  
```

```

oadrCreatedOpt XML{
    "responseCode" : Integer,
    "responDescription" : String,
    "requestID" : String,
    "optID" : String,
}
    ↳ <ns3:oadrCreatedOpt
        ↳ ns5:schemaVersion="2.0b"
            ↳ <ns5:eiResponse>
                ↳ <ns5:responseCode>
                ↳ <ns5:responseDescription>
                ↳ <ns4:requestID>
                    ↳ </ns5:eiResponse>
            ↳ <ns5:optID>
                ↳ </ns3:oadrCreatedOpt>
  
```

## 2. Profile : OpenADR 2.0b

### 2.4 Services : EiOpt (HTTP / XML)



(3) oadrCancelOpt

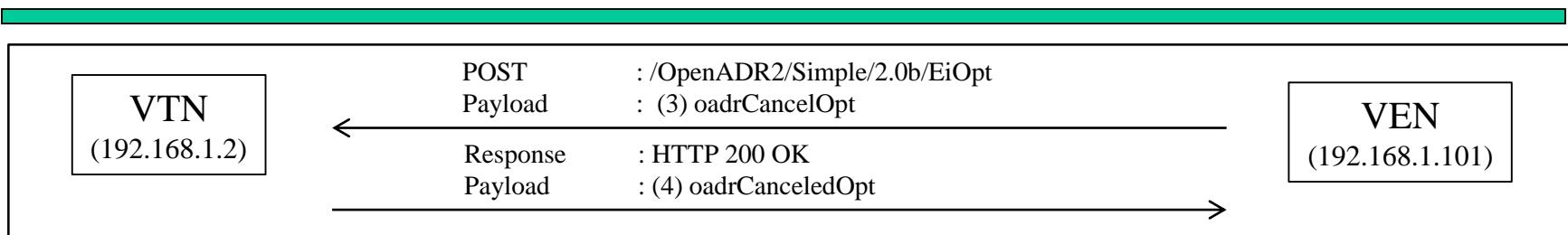
Key name	Comments
requestID	request identifier
optID	opt identifier
venID	requested VEN ID
service	message type

(4) oadrCanceledOpt

Key name	Comments
requestID	request identifier
responseCode	response code
responseDescription	description of response code
optID	opt identifier
service	message type

# 2. Profile : OpenADR 2.0b

## 2.4 Services : EiOpt (HTTP / XML)



(3) oadrCancelOpt  
 (4) oadrCanceledOpt

192.168.1.101	192.168.1.2	POST /OpenADR2/Simple/2.0b/EiOpt	HTTP/1.1
192.168.1.2	192.168.1.101	HTTP/1.1	200 OK

### oadrCancelOpt XML{

```

    "requestID" : String,
    "optID" : String,
    "venID" : String
}
  <oadrCancelOpt
    d3p1:schemaVersion="2.0b"
    xmlns:d3p1="http://docs.oasis-open.org/ns/energyinterop/201110">
    <requestID
      xmlns="http://docs.oasis-open.org/ns/energyinterop/201110/payloads">
        9c1c8dfc42
      </requestID>
    <d3p1:optID>
      109362332c-IN
    </d3p1:optID>
    <d3p1:venID>
      0ef0dd4d4e3eb592e74e
    </d3p1:venID>
  </oadrCancelOpt>

```

### oadrCanceledOpt XML{

```

    "responseCode" : Integer,
    "responDescription" : String,
    "requestID" : String,
    "optID" : String
}
  <ns3:oadrCanceledOpt
    ns5:schemaVersion="2.0b">
    <ns5:eiResponse>
      <ns5:responseCode>
        200
      </ns5:responseCode>
      <ns5:responseDescription>
        OK
      </ns5:responseDescription>
      <ns4:requestID>
        9c1c8dfc42
      </ns4:requestID>
    </ns5:eiResponse>
    <ns5:optID>
      109362332c-IN
    </ns5:optID>
  </ns3:oadrCanceledOpt>

```

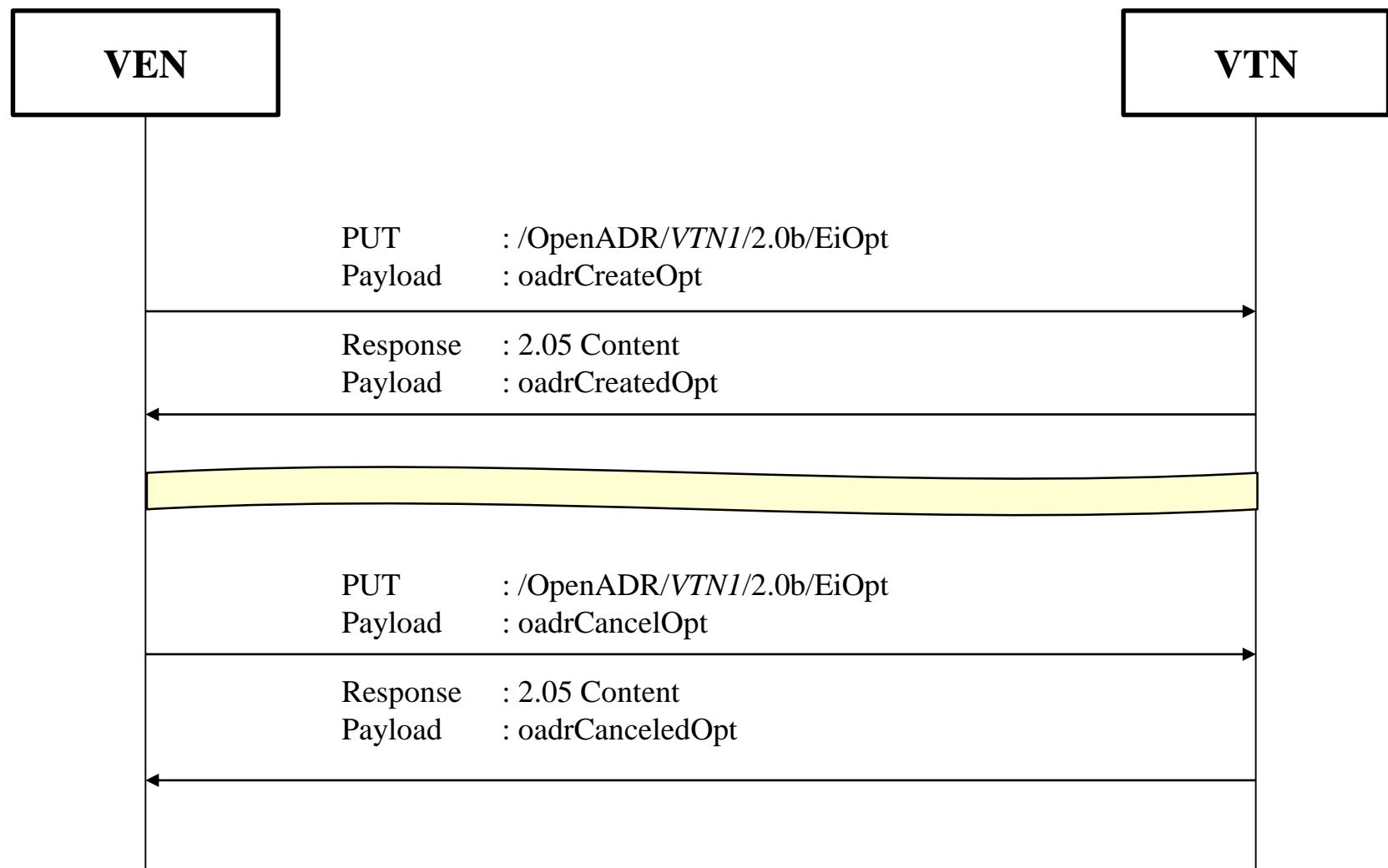
# **OpenADR 2.0b**

## **(4) EiOpt**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

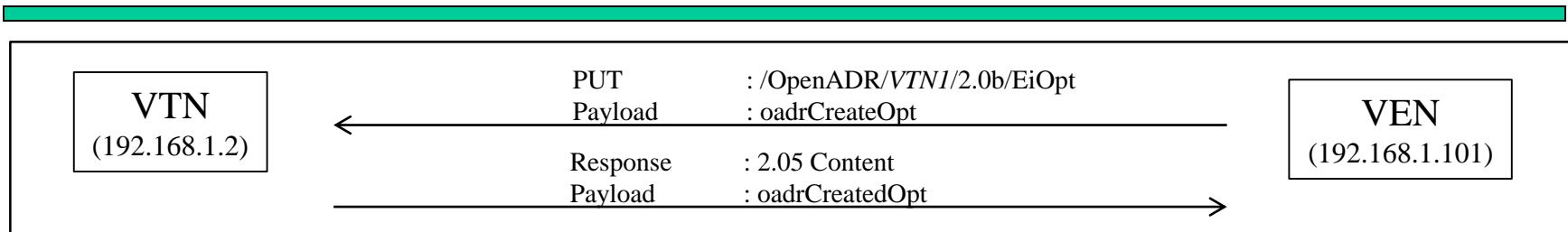
# 2. Profile : OpenADR 2.0b

## 2.3 Services : EiEvent (**CoAP / JSON**)



## 2. Profile : OpenADR 2.0b

### 2.4 Services : EiOpt (CoAP / JSON)



- (1) CreateOpt
- (2) CreatedOpt

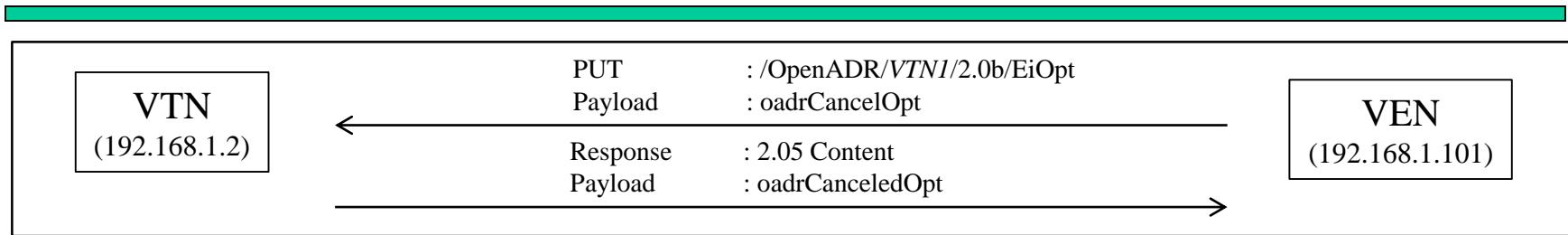
```
oadrCreateOpt JSON{
    "optID": String,
    "optType" : String,
    "optReason" : String,
    "venID" : String,
    "marketContext" : String,
    "available" : Array,
    "createdDateTime" : Date,
    "requestID" : String,
    "service" : String
}
```

```
available Array{
    "dtstart" : Date,
    "duration" : String,
}
```

```
oadrCreatedOpt JSON{
    "responseCode" : Integer,
    "responDescription" : String,
    "requestID" : String,
    "optID" : String,
    "service" : String
}
```

## 2. Profile : OpenADR 2.0b

### 2.4 Services : EiOpt (CoAP / JSON)



- (3) CancelOpt  
(4) CanceledOpt

```
oadrCancelOpt JSON{
    "requestID" : String,
    "optID" : String,
    "venID" : String,
    "service" : String
}
```

```
CanceledOpt JSON{
    "responseCode" : Integer,
    "responDescription" : String,
    "requestID" : String,
    "optID" : String,
    "service" : String
}
```

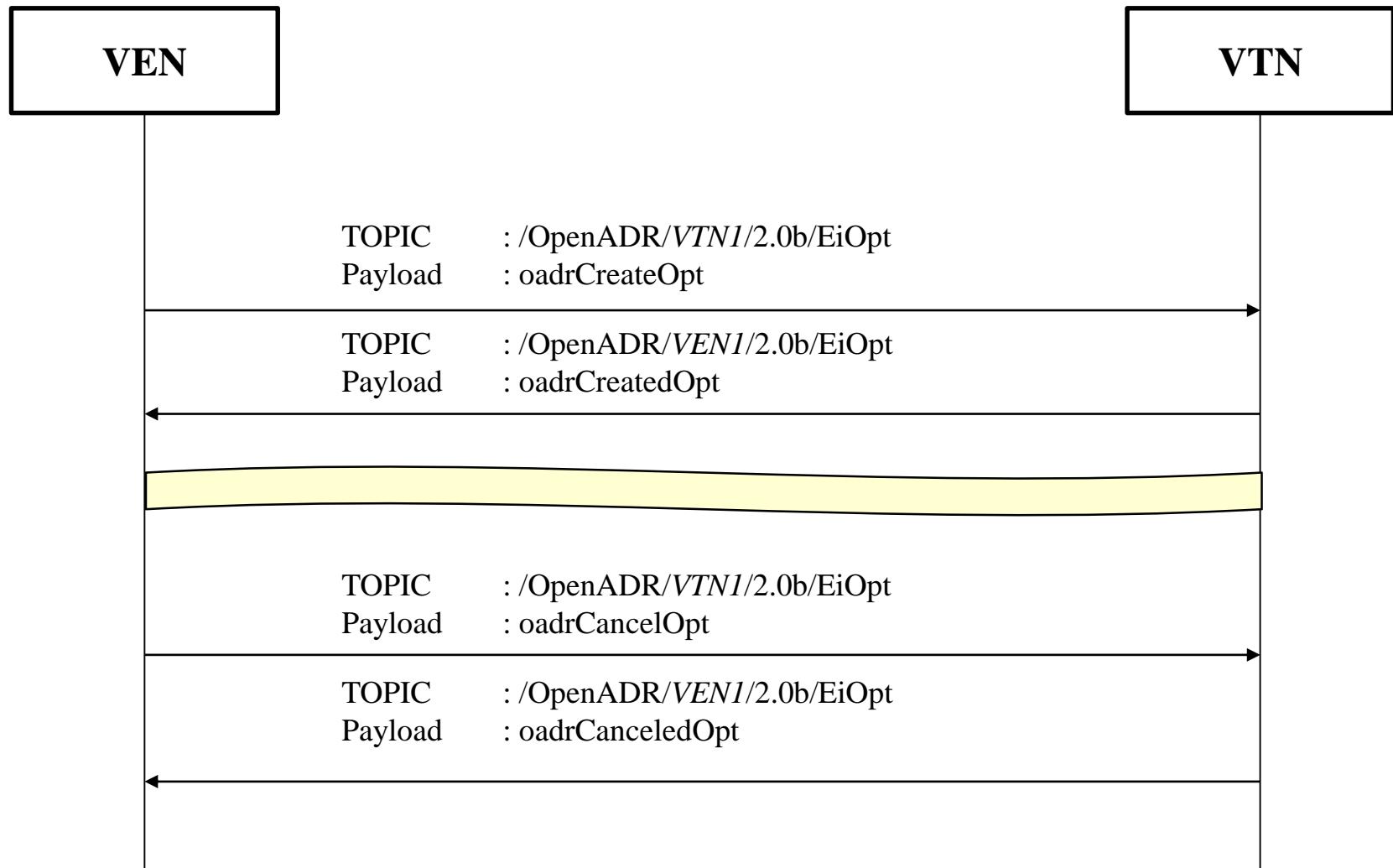
# **OpenADR 2.0b**

## **(4) EiOpt**

- **HTTP / XML**
- **CoAP / JSON**
- **MQTT / JSON**

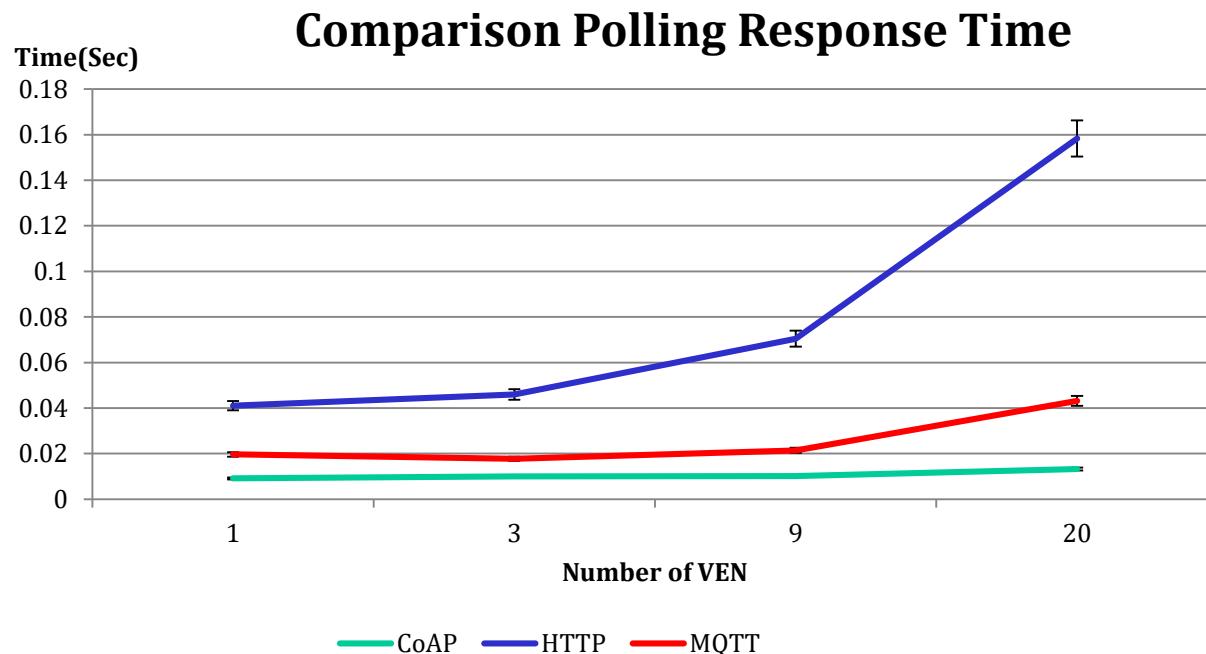
## 2. Profile : OpenADR 2.0b

### 2.3 Services : EiEvent ([MQTT / JSON](#))



## 2. Smart Home Energy Framework :

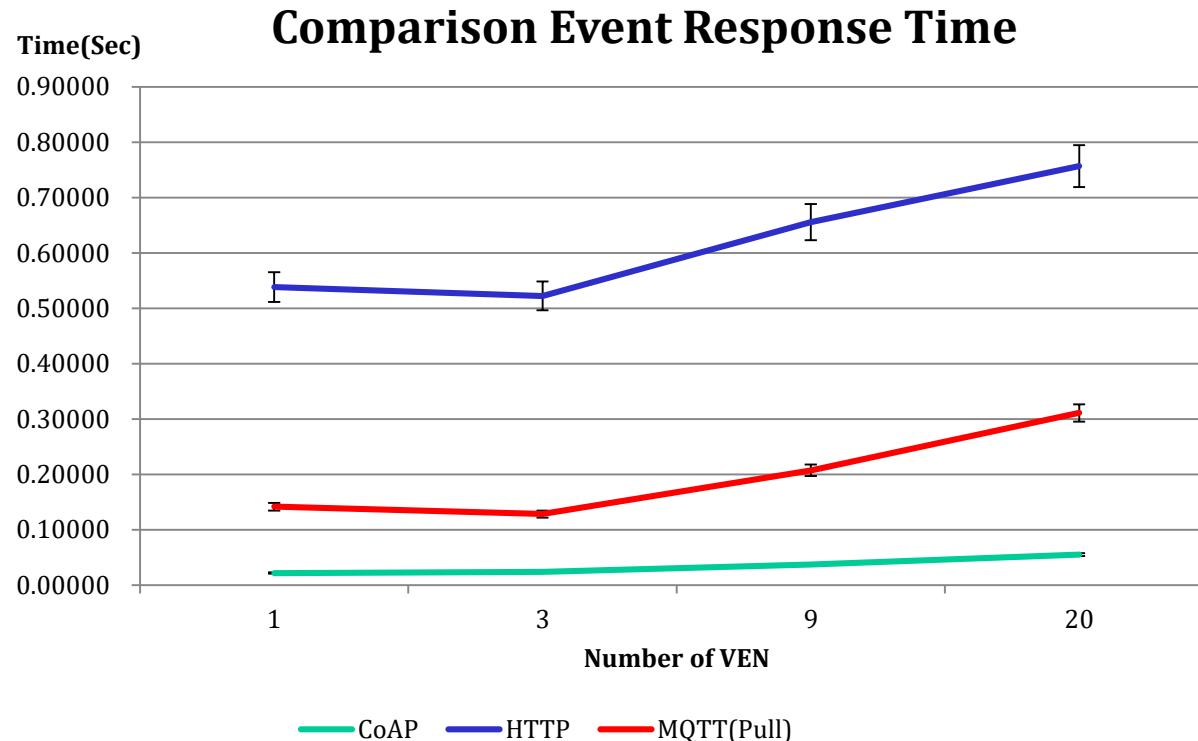
### 2.1 OpenADR(HTTP/XML, MQTT/JSON, CoAP/JSON) : Data Time Comparison (1/3)



Polling Respond Comparison (Sec)				
	1	3	9	20
HTTP	0.0410224	0.045946	0.070413	0.158386
CoAP	0.009088	0.00992	0.010096	0.013202
MQTT	0.0196723	0.017697	0.021437	0.043106

## 2. Smart Home Energy Framework :

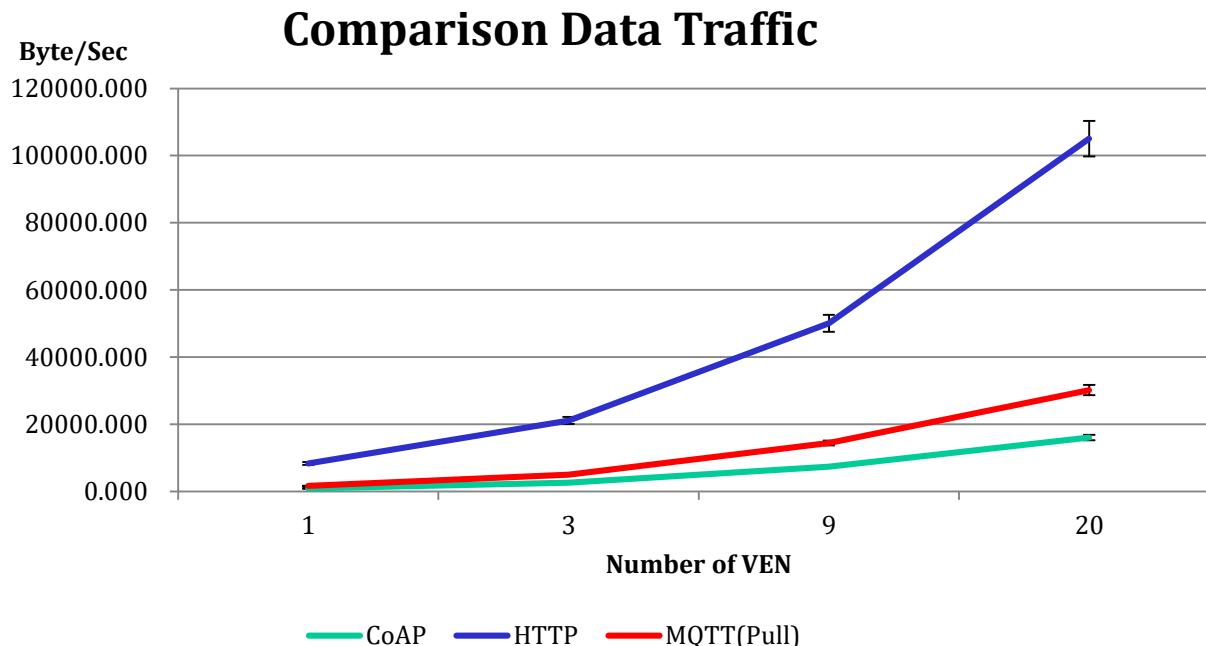
### 2.1 OpenADR(HTTP/XML, MQTT/JSON, CoAP/JSON) : Data Time Comparison (2/3)



Event-Response (Sec)				
	1	3	9	20
HTTP	0.538178	0.522328	0.655566	0.75686
CoAP	0.02196	0.02394	0.03703	0.05531
MQTT(Pull)	0.1416421	0.128474	0.207446	0.310962
MQTT(Push)	0.0271353	0.029069	0.025261	0.027319

## 2. Smart Home Energy Framework :

### 2.1 OpenADR(HTTP/XML, MQTT/JSON, CoAP/JSON) : Data Traffic Comparison (3/3)



Data traffic (Bytes)				
	1	3	9	20
HTTP	8340.1966	21130.51	50062.6	105051.3
CoAP	877.392	2592.325	7368.251	16052.44
MQTT(Pull)	1676.794	5001.758	14395.77	30195.19
MQTT(Push)	397.9302	897.7762	3350.054	6077.086

# Energy Framework

## 2. Profile & Protocol : OpenADR, EMAP, **DLMS/COSEM**

2.1 OpenADR (HTTP/XML, MQTT, COAP)

*VTN ↔ VEN*

2.2 EMAP (MQTT, COAP/JSON)

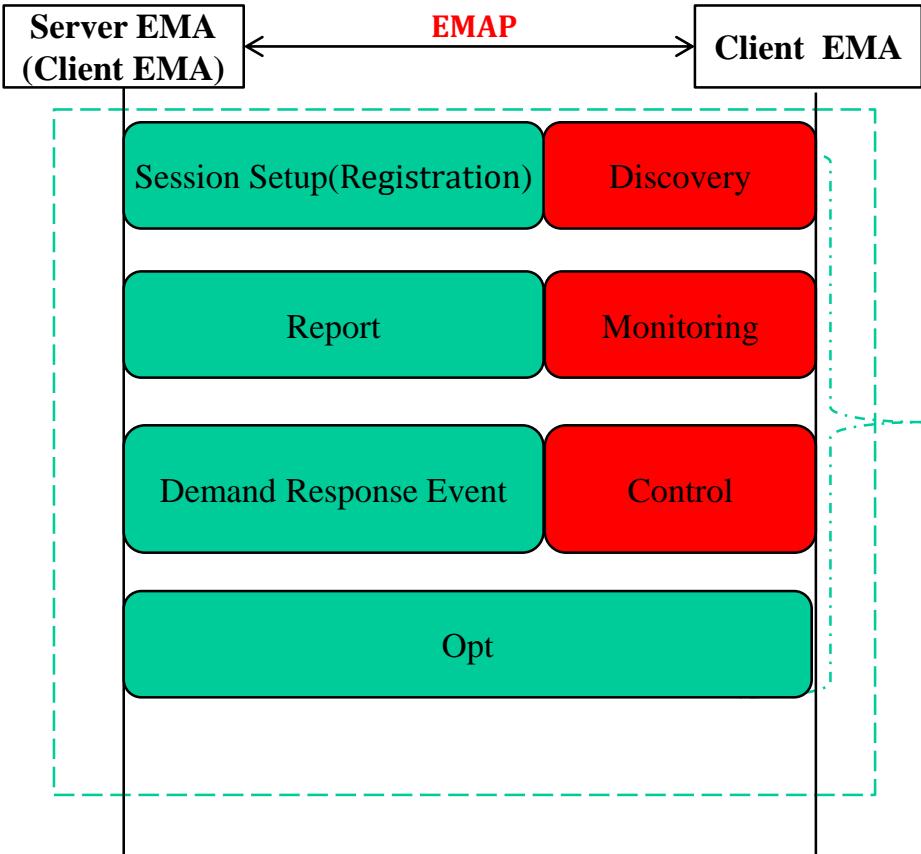
*Server EMA ↔ Client EMA*

2.3 Device Control

*Client EMA ↔ Device*

2.4 DLMS/COSEM

## 2.2 EMAP(MQTT, CoAP/JSON)



- Session Setup(Registration)
  - 에너지관리에이전트 간 서로 연결을 수립하고 Report을 교환할 때 기존의 Report에 대한 에너지 가격 정보나 클라이언트 에너지관리에이전트의 에너지 관리 정보를 얻음.
  - **기존 OpenADR 2.0b의 EiRegister, EiReport, EiEvent의 초기 등록과정을 포함하여 Session Setup과정으로 통일하였음.**
- Report(UpdateReport)
  - 에너지관리에이전트 간 서로 연결을 수립 할 때 Report을 교환할 때 실시간 에너지에 대한 가격 정보나 클라이언트 에너지관리에이전트의 디바이스 정보 등을 얻음.
  - **하위의 EMA의 정보를 모니터링 하는 단계, Explicit/Abstract 방식으로 Monitoring.**
  - Explicit에는 Device Type (LED, ESS, Recloser, Resource, PV)에 대한 정보가 포함
  - Implicit에는 Device Type을 제외한 정보가 포함되어 있어 필요로 하는 정보에 따라 데이터 트래픽을 다르게 설정할 수 있음.
- Event(Demand Response Event)
  - 에너지관리에이전트 프로토콜은 다양한 이벤트 종류가 있다. 예로 가격기반 수요반응 이벤트에서 Initial Price, Incentive Price, Negotiation Price가 있다. 서버 에너지관리에이전트는 실시간적으로 가격에 대한 정보를 이벤트로 발생
  - **EMA가 최하위 Device 컨트롤하기 위해서 Device를 총괄적으로 관리하는 Smart Appliances에게 Control 메시지를 전달.**
  - 대표적인 예로 기존 OpenADR 2.0b의 marketContext에 price정보만 있지만, 하위 노드의 Price가 포함되어 있는 Local Price, Local Incentive가 포함되어 있음.
- Opt
  - 에너지관리에이전트 프로토콜의 Opt는 클라이언트 에너지관리에이전트가 상위 서버 에너지관리에이전트에게 수요반응 이벤트의 가용상태 또는 수요반응 이벤트 프로그램 변경, 수요반응 스케줄링을 요청을 알려주는 서비스.

## 2.2 EMAP(MQTT, CoAP/JSON) Profile Spec (1/2)

---

- EMAP 모델에서 필요한 서비스 4가지
  - Session Setup, UpdateReport, Event, Opt
  - **Session Setup**
    - ConnectRegistration/ConnectedRegistration : 상위노드가 등록전에 등록에 필요한 정보를 하위노드로부터 얻는다. 응답메시지에는 상위노드에 의해 지원되는 모든 프로파일과 전송 프로토콜 정보가 포함된다.
    - CreatePartyRegistration/CreatedPartyRegistration : 하위노드가 상위노드의 정보 중에 등록할 내용을 입력하여 등록을 요청하고 가 받아들인다.
    - RegisterReport/RegisteredReport : 등록된 자원에 대한 상태와 최대값 최소값 등에 대하여 보고하고 응답한다.
    - Poll(RegisterReport Request) / RegisterReport : 주기적으로 Report를 보내고 해당 Report를 받아들임
    - RequestEvent/DistributeEvent : 하위노드가 DR이벤트가 있는지 요청, 하고 상위노드가 DR이벤트를 보냄
    - CancelPartyRegistration/CanceledPartyRegistration : 하위노드 측에서 등록을 취소하고 상위노드 측에서 받아들인다.

## 2.2 EMAP(MQTT, CoAP/JSON) Profile Spec (2/2)

- EMAP 모델에서 필요한 서비스 4가지

- **UpdateReport**

- UpdateReport/UpdatedReport: 주기적으로 등록된 자원에 대한 상태와 최대값 최소값 등에 대하여 보고하고 응답한다. (Type에 따라 Implicit/Explicit로 구분)

- **Event**

- **PULL**

- Poll/DistributeEvent: 주기적으로 하위노드에서 Event가 있는지 요청하고 상위노드 쪽에서 이에 대한 Event를 내린다.
    - CreateEvent/Response: 하위노드에서 Event에 참여할 것인지에 대하여 OptIn/OptOut으로 응답하고 상위노드에서 확인하고 응답한다.

- **Push**

- DistributeEvent/Response: 상위노드가 이벤트에 대하여 바로 하위노드에게 요청하고 하위노드는 이에 대하여 응답한다.
    - CreateEvent/Response: 하위노드에서 Event에 참여할 것인지에 대하여 OptIn/OptOut으로 응답하고 상위노드에서 확인하고 응답한다.

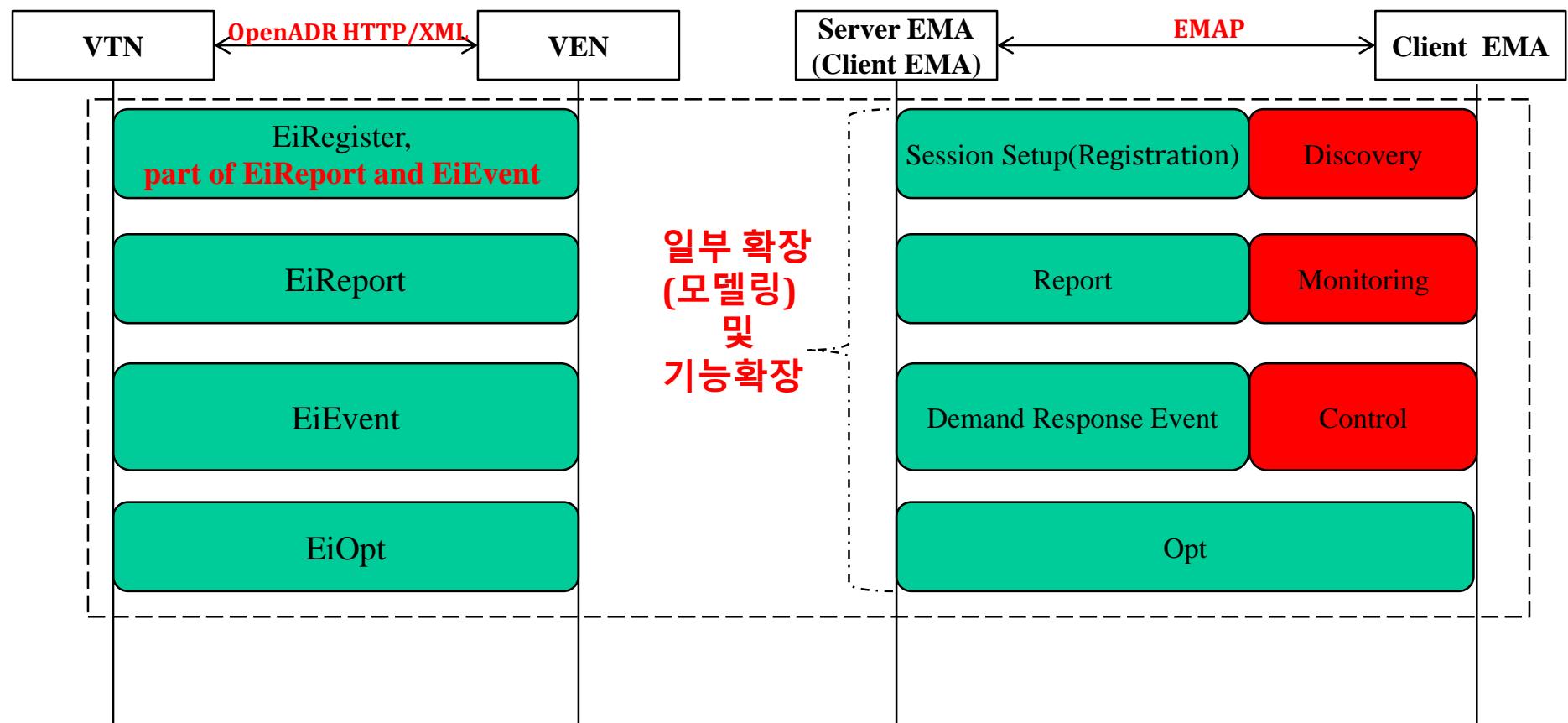
- **Opt**

- CreateOpt/CreatedOpt: 하위노드가 opt 스케줄을 보내고 상위노드가 응답한다.
    - CancelOpt/CanceledOpt: 하위노드가 opt 스케줄을 취소하고 S-EMA가 응답한다.

## 2.2 EMAP(MQTT, CoAP/JSON)

### EMAP

- EMA 사이의 수요반응 통신 프로토콜이며 OpenADR 2.0b의 모델링을 따랐으며 일부 모델링을 확장했고 Discovery와 Monitoring, Control 부분의 통신 프로토콜이 확장.



# **EMAP (CoAP & MQTT/JSON)**

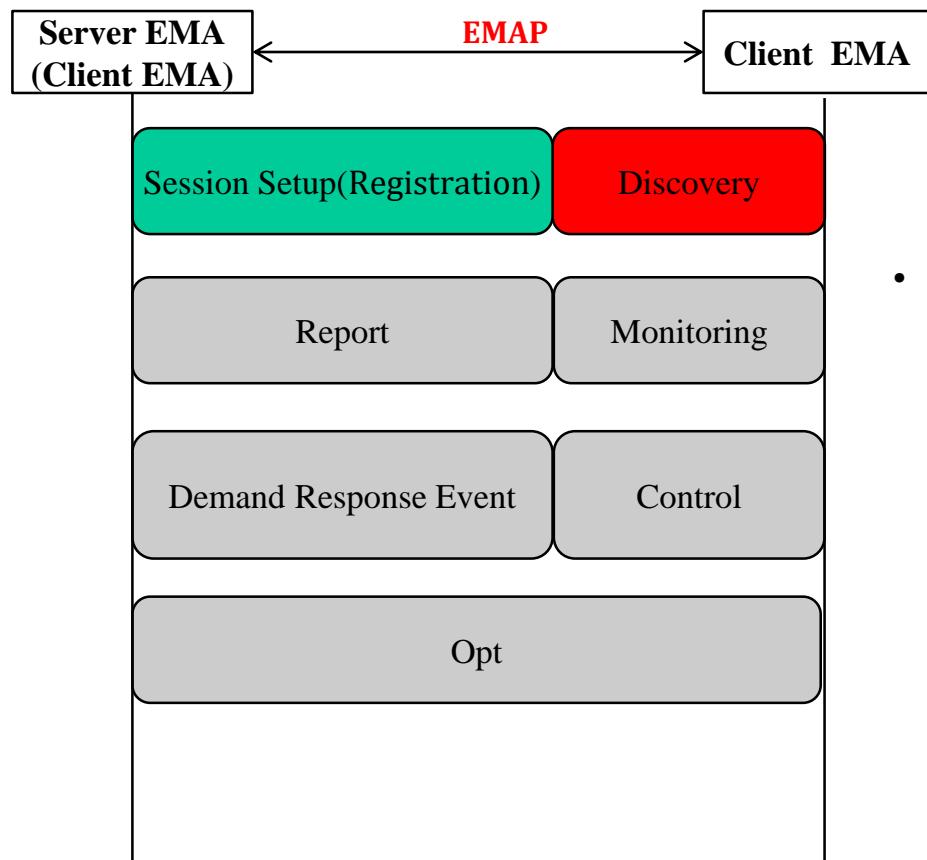
## **Session Setup (Registration)**

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup

#### EMAP

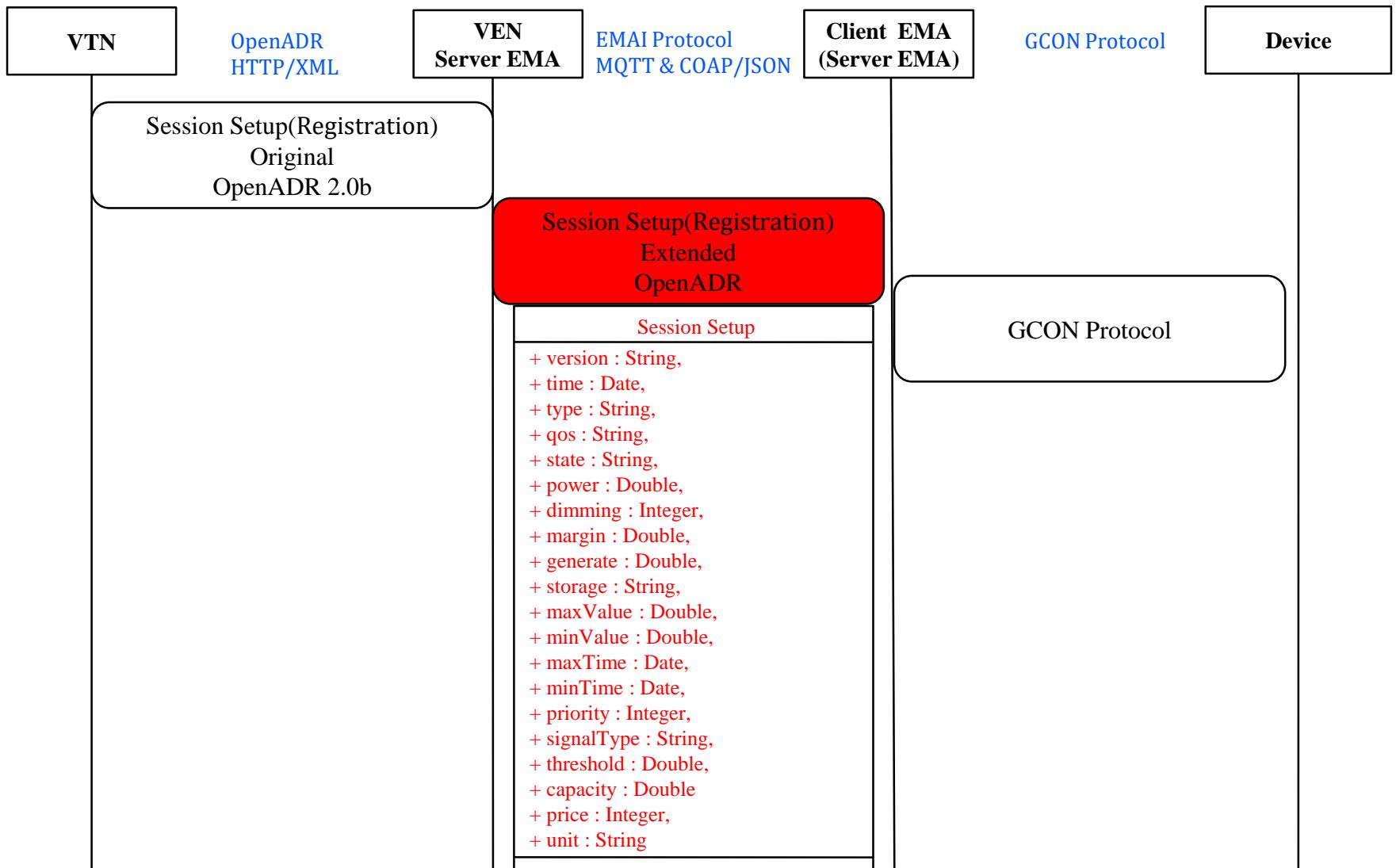
- EMA 사이의 수요반응 통신 프로토콜이며 OpenADR 2.0b의 모델링을 따랐으며 일부 모델링을 확장했고 Discovery와 Monitoring, Control 부분의 통신 프로토콜이 확장.



- Session Setup(Registration)
  - 에너지관리에이전트 간 서로 연결을 수립하고 Report을 교환할 때 기존의 Report에 대한 에너지 가격 정보나 클라이언트 에너지관리에이전트의 에너지 관리 정보를 얻음.
  - **기존 OpenADR 2.0b의 EiRegister, EiReport, EiEvent의 초기 등록과정을 포함하여 Session Setup과정으로 통일하였다.**

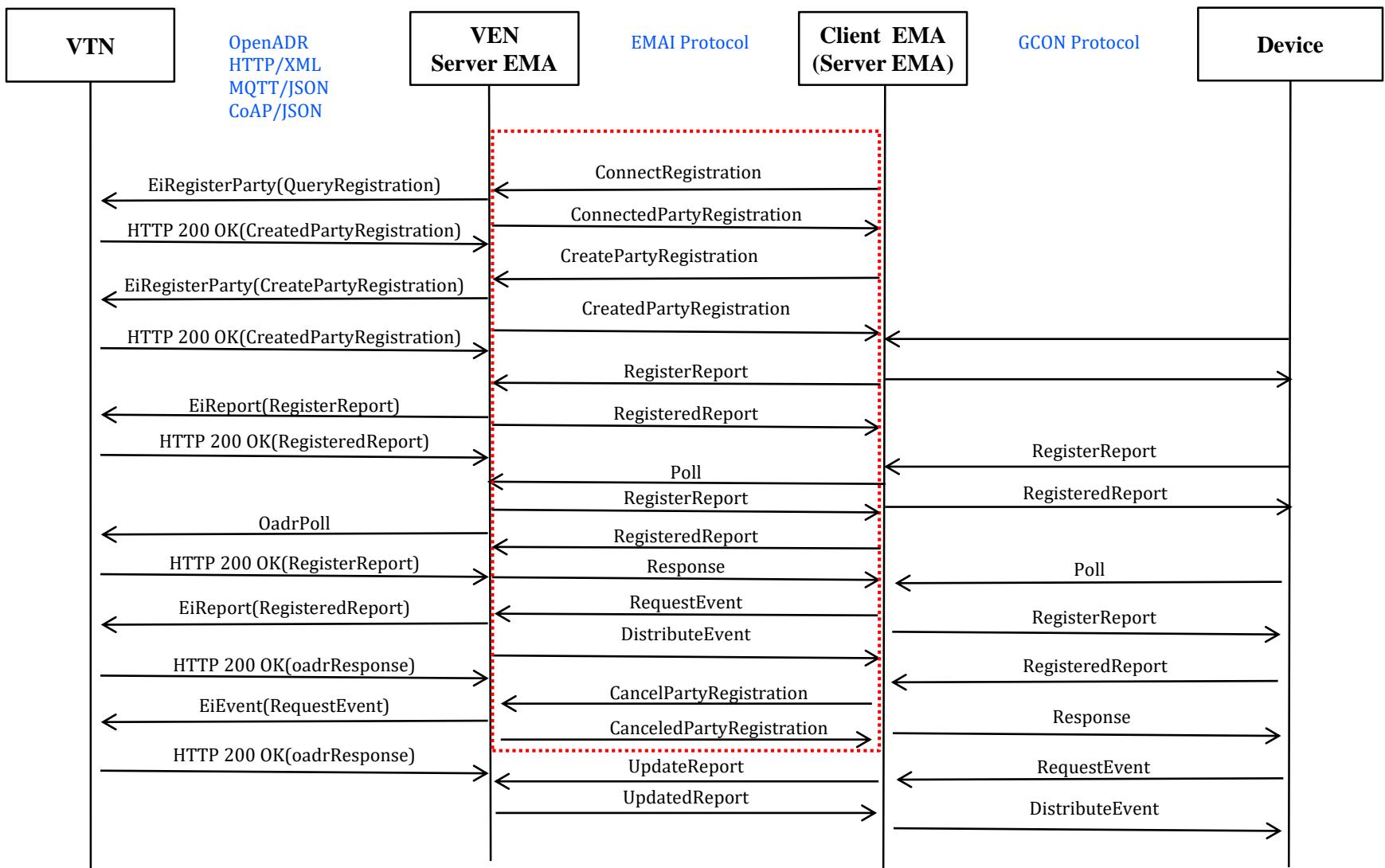
## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



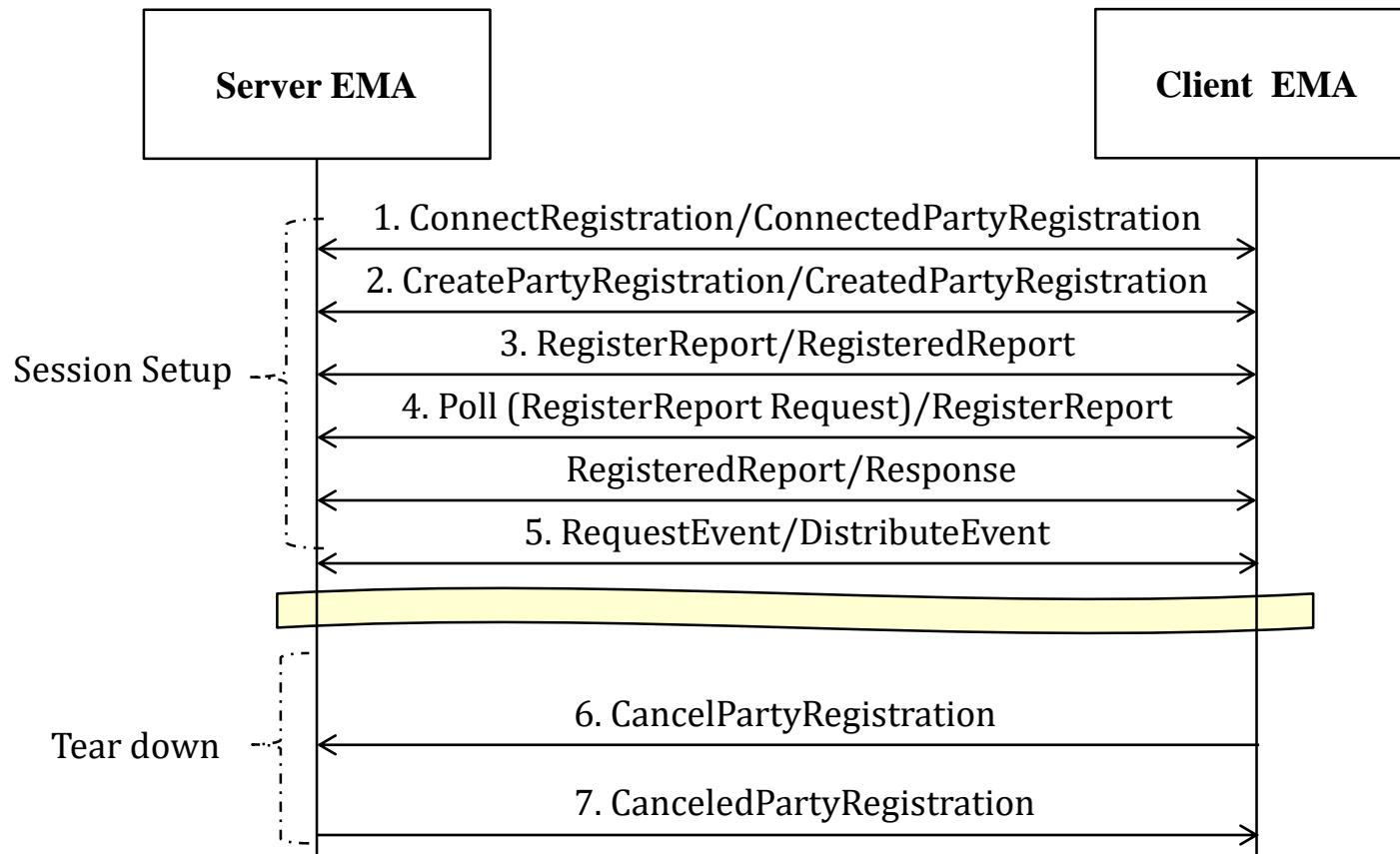
# 2.2 EMAP(MQTT, CoAP/JSON)

## Service : Session Setup



## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup

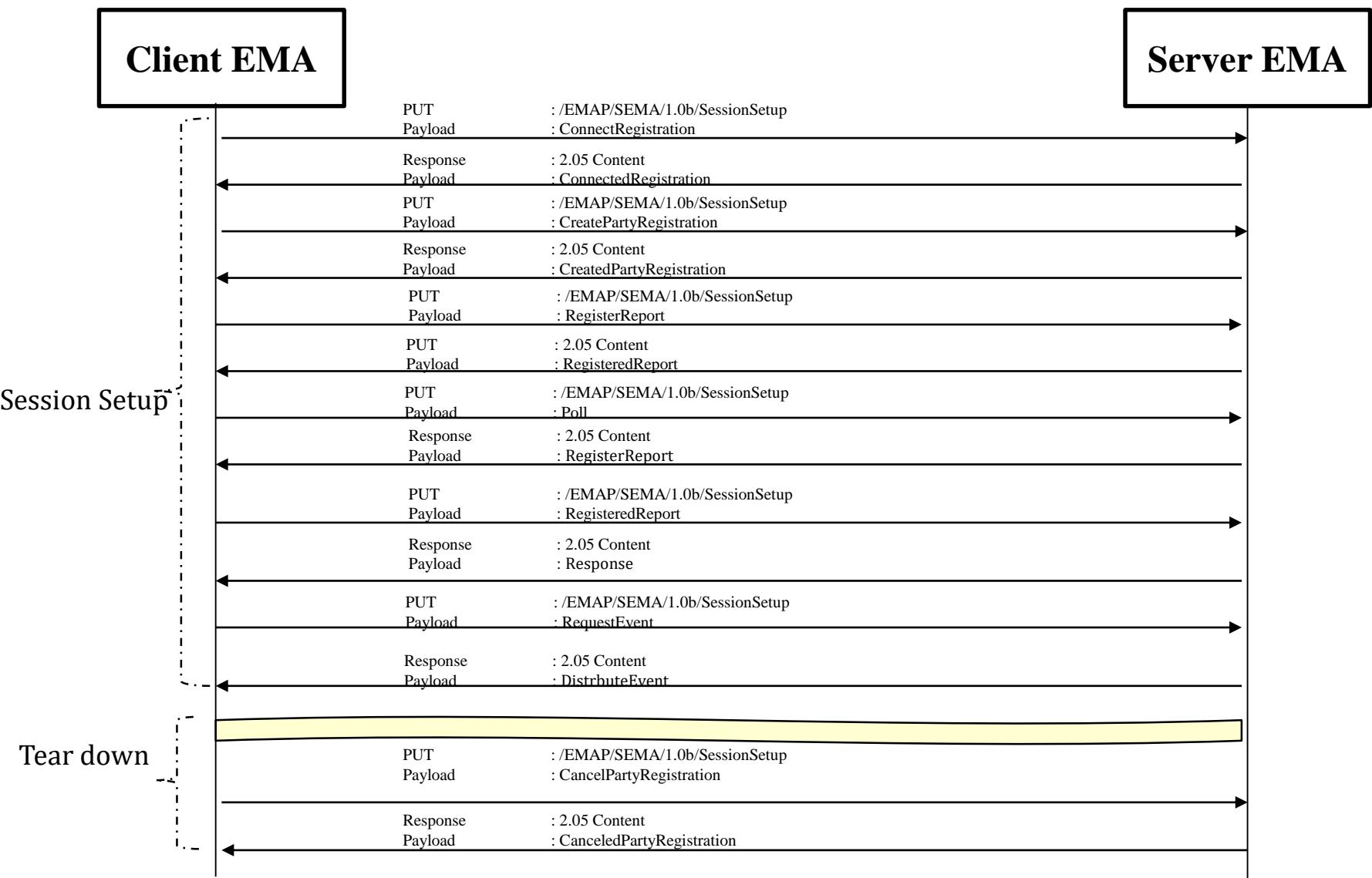


# EMAP

## (1) Session Setup (Registration)

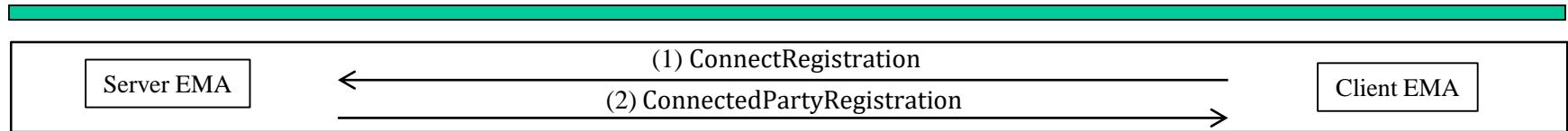
- CoAP / JSON
- MQTT / JSON

## 2.2 EMAP(CoAP/JSON) Service : Session Setup



## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



(1) ConnectRegistration

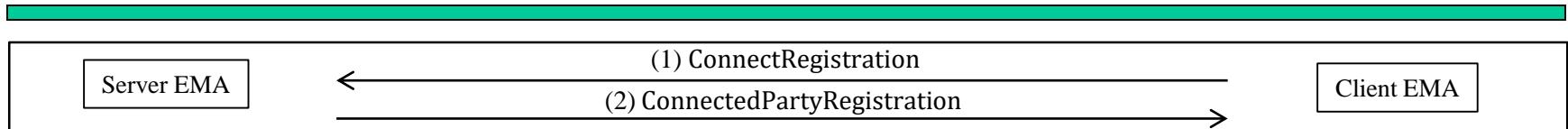
Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei: venID	
DestEMA	ei: vtnID	
requestID	pyld:requestID	
service	(Tag 이름으로 존재)	
version		IdentifiedObject:version
time		RandomizableEvent:creation Time

(2) ConnectedRegistration

Key Name	Reference		
	OpenADR 2.0b	SEP 2.0(IEC 61968)	
SrcEMA		ei:vtnID	
DestEMA		ei:venID	
responseCode			ei:responseCode
responseDescription		ei:eiResponse	ei:responseDescription
requestID			Pyld:requestID
profile	transports	Oadr:oadrProfile	oadrTransport:Array
	profileName	Oadr:oadrProfile	oadrProfileName:String
profile:profileName:transports	transportName	Oadr:oadrProfile:oadrTransportname	oadrTransportName:String
duration			RandomizableEvent:randomizeDuration
registrationID			
service		(Tag 이름으로 존재)	
version			IdentifiedObject:version
time			RandomizableEvent:creationTime

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



(1) ConnectRegistration

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
service	type of service
version	EMAP protocol version
time	service creation time

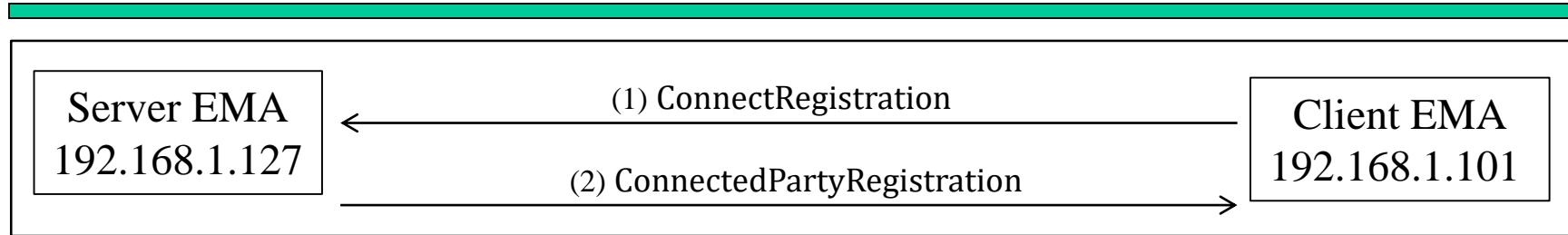
(2) ConnectedRegistration

Key name	Comments					
SrcEMA	source EMA identifier					
DestEMA	destination EMA identifier					
responseCode	response code					
responseDescription	description of response code					
requestID	request identifier					
duration	requested polling frequency					
registrationID	registration identifier					
profile	<table border="1"> <tr> <td>profileName</td> <td>type of profile</td> </tr> <tr> <td>transports</td> <td>transportName</td> <td>type of transport protocol</td> </tr> </table>	profileName	type of profile	transports	transportName	type of transport protocol
profileName	type of profile					
transports	transportName	type of transport protocol				
version	EMAP protocol version					
service	type of service					
time	service creation time					

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



- (1) ConnectRegistration  
(2) ConnectedPartyRegistration

192.168.1.101	192.168.1.127	CoAP	CON, MID:44318, PUT, /ConnectRegistration (application/json)
192.168.1.127	192.168.1.101	CoAP	ACK, MID:44318, 2.05 Content (application/json)

ConnectRegistration Object{  
    “SrcEMA” : String,  
    “DestEMA” : String,  
    “requestID”: String,  
    “service”: String,  
    “version” : String,  
    “time” : Date,  
    “type”: String => RegisteredReport로 변경  
    “QoS” : String => RegisterReport로 변경  
}

▪ JavaScript Object Notation: application/json  
    ▪ Object  
        ▷ Member Key: SrcEMA  
        ▷ Member Key: DestEMA  
        ▷ Member Key: requestID  
        ▷ Member Key: version  
        ▷ Member Key: customerPriority  
        ▷ Member Key: QoS  
        ▷ Member Key: type  
        ▷ Member Key: service  
        ▷ Member Key: time

ConnectedRegistration Object{  
    “SrcEMA” : String,  
    “DestEMA” : String,  
    “responseCode” : Integer,  
    “responseDescription”: String,  
    “requestID”: String,  
    “duration” : Integer,  
    “profile”: Array,  
    “profileName”: String,  
    “TransportName”: String,  
    “service”: String,  
    “version” : String,  
    “time” : Date,  
    “type”: String => RegisteredReport로 변경  
    “QoS” : String => RegisterReport로 변경,  
    “registrationID” : String  
}

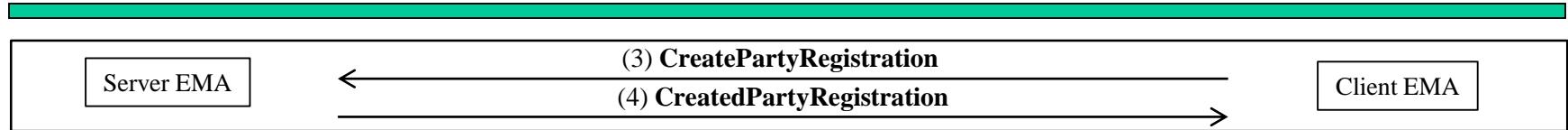
oadrProfile Array{  
    “oadrTransports”: Array,  
    “oadrProfileName”: String  
}

oadrTransports Array{  
    “oadrTransportName” : String,  
}

▪ JavaScript Object Notation: application/json  
    ▪ Object  
        ▷ Member Key: profileName  
        ▷ Member Key: transportName  
        ▷ Member Key: SrcEMA  
        ▷ Member Key: type  
        ▷ Member Key: version  
        ▷ Member Key: responseCode  
        ▷ Member Key: customerPriority  
        ▷ Member Key: duration  
        ▷ Member Key: QoS  
        ▷ Member Key: responseDescription  
        ▷ Member Key: requestID  
        ▷ Member Key: service  
        ▷ Member Key: registrationID  
        ▷ Member Key: time  
        ▷ Member Key: DestEMA

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



(3) CreatePartyRegistration

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:venID	
DestEMA	ei:vtnID	
requestID	pyld:requestID	
profileName	oadr:oadrProfileName	
transportName	oadr:oadrTransportName	
reportOnly	oadr:oadrReportOnly	
xmlSignature	oadr:oadrXMLSignature	
httpPullModel	oadr:oadrHttpPullModel	
service	(Tag 0 름으로 존재)	
time		RandomizableEvent:creationTime

(4) CreatedPartyRegistration

Key Name	Reference		
	OpenADR 2.0b	SEP 2.0(IEC 61968)	
SrcEMA		ei:vtnID	
DestEMA		ei:venID	
responseCode			Ei:responseCode
responseDescription		ei:eiResponse	ei:responseDescription
requestID			Pyld:requestID
registrationID			ei:registrationID
profile	transports	Oadr:oadrProfile	oadrTransport:Array
	profileName	Oadr:oadrProfile	oadrProfileName:String
profile:profileName:transports	transportName	Oadr:oadrProfile:oadrTransportName	oadrTransportName:String
duration		RandomizableEvent:randomizedDuration	
service		(Tag 0 름으로 존재)	
time			RandomizableEvent:creationTime

## 2.2 EMAP(MQTT, CoAP/JSON)

# Service : Session Setup



(3) CreatePartyRegistration

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
profileName	profile name used by client EMA
transportName	transport name used by client EMA
reportOnly	EMA type (report only or full functional)
xmlSignature	xml 사용여부 true/false
httpPullMode	communication mode used by EMA (pull or push)
service	message type
time	service creation time

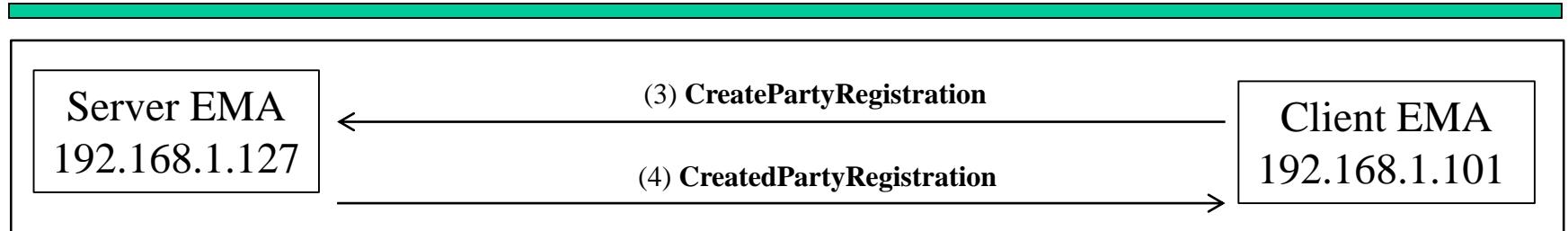
(4) CreatedPartyRegistration

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
responseCode	response code
responseDescription	description of response code
requestID	request identifier
duration	requested polling frequency
registrationID	registration identifier
profile	profileName
	transports transportName
version	EMAP protocol version
service	type of service
time	service creation time

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(3) CreatePartyRegistration  
(4) CreatedPartyRegistration

CoAP	192.168.1.101	192.168.1.127	CON, MID:29975, PUT, /CreatePartyRegistration (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:29975, 2.05 Content (application/json)

#### CreatePartyRegistration Object{

“SrcEMA” : String,  
“DestEMA” : String,  
“version” : Integer => 삭제,  
“requestID” : String,  
“transportName” : String,  
“reportOnly” : Integer,  
“httpPullModel” : Boolean,  
“profileName” : String,  
“xmlSignature” : String,  
“registrationID” : String 삭제  
“service” : String,  
“time” : Date  
}

JavaScript Object Notation: application/json  
↳ Object  
↳ Member Key: SrcEMA  
↳ Member Key: DestEMA  
↳ Member Key: requestID  
↳ Member Key: version  
↳ Member Key: transportName  
↳ Member Key: transportAddress  
↳ Member Key: reportOnly  
↳ Member Key: httpPullModel  
↳ Member Key: profileName  
↳ Member Key: xmlSignature  
↳ Member Key: registrationID  
↳ Member Key: service  
↳ Member Key: time

#### CreatedPartyRegistration Object{

“SrcEMA” : String,  
“DestEMA” : String,  
“version” : Integer => 삭제,  
“requestID” : String,  
“oadrProfile” : Array,  
“registrationID” : String,  
“duration” : String,  
“responseCode” : Integer,  
“responseDescription” : String,  
“service” : String  
“time” : Date,  
}

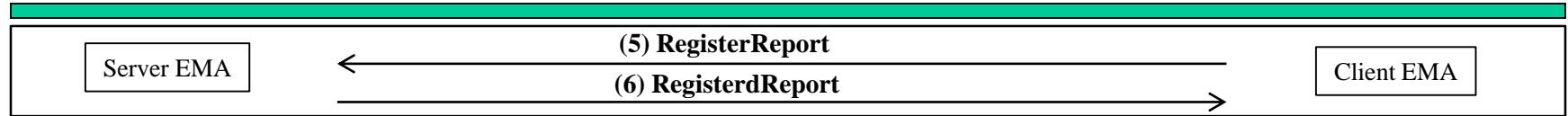
oadrProfile Array{  
“oadrTransports” : Array,  
“oadrProfileName” : String  
}

oadrTransports Array{  
“oadrTransportName” : String,  
}

JavaScript Object Notation: application/json  
↳ Object  
↳ Member Key: duration  
↳ Member Key: profileName  
↳ Member Key: transportName  
↳ Member Key: SrcEMA  
↳ Member Key: responseDescription  
↳ Member Key: requestID  
↳ Member Key: service  
↳ Member Key: registrationID  
↳ Member Key: time  
↳ Member Key: DestEMA  
↳ Member Key: version  
↳ Member Key: responseCode

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup

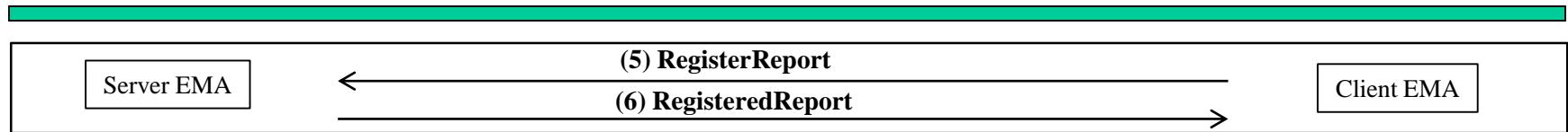


(5) RegisterReport

Key Name		Reference		
		OpenADR 2.0b	SEP 2.0(IEC 61968)	OpenFMB(IEC 61850)
SrcEMA		ei:venID		
DestEMA		ei:vtnID		
service		(tag이름으로 존재)		
time			RandomizableEvent:creation Time	
requestID		requestID		
<b>type(Implicit, Explicit)</b>			TariffProfile:serviceCategoryKind:ServiceKind	
report	duration	oadrReport	duration	
	reportDescription		oadrReportDescription	
	reportRequestID		reportRequestID	
	reportSpecifierID		reportSpecifierID	
	reportName		reportName	
	createdDateTime		createdDateTime	
	rID		rID	
	resourceID		resourceID	
	<b>deviceType</b>			EndDeviceControlType:type
	reportType		reportType	
report:reportDescription	itemUnits	oadrReport:oadrReportDescription	itemUnits	
	siScaleCode		siScaleCode	
	marketContext		marketContext	
	MinPeriod		oadrMinPeriod	
	MaxPeriod		oadrMaxPeriod	
	OnChange		oadrOnChange	
	itemDescription		itemDescription	
	<b>powerAttributes</b>		powerAttributes	
	qos			EndDeviceControl:loadShiftForward
	state			DeviceStatus:opState
report:reportDescription	power			Readings: value
	dimming			
	margin			Subscription:Level
	generate			IdentifiedObject:DemandResponseProgram:availabilityUpdatePowerChangeThreshold
				SolarEventProfile:SolarInverterStatus:value
				BatteryEventProfile:Batter

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup

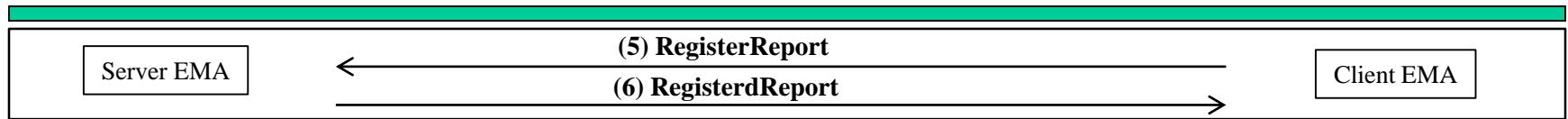


(6) RegisteredReport

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:venID	
DestEMA	ei:vtnID	
responseCode	ei:responseCode	
responseDescription	ei:responseDescription	
requestID	pyld:requestID	
service	(tag 이름으로 존재)	
time		RandomizableEvent:creation Time

# 2.2 EMAP(MQTT, CoAP/JSON)

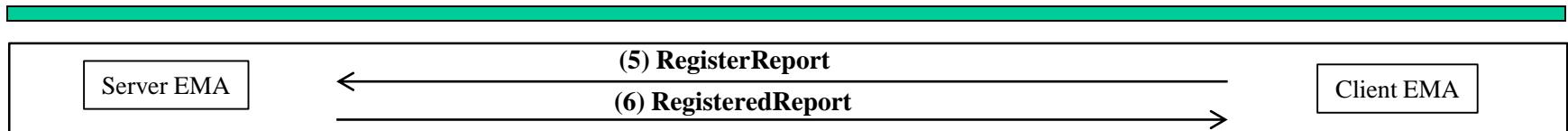
## Service : Session Setup



(5) RegisterReport

	Key name	Comments
SrcEMA		source EMA identifier
DestEMA		destination EMA identifier
requestID		request identifier
report	duration	report duration
	reportRequestID	report request identifier
	reportSpecifierID	report specific id (created from EMA)
	reportName	report name
	createdDateTime	created time of this report
	rID	
	resourceID	resource identifier
	reportType	report type
	deviceType	type of device
	itemUnits	unit of item that report
	siScaleCode	
	marketContext	refer marketContext address
	minPeriod	Energy usage minimum period
	maxPeriod	Energy usage maximum period
	onChange	
	itemDescription	type of item that report
	qos	device QoS
	state	device current state
	power	power usage
	dimming	dimming state
	margin	available amount of energy (Including generated, storaged Energy)
	generate	generated energy
	storage	storaged energy
	maxValue	energy max usage value
	minValue	energy min usage value
	avgValue	energy average usage value
	maxTime	energy max usage time
	minTime	energy min usage time
	priority	priority of this device
service	powerAttributes	hertz
		voltage
		ac
time		pulse frequency of power
type		voltage of power
		Is this AC power? (True or False)
		type of service
		service creation time
		report message type (implicit or explicit)

## 2.2 EMAP(MQTT, CoAP/JSON) Service : Session Setup



(6) RegisteredReport

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	message type
time	service creation time

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(5) RegisterReport	192.168.1.101	192.168.1.127	CoAP	CON, MID:36645, PUT, /RegisterReport (application/json)
(6) RegisteredReport	192.168.1.127	192.168.1.101	CoAP	ACK, MID:36645, 2.05 Content (application/json)

```
RegisterReport Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "reportType" : String,
    "EMAreisteredDRInformation" : Object -> 변경
    "EMAreisteredMgnInformation" : Object -> 변경,
    - "report" : Array,
    "time" : Date,
    "service" : String,
    "type" : String (Explicit, Implicit인지 구분)
}
```

JavaScript Object Notation: application/json

- Object
  - Member Key: SrcEMA
  - Member Key: DestEMA
  - Member Key: requestID
  - Member Key: reportName
  - Member Key: reportType
  - Member Key: EMAreisteredDRInformation
  - Member Key: EMAreisteredMgnInformation
  - Member Key: service
  - Member Key: time

```
report Object{
    "duration" : String,
    "reportRequestID" : Integer,
    "reportSpecifierID" : String,
    "reportName" : String,
    "createdDateTime" : Date,
    "reportDescription" : Array,
}
```

```
reportDescription Object{
    "rID" : String,
    "resourceID" : String,
    "deviceType" : String,
    "reportType" : String,
    "itemUnits" : String,
    "siScaleCode" : String,
    "marketContext" : String,
    "oadrMinPeriod" : String,
    "oadrMaxPeriod" : String,
    "oadrOnChange" : String,
    "itemDescription" : String,
    "powerAttributes" : Array,
    "qos" : String
    "state" : String,
    "power" : Double,
    "dimming" : Integer,
    "margin" : double,
    "generate" : double,
    "storage" : String,
    "maxValue" : Double,
    "minValue" : Double,
    "avgValue" : Double,
    "maxTime" : Date,
    "minTime" : Date,
    "priority" : Integer
}
```

```
powerAttributes Object{
    "hertz" : Double,
    "voltage" : Double,
    "ac" : Boolean
}
```

파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(5) RegisterReport	192.168.1.101	192.168.1.127	CoAP	CON, MID:36645, PUT, /RegisterReport (application/json)
(6) RegisteredReport	192.168.1.127	192.168.1.101	CoAP	ACK, MID:36645, 2.05 Content (application/json)

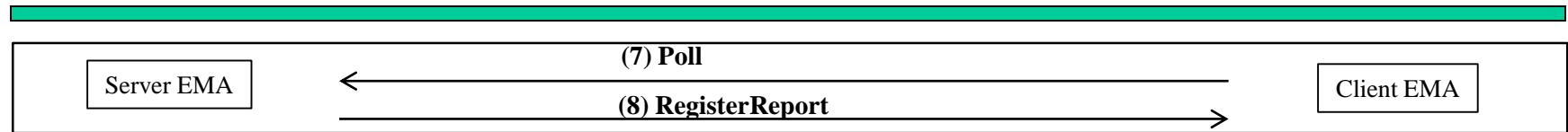
```
RegisteredReport Object{
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription": String,
    "transportName": String > 기존 OpenADR에 없는 내용,
    "version": Integer > 삭제,
    "threshold": Double > DistributeEvent로 이동,
    "service": String,
    "time": Date,
}
```

JavaScript Object Notation: application/json

- Object
  - Member Key: SrcEMA
  - Member Key: DestEMA
  - Member Key: transportName
  - Member Key: requestID
  - Member Key: responseCode
  - Member Key: version
  - Member Key: responseDescription
  - Member Key: threshold
  - Member Key: service
  - Member Key: time

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



(7) Poll

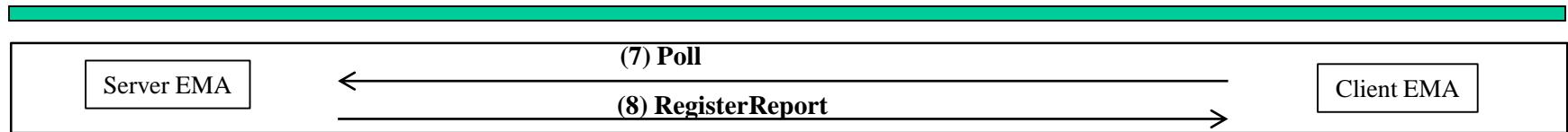
(8) RegisterReport

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:venID	
DestEMA	ei:vtnID	
service	(tag이름으로 존재)	
time		RandomizableEvent:creation Time

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
requestID	ei:eiResponse	Pyld:requestID
service	(tag이름으로 존재)	
time		RandomizableEvent:creation Time

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



(7) Poll

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
service	message type
time	service creation time

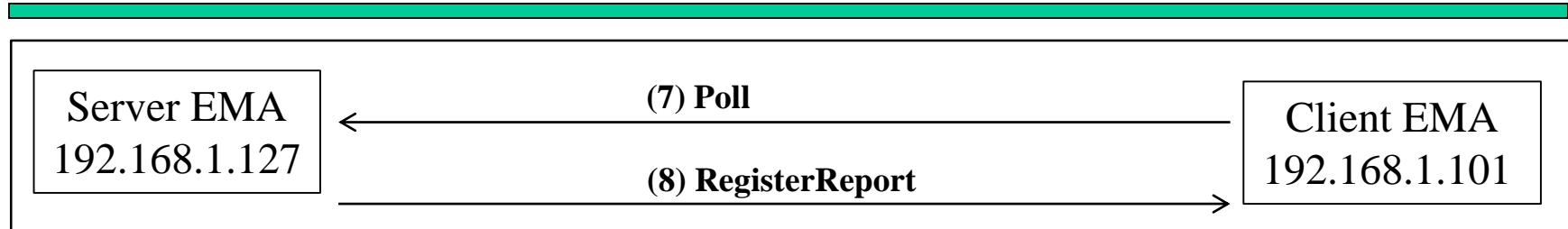
(8) RegisterReport

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
service	type of service
time	service creation time

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(7) Poll

CoAP	192.168.1.101	192.168.1.127	CON, MID:22170, PUT, /Poll (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:22170, 2.05 Content (application/json)

(8) RegisterReport

```
Poll JSON Object{
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": Integer, -> 삭제
    "version": Integer, -> 삭제,
    "type": String -> 삭제(RegisteredReport로 변경)
    "service": String,
    "time": Date
}
```

- JavaScript Object Notation: application/json
- Object
  - ▷ Member Key: SrcEMA
  - ▷ Member Key: DestEMA
  - ▷ Member Key: requestID
  - ▷ Member Key: version
  - ▷ Member Key: type
    - String value: Registration
    - Key: type
  - ▷ Member Key: service
  - ▷ Member Key: time

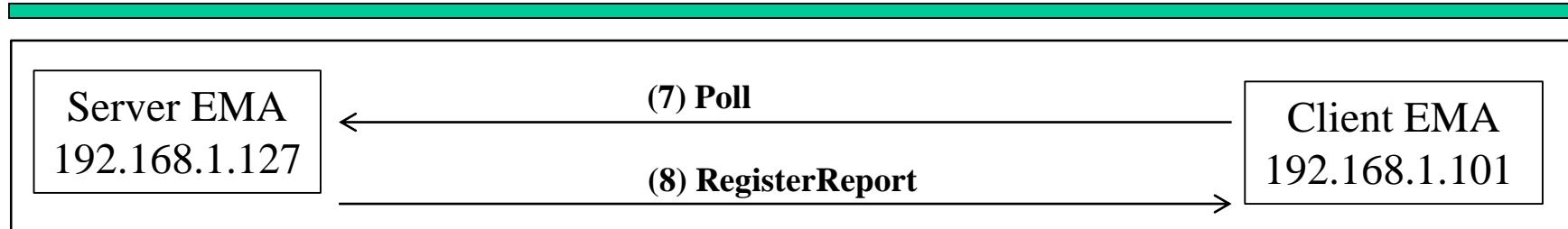
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(7) Poll

CoAP	192.168.1.101	192.168.1.127	CON, MID:22170, PUT, /Poll (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:22170, 2.05 Content (application/json)

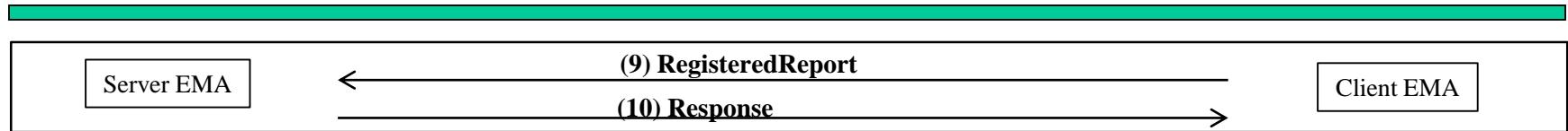
(8) RegisterReport

```
RegisterReport Object{
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "time": Date,
    "service": String
}
```

```
JavaScript Object Notation: application/json
Object
    Member Key: SrcEMA
    Member Key: DestEMA
    Member Key: requestID
    Member Key: reportName
    Member Key: reportType
    Member Key: EMAreisteredDRInformation
    Member Key: EMAreisteredMgnInformation
    Member Key: service
    Member Key: time
```

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



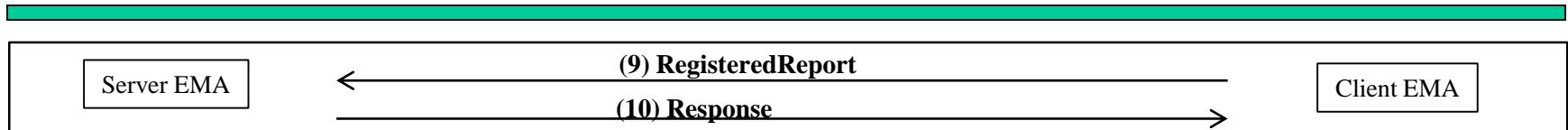
(9) RegisteredReport

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
responseCode	ei:eiResponse	ei:responseCode
responseDescription	ei:eiResponse	ei:responseDescription
requestID		Pyld:requestID
service	(tag 이름으로 존재)	
time		RandomizableEvent:creationTime

(10) Response

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA		ei:vtnID
DestEMA		ei:venID
responseCode		ei:responseCode
responseDescription	ei:eiResponse	ei:responseDescription
requestID		Pyld:requestID
service	(tag 이름으로 존재)	
time		RandomizableEvent:creationTime

## 2.2 EMAP(MQTT, CoAP/JSON) Service : Session Setup



(9) RegisteredReport

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	message type
time	service creation time

(10) Response

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	type of service
time	service creation time

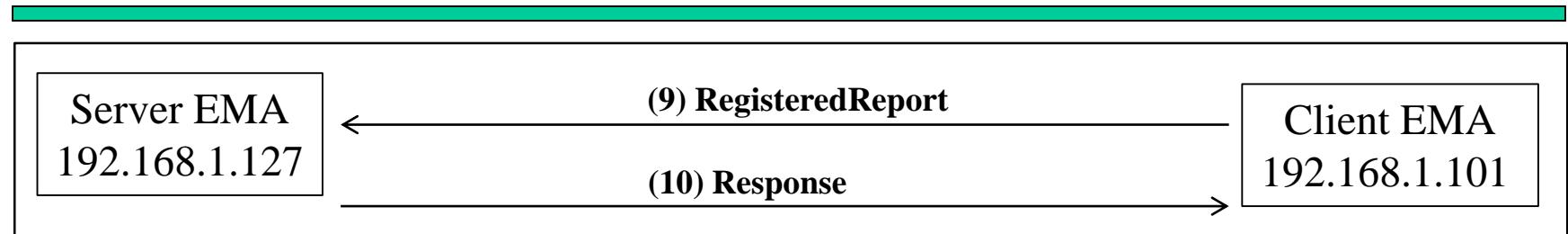
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(9) RegisteredReport

CoAP	192.168.1.101	192.168.1.127	CON, MID:15201, PUT, /RegisteredReport (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:15201, 2.05 Content (application/json)

(10) Response

RegisteredReport Object{

“SrcEMA” : String,  
“DestEMA” : String,  
“requestID”: String,  
“responseCode” : Integer,  
“responseDescription”: String,  
“transportName: String”  
→ 기존 OpenADR에 없는 내용,  
“version”: Integer  
→ 삭제,  
“threshold”: Double  
→ DistributeEvent로 이동,  
“service”: String,  
“type” : String,  
“time” : Date,  
}

- JavaScript Object Notation: application/json
  - Object
    - ▷ Member Key: SrcEMA
    - ▷ Member Key: DestEMA
    - ▷ Member Key: transportName
    - ▷ Member Key: requestID
    - ▷ Member Key: responseCode
    - ▷ Member Key: version
    - ▷ Member Key: responseDescription
    - ▷ Member Key: threshold
    - ▷ Member Key: service
    - ▷ Member Key: time

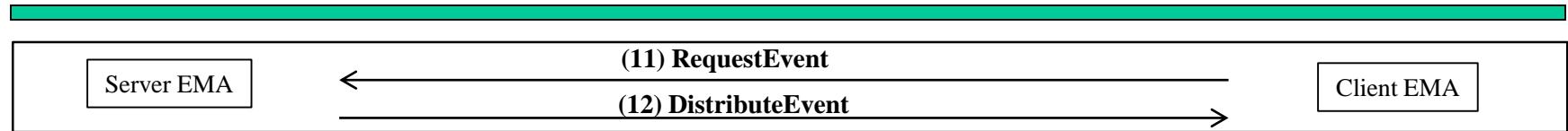
Response Object{

“SrcEMA” : String,  
“DestEMA” : String,  
“requestID”: String,  
“responseCode” : Integer,  
“responseDescription”: String,  
“version”: Integer  
→ 삭제,  
“service”: String,  
“time” : Date

- JavaScript Object Notation: application/json
- Object
  - ▷ Member Key: SrcEMA
  - ▷ Member Key: responseDescription
  - ▷ Member Key: requestID
  - ▷ Member Key: service
  - ▷ Member Key: time
  - ▷ Member Key: DestEMA
  - ▷ Member Key: version
  - ▷ Member Key: responseCode

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup

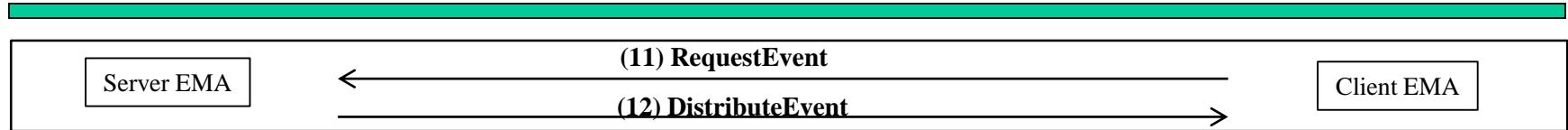


(11) RequestEvent

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:venID	
DestEMA	ei:vtnID	
requestID	pyld:eiRequestEvent	pyld:requestID
service	(tag 이름으로 존재)	
time		RandomizableEvent:creation Time

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



(12) DistributeEvent

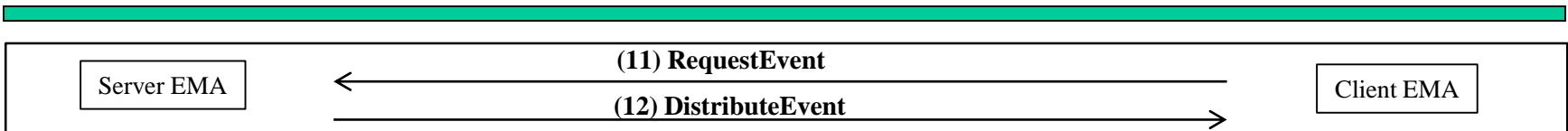
Key Name	Reference	
SrcEMA	OpenADR 2.0b	SEP 2.0(IEC 61968)
DestEMA	ei:vtmID	
requestID	ei:venID	
responseRequired	ei:requestID	
response	Ei:reponseRequired	
requestID	pyld:requestID	
responseCode	ei:responseCode	
responseDescription	ei:responseDescription	
eventID	eventID	
<b>eventSignals</b>	eventSignals	
modificationNumber	modificationNumber	
modificationReason	modificationReason	
priority	priority	
marketContext	eiMarketContext	
createdDateTime	createdDateTime	
eventStatus	eventStatus	
testEvent	testEvent	
vtnComment	vtnComment	
properties	properties	
components	components	
<b>specificDestEMA</b>	venID	
dtStart	dtstart	
Duration	duration	
Tolerance	tolerance	
notification	x-eiNotification	
rampUp	x-eiRampUp	
Recovery	x-eiRecovery	
eventSignal	eiEventSignal	
<b>Intervals</b>	intervals	
signalName	signalName	
<b>signalType</b>	signalType	
(Price Event, Control Event, Reserve Mode, RealtimeDR)		
signalID	signalID	
currentValue	currentValue	
event: eventSignals		

signalType으로 Price Event인지

Control Event, Reserve Mode, RealtimeDR인지 구분한다

event: eventSignals

## 2.2 EMAP(MQTT, CoAP/JSON) Service : Session Setup

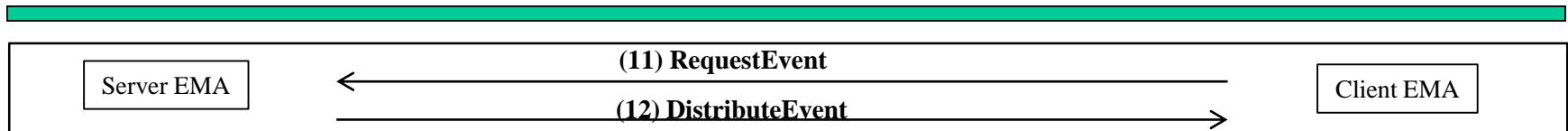


(11) RequestEvent

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
service	type of service
time	service creation time

## 2.2 EMAP(MQTT, CoAP/JSON)

# Service : Session Setup



### (12) DistributeEvent

	Key name	Comments
SrcEMA		source EMA identifier
DestEMA		destination EMA identifier
response	requestID	request identifier
	responseCode	response code
	responseDescription	description of response code
	eventID	event identifier
event	eventSignals	duration
		event signal interval duration
		uid
		event user id
		value
		event value
		signalName
		event signal name
		signalType
		event signal type (bi direct, level)
		signalID
		event signal ID
		currentValue
		current usage value
responseRequired	service	threshold
		available amount of energy
		capacity
		사용 가능량 (threshold - power)
		price
		price of energy
		단위
		modificationNumber
		modification Number(count)
		modificationReason
		modification reason(event reason)
		priority
		priority
		marketContext
time	service	market address(market reference)
		createdDateTime
		event create date & time
		eventStatus
		event status
		testEvent
		if event test or not
		vtnComment
		dtStart
		event start time
		duration
		event duration
		properties
		components
		specificDestEMA
notification	service	specific target EMA
		tolerance
		tolerance duration
		notification
rampUp	service	notification duration
		rampUp
		ramp up duration
		recovery
service	time	response mandatory or not
		type of service
		service creation time

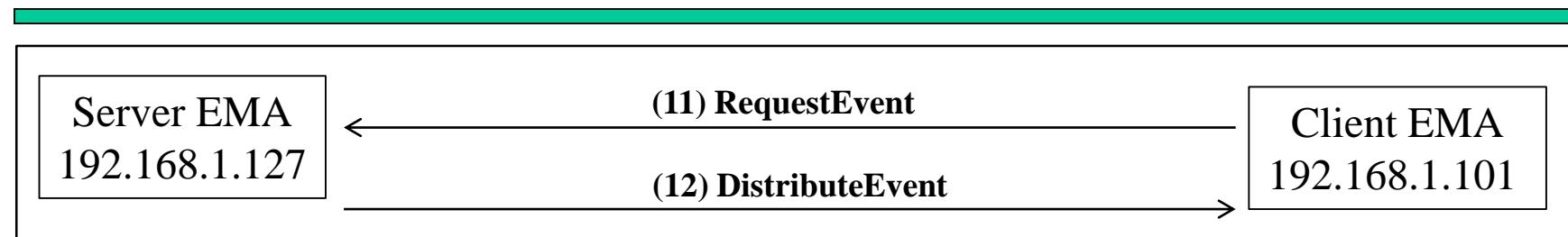
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(11) RequestEvent	CoAP	192.168.1.101	192.168.1.127	CON, MID:44189, PUT, /RequestEvent (application/json)
(12) DistributeEvent	CoAP	192.168.1.127	192.168.1.101	ACK, MID:44189, 2.05 Content (application/json)

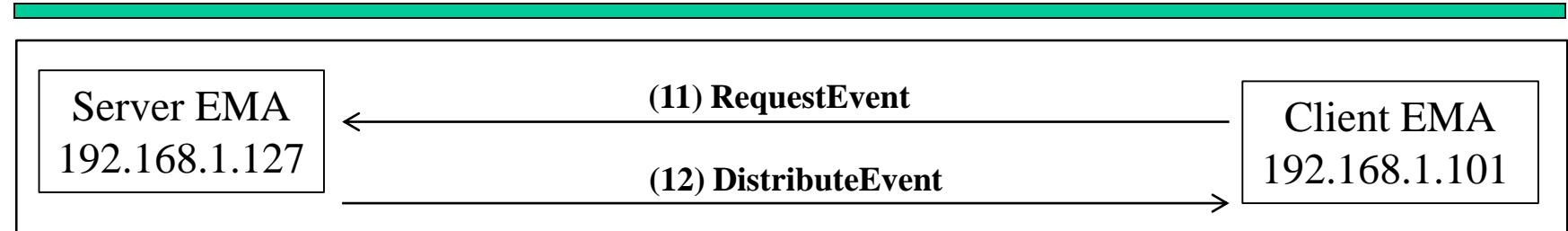
```
RequestEvent Object{
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "replyLimit": Integer => 삭제,
    "time": Date,
    "service": String
}
```

- JavaScript Object Notation: application/json
  - Object
    - ▷ Member Key: SrcEMA
    - ▷ Member Key: DestEMA
    - ▷ Member Key: requestID
    - ▷ Member Key: replyLimit
    - Member Key: service
      - String value:
      - Key: service
    - Member Key: time
      - String value: 2018-04-18 07:18:39
      - Key: time

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(11) RequestEvent	CoAP	192.168.1.101	192.168.1.127	CON, MID:44189, PUT, /RequestEvent (application/json)
(12) DistributeEvent	CoAP	192.168.1.127	192.168.1.101	ACK, MID:44189, 2.05 Content (application/json)

DistributeEvent Object{  
    “SrcEMA” : String,  
    “DestEMA” : String,  
    “requestID” : String,  
    “response” : Array,  
    “event” : Array,  
    “responseRequired” : String,  
    “service” : String,  
    “time” : Date  
}

response Array{  
    “requestID” : String,  
    “responseCode” : Integer,  
    “responseDescription” : String  
}

JavaScript Object Notation: application/json  
Object  
    Member Key: SrcEMA  
    Member Key: responseDescription  
    Member Key: requestID  
    Member Key: service  
    Member Key: EMADREventInformation  
    Member Key: time  
    Member Key: DestEMA  
    Member Key: type  
    Member Key: EMADRPriceInformation  
    Member Key: responseCode

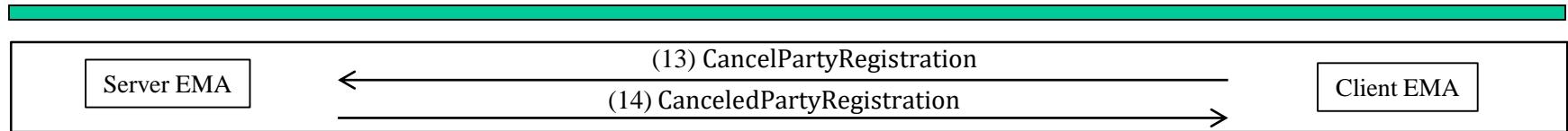
event Array{  
    “eventID” : String,  
    “eventSignals” : Array,  
    “modificationNumber” : Integer,  
    “modificationReason” : String,  
    “priority” : Integer,  
    “marketContext” : String,  
    “createdDataTime” : Date,  
    “eventStatus” : String,  
    “testEvent” : Boolean,  
    “vtnComment” : String,  
    “dtstart” : Date,  
    “duration” : String,  
    “properties” : String,  
    “components” : String,  
    “specificDestEMA” : String,  
    “tolerance” : String,  
    “notification” : String,  
    “rampUp” : String,  
    “recovery” : String  
}

eventSignals Object{  
    “eventSignal” : String,  
    “intervals” : Array,  
    “signalName” : String,  
    “signalType” : String, (Price Event, Control Event, Reserve Mode, RealtimeDR인지 구분)  
    “signalID” : String,  
    “currentValue” : Double,  
    “threshold” : Double,  
    “capacity” : Double,  
    “price” : Integer,  
    “unit” : String,  
}

intervals Array{  
    “duration” : String,  
    “uid” : Integer,  
    “value” : Double  
}

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Session Setup



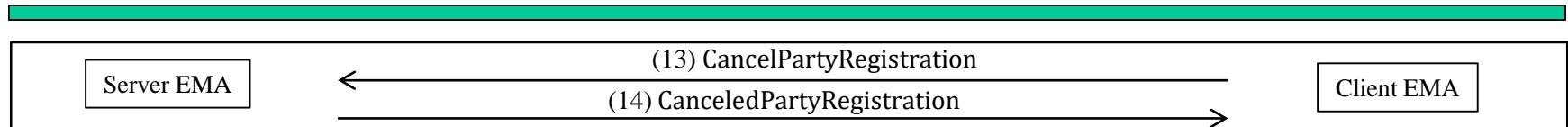
(13) CancelPartyRegistration

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei: venID	
DestEMA	ei: vtnID	
requestID	pyld:requestID	
regatistronID	regatistronID	
service	(Tag 이름으로 존재)	
time		RandomizableEvent:creation Time

(14) CanceledPartyRegistration

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA		ei:vtnID
DestEMA		ei:venID
responseCode		ei:responseCode
responseDescription	ei:eiResponse	ei:responseDescription
requestID		Pyld:requestID
service		(Tag 이름으로 존재)
regatistronID		
time		RandomizableEvent:creationTime

## 2.2 EMAP(MQTT, CoAP/JSON) Service : Session Setup



(13) CancelPartyRegistration

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
registrationID	registration identifier
service	type of service
time	service creation time

(14) CanceledPartyRegistration

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
registrationID	registration identifier
service	type of service
time	service creation time

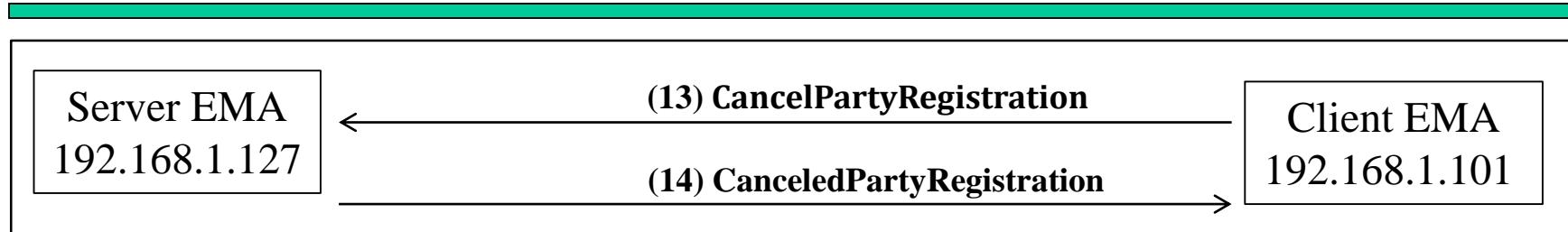
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(13) CancelPartyRegistration

(14) CanceledPartyRegistration

```
CancelPartyRegistration Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID" : String,
    "registrationID" : String
    "service" : String,
    "time" : Date
}
```

```
CanceledPartyRegistration Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID" : String,
    "responseCode" : Integer,
    "responseDescription" : String,
    "registrationID" : String
    "service" : String,
    "time" : Date
}
```

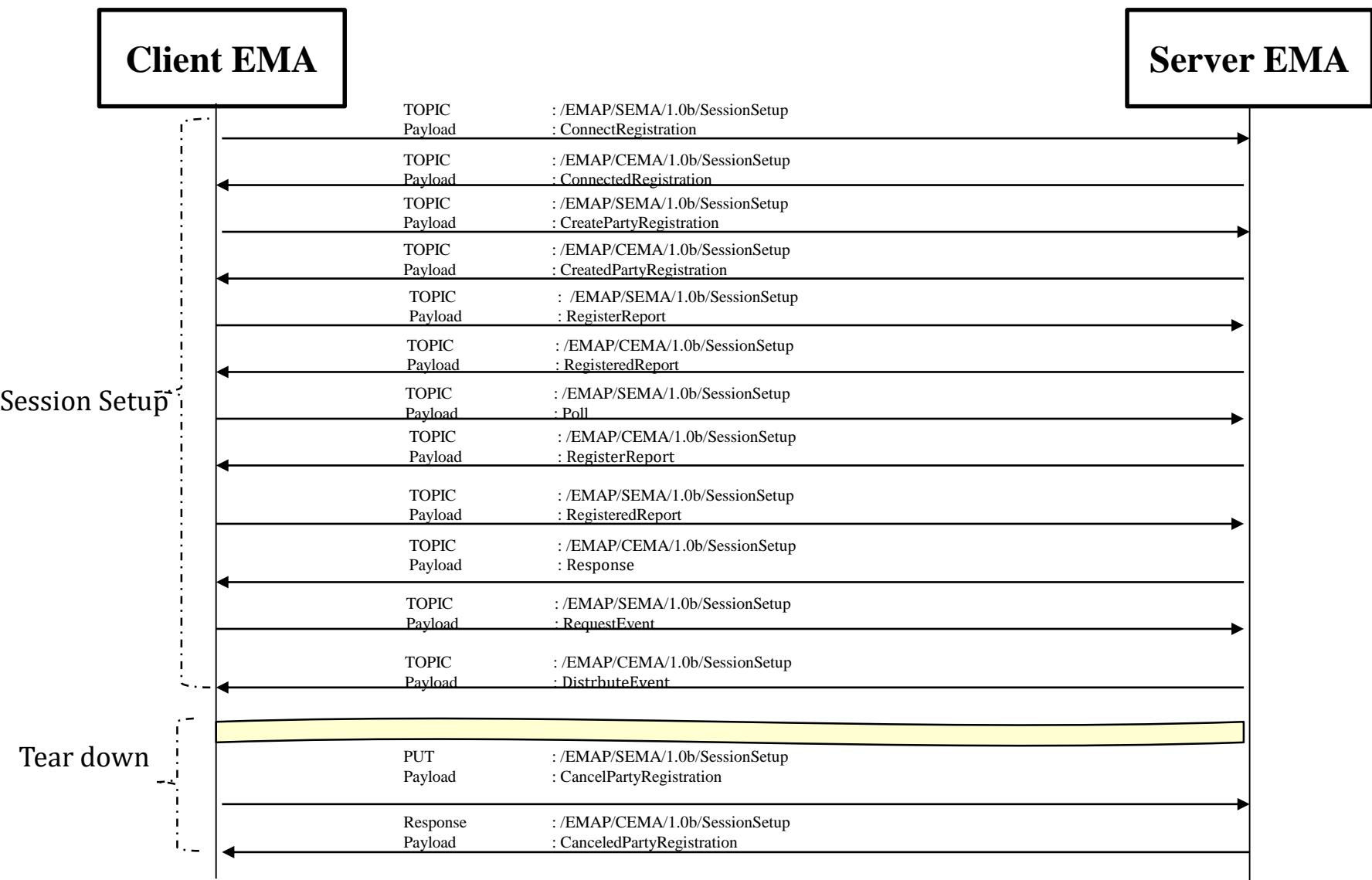
# EMAP

## (1) Session Setup (Registration)

- CoAP / JSON
- MQTT / JSON

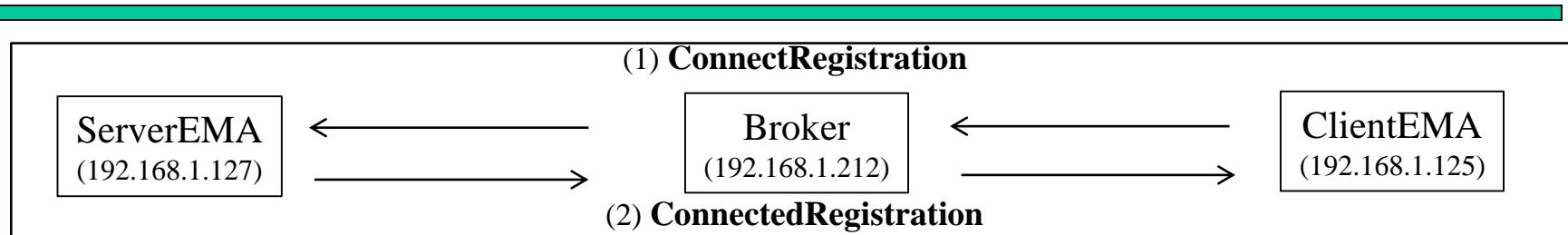
## 2.2 EMAP(MQTT/JSON)

# Service : Session Setup



## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(1) ConnectRegistration

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/ConnectRegistration]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/ConnectedPartyRegistration]

(2) ConnectedRegistration

ConnectRegistration Object{

```

    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "service": String,
    "version": String,
    "time": Date,
    "type": String => RegisteredReport로 변경
    "QoS": String => RegisterReport로 변경
}

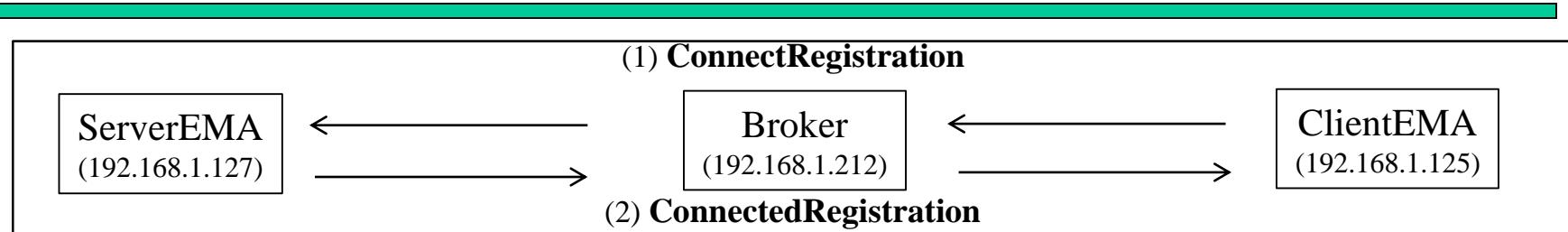
```

Topic: SEMA/SERVER\_EMA/oadrRegistration/ConnectRegistration

Message: { "venID": "VEN\_M1R1", "requestID": 1, "Version": 3, "type": "CEMA", "priority": 1, "QoS": "Controllable", "time": "2018-5-29 2:17:15", "emaCNT": 0 }

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



- (1) ConnectRegistration  
 (2) ConnectedRegistration

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/ConnectRegistration]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/ConnectedPartyRegistration]

ConnectedRegistration Object{

```

    "SrcEMA": String,
    "DestEMA": String,
    "responseCode": Integer,
    "responseDescription": String,
    "requestID": String,
    "duration": Integer,
    "profile": Array,
    "profileName": String,
    "TransportName": String,
    "service": String,
    "version": String,
    "time": Date,
    "registrationID": String
    "type": String => RegisteredReport로 변경
    "QoS": String => RegisterReport로 변경
}

```

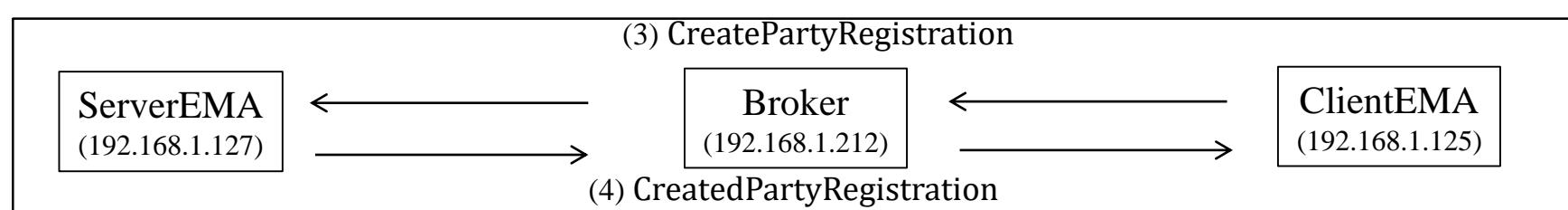
```

        oadrProfile Array{
            "oadrTransports": Array,
            "oadrProfileName": String
        }
        oadrTransports Array{
            "oadrTransportName": String,
        }

        oadrRegistration/ConnectedPartyRegistration{"profileName": "MIR_EnergyManagement_System", "transportName": "MQTT", "priority": 2, "type": "CEMA", "version": "version 1.1", "responseCode": 200, "duration": 60, "venID": "VEN_MIRE1", "QoS": "Contractable", "responseDescription": "OK", "requestID": 1, "emaCNT": 0, "registrationID": "c92c0babdc764d8674bce14a55d867d", "time": "Tue May 29 02:17:15 KST 2018", "vtID": "SERVER_EMA1"}
    
```

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(3) CreatePartyRegistration  
 (4) CreatedPartyRegistration

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/CreatePartyRegistration]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/CreatedPartyRegistration]

CreatePartyRegistration Object{

```

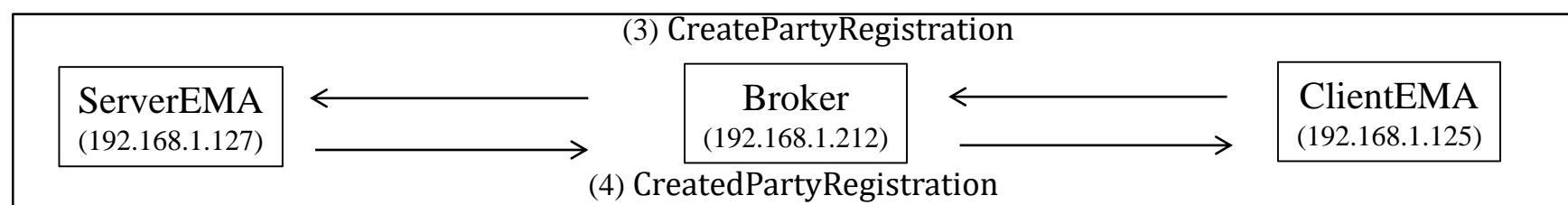
    "SrcEMA": String,
    "DestEMA": String,
    "version": Integer → 삭제,
    "requestID": String,
    "transportName": String,
    "reportOnly": Integer,
    "httpPullModel": Boolean,
    "profileName": String,
    "xmlSignature": String,
    "registrationID": String 삭제,
    "service": String,
    "time": Date
}
  
```

```

CreatePartyRegistration{ "venID": "VEN_M_1R1", "transport Name": "OpenADR MQTT",
"transpor tAddress": "192. 168.1.212", "req uestID": 1, "ver sion": 3, "repor
tOnly": 0, "http PullModel": 0, " profileName": 3, "xmlSignature": 0,
"registratio nID": "VEN_MIR1 ", "time": "2018 -5-29 2:17:15" }
  
```

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(3) CreatePartyRegistration  
 (4) CreatedPartyRegistration

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/CreatePartyRegistration]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/CreatedPartyRegistration]

**CreatedPartyRegistration Object{**

- “SrcEMA” : String,
- “DestEMA” : String,
- “version”: Integer => 삭제,
- “requestID” : String,
- “oadrProfile”: Array,
- “registrationID”: String,
- “duration” : String,
- “responseCode” : Integer,
- “responseDescription”: String,
- “service”: String
- “time” : Date,

**}**

```

oadrProfile Array{
    "oadrTransports": Array,
    "oadrProfileName": String
}

```

```

oadrTransports Array{
    "oadrTransportName": String,
}

```

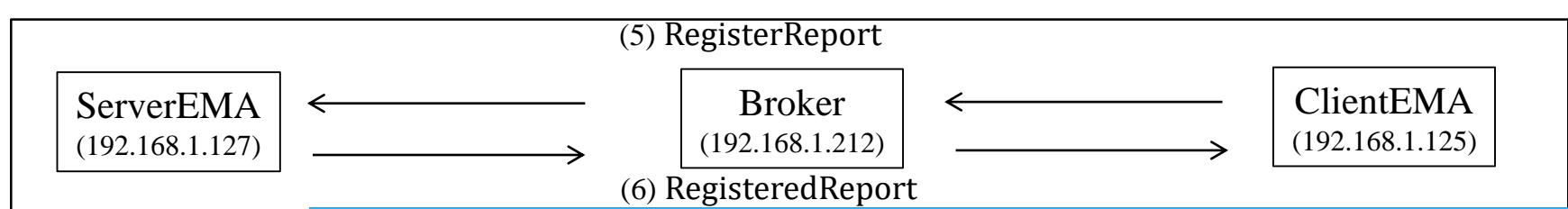
```

"odrRegistration/CreatedPartyRegistration": {
    "duration": 60,
    "profileName": "MIR Energy Management System",
    "transportName": "MQTT",
    "venID": "VEN_MIR1",
    "responseDescription": "OK",
    "requestID": 1,
    "registrationID": "c92c0babdc764d8674bceaa14a55d867d",
    "time": "Tue May 29 02:17:15 KST 2018",
    "version": "version_1.1",
    "vtID": "SERVER_EMA1",
    "responseCode": 200
}

```

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(5) RegisterReport	MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/RegisterReport]
(6) RegisteredReport	MQTT	192.168.1.125	192.168.1.212	Disconnect Req
	MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/RegisteredReport]

#### RegisterReport Object{

```

    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "reportType" : String,
    "EMAreigisteredDRinformation" : Object => 변경
    "EMAreigisteredMgnInformation" : Object => 변경,
    - "report" : Array,
    "time": Date,
    "service" : String
}

```

```

/oadrRegistration/Register Report{ "reportName": 3, "venID" : "VEN_MIR1", "i
semunit": "MIR" , "maketContext" : "192.168.1.212 ", "createDataTi me": "2018-5-
29 2:17:15", "siScl encode": 0, "requ estID": 1, "reso urceID": 1, "rep ortType":
3, "re adingType": 1, "oadrReport": "Re gisterReport", "duration": 2, "r id":
1, "hertz": 60, "voltage": 270, "ac": 1, "oadrMinPeriod": 0 , "oadrMAXPeriod ":
0, "oadrReport Description": "M IRLAB RegisterRe port", "energyRe al": "OpenADRPower
erTest", "oadrOn Change": 0, "rep ortRequestID": 1 , "reportSpecifi erID": 1,
"state " : "ON", "dimmin g": 0, "power": 0.100000, "maxVa lue": 0.100000,
"minValue": 0.10 0000, "avgValue" : 0.100000, "tim e": "2018-5-29 2 :17:15",
"maxTim e": "2018-5-29 2 :17:15", "minTim e": "2018-5-29 2 :17:15" }

```

#### (5) RegisterReport

#### Broker (192.168.1.212)

#### (6) RegisteredReport

#### ClientEMA (192.168.1.125)

```

report Object{
    "duration" : String,
    "reportRequestID" : Integer,
    "reportSpecifierID" : String,
    "reportName" : String,
    "createdDateTime" : Date,
    "reportDescription" : Array,
}

```

#### reportDescription Object{

```

    "rID" : String,
    "resourceID" : String,
    "deviceType" : String,
    "reportType" : String,
    "itemUnits" : String,
    "siScaleCode" : String,
    "marketContext" : String,
    "oadrMinPeriod" : String,
    "oadrMaxPeriod" : String,
    "oadrOnChange" : String,
    "itemDescription" : String,
    "powerAttributes" : Array,
    "qos" : String
    "state" : String,
    "power" : Double,
    "dimming" : Integer,
    "margin" : double,
    "generate" : double,
    "storage" : String,
    "maxValue" : Double,
    "minValue" : Double,
    "avgValue" : Double,
    "maxTime" : Date,
    "minTime" : Date,
    "priority" : Integer
}

```

#### powerAttributes Object{

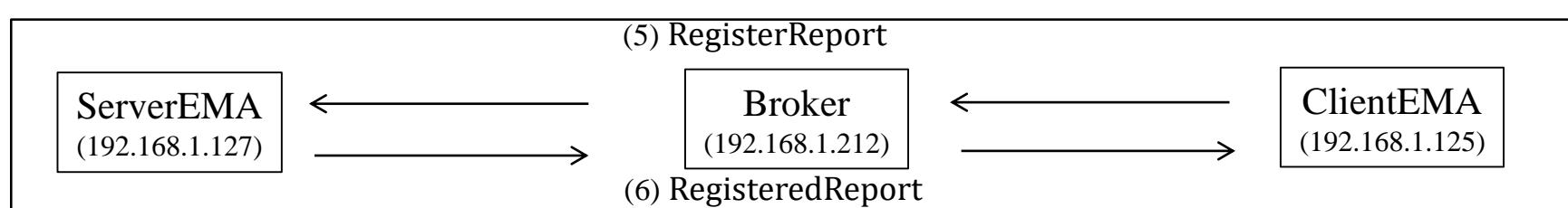
```

    "hertz" : Double,
    "voltage" : Double,
    "ac" : Boolean
}

```

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(5) RegisterReport	MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/RegisterReport]
(6) RegisteredReport	MQTT	192.168.1.125	192.168.1.212	Disconnect Req
	MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/RegisteredReport]

```

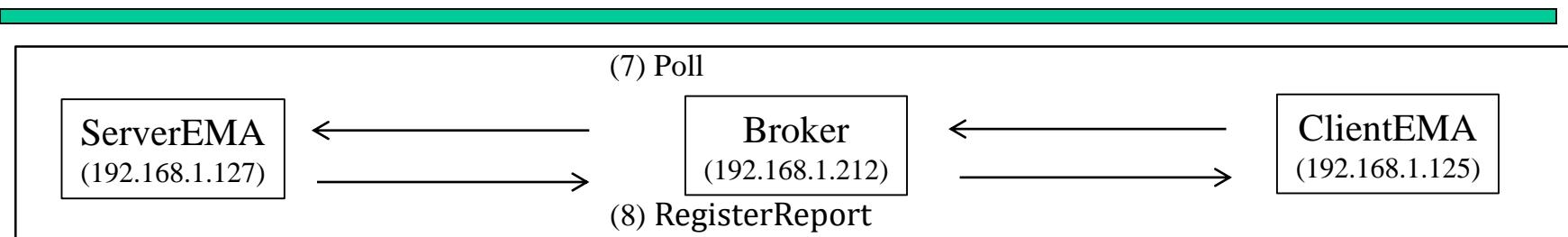
RegisteredReport Object{
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription": String,
    "transportName": String => 기존 OpenADR에 없는 내용,
    "version": Integer => 삭제,
    "threshold": Double => DistributeEvent로 이동,
    "service": String,
    "time": Date,
}
    
```

Topic: CEMA/1/oadrRegistration/RegisteredReport

Message: {"transportName": "MQTT", "venID": "VEN\_MIR1", "responseDescription": "OK", "requestID": 1, "threshold": 0, "time": "Tue May 29 02:17:15 KST 2018", "version": "version\_1.1", "vtnID": "SERVER\_EMA1", "responseCode": 200}

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(7) Poll

(8) RegisterReport

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/Poll]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/RegisterReport]

**Poll JSON Object{**

“SrcEMA” : String,  
“DestEMA” : String,  
“requestID”: Integer, -> 삭제

“version”: Integer, -> 삭제,

“type” : String -> 삭제(RegisteredReport로 변경)

“service” : String,  
“time” : Date

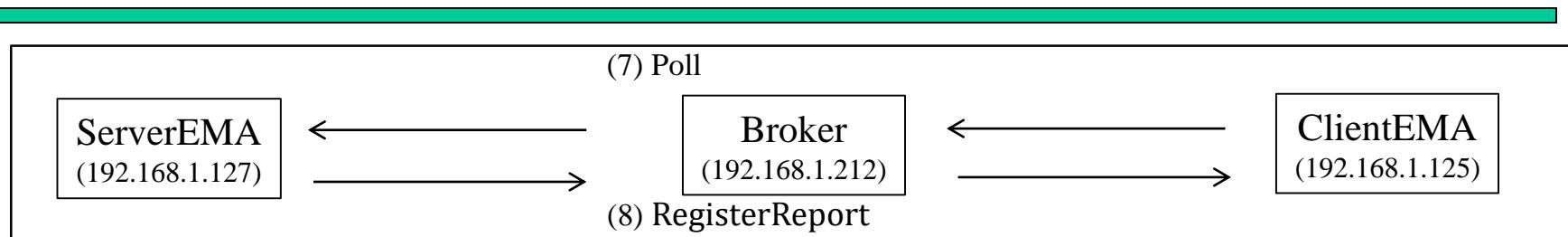
}

Topic: SEMA/SERVER\_EMA/oadrRegistration/Poll

Message: { “venID”: “VEN\_MIR1”, “requestID”: 1, “version”: 3, “time”: “2018-5-29 2:17:15” }

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(7) Poll

(8) RegisterReport

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/Poll]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/RegisterReport]

```

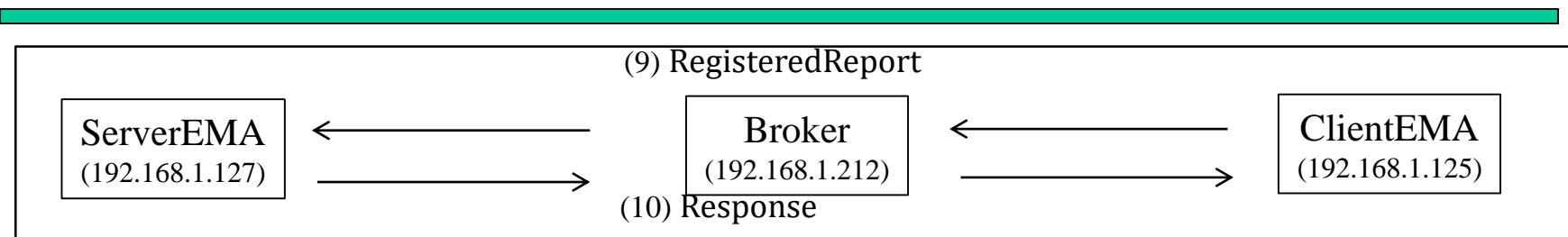
RegisterReport Object{
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "time": Date,
    "service": String
}
    
```

Topic: CEMA/1/oadrRegistration/RegisterReport

Message: {"duration":60,"transportName":"MQTT","venID":"VEN\_MIRI","responseDescription":"OK","requestID":1,"time":"Tue May 29 02:17:15 KST 2018","version":"version\_1.1","vtnID":"SERVER\_EMA1","responseCode":200}

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(9) RegisteredReport

(10) Response

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/RegisteredReport]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/Response]

**RegisteredReport Object{**

```

    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription": String,
    "transportName": String => 기존 OpenADR에 없는 내용,
    "version": Integer => 삭제,
    "threshold": Double => DistributeEvent로 이동,
    "service": String,
    "time": Date,
}
    
```

Topic: SEMA/SERVER\_EMA/oadrRegistration/RegisteredReport

Message: { "venID": "VEN\_MIR1", "requestID": 1, "version": 3, "responseDescription": "200 OK", "time": "2018-5-29 2:17:15" }

**Response Object{**

```

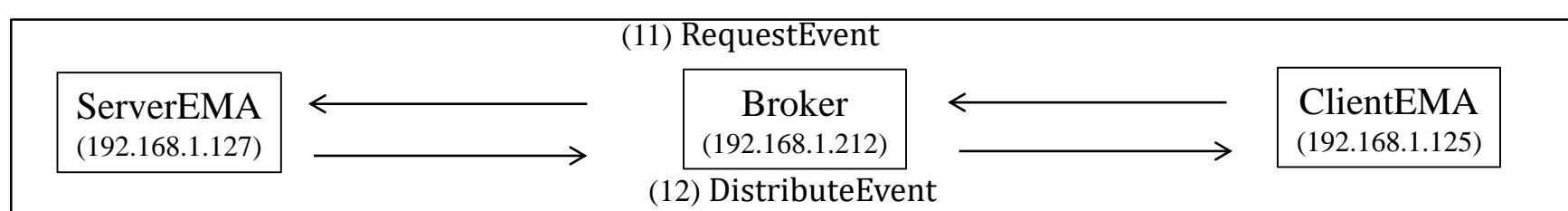
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription": String,
    "version": Integer => 삭제,
    "service": String,
    "time": Date
}
    
```

Topic: CEMA/1/oadrRegistration/Response

Message: { "venID": "VEN\_MIR1", "responseDescription": "OK", "requestID": 1, "time": "Tue May 29 02:17:15 KST 2018", "version": "version\_1.1", "venID": "SERVER\_EMA", "responseCode": 200 }

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(11) RequestEvent

(12) DistributeEvent

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/RequestEvent]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/DistributeEvent]

RequestEvent Object{

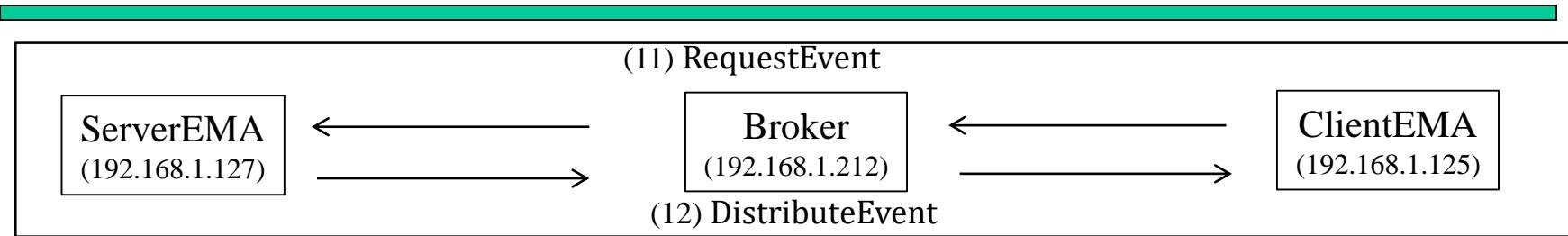
    “SrcEMA” : String,  
    “DestEMA” : String,  
    “requestID”: String,  
        “replyLimit” : Integer => 삭제,  
    “time” : Date,  
    “Service”: String  
}

Topic: SEMA/SERVER\_EMA/oadrRegistration/RequestEvent

Message: { “venID”: “VEN\_MIR1”, “requestID”: 1, “replyLimit”: 0, “time”: “2018-5-29 2:17:15” }

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(11) RequestEvent

(12) DistributeEvent

MQTT	192.168.1.125	192.168.1.212	Publish Message [SEMA/SERVER_EMA/oadrRegistration/RequestEvent]
MQTT	192.168.1.125	192.168.1.212	Disconnect Req
MQTT	192.168.1.212	192.168.1.125	Publish Message [CEMA/1/oadrRegistration/DistributeEvent]

**DistributeEvent Object{**

```

    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID" : String,
    "response" : Array,
    "event" : Array,
    "responseRequired" : String,
    "service" : String,
    "time" : Date
}

```

**response Array{**

```

    "requestID" : String,
    "responseCode" : Integer,
    "responseDescription" : String
}

```

**event Array{**

```

    "eventID" : String,
    "eventSignals" : Array,
    "modificationNumber" : Integer,
    "modificationReason" : String,
    "priority" : Integer,
    "eiMarketContext" : String,
    "createdDataTime" : Date,
    "eventStatus" : String,
    "testEvent" : Boolean,
    "vtnComment" : String,
    "dtstart" : Date,
    "duration" : String,
    "properties" : String,
    "components" : String,
    "specificDestEMA" : String,
    "tolerance" : String,
    "notification" : String,
    "rampUp" : String,
    "recovery" : String
}

```

**eventSignals Object{**

```

    "eiEventSignal" : String,
    "intervals" : Array,
    "signalName" : String,
    "signalType" : String, (Price Event, Control Event, Reserve Mode, RealtimeDR인지 구분)
    "signalID" : String,
    "currentValue" : Double,
    "threshold" : Double,
    "capacity" : Double,
    "price" : Integer,
    "unit" : String,
}

```

**intervals Array{**

```

    "duration" : String,
    "uid" : Integer,
    "value" : Double
}

```

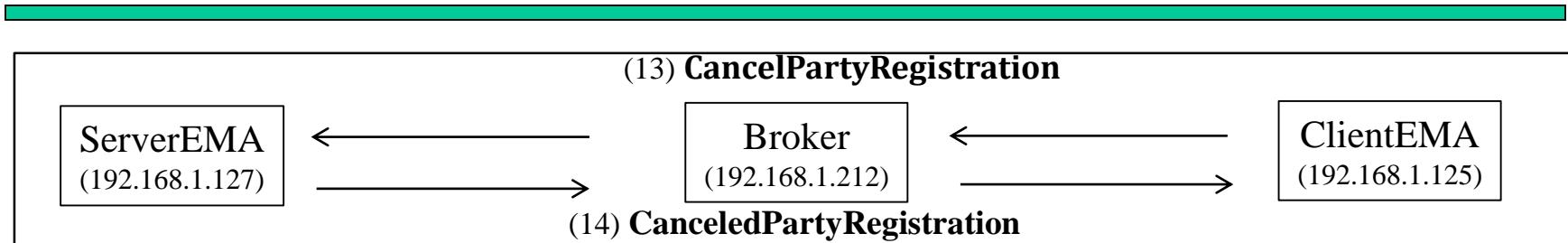
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



**CancelPartyRegistration Object{**

“SrcEMA” : String,  
“DestEMA” : String,  
“requestID” : String,  
“registrationID” : String  
“service” : String,  
“time” : Date  
}

**CanceledPartyRegistration Object{**

“SrcEMA” : String,  
“DestEMA” : String,  
“requestID” : String,  
“responseCode” : Integer,  
“responseDescription” : String,  
“registrationID” : String  
“service” : String,  
“time” : Date  
}

# **EMAP (CoAP & MQTT/JSON)**

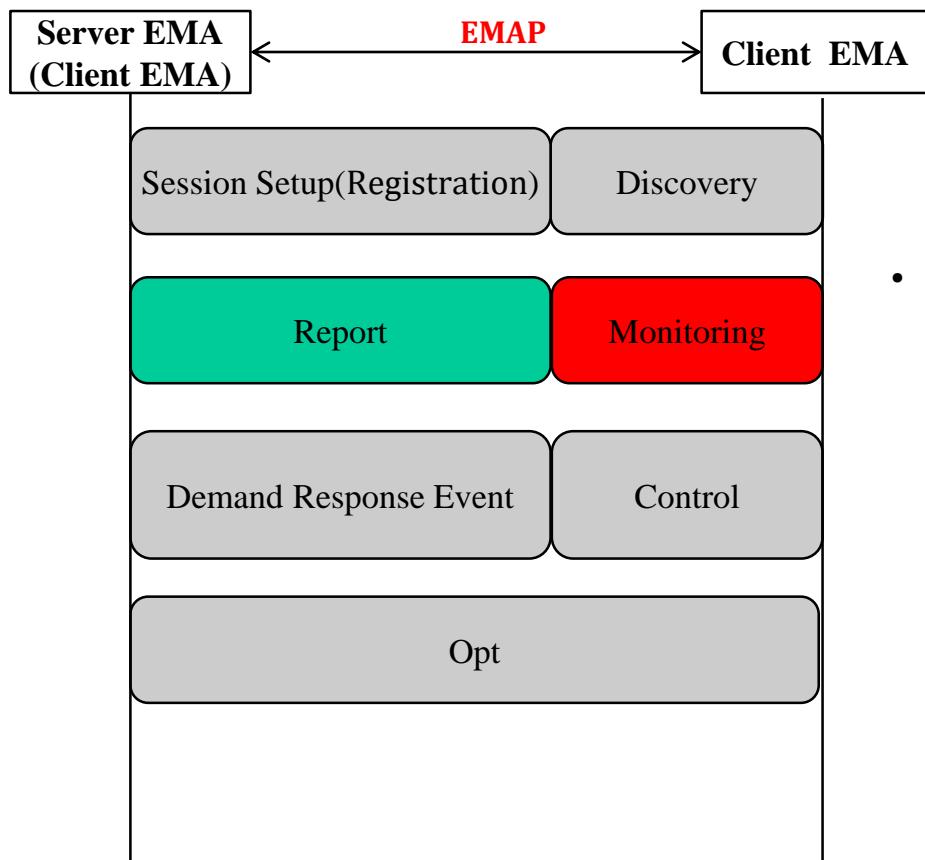
## **UpdateReport**

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Update Report

#### EMAP

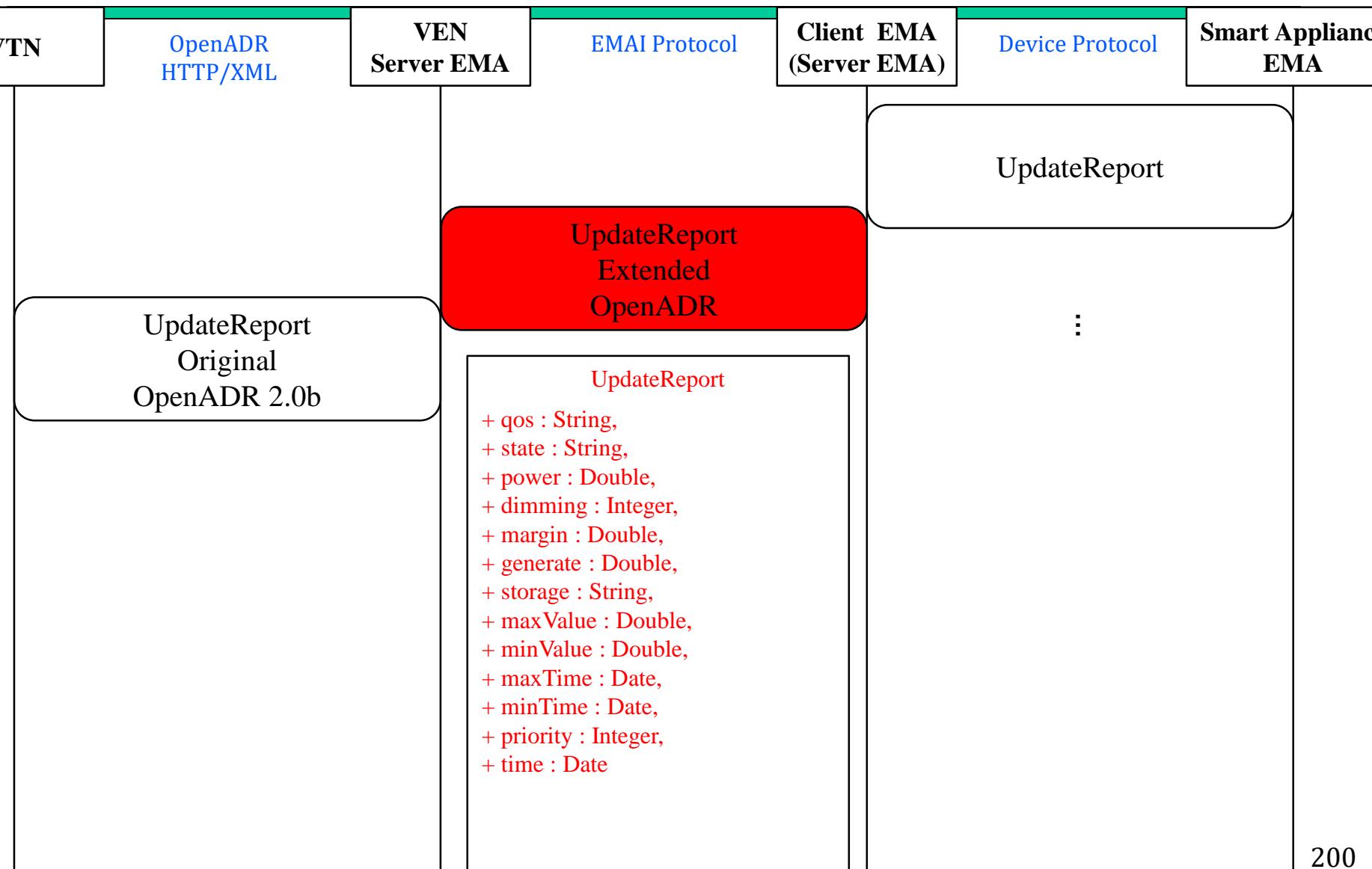
- EMA 사이의 수요반응 통신 프로토콜이며 OpenADR 2.0b의 모델링을 따랐으며 일부 모델링을 확장했고 Discovery와 Monitoring, Control 부분의 통신 프로토콜이 확장.



- Report(updateReport)
  - 에너지관리에이전트 간 서로 연결을 수립 할 때 Report을 교환할 때 실시간 에너지에 대한 가격 정보나 클라이언트 에너지관리에이전트의 디바이스 정보 등을 얻음.
  - 하위의 EMA의 정보를 모니터링 하는 단계, Explicit/Abstract 방식으로 Monitoring.
  - Explicit에는 Device Type (LED, ESS, Recloser, Resource, PV)에 대한 정보가 포함
  - Implicit에는 Device Type을 제외한 정보가 포함되어 있어 필요로 하는 정보에 따라 데이터 트래픽을 다르게 설정할 수 있음.

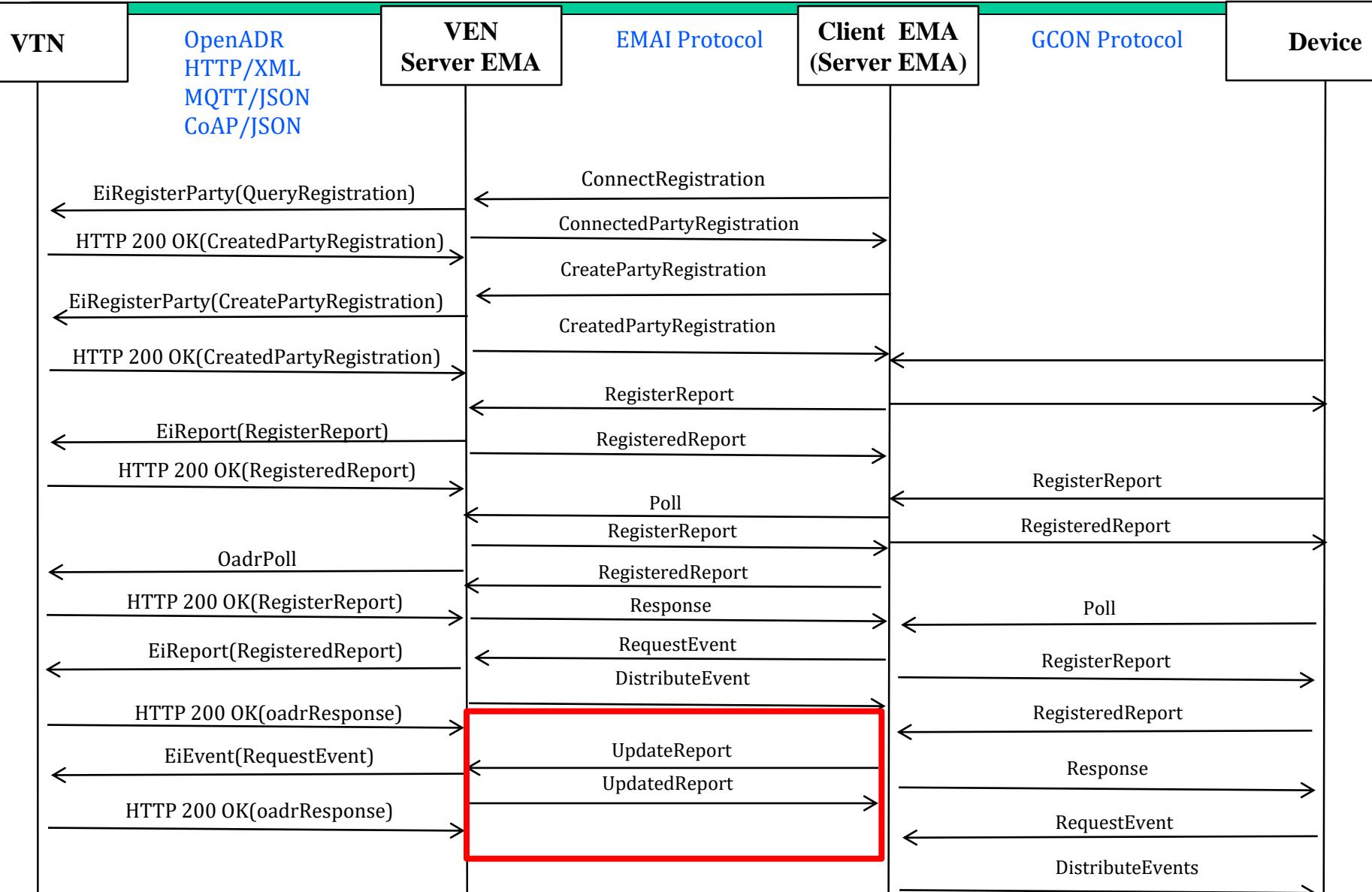
## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Update Report



# 2.2 EMAP(MQTT, CoAP/JSON)

## Service : Update Report

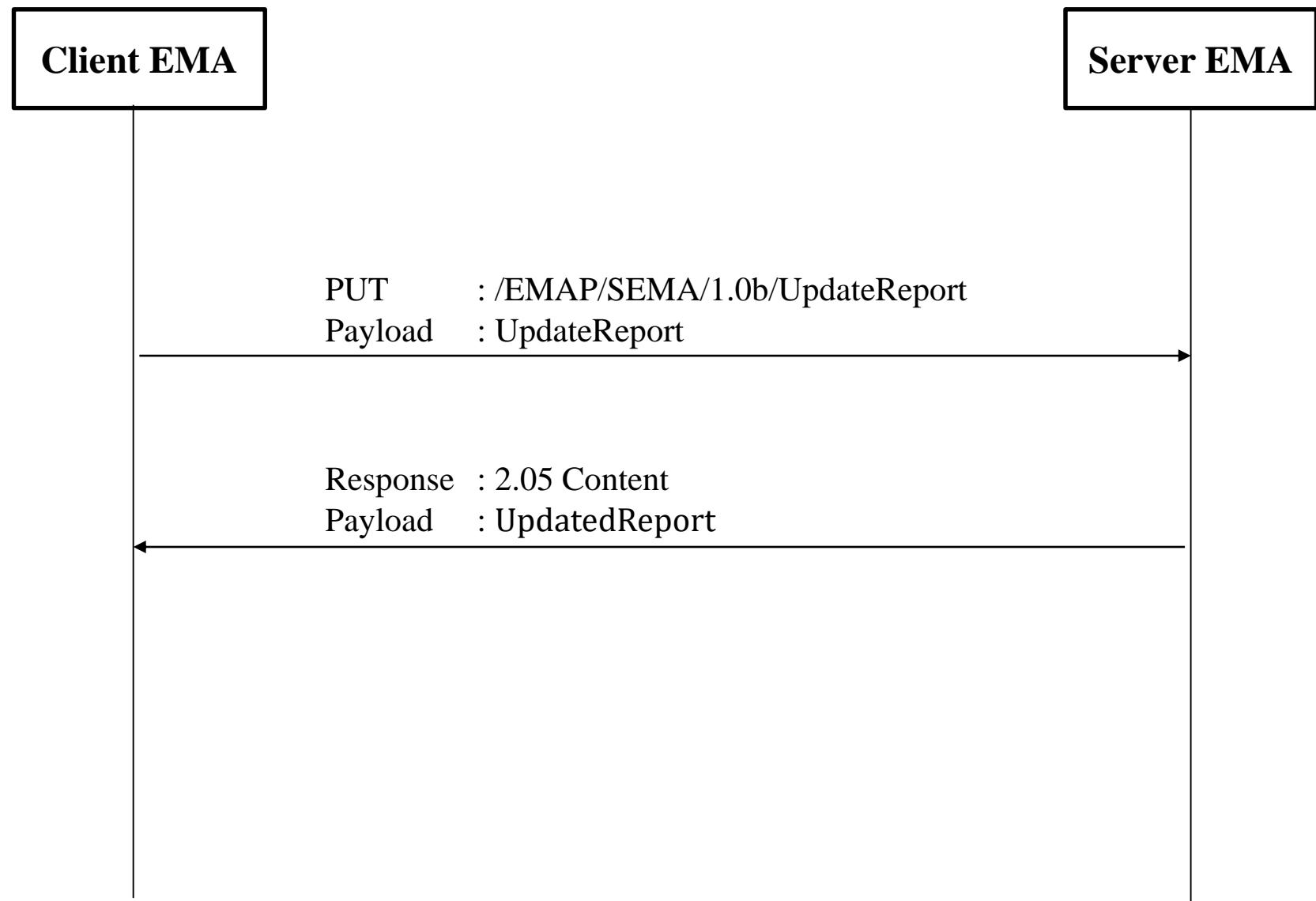


# **EMAP**

## **(2) UpdateReport**

- **CoAP / JSON**
- **MQTT / JSON**

## 2.2 EMAP(CoAP/JSON) Service : Update Report



## 2. Smart Home Energy Framework :

### 2.2 EMAP(MQTT/JSON, CoAP/JSON) : UpdateReport

#### (1) UpdateReport

Key Name		Reference		
		OpenADR 2.0b	SEP 2.0(IEC 61968)	OpenFMB(IEC 61850)
SrcEMA		ei:venID		
DestEMA		ei:vtnID		
service		(Tag 0 름으로 존재)		
time				RandomizableEvent:creation Time
requestID		requestID		
<b>type(Implicit, Explicit)</b>				
report	duration	oadrReport	duration	
	reportDescription		oadrReportDescription	
	reportRequestID		reportRequestID	
	reportSpecifierID		reportSpecifierID	
	reportName		reportName	
	createdDateTime		createdDateTime	
	rID		rID	
	resourceID		resourceID	
	<b>deviceType</b>			EndDeviceControlType:type
	reportType		reportType	
report:reportDescription	itemUnits	oadrReport:oadrReportDescrip tion	itemUnits	
	siScaleCode		siScaleCode	
	marketContext		marketContext	
	MinPeriod		oadrMinPeriod	
	MaxPeriod		oadrMaxPeriod	
	OnChange		oadrOnChange	
	itemDescription		itemDescription	
	<b>powerAttributes</b>		powerAttributes	
	qos			EndDeviceControl:loadShiftForward
	state			DeviceStatus:opState
<b>power</b>	power			Readings: value
	dimming			
	margin			Subscription:Level
	generate			IdentifiedObject:DemandResponseP rogram:availabilityUpdatePowerCh nageThreshold
	storage			SolarEventProfile:SolarInve terrStatus:value
	maxValue			BatteryEventProfile:Batter yStatus:value
	minValue			solarModule:SolarCapabilit y:MaxVal solarModule:SolarCapabilit y:MinVal

## 2. Smart Home Energy Framework :

### 2.2 EMAP(MQTT/JSON, CoAP/JSON) : UpdateReport

(2) UpdatedReport

Key Name	Reference		
	OpenADR 2.0b	SEP 2.0(IEC 61968)	OpenFMB(IEC 61850)
SrcEMA	ei:venID		
DestEMA	ei:vtnID		
requestID		pyld:requestID	
responseCode	ei:eiResponse	ei:responseCode	
responseDescription		ei:responseDescription	
service	(Tag 이름으로 존재)		
type			IdentifiedObject:TrafiProfile: ServiceKind
time			RandomizableEvent:creation Time

type: Explicit, Implicit

## 2. Smart Home Energy Framework :

### 2.2 EMAP(MQTT/JSON, CoAP/JSON) : UpdateReport

#### (1) UpdateReport

	<b>Key name</b>	<b>Comments</b>
SrcEMA		source EMA identifier
DestEMA		destination EMA identifier
requestID		request identifier
duration		report duration
reportRequestID		report request identifier
reportSpecifierID		report specific id (created from ven)
reportName		report name
createdDateTime		created time of this report
report	rID	
	resourceID	resource identifier
	reportType	report type
	deviceType	type of device
	itemUnits	unit of item that report
	siScaleCode	
	marketContext	refer marketContext address
	minPeriod	Energy usage minimum period
	maxPeriod	Energy usage maximum period
	onChange	
	itemDescription	type of item that report
	qos	device QoS
	state	device current state
	power	power usage
	dimming	dimming state
	margin	available amount of energy (Including generated, storaged Energy)
	generate	generated energy
	storage	soraged energy
	maxValue	energy max usage value
	minValue	energy min usage vaalue
	avgValue	energy average usage value
	maxTime	energy max usage time
	minTime	energy min usage time
	priority	priority of this device
service	powerAttributes	hertz
		voltage
		ac
time		pulse frequency of power
type		voltage of power
		Is this AC power? (True or False)
		type of service
		service creation time
		report message type (implicit or explicit)

## 2. Smart Home Energy Framework :

### 2.2 EMAP(MQTT/JSON, CoAP/JSON) : UpdateReport

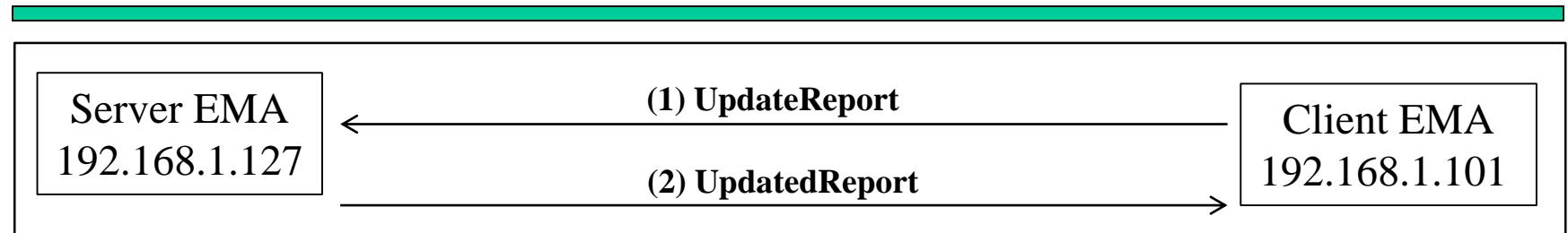
---

(2) UpdatedReport

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	type of service
time	service creation time

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)



(1) UpdateReport	192.168.1.101	192.168.1.127	CoAP	1295 CON, MID:8526, PUT, /updateReport (application/json)
	192.168.1.127	192.168.1.101	CoAP	802 ACK, MID:8526, 2.05 Content (application/json)

```

UpdateReport Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "reportType" : String,
    "EMAreisteredDRinformation" : Object -> 변경
    "EMAreisteredMgnInformation" : Object -> 변경,
    - "report" : Array,
    "time" : Date,
    "service" : String,
    "type" : String (Explicit, Implicit인지 구분)
}

```

```

report Object{
    "duration" : String,
    "reportRequestID" : Integer,
    "reportSpecifierID" : String,
    "reportName" : String,
    "createdDateTime" : Date,
    "reportDescription" : Array,
}

```

▣ Javascript Object Notation: application/json  
 Object  
 + Member Key: "SrcEMA"  
 + Member Key: "DestEMA"  
 + Member Key: "requestID"  
 + Member Key: "reportName"  
 + Member Key: "reportType"  
 + Member Key: "EMAreisteredDRInfo"  
 + Member Key: "EMAreisteredMgnInfo"  
 + Member Key: "service"  
 + Member Key: "time"

```

reportDescription Object{
    "rID" : String,
    "resourceID" : String,
    "deviceType" : String,
    "reportType" : String,
    "itemUnits" : String,
    "siScaleCode" : String,
    "marketContext" : String,
    "oadrMinPeriod" : String,
    "oadrMaxPeriod" : String,
    "oadrOnChange" : String,
    "itemDescription" : String,
    "powerAttributes" : Array,
    "qos" : String,
    "state" : String,
    "power" : Double,
    "dimming" : Integer,
    "margin" : double,
    "generate" : double,
    "storage" : String,
    "maxValue" : Double,
    "minValue" : Double,
    "avgValue" : Double,
    "maxTime" : Date,
    "minTime" : Date,
    "priority" : Integer
}

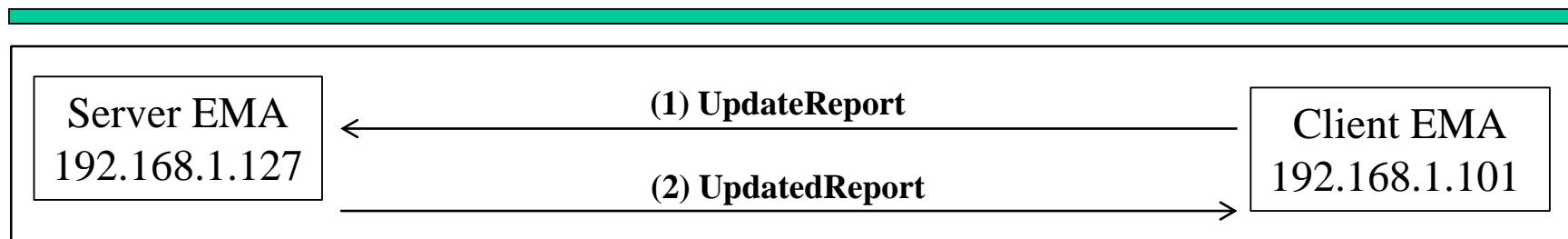
```

```

powerAttributes Object{
    "hertz" : Double,
    "voltage" : Double,
    "ac" : Boolean
}

```

**2. Smart Home Energy Framework :**  
**(HTTP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)**



(2) UpdatedReport	192.168.1.101	192.168.1.127	CoAP	1295 CON, MID:8526, PUT, /updateReport (application/json)
	192.168.1.127	192.168.1.101	CoAP	802 ACK, MID:8526, 2.05 Content (application/json)

```
UpdatedReport Object{  
    "SrcEMA" : String,  
    "DestEMA" : String,  
    "requestID": String,  
    "responseCode": Integer,  
    "responseDescription" : String,  
    "EMAEnergyinfo" : Object -> 사자  
    "service": String,  
    "time": Date  
}
```

```
    □ Payload: Payload Content-Format: application/json, Length: 753  
        Payload Desc: application/json  
        □ Javascript object Notation: application/json  
            □ Object  
                □ Member Key: "EMAEnergyinfo"  
                □ Member Key: "SrcEMA"  
                □ Member Key: "responseDescription"  
                □ Member Key: "requestID"  
                □ Member Key: "service"  
                □ Member Key: "EMATopologyinfo"  
                □ Member Key: "time"  
                □ Member Key: "DestEMA"  
                □ Member Key: "responsecode"
```

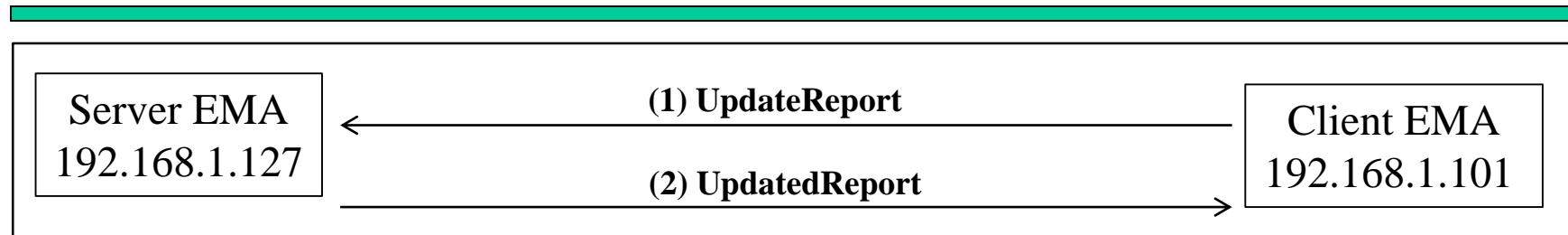
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)



(1) UpdateReport	192.168.1.101	192.168.1.127	CoAP	1295 CON, MID:8526, PUT, /updateReport (application/json)
	192.168.1.127	192.168.1.101	CoAP	802 ACK, MID:8526, 2.05 Content (application/json)

#### UpdateReport Object{

“SrcEMA” : String,  
“DestEMA” : String,  
“requestID”: String,  
“reportType” : String,  
“EMAreisteredDRinformation” : Object -> 변경  
“EMAreisteredMgnInformation” : Object -> 변경,  
- “report” : Array,  
“time”: Date,  
“service” : String,  
“type” : String (Explicit, Implicit인지 구분)  
}

```
report Object{
    “duration” : String,
    “reportRequestID” : Integer,
    “reportSpecifierID” : String,
    “reportName” : String,
    “createdDateTime” : Date,
    “reportDescription” : Array,
}
```

JavaScript Object Notation: application/json  
object  
└ Member Key: "SrcEMA"  
└ Member Key: "DestEMA"  
└ Member Key: "requestID"  
└ Member Key: "reportName"  
└ Member Key: "reportType"  
└ Member Key: "EMAreisteredDRInfo"  
└ Member Key: "EMAreisteredMgnInfo"  
└ object  
 └ Member Key: "emacNT"  
 └ Number value: 5  
 └ Member Key: "topology"  
 └ Array  
 └ Member Key: "state"  
 └ String value:  
 └ Member Key: "dimming"  
 └ Number value: 0  
 └ Member Key: "power"  
 └ Number value: 0  
 └ Member Key: "margin"  
 └ Number value: 0  
 └ Member Key: "generate"  
 └ Number value: 0  
 └ Member Key: "storage"  
 └ Number value: 0  
 └ Member Key: "maxvalue"  
 └ Number value: 0  
 └ Member Key: "minvalue"  
 └ Number value: 0  
 └ Member Key: "avgvalue"  
 └ Number value: 0  
 └ Member Key: "maxTime"  
 └ Date value:  
 └ Member Key: "minTime"  
 └ String value:  
 └ Member Key: "service"  
 └ Member Key: "time"

#### reportDescription Object{

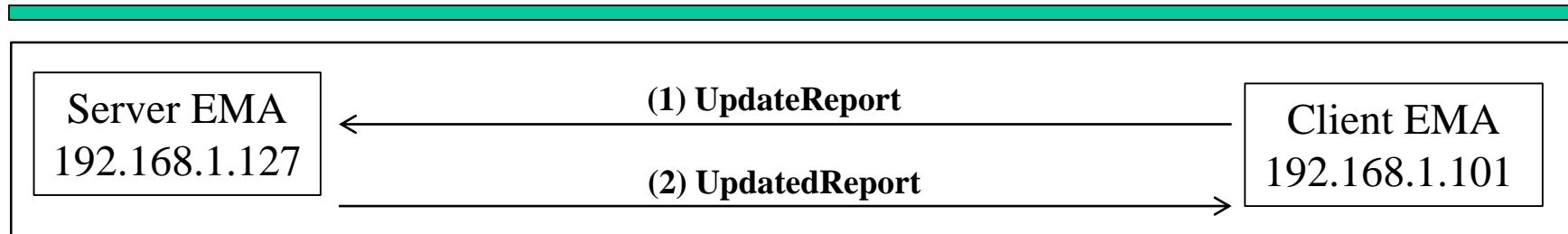
```
“rID” : String,
“resourceID” : String,
“deviceType” : String,
“reportType” : String,
“itemUnits” : String,
“siScaleCode” : String,
“marketContext” : String,
“oadrMinPeriod” : String,
“oadrMaxPeriod” : String,
“oadrOnChange” : String,
“itemDescription” : String,
“powerAttributes” : Array,
“qos” : String
“state” : String,
“power” : Double,
“dimming” : Integer,
“margin” : double,
“generate” : double,
“storage” : String,
“ maxValue” : Double,
“ minValue” : Double,
“ avgValue” : Double,
“ maxTime” : Date,
“ minTime” : Date,
“ priority” : Integer
```

#### powerAttributes Object{

```
“hertz” : Double,
“voltage” : Double,
“ac” : Boolean
}
```

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)



(2) UpdatedReport	192.168.1.101	192.168.1.127	CoAP	1295 CON, MID:8526, PUT, /updateReport (application/json)
	192.168.1.127	192.168.1.101	CoAP	802 ACK, MID:8526, 2.05 Content (application/json)

```
UpdatedReport Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription" : String,
    "EMAEnergyinfo" : Object -> 삭제
    "service": String,
    "time": Date
}
```

```
ayload: Payload Content-Format: application/json, Length: 753
Payload Desc: application/json
JavaScript Object Notation: application/json
Object
+ Member Key: "EMAEnergyinfo"
+ Member Key: "SrcEMA"
+ Member Key: "responseDescription"
+ Member Key: "requestID"
+ Member Key: "service"
- Member Key: "EMATopologyinfo"
- Object
- Member Key: "emaCNT"
    Number value: 5
- Member Key: "topology"
- Array
+ Member Key: "time"
+ Member Key: "DestEMA"
+ Member Key: "responseCode"
```

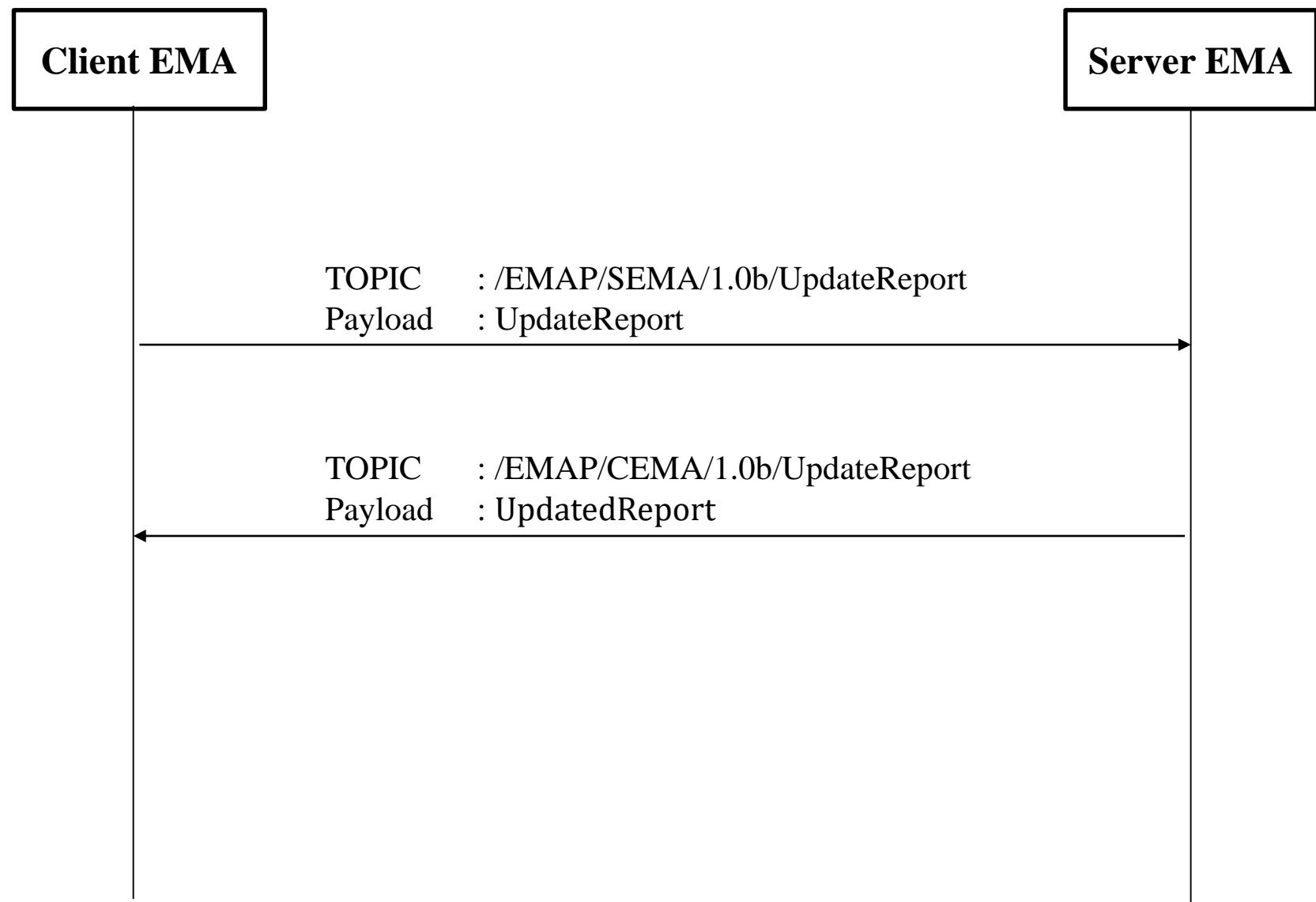
# **EMAP**

## **(2) UpdateReport**

- **CoAP / JSON**
- **MQTT / JSON**

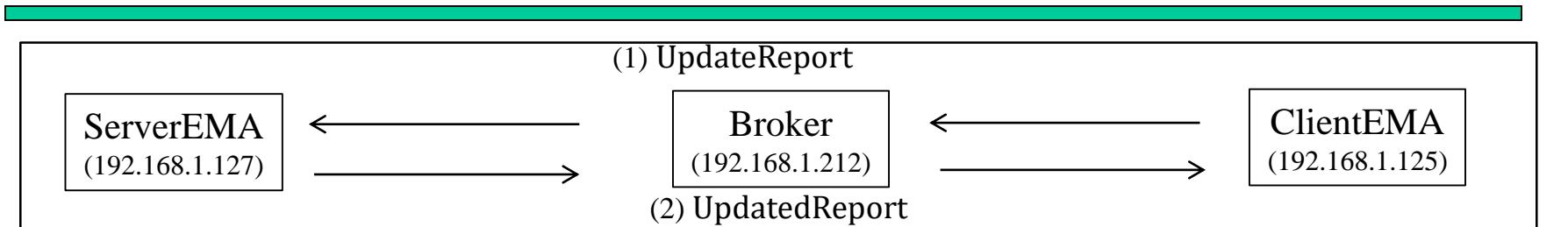
## 2.2 EMAP(MQTT/JSON)

### Service : Update Report



## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)



(1) UpdateReport      MQTT      192.168.1.125      192.168.1.212      Publish Message [SEMA/SERVER\_EMA/oadrReport/UpdateReport]

**UpdateReport Object{**

- “SrcEMA” : String,
- “DestEMA” : String,
- “requestID”: String,
- “reportType” : String,
- “EMAreisteredDRinformation” : Object => 변경
- “EMAreisteredMgnInformation” : Object => 변경,
- “report” : Array,
- “time”: Date,
- “service” : String,
- “type” : String (Explicit, Implicit인지 구분)

**}**

```

CEMA/1/Report/UpdatedReport {"EMAEnergyinf o": {"threshold": 0, "priority": 1,
"currentValue": 320, "capacity": -320}, "SrcEMA": "SER VER EMA1", "respo
nseDescription": "OK", "requestID": 0, "service": "UpdatedReport", "EM
ATopologyinfo": { "emaCNT": 5}, "time": "Mon Jun 25 21:00:01 KST 2018
", "DestEMA": "1", "responseCode": 200}
    
```

**report Object{**

- “duration” : String,
- “reportRequestID” : Integer,
- “reportSpecifierID” : String,
- “reportName” : String,
- “createdDateTime” : Date,
- “reportDescription” : Array,

**}**

**reportDescription Object{**

- “rID” : String,
- “resourceID” : String,
- “deviceType” : String,
- “reportType” : String,
- “itemUnits” : String,
- “siScaleCode” : String,
- “marketContext” : String,
- “oadrMinPeriod” : String,
- “oadrMaxPeriod” : String,
- “oadrOnChange” : String,
- “itemDescription” : String,
- “powerAttributes” : Array,
- “qos” : String
- “state” : String,
- “power” : Double,
- “dimming” : Integer,
- “margin” : double,
- “generate” : double,
- “storage” : String,
- “maxValue” : Double,
- “minValue” : Double,
- “avgValue” : Double,
- “maxTime” : Date,
- “minTime” : Date,
- “priority” : Integer

**}**

**powerAttributes Object{**

- “hertz” : Double,
- “voltage” : Double,
- “ac” : Boolean

**}**

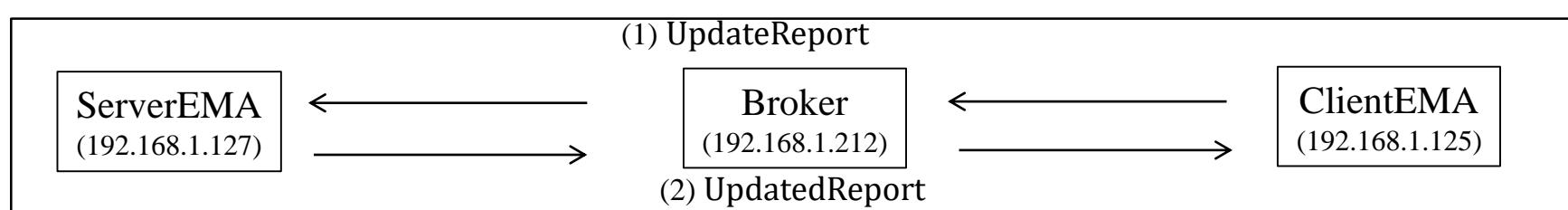
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)



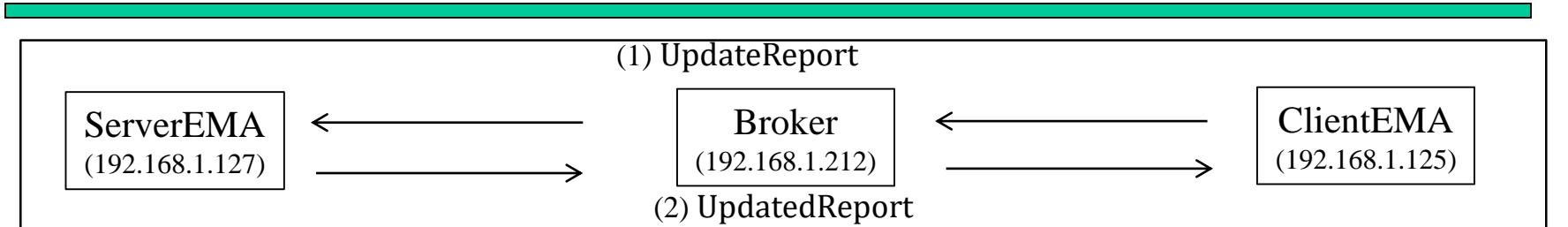
(2) UpdatedReport    MQTT    192.168.1.212    192.168.1.125    Publish Message [CEMA/1/oadrReport/UpdatedReport]

```
UpdatedReport Object{
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription": String,
    "EMAEnergyinfo": Object => 삭제
    "service": String,
    "time": Date
}
```

```
/UpdatedReport{"EMAEnergyinf o": {"threshold": 0, "priority": 1, "currentValue": 40
0, "capacity": -40 0}, "SrcEMA": "SER VER EMA1", "respo nseDescription": "OK", "requestID": 0, "service": "Up datedReport", "EM ATopologyinfo": { "topology": [{"de viceType": "LED", "dimming": 9, "dev iceEMAID": "104", "power": 80, "stat e": "ON", "priorit y": 0}, {"deviceTy pe": "LED", "dimmi ng": 9, "deviceEM
AID": "103", "power ": 80, "state": "ON ", "priority": 0}, {"deviceType": "L
ED", "dimming": 9, "deviceEMAID": "1 02", "power": 80, " state": "ON", "pri ority": 0}, {"deviceType": "LED", "d imming": 9, "devic eEMAID": "101", "p ower": 80, "state": "ON", "priority": 0}, {"deviceType": "LED", "dimming": 9, "deviceEMAID": "100", "power": 80, "state": "ON", "priority": 0}], "emaCNT": 5}, "time ": "Mon Jun 25
23 :46:51 KST 2018", "DestEMA": "1", " responseCode": 200}
```

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)



(1) UpdateReport      MQTT      192.168.1.125      192.168.1.212      Publish Message [SEMA/SERVER\_EMA/oadrReport/UpdateReport]

**UpdateReport Object{**

- “SrcEMA” : String,
- “DestEMA” : String,
- “requestID”: String,
- “reportType” : String,
- “EMAreisteredDRinformation” : Object => 변경
- “EMAreisteredMgnInformation” : Object => 변경,
- “report” : Array,
- “time”: Date,
- “service” : String,
- “type” : String (Explicit, Implicit인지 구분)

**}**

```

CEMA/1/Report/UpdatedReport {"EMAEnergyinfo": {"threshold": 0, "priority": 1,
"currentValue": 320, "capacity": -320}, "SrcEMA": "SER VER EMA1", "respo
nseDescription": "OK", "requestID": 0, "service": "UpdatedReport", "EM
ATopologyinfo": {"emaCNT": 5}, "time": "Mon Jun 25 2 1:00:01 KST 2018
", "DestEMA": "1", "responseCode": 200}
    
```

**report Object{**

- “duration” : String,
- “reportRequestID” : Integer,
- “reportSpecifierID” : String,
- “reportName” : String,
- “createdDateTime” : Date,
- “reportDescription” : Array,

**}**

**reportDescription Object{**

- “rID” : String,
- “resourceID” : String,
- “deviceType” : String,
- “reportType” : String,
- “itemUnits” : String,
- “siScaleCode” : String,
- “marketContext” : String,
- “oadrMinPeriod” : String,
- “oadrMaxPeriod” : String,
- “oadrOnChange” : String,
- “itemDescription” : String,
- “powerAttributes” : Array,
- “qos” : String
- “state” : String,
- “power” : Double,
- “dimming” : Integer,
- “margin” : double,
- “generate” : double,
- “storage” : String,
- “maxValue” : Double,
- “minValue” : Double,
- “avgValue” : Double,
- “maxTime” : Date,
- “minTime” : Date,
- “priority” : Integer

**}**

**powerAttributes Object{**

- “hertz” : Double,
- “voltage” : Double,
- “ac” : Boolean

**}**

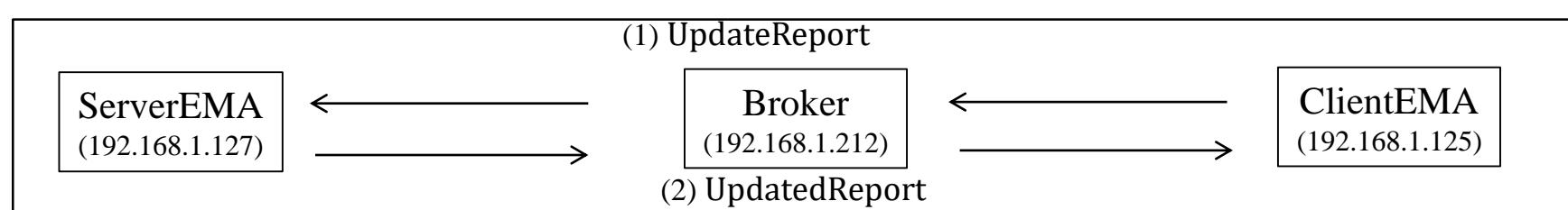
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)



(2) UpdatedReport    MQTT    192.168.1.212    192.168.1.125    Publish Message [CEMA/1/oadrReport/UpdatedReport]

```
UpdatedReport Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription" : String,
    "EMAEnergyinfo" : Object=>삭제
    "service": String,
    "time": Date
}
```

```
/UpdatedReport{"EMAEnergyinf o": {"threshold": 0, "priority": 1, "currentValue": 40
0, "capacity": -40 0}, "SrcEMA": "SER VER EMA1", "respo nseDescription": "OK", "requestID": 0, "service": "Up datedReport", "EM ATopologyinfo": { "topology": [{"de viceType": "LED", "dimming": 9, "dev iceEMAID": "104", "power": 80, "stat e": "ON", "priorit y": 0}, {"deviceTy pe": "LED", "dimmi ng": 9, "deviceEMA ID": "103", "power": 80, "state": "ON", "priorit y": 0}, {"deviceType": "LED", "dimming": 9, "deviceEMAID": "1 02", "power": 80, "state": "ON", "pri ority": 0}, {"deviceType": "LED", "d imming": 9, "devic eEMAID": "101", "p ower": 80, "state": "ON", "priority": 0}, {"deviceType": "LED", "dimming": 9, "deviceEMAID": "100", "power": 80, "state": "ON", "priority": 0}], "emaCNT": 5}, "time": "Mon Jun 25
23 :46:51 KST 2018", "DestEMA": "1", "responseCode": 200}
```

# **EMAP (CoAP & MQTT/JSON)**

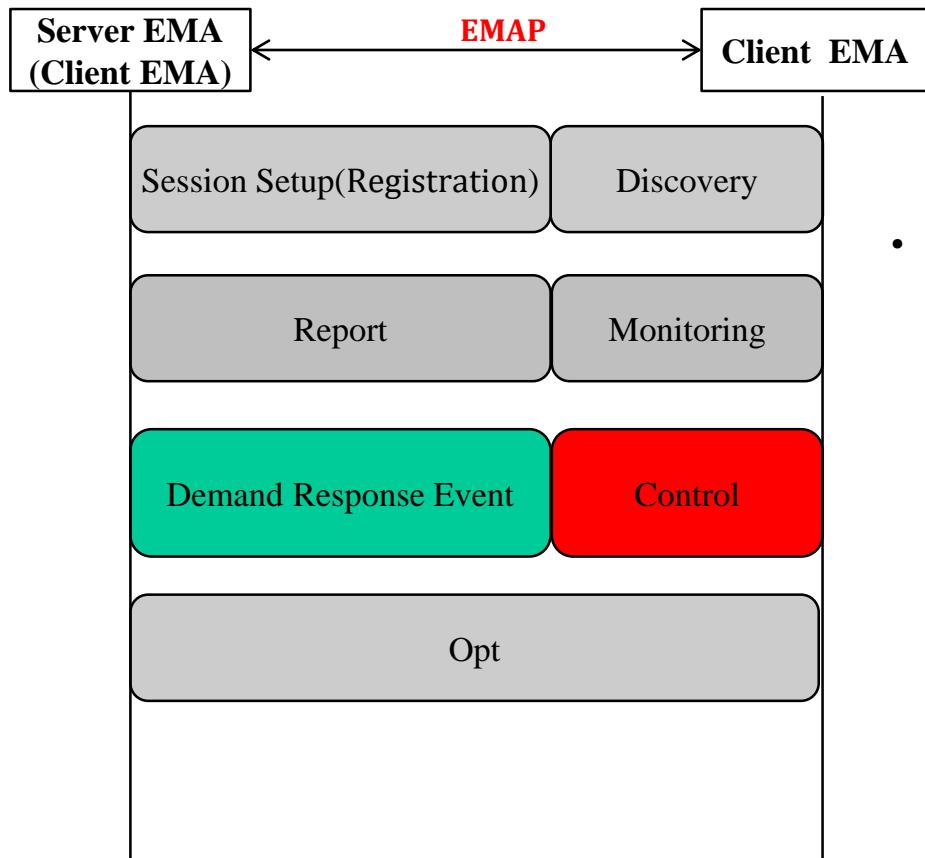
**Event**

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : EiEvent

#### EMAP

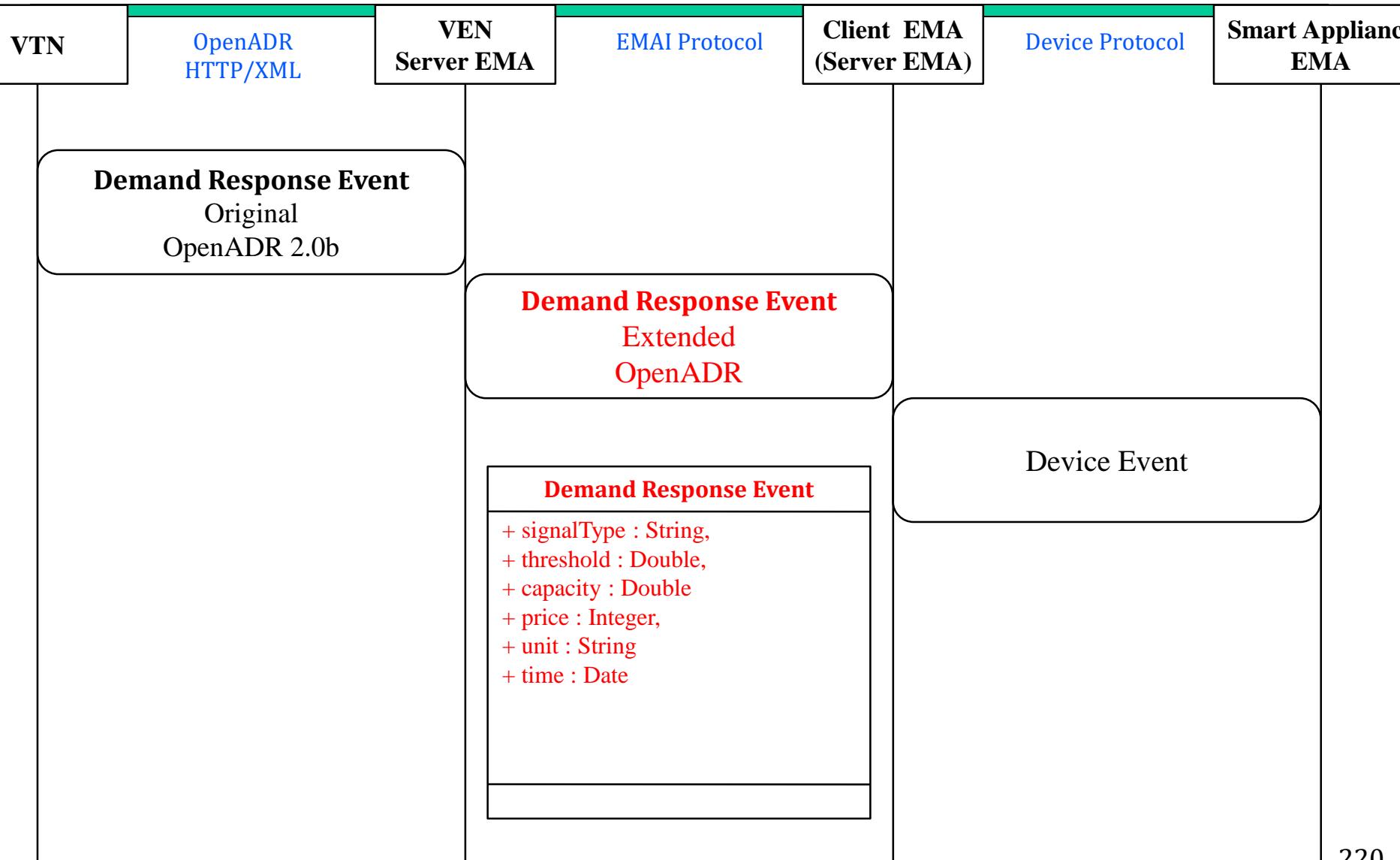
- EMA 사이의 수요반응 통신 프로토콜이며 OpenADR 2.0b의 모델링을 따랐으며 일부 모델링을 확장했고 Discovery와 Monitoring, Control 부분의 통신 프로토콜이 확장.



- Event(Demand Response Event)
  - 에너지관리에이전트 프로토콜은 다양한 이벤트 종류가 있다. 예로 가격기반 수요반응 이벤트에서 Initial Price, Incentive Price, Negotiation Price가 있다. 서버 에너지관리에이전트는 실 시간적으로 가격에 대한 정보를 이벤트로 발생
  - EMA가 최하위 Device 컨트롤 하기 위해서 Device를 총괄적으로 관리하는 Smart Appliances에게 Control 메시지를 전달.
  - 대표적인 예로 기존 OpenADR 2.0b의 marketContext에 price정보만 있지만, 하위 노드의 Price가 포함되어 있는 Local Price, Local Incentive가 포함되어 있음.

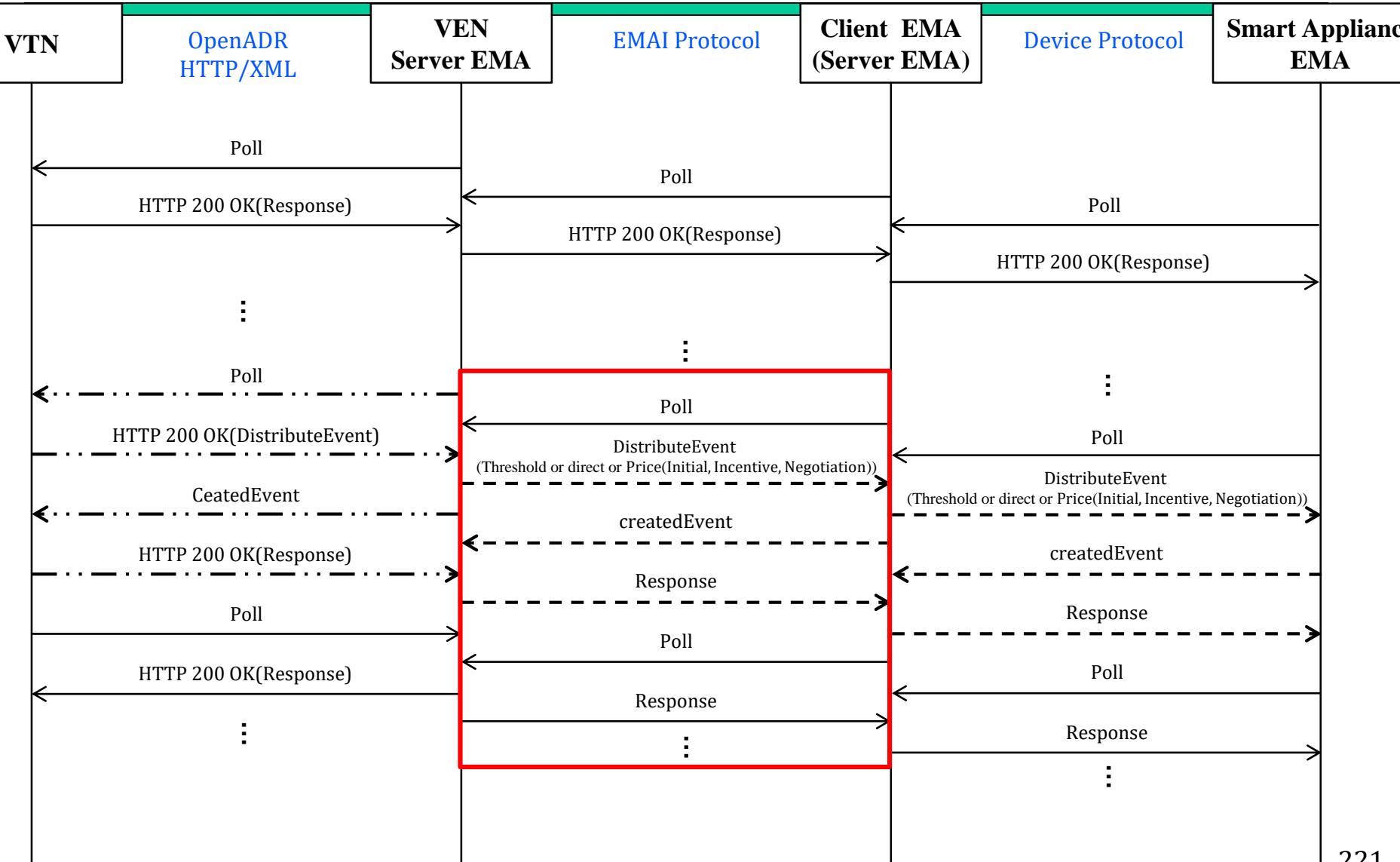
## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event



## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



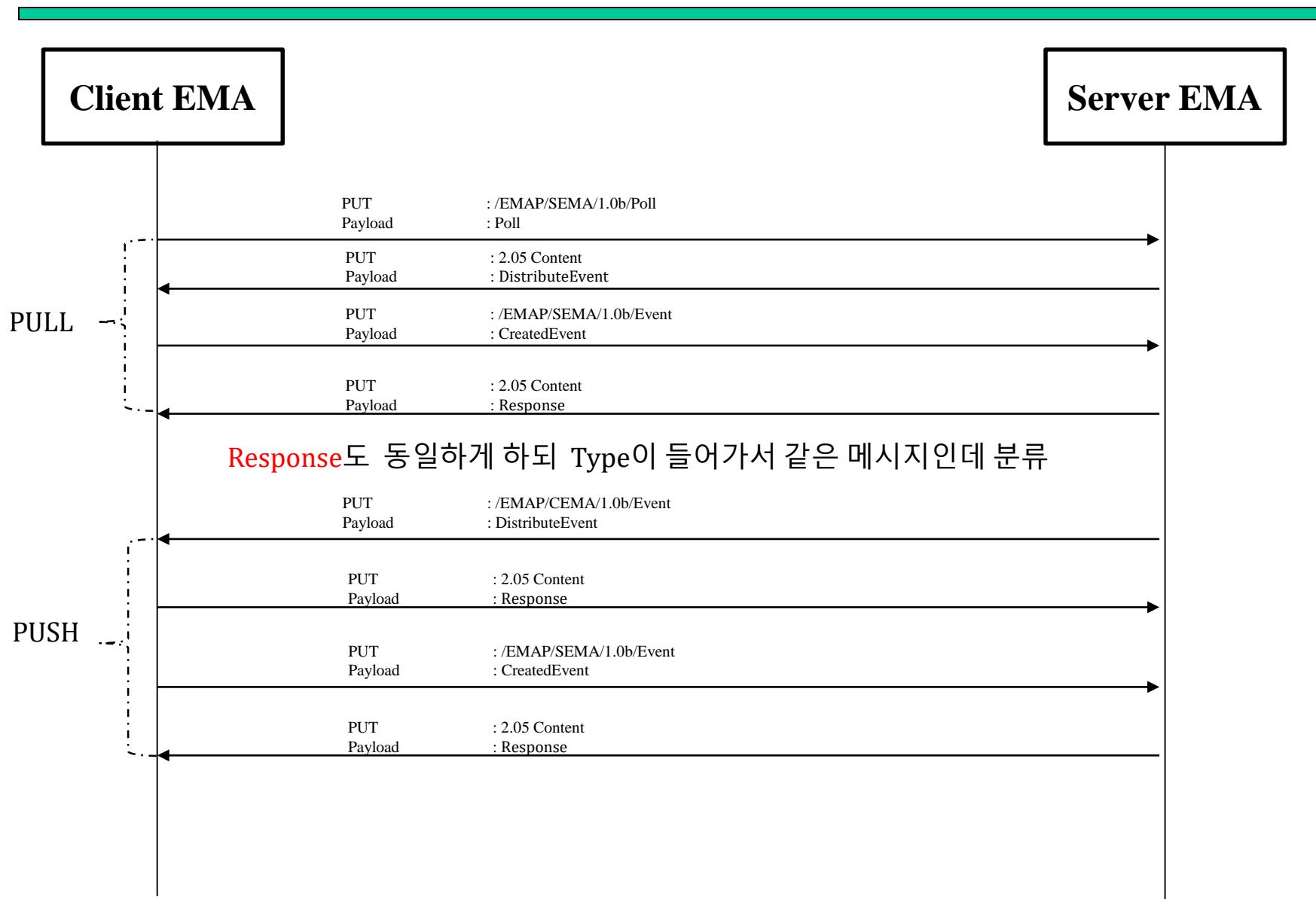
# **EMAP**

## **(3) Event\_PULL**

- **CoAP/JSON**
- **MQTT / JSON**

## 2.2 EMAP(CoAP/JSON)

### Service : Event



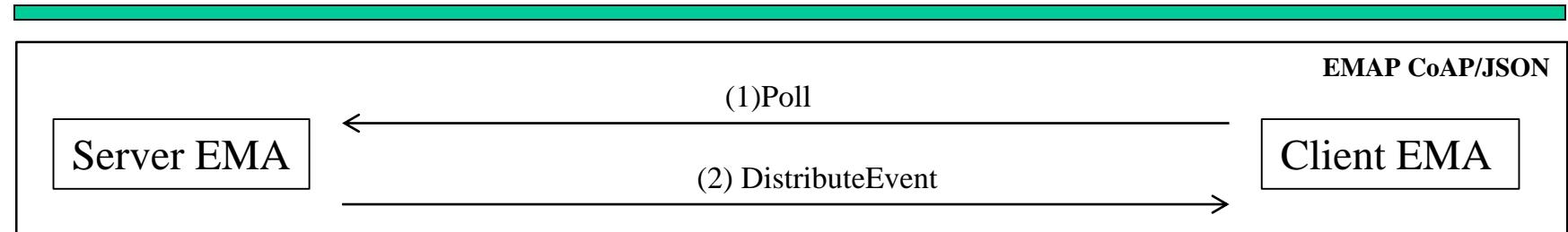
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(1) Poll

(2) DistributeEvent

Key Name

Reference

OpenADR 2.0b

SEP 2.0(IEC 61968)

SrcEMA

ei:venID

DestEMA

ei:vtnID

service

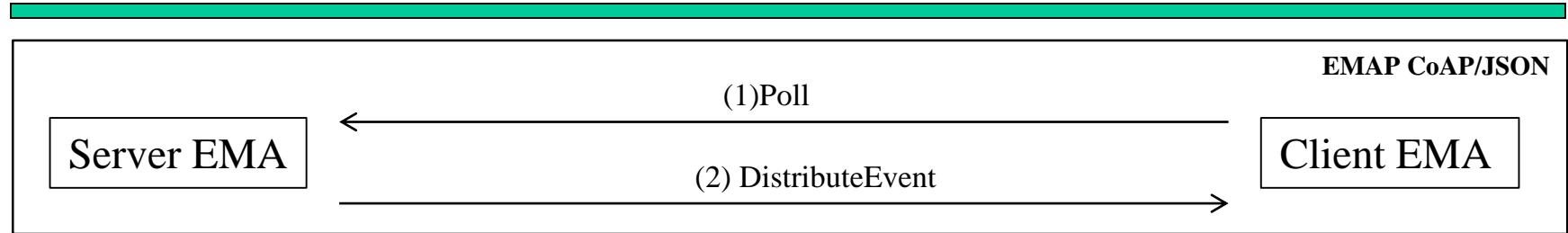
(Tag 0|름으로 존재)

time

RandomizableEvent:creation Time

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH

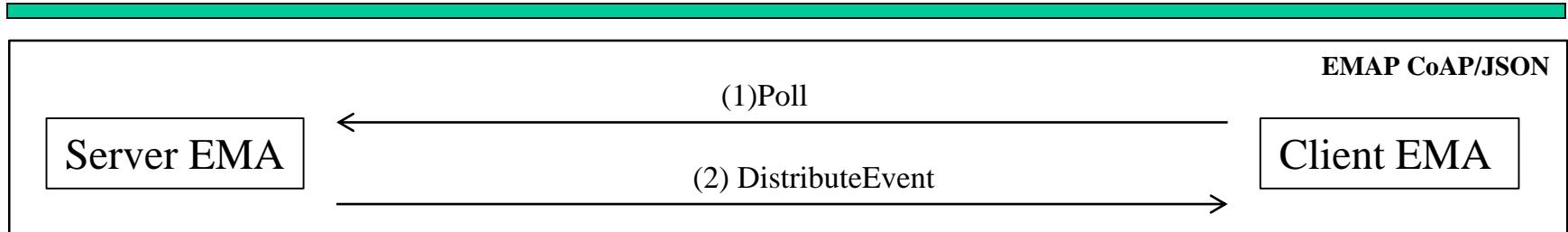


#### (2) DistributeEvent

Key Name		Reference	
		OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA		ei:vtnID	
DestEMA		ei:venID	
requestID		ei:requestID	
responseRequired		Ei:reponseRequired	
response	requestID	pyld:requestID	
	responseCode	ei:responseCode	
	responseDescription	ei:responseDescription	
	eventID	eventID	
	eventSignals	eventSignals	
	modificationNumber	modificationNumber	
	modificationReason	modificationReason	
	priority	priority	
	marketContext	eiMarketContext	
	createdDateTime	createdDateTime	
event	eventStatus	eventStatus	
	testEvent	testEvent	
	vtnComment	vtnComment	
	properties	properties	
	components	components	
	specificDestEMA	venID	
	dtStart	dtstart	
	Duration	duration	
	Tolerance	tolerance	
	notification	x-eiNotification	
signalType으로 Price Event 인지	rampUp	x-eiRampUp	
	Recovery	x-eiRecovery	
	eventSignal	eiEventSignal	
	Intervals	intervals	
	signalName	signalName	
	signalType	signalType	
	Control Event, Reserve Mode	oadrEvent:eiEventSignals:eiEventSig	
Price Event, Control Event, Reserve			

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH

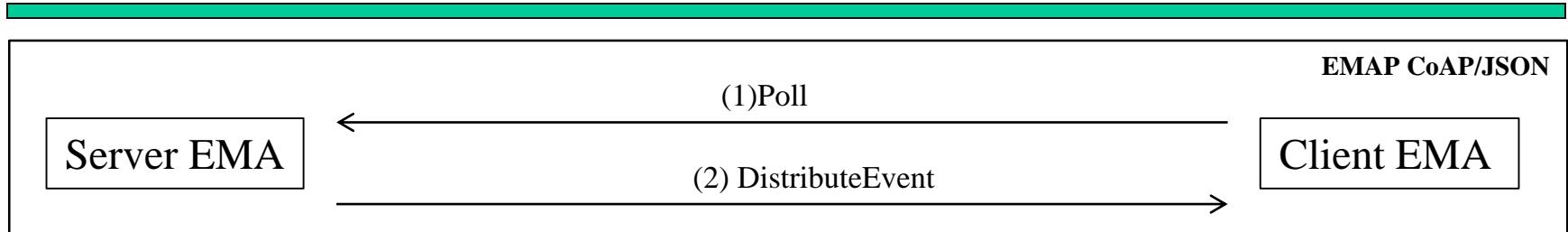


(1)Poll

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
service	type of service
time	service creation time

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH

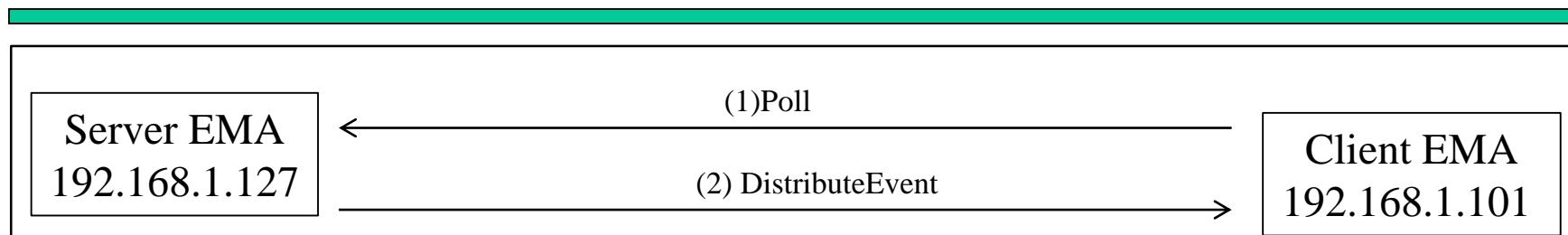


#### (2) DistributeEvent

Key name		Comments	
SrcEMA		source EMA identifier	
DestEMA		destination EMA identifier	
response	requestID	request identifier	
	responseCode	response code	
	responseDescription	description of response code	
	eventID	event identifier	
event	eventSignals	duration	event signal interval duration
		uid	event user id
		value	event value
		signalName	event signal name
		signalType	event signal type (bi direct, level)
		signalID	event signal ID
		currentValue	current usage value
		threshold	available amount of energy
		capacity	사용 가능량 (threshold - power)
		price	price of energy
		unit	단위
	modificationNumber	modification Number(count)	
	modificationReason	modification reason(event reason)	
	priority	priority	
	marketContext	market address(market reference)	
	createdDateTime	event create date & time	
	eventStatus	event status	
	testEvent	if event test or not	
	vtnComment		
	dtStart	event start time	
	duration	event duration	
	properties		
	components		
	specificDestEMA	specific target EMA	
	tolerance	tolerance duration	
	notification	notification duration	
	rampUp	ramp up duration	
	recovery		
responseRequired		response mandatory or not	
service		type of service	
time		service creation time	

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(1) Poll	MQTT	192.168.1.101	192.168.1.201	Publish Message [SEMA/SERVER_EMA1/Poll/Poll]
(2) DistributeEvent	MQTT	192.168.1.201	192.168.1.101	Publish Message [CEMA/1/Poll/DistributeEvent]

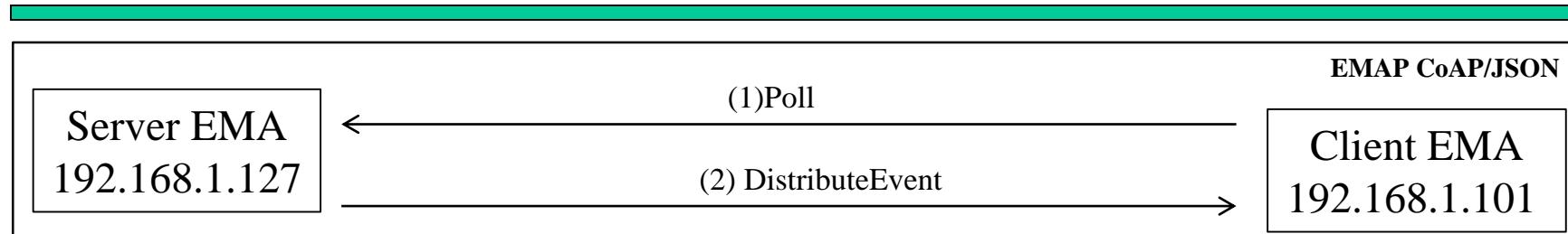
```

Poll JSONObject{
    "SrcEMA" : String,
    "DestEMA" : String,
    "version": Integer -> 삭제,
    "type": String -> 삭제(RegisteredReport로 변경)
    "service" : String,
    "time" : Date
}
  
```

- JavaScript Object Notation: application/json
- Object
  - ▷ Member Key: SrcEMA
  - ▷ Member Key: DestEMA
  - ▷ Member Key: requestID
  - ▷ Member Key: version
  - ▷ Member Key: type
    - String value: Registration
    - Key: type
  - ▷ Member Key: service
  - ▷ Member Key: time

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(1) Poll  
 (2) DistributeEvent

CoAP	192.168.1.101	192.168.1.127	CON, MID:25500, PUT, /Poll (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:25500, 2.05 Content (application/json)

DistributeEvent Object{  
 “SrcEMA” : String,  
 “DestEMA” : String,  
 “requestID” : String,  
 “response” : Array,  
 “event” : Array,  
 “responseRequired” : String,  
 “service” : String,  
 “time” : Date  
 }

response Array{  
 “requestID” : String,  
 “resultCode” : Integer,  
 “responseDescription” : String  
 }

```

// JavaScript Object Notation: application/json
Object
  Member Key: SrcEMA
  Member Key: responseDescription
  Member Key: requestID
  Member Key: service
  Member Key: EMADREventInformation
  Member Key: time
  Member Key: DestEMA
  Member Key: type
  Member Key: EMADRPriceInformation
  Member Key: responseCode
  
```

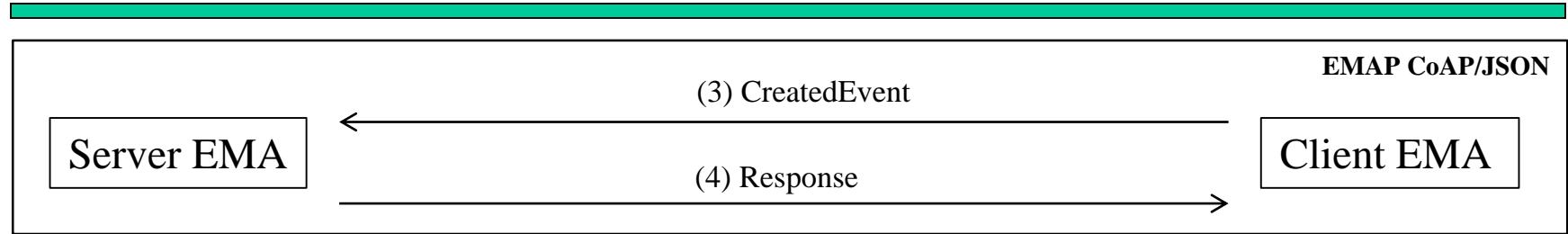
event Array{  
 “eventID” : String,  
 “eventSignals” : Array,  
 “modificationNumber” : Integer,  
 “modificationReason” : String,  
 “priority” : Integer,  
 “marketContext” : String,  
 “createdDataTime” : Date,  
 “eventStatus” : String,  
 “testEvent” : Boolean,  
 “vtnComment” : String,  
 “dtstart” : Date,  
 “duration” : String,  
 “properties” : String,  
 “components” : String,  
 “specificDestEMA” : String,  
 “tolerance” : String,  
 “notification” : String,  
 “rampUp” : String,  
 “recovery” : String  
 }

eventSignals Object{  
 “eventSignal” : String,  
 “intervals” : Array,  
 “signalName” : String,  
 “signalType” : String, (Price Event, Control Event, Reserve Mode, RealtimeDR인지 구분)  
 “signalID” : String,  
 “currentValue” : Double,  
 “threshold” : Double,  
 “capacity” : Double,  
 “price” : Integer,  
 “unit” : String,  
 }

intervals Array{  
 “duration” : String,  
 “uid” : Integer,  
 “value” : Double  
 }

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH

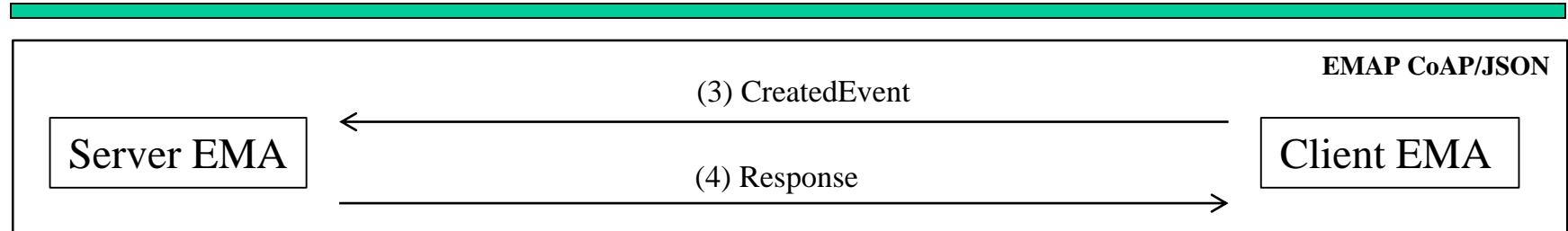


(3) CreatedEvent

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcMEA	ei:venID	
DestEMA	ei:vtnID	
responseCode	ei:eiResponse	Ei:responseCode
responseDescription		ei:responseDescription
optType	ei:eventResponse	ei:optType
eventID		ei:eventID
modificationNumber		ei:modificationNumber
requestID		pyld:requestID
service	(Tag 이름으로 존재)	
time		RandomizableEvent:creation Time

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH

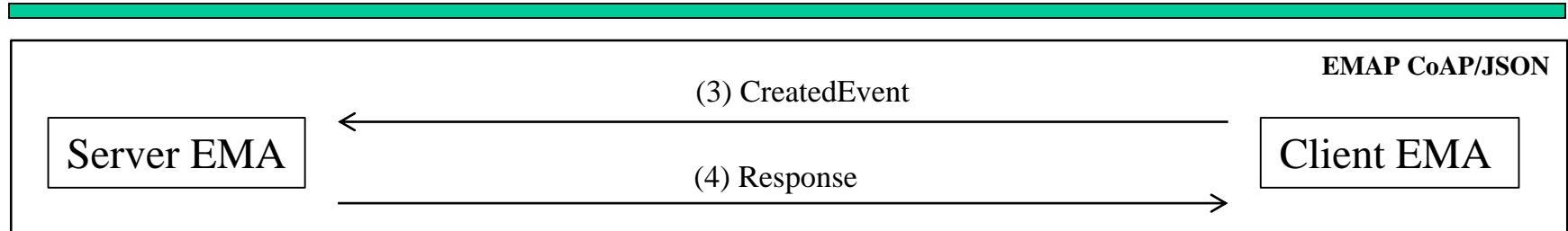


(4) Response

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
responseCode		ei:responseCode
responseDescription	ei:eiResponse	ei:responseDescription
requestID		Pyld:requestID
service	(Tag 이름으로 존재)	
time		RandomizableEvent:creationTime

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(3) CreatedEvent

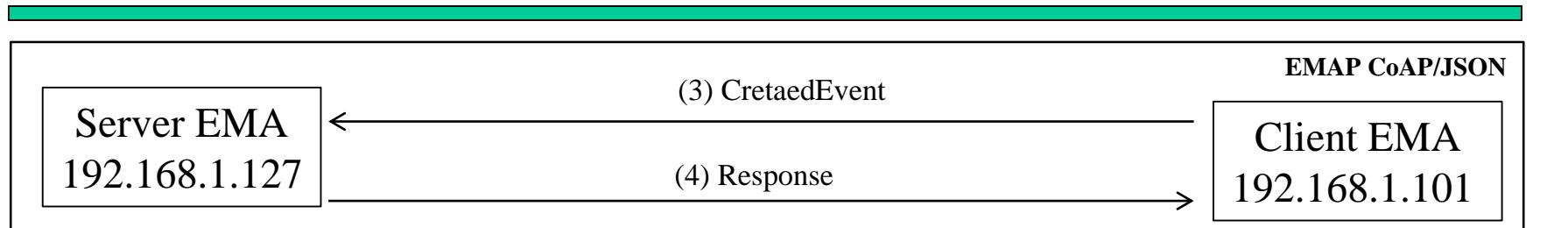
Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
eventID	Event identifier
modificationNumber	modification number(count)
optType	if participate event or not
service	type of service
time	service creation time

(4) Response

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	type of service
time	service creation time

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(3) CreatedEvent	CoAP	192.168.1.101	192.168.1.127	CON, MID:8526, PUT, /CreatedEvent (application/json)
(4) Response	CoAP	192.168.1.127	192.168.1.101	ACK, MID:8526, 2.05 Content (application/json)

```

CreatedEvent Object{
  "SrcEMA" : String,
  "DestEMA" : String,
  "requestID": String,
  "responseCode": Integer,
  "reponseDescription": String,
  "optType": String,
  "eventID" : String,
  "modificationNumber" : Integer,
  "service" : String,
  "type" : String,
  "time" : Date
}
  
```

▪ JavaScript Object Notation: application/json

- Object
  - ▷ Member Key: SrcEMA
  - ▷ Member Key: DestEMA
  - ▷ Member Key: requestID
  - ▷ Member Key: responseCode
  - ▷ Member Key: reponseDescription
  - ▷ Member Key: optType
  - ▷ Member Key: eventID
  - ▷ Member Key: modificationNumber
  - ▷ **Member Key: service**
  - Member Key: time
    - String value: 2018-04-18 10:47:15
    - Key: time

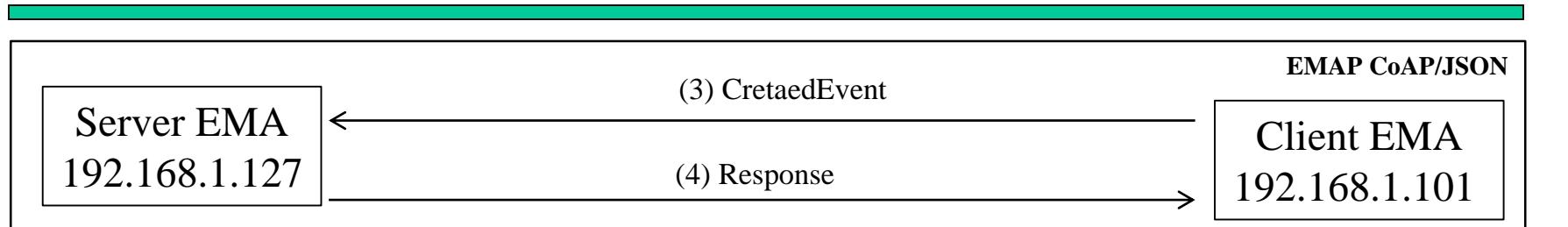
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(3) CreatedEvent	CoAP	192.168.1.101	192.168.1.127	CON, MID:8526, PUT, /CreatedEvent (application/json)
(4) Response	CoAP	192.168.1.127	192.168.1.101	ACK, MID:8526, 2.05 Content (application/json)

#### Response Object{

```
    "SrcEMA" : String,  
    "DestEMA" : String,  
    "requestID": String,  
    "responseCode" : Integer,  
    "responseDescription": String,  
    "version": Integer => 삭제,  
    "service": String,  
    "time" : Date  
}
```

- ↳ JavaScript Object Notation: application/json
- ↳ Object
  - ↳ Member Key: SrcEMA
  - ↳ Member Key: responseDescription
  - ↳ Member Key: requestID
  - ↳ Member Key: service
  - ↳ Member Key: time
  - ↳ Member Key: DestEMA
  - ↳ Member Key: version
  - ↳ Member Key: responseCode

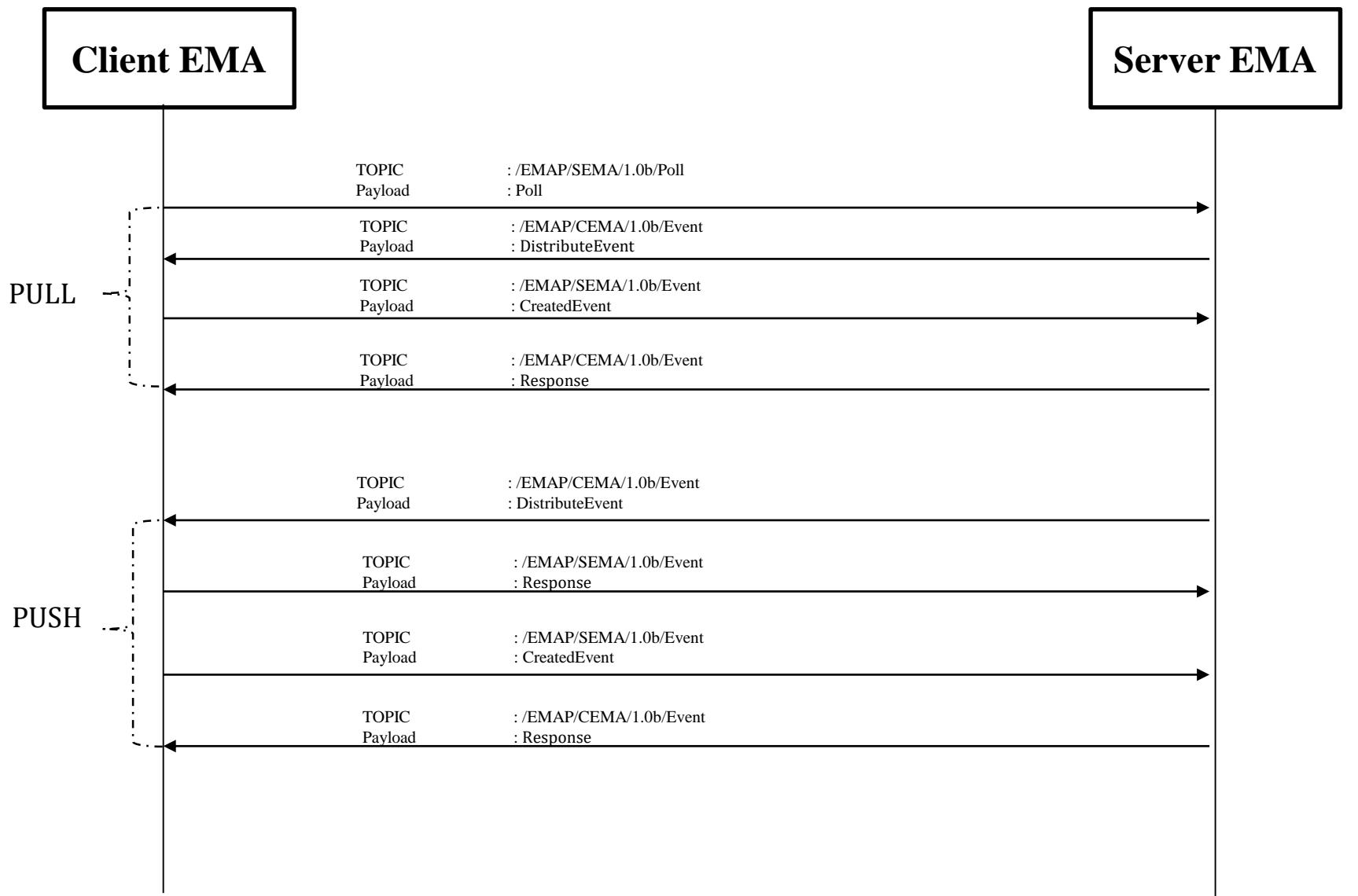
# **EMAP**

## **(3) Event\_PULL**

- **CoAP/JSON**
- **MQTT / JSON**

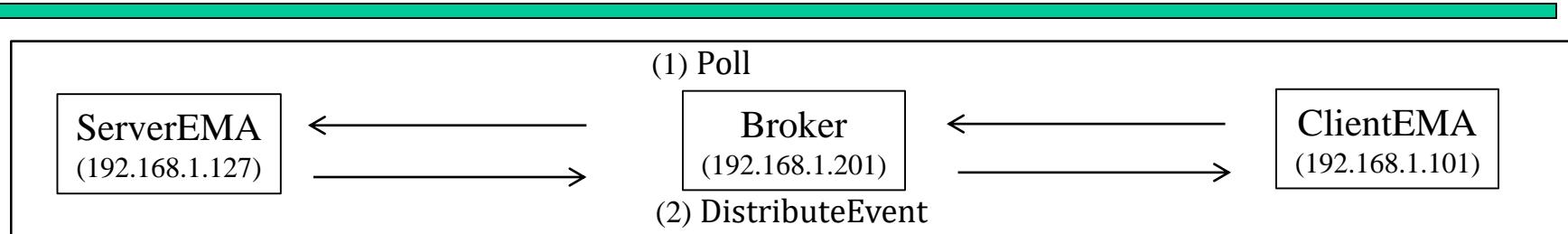
## 2.2 EMAP(MQTT/JSON)

### Service : Event



## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(1) Poll	MQTT	192.168.1.101	192.168.1.201	Publish Message [SEMA/SERVER_EMA1/Poll/Poll]
(2) DistributeEvent	MQTT	192.168.1.201	192.168.1.101	Publish Message [CEMA/1/Poll/DistributeEvent]

```
Poll JSONObject{
    "SrcEMA": String,
    "DestEMA": String,
    "requestID": String,
    "version": Integer >삭제,
    "type": String >삭제(RegisteredReport로 변경)
    "service": String,
    "type": String,
    "time": Date
}
```

type : Registration ,Periodic , report, Event, Price 인지 구분

Topic: SEMA/SERVER\_EMA1/Poll/Poll

Message: { "SrcEMA": "1", "DestEMA": "SERVER\_EMA1", "requestID": 189, "version": 4, "type": "Periodical", "service": "Poll", "time": "2018-06-27 05:31:08" }

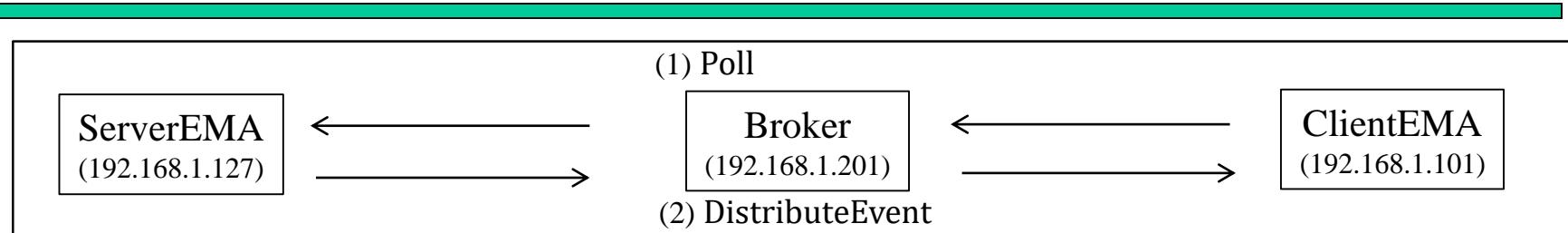
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(1) Poll	MQTT	192.168.1.101	192.168.1.201	Publish Message [SEMA/SERVER_EMA1/Poll/Poll]
(2) DistributeEvent	MQTT	192.168.1.201	192.168.1.101	Publish Message [CEMA/1/Poll/DistributeEvent]

#### DistributeEvent Object{

```

    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID" : String,
    "response" : Array,
    "event" : Array,
    "responseRequired" : String,
    "service" : String,
    "time" : Date
}

```

```

    response Array{
        "requestID" : String,
        "responseCode" : Integer,
        "responseDescription" : String
    }

```

```

DistributeEvent("SrcEMA": "SER VER EMA1", "respo nseDescription": "OK", "requestID": "189", "service": "DistributeEvent", "EMAPEventInfo rmation": [{"prof ileName": "EMAPro tocol", "eventID": 0, "signalName": "DREVENT", "endYM D": "20180627", "mar ketContext": 1, "t hreshold": 0, "pri ority": 0, "capaci ty": -198, "durati on": 1000, "uid": 0, "startYMD": 2018 0627, "testEvent": false, "eventSta tus": "DREVENT", "signalType": "Dis tributeEvent", "sig nalID": 1, "star tTime": 440, "endT ime": 441, "curr en tValue": 198, "mod ificationNumber": 0}], "time": "Wed Jun 27 14:31:07 KST 2018", "Dest EMA": "1", "type": "Level", "EMAPProfileInformation": [{"unit": "KW", "price": 1000}], "re sponseCode": 200)

```

#### event Array{

```

    "eventID" : String,
    "eventSignals" : Array,
    "modificationNumber" : Integer,
    "modificationReason" : String,
    "priority" : Integer,
    "marketContext" : String,
    "createdDataTime" : Date,
    "eventStatus" : String,
    "testEvent" : Boolean,
    "vtnComment" : String,
    "dtstart" : Date,
    "duration" : String,
    "properties" : String,
    "components" : String,
    "specificDestEMA" : String,
    "tolerance" : String,
    "notification" : String,
    "rampUp" : String,
    "recovery" : String
}

```

#### eventSignals Object{

```

    "eventSignal" : String,
    "intervals" : Array,
    "signalName" : String,
    "signalType" : String, (Price Event, Control Eve nt, Reserve
    Mode, RealtimeDR인지 구분)
    "signalID" : String,
    "currentValue" : Double,
    "threshold" : Double,
    "capacity" : Double,
    "price" : Integer,
    "unit" : String,
}

```

#### intervals Array{

```

    "duration" : String,
    "uid" : Integer,
    "value" : Double
}

```

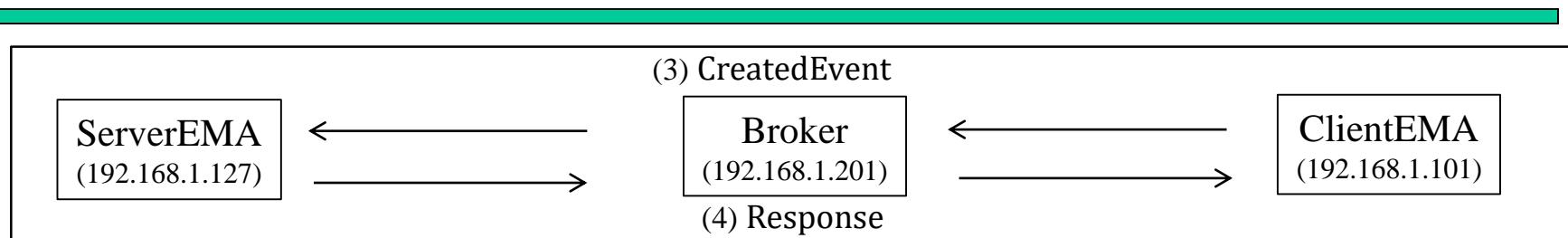
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(3) CreatedEvent	MQTT	192.168.1.101	192.168.1.201	Publish Message [SEMA/SERVER_EMA1/Poll/CreatedEvent]
(4) Response	MQTT	192.168.1.201	192.168.1.101	Publish Message [CEMA/1/Poll/Response]

#### CreatedEvent Object{

```
    "SrcEMA" : String,  
    "DestEMA" : String,  
    "requestID": String,  
    "responseCode": Integer,  
    "reponseDescription": String,  
    "optType": String,  
    "eventID" : String,  
    "modificationNumber" : Integer,  
    "service" : String,  
    "time" : Date  
}
```

Topic: SEMA/SERVER\_EMA1/Poll/CreatedEvent

Message [truncated]: { "SrcEMA": "1", "DestEMA": "SERVER\_EMA1", "requestID": 190, "responseCode": 200, "responseDescription": "OK", "optType": "OPT\_TYPE", "eventID": "1", "modificationNumber": 1, "service": "CreatedEvent", "time": "20

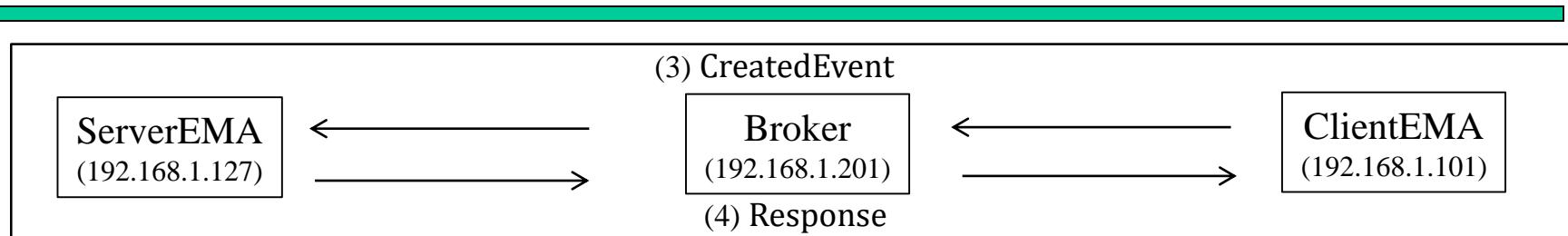
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(3) CreatedEvent	MQTT	192.168.1.101	192.168.1.201	Publish Message [SEMA/SERVER_EMA1/Poll/CreatedEvent]
(4) Response	MQTT	192.168.1.201	192.168.1.101	Publish Message [CEMA/1/Poll/Response]

```
Response Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode" : Integer,
    "responseDescription": String,
    "version": Integer // 삭제,
    "service": String,
    "time" : Date
}
```

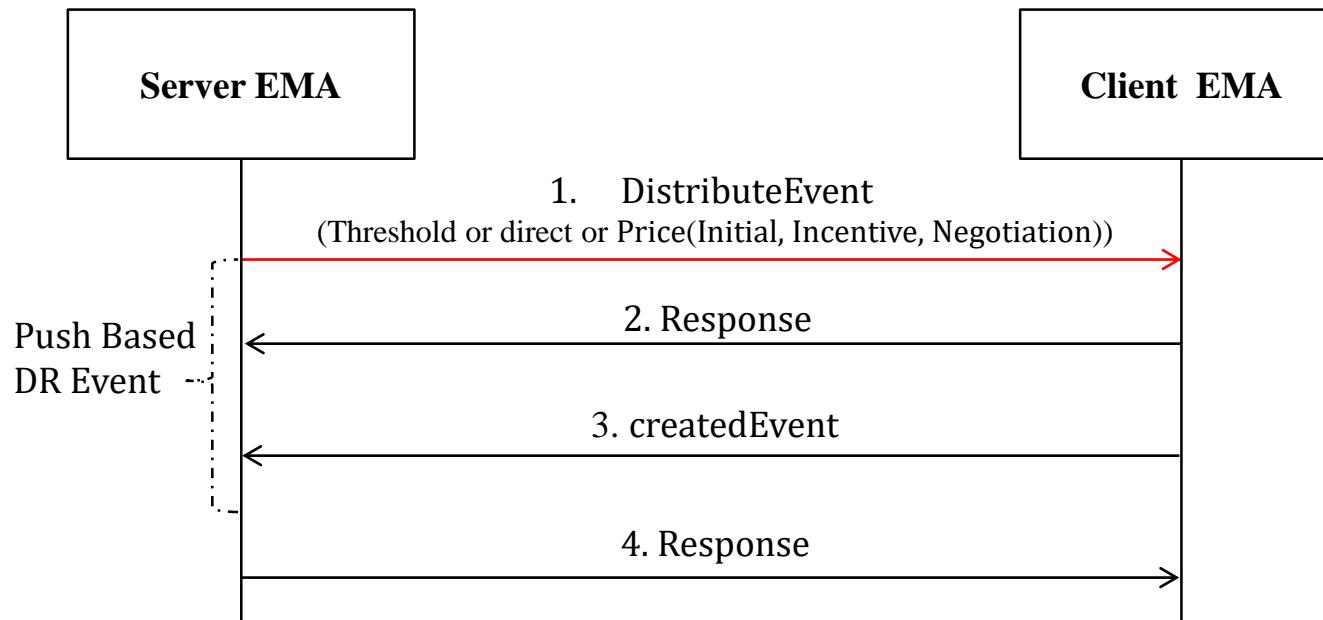
Topic: CEMA/1/Poll/Response

Message: {"SrcEMA": "SERVER\_EMA1", "responseDescription": "OK", "requestID": 190, "service": "Response", "time": "Wed Jun 27 14:31:07 KST 2018", "DestEMA": "1", "version": 0, "responseCode": 200}

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH

(2) Push 방식 – Event Based DR



Response도 동일하게 하되 Type을 넣어 분류

# **EMAP**

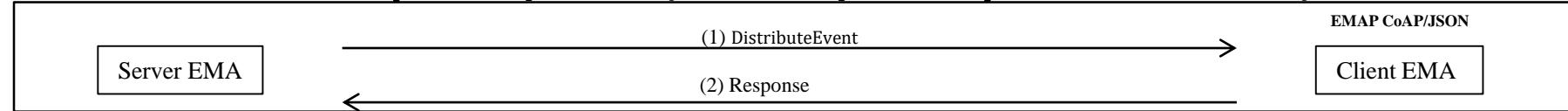
## **(3) Event\_PUSH**

- **CoAP/JSON**
- **MQTT / JSON**

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH

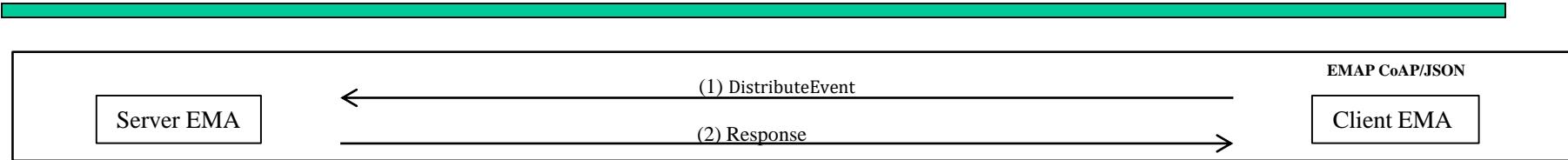


(1) DistributeEvent

Key Name		Reference	
		OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA		ei:vtID	
DestEMA		ei:venID	
requestID		ei:requestID	
responseRequired		Ei:reponseRequired	
response	requestID	ei:Response	pyld:requestID
	responseCode		ei:responseCode
	responseDescription		ei:responseDescription
	eventID		eventID
	<b>eventSignals</b>		eventSignals
	modificationNumber		modificationNumber
	modificationReason		modificationReason
	priority		priority
	marketContext		eiMarketContext
	createdDateTime		createdDateTime
event	eventStatus	oadrEvent:eiActivePeriod:eventDescriptor	eventStatus
	testEvent		testEvent
	vtIDComment		vtIDComment
	properties		properties
	components		components
	<b>specificDestEMA</b>		venID
	dtStart		dtstart
	Duration		duration
	Tolerance		tolerance
	notification		x-eiNotification
signalType으로 Price Event인지 Control Event, Reserve Mode, RealtimeDR 인지 구분한다	rampUp	oadrEvent:eiActivePeriod:properties	x-eiRampUp
	Recovery		x-eiRecovery
	eventSignal		eiEventSignal
	<b>intervals</b>		intervals
	signalName		signalName
	<b>signalType</b> (Price Event, Control Event, Reserve Mode, RealtimeDR)		signalType
	signalID		signalID
	currentValue		currentValue
	<b>threshold</b>		IdentifiedObject:DemandResponseProgram:availabilityUpdatePowerChangeThreshold
	<b>capacity</b>		AccountBalance/availableCredit

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH

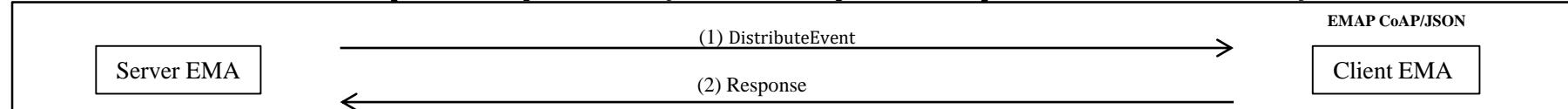


(2) Response

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtID	
DestEMA	ei:venID	
responseCode		ei:responseCode
responseDescription	ei:eiResponse	ei:responseDescription
requestID		Pyld:requestID
service	(Tag 이름으로 존재)	
type		TariffProfile:serviceCategoryKind:ServiceKind
time		RandomizableEvent:creationTime

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH

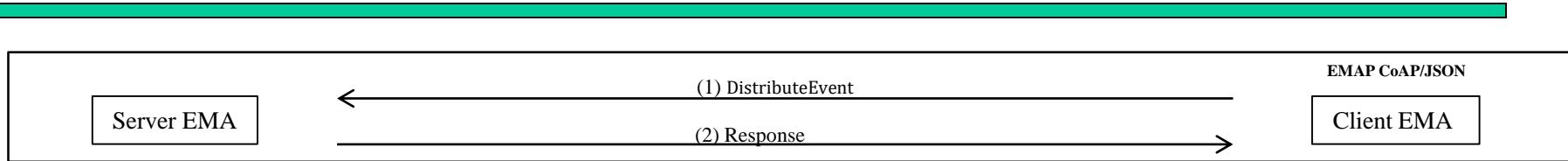


(1) DistributeEvent

Key name		Comments	
SrcEMA		source EMA identifier	
DestEMA		destination EMA identifier	
response	requestID	request identifier	
	responseCode	response code	
	responseDescription	description of response code	
	eventID	event identifier	
event	intervals	duration	event signal interval duration
		uid	event user id
		value	event value
	eventSignals	signalName	event signal name
		signalType	event signal type (bi direct, level)
		signalID	event signal ID
		currentValue	current usage value
		threshold	available amount of energy
		capacity	사용 가능량 (threshold - power)
		price	price of energy
		unit	단위
	modificationNumber	modification Number(count)	
	modificationReason	modification reason(event reason)	
	priority	priority	
	marketContext	market address(market reference)	
	createdDateTime	event create date & time	
	eventStatus	event status	
	testEvent	if event test or not	
	vtnComment		
	dtStart	event start time	
	duration	event duration	
	properties		
	components		
	specificDestEMA	specific target EMA	
	tolerance	tolerance duration	
	notification	notification duration	
	rampUp	ramp up duration	
	recovery		
responseRequired		response mandatory or not	
service		type of service	
time		service creation time	

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



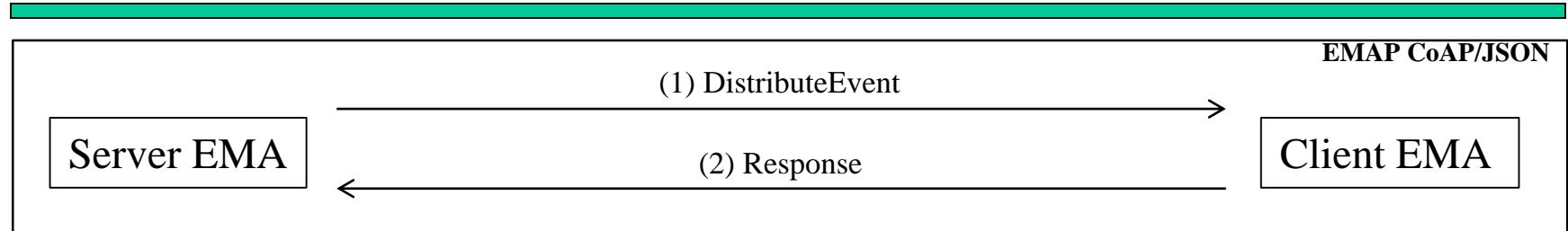
(2) Response

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	type of service
time	service creation time

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(1) DistributeEvent

#### DistributeEvent Object{

```
"SrcEMA" : String,  
"DestEMA" : String,  
"requestID" : String,  
"response" : Array,  
"event" : Array,  
"responseRequired" : String,  
"service" : String,  
"time" : Date  
}
```

```
response Array{  
    "requestID" : String,  
    "responseCode" : Integer,  
    "responseDescription" : String  
}
```

▪ JavaScript Object Notation: application/json  
  ▫ Object  
    ▫ Member Key: SrcEMA  
    ▫ Member Key: responseDescription  
    ▫ Member Key: requestID  
    ▫ Member Key: service  
    ▫ Member Key: EMADREventInformation  
    ▫ Member Key: time  
    ▫ Member Key: DestEMA  
    ▫ Member Key: type  
    ▫ Member Key: EMADRPriceInformation  
    ▫ Member Key: responseCode

#### event Array{

```
    "eventID" : String,  
    "eventSignals" : Array,  
    "modificationNumber" : Integer,  
    "modificationReason" : String,  
    "priority" : Integer,  
    "marketContext" : String,  
    "createdDataTime" : Date,  
    "eventStatus" : String,  
    "testEvent" : Boolean,  
    "vtnComment" : String,  
    "dtstart" : Date,  
    "duration" : String,  
    "properties" : String,  
    "components" : String,  
    "specificDestEMA" : String,  
    "tolerance" : String,  
    "notification" : String,  
    "rampUp" : String,  
    "recovery" : String  
}
```

#### eventSignals Object{

```
    "eventSignal" : String,  
    "intervals" : Array,  
    "signalName" : String,  
    "signalType" : String, (Price Event, Control Event, Reserve Mode, RealtimeDR인지 구분)  
    "signalID" : String,  
    "currentValue" : Double,  
    "threshold" : Double,  
    "capacity" : Double,  
    "price" : Integer,  
    "unit" : String,  
}
```

#### intervals Array{

```
    "duration" : String,  
    "uid" : Integer,  
    "value" : Double  
}
```

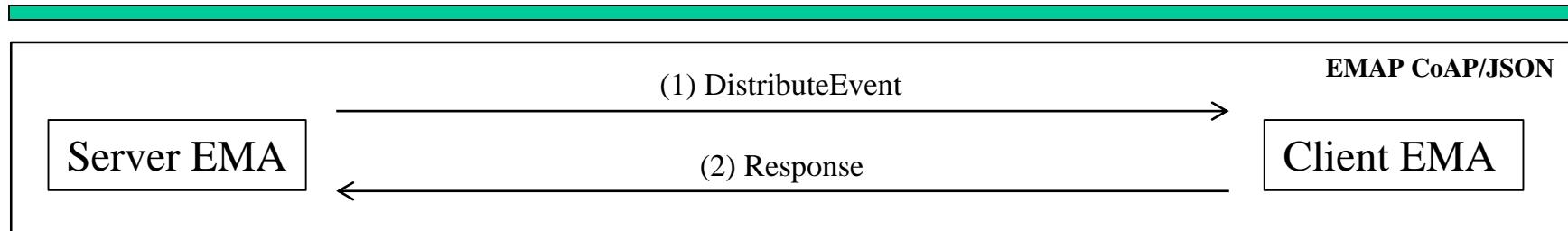
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



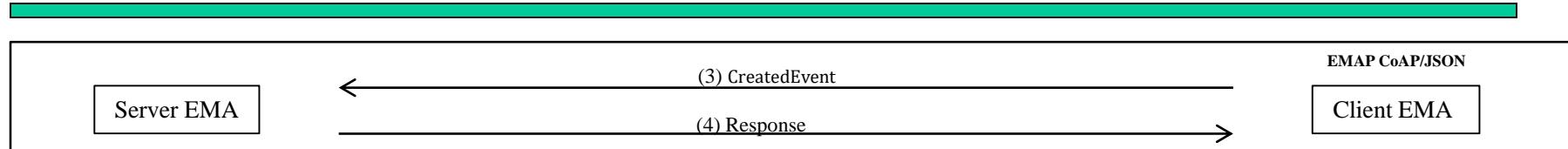
(2) Response

```
Response Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode" : Integer,
    "responseDescription": String,
    "service" : String,
    "time" : Date
}
```

```
Object
    ▷ Member Key: SrcEMA
    ▷ Member Key: DestEMA
    ▷ Member Key: requestID
    ▷ Member Key: version
    ▷ Member Key: responseCode
    ▷ Member Key: responseDescription
    ▷ Member Key: type
    ▷ Member Key: service
    ▷ Member Key: time
        String value: 2018-05-01 03:35:46
        Key: time
```

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH

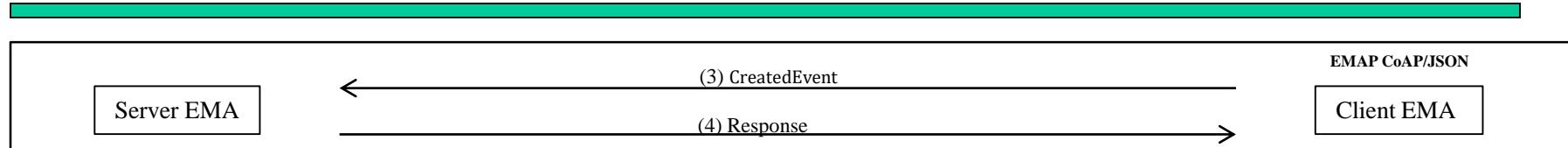


(3) CreatedEvent

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcMEA	ei:venID	
DestEMA	ei:vtnID	
responseCode	ei:eiResponse	Ei:responseCode
responseDescription		ei:responseDescription
optType	ei:eventResponse	ei:optType
eventID		ei:eventID
modificationNumber		ei:modificationNumber
requestID		pyld:requestID
service	(Tag 이름으로 존재)	
time		RandomizableEvent:creation Time

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH

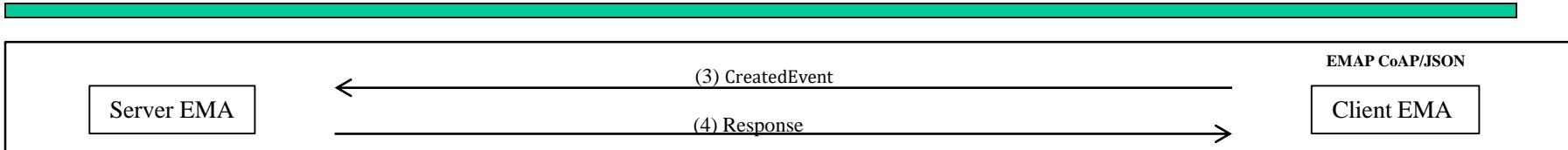


(4)Response

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
responseCode	ei:responseCode	
responseDescription	ei:eiResponse	ei:responseDescription
requestID		Pyld:requestID
service	(Tag 이름으로 존재)	
time		RandomizableEvent:creationTime

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(3) CreatedEvent

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
eventID	Event identifier
modificationNumber	modification number(count)
optType	if participate event or not
service	type of service
time	service creation time

(4)Response

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	type of service
time	service creation time

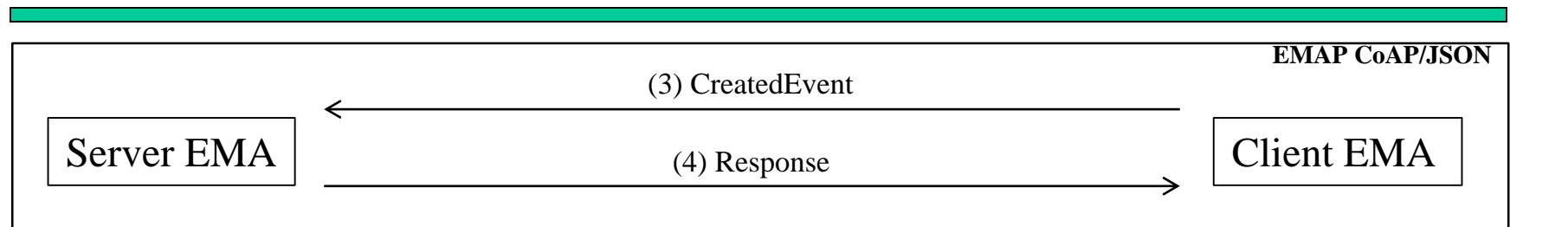
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(3) CreatedEvent  
(4) Response

CoAP	192.168.1.101	192.168.1.127	CON, MID:17396, PUT, /CreatedEvent (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:17396, 2.05 Content (application/json)

**CreatedEvent Object{**

“SrcEMA” : String,  
“DestEMA” : String,  
“requestID”: String,  
“responseCode”: Integer,  
“reponseDescription”: String,  
“optType”: String,  
“eventID” : String,  
“modificationNumber” : Integer,  
“service” : String,  
“time” : Date  
}

```
JavaScript Object Notation: application/json
  Object
    ▷ Member Key: SrcEMA
      String value: 1
      Key: SrcEMA
    ▷ Member Key: DestEMA
    ▷ Member Key: requestID
    ▷ Member Key: responseCode
    ▷ Member Key: reponseDescription
    ▷ Member Key: optType
    ▷ Member Key: eventID
    ▷ Member Key: modificationNumber
      Number value: 1
      Key: modificationNumber
    ▷ Member Key: service
    ▷ Member Key: time
      String value: 2018-05-01 03:35:46
      Key: time
```

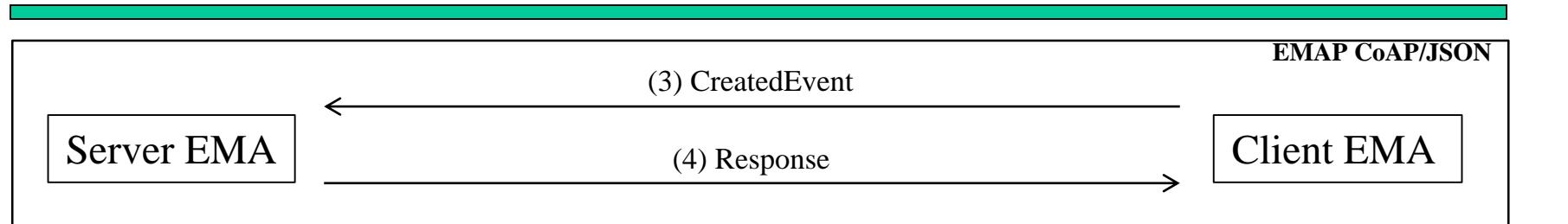
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(3) CreatedEvent  
(4) Response

CoAP	192.168.1.101	192.168.1.127	CON, MID:17396, PUT, /CreatedEvent (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:17396, 2.05 Content (application/json)

#### Response Object{

```
"SrcEMA" : String,  
"DestEMA" : String,  
"requestID": String,  
"responseCode" : Integer,  
"responseDescription": String,  
"service": String,  
"time" : Date  
}
```

```
Object  
  ▷ Member Key: SrcEMA  
  ▷ Member Key: responseDescription  
  ▷ Member Key: requestID  
  ▷ Member Key: service  
    String value: Response  
    Key: service  
  ▷ Member Key: time  
    String value: Tue May 01 12:35:29 KST 2018  
    Key: time  
  ▷ Member Key: DestEMA  
  ▷ Member Key: version  
  ▷ Member Key: responseCode  
    Number value: 200  
    Key: responseCode
```

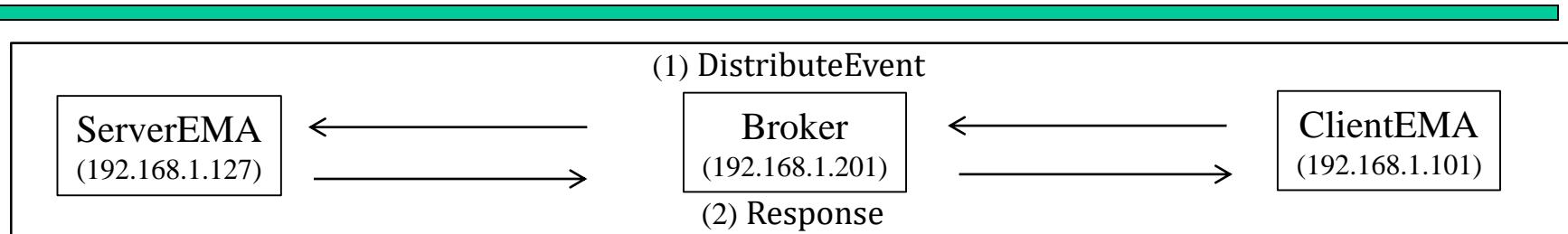
# **EMAP**

## **(3) Event\_PUSH**

- **CoAP/JSON**
- **MQTT / JSON**

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(1) DistributeEvent  
 (2) Response

#### DistributeEvent Object{

```

    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID" : String,
    "response" : Array,
    "event" : Array,
    "responseRequired" : String,
    "service" : String,
    "time" : Date
}
    
```

**response Array{**

- “requestID” : String,
- “responseCode” : Integer,
- “responseDescription” : String

**}**

```

DistributeEvent("SrcEMA": "SER VER EMA1", "respo nseDescription": "OK", "requestID": "189", "service": "DistributeEvent", "EMADREventInfo rmation": [{"prof ileName": "EMAPro tocol1", "eventID": 0, "signalName": "DREVENT", "endYM D": "20180627", "mar ketContext": 1, "t hreshold": 0, "pri ority": 0, "capaci ty": -198, "durati on": 1000, "uid": 0, "startYMD": "2018 0627", "testEvent": false, "eventSta tus": "DREVENT", "signalType": "Dis tributeEvent", "sig nalID": 1, "star tTime": 440, "endT ime": 441, "curren tValue": 198, "mod ificationNumber": 0}], "time": "Wed Jun 27 14:31:07 KST 2018", "Dest EMA": "1", "type": "Level", "EMADRProfileInformation": [{"unit": "KW", "price": 1000}], "re sponseCode": 200)
    
```

#### event Array{

```

    "eventID" : String,
    "eventSignals" : Array,
    "modificationNumber" : Integer,
    "modificationReason" : String,
    "priority" : Integer,
    "marketContext" : String,
    "createdDataTime" : Date,
    "eventStatus" : String,
    "testEvent" : Boolean,
    "vtnComment" : String,
    "dtstart" : Date,
    "duration" : String,
    "properties" : String,
    "components" : String,
    "specificDestEMA" : String,
    "tolerance" : String,
    "notification" : String,
    "rampUp" : String,
    "recovery" : String
}
    
```

#### eventSignals Object{

```

    "eventSignal" : String,
    "intervals" : Array,
    "signalName" : String,
    "signalType" : String, (Price Event, Control Eve nt, Reserve
    Mode, RealtimeDR인지 구분)
    "signalID" : String,
    "currentValue" : Double,
    "threshold" : Double,
    "capacity" : Double,
    "price" : Integer,
    "unit" : String,
}
    
```

#### intervals Array{

```

    "duration" : String,
    "uid" : Integer,
    "value" : Double
}
    
```

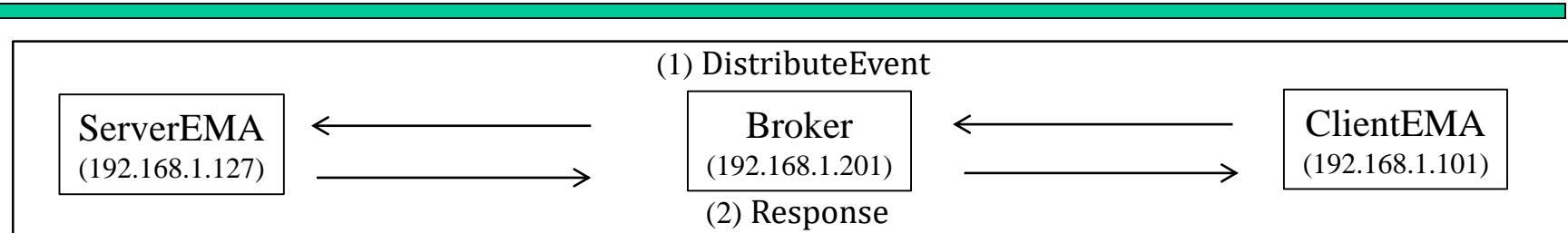
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



- (1) DistributeEvent  
(2) Response

```
Response Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “requestID”: String,
    “responseCode” : Integer,
    “responseDescription”: String,
    “service” : String,
    “time” : Date
}
```

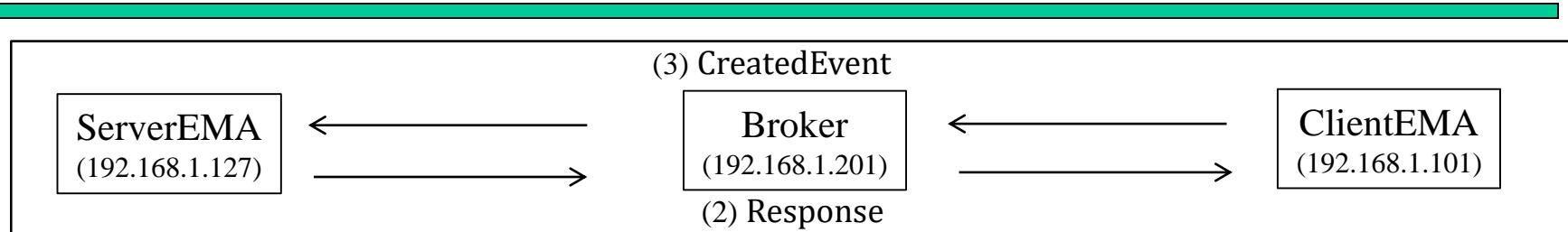
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



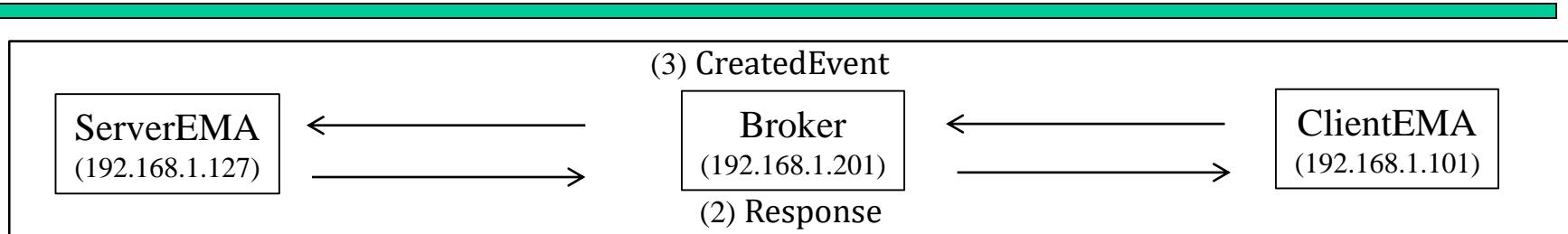
(3) CreatedEvent  
(4) Response

```
CreatedEvent Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode": Integer,
    "reponseDescription": String,
    "optType": String,
    "eventID" : String,
    "modificationNumber" : Integer,
    "service" : String,
    "time" : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(3) CreatedEvent  
(4) Response

```
Response Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode" : Integer,
    "responseDescription": String,
    "service": String,
    "time" : Date
}
```

# **EMAP (CoAP & MQTT/JSON)**

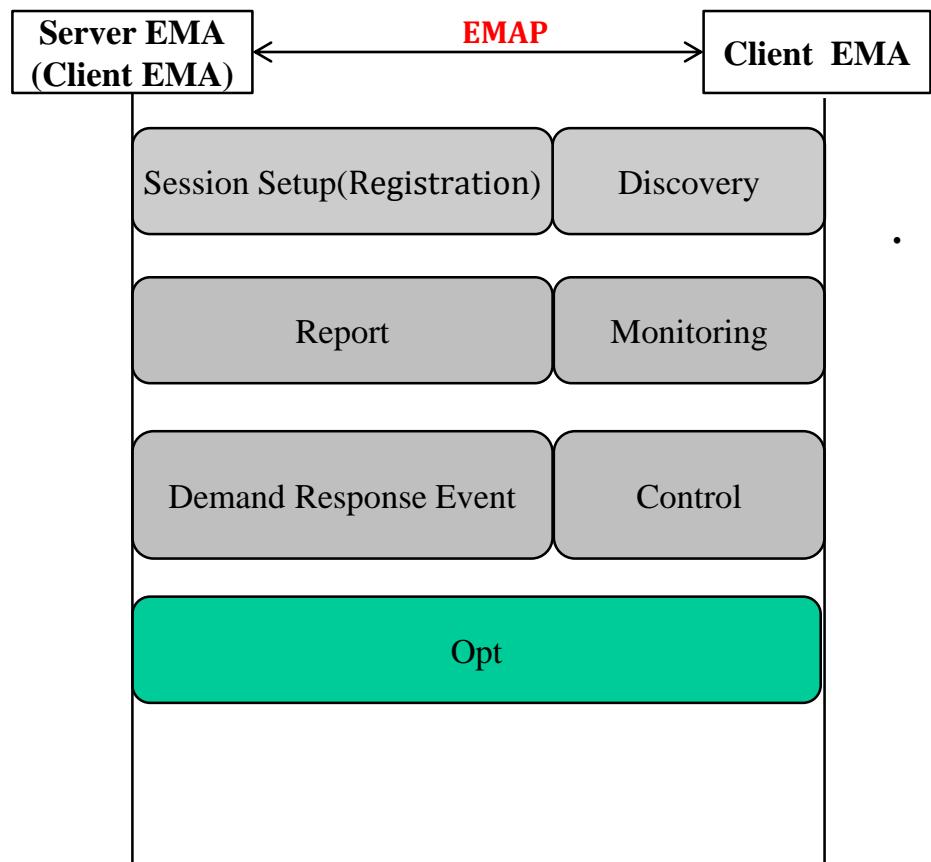
**Opt**

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : EiEvent

#### EMAP

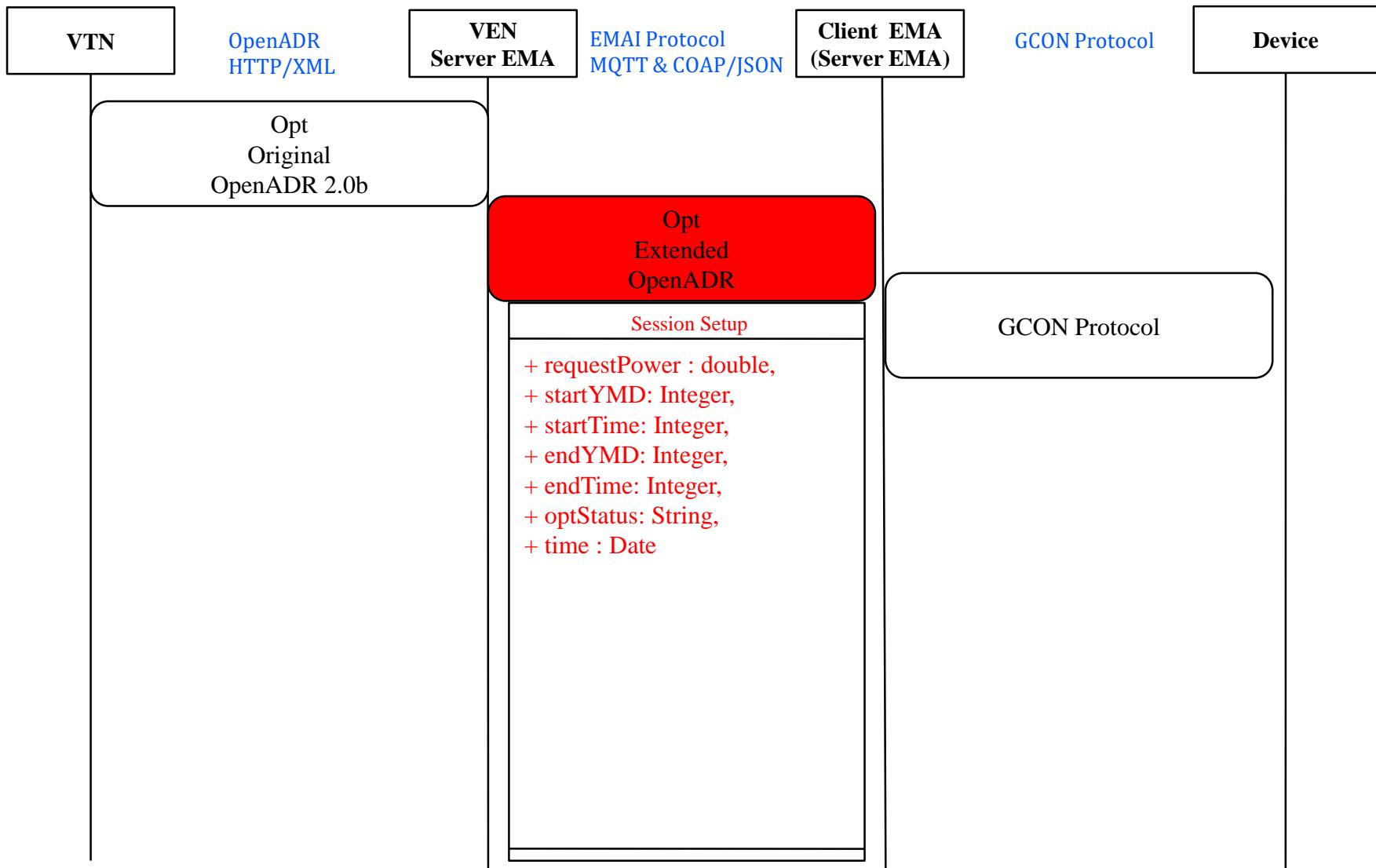
- EMA 사이의 수요반응 통신 프로토콜이며 OpenADR 2.0b의 모델링을 따랐으며 일부 모델링을 확장했고 Discovery와 Monitoring, Control 부분의 통신 프로토콜이 확장.



- Opt
  - 에너지 관리 에이전트 프로토콜의 Opt는 클라이언트 에너지 관리 에이전트가 상위 서버 에너지 관리 에이전트에게 수요 반응 이벤트의 가용 상태 또는 수요 반응 이벤트 프로그램 변경, 수요 반응 스케줄링을 요청하는 서비스.

## 2.2 EMAP(MQTT, CoAP/JSON)

### Service : Opt



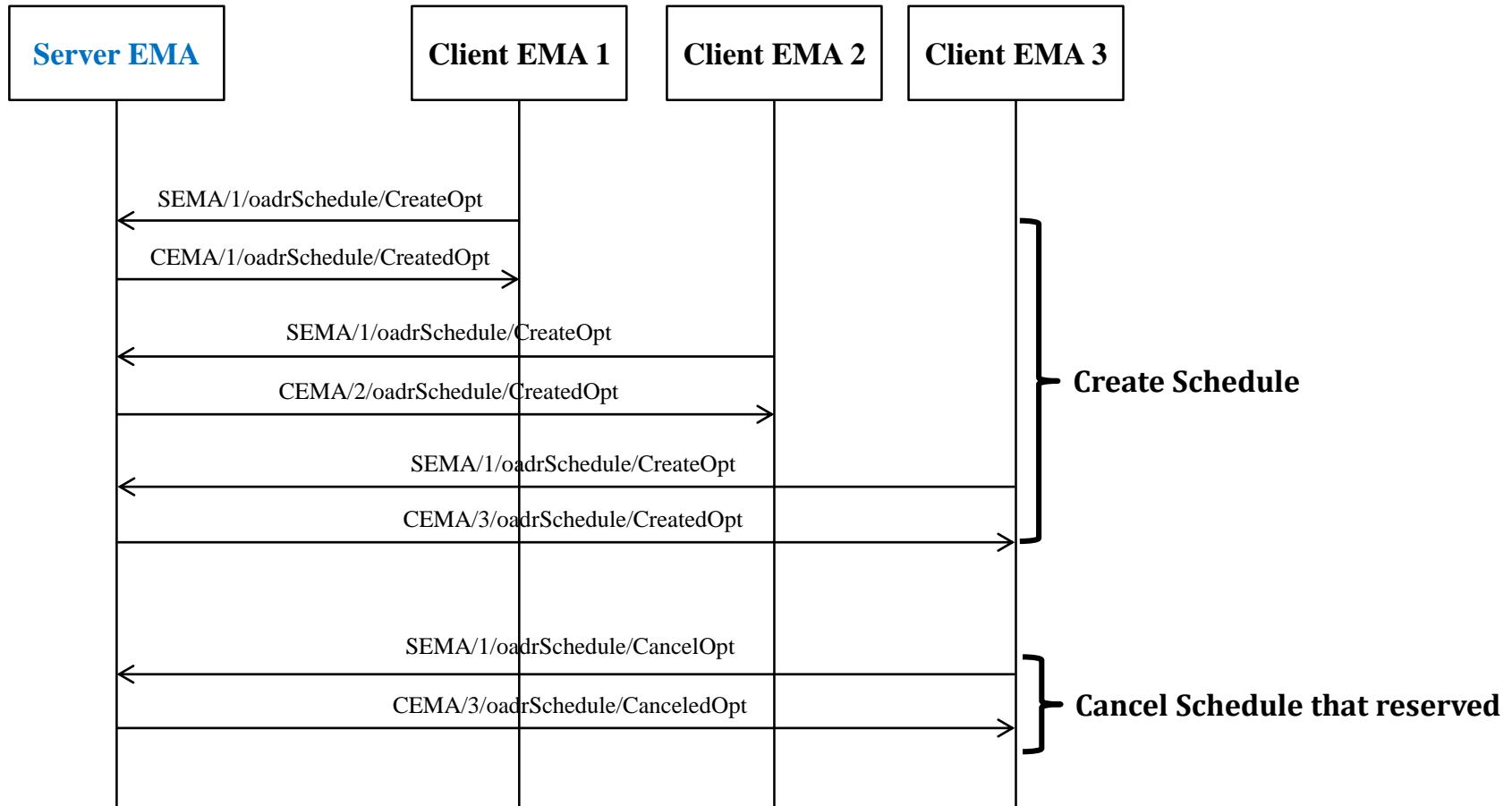
# **EMAP**

## **(4) Opt**

- **CoAP/JSON**
- **MQTT / JSON**

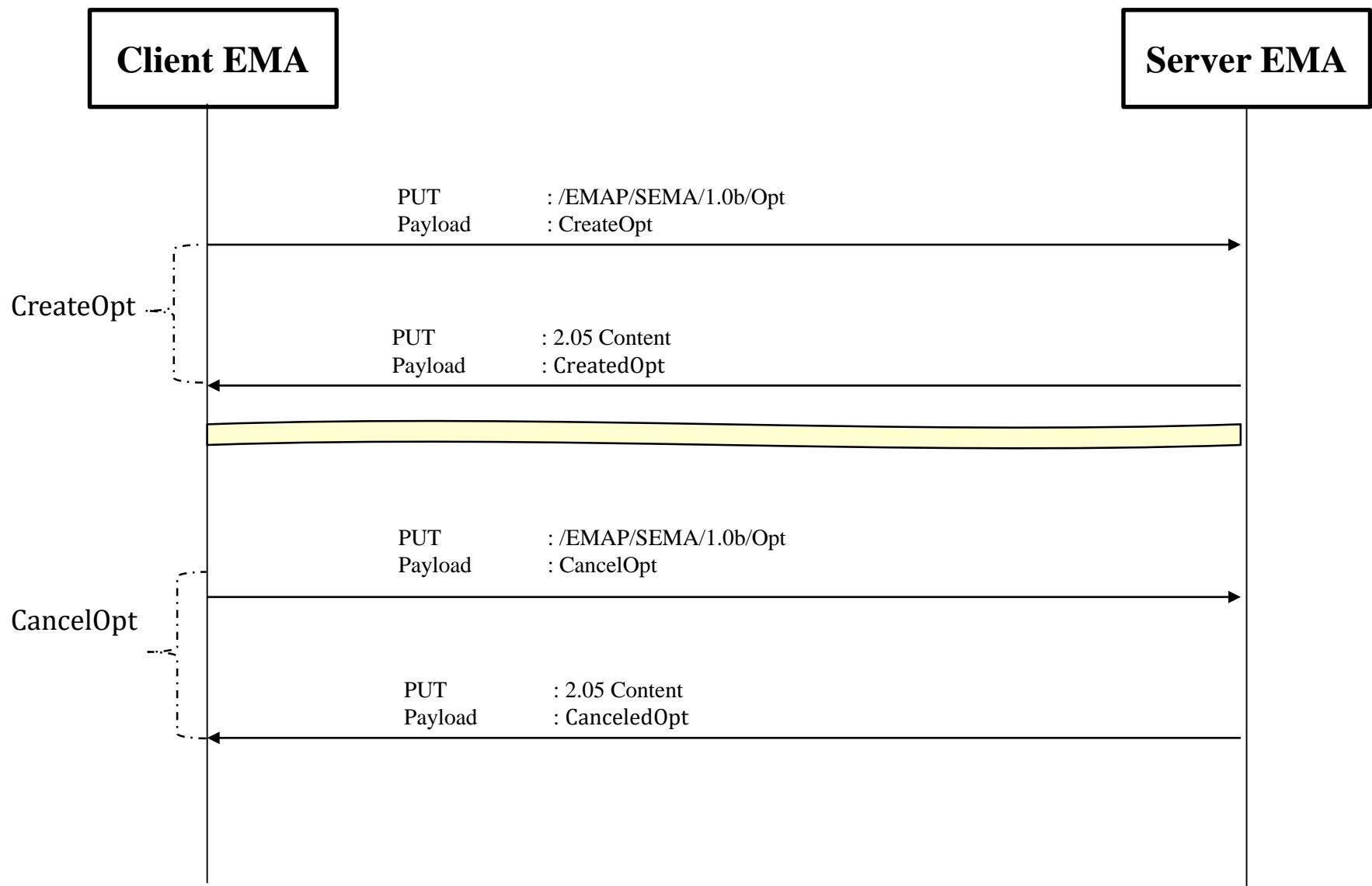
## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Schedule by using Opt



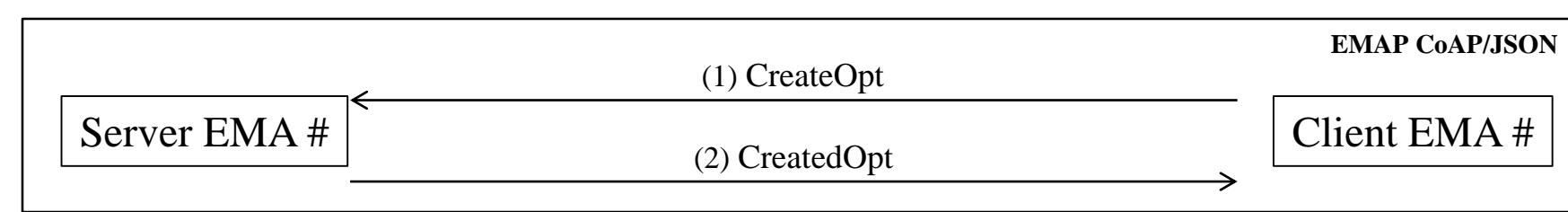
## 2.2 EMAP(CoAP/JSON)

### Service : Opt



# 2. Smart Home Energy Framework :

## 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Schedule (Opt)

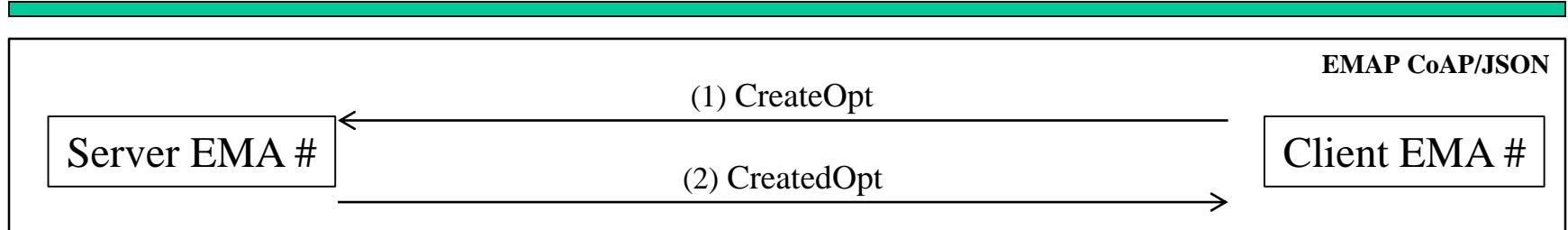


(1) CreateOpt

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
optID	ei:qualifiedEventID	
optType	ei:optType	
optReason	ei:optReason	
requestID	pyld:requestID	
marketContext	ei:marektContext	
createdDateTime	ei:createdDateTime	
service	(Tag 이름으로 존재)	
available	dtstart	
	duration	RandomizableEvent:randomizeDuration
	requestPower	xcal :availability
	startYMD	TimeObject:dstStartTime
	startTime	TimeObject:dstStartTime
	endYMD	TimeObject:dstEndTime
	endTime	TimeObject:dstEndTime

# 2. Smart Home Energy Framework :

## 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Schedule (Opt)

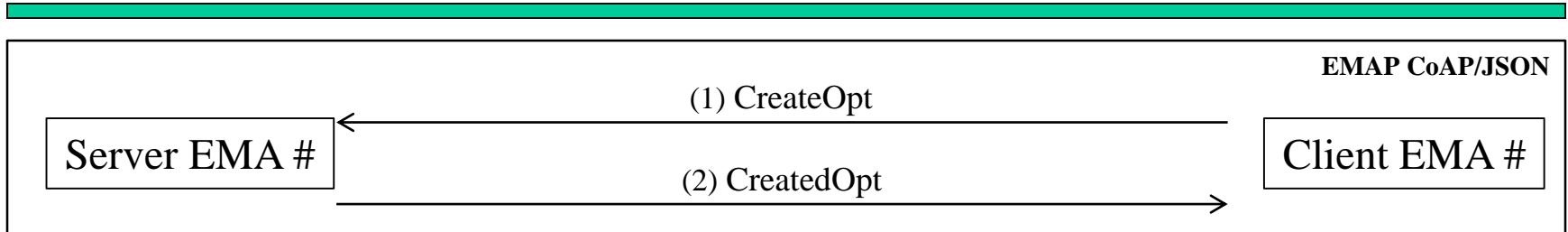


(2) CreatedOpt

Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
responseCode	ei:responseCode	
responseDescription	ei:responseDescription	
requestID	pyld:requestID	
optID	ei:optID	
optStatus		IdentifiedObject:TraiilProfile:ServiceKind
service	(Tag 이름으로 존재)	

# 2. Smart Home Energy Framework :

## 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Schedule (Opt)



(1) CreateOpt

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
optID	opt identifier
optType	type of opt
optReason	opt reason(e.g. emergency)
marketContext	refer market address
available	dtstart opt start time
	duration opt duration
	requestPower opt 요청 전력량
	startYMD opt start date
	startTime opt start time
requestID	endYMD opt end date
	endTime opt end time
service	request identifier
service	type of service

(2) CreatedOpt

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
optID	opt identifier
service	type of service
optStatus	

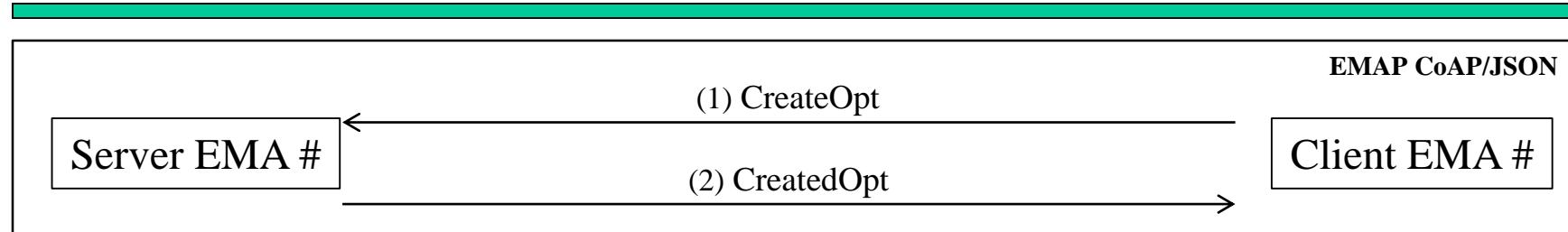
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Schedule (Opt)



(1) CreateOpt

444 60.023690	322 192.168.1.102	192.168.1.127	CoAP	CON, MID:39442, PUT, /CreateOpt (application/json)
445 60.024630	202 192.168.1.127	192.168.1.102	CoAP	ACK, MID:39442, 2.05 Content (application/json)

(2) CreatedOpt

```
CreateOpt Object{
  "SrcEMA" : String,
  "DestEMA" : String,
  "optID": String,
  "optType": String,
  "optReason": String,
  "requestID" : String,
  "marketContext" : String,
  "createdDateTime": Date,
  "service" : String,
  "available" : Array,
}
```

```
available Array{
  "dtstart" : Date,
  "duration" : String,
  "requestPower" : double,
  "startYMD": Integer,
  "startTime": Integer,
  "endYMD": Integer,
  "endTime": Integer,
}
```

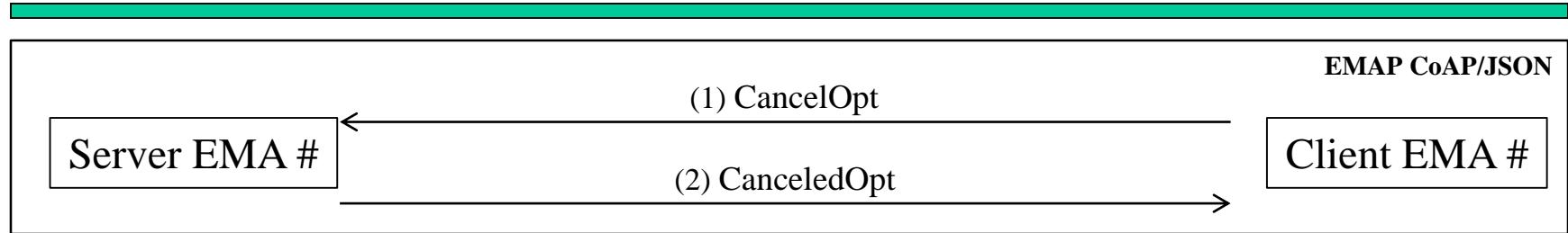
```
..00 .... = Type: Confirmable (0)
...0000 = Token Length: 0
Code: PUT (3)
Message ID: 39442
> Opt Name: #1: Uri-Path: CreateOpt
> Opt Name: #2: Content-Format: application/json
End of options marker: 255
[Response In: 5093]
Payload: Payload Content-Format: application/json, Length: 263
Payload Desc: application/json
JavaScript Object Notation: application/json
  Object
    > Member Key: SrcEMA
    > Member Key: DestEMA
    > Member Key: optID
    > Member Key: optType
    > Member Key: optReason
    > Member Key: requestID
    > Member Key: requestPower
    > Member Key: startYMD
    > Member Key: startTime
    > Member Key: endYMD
    > Member Key: endTime
    > Member Key: duration
    > Member Key: service
    > Member Key: createdDateTime
```

```
CreatedOpt Object{
  "SrcEMA" : String,
  "DestEMA" : String,
  "optID": String,
  "requestID": String,
  "responseCode": Integer,
  "responseDescription" : String,
  "service" : String,
  "optStatus": String
}
```

```
Payload Desc: application/json
JavaScript Object Notation: application/json
  Object
    > Member Key: optStatus
    > Member Key: SrcEMA
    > Member Key: responseDescription
    > Member Key: requestID
    > Member Key: service
    > Member Key: DestEMA
    > Member Key: optID
    > Member Key: responseCode
```

# 2. Smart Home Energy Framework :

## 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Schedule \_CancelOpt



(1) CancelOpt

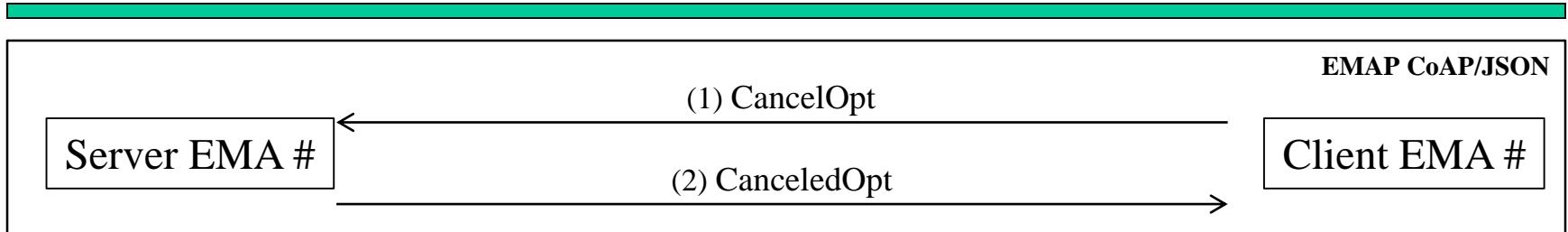
Key Name	Reference	
Key Name	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
requestID	pyld:requestID	
optID	ei:OptID	
time		RandomizableEvent:creation Time
service	(Tag 이름으로 존재)	

(2) CanceledOpt

Key Name	Reference	
Key Name	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
responseCode	Ei:responseCode	
responseDescription	ei:responseDescription	
requestID	Pyld:requestID	
optID	ei:optID	

# 2. Smart Home Energy Framework :

## 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Schedule \_CancelOpt



(1) CancelOpt

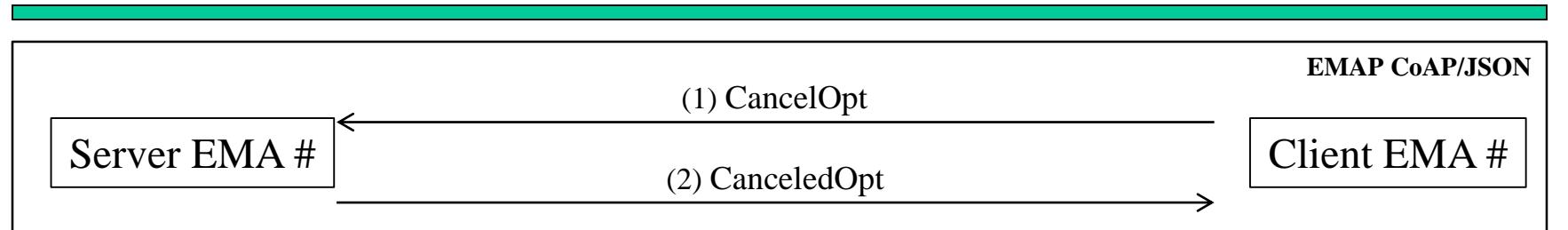
Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
optID	opt identifier
service	type of service
time	service creation time

(2) CanceledOpt

Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
responseCode	response code
responseDescription	description of response code
optID	opt identifier
service	type of service

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Schedule \_CancelOpt



(1) CancelOpt  
 (2) CanceledOpt

484 64.751022	179 192.168.1.102	192.168.1.127	CoAP	CON, MID:42798, PUT, /CancelOpt (application/json)
486 64.751910	224 192.168.1.127	192.168.1.102	CoAP	ACK, MID:42798, 2.05 Content (application/json)

```

CancelOpt JSONObject{
    "optID": String,
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "service" : String
    "time": Date
}
    
```

```

Payload Desc: application/json
JavaScript Object Notation: application/json
Object
    Member Key: optID
    Member Key: SrcEMA
    Member Key: DestEMA
    Member Key: requestID
    Member Key: service
    Member Key: time
    
```

```

CanceledOpt Object{
    "optID": String,
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription" : String
    "service": String
}
    
```

```

JavaScript Object Notation: application/json
Object
    Member Key: SrcEMA
    Member Key: responseDescription
    Member Key: requestID
    Member Key: service
    Member Key: time
    Member Key: DestEMA
    Member Key: optID
    Member Key: responseCode
    
```

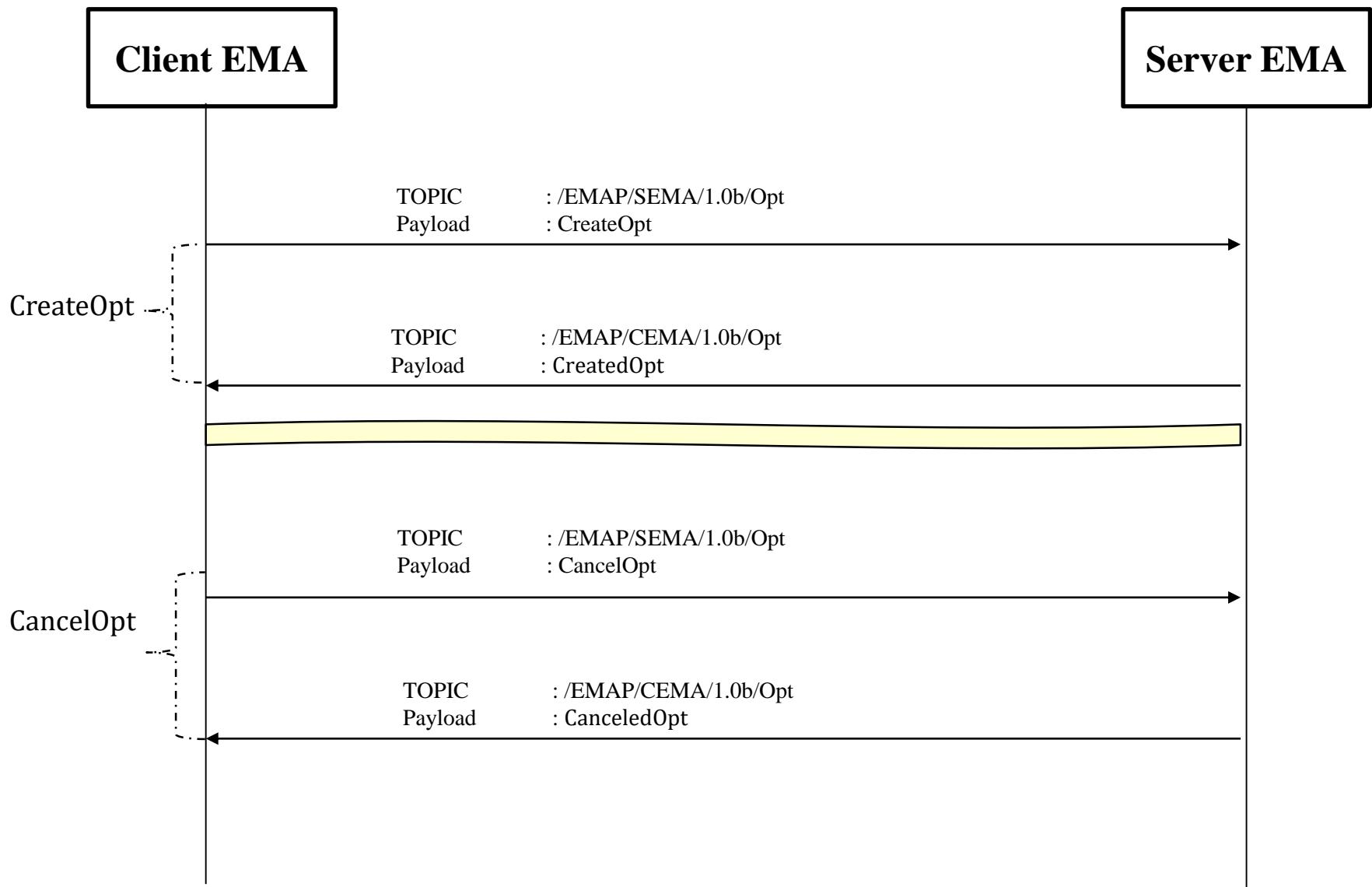
# **EMAP**

## **(4) Opt**

- **CoAP/JSON**
- **MQTT / JSON**

## 2.2 EMAP(MQTT/JSON)

### Service : Opt



# **Scenario**

- **Discovery**
- **Report**
- **DR Event**

# Smart Home overall Terms

## 스마트 에너지 홈 (Smart Energy Home)

: 스마트 에너지 홈이란 전력 기술과 ICT 기술이 융합되어 탄생한 지능형 전력망 기술이 댁내 환경에 도입됨에 따라, 생산자 중심으로 이루어지는 에너지의 관리 기능을 소비자 중심으로 전환하여 소비자 측면에서의 에너지 효율성 향상 및 이를 위한 서비스 제공이 가능하도록 하는 기술.

## 지능형 홈 네트워크

: 홈네트워크를 기본으로 댁내에 존재하는 정보 가전 기기들을 통합, 연결하는 동시에 외부에 존재하는 여러 종류의 서비스 네트워크들과의 연결성을 제공하여 누구나 기기, 시간, 장소에 구애받지 않고 다양한 컨텐츠와 서비스를 제공할 수 있는 환경으로, 지능형 홈, 홈네트워크, 디지털 홈, 스마트 홈 환경과 인프라를 종합하는 개념

## OpenADR(Automated Demand Response)

: 지능형 DR(Demand Response)에 적용되는 표준 통신 프로토콜 특히 전력공급자와 소비자 간 양방향 통신을 가능케 함으로써 송전과 배전의 효율성을 극대화하는 게 주목적. OpenADR 물리적 통신방식이 아닌 명령, 요청 등과 같은 DR 신호 교환을 위한 응용계층의 표준기술로써 OpenADR을 적용함으로써 전력공급자가 실시간으로 수요관리 명령을 내렸을 때, 소비자가 이에 대응해 자동으로 전기사용량을 줄이는 지능형 DR을 구현

## OpenFMB(Field Message Bus)

: OpenFMB는 그리드 말단의 상호운용성(interoperability)과 분산지능화(distributed intelligence)를 위한 프레임워크이자 참고 아키텍쳐

## OpenHAB(Open Home Automation Bus)

: 홈 자동화에 대한 모든 것들의 전반적인 통합 플랫폼을 제공하기 위한 프로젝트

## Demand Response

: 스마트그리드에서 수요반응이란 Utility company와 ISO, third part service provider가 소비자들의 최대 전기 사용량 차이를 완화시키고, 전기 공급과 수요의 균형을 맞추기 위하여 일시적으로 전기 사용량을 축소시키기 위해 소비자의 전력사용패턴 변화를 유도하는 기술

# Energy Management System

---

## 에너지 관리 시스템(Energy Management System; EMS)

: 에너지소비자에게 에너지 절감 및 에너지효율화를 포함한 통합적인 에너지관리 서비스를 제공하기 위한 시스템

## 분산 전원 시스템(Distributed Energy Resources; DER)

: 중앙집중 형 발전에서 벗어나, 수요자 주변에 발전소를 분산 배치하여 마이크로그리드 망에서 전력 수송을 효율적으로 해 주는 차세대 시스템

## BAS(Building Automation System)

: 전력, 공조, 냉난방, 조명 설비 등과 같은 건물에너지 설비에 대한 상태 감시 및 자동화 된 감시 조작 시스템으로 건물 설비에 대한 자동화 운용 및 중앙 감시를 목적으로 하는 시스템

## Home Gateway

: 다양한 가정 내 통신 기기들을 외부의 통신망과의 통신을 위해 연결하기 위한 가정 내 네트워크 장치로 외부와의 통신뿐만 아니라 가정 내의 통신 기기들의 통신을 지원

## SCADA(Supervisory Control And Data Acquisition)

: ICT 기술을 이용하여 산업 공정, 기반시설, 설비를 바탕으로 한 작업 공정을 감시하고 제어하는 시스템

## AMI(Advanced Metering Infrastructure)

: 기존의 인력에 의한 검침 방법의 검침 효율 및 정확성 결여 등의 문제를 개선하고자 제안 된 방법

## IHID(In Home Display)

: AMI의 주요 구성요소로서 고객의 건물 내에서 AMI와 연계하여 에너지 사용 및 통계에 대한 정보, 서비스 정보 등을 실시간으로 소비자에게 제공하여 사용자에게 자발적 에너지 절감을 유도하는 방식

# Energy Management Agent

**홈 에너지관리에이전트(Home Energy Management Agnet)**

: 홈 소비자 영역에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치 (specified in ISO/IEC 15067-3)

**커뮤니티 에너지관리에이전트(Community Energy Management Agnet)**

: 커뮤니티 영역에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치

**온라인 에너지관리에이전트(Online Energy Management Agnet)**

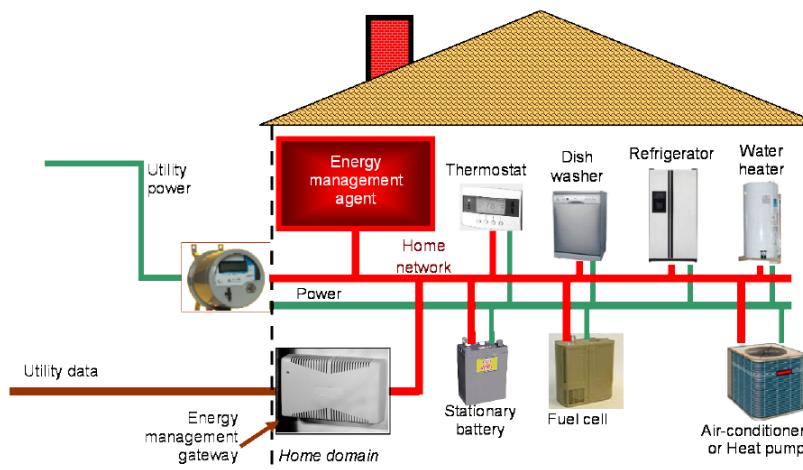
: 클라우드 환경에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치

**커뮤니티 수요반응(Community Demand Response)**

: 소규모 수요반응 자원을 하나의 수요반응 자원으로 통합하여 보다 커다란 규모의 수요반응 자원 제어하는 기술

**커뮤니티 에너지관리시스템(Community Energy Management System)**

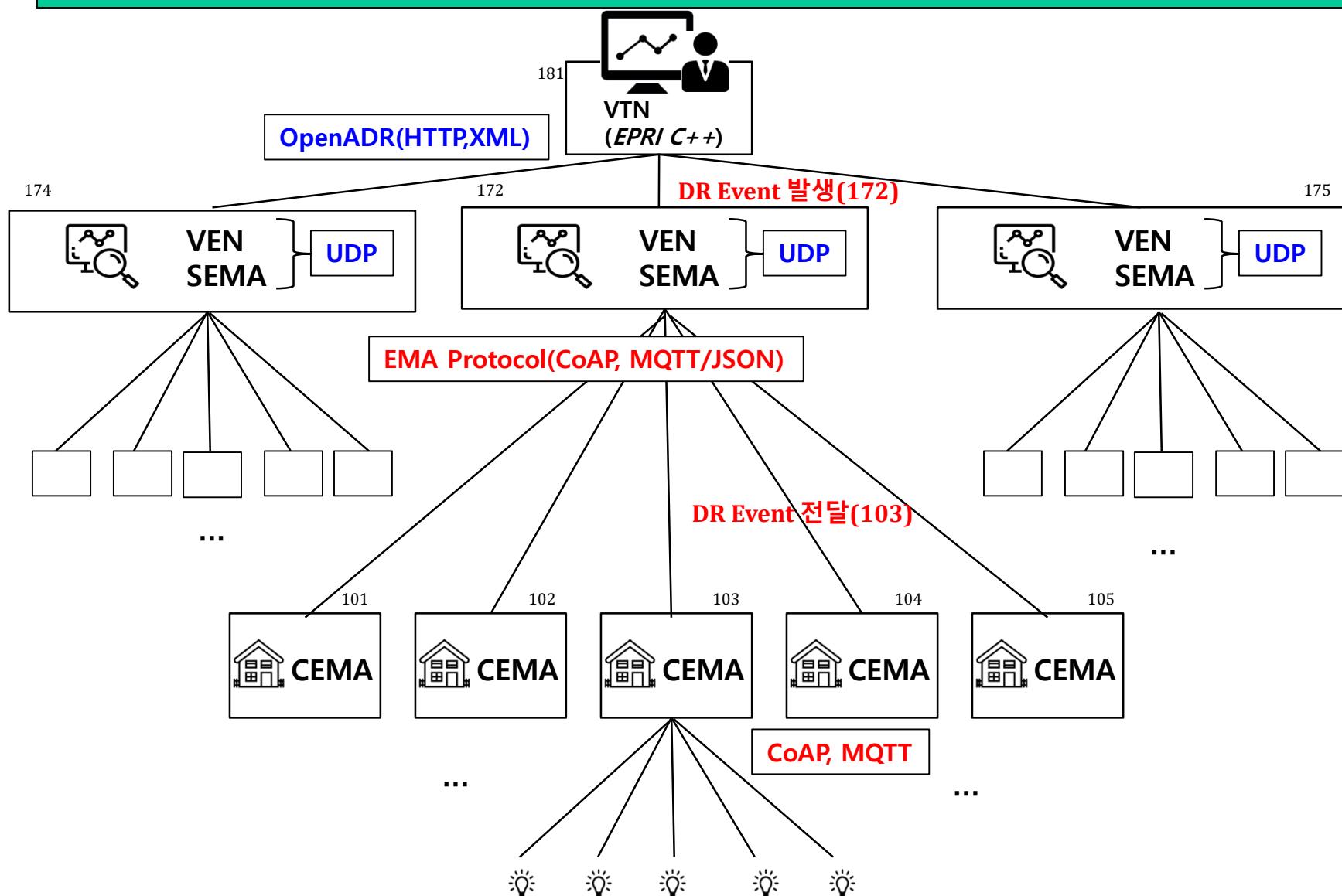
: 다수의 개별 홈 소비자로부터 에너지 정보를 수집, 통합하여 커뮤니티 또는 온라인 에너지관리시스템을 통한 에너지 관리 시스템



# **Main Part**

# AS-IS

VTN : 1대 (리눅스)  
SEMA + VEN: 3대(리눅스)  
CEMA : SEMA 1대당 5대(OpenWRT)  
Device : CEMA 1대당 가상5대

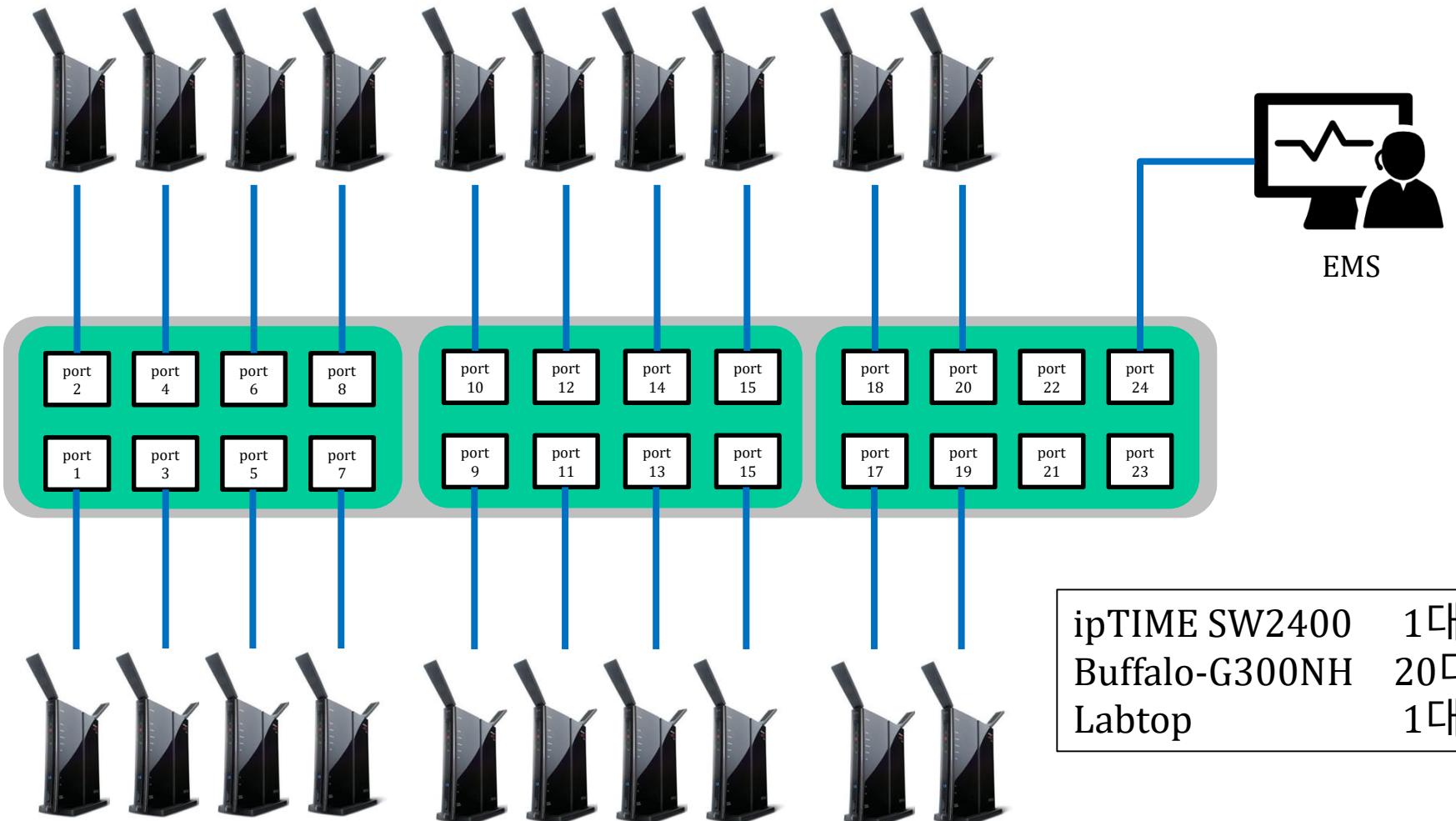


## 2. EMAP 패킷 및 이벤트 속도 측정

### 실험 환경

VTN (EPRI) Local IP : 192.168.1.181	Host PC	CPU	Intel Core i7-7770
		CPU Speed	2.6GHz
		RAM	16GB
		OS	LINUX
		Language	RUBY, C++
VEN-SEMA (UDP)	Gateway	Model	Buffalo WZR-HP-G300NH
		CPU	Atheros AR9132 rev 2 (0xb9)
		CPU Speed	400 MHz
		RAM	64 MB
		OS	OpenWRT 15.05 Chaos Calmer
		Language	C, C++
MQTT Broker 166.104.28.51	Broker	OS	Ubuntu

# Experiment Testbed (Server EMS- Client EMA)



# Smart Energy Framework

IP : 192.168.1.xxx

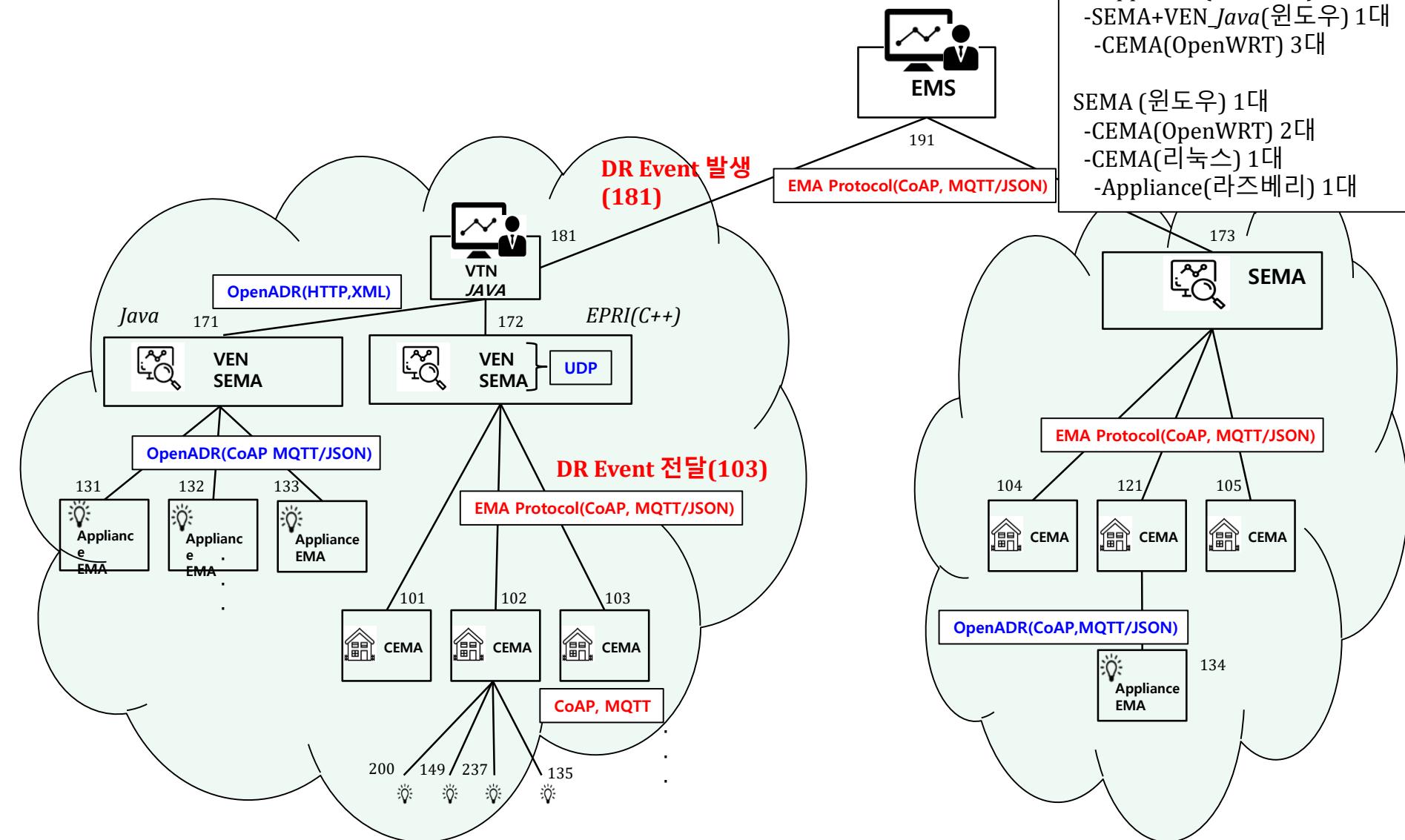
EMS 1대 (Window)

VTN 1대 (리눅스)

- SEMA+VEN (리눅스) 1대
- Appliance(라즈베리) 3대
- SEMA+VEN\_Java(윈도우) 1대
- CEMA(OpenWRT) 3대

SEMA (윈도우) 1대

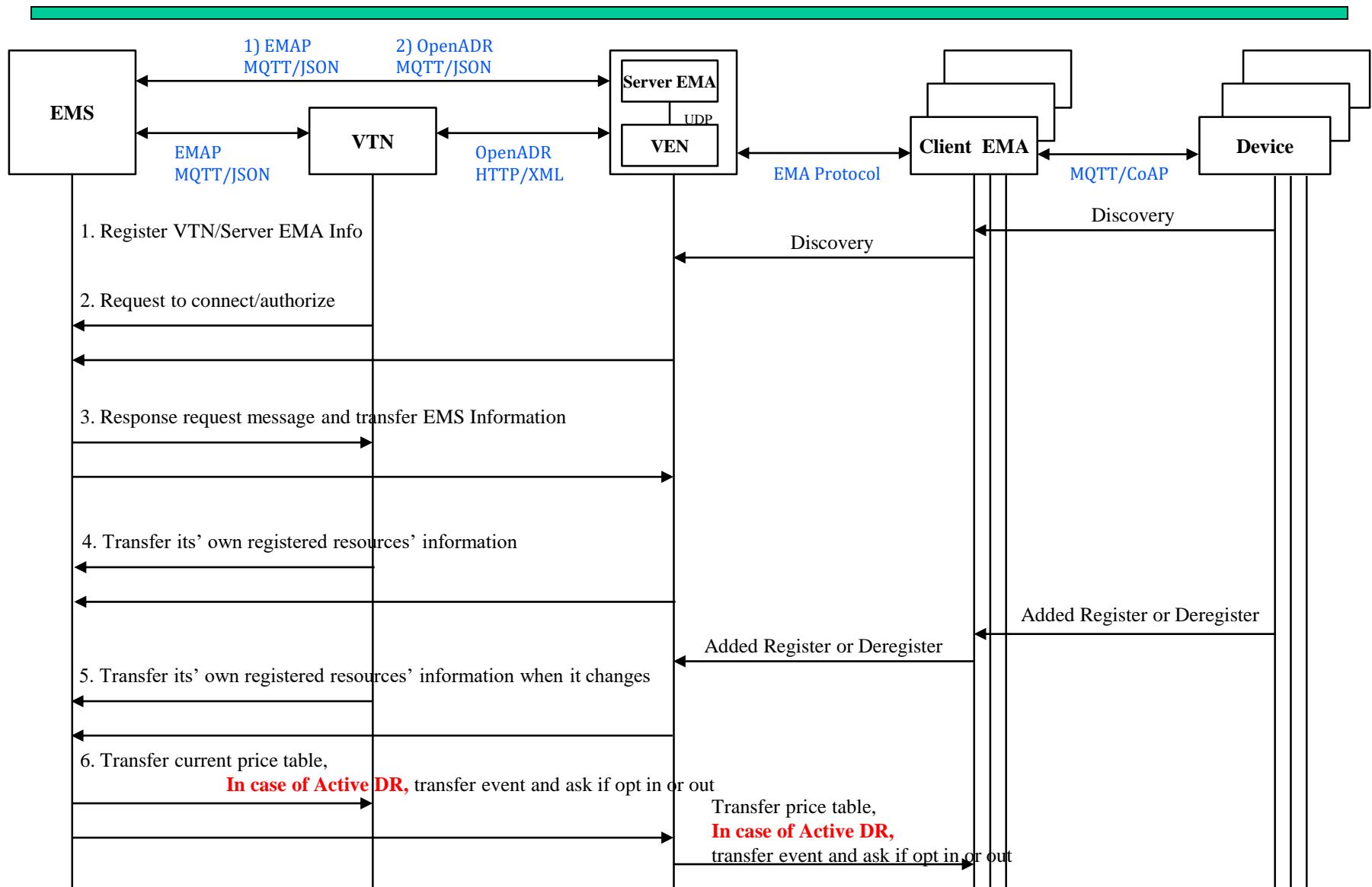
- CEMA(OpenWRT) 2대
- CEMA(리눅스) 1대
- Appliance(라즈베리) 1대



## 2. EMAP 패킷 및 이벤트 속도 측정 실험 환경

EMS Server Local IP : 192.168.1.218	Host PC	CPU	Intel Core i7-7770
		CPU Speed	2.6GHz
		RAM	16GB
		OS	Window 10
		Language	JAVA,
EMA/ Client Local IP : 192.168.1.101~120	Gateway	Model	Buffalo WZR-HP-G300NH
		CPU	Atheros AR9132 rev 2 (0xb9)
		CPU Speed	400 MHz
		RAM	64 MB
		OS	OpenWRT 15.05 Chaos Calmer
		Language	C, C++
MQTT Broker 166.104.28.51	Broker	OS	Ubuntu

# Service Scenario – Discovery



# Service Scenario - Discovery

1. Register VTN/Server EMA Information on Energy Management System(EMS)

2. VTN/Server EMA Request connection/authorization to EMS

3.1 Response to request message and transfer EMS Information(e.g. Version, Profile, Protocol and etc.)

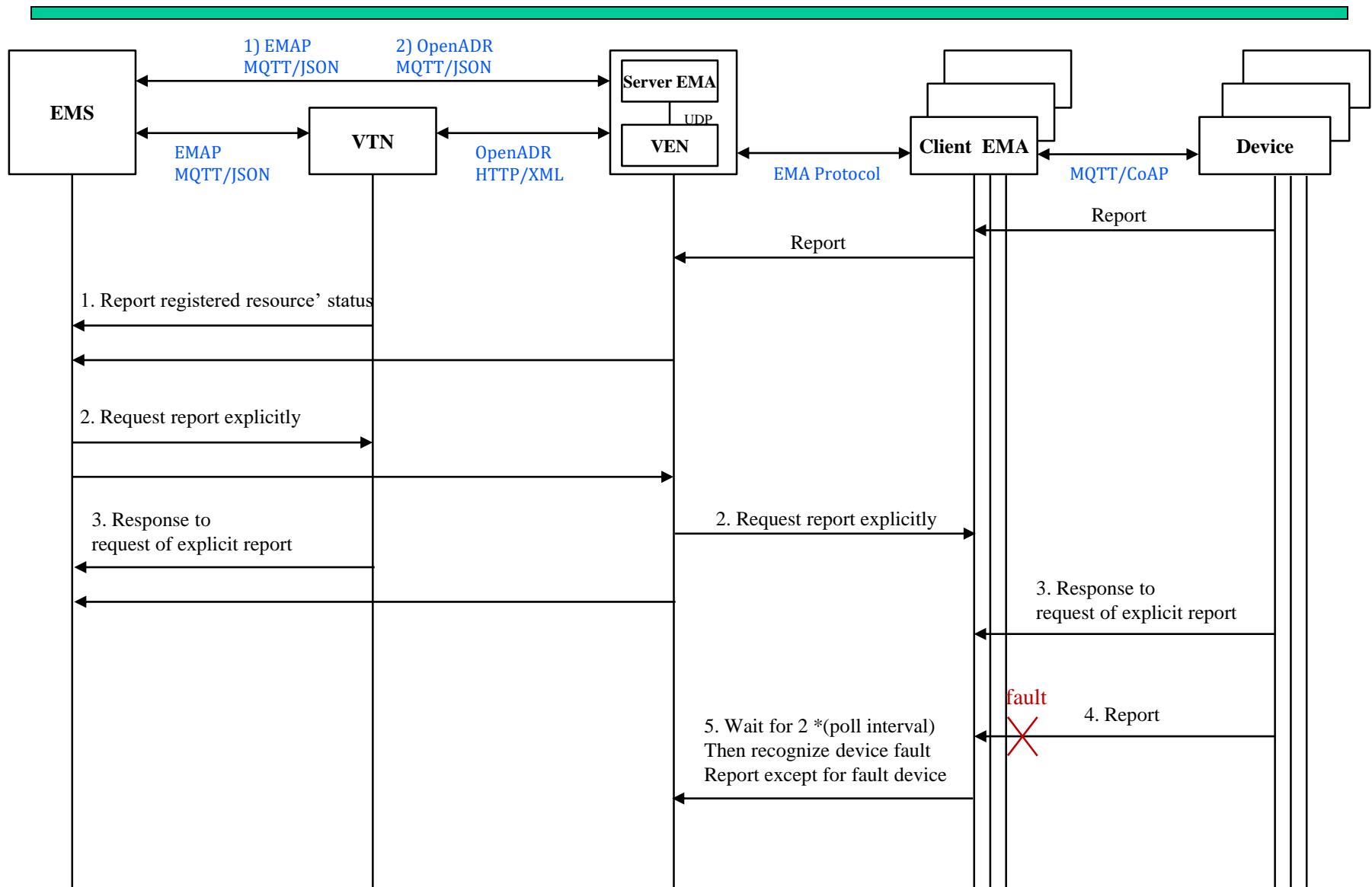
3.2 If VTN/Server EMA is not registered in advance on EMS, response as reject to request message

4. Transfer its' own registered resources' information (e.g. Appliance, Child EMA and etc.)

5. Transfer its' own registered resources' information when it changes(e.g. added de/registered device and etc.)

6. Transfer current price table. In case of Active DR, transfer event and ask if opt in or out

# Service Scenario – Report(Monitor)



# Service Scenario – Report(Monitor)

## Periodical

1. Report registered resource' status (e.g. KW, State)

## In case of request implicit / explicit information

2. If EMS, VTN or Server EMA need explicit information request explicit report by using acknowledge and vice versa(implicit)

3. Report to explicit report ( \* Considering customer privacy )

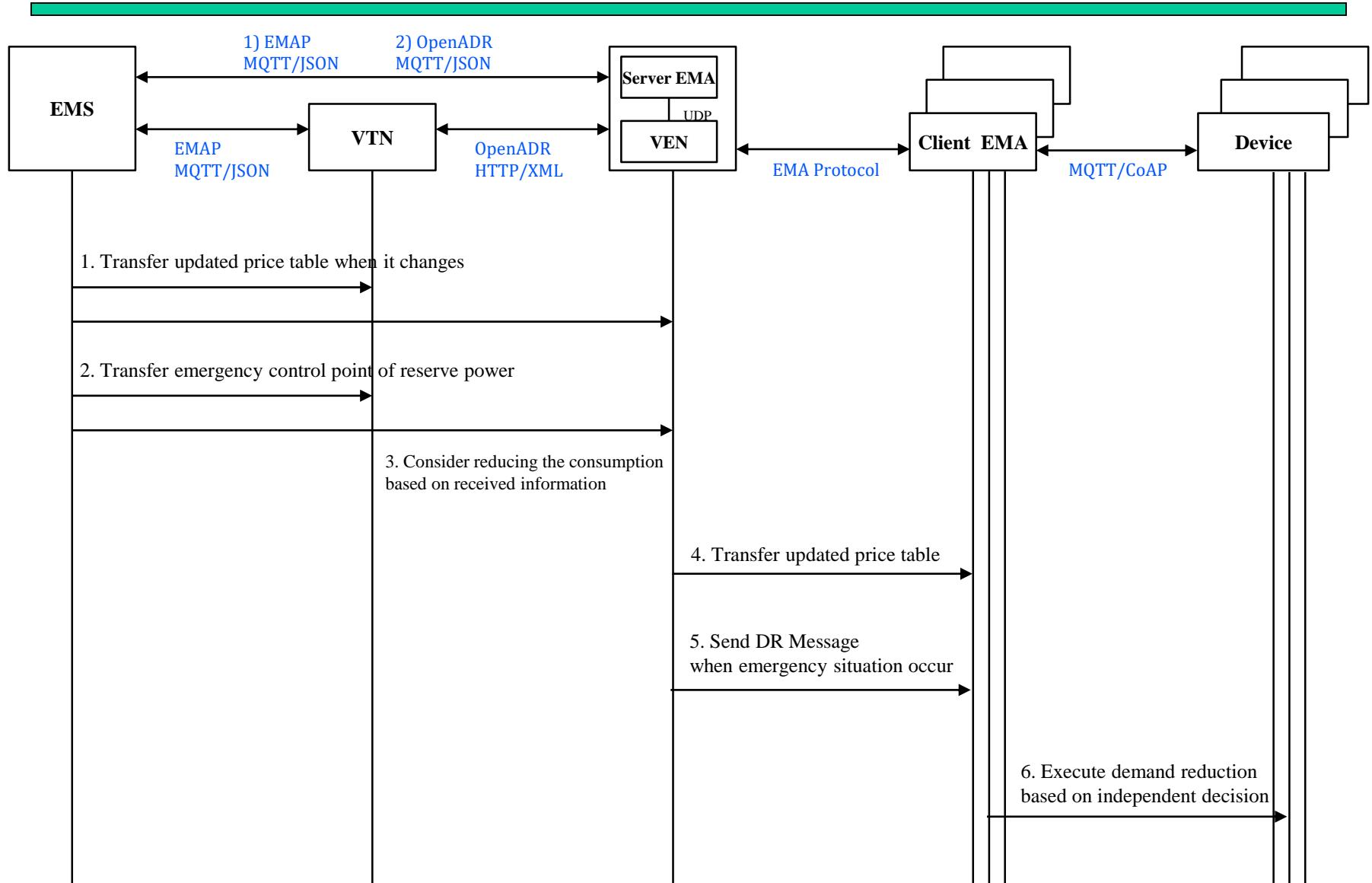
## Fault monitor

4. Device, Server/Client EMA and VEN report periodically.

5. Wait for 2 \*(poll interval) then it recognizes device fault.  
Report resources' information to Server EMA or VTN except for fault device

Report가 Poll 주기의 2배만큼 오지 않을 경우 Fault로 탐지하도록 하였습니다.  
추가로 Fault를 Recovery 생각해 보았는데 이 방법은 연구와 도움이 필요합니다.

# Service Scenario – DR Event (Pull)



# Service Scenario – DR (Passive)

1. EMS transfers updated price table (e.g. TOU, CPP, RTP)  
including registered program, unit and currency to VTN or Server EMA when it changes

2. Transfer emergency control point of reserve power (e.g. under 20% of reserve power)

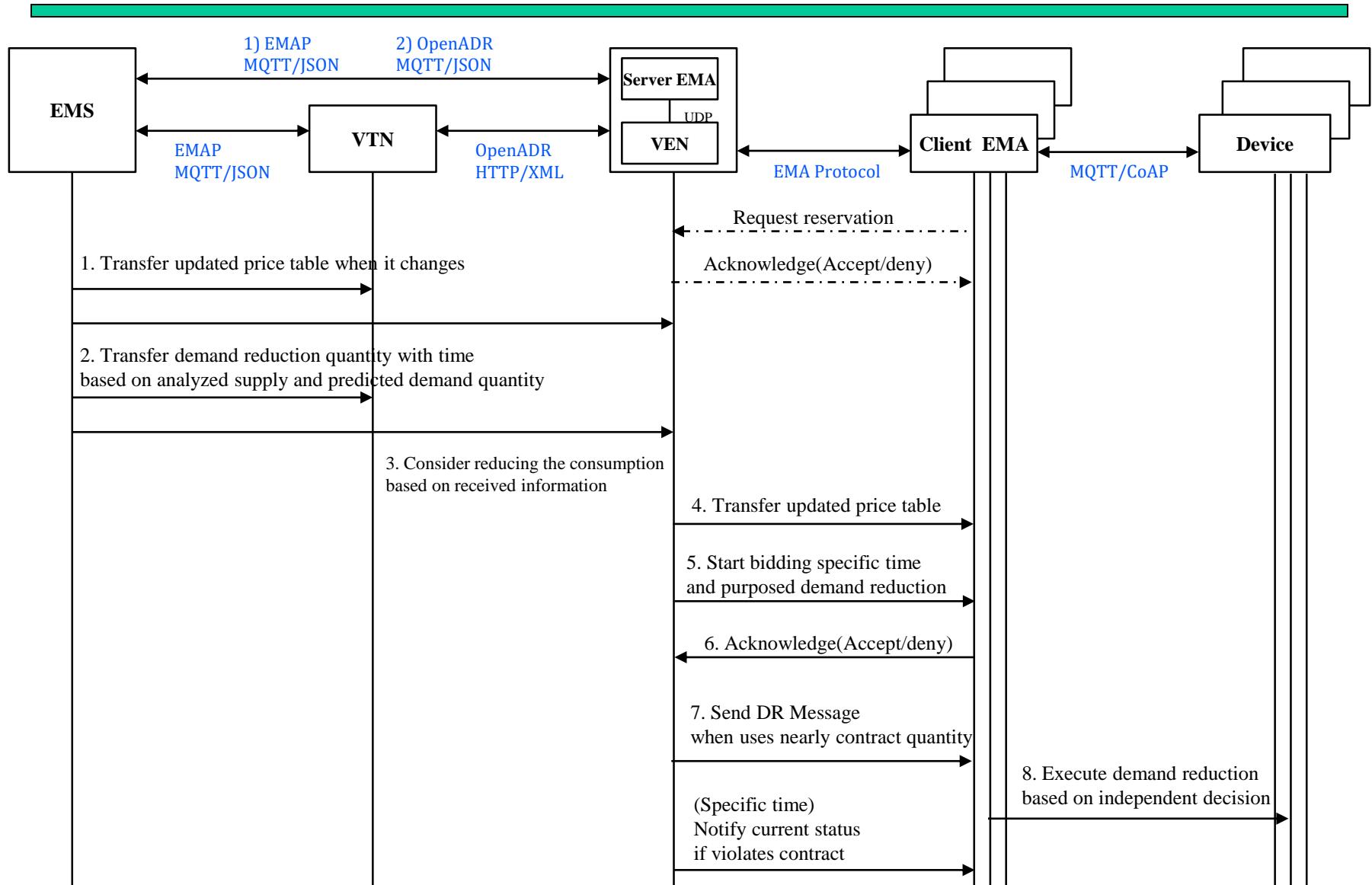
3. Consider reducing the consumption of VEN or Client EMA based on received information

4. VTN or Server EMA transfers updated price table (e.g. TOU, CPP, RTP)  
including registered program, unit and currency to VEN or Client EMA when it changes

4. Send DR Message when emergency situation occur when VTN or Server EMA have under 20% of reserve power

5. Execute demand reduction based on independent decision (e.g. Priority based control)

# Service Scenario – DR (Active)



# Service Scenario – DR (Active)

1. EMS transfers updated price table (e.g. TOU, CPP, RTP)  
including registered program, unit and currency to VTN or Server EMA when it changes

2. Transfer demand reduction quantity with time based on analyzed supply and predicted demand quantity

3. Consider reducing the consumption based on received information

4. VTN or Server EMA transfers updated price table (e.g. TOU, CPP, RTP)  
including registered program, unit and currency to VEN or Client EMA when it changes

5. Start bidding specific time and purposed demand reduction

6. Acknowledge(Accept/deny)

7. Send DR Message when uses nearly contract quantity

8. Execute demand reduction based on independent decision

(Specific time)Notify current status if violates contract

# 계층 실험 계획서

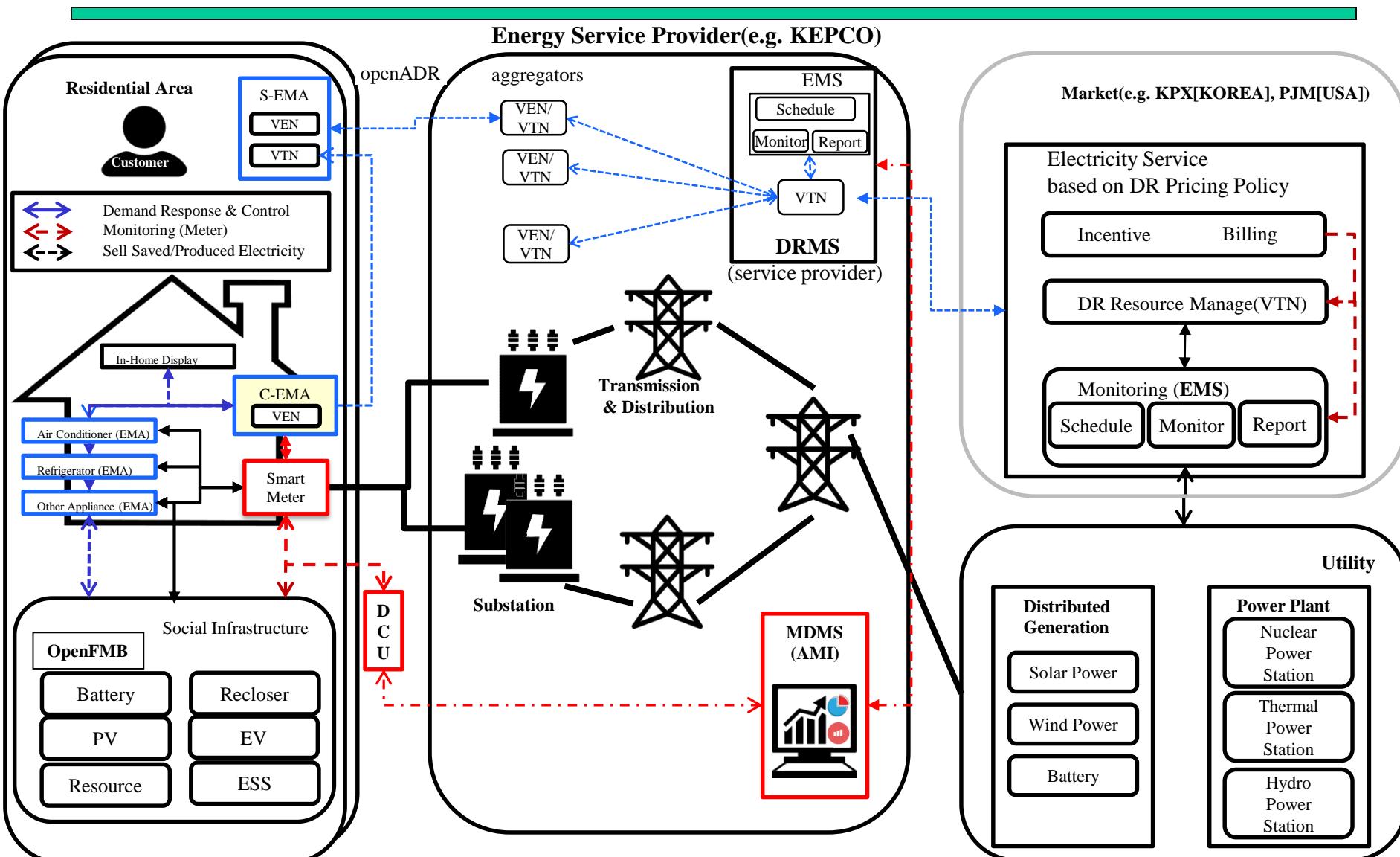
## DEMO

Hanyang University  
MIR-Lab



# Background

# Smart Energy Framework including Homes



# Smart Home overall Terms

## 스마트 에너지 홈 (Smart Energy Home)

: 스마트 에너지 홈이란 전력 기술과 ICT 기술이 융합되어 탄생한 지능형 전력망 기술이 댁내 환경에 도입됨에 따라, 생산자 중심으로 이루어지는 에너지의 관리 기능을 소비자 중심으로 전환하여 소비자 측면에서의 에너지 효율성 향상 및 이를 위한 서비스 제공이 가능하도록 하는 기술.

## 지능형 홈 네트워크

: 홈네트워크를 기본으로 댁내에 존재하는 정보 가전 기기들을 통합, 연결하는 동시에 외부에 존재하는 여러 종류의 서비스 네트워크들과의 연결성을 제공하여 누구나 기기, 시간, 장소에 구애받지 않고 다양한 컨텐츠와 서비스를 제공할 수 있는 환경으로, 지능형 홈, 홈네트워크, 디지털 홈, 스마트 홈 환경과 인프라를 종합하는 개념

## OpenADR(Automated Demand Response)

: 지능형 DR(Demand Response)에 적용되는 표준 통신 프로토콜 특히 전력공급자와 소비자 간 양방향 통신을 가능케 함으로써 송전과 배전의 효율성을 극대화하는 게 주목적. OpenADR 물리적 통신방식이 아닌 명령, 요청 등과 같은 DR 신호 교환을 위한 응용계층의 표준기술로써 OpenADR을 적용함으로써 전력공급자가 실시간으로 수요관리 명령을 내렸을 때, 소비자가 이에 대응해 자동으로 전기사용량을 줄이는 지능형 DR을 구현

## OpenFMB(Field Message Bus)

: OpenFMB는 그리드 말단의 상호운용성(interoperability)과 분산지능화(distributed intelligence)를 위한 프레임워크이자 참고 아키텍쳐

## OpenHAB(Open Home Automation Bus)

: 홈 자동화에 대한 모든 것들의 전반적인 통합 플랫폼을 제공하기 위한 프로젝트

## Demand Response

: 스마트그리드에서 수요반응이란 Utility company와 ISO, third part service provider가 소비자들의 최대 전기 사용량 차이를 완화시키고, 전기 공급과 수요의 균형을 맞추기 위하여 일시적으로 전기 사용량을 축소시키기 위해 소비자의 전력사용패턴 변화를 유도하는 기술

# Energy Management System

---

## 에너지 관리 시스템(Energy Management System; EMS)

: 에너지소비자에게 에너지 절감 및 에너지효율화를 포함한 통합적인 에너지관리 서비스를 제공하기 위한 시스템

## 분산 전원 시스템(Distributed Energy Resources; DER)

: 중앙집중 형 발전에서 벗어나, 수요자 주변에 발전소를 분산 배치하여 마이크로그리드 망에서 전력 수송을 효율적으로 해 주는 차세대 시스템

## BAS(Building Automation System)

: 전력, 공조, 냉난방, 조명 설비 등과 같은 건물에너지 설비에 대한 상태 감시 및 자동화 된 감시 조작 시스템으로 건물 설비에 대한 자동화 운용 및 중앙 감시를 목적으로 하는 시스템

## Home Gateway

: 다양한 가정 내 통신 기기들을 외부의 통신망과의 통신을 위해 연결하기 위한 가정 내 네트워크 장치로 외부와의 통신뿐만 아니라 가정 내의 통신 기기들의 통신을 지원

## SCADA(Supervisory Control And Data Acquisition)

: ICT 기술을 이용하여 산업 공정, 기반시설, 설비를 바탕으로 한 작업 공정을 감시하고 제어하는 시스템

## AMI(Advanced Metering Infrastructure)

: 기존의 인력에 의한 검침 방법의 검침 효율 및 정확성 결여 등의 문제를 개선하고자 제안 된 방법

## IHID(In Home Display)

: AMI의 주요 구성요소로서 고객의 건물 내에서 AMI와 연계하여 에너지 사용 및 통계에 대한 정보, 서비스 정보 등을 실시간으로 소비자에게 제공하여 사용자에게 자발적 에너지 절감을 유도하는 방식

# Energy Management Agent

**홈 에너지관리에이전트(Home Energy Management Agnet)**

: 홈 소비자 영역에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치 (specified in ISO/IEC 15067-3)

**커뮤니티 에너지관리에이전트(Community Energy Management Agnet)**

: 커뮤니티 영역에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치

**온라인 에너지관리에이전트(Online Energy Management Agnet)**

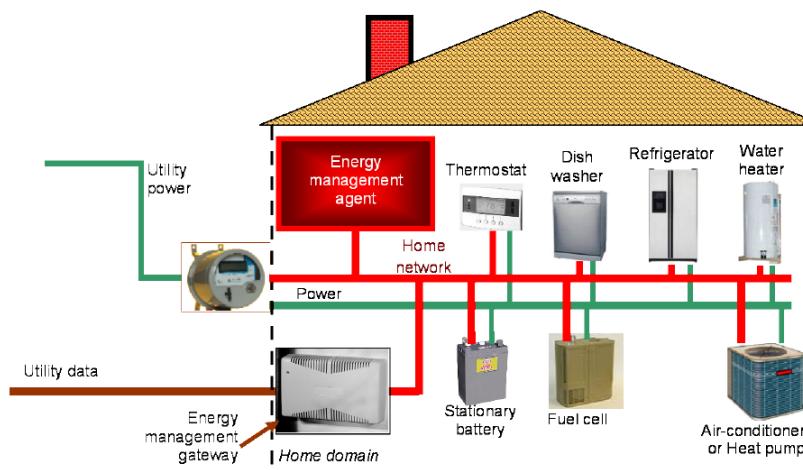
: 클라우드 환경에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치

**커뮤니티 수요반응(Community Demand Response)**

: 소규모 수요반응 자원을 하나의 수요반응 자원으로 통합하여 보다 커다란 규모의 수요반응 자원 제어하는 기술

**커뮤니티 에너지관리시스템(Community Energy Management System)**

: 다수의 개별 홈 소비자로부터 에너지 정보를 수집, 통합하여 커뮤니티 또는 온라인 에너지관리시스템을 통한 에너지 관리 시스템



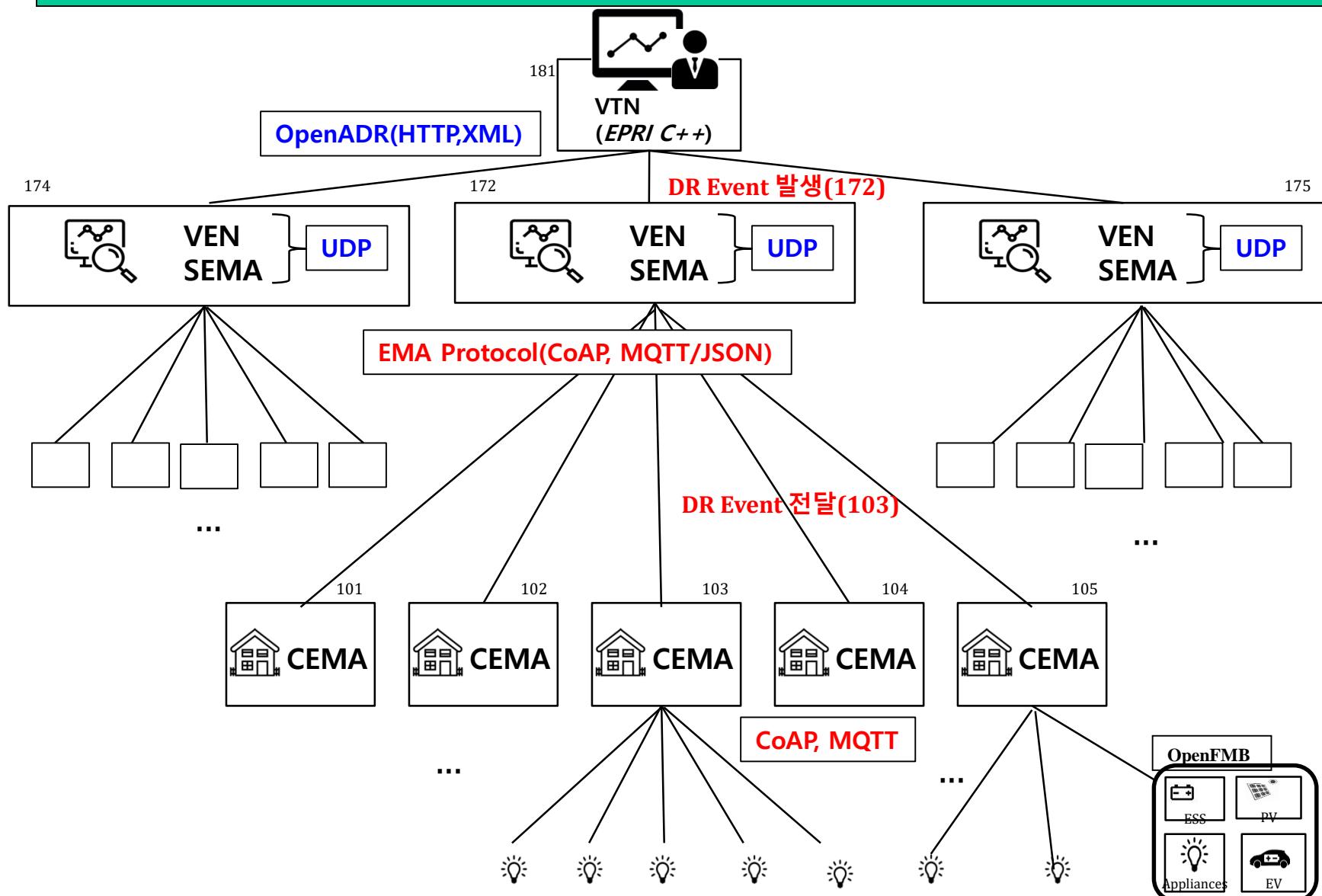
# Main Part

1. 계층 구조 AS-IS
2. 전체 Framework

# 계층 구조

## AS-IS

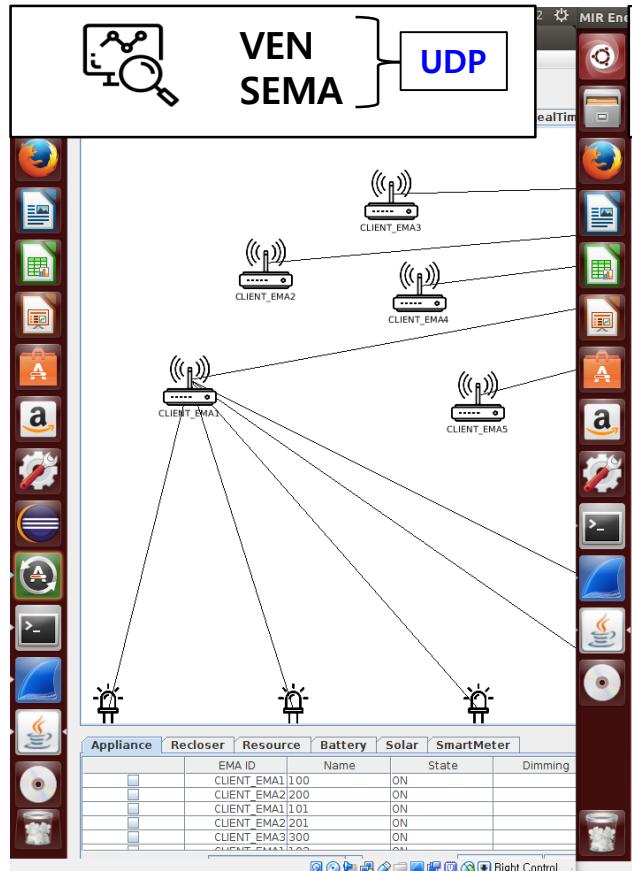
VTN : 1대 (리눅스)  
 SEMA +VEN: 3대(리눅스)  
 CEMA : SEMA 1대당 5대(OpenWRT)  
 Device : CEMA 1대당 가상5대



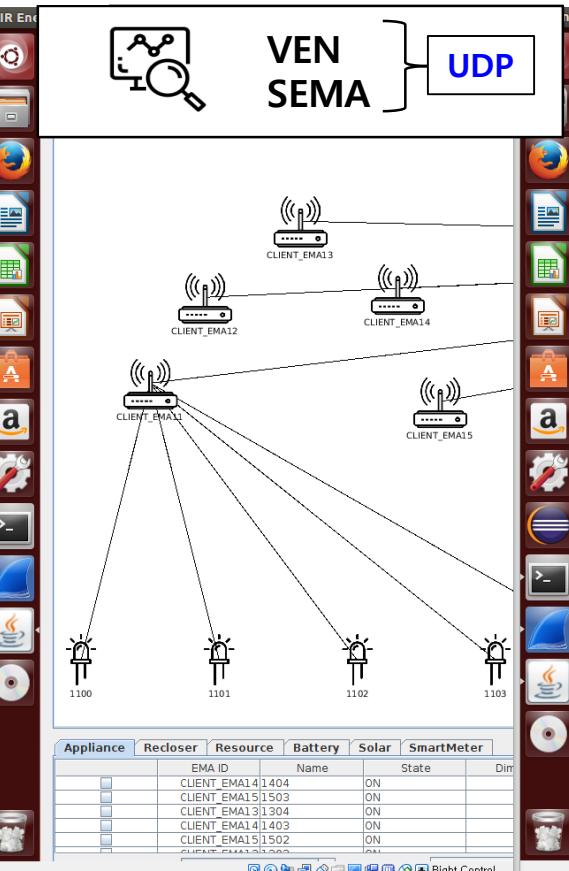
# 계층 구조

## AS-IS

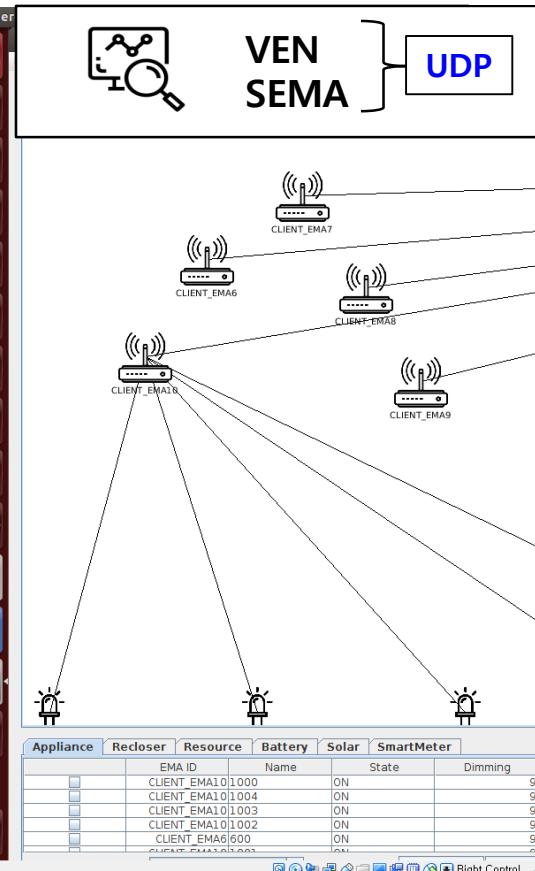
174



172



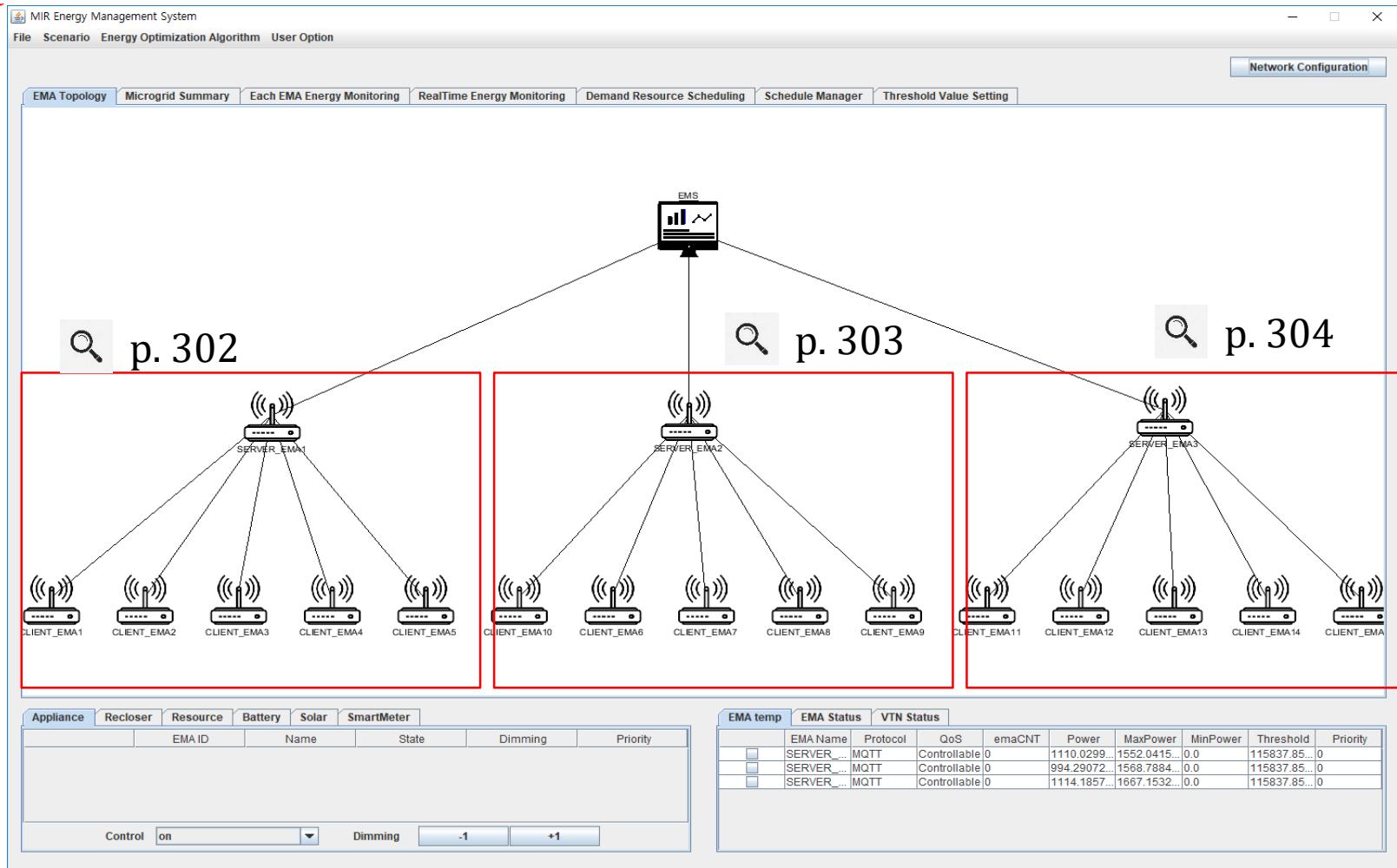
175



# 계층 구조 AS-IS

EMS : 1대  
SEMA : 3대  
CEMA : SEMA 1대당 5대 ( $3 \times 5 = 15$ 대)

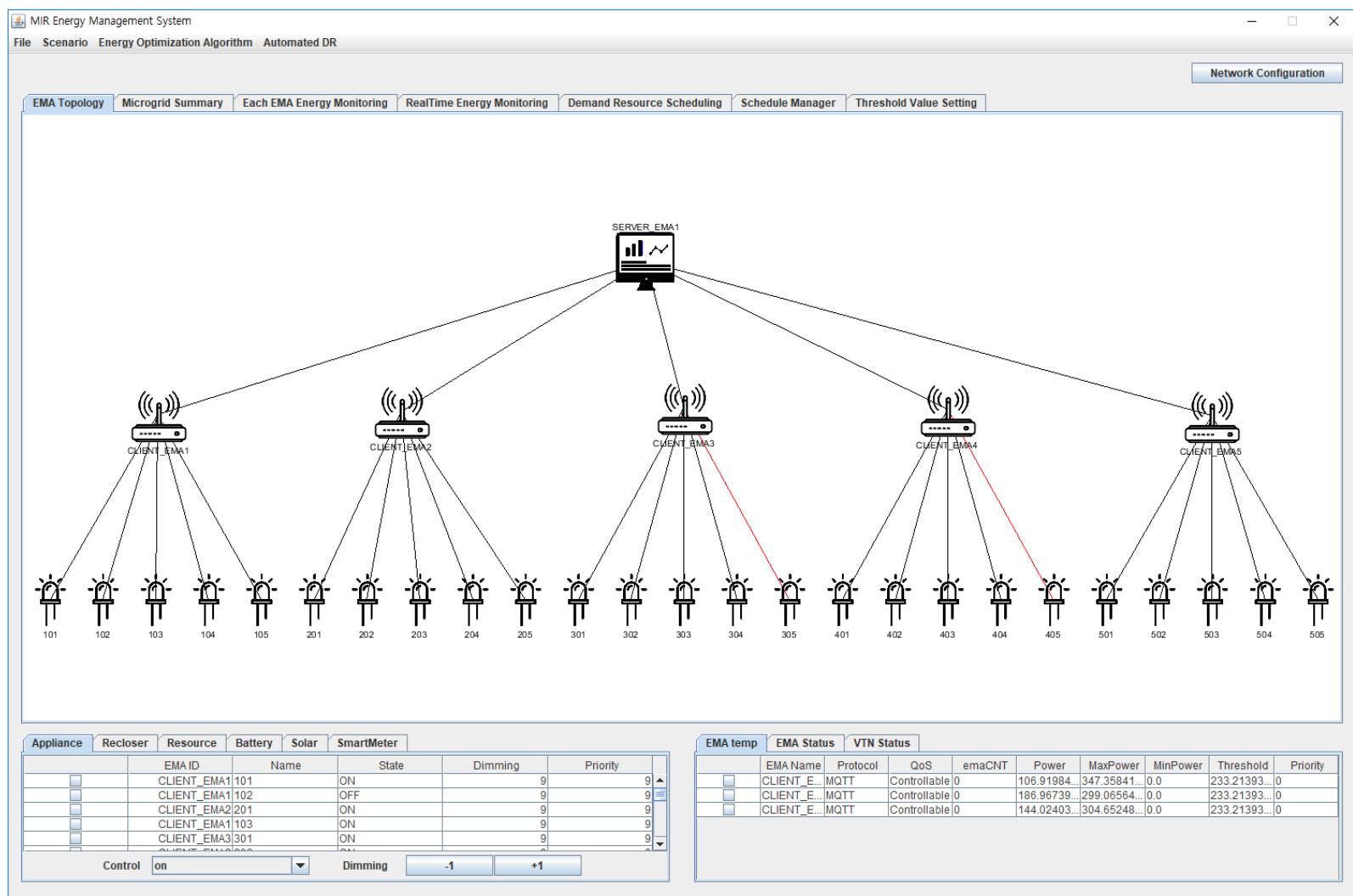
- Connect
- Disconnect



# 계층 구조 AS-IS

SEMA : 1대 (SERVER\_EMA1)  
 CEMA : 5대  
 Device : CEMA 1대당 5대 ( $5 \times 5 = 25$ 대)

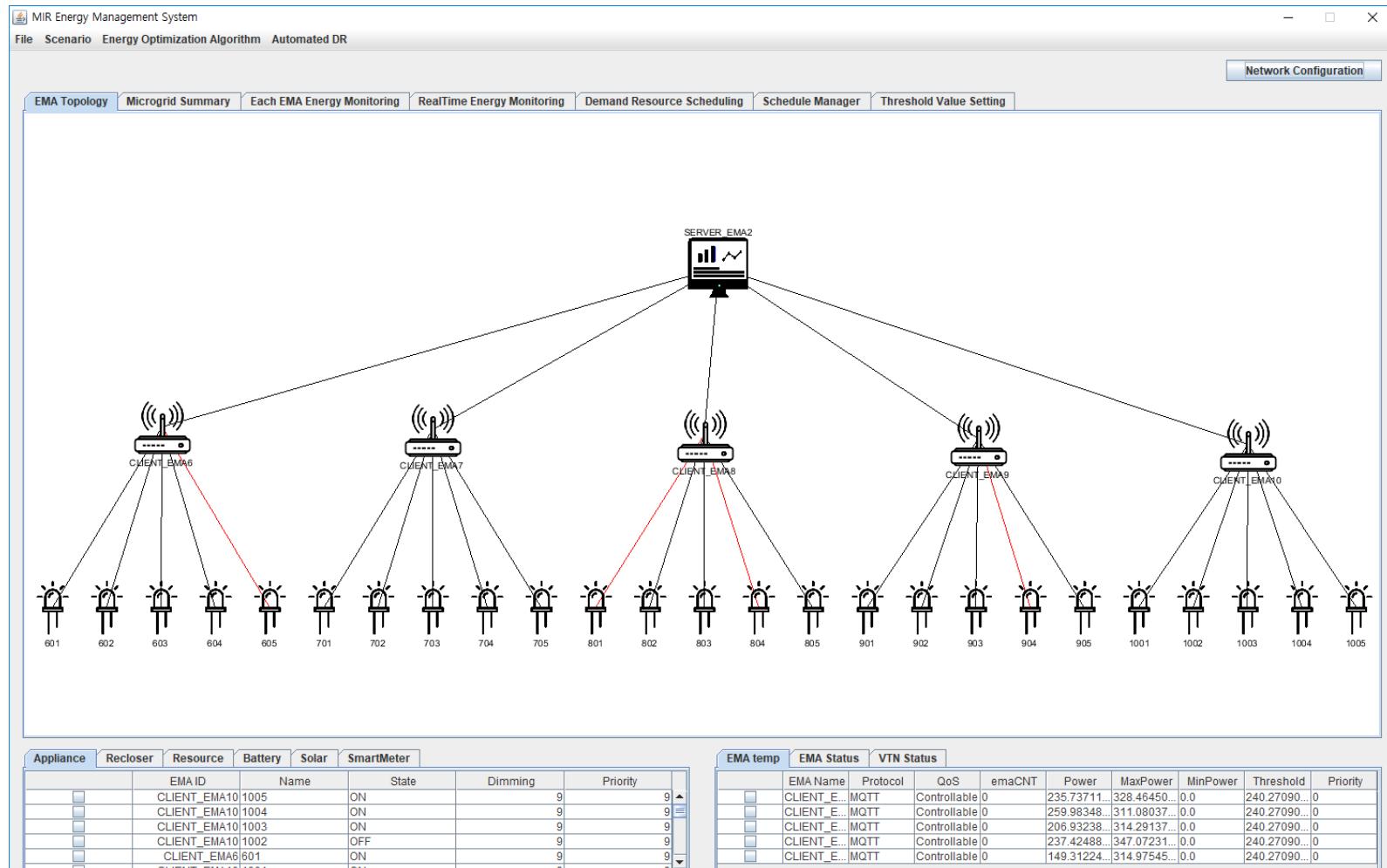
- Connect
- Disconnect



# 계층 구조 AS-IS

SEMA : 1대 (SERVER\_EMA2)  
 CEMA : 5대  
 Device : CEMA 1대당 5대 ( $5 \times 5 = 25$ 대)

- Connect
- Disconnect



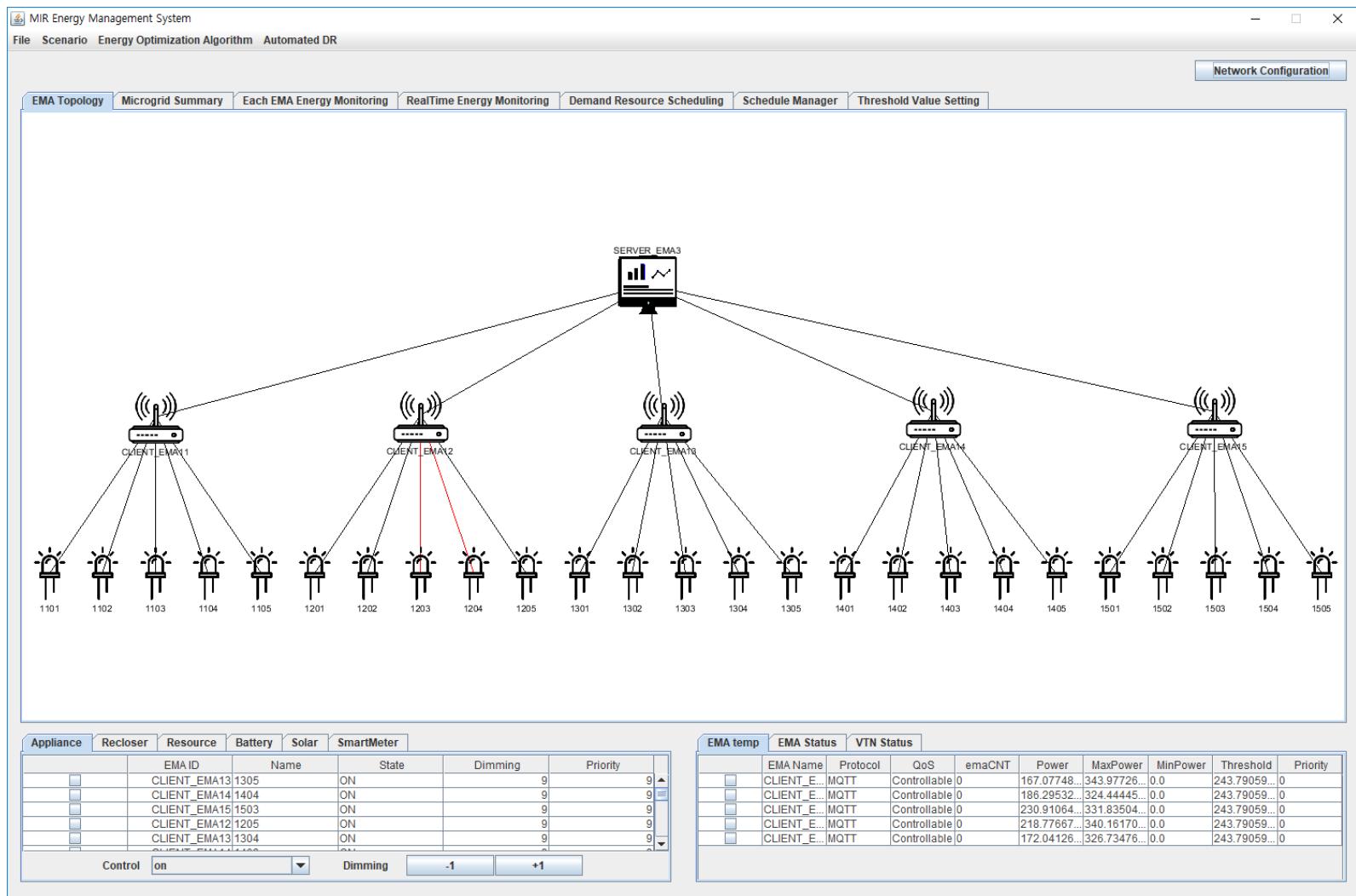
# 계층 구조 AS-IS

SEMA : 1대 (SERVER\_EMA3)

CEMA : 5 대

Device : CEMA 1대당 5대 ( $5 \times 5 = 25$  대)

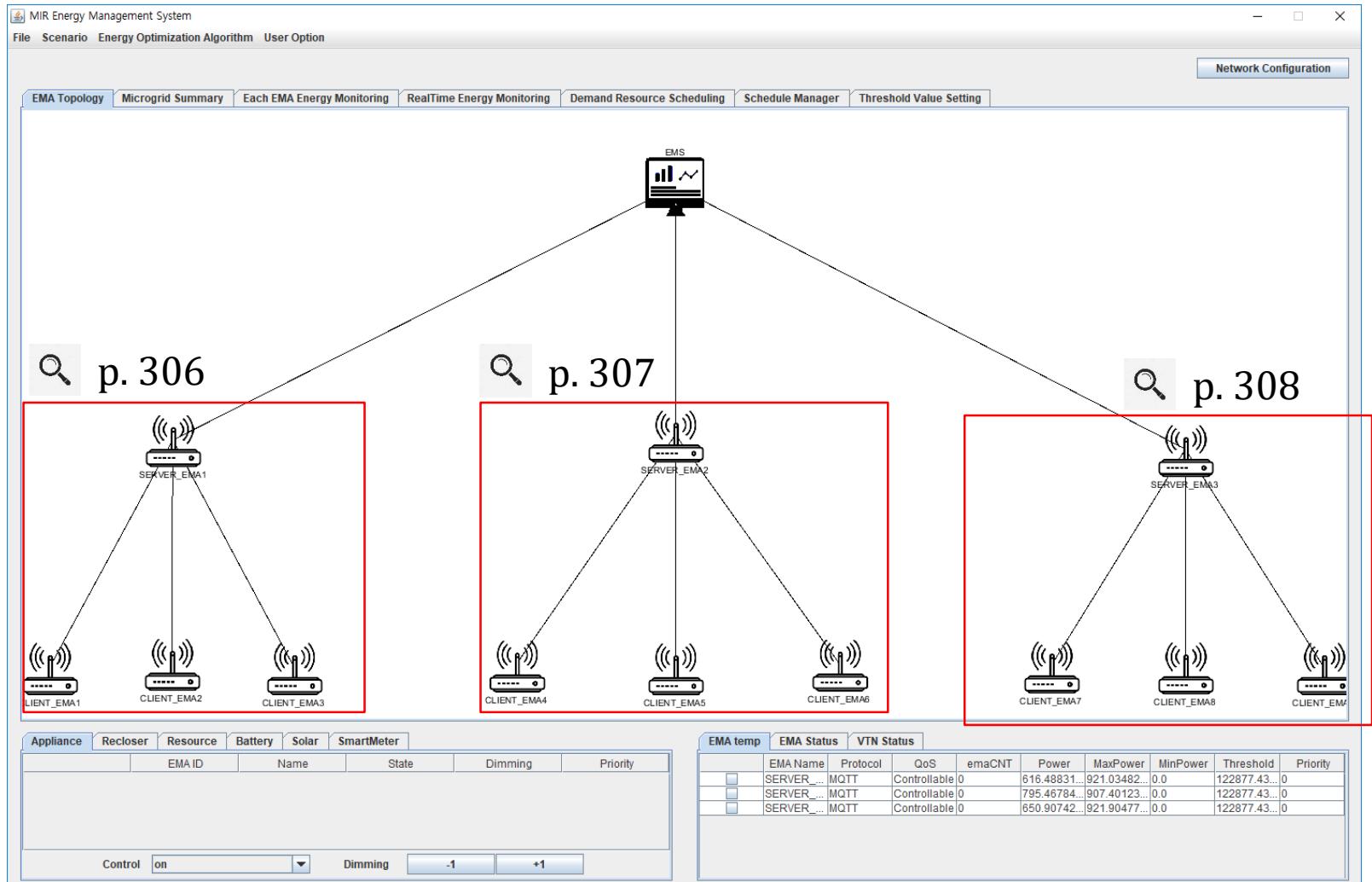
- Connect
  - Disconnect



# 계층 구조 AS-IS

EMS : 1대  
SEMA : 3대  
CEMA : SEMA 1대당 3대 ( $3 \times 3 = 9$  대)

- Connect
- Disconnect



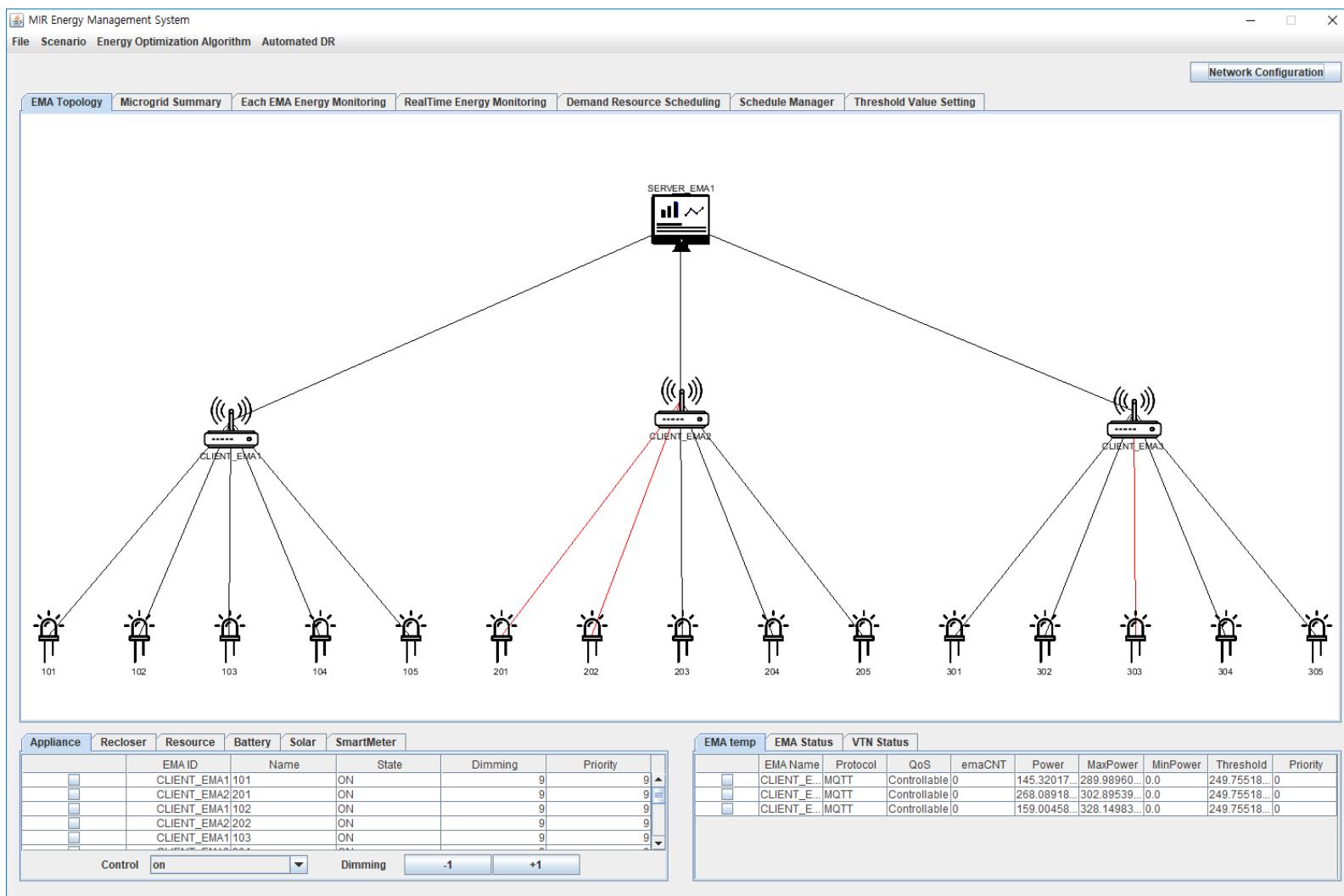
# 계층 구조 AS-IS

## SEMA : 1대 (SEMA1)

CEMA : 3

Device : CEMA 1대당 5대 ( $3 \times 5 = 15$ 대)

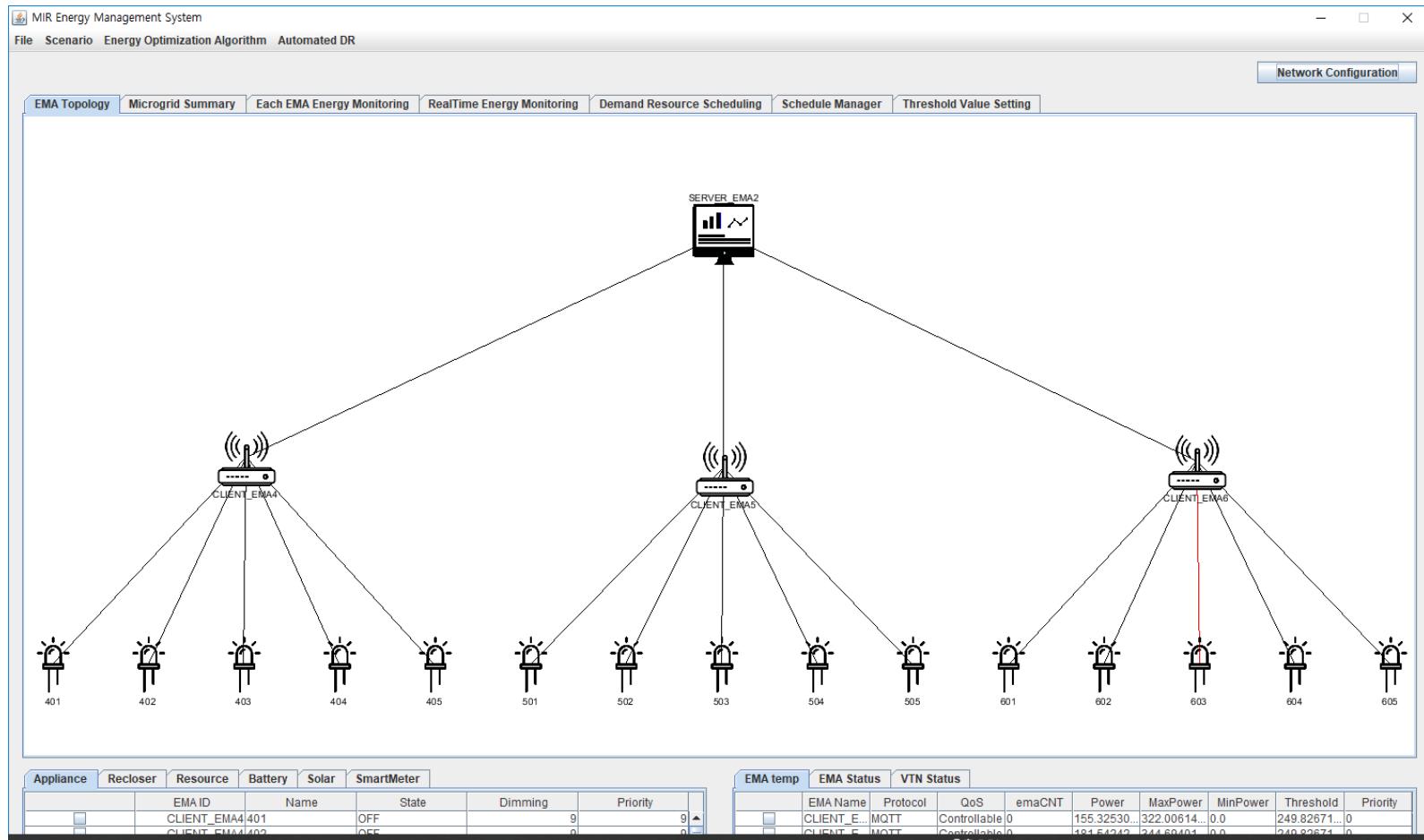
- Connect
  - Disconnect



# 계층 구조 AS-IS

SEMA : 1대 (SEMA2)  
CEMA : 3대  
Device : CEMA 1대당 5대 ( $3 \times 5 = 15$ 대)

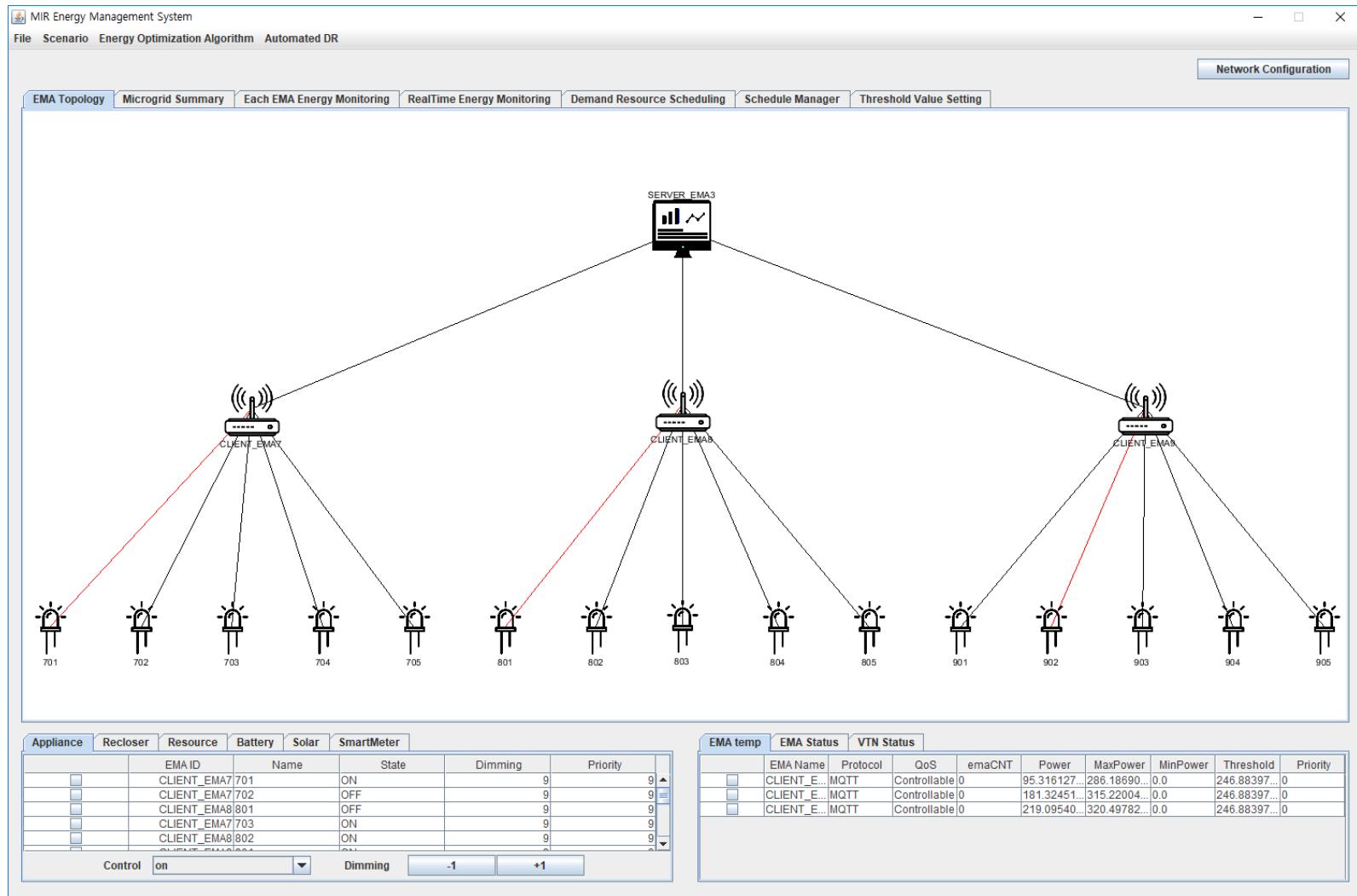
- Connect
- Disconnect



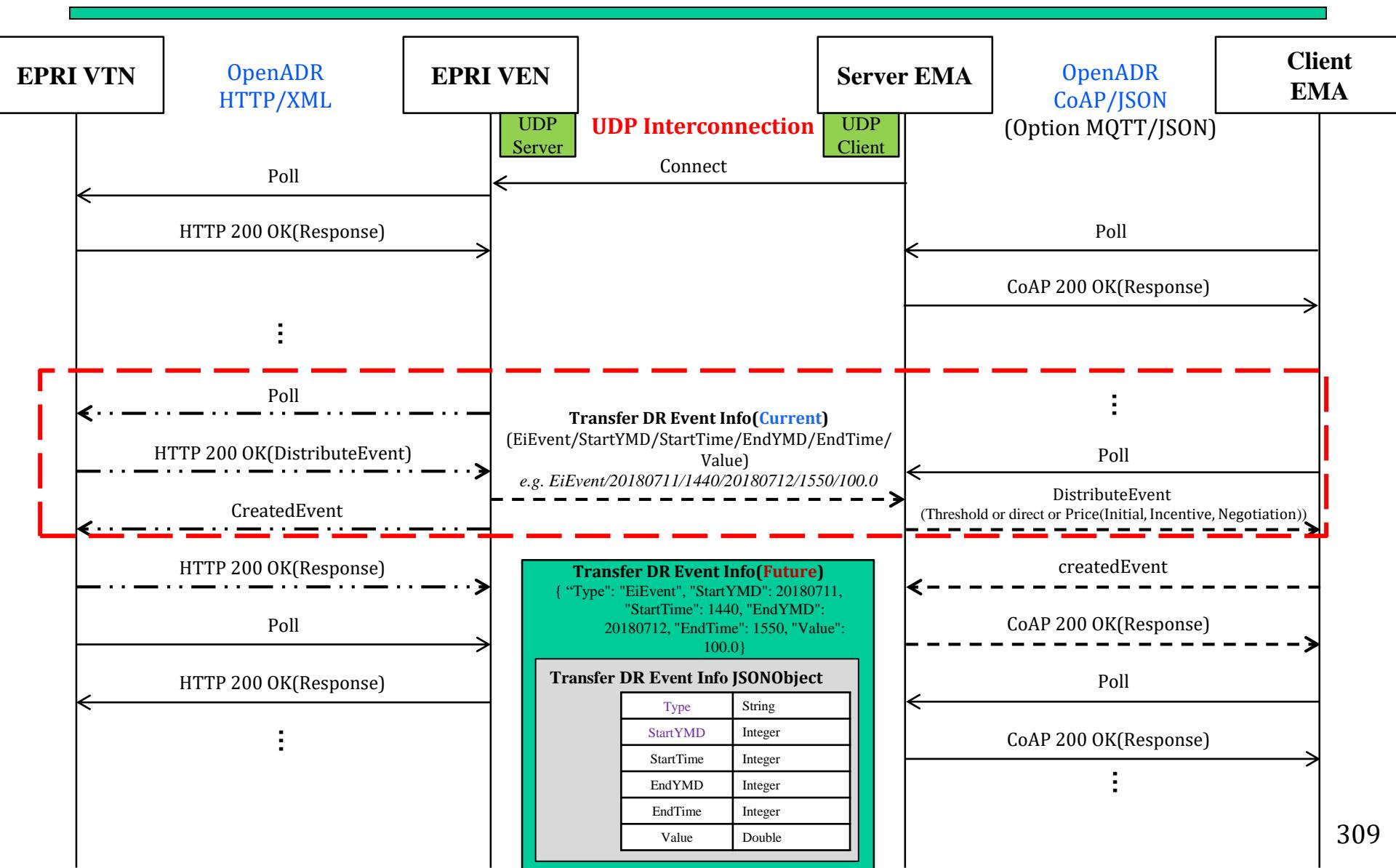
# 계층 구조 AS-IS

SEMA : 1대 (SEMA3)  
 CEMA : 3대  
 Device : CEMA 1대당 5대 ( $3 \times 5 = 15$ 대)

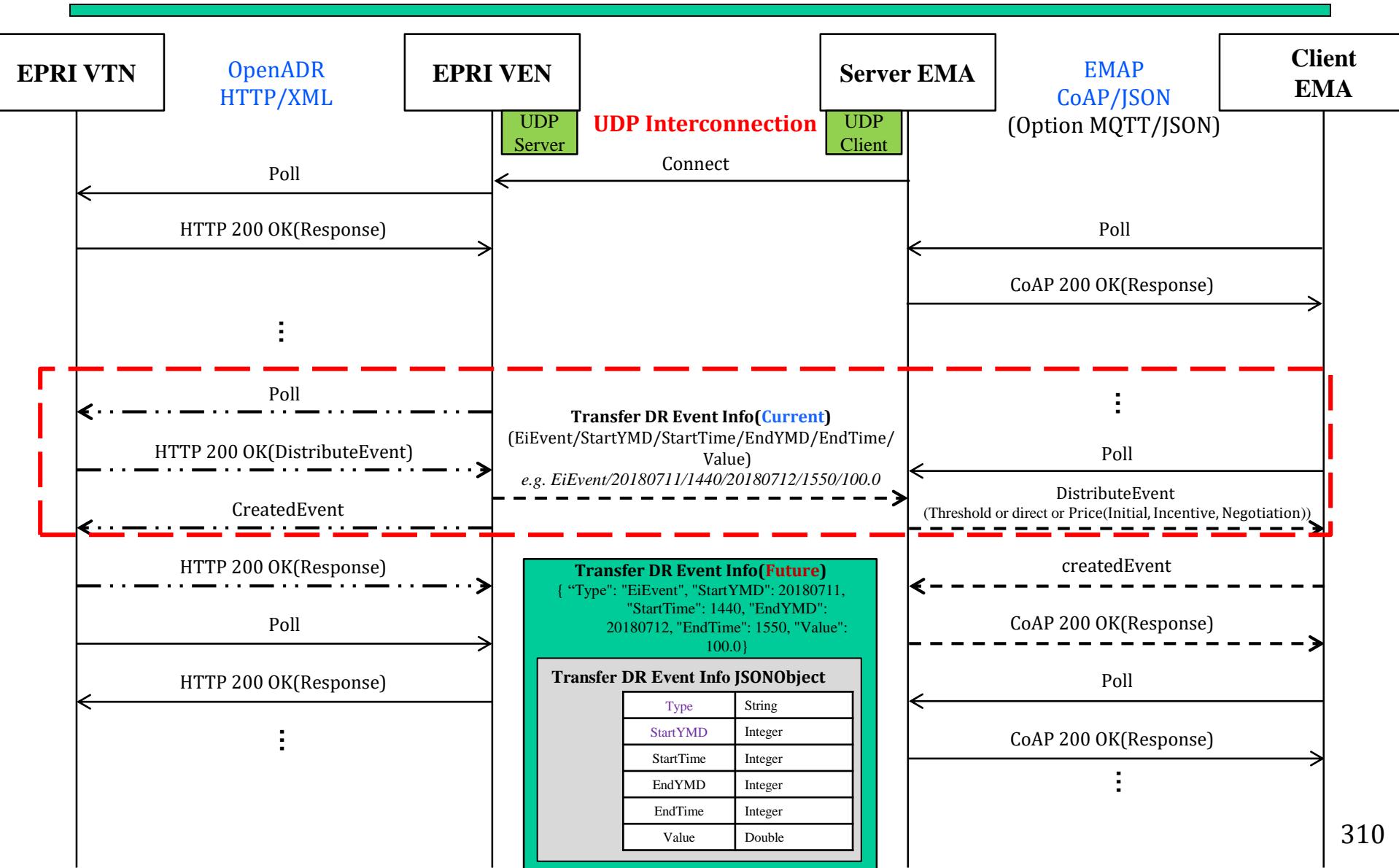
- Connect
- Disconnect



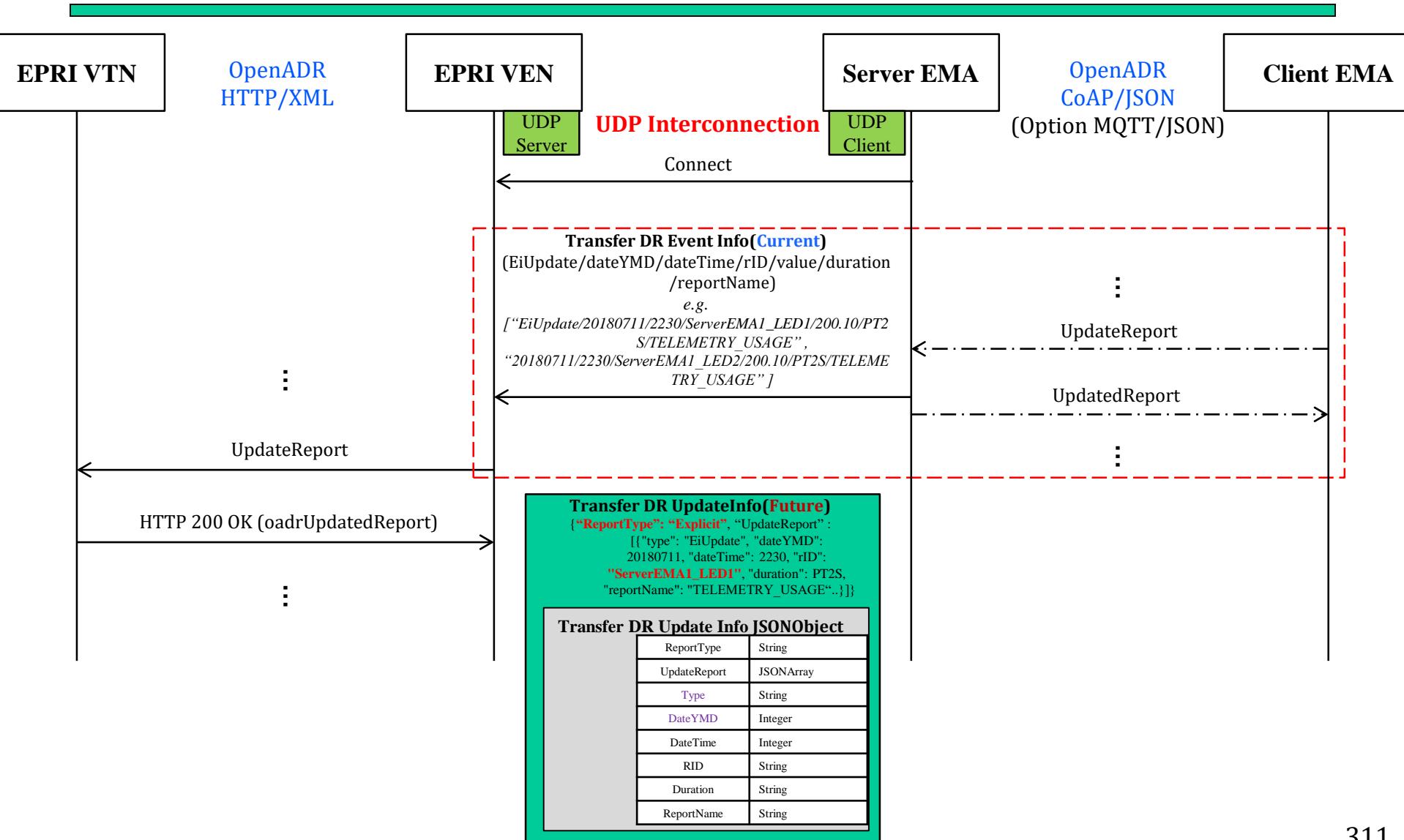
# Demand Response Event(Lightweight OpenADR)



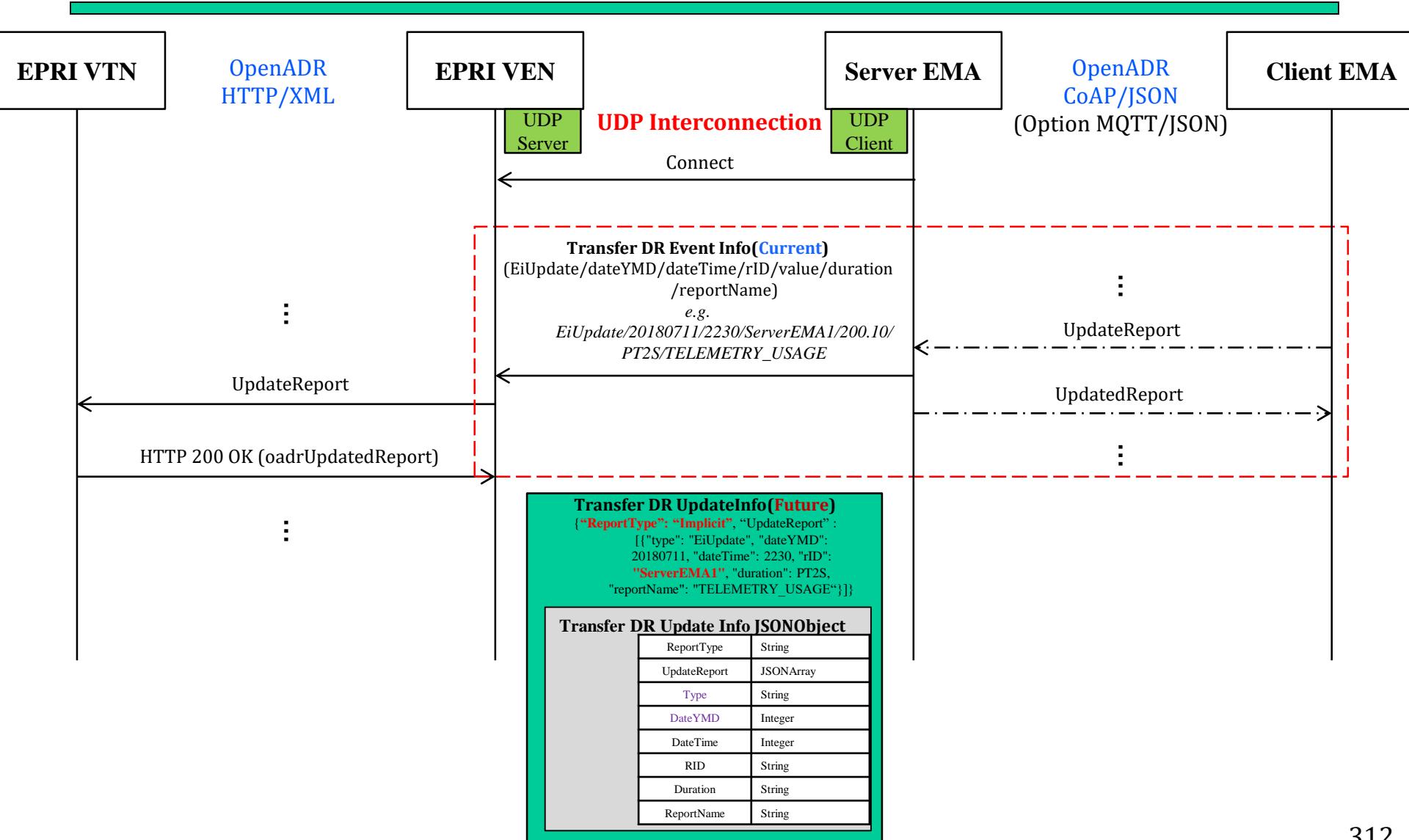
# Demand Response Event(EMAP)



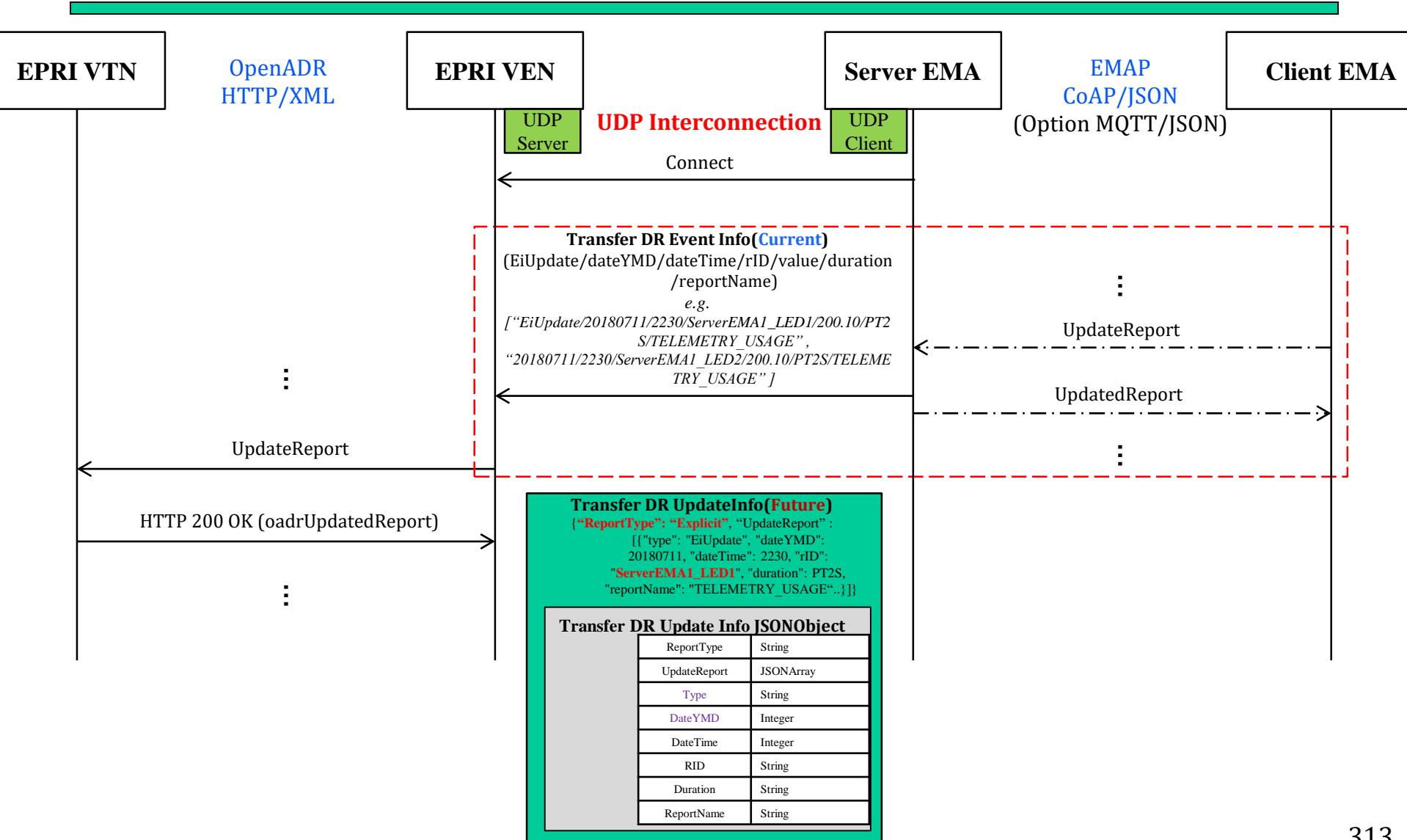
# Update Report(Lightweight OpenADR - Explicit)



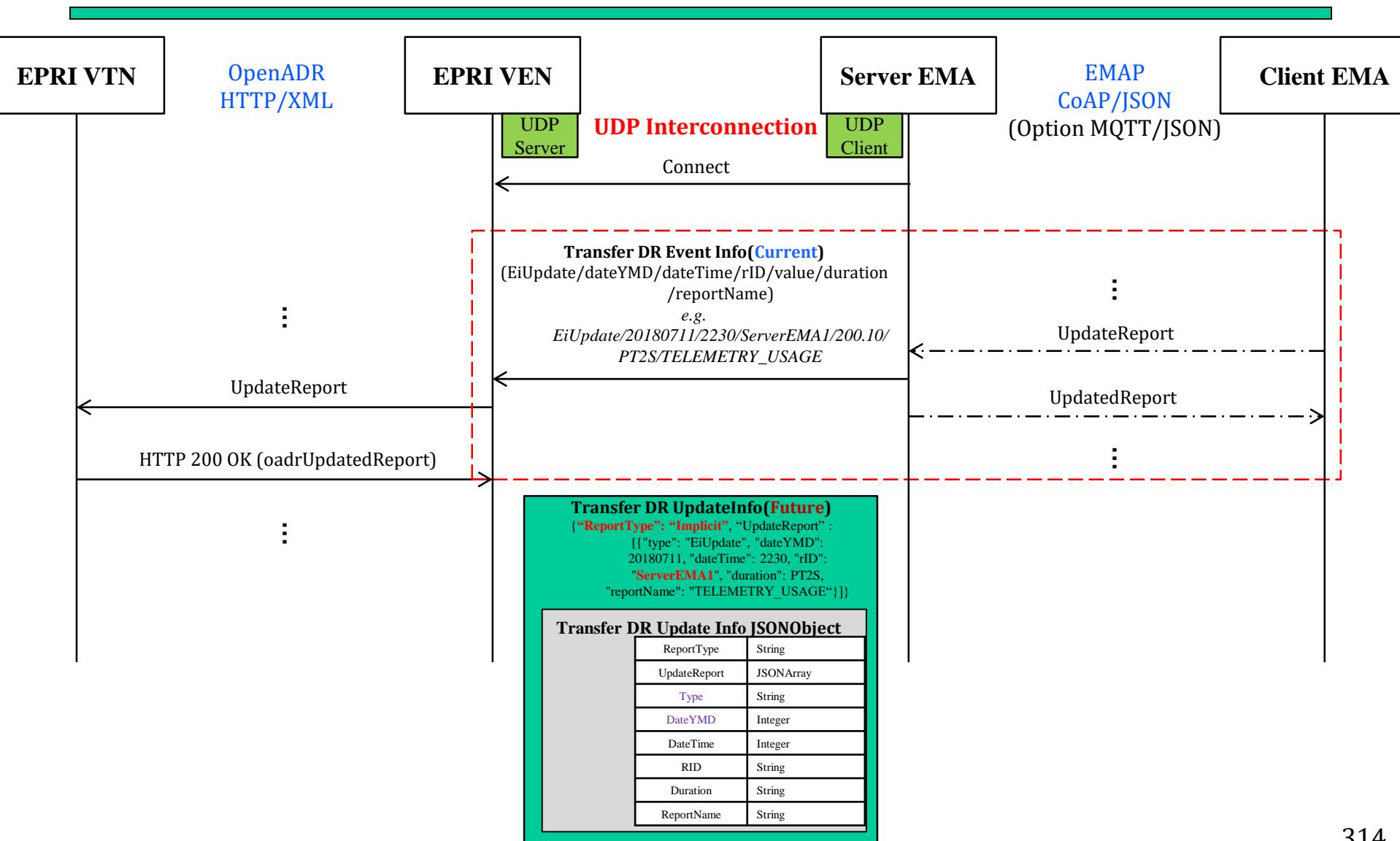
# Update Report(Lightweight OpenADR - Implicit)



# Update Report(EMAP - Explicit)



# Update Report(EMAP - Implicit)



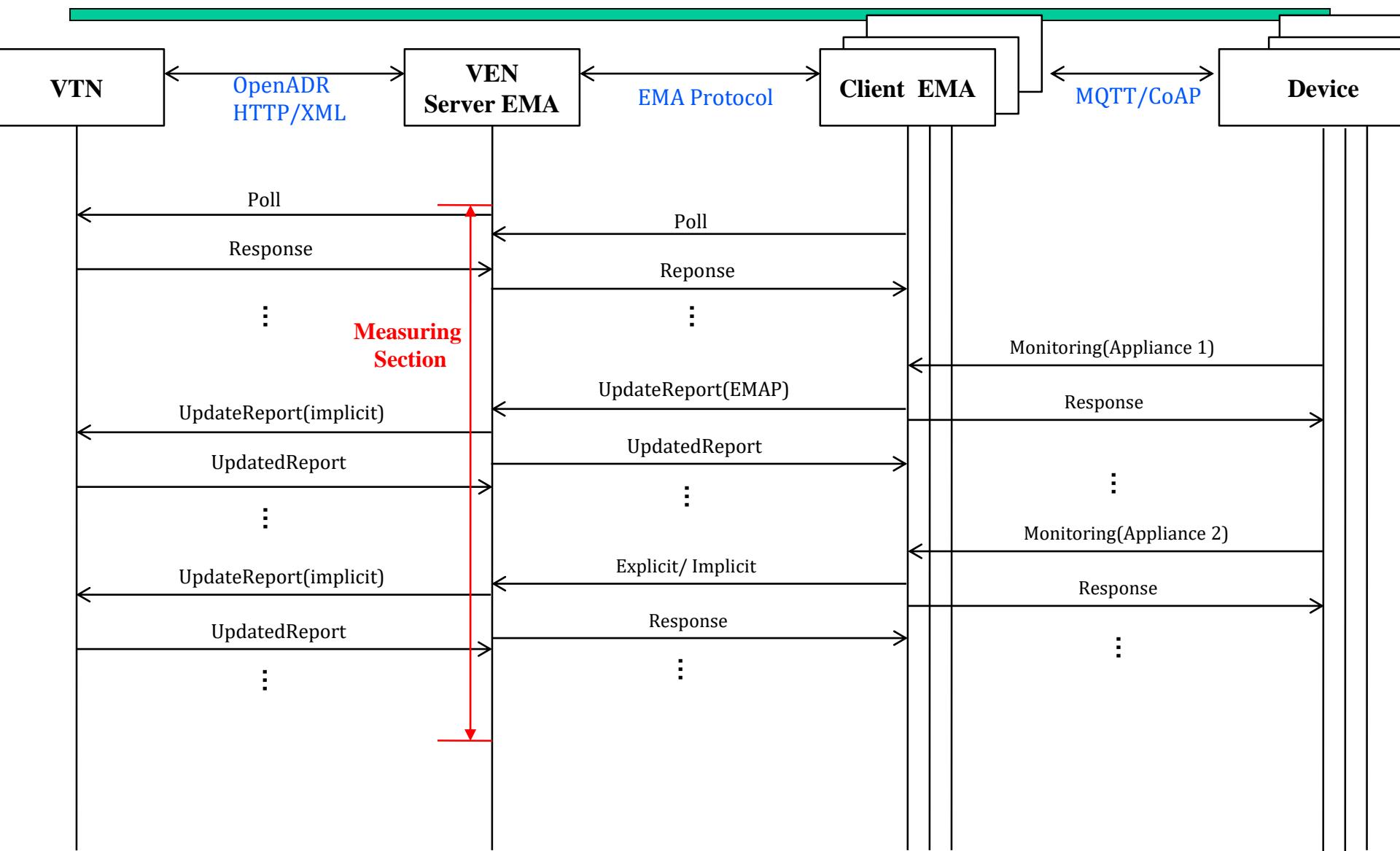
# AS-IS

## 실험 측정 방법

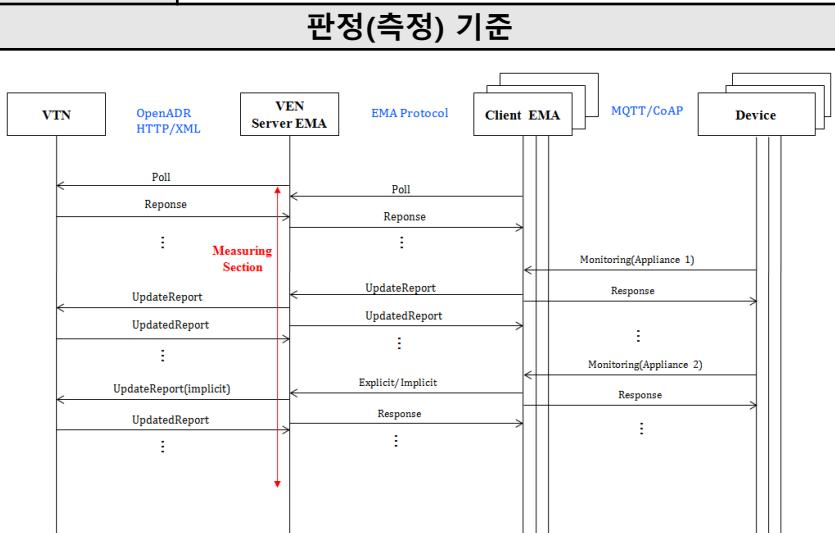
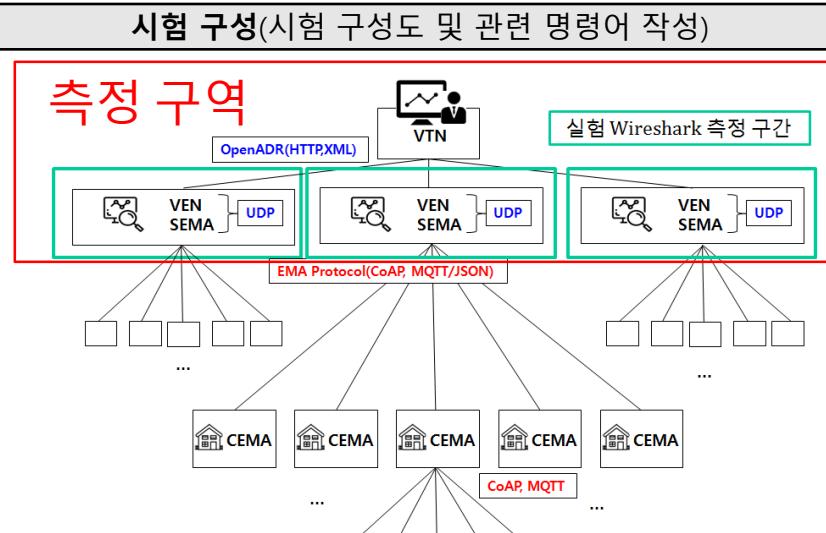
---

1. OpenADR2.0b HTTP/XML+EMA Protocol 이벤트 응답 시간
2. VTN - [VEN/SEMA 3대] - CEMA [1,3,5대] 데이터 트래픽 측정
  - Poll, UpdateReport, Explicit/Implicit의 데이터 트래픽 비교
  - 총 CEMA 개수는 3, 9, 15대
3. VTN - SEMA - CEMA 이벤트 응답시간 측정
  - VTN에서 SEMA 까지 Polling 주기 1,3,5sec
  - SEMA에서 CEMA까지 Polling 주기 1,3,5sec

# Message Flow(1)



# Experiment Procedure

항목 번호		시험 일자		시 험 자	
대 항 목		증 항 목		소 항 목	
목적	실험 측정 구간 SEMA 데이터 트래픽(OpenADR-HTTP/XML)				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. VTN 1대, VEN+ServerEMA 3대를 실행시킨다.</li> <li>3. Wireshark 실행한다.</li> <li>4. VEN을 통해 VTN과 HTTP/XML에 대한 구간을 측정한다.</li> <li>5. 주기적인 Poll/Reponse 메시지, UpdateReport의 트래픽 양을 SEMA에서 측정한다.</li> <li>6. SEMA에서 Wireshark로 60초 동안 측정한다. (Wireshark 종료 옵션)</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준	<p>판정(측정) 기준</p> 				
시험 구성(시험 구성도 및 관련 명령어 작성)	<p>시험 구성(시험 구성도 및 관련 명령어 작성)</p> 				
판정	비고				

# Experiment Procedure

AS-IS-SAMPLE(172).pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

xml || coap

No.	Time	Source	Destination	Protocol	Info
113...	221.35528845	192.168.1.115	192.168.1.175	CoAP	CON, MID:523578, T10, /EMAP/SERVER_EMA3/1.0b/col (application)
113...	222.135924939	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
113...	222.173318939	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
113...	222.177077196	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
113...	222.228324751	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
113...	222.257153775	192.168.1.187	192.168.1.174	CoAP	CON, MID:21186, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.259860624	192.168.1.110	192.168.1.174	CoAP	CON, MID:19114, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.272834923	192.168.1.108	192.168.1.174	CoAP	CON, MID:63583, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.275155582	192.168.1.109	192.168.1.174	CoAP	CON, MID:6277, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.293367852	192.168.1.106	192.168.1.174	CoAP	CON, MID:24280, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.300150102	192.168.1.110	192.168.1.174	CoAP	CON, MID:19115, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.328148871	192.168.1.187	192.168.1.174	CoAP	CON, MID:21187, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.345245883	192.168.1.108	192.168.1.174	CoAP	CON, MID:63584, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.358328950	192.168.1.109	192.168.1.174	CoAP	CON, MID:6278, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.374082176	192.168.1.106	192.168.1.174	CoAP	CON, MID:24281, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.414409412	192.168.1.101	192.168.1.172	CoAP	CON, MID:40074, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)
113...	222.415399588	192.168.1.172	192.168.1.101	CoAP	ACK, MID:40074, 2.05 Content (application/json)
113...	222.423766636	192.168.1.105	192.168.1.172	CoAP	CON, MID:29336, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)
113...	222.424558214	192.168.1.172	192.168.1.195	CoAP	ACK, MID:29336, 2.05 Content (application/json)
113...	222.432774726	192.168.1.104	192.168.1.172	CoAP	CON, MID:62500, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)
113...	222.433611475	192.168.1.172	192.168.1.104	CoAP	ACK, MID:62500, 2.05 Content (application/json)
113...	222.436261802	192.168.1.103	192.168.1.172	CoAP	CON, MID:43398, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)
113...	222.436990368	192.168.1.172	192.168.1.103	CoAP	ACK, MID:43398, 2.05 Content (application/json)
114...	222.440330073	192.168.1.102	192.168.1.172	CoAP	CON, MID:12749, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)

```

< p1:oadrUpdateReport
  xmlns:p3="http://docs.oasis-open.org/ns/energyinterop/201110"
  p3:schemaVersion="2.0b"
  xmlns:p2="http://docs.oasis-open.org/ns/energyinterop/201110/payments">
  <p2:requestID>
    <p1:oadrReport
      xmlns:p3="urn:ietf:params:xml:ns:icalendar-2.0:stream"
      xmlns:p4="http://docs.oasis-open.org/ns/energyinterop/201110">
        <p3:intervals
        <p4:reportRequestID>
        <p4:reportSpecifierID>
        <p4:reportName>
        <p4:createdDateTime>
        </p1:oadrReport>
      <p3:venID>
        0a0f2f61e2e4d1989397
    
```

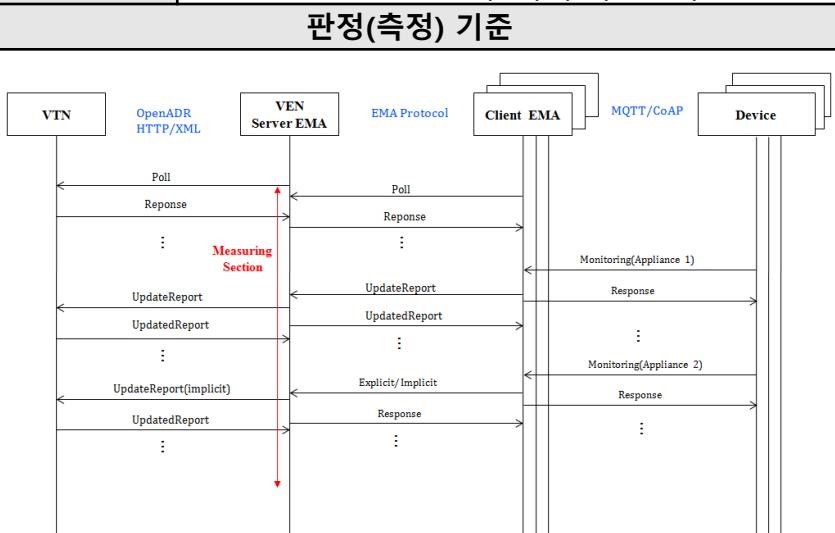
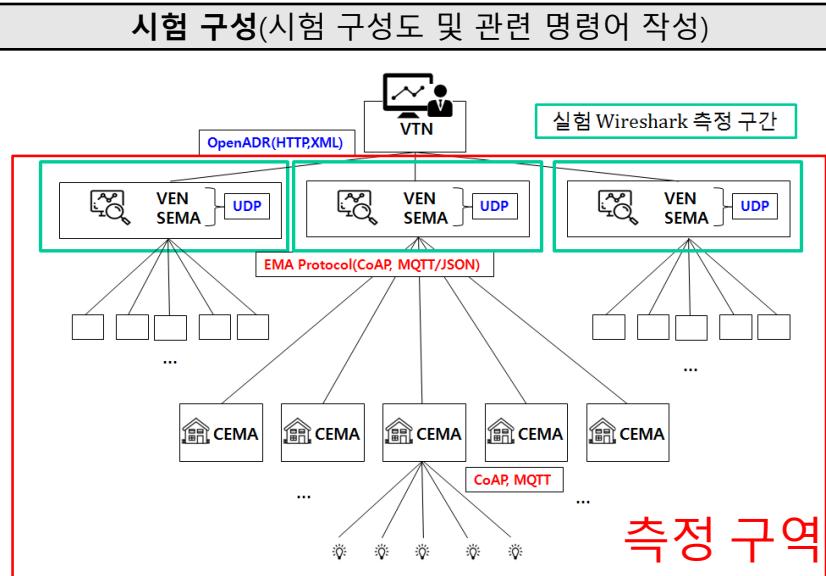
Frame (2041 bytes) Reassembled TCP (15146 bytes)

Tag (xml.tag), 41 bytes

Packets: 13706 Displayed: 6944 (50.7%)

Profile: Default

# Experiment Procedure

항목 번호		시험 일자		시 험 자	
대 항 목		증 항 목		소 항 목	
목적	실험 측정 구간 SEMA 데이터 트래픽(EMAP/CoAP, MQTT 사용)				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. VTN 1대, VEN+Server EMA 3대를 실행시킨다.</li> <li>3. Wireshark 실행한다.</li> <li>4. 각 Client EMA 5대(실험에 따라 1, 3, 5대)를 실행한다.           <ul style="list-style-type: none"> <li>• Client EMA starter를 이용하여 각 EMA들을 가능한 동시에 실행</li> <li>• 각 Client EMA의 가상 Device 5대를 실행한다.</li> </ul> </li> <li>5. 주기적인 Poll/Reponse 메시지, Explicit/Implicit의 트래픽 양을 SEMA에서 측정한다.</li> <li>6. SEMA에서 Wireshark로 60초 동안 측정한다. (Wireshark 종료 옵션)</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준	<p>판정(측정) 기준</p> 				
시험 구성(시험 구성도 및 관련 명령어 작성)	<p>시험 구성(시험 구성도 및 관련 명령어 작성)</p> 				
판정					
비고					

# Experiment Procedure

AS-IS-SAMPLE(172).pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

xml || coap

No.	Time	Source	Destination	Protocol	Info
113...	221.955920845	192.168.1.113	192.168.1.175	CoAP	CON, MID:62970, PUT, /EMAP/SERVER_EMA3/1.0b/Poll (application)
113...	222.135924939	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
113...	222.173318939	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
113...	222.177077196	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
113...	222.228324751	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
113...	222.257153775	192.168.1.107	192.168.1.174	CoAP	CON, MID:21186, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.259860624	192.168.1.110	192.168.1.174	CoAP	CON, MID:19114, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.272834923	192.168.1.108	192.168.1.174	CoAP	CON, MID:63583, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.275155582	192.168.1.109	192.168.1.174	CoAP	CON, MID:6277, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.293367852	192.168.1.106	192.168.1.174	CoAP	CON, MID:24280, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application)
113...	222.300150102	192.168.1.110	192.168.1.174	CoAP	CON, MID:19115, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.328148871	192.168.1.107	192.168.1.174	CoAP	CON, MID:21187, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.345245883	192.168.1.108	192.168.1.174	CoAP	CON, MID:63584, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.358328950	192.168.1.109	192.168.1.174	CoAP	CON, MID:6278, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.374082176	192.168.1.106	192.168.1.174	CoAP	CON, MID:24281, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application)
113...	222.414409412	192.168.1.101	192.168.1.172	CoAP	CON, MID:40074, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)
113...	222.415399588	192.168.1.172	192.168.1.101	CoAP	ACK, MID:40074, 2.05 Content (application/json)
113...	222.423766336	192.168.1.105	192.168.1.172	CoAP	CON, MID:29336, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)
113...	222.424558214	192.168.1.172	192.168.1.105	CoAP	ACK, MID:29336, 2.05 Content (application/json)
113...	222.432774726	192.168.1.104	192.168.1.172	CoAP	CON, MID:62500, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)
113...	222.433611475	192.168.1.172	192.168.1.104	CoAP	ACK, MID:62500, 2.05 Content (application/json)
113...	222.436261802	192.168.1.103	192.168.1.172	CoAP	CON, MID:43398, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)
113...	222.436990368	192.168.1.172	192.168.1.103	CoAP	ACK, MID:43398, 2.05 Content (application/json)
114...	222.440330073	192.168.1.102	192.168.1.172	CoAP	CON, MID:12749, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application)

< >

```

> Linux cooked capture
> Internet Protocol Version 4, Src: 192.168.1.108, Dst: 192.168.1.174
> User Datagram Protocol, Src Port: 49297, Dst Port: 5683
> Constrained Application Protocol, Confirmable, PUT, MID:63583
> JavaScript Object Notation: application/json
  < Object
    < Member Key: SrcEMA
      String value: CLIENT_EMA8
      Key: SrcEMA
    > Member Key: DestEMA
    > Member Key: requestID
    > Member Key: service
    > Member Key: type
    > Member Key: time
    > Member Key: report
  
```

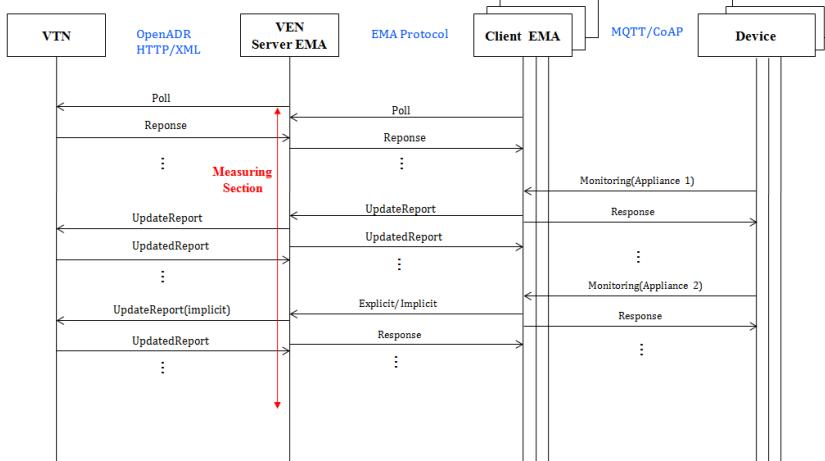
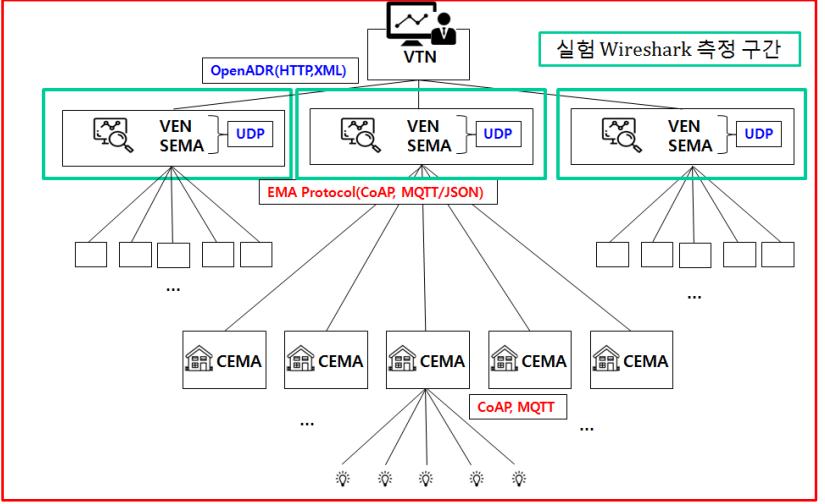
Frame (412 bytes) Reassembled IPv4 (336 bytes)

JSON object member (json.member), 23 bytes

Packets: 13706 · Displayed: 6944 (50.7%)

Profile: Default

# Experiment Procedure

항목 번호		시험 일자		시 험 자	
대 항 목		증 항 목		소 항 목	
목적	계층 실험 측정 구간 VTN -SEMA-CEMA 데이터 트래픽 측정 (OpenADR- HTTP/XML, EMAP/CoAP, MQTT 사용)				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. VTN 1대, VEN+Server EMA 3대를 실행시킨다.</li> <li>3. Wireshark 실행한다.</li> <li>4. 각 Client EMA 5대(실험에 따라1, 3, 5대)를 실행한다.           <ul style="list-style-type: none"> <li>• Client EMA starter를 이용하여 각 EMA들을 가능한 동시에 실행</li> <li>• 각 Client EMA의 가상 Device 5대를 실행한다.</li> </ul> </li> <li>5. SEMA에서 VTN – SEMA(VEN)까지의 데이터 트래픽(HTTP/XML)을 측정한다.</li> <li>6. 주기적인 Poll/Reponse 메시지, 트래픽 양을 SEMA(EMAP/CoAP, MQTT)에서 측정한다.</li> <li>7. SEMA에서 Wireshark로 60초 동안 측정한다. (Wireshark 종료 옵션)</li> <li>8. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준	시험 구성(시험 구성도 및 관련 명령어 작성)				
 <p>Sequence diagram illustrating the communication flow between VTN, VEN Server EMA, Client EMA, and Device. The process involves multiple steps including Poll, Response, Measuring Section, UpdateReport, and Monitoring. Protocols used include OpenADR (HTTP/XML), EMA Protocol, and MQTT/CoAP.</p>	 <p>Network diagram showing the experimental setup. The VTN layer at the top connects to three SEMA (VEN SEMA) nodes via UDP. Each SEMA node connects to five CEMA (House) nodes via EMA Protocol (CoAP, MQTT/JSON). The CEMA nodes are further connected to various sensors (light, temperature, etc.) at the bottom. A red box highlights the "실험 Wireshark 측정 구간" (Experimental Wireshark Measurement Interval) which covers the communication between VTN and SEMA layers.</p>				
판정	비고				

# Experiment Procedure

AS-IS-SAMPLE(172).pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

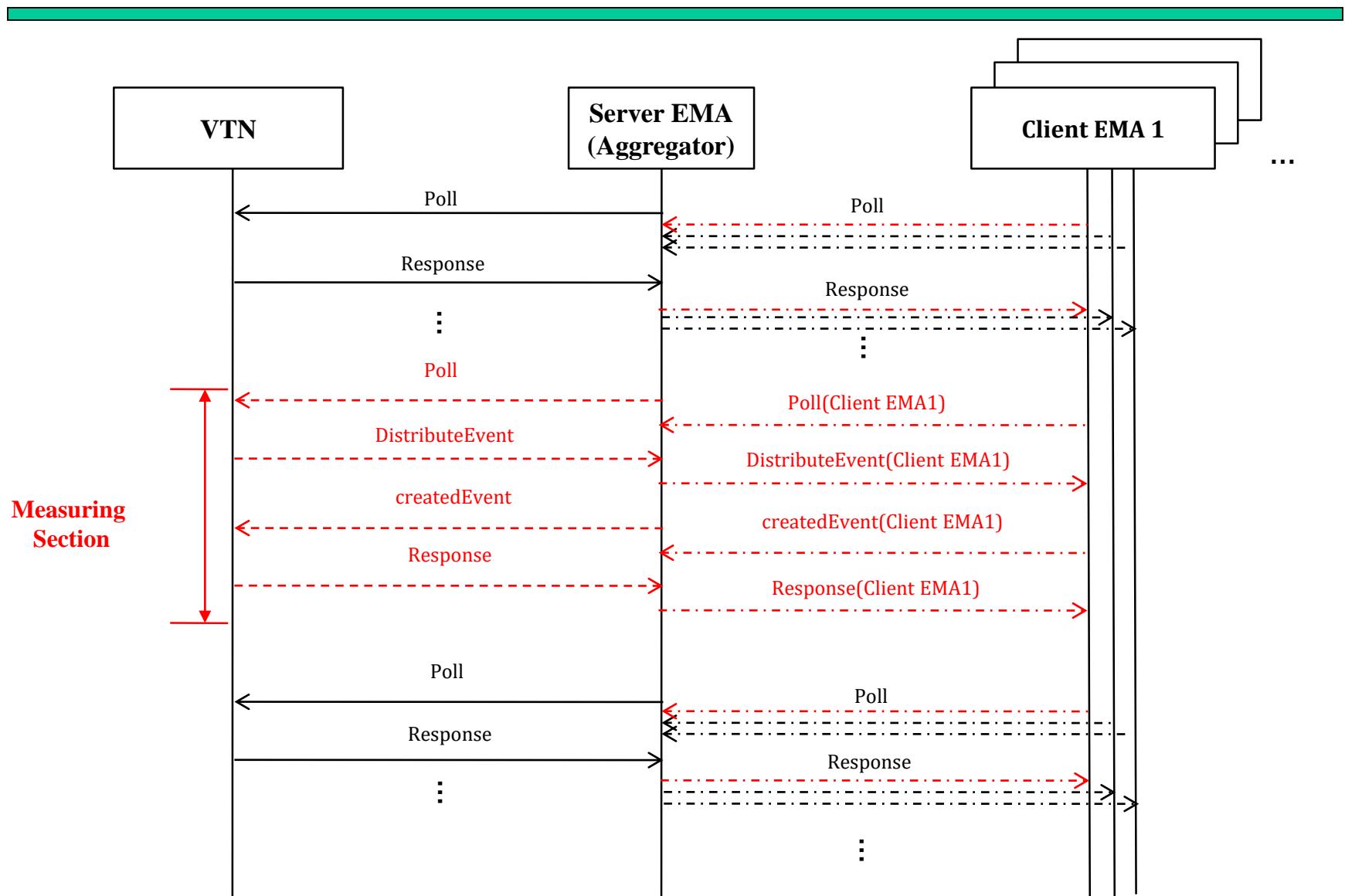
xml || coap

No.	Time	Source	Destination	Protocol	Info
113...	201.055030045	192.168.1.113	192.168.1.176	CoAP	CON, MID:20001, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application/json)
113...	222.135924939	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
113...	222.173318939	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
113...	222.177077196	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
113...	222.228324751	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
113...	222.257153775	192.168.1.107	192.168.1.174	CoAP	CON, MID:21186, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application/json)
113...	222.259860624	192.168.1.110	192.168.1.174	CoAP	CON, MID:19114, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application/json)
113...	222.272834923	192.168.1.108	192.168.1.174	CoAP	CON, MID:63583, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application/json)
113...	222.275155582	192.168.1.109	192.168.1.174	CoAP	CON, MID:6277, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application/json)
113...	222.293367852	192.168.1.106	192.168.1.174	CoAP	CON, MID:24280, PUT, /EMAP/SERVER_EMA1/1.0b/Report (application/json)
113...	222.300150102	192.168.1.110	192.168.1.174	CoAP	CON, MID:19115, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application/json)
113...	222.328148871	192.168.1.107	192.168.1.174	CoAP	CON, MID:21187, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application/json)
113...	222.345245883	192.168.1.108	192.168.1.174	CoAP	CON, MID:63584, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application/json)
113...	222.358328950	192.168.1.109	192.168.1.174	CoAP	CON, MID:6278, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application/json)
113...	222.374082176	192.168.1.106	192.168.1.174	CoAP	CON, MID:24281, PUT, /EMAP/SERVER_EMA1/1.0b/Poll (application/json)
113...	222.414409412	192.168.1.101	192.168.1.172	CoAP	CON, MID:40074, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application/json)
113...	222.415399588	192.168.1.172	192.168.1.101	CoAP	ACK, MID:40074, 2.05 Content (application/json)
113...	222.423766336	192.168.1.105	192.168.1.172	CoAP	CON, MID:29336, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application/json)
113...	222.424558214	192.168.1.172	192.168.1.105	CoAP	ACK, MID:29336, 2.05 Content (application/json)
113...	222.432774726	192.168.1.104	192.168.1.172	CoAP	CON, MID:62500, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application/json)
113...	222.433611475	192.168.1.172	192.168.1.104	CoAP	ACK, MID:62500, 2.05 Content (application/json)
113...	222.436261802	192.168.1.103	192.168.1.172	CoAP	CON, MID:43398, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application/json)
113...	222.436990368	192.168.1.172	192.168.1.103	CoAP	ACK, MID:43398, 2.05 Content (application/json)
114...	222.440330073	192.168.1.102	192.168.1.172	CoAP	CON, MID:12749, PUT, /EMAP/SERVER_EMA2/1.0b/Report (application/json)

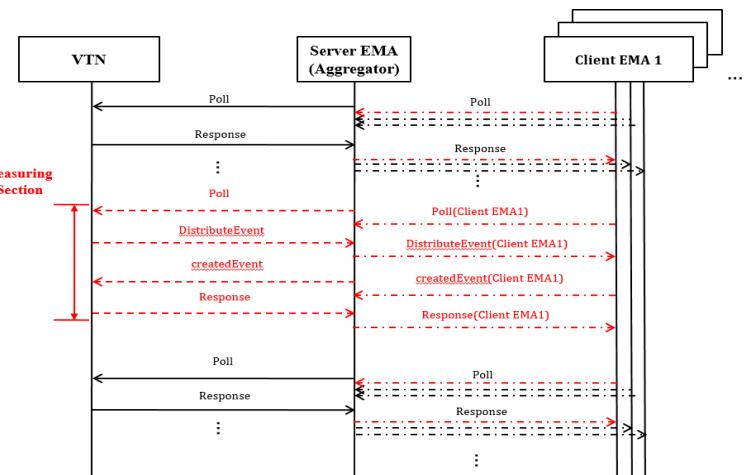
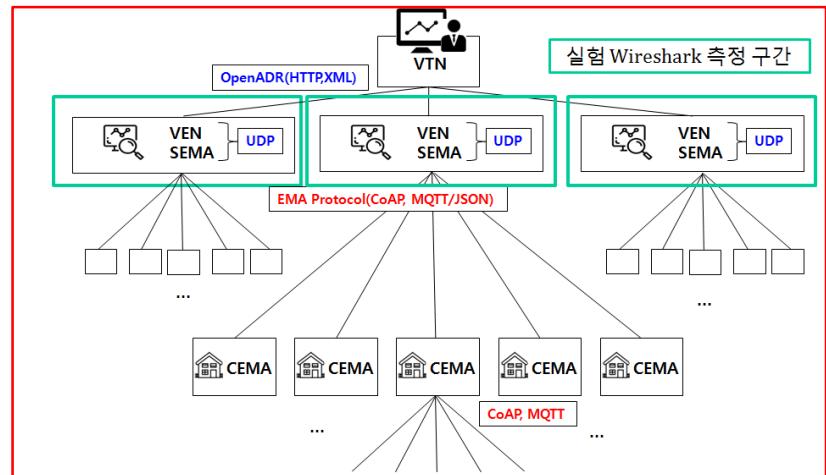
> Linux cooked capture  
> Internet Protocol Version 4, Src: 192.168.1.108, Dst: 192.168.1.174  
> User Datagram Protocol, Src Port: 49297, Dst Port: 5683  
> Constrained Application Protocol, Confirmable, PUT, MID:63583  
> JavaScript Object Notation: application/json  
 > Object  
 > Member Key: SrcEMA  
 String value: CLIENT\_EMA8  
 Key: SrcEMA  
 > Member Key: DestEMA  
 > Member Key: requestID  
 > Member Key: service  
 > Member Key: type  
 > Member Key: time  
 > Member Key: report

Frame (412 bytes) Reassembled IPv4 (336 bytes)  
 JSON object member (json.member), 23 bytes  
 Packets: 13706 · Displayed: 6944 (50.7%) · Profile: Default

# Message Flow(2)



# Experiment Procedure

항목 번호		시험 일자		시 험 자	
대 항 목		증 항 목		소 항 목	
목적	OpenADR2.0b HTTP/XML+EMA Protocol 이벤트 응답 시간 (VTN - VEN+SEMA - CEMA)				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. VTN 1대, VEN+Server EMA 3대를 실행시킨다.</li> <li>3. Wireshark 실행한다.</li> <li>4. 각 Client EMA 5대를 실행한다.           <ul style="list-style-type: none"> <li>• Client EMA starter를 이용하여 각 EMA들을 가능한 동시에 실행</li> <li>• 각 Client EMA의 가상 Device 5대를 실행한다.</li> </ul> </li> <li>5. VTN에서 해당 VEN으로 DR을 내리면 VEN이 UDP로 SEMA에게 DR을 전달, SEMA는 CEMA에게 DR을 내린다.           <ul style="list-style-type: none"> <li>• SEMA는 VTN에게 받은 이벤트를 1/N로 분배하여 CEMA에게 DR을 내린다.</li> </ul> </li> <li>6. Wireshark 측정을 종료한다.</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준	시험 구성(시험 구성도 및 관련 명령어 작성)				
 <p><b>Measuring Section</b></p>			 <p>실험 Wireshark 측정 구간</p>		
판정	비고				

# Experiment Procedure

Screenshot of Wireshark showing network traffic analysis. A red box highlights a sequence of 11 packets (108.. to 118..) from source 192.168.1.181 to destination 192.168.1.172. These packets represent a CoAP POST request to /OpenADR2/Simple/2.0b/OadrPoll followed by several EiEvent and EiReport responses.

No.	Time	Source	Destination	Protocol	Info
108..	216.745827318	192.168.1.181	192.168.1.174	HTTP/XML	HTTP/1.1 200 OK
109..	216.752459724	192.168.1.181	192.168.1.175	HTTP/XML	HTTP/1.1 200 OK
109..	216.789714056	192.168.1.181	192.168.1.174	HTTP/XML	HTTP/1.1 200 OK
109..	217.846629804	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
110.. *REF*	0.002966394	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
110..	0.002966394	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/EiEvent HTTP/1.1
110..	0.052657636	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
110..	0.056820121	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
110..	0.131974048	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
110..	0.669839885	192.168.1.172	192.168.1.102	CoAP	ACK, MID:12741, 2.05 Content (application/json)
110..	0.680070092	192.168.1.102	192.168.1.172	CoAP	CON, MID:12742, PUT, /EMAP/SERVER_EMA2/1.0b/Event (application
110..	0.942024695	192.168.1.181	192.168.1.175	HTTP/XML	HTTP/1.1 200 OK
110..	0.976263729	192.168.1.181	192.168.1.174	HTTP/XML	HTTP/1.1 200 OK
111..	0.993879755	192.168.1.181	192.168.1.175	HTTP/XML	HTTP/1.1 200 OK
111..	1.029808163	192.168.1.181	192.168.1.174	HTTP/XML	HTTP/1.1 200 OK
111..	2.134342460	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
111..	2.197254124	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
111..	2.200700819	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
111..	2.246045142	192.168.1.181	192.168.1.172	HTTP/XML	HTTP/1.1 200 OK
112..	3.070539105	192.168.1.181	192.168.1.175	HTTP/XML	HTTP/1.1 200 OK
112..	3.121628969	192.168.1.181	192.168.1.174	HTTP/XML	HTTP/1.1 200 OK
112..	3.183438371	192.168.1.181	192.168.1.175	HTTP/XML	HTTP/1.1 200 OK
112..	3.201099704	192.168.1.181	192.168.1.174	HTTP/XML	HTTP/1.1 200 OK
113..	4.248651890	192.168.1.172	192.168.1.181	HTTP/XML	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1

Selected packet details:

```

version="1.0"
encoding="UTF-8"
standalone="yes"
?>
<ns2:oadrPayload
  xmlns="http://www.w3.org/2000/09/xmldsig#"
  xmlns:ns2="http://openadr.org/oadr-2.0b/2012/07"
  xmlns:ns3="http://docs.oasis-open.org/ns/energyinterop/201110"
  xmlns:ns4="http://docs.oasis-open.org/ns/energyinterop/201110/payments"
  xmlns:ns5="urn:ietf:params:xml:ns:icalendar-2.0"
  xmlns:ns6="http://docs.oasis-open.org/ns/emix/2011/06"
  xmlns:ns7="urn:ietf:params:xml:ns:icalendar-2.0:stream"
  xmlns:ns8="http://www.w3.org/2005/Atom"
  xmlns:ns9="http://docs.oasis-open.org/ns/emix/2011/06/power"
  xmlns:ns10="http://www.opengis.net/gml/3.2"
  xmlns:ns11="http://docs.oasis-open.org/ns/emix/2011/06/siscale"
  
```

Selected bytes:

```
0000  00 00 00 01 00 00 06 08 00  27 fe 25 6b 74 61 08 00  . .... : %kta..
```

Selected hex:

```
0000  00 00 00 01 00 00 06 08 00  27 fe 25 6b 74 61 08 00  . .... : %kta..
```

Selected ASCII:

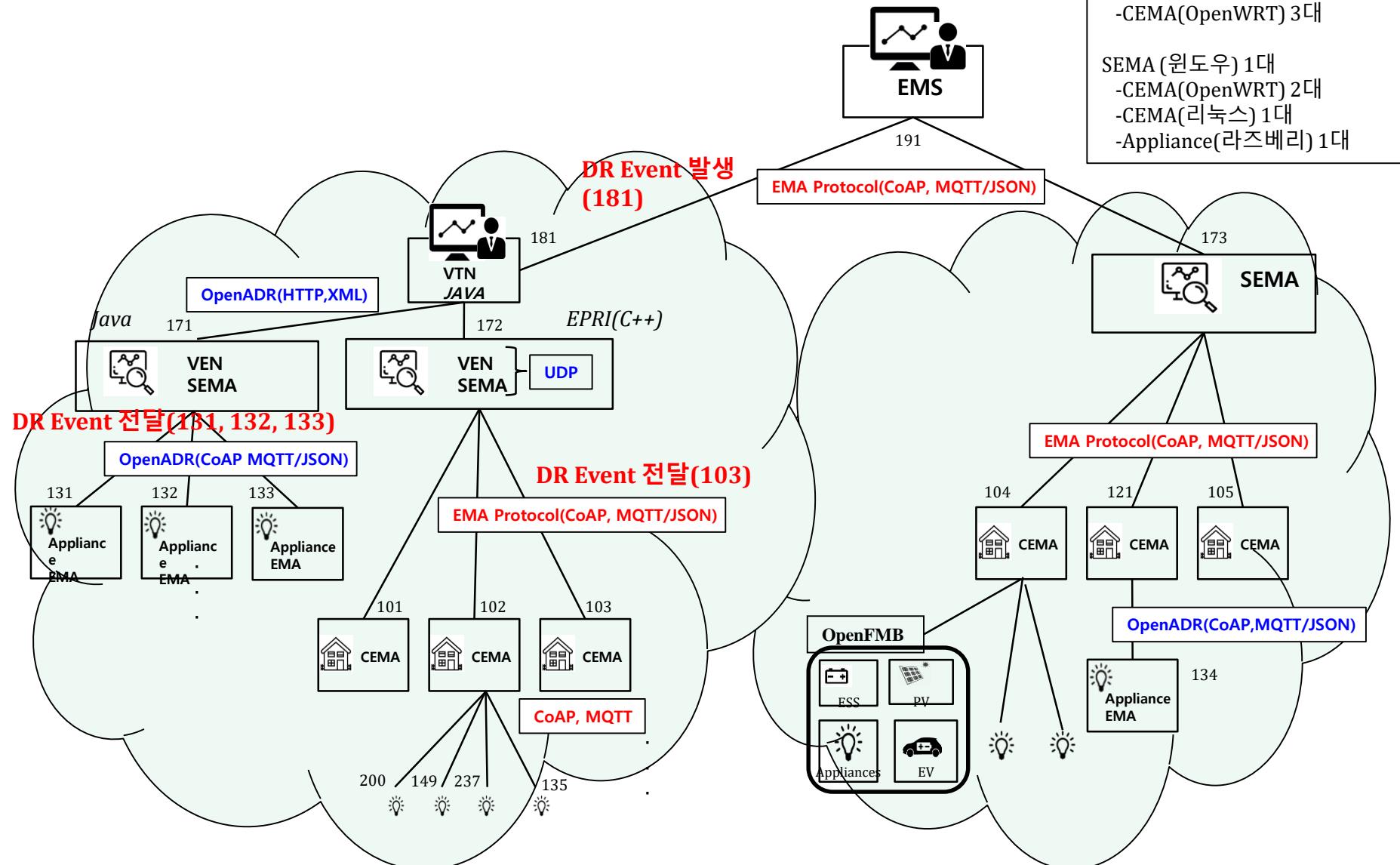
```
0000  JSON object member (json.member), 23 bytes
```

Statistics:

```
Packets: 13706 · Displayed: 804 (5.9%) · Marked: 2 (0.0%) · Profile: Default
```

# 계층 구조

## Smart Energy Framework



# 계층 구조

## Smart Energy Framework

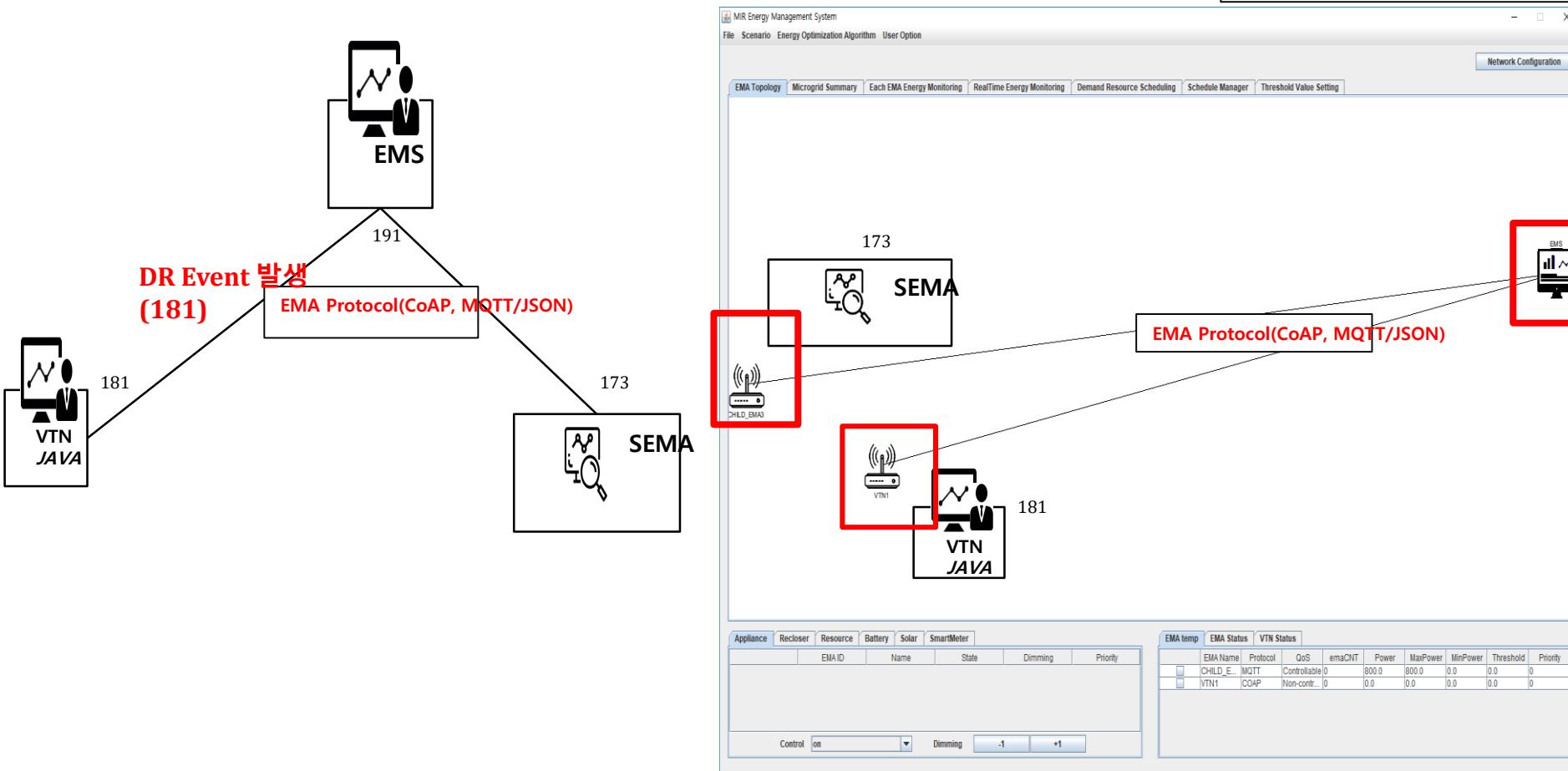
**EMS 1대 (Window)**

**VTN 1대 (리눅스)**

- SEMA+VEN (리눅스) 1대
- Appliance(라즈베리) 3대
- SEMA+VEN\_Java(윈도우) 1대
- CEMA(OpenWRT) 3대

**SEMA (윈도우) 1대**

- CEMA(OpenWRT) 2대
- CEMA(리눅스) 1대
- Appliance(라즈베리) 1대



# 계층 구조

## Smart Energy Framework

IP : 192.168.1.xxx

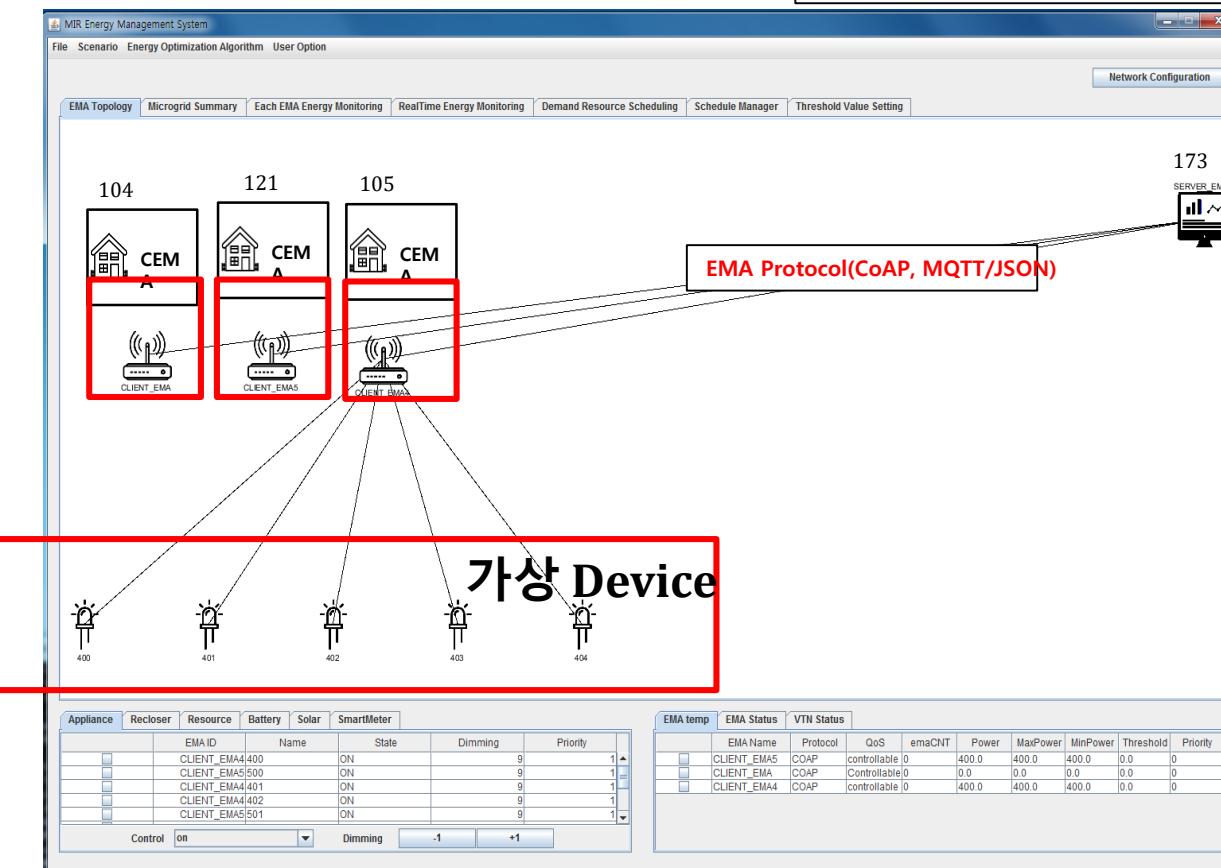
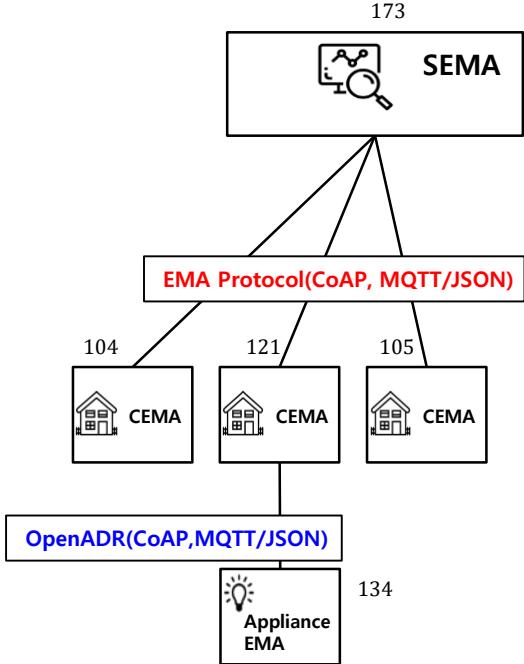
EMS 1대 (Window)

VTN 1대 (리눅스)

- SEMA+VEN (리눅스) 1대
- Appliance(라즈베리) 3대
- SEMA+VEN\_Java(윈도우) 1대
- CEMA(OpenWRT) 3대

SEMA (윈도우) 1대

- CEMA(OpenWRT) 2대**
- CEMA(리눅스) 1대**
- Appliance(라즈베리) 1대**



IP : 192.168.1.xxx

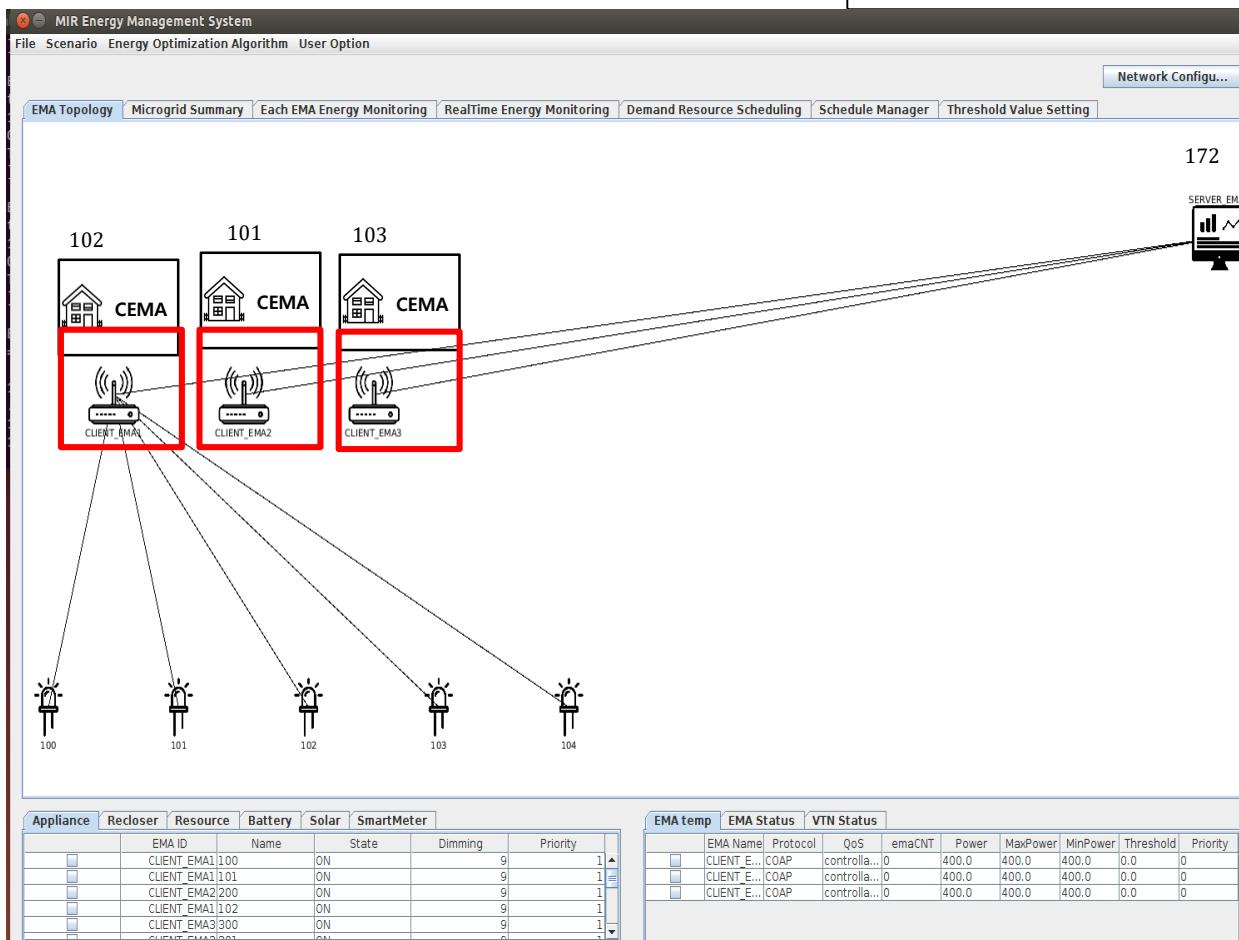
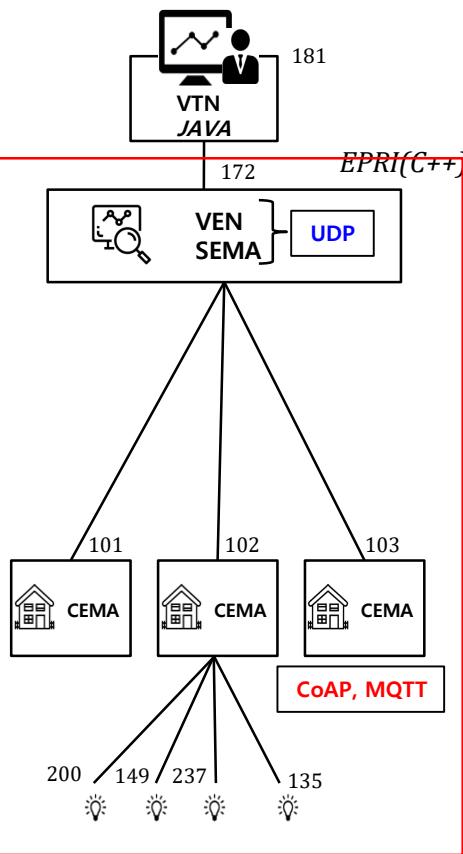
EMS 1대 (Window)

VTN 1대 (리눅스)

- SEMA+VEN (리눅스) 1대
- Appliance(라즈베리) 3대
- SEMA+VEN\_Java(윈도우) 1대
- CEMA(OpenWRT) 3대

SEMA (윈도우) 1대

- CEMA(OpenWRT) 2대
- CEMA(리눅스) 1대
- Appliance(라즈베리) 1대



# 계층 구조

## Smart Energy Framework

IP : 192.168.1.xxx

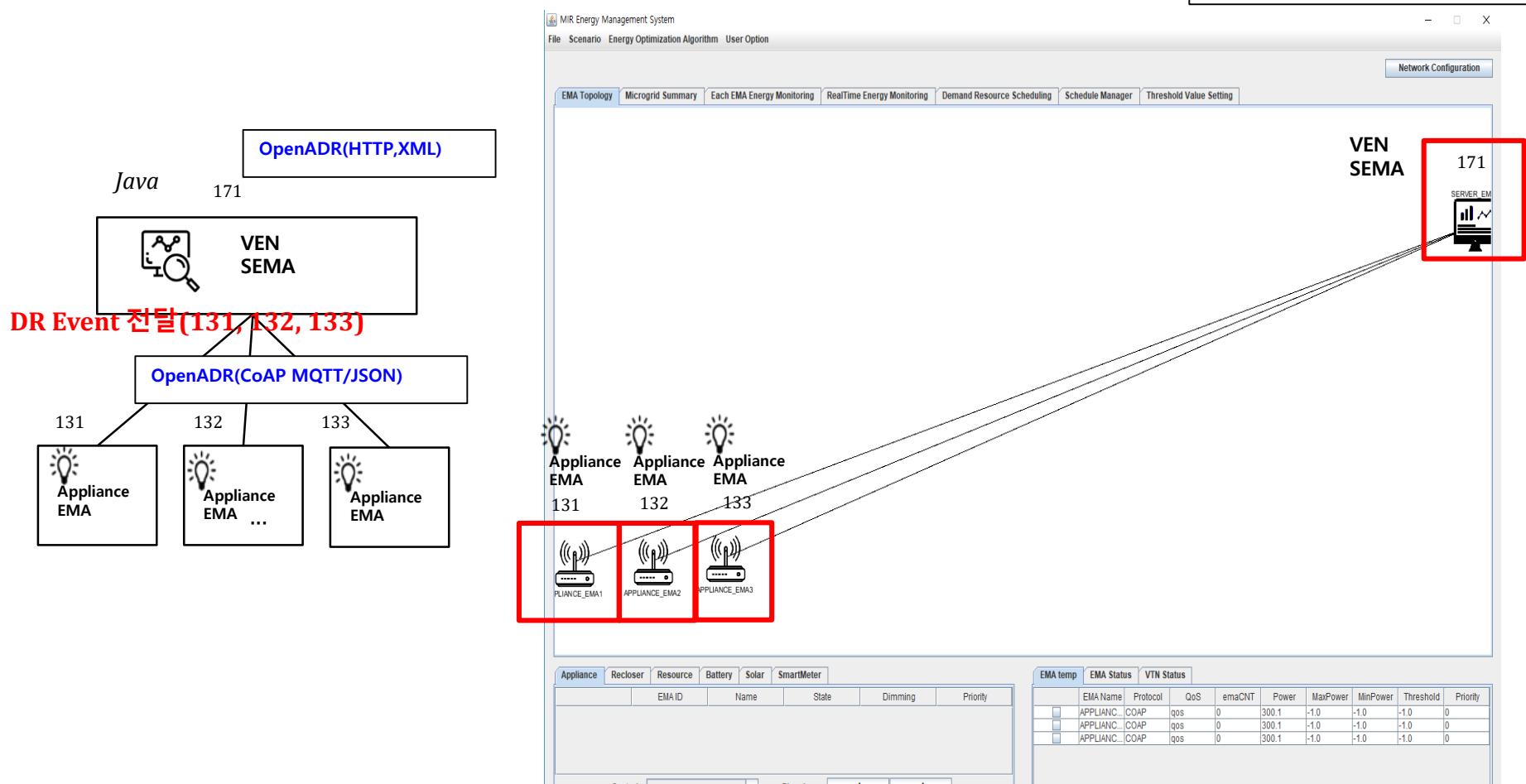
EMS 1대 (Window)

VTN 1대 (리눅스)

- SEMA+VEN (리눅스) 1대
- Appliance(라즈베리) 3대
- SEMA+VEN\_Java(윈도우) 1대
- CEMA(OpenWRT) 3대

SEMA (윈도우) 1대

- CEMA(OpenWRT) 2대
- CEMA(리눅스) 1대
- Appliance(라즈베리) 1대



# Energy Framework

## 실험 측정 방법

➤ OpenADR2.0b HTTP/XML+EMA Protocol 계층 실험

### 1. 데이터 트래픽 측정 (Poll, UpdateReport)

- EMS – VTN – [VEN-SEMA 1대] – ApplianceEMA [3대] 데이터 트래픽 측정
- EMS – VTN – [VEN-SEMA(UDP)] – CEMA 3대 – Appliance 데이터 트래픽 측정
- EMS – SEMA – CEMA(3대) – ApplianceEMA (1대) 데이터 트래픽 측정

### 2. 이벤트 응답시간 측정 (EMS에서 응답 시간 측정)

- EMS – VTN – [VEN-SEAM] – ApplianceEMA 3대 (OpenADR)
- EMS – VTN – [VEN-SEMA(UDP)] – CEMA 3대 – Appliance
- EMS – SEMA – CEMA(3대) – ApplianceEMA (1대)

## Message Flow(1) – Traffic

192.168.1.191

173

181

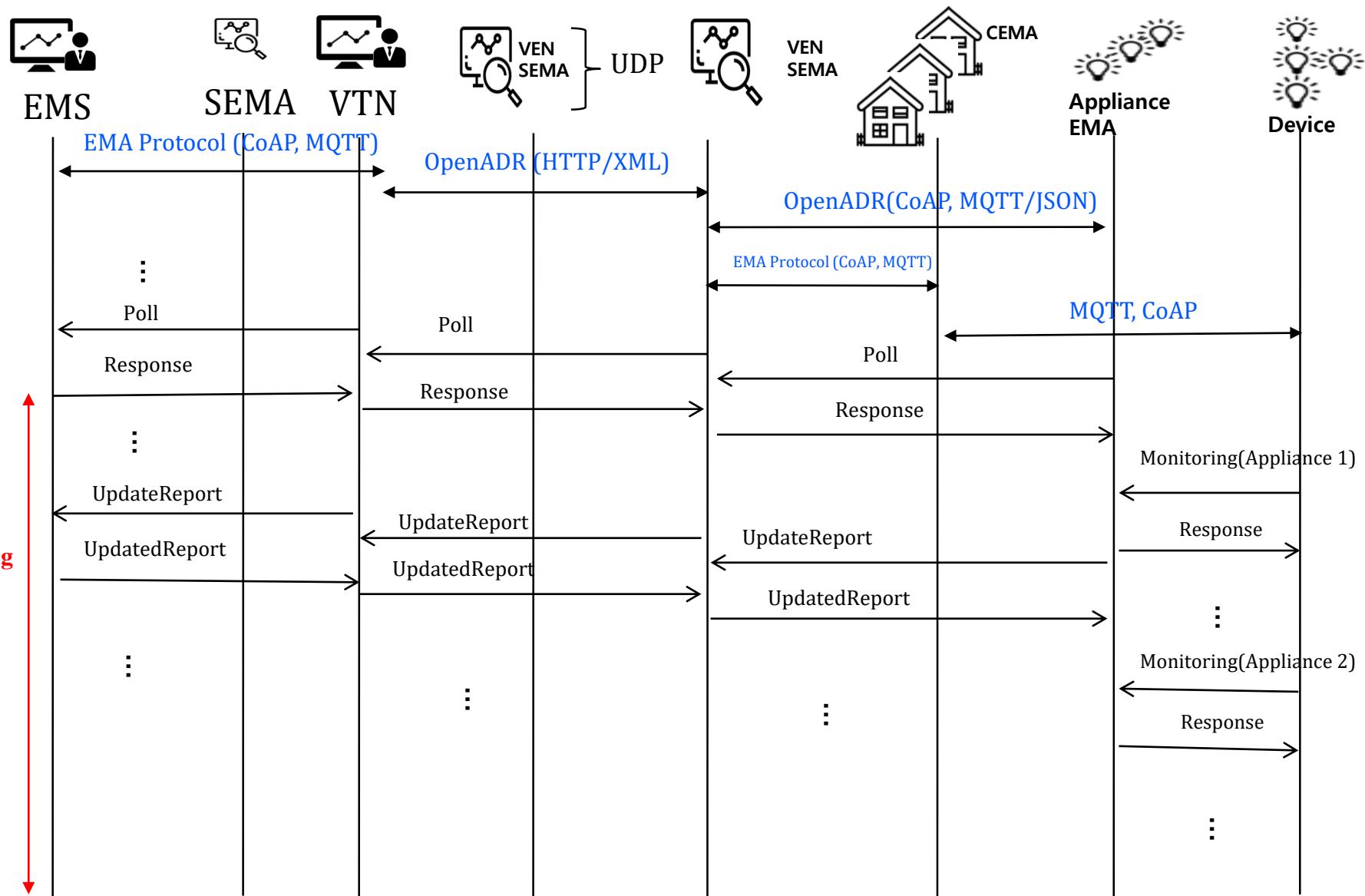
172

171

101~103

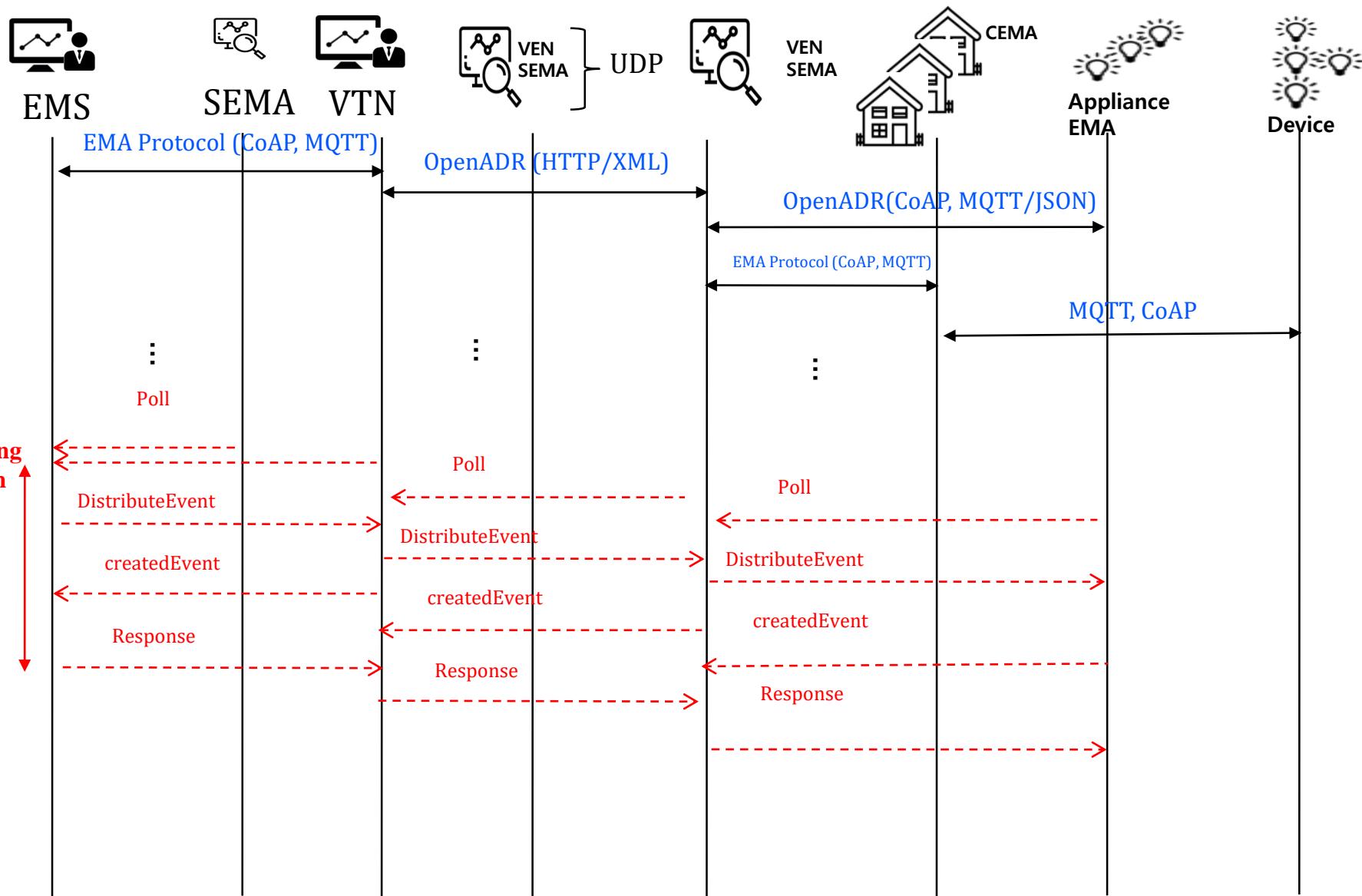
131~134

135, 149,  
200, 237

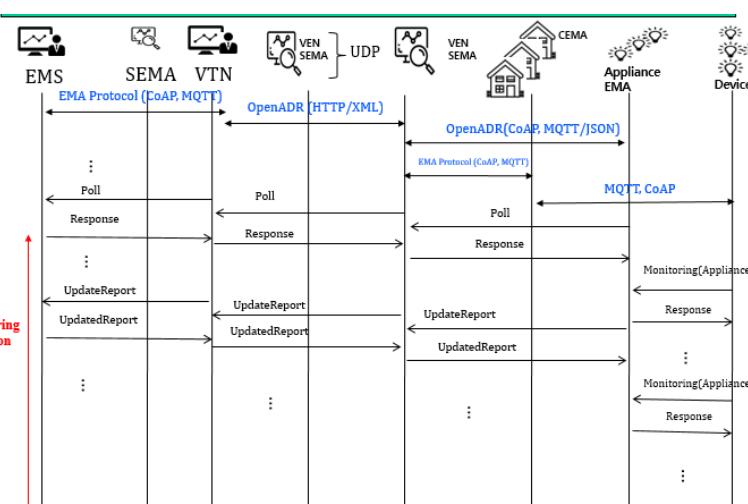
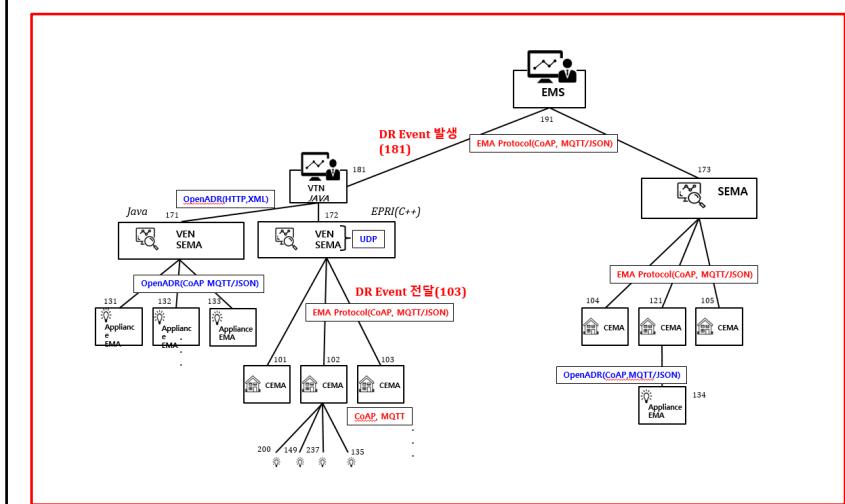


# Message Flow(2) \_Event Response Time

192.168.1.191 173 181 172 171 101~103 131~134 135, 149,  
200, 237

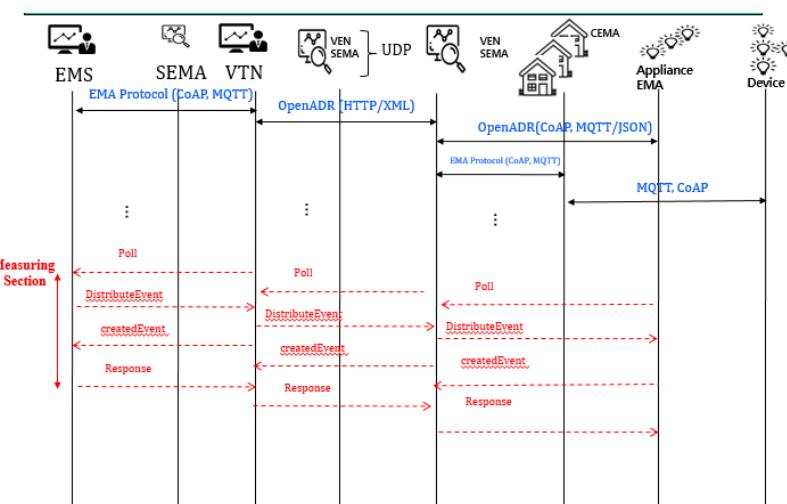
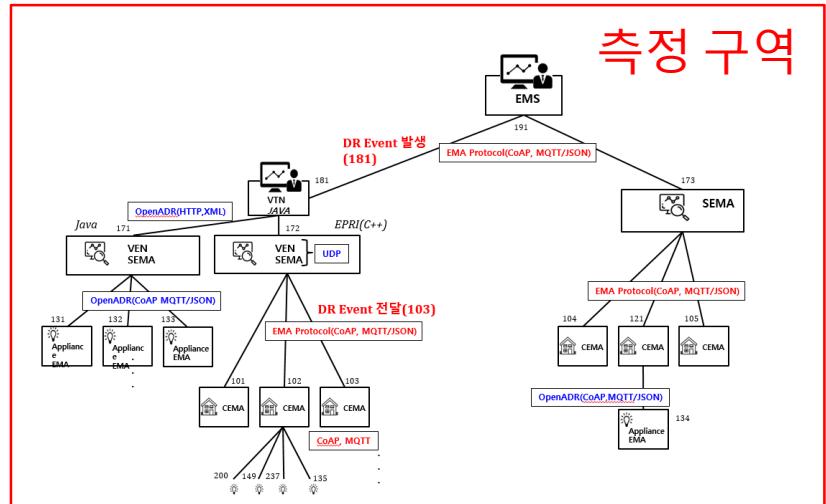


# Experiment Procedure

항목 번호		시험 일자	<th>시 험 자</th> <td></td>	시 험 자	
대 항 목		증 항 목	<th>소 항 목</th> <td></td>	소 항 목	
목 적	실험 측정 구간 SEMA 데이터 트래픽				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. EMS 1대, VTN 1대, SEMA 3대, CEMA 6대, ApplianceEMA 4대, Device 4대를 실행한다.</li> <li>3. Wireshark 실행한다.</li> <li>4. EMS 각 계층별 VTN JAVA, SEMA의 하위 계층 부터 데이터 트래픽을 구한다.</li> <li>5. 주기적인 Poll/Response 메시지, UpdateReport의 트래픽 양을 SEMA에서 측정한다.</li> <li>6. SEMA에서 Wireshark로 60초 동안 측정한다. (Wireshark 종료 옵션)</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준			시험 구성(시험 구성도 및 관련 명령어 작성)		
 <p>Measuring Section</p>					
판정	비고				

측정

# Experiment Procedure

항목 번호		시험 일자		시 험 자	
대 항 목		증 항 목	<th>소 항 목</th> <td></td>	소 항 목	
목 적	OpenADR2.0b HTTP/XML+EMA Protocol 이벤트 응답 시간 (VTN - VEN+SEMA - CEMA)				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. EMS 1대, VTN 1대, SEMA 3대, CEMA 6대, ApplianceEMA 4대, Device 4대를 실행한다.</li> <li>3. Wireshark 실행한다.</li> <li>4. 각 Client EMA 5대를 실행한다.           <ul style="list-style-type: none"> <li>• Client EMA starter를 이용하여 각 EMA들을 가능한 동시에 실행</li> <li>• 각 Client EMA의 가상 Device 5대를 실행한다.</li> </ul> </li> <li>5. VTN에서 해당 VEN으로 DR을 내리면 VEN이 UDP로 SEMA에게 DR을 전달, SEMA는 CEMA에게 DR을 내린다.           <ul style="list-style-type: none"> <li>• SEMA는 VTN에게 받은 이벤트를 1/N로 분배하여 CEMA에게 DR을 내린다.</li> </ul> </li> <li>6. Wireshark 측정을 종료한다.</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준	시험 구성(시험 구성도 및 관련 명령어 작성)				
 <p>Measuring Section</p>	 <p>측정 구역</p>				
판정	비고				

# 계층 실험 계획서

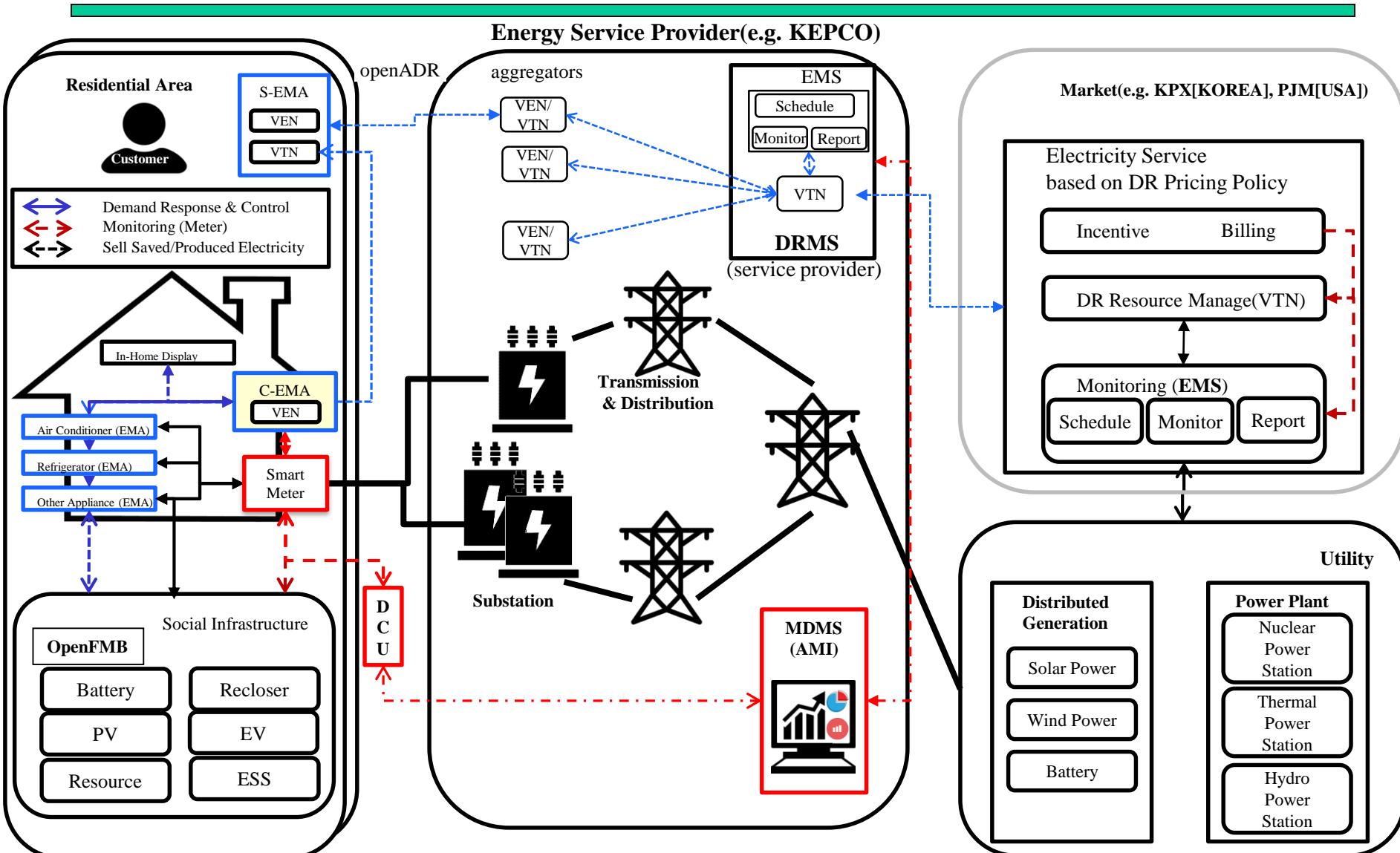
## DEMO

Hanyang University  
MIR-Lab



# Background

# Smart Energy Framework including Homes



# Smart Home overall Terms

## 스마트 에너지 홈 (Smart Energy Home)

: 스마트 에너지 홈이란 전력 기술과 ICT 기술이 융합되어 탄생한 지능형 전력망 기술이 댁내 환경에 도입됨에 따라, 생산자 중심으로 이루어지는 에너지의 관리 기능을 소비자 중심으로 전환하여 소비자 측면에서의 에너지 효율성 향상 및 이를 위한 서비스 제공이 가능하도록 하는 기술.

## 지능형 홈 네트워크

: 홈네트워크를 기본으로 댁내에 존재하는 정보 가전 기기들을 통합, 연결하는 동시에 외부에 존재하는 여러 종류의 서비스 네트워크들과의 연결성을 제공하여 누구나 기기, 시간, 장소에 구애받지 않고 다양한 컨텐츠와 서비스를 제공할 수 있는 환경으로, 지능형 홈, 홈네트워크, 디지털 홈, 스마트 홈 환경과 인프라를 종합하는 개념

## OpenADR(Automated Demand Response)

: 지능형 DR(Demand Response)에 적용되는 표준 통신 프로토콜 특히 전력공급자와 소비자 간 양방향 통신을 가능케 함으로써 송전과 배전의 효율성을 극대화하는 게 주목적. OpenADR 물리적 통신방식이 아닌 명령, 요청 등과 같은 DR 신호 교환을 위한 응용계층의 표준기술로써 OpenADR을 적용함으로써 전력공급자가 실시간으로 수요관리 명령을 내렸을 때, 소비자가 이에 대응해 자동으로 전기사용량을 줄이는 지능형 DR을 구현

## OpenFMB(Field Message Bus)

: OpenFMB는 그리드 말단의 상호운용성(interoperability)과 분산지능화(distributed intelligence)를 위한 프레임워크이자 참고 아키텍쳐

## OpenHAB(Open Home Automation Bus)

: 홈 자동화에 대한 모든 것들의 전반적인 통합 플랫폼을 제공하기 위한 프로젝트

## Demand Response

: 스마트그리드에서 수요반응이란 Utility company와 ISO, third part service provider가 소비자들의 최대 전기 사용량 차이를 완화시키고, 전기 공급과 수요의 균형을 맞추기 위하여 일시적으로 전기 사용량을 축소시키기 위해 소비자의 전력사용패턴 변화를 유도하는 기술

# Energy Management System

---

## 에너지 관리 시스템(Energy Management System; EMS)

: 에너지소비자에게 에너지 절감 및 에너지효율화를 포함한 통합적인 에너지관리 서비스를 제공하기 위한 시스템

## 분산 전원 시스템(Distributed Energy Resources; DER)

: 중앙집중 형 발전에서 벗어나, 수요자 주변에 발전소를 분산 배치하여 마이크로그리드 망에서 전력 수송을 효율적으로 해 주는 차세대 시스템

## BAS(Building Automation System)

: 전력, 공조, 냉난방, 조명 설비 등과 같은 건물에너지 설비에 대한 상태 감시 및 자동화 된 감시 조작 시스템으로 건물 설비에 대한 자동화 운용 및 중앙 감시를 목적으로 하는 시스템

## Home Gateway

: 다양한 가정 내 통신 기기들을 외부의 통신망과의 통신을 위해 연결하기 위한 가정 내 네트워크 장치로 외부와의 통신뿐만 아니라 가정 내의 통신 기기들의 통신을 지원

## SCADA(Supervisory Control And Data Acquisition)

: ICT 기술을 이용하여 산업 공정, 기반시설, 설비를 바탕으로 한 작업 공정을 감시하고 제어하는 시스템

## AMI(Advanced Metering Infrastructure)

: 기존의 인력에 의한 검침 방법의 검침 효율 및 정확성 결여 등의 문제를 개선하고자 제안 된 방법

## IHID(In Home Display)

: AMI의 주요 구성요소로서 고객의 건물 내에서 AMI와 연계하여 에너지 사용 및 통계에 대한 정보, 서비스 정보 등을 실시간으로 소비자에게 제공하여 사용자에게 자발적 에너지 절감을 유도하는 방식

# Energy Management Agent

**홈 에너지관리에이전트(Home Energy Management Agnet)**

: 홈 소비자 영역에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치 (specified in ISO/IEC 15067-3)

**커뮤니티 에너지관리에이전트(Community Energy Management Agnet)**

: 커뮤니티 영역에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치

**온라인 에너지관리에이전트(Online Energy Management Agnet)**

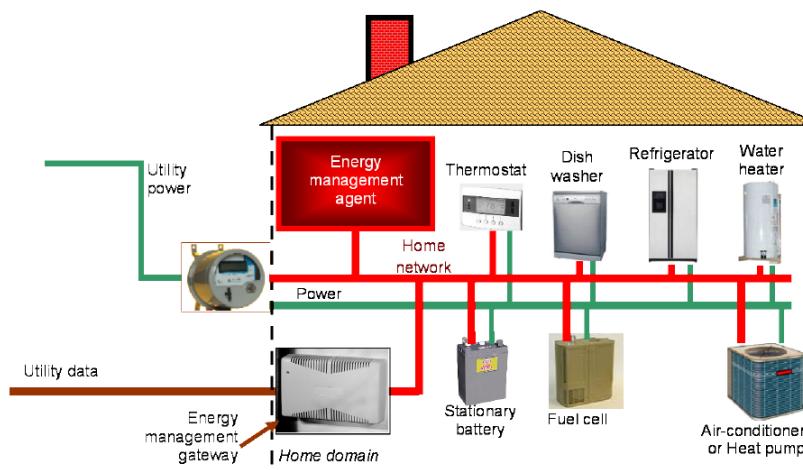
: 클라우드 환경에 위치하여 에너지관리 서비스를 제공하는 제어 기능을 가진 장치

**커뮤니티 수요반응(Community Demand Response)**

: 소규모 수요반응 자원을 하나의 수요반응 자원으로 통합하여 보다 커다란 규모의 수요반응 자원 제어하는 기술

**커뮤니티 에너지관리시스템(Community Energy Management System)**

: 다수의 개별 홈 소비자로부터 에너지 정보를 수집, 통합하여 커뮤니티 또는 온라인 에너지관리시스템을 통한 에너지 관리 시스템

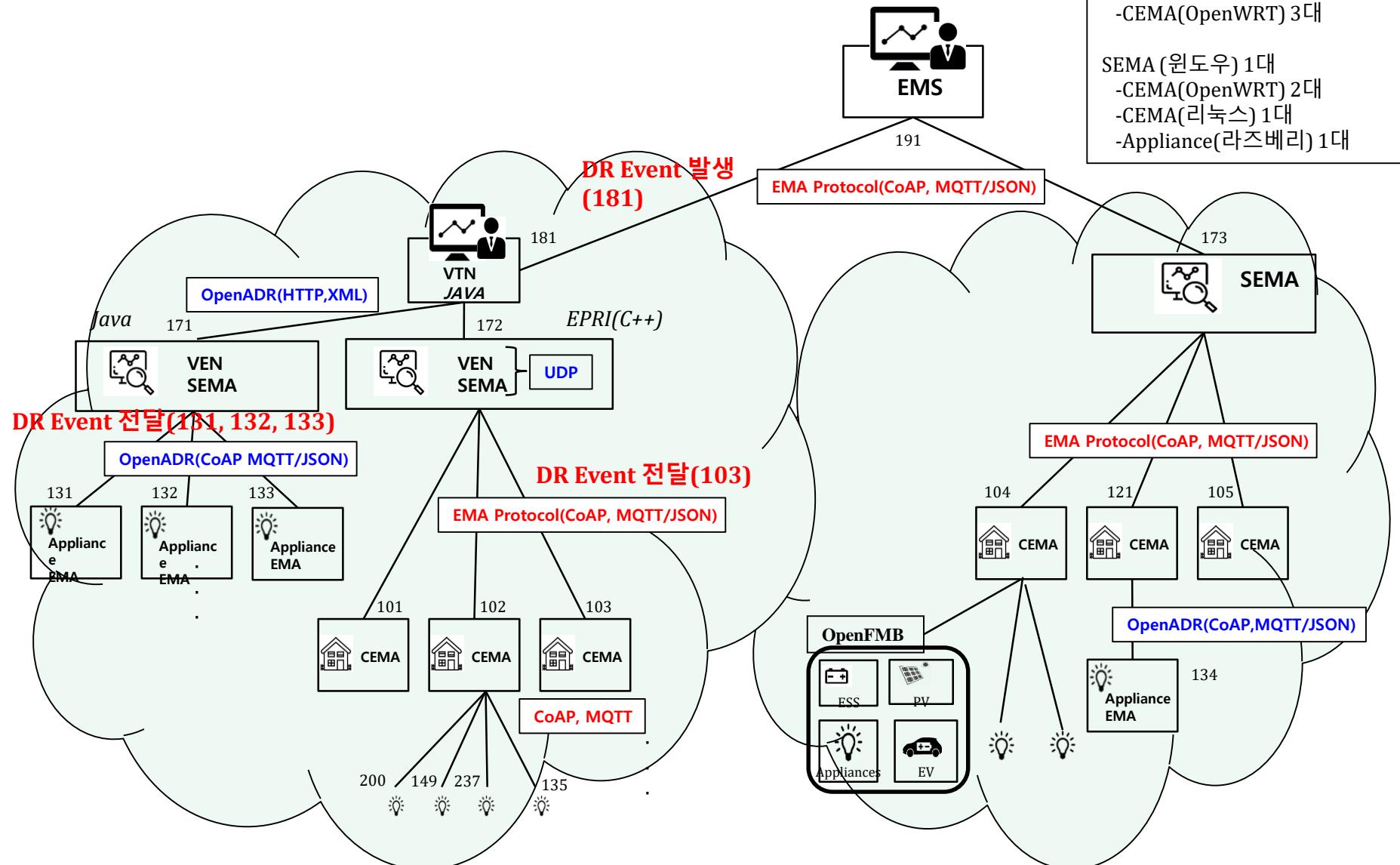


# Main Part

계층 구조 실험

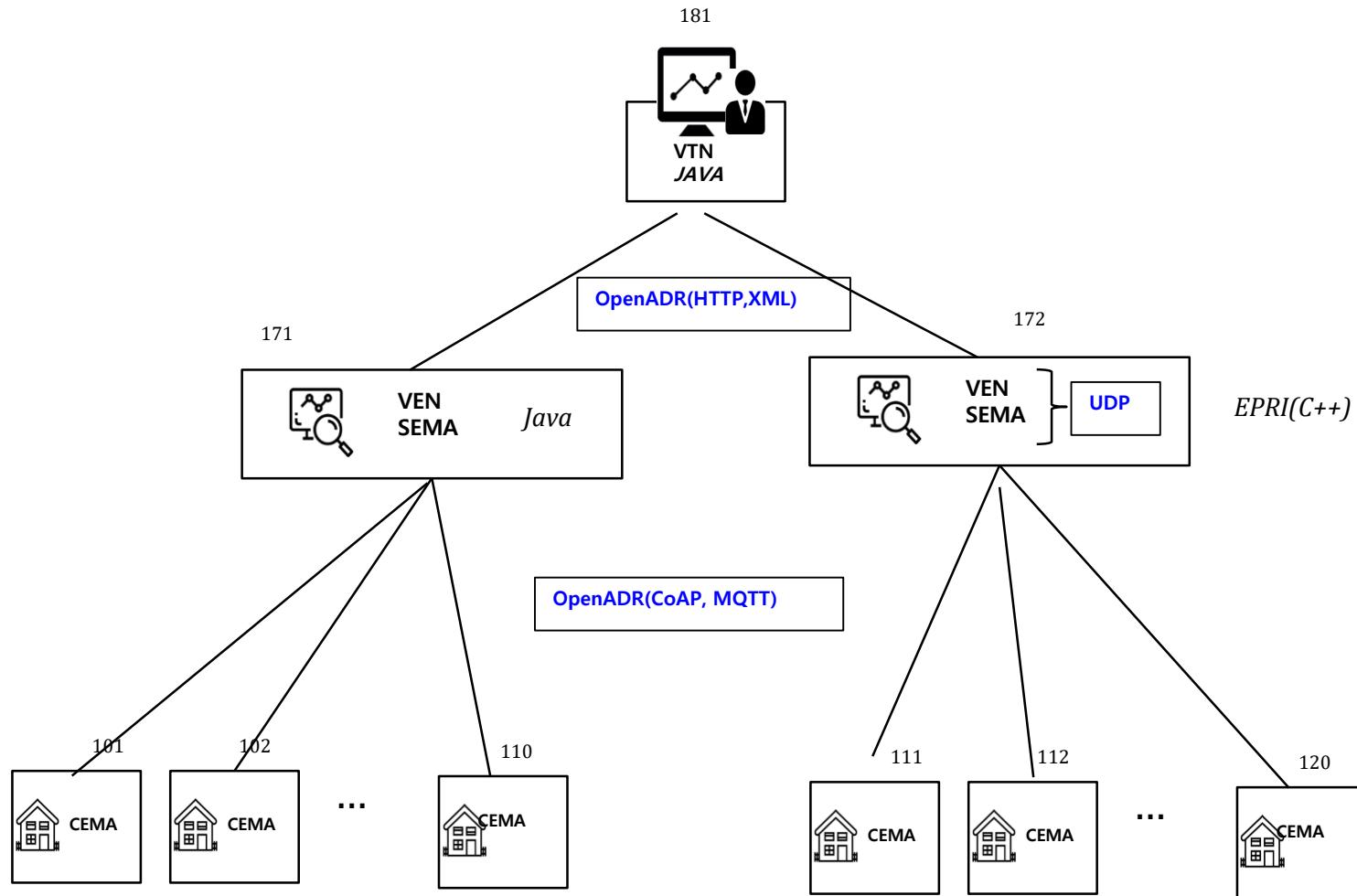
# 계층 구조

## Smart Energy Framework



# 계층 구조 실험

VTN : 1대 (리눅스)  
SEMA + VEN: 2대(리눅스)  
CEMA : SEMA 1대당 10대(OpenWRT)  
Device : SEMA 당 가상 device 5대



# AS-IS

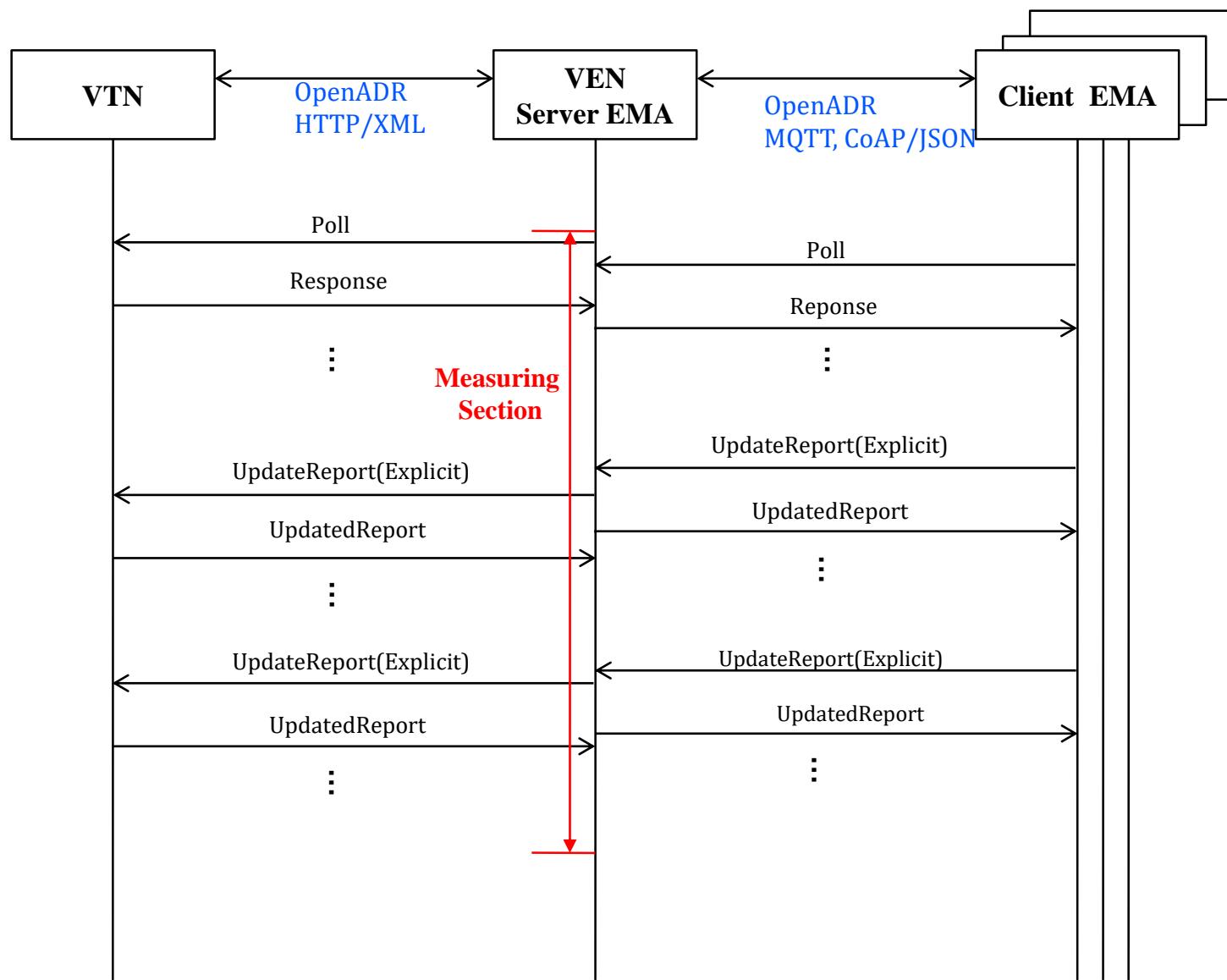
## 실험 측정 방법

---

1. OpenADR2.0b HTTP/XML+ OpenADR 2.0b MQTT/CoAP JSON 데이터 트래픽
2. VTN – [VEN/SEMA 2대] – CEMA [20대] 데이터 트래픽 측정
  - PULL, PUSH 데이터 트래픽 비교
  - 총 CEMA 개수는 20대
3. VTN – SEMA - CEMA 이벤트 응답시간 측정
  - VTN에서 SEMA 까지 Polling 주기 1,3,5sec , Push
  - SEMA에서 CEMA까지 Polling 주기 1,3,5sec, Push Event Response

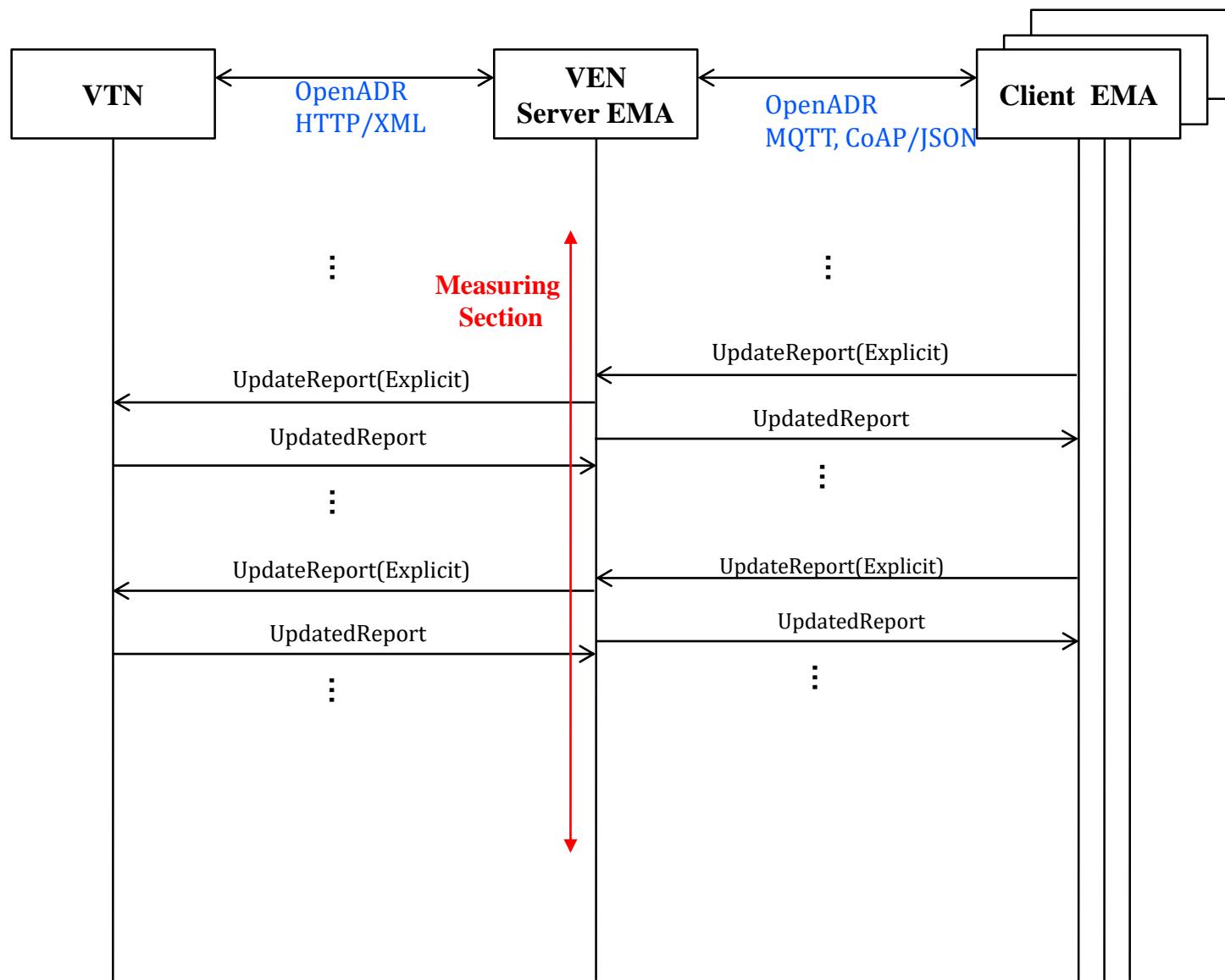
# Message Flow(1) \_

## Data Traffic \_ MQTT,CoAP \_ PULL

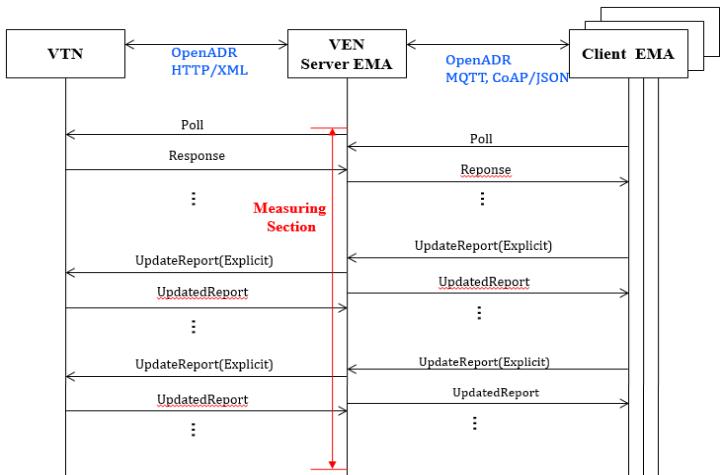
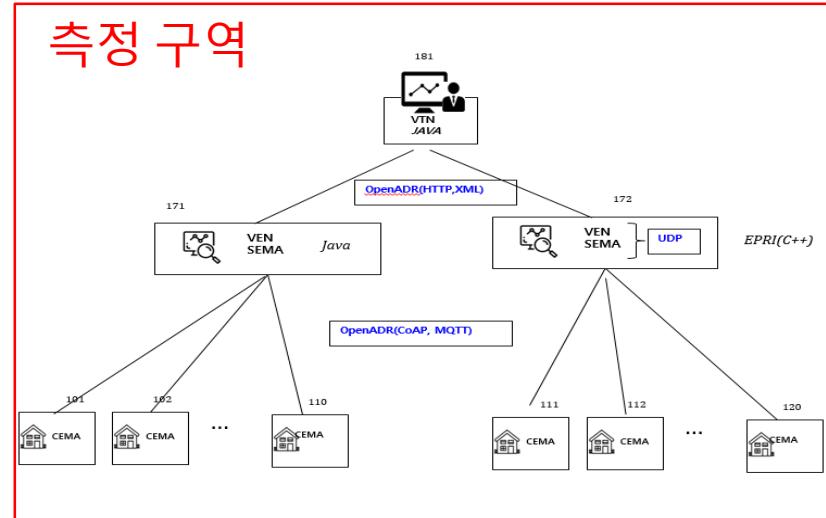


# Message Flow(2)

## Data Traffic - MQTT, CoAP, PUSH



# Experiment Procedure

항목 번호		시험 일자	<th>시 험 자</th> <td></td>	시 험 자	
대 항 목		증 항 목	<th>소 항 목</th> <td></td>	소 항 목	
목 적	실험 측정 구간 SEMA, PULL데이터 트래픽(OpenADR-HTTP/XML, MQTT/JSON, CoAP/JSON)				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. VTN 1대, VEN+ServerEMA 2대를 실행시킨다.</li> <li>3. Wireshark 실행한다.</li> <li>4. VEN을 통해 VTN과 HTTP/XML에 대한 구간을 측정한다.</li> <li>5. 주기적인 Poll/Reponse 메시지, UpdateReport의 트래픽 양을 SEMA에서 측정한다.</li> <li>6. SEMA에서 Wireshark로 60초 동안 측정한다. (Wireshark 종료 옵션)</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준	<p>시험 구성(시험 구성도 및 관련 명령어 작성)</p>  <p><b>측정 구역</b></p> 				
판정	비고				

# Experiment Procedure

항목 번호		시험 일자	<th>시 험 자</th> <td></td>	시 험 자	
대 항 목		증 항 목	<th>소 항 목</th> <td></td>	소 항 목	
목 적	실험 측정 구간 SEMA, PUSH데이터 트래픽(OpenADR-HTTP/XML, MQTT/JSON, CoAP/JSON)				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. VTN 1대, VEN+ServerEMA 2대를 실행시킨다.</li> <li>3. Wireshark 실행한다.</li> <li>4. VEN을 통해 VTN과 HTTP/XML에 대한 구간을 측정한다.</li> <li>5. 주기적인 UpdateReport의 트래픽양을 SEMA에서 측정한다.</li> <li>6. SEMA에서 Wireshark로 60초 동안 측정한다. (Wireshark 종료 옵션)</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준	<pre> sequenceDiagram     participant VTN     participant VENServerEMA     participant ClientEMA     VTN-&gt;&gt;VENServerEMA: OpenADR HTTP/XML     VENServerEMA-&gt;&gt;ClientEMA: OpenADR MQTT, CoAP/JSON     activate ClientEMA     Note over ClientEMA: Measuring Section     ClientEMA-&gt;&gt;VENServerEMA: UpdateReport(Explicit)     VENServerEMA-&gt;&gt;ClientEMA: UpdatedReport     deactivate ClientEMA     </pre>				
시험 구성(시험 구성도 및 관련 명령어 작성)	<p style="color: red; font-size: 1.5em;">측정 구역</p>				
판정	비고				

# Experiment Procedure \_CoAP\_PULL

No.	Time	Source	Destination	Protocol	Length	Message	Message	Info
2	0.042525927	192.168.1.172	192.168.1.181	HTTP/XML	826			POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1
4	0.046018999	192.168.1.181	192.168.1.172	HTTP/XML	1396			HTTP/1.1 200
8	0.146835104	192.168.1.102	192.168.1.171	CoAP	597			CON, MID:40334, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
9	0.158932316	192.168.1.102	192.168.1.171	CoAP	134			CON, MID:40335, PUT, /OpenADR/SERVER_EMA1/2.0b/OadrPoll (application/json)
11	0.353895061	192.168.1.181	192.168.1.171	HTTP/XML	1396			HTTP/1.1 200
13	0.420872017	192.168.1.118	192.168.1.172	CoAP	610			CON, MID:22912, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
14	0.423227789	192.168.1.172	192.168.1.118	CoAP	178			ACK, MID:22912, 2.05 Content (application/json)
16	0.431852883	192.168.1.101	192.168.1.171	CoAP	597			CON, MID:57748, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
18	0.431915752	192.168.1.115	192.168.1.172	CoAP	610			CON, MID:63748, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
19	0.431939736	192.168.1.118	192.168.1.172	CoAP	135			CON, MID:22913, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
20	0.432877744	192.168.1.172	192.168.1.118	CoAP	161			ACK, MID:22913, 2.05 Content (application/json)
21	0.433077347	192.168.1.172	192.168.1.115	CoAP	178			ACK, MID:63748, 2.05 Content (application/json)
23	0.436455350	192.168.1.109	192.168.1.171	CoAP	597			CON, MID:24613, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
25	0.437406158	192.168.1.113	192.168.1.172	CoAP	610			CON, MID:11491, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
26	0.438894791	192.168.1.172	192.168.1.113	CoAP	178			ACK, MID:11491, 2.05 Content (application/json)
27	0.439124485	192.168.1.101	192.168.1.171	CoAP	134			CON, MID:57749, PUT, /OpenADR/SERVER_EMA1/2.0b/OadrPoll (application/json)
28	0.440931089	192.168.1.115	192.168.1.172	CoAP	135			CON, MID:63749, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
29	0.441665969	192.168.1.172	192.168.1.115	CoAP	161			ACK, MID:63749, 2.05 Content (application/json)
31	0.442809202	192.168.1.116	192.168.1.172	CoAP	610			CON, MID:6964, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
32	0.443501911	192.168.1.172	192.168.1.116	CoAP	178			ACK, MID:6964, 2.05 Content (application/json)
34	0.446213120	192.168.1.105	192.168.1.171	CoAP	597			CON, MID:15336, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
35	0.446982217	192.168.1.109	192.168.1.171	CoAP	134			CON, MID:24614, PUT, /OpenADR/SERVER_EMA1/2.0b/OadrPoll (application/json)
37	0.448574647	192.168.1.113	192.168.1.172	CoAP	135			CON, MID:11492, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
38	0.448594019	192.168.1.104	192.168.1.171	CoAP	597			CON, MID:46584, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
39	0.449151778	192.168.1.172	192.168.1.113	CoAP	161			ACK, MID:11492, 2.05 Content (application/json)
40	0.460885156	192.168.1.116	192.168.1.172	CoAP	135			CON, MID:6965, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
41	0.460920702	192.168.1.105	192.168.1.171	CoAP	134			CON, MID:15337, PUT, /OpenADR/SERVER_EMA1/2.0b/OadrPoll (application/json)
42	0.460925679	192.168.1.104	192.168.1.171	CoAP	134			CON, MID:46585, PUT, /OpenADR/SERVER_EMA1/2.0b/OadrPoll (application/json)
43	0.461612445	192.168.1.172	192.168.1.116	CoAP	161			ACK, MID:6965, 2.05 Content (application/json)
45	0.461768768	192.168.1.106	192.168.1.171	CoAP	597			CON, MID:20194, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
47	0.463098100	192.168.1.107	192.168.1.171	CoAP	597			CON, MID:10853, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
49	0.469137477	192.168.1.119	192.168.1.172	CoAP	610			CON, MID:54069, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
50	0.470117490	192.168.1.172	192.168.1.119	CoAP	178			ACK, MID:54069, 2.05 Content (application/json)

# Experiment Procedure \_MQTT\_PULL

No.	Time	Source	Destination	Protocol	Length	Message	Message	Info
3	0.003309026	192.168.1.181	192.168.1.171	HTTP/XML	1538			HTTP/1.1 200
5	0.004328401	192.168.1.172	166.104.28.51	MQTT	223	{"venID": "CLIENT... { "venID": "CLIENT... Publish Message [/OpenADR/CLIENT_EMA16/2.0b/EiReport]		
7	0.010177174	166.104.28.51	192.168.1.172	MQTT	144	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA2/2.0b/OadrPo11]		
8	0.011310567	192.168.1.172	166.104.28.51	MQTT	206	{"venID": "CLIENT... { "venID": "CLIENT... Publish Message [/OpenADR/CLIENT_EMA16/2.0b/OadrPo11]		
9	0.043276437	166.104.28.51	192.168.1.171	MQTT	2156	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
10	0.043303796	166.104.28.51	192.168.1.172	MQTT	2156	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
12	0.052978165	166.104.28.51	192.168.1.171	MQTT	143	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPo11]		
13	0.053649287	166.104.28.51	192.168.1.172	MQTT	143	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPo11]		
19	0.178820168	192.168.1.172	192.168.1.181	HTTP/XML	1854			POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
21	0.184599584	192.168.1.181	192.168.1.172	HTTP/XML	1538			HTTP/1.1 200
23	0.214332910	192.168.1.172	192.168.1.181	HTTP/XML	826			POST /OpenADR2/Simple/2.0b/OadrPo11 HTTP/1.1
24	0.216153486	192.168.1.181	192.168.1.172	HTTP/XML	1396			HTTP/1.1 200
27	0.585887234	166.104.28.51	192.168.1.172	MQTT	2151	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiReport]		
29	0.587207240	192.168.1.172	166.104.28.51	MQTT	223	{"venID": "CLIENT... { "venID": "CLIENT... Publish Message [/OpenADR/CLIENT_EMA12/2.0b/EiReport]		
30	0.596471031	166.104.28.51	192.168.1.172	MQTT	144	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA2/2.0b/OadrPo11]		
31	0.597447713	192.168.1.172	166.104.28.51	MQTT	206	{"venID": "CLIENT... { "venID": "CLIENT... Publish Message [/OpenADR/CLIENT_EMA12/2.0b/OadrPo11]		
33	0.642178409	192.168.1.181	192.168.1.171	HTTP/XML	1396			HTTP/1.1 200
35	0.724756466	166.104.28.51	192.168.1.171	MQTT	696	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
36	0.724763271	166.104.28.51	192.168.1.172	MQTT	2156	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
38	0.734241398	166.104.28.51	192.168.1.171	MQTT	143	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPo11]		
39	0.735326667	166.104.28.51	192.168.1.172	MQTT	143	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPo11]		
40	0.761060366	166.104.28.51	192.168.1.171	MQTT	2202	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
41	0.761076017	166.104.28.51	192.168.1.172	MQTT	2202	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
43	0.770304993	166.104.28.51	192.168.1.171	MQTT	144	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPo11]		
44	0.771274383	166.104.28.51	192.168.1.172	MQTT	144	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPo11]		
47	0.772109197	166.104.28.51	192.168.1.172	MQTT	710	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiReport]		
49	0.773092042	166.104.28.51	192.168.1.171	MQTT	2156	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
50	0.773097385	166.104.28.51	192.168.1.172	MQTT	2156	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
52	0.773368984	192.168.1.172	166.104.28.51	MQTT	223	{"venID": "CLIENT... { "venID": "CLIENT... Publish Message [/OpenADR/CLIENT_EMA15/2.0b/EiReport]		

# Experiment Procedure \_COAP\_PUSH

Coap    http								
No.	Time	Source	Destination	Protocol	Length	Message	Message	Info
4	0.000448820	192.168.1.172	192.168.1.181	HTTP/XML	1854			POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
6	0.005421731	192.168.1.181	192.168.1.172	HTTP/XML	1538			HTTP/1.1 200
10	0.173380215	192.168.1.108	192.168.1.171	CoAP	667			CON, MID:55315, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
12	0.174179067	192.168.1.112	192.168.1.172	CoAP	680			CON, MID:35322, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
14	0.174194898	192.168.1.101	192.168.1.171	CoAP	666			CON, MID:14956, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
15	0.176748421	192.168.1.172	192.168.1.112	CoAP	178			ACK, MID:35322, 2.05 Content (application/json)
17	0.176956133	192.168.1.103	192.168.1.171	CoAP	650			CON, MID:47591, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
19	0.176959474	192.168.1.106	192.168.1.171	CoAP	664			CON, MID:29768, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
21	0.184658649	192.168.1.117	192.168.1.172	CoAP	678			CON, MID:28098, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
23	0.184673932	192.168.1.109	192.168.1.171	CoAP	681			CON, MID:52804, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
24	0.185798537	192.168.1.172	192.168.1.117	CoAP	178			ACK, MID:28098, 2.05 Content (application/json)
26	0.187530772	192.168.1.118	192.168.1.172	CoAP	679			CON, MID:35674, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
27	0.190535459	192.168.1.172	192.168.1.118	CoAP	178			ACK, MID:35674, 2.05 Content (application/json)
29	0.191499189	192.168.1.107	192.168.1.171	CoAP	699			CON, MID:34413, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
32	0.199756597	192.168.1.119	192.168.1.172	CoAP	662			CON, MID:27377, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
33	0.199765718	192.168.1.115	192.168.1.172	CoAP	679			CON, MID:41389, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
34	0.203229740	192.168.1.172	192.168.1.119	CoAP	178			ACK, MID:27377, 2.05 Content (application/json)
35	0.211177903	192.168.1.172	192.168.1.115	CoAP	178			ACK, MID:41389, 2.05 Content (application/json)
37	0.228490239	192.168.1.104	192.168.1.171	CoAP	665			CON, MID:7680, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
39	0.233891090	192.168.1.105	192.168.1.171	CoAP	633			CON, MID:2614, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
41	0.242894092	192.168.1.111	192.168.1.172	CoAP	647			CON, MID:48398, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
42	0.246048721	192.168.1.172	192.168.1.111	CoAP	178			ACK, MID:48398, 2.05 Content (application/json)
44	0.248016916	192.168.1.116	192.168.1.172	CoAP	630			CON, MID:28933, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
45	0.248817072	192.168.1.172	192.168.1.116	CoAP	178			ACK, MID:28933, 2.05 Content (application/json)
47	0.251797129	192.168.1.114	192.168.1.172	CoAP	712			CON, MID:52175, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
48	0.254388061	192.168.1.172	192.168.1.114	CoAP	178			ACK, MID:52175, 2.05 Content (application/json)
50	0.274418544	192.168.1.110	192.168.1.171	CoAP	645			CON, MID:36449, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
52	0.304707559	192.168.1.102	192.168.1.171	CoAP	648			CON, MID:18486, PUT, /OpenADR/SERVER_EMA1/2.0b/EiReport (application/json)
54	0.309789402	192.168.1.120	192.168.1.172	CoAP	695			CON, MID:21895, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
55	0.310910765	192.168.1.172	192.168.1.120	CoAP	178			ACK, MID:21895, 2.05 Content (application/json)
58	0.358873150	192.168.1.181	192.168.1.171	HTTP/XML	1538			HTTP/1.1 200
60	0.403989135	192.168.1.113	192.168.1.172	CoAP	662			CON, MID:18001, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
61	0.404919802	192.168.1.172	192.168.1.113	CoAP	178			ACK, MID:18001, 2.05 Content (application/json)
67	1.011408095	192.168.1.172	192.168.1.181	HTTP/XML	1854			POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
69	1.013892722	192.168.1.181	192.168.1.172	HTTP/XML	1538			HTTP/1.1 200

# Experiment Procedure \_MQTT\_PUSH

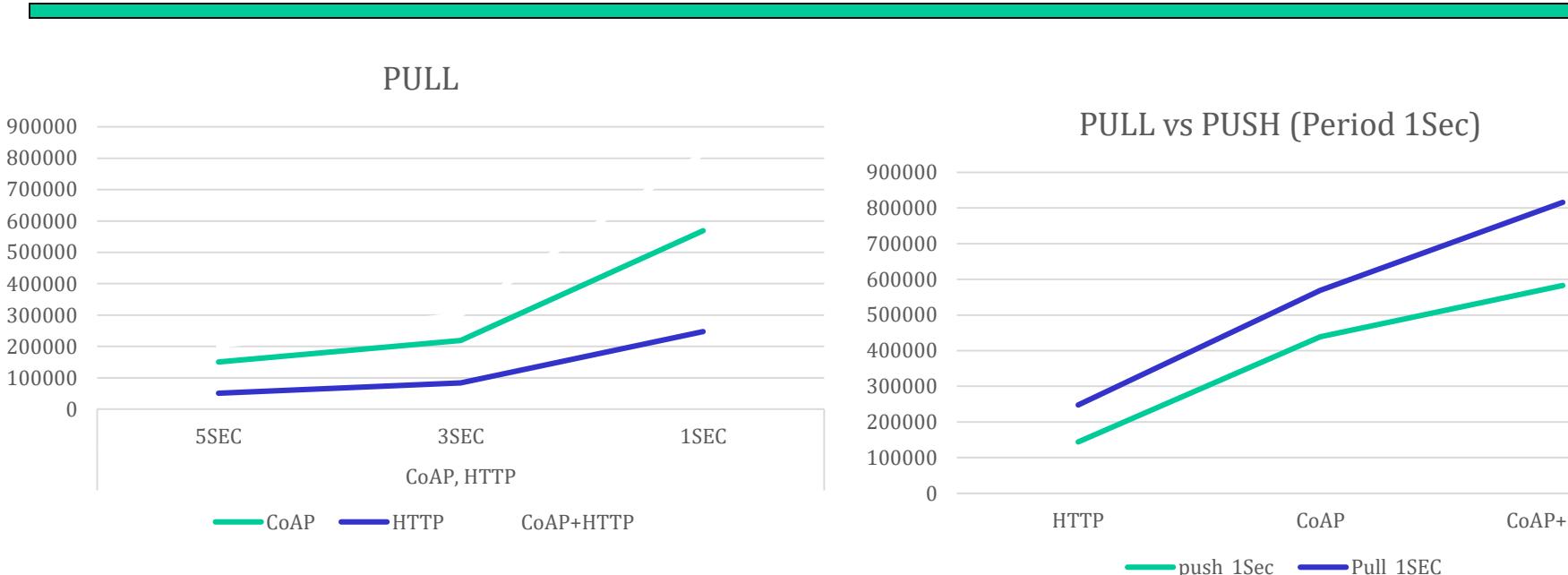
mqtt    http								
No.	Time	Source	Destination	Protocol	Length	Message	Message	Info
1	0.000000000	166.104.28.51	192.168.1.172	MQTT	2171	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiReport]		
3	0.001165486	192.168.1.172	166.104.28.51	MQTT	223	{"venID":"CLIENT... {"venID":"CLIENT... Publish Message [/OpenADR/CLIENT_EMA16/2.0b/EiReport]		
7	0.101410594	192.168.1.181	192.168.1.171	HTTP/XML	1538			HTTP/1.1 200
11	0.220219601	192.168.1.172	192.168.1.181	HTTP/XML	1854			POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
13	0.226212005	192.168.1.181	192.168.1.172	HTTP/XML	1538			HTTP/1.1 200
15	0.268305251	166.104.28.51	192.168.1.172	MQTT	2172	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
16	0.268339588	166.104.28.51	192.168.1.171	MQTT	2172	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
20	0.370149098	166.104.28.51	192.168.1.172	MQTT	695	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiReport]		
22	0.373532421	192.168.1.172	166.104.28.51	MQTT	223	{"venID":"CLIENT... {"venID":"CLIENT... Publish Message [/OpenADR/CLIENT_EMA14/2.0b/EiReport]		
25	0.605830872	166.104.28.51	192.168.1.172	MQTT	2108	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
26	0.605847851	166.104.28.51	192.168.1.171	MQTT	2108	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
28	0.607736364	166.104.28.51	192.168.1.172	MQTT	2140	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
29	0.607752825	166.104.28.51	192.168.1.171	MQTT	2140	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
31	0.624334650	166.104.28.51	192.168.1.172	MQTT	2140	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
32	0.624358338	166.104.28.51	192.168.1.171	MQTT	2140	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
35	0.667603972	166.104.28.51	192.168.1.172	MQTT	697	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
36	0.667613362	166.104.28.51	192.168.1.171	MQTT	2157	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
40	0.687221628	166.104.28.51	192.168.1.172	MQTT	2153	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
42	0.688680188	166.104.28.51	192.168.1.171	MQTT	2153	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
43	0.694540422	166.104.28.51	192.168.1.172	MQTT	2140	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
44	0.694558678	166.104.28.51	192.168.1.171	MQTT	2140	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
46	0.720977628	166.104.28.51	192.168.1.172	MQTT	2124	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
47	0.721679522	166.104.28.51	192.168.1.171	MQTT	2124	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
49	0.731106560	166.104.28.51	192.168.1.172	MQTT	2107	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
51	0.731838328	166.104.28.51	192.168.1.171	MQTT	2107	{"venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		
54	0.781056058	166.104.28.51	192.168.1.172	MQTT	680	{ "venID": "CLIE... { "venID": "CLIE... Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]		

# 실험 결과 \_ MQTT & HTTP \_ PULL, PUSH



MQTT	Push	Polling 1sec	Polling 3sec	Polling 5sec
평균	1609510	2009274.6	469736	412246.4
HTTP	Push	Polling 1sec	Polling 3sec	Polling 5sec
평균	729616	208577.8	68911.8	41155.2
MQTT+HTTP	Push	Polling 1sec	Polling 3sec	Polling 5sec
평균	1755433	2217852.4	538647.8	453401.6

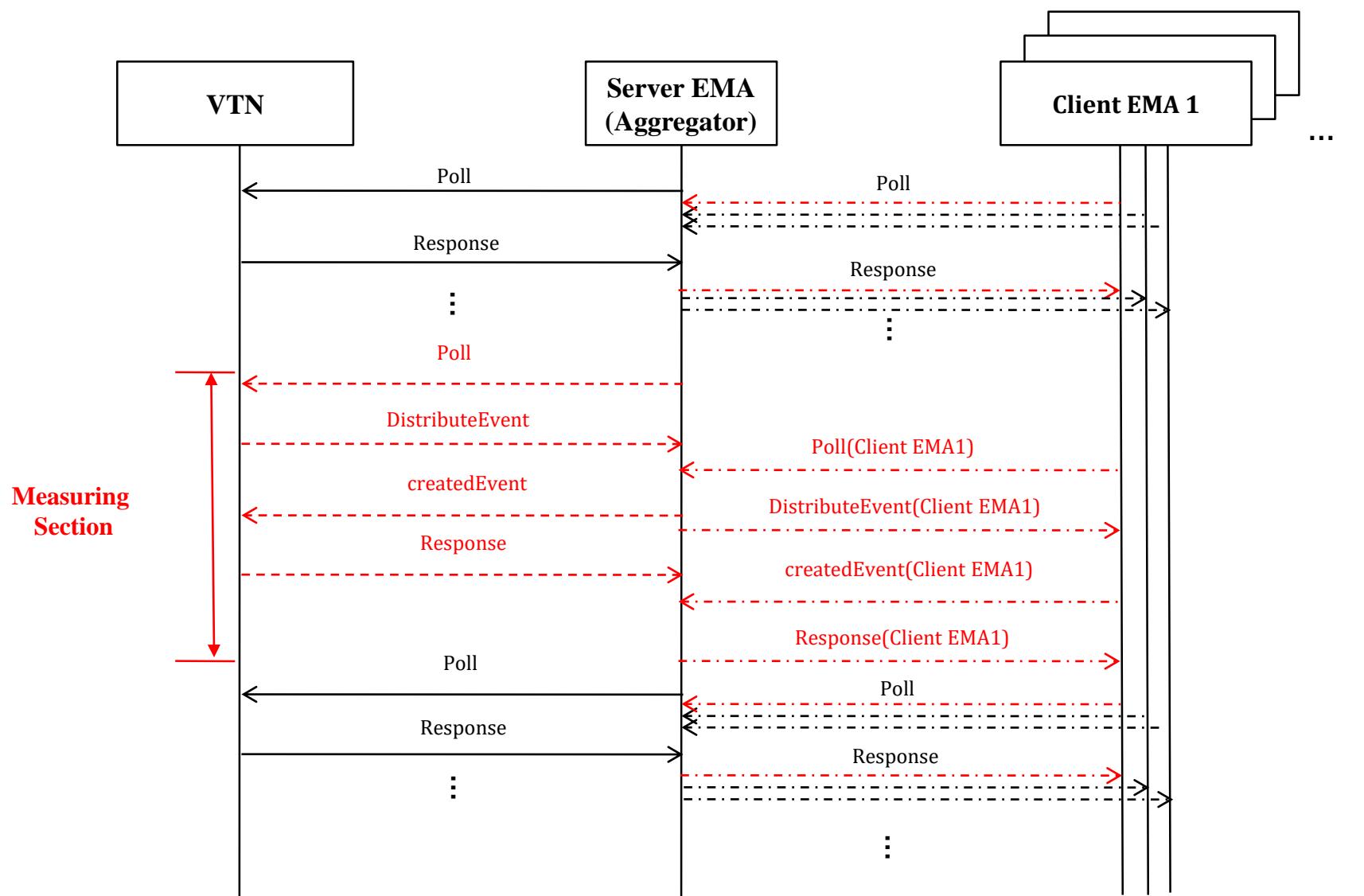
# 실험 결과\_ CoAP & HTTP\_ PULL, PUSH



	Push	Polling 1sec	Polling 3sec	Polling 5sec
CoAP	439214.4	568731.8	218730.2	150650
HTTP	143640.8	247359.6	83673.6	50608.4
CoAP+HTTP	582855.2	816091.4	302403.8	201258.4
평균	439214.4	568731.8	218730.2	150650

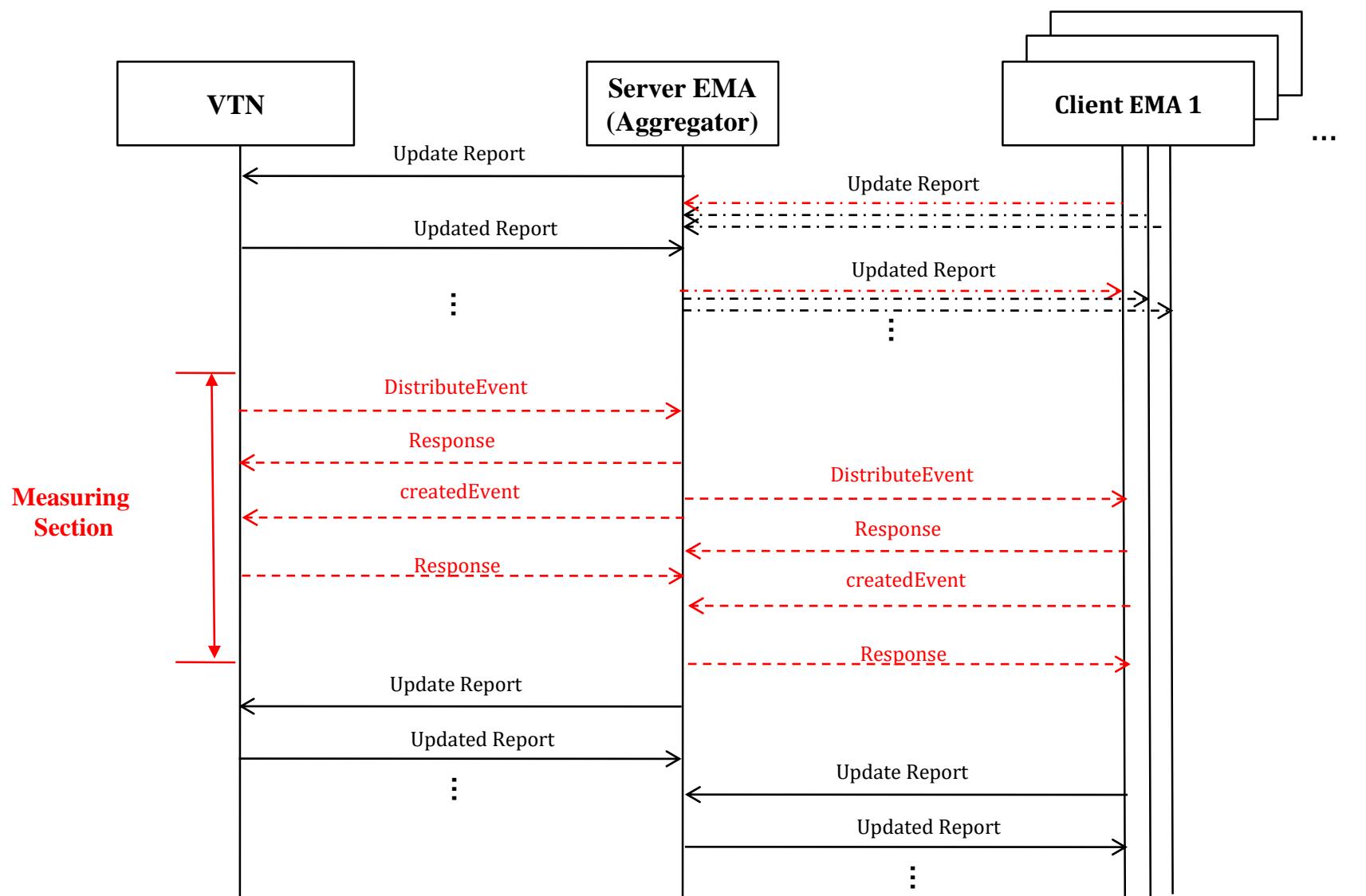
# Message Flow(3) PULL

## Event Response Time - MQTT/CoAP

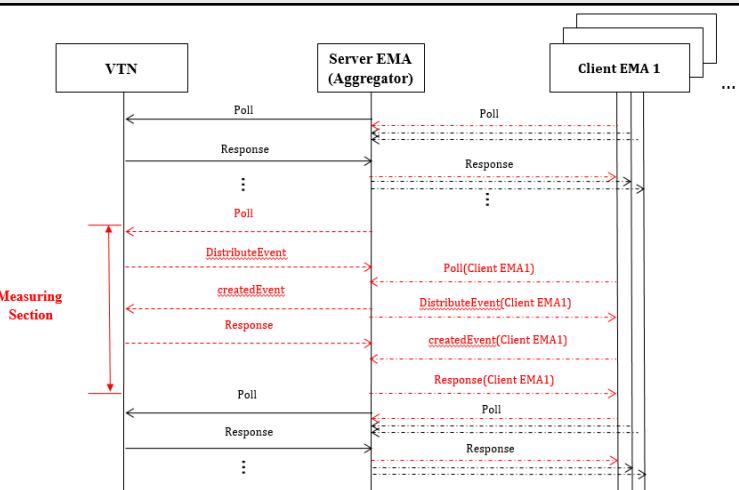
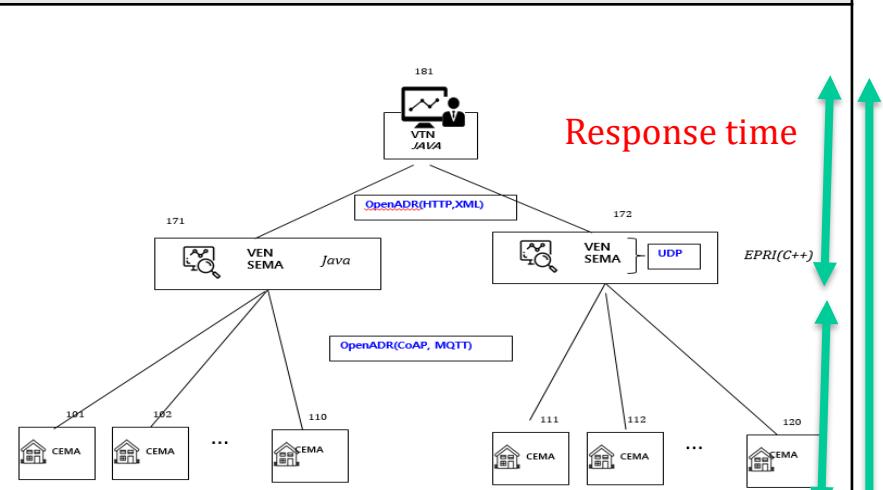


# Message Flow(4) PUSH

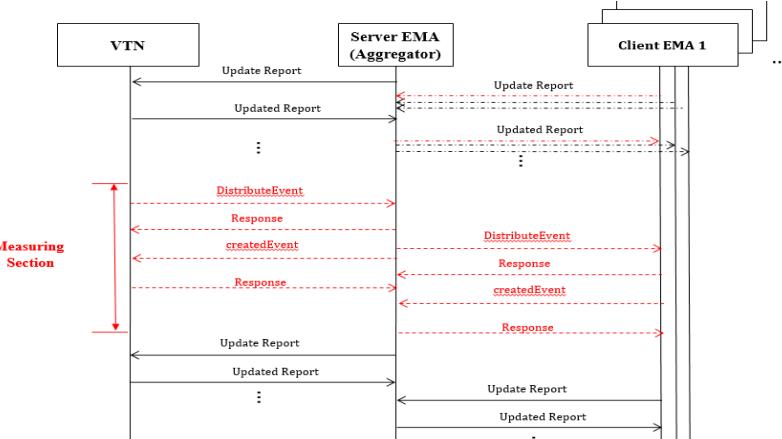
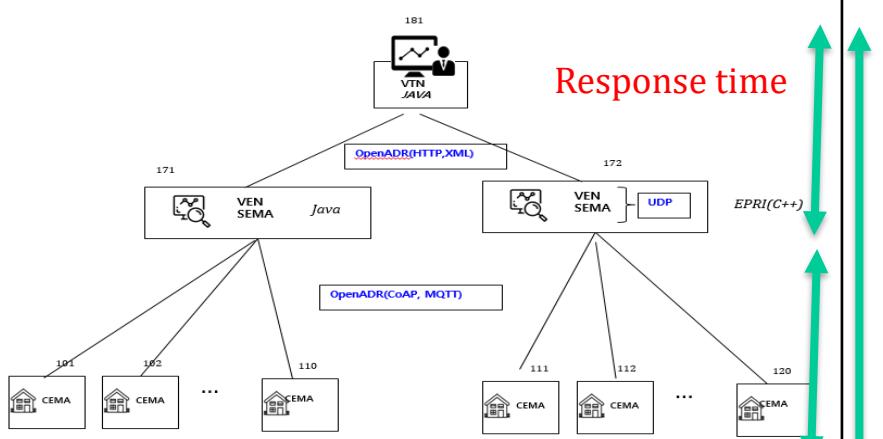
## Event Response Time - MQTT/CoAP



# Experiment Procedure

항목 번호		시험 일자		시 험 자	
대 항 목		증 항 목	<th>소 항 목</th> <td></td>	소 항 목	
목 적	OpenADR2.0b HTTP/XML+ MQTT/CoAP JSON 이벤트 응답 시간 (VTN - VEN+SEMA - CEMA) *PULL				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. VTN 1대, VEN+Server EMA 2대를 실행시킨다. (VTN 및 SEMA의 Report 주기는 1초로 한다)</li> <li>3. Wireshark 실행한다.</li> <li>4. 각 Client EMA 10대를 실행한다.           <ul style="list-style-type: none"> <li>• Client EMA starter를 이용하여 각 EMA들을 가능한 동시에 실행</li> </ul> </li> <li>5. VTN에서 해당 VEN으로 DR을 내리면 VEN이 UDP로 SEMA에게 DR을 전달, SEMA는 CEMA에게 DR을 내린다.           <ul style="list-style-type: none"> <li>• SEMA는 VTN에게 받은 이벤트를 1/N로 분배하여 CEMA에게 DR을 내린다.</li> </ul> </li> <li>6. Wireshark 측정을 종료한다.</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준	시험 구성(시험 구성도 및 관련 명령어 작성)				
 <p>The sequence diagram illustrates the communication flow between VTN, Server EMA (Aggregator), and Client EMA 1. It shows a series of Poll and Response messages. A red double-headed arrow labeled "Measuring Section" spans the duration from the initial Poll from VTN to the final Response from Client EMA 1. Within this section, specific events like "DistributeEvent" and "createdEvent" are highlighted in red, indicating the time points for measurement.</p>	 <p>The system architecture diagram shows the experimental setup. VTN JAVA acts as the central node, connected to VEN SEMA Java and multiple CEMA nodes. The connections are established through different protocols: OpenADR(HTTP.XML), UDP, and OpenADR(CoAP, MQTT). A red double-headed arrow labeled "Response time" spans the distance between VTN JAVA and the CEMA nodes, indicating the total response time for the experiment.</p>				
판정	비고				

# Experiment Procedure

항목 번호		시험 일자		시 험 자	
대 항 목		증 항 목	<th>소 항 목</th> <td></td>	소 항 목	
목 적	OpenADR2.0b HTTP/XML+ MQTT/CoAP JSON 이벤트 응답 시간 (VTN - VEN+SEMA - CEMA) *PUSH				
시험 절차 (시험 절차 또는 방법 작 성)	<ol style="list-style-type: none"> <li>1. 시험 구성도와 같이 시험 환경을 구성한다.</li> <li>2. VTN 1대, VEN+Server EMA 2대를 실행시킨다. (VTN 및 SEMA의 Report 주기는 1초로 한다)</li> <li>3. Wireshark 실행한다.</li> <li>4. 각 Client EMA 10대를 실행한다.           <ul style="list-style-type: none"> <li>• Client EMA starter를 이용하여 각 EMA들을 가능한 동시에 실행</li> </ul> </li> <li>5. VTN에서 해당 VEN으로 DR을 내리면 VEN이 UDP로 SEMA에게 DR을 전달, SEMA는 CEMA에게 DR을 내린다.           <ul style="list-style-type: none"> <li>• SEMA는 VTN에게 받은 이벤트를 1/N로 분배하여 CEMA에게 DR을 내린다.</li> </ul> </li> <li>6. Wireshark 측정을 종료한다.</li> <li>7. 프로그램을 종료 후 다시 측정한다.</li> </ol>				
판정(측정) 기준			시험 구성(시험 구성도 및 관련 명령어 작성)		
 <p>Sequence diagram illustrating the communication flow between VTN, Server EMA (Aggregator), and Client EMA 1. The process involves multiple 'Update Report' messages and 'DistributeEvent' responses. A red double-headed arrow labeled 'Measuring Section' spans across the sequence to indicate the period of measurement for specific events.</p>			 <p>Hierarchical network diagram showing the experiment setup. VTN Java sends OpenADR(HTTP,XML) requests to VEN SEMA Java. VEN SEMA Java uses UDP to send OpenADR(CoAP, MQTT) requests to CEMA nodes. Red arrows labeled 'Response time' and 'EPRI(C++)' indicate the measurement points for response times and EPRI(C++) processing.</p>		
판정	비고				

# Experiment Procedure \_HTTP, MQTT PULL

*REF*	IP	IP	Protocol	Port	Action	Message Content	
0 0.002677046	192.168.1.172	192.168.1.181	HTTP/XML	826	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1		
0 0.009634203	192.168.1.172	192.168.1.181	HTTP/XML	3948	HTTP/1.1 200		
0 0.011744780	192.168.1.181	192.168.1.172	HTTP/XML	1922	POST /OpenADR2/Simple/2.0b/EiEvent HTTP/1.1		
0 0.264331760	166.104.28.51	192.168.1.172	MQTT	143	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPoll]	
0 0.300142833	166.104.28.51	192.168.1.172	MQTT	2340	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiReport]
0 0.319687957	166.104.28.51	192.168.1.172	MQTT	144	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/OadrPoll]
0 0.320793069	192.168.1.172	166.104.28.51	MQTT	973	{ "vtnID": "SERVER..." }	{ "vtnID": "SERVER..." }	Publish Message [/OpenADR/CLIENT_EMA17/2.0b/EiEvent]
0 0.331260700	166.104.28.51	192.168.1.172	MQTT	310	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent]
0 0.332392291	192.168.1.172	166.104.28.51	MQTT	205	{ "venID": "CLIENT..." }	{ "venID": "CLIENT..." }	Publish Message [/OpenADR/CLIENT_EMA17/2.0b/EiEvent]
0 0.366534178	166.104.28.51	192.168.1.172	MQTT	640	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiReport]
0 0.3676655964	192.168.1.172	166.104.28.51	MQTT	223	{ "venID": "CLIENT..." }	{ "venID": "CLIENT..." }	Publish Message [/OpenADR/CLIENT_EMA17/2.0b/EiReport]
0 0.42330262	166.104.28.51	192.168.1.172	MQTT	143	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPoll]
0 0.2181215767	166.104.28.51	192.168.1.172	MQTT	309	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]
0 0.530326826	166.104.28.51	192.168.1.172	MQTT	143	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPoll]
0 0.539161629	166.104.28.51	192.168.1.172	MQTT	144	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/OadrPoll]
0 0.540581763	192.168.1.172	166.104.28.51	MQTT	973	{ "vtnID": "SERVER..." }	{ "vtnID": "SERVER..." }	Publish Message [/OpenADR/CLIENT_EMA11/2.0b/EiEvent]
0 0.541720911	166.104.28.51	192.168.1.172	MQTT	309	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]
0 0.580901141	166.104.28.51	192.168.1.172	MQTT	1088	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/OadrPoll], Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiReport]
0 0.584578042	192.168.1.172	166.104.28.51	MQTT	973	{ "vtnID": "SERVER..." }	{ "vtnID": "SERVER..." }	Publish Message [/OpenADR/CLIENT_EMA19/2.0b/EiEvent]
0 0.584663174	166.104.28.51	192.168.1.172	MQTT	396	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPoll]
0 0.587797594	166.104.28.51	192.168.1.172	MQTT	144	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/OadrPoll]
0 0.587966294	192.168.1.172	166.104.28.51	MQTT	1122	{ "vtnID": "SERVER..." }	{ "vtnID": "SERVER..." }	Publish Message [/OpenADR/CLIENT_EMA14/2.0b/EiEvent], Publish Message [/OpenADR/CLIENT_EMA11/2.0b/EiEvent]
0 0.592866874	166.104.28.51	192.168.1.172	MQTT	143	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPoll]
0 0.593163878	192.168.1.172	166.104.28.51	MQTT	973	{ "vtnID": "SERVER..." }	{ "vtnID": "SERVER..." }	Publish Message [/OpenADR/CLIENT_EMA16/2.0b/EiEvent]
0 0.594656719	166.104.28.51	192.168.1.172	MQTT	143	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/OadrPoll]
0 0.597280560	166.104.28.51	192.168.1.172	MQTT	3101	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]
0 0.598730899	192.168.1.172	166.104.28.51	MQTT	205	{ "venID": "CLIENT..." }	{ "venID": "CLIENT..." }	Publish Message [/OpenADR/CLIENT_EMA14/2.0b/EiEvent]
0 0.598792584	166.104.28.51	192.168.1.172	MQTT	232	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/OadrPoll], Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent]
0 0.611387668	192.168.1.172	166.104.28.51	MQTT	1516	{ "venID": "CLIENT..." }	{ "venID": "CLIENT..." }	Publish Message [/OpenADR/CLIENT_EMA19/2.0b/EiEvent], Publish Message [/OpenADR/CLIENT_EMA13/2.0b/EiEvent]
0 0.612898456	166.104.28.51	192.168.1.172	MQTT	1158	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]
0 0.612970930	192.168.1.172	166.104.28.51	MQTT	579	{ "vtnID": "SERVER..." }	{ "vtnID": "SERVER..." }	Publish Message [/OpenADR/CLIENT_EMA18/2.0b/EiEvent]
0 0.620640275	166.104.28.51	192.168.1.172	MQTT	310	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent]
0 0.620731731	192.168.1.172	166.104.28.51	MQTT	1122	{ "venID": "CLIENT..." }	{ "venID": "CLIENT..." }	Publish Message [/OpenADR/CLIENT_EMA16/2.0b/EiEvent], Publish Message [/OpenADR/CLIENT_EMA12/2.0b/EiEvent]
0 0.622606249	166.104.28.51	192.168.1.172	MQTT	310	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent]
0 0.631205748	192.168.1.172	166.104.28.51	MQTT	205	{ "venID": "CLIENT..." }	{ "venID": "CLIENT..." }	Publish Message [/OpenADR/CLIENT_EMA13/2.0b/EiEvent]
0 0.632270349	166.104.28.51	192.168.1.172	MQTT	310	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent]
0 0.637032572	166.104.28.51	192.168.1.172	MQTT	2100	{ "venID": "CLIE..." }	{ "venID": "CLIE..." }	Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiReport]
0 0.639938849	192.168.1.172	166.104.28.51	MQTT	205	{ "venID": "CLIENT..." }	{ "venID": "CLIENT..." }	Publish Message [/OpenADR/CLIENT_EMA18/2.0b/EiEvent]

# Experiment Procedure \_HTTP, CoAP PULL

1783 *REF*	192.168.1.172	192.168.1.181	HTTP/XML	826	POST /OpenADR2/Simple/2.0b/OadrPoll HTTP/1.1 HTTP/1.1 200
1784 0.003013216	192.168.1.181	192.168.1.172	HTTP/XML	3946	POST /OpenADR2/Simple/2.0b/EiEvent HTTP/1.1
1786 0.025140033	192.168.1.172	192.168.1.181	HTTP/XML	1922	HTTP/1.1 200
1788 0.027334410	192.168.1.181	192.168.1.172	HTTP/XML	1396	CON, MID:11147, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json) CON, MID:63399, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
1801 0.243000309	192.168.1.113	192.168.1.172	CoAP	135	CON, MID:6620, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json) CON, MID:22574, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
1802 0.243017272	192.168.1.115	192.168.1.172	CoAP	135	ACK, MID:11147, 2.05 Content (application/json)
1804 0.243024383	192.168.1.116	192.168.1.172	CoAP	135	ACK, MID:6620, 2.05 Content (application/json)
1805 0.243027034	192.168.1.118	192.168.1.172	CoAP	135	ACK, MID:63399, 2.05 Content (application/json)
1809 0.245534829	192.168.1.172	192.168.1.113	CoAP	929	ACK, MID:22574, 2.05 Content (application/json)
1810 0.245578080	192.168.1.172	192.168.1.181	HTTP/XML	1852	POST /OpenADR2/Simple/2.0b/EiReport HTTP/1.1
1812 0.247093978	192.168.1.172	192.168.1.116	CoAP	929	ACK, MID:6620, 2.05 Content (application/json)
1813 0.250798221	192.168.1.112	192.168.1.172	CoAP	135	CON, MID:13620, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
1814 0.251357472	192.168.1.172	192.168.1.115	CoAP	929	ACK, MID:63399, 2.05 Content (application/json)
1815 0.251885114	192.168.1.172	192.168.1.118	CoAP	929	ACK, MID:22574, 2.05 Content (application/json)
1817 0.255590113	192.168.1.116	192.168.1.172	CoAP	300	CON, MID:6621, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
1818 0.255628022	192.168.1.181	192.168.1.172	HTTP/XML	1538	HTTP/1.1 200
1820 0.257640232	192.168.1.118	192.168.1.172	CoAP	300	CON, MID:22575, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
1821 0.257685172	192.168.1.119	192.168.1.172	CoAP	135	CON, MID:53724, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
1822 0.259214481	192.168.1.120	192.168.1.172	CoAP	135	CON, MID:25014, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
1823 0.260093351	192.168.1.111	192.168.1.172	CoAP	135	CON, MID:19351, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
1824 0.260117277	192.168.1.113	192.168.1.172	CoAP	300	CON, MID:11148, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
1825 0.261607287	192.168.1.115	192.168.1.172	CoAP	300	CON, MID:63400, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
1826 0.266274637	192.168.1.172	192.168.1.112	CoAP	929	ACK, MID:13620, 2.05 Content (application/json)
1827 0.266918428	192.168.1.117	192.168.1.172	CoAP	135	CON, MID:29180, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
1828 0.268832149	192.168.1.172	192.168.1.116	CoAP	161	ACK, MID:6621, 2.05 Content (application/json)
1829 0.269451103	192.168.1.172	192.168.1.120	CoAP	929	ACK, MID:25014, 2.05 Content (application/json)
1830 0.272696761	192.168.1.172	192.168.1.111	CoAP	929	ACK, MID:19351, 2.05 Content (application/json)
1831 0.274816342	192.168.1.172	192.168.1.113	CoAP	161	ACK, MID:11148, 2.05 Content (application/json)
1832 0.275161656	192.168.1.112	192.168.1.172	CoAP	300	CON, MID:13621, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
1833 0.275185800	192.168.1.114	192.168.1.172	CoAP	135	CON, MID:41978, PUT, /OpenADR/SERVER_EMA2/2.0b/OadrPoll (application/json)
1834 0.278887391	192.168.1.172	192.168.1.115	CoAP	161	ACK, MID:63400, 2.05 Content (application/json)
1835 0.284410502	192.168.1.120	192.168.1.172	CoAP	300	CON, MID:25015, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
1836 0.284441166	192.168.1.111	192.168.1.172	CoAP	300	CON, MID:19352, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
1837 0.285278142	192.168.1.172	192.168.1.118	CoAP	161	ACK, MID:22575, 2.05 Content (application/json)
1838 0.285573779	192.168.1.172	192.168.1.112	CoAP	161	ACK, MID:13621, 2.05 Content (application/json)
1839 0.286185214	192.168.1.172	192.168.1.114	CoAP	929	ACK, MID:41978, 2.05 Content (application/json)
1840 0.286257054	192.168.1.172	192.168.1.117	CoAP	929	ACK, MID:29180, 2.05 Content (application/json)
1841 0.289182760	192.168.1.172	192.168.1.119	CoAP	929	ACK, MID:53724, 2.05 Content (application/json)
1842 0.290800930	192.168.1.172	192.168.1.111	CoAP	161	ACK, MID:19352, 2.05 Content (application/json)

# Experiment Procedure \_HTTP, MQTT PUSH

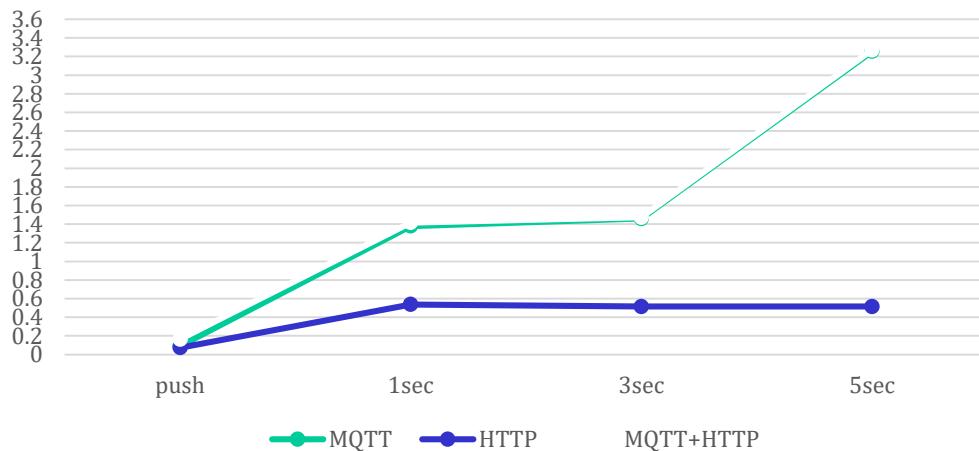
0 *REF*	192.168.1.181	192.168.1.172	HTTP	929	HTTP/1.1 200 OK
5 0.018597334	192.168.1.172	166.104.28.51	MQTT	983 {"vtnID":"SERVER.. {"vtnID":"SERVER.. Publish Message [/OpenADR/CLIENT_EMA16/2.0b/EiEvent]	
9 0.018631676	166.104.28.51	192.168.1.172	MQTT	93 Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]	
1 0.026189192	192.168.1.172	166.104.28.51	MQTT	1516 {"vtnID":"SERVER.. {"vtnID":"SERVER.. Publish Message [/OpenADR/CLIENT_EMA17/2.0b/EiEvent]	
3 0.027365097	192.168.1.172	166.104.28.51	MQTT	1377 {"vtnID":"SERVER.. {"vtnID":"SERVER.. Publish Message [/OpenADR/CLIENT_EMA14/2.0b/EiEvent], Publish Message [/OpenADR/CLIENT_EMA15/2.0b/EiEvent]	
5 0.028651843	192.168.1.172	166.104.28.51	MQTT	1516 {"vtnID":"SERVER.. {"vtnID":"SERVER.. Publish Message [/OpenADR/CLIENT_EMA18/2.0b/EiEvent]	
6 0.030732126	192.168.1.172	166.104.28.51	MQTT	1516 {"vtnID":"SERVER.. {"vtnID":"SERVER.. Publish Message [/OpenADR/CLIENT_EMA19/2.0b/EiEvent], Publish Message [/OpenADR/CLIENT_EMA20/2.0b/EiEvent]	
0 0.031050232	166.104.28.51	192.168.1.172	MQTT	93 Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]	
3 0.031803344	166.104.28.51	192.168.1.172	MQTT	93 Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]	
4 0.031884300	192.168.1.172	166.104.28.51	MQTT	844 {"vtnID":"SERVER.. {"vtnID":"SERVER.. Publish Message [/OpenADR/CLIENT_EMA12/2.0b/EiEvent]	
7 0.035849812	192.168.1.172	166.104.28.51	MQTT	1516 {"vtnID":"SERVER.. {"vtnID":"SERVER.. Publish Message [/OpenADR/CLIENT_EMA13/2.0b/EiEvent]	
9 0.036904267	192.168.1.172	166.104.28.51	MQTT	450 {"vtnID":"SERVER.. {"vtnID":"SERVER.. Publish Message [/OpenADR/CLIENT_EMA11/2.0b/EiEvent]	
1 0.038430458	192.168.1.172	192.168.1.181	HTTP/XML	1922 POST /OpenADR2/Simple/2.0b/EiEvent HTTP/1.1	
2 0.040245286	166.104.28.51	192.168.1.172	MQTT	93 Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent]	
4 0.040277677	166.104.28.51	192.168.1.172	MQTT	130 Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent]	
3 0.043974457	166.104.28.51	192.168.1.172	MQTT	93 Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent]	
0 0.047763805	192.168.1.181	192.168.1.172	HTTP/XML	1396 HTTP/1.1 200	
1 0.079144338	192.168.1.172	192.168.1.181	HTTP	234 GET /EventObserve HTTP/1.1	
4 0.083292987	166.104.28.51	192.168.1.172	MQTT	1551 {"venID": "CLIE.. {"venID": "CLIE.. Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent], Publ	
8 0.085193843	166.104.28.51	192.168.1.172	MQTT	817 {"venID": "CLIE.. {"venID": "CLIE.. Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent], Publ	
9 0.087528680	192.168.1.172	166.104.28.51	MQTT	205 {"venID": "CLIENT.. {"venID": "CLIENT.. Publish Message [/OpenADR/CLIENT_EMA16/2.0b/EiEvent]	
0 0.099559838	166.104.28.51	192.168.1.172	MQTT	309 {"venID": "CLIE.. {"venID": "CLIE.. Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]	
1 0.099583100	166.104.28.51	192.168.1.172	MQTT	1576 {"venID": "CLIE.. {"venID": "CLIE.. Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent], Publi	
4 0.102948270	192.168.1.172	166.104.28.51	MQTT	503 {"venID": "CLIENT.. {"venID": "CLIENT.. Publish Message [/OpenADR/CLIENT_EMA15/2.0b/EiEvent], Publish Message [/OpenADR/CLIENT_EMA17/2.0b/EiEvent], Pub	
7 0.105693345	166.104.28.51	192.168.1.172	MQTT	1072 {"venID": "CLIE.. {"venID": "CLIE.. Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA2/2.0b/EiEvent], Publi	
3 0.140154351	192.168.1.172	166.104.28.51	MQTT	205 {"venID": "CLIENT.. {"venID": "CLIENT.. Publish Message [/OpenADR/CLIENT_EMA19/2.0b/EiEvent]	
2 0.141442143	166.104.28.51	192.168.1.172	MQTT	563 {"venID": "CLIE.. {"venID": "CLIE.. Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent], Publish Message [/OpenADR/SERVER_EMA1/2.0b/EiEvent]	
3 0.141552566	192.168.1.172	166.104.28.51	MQTT	205 {"venID": "CLIENT.. {"venID": "CLIENT.. Publish Message [/OpenADR/CLIENT_EMA13/2.0b/EiEvent]	
5 0.189793838	192.168.1.172	166.104.28.51	MQTT	652 {"venID": "CLIENT.. {"venID": "CLIENT.. Publish Message [/OpenADR/CLIENT_EMA11/2.0b/EiEvent], Publish Message [/OpenADR/CLIENT_EMA12/2.0b/EiEvent], Publ	

# Experiment Procedure \_HTTP, CoAP PUSH

280 *REF*	192.168.1.181	192.168.1.172	HTTP	929	HTTP/1.1 200 OK
290 0.008911077	192.168.1.172	192.168.1.181	HTTP/XML	1922	POST /OpenADR2/Simple/2.0b/EiEvent HTTP/1.1
292 0.020287870	192.168.1.181	192.168.1.172	HTTP/XML	1396	HTTP/1.1 200
301 0.041557404	192.168.1.172	192.168.1.181	HTTP	234	GET /EventObserve HTTP/1.1
309 0.068195791	192.168.1.172	192.168.1.115	CoAP	941	NON, MID:1802, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
310 0.068295814	192.168.1.172	192.168.1.116	CoAP	941	NON, MID:1803, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
311 0.068349867	192.168.1.172	192.168.1.114	CoAP	941	NON, MID:1804, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
312 0.068403316	192.168.1.172	192.168.1.117	CoAP	941	NON, MID:1805, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
314 0.078983904	192.168.1.172	192.168.1.113	CoAP	941	NON, MID:1806, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
316 0.079684095	192.168.1.116	192.168.1.172	CoAP	300	CON, MID:29080, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
317 0.081162095	192.168.1.114	192.168.1.172	CoAP	300	CON, MID:52322, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
320 0.084386557	192.168.1.115	192.168.1.172	CoAP	300	CON, MID:41536, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
322 0.084407286	192.168.1.117	192.168.1.172	CoAP	300	CON, MID:28245, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
323 0.089960871	192.168.1.113	192.168.1.172	CoAP	300	CON, MID:18148, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
325 0.096355613	192.168.1.112	192.168.1.172	CoAP	664	CON, MID:35469, PUT, /OpenADR/SERVER_EMA2/2.0b/EiReport (application/json)
326 0.098091957	192.168.1.172	192.168.1.112	CoAP	941	NON, MID:1807, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
327 0.098189018	192.168.1.172	192.168.1.118	CoAP	941	NON, MID:1808, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
329 0.101177127	192.168.1.172	192.168.1.111	CoAP	941	NON, MID:1809, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
330 0.102156841	192.168.1.172	192.168.1.120	CoAP	941	NON, MID:1810, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
335 0.107710040	192.168.1.118	192.168.1.172	CoAP	300	CON, MID:35821, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
336 0.111462326	192.168.1.111	192.168.1.172	CoAP	300	CON, MID:48545, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
339 0.117440050	192.168.1.172	192.168.1.119	CoAP	941	NON, MID:1811, 2.05 Content, TKN:91 61 16 c9 f0 76 eb 00 (application/json)
340 0.117523482	192.168.1.172	192.168.1.116	CoAP	161	ACK, MID:29080, 2.05 Content (application/json)
341 0.117575893	192.168.1.172	192.168.1.114	CoAP	161	ACK, MID:52322, 2.05 Content (application/json)
342 0.117626713	192.168.1.172	192.168.1.115	CoAP	161	ACK, MID:41536, 2.05 Content (application/json)
343 0.117676986	192.168.1.172	192.168.1.117	CoAP	161	ACK, MID:28245, 2.05 Content (application/json)
344 0.117726909	192.168.1.172	192.168.1.113	CoAP	161	ACK, MID:18148, 2.05 Content (application/json)
345 0.117777310	192.168.1.172	192.168.1.112	CoAP	178	ACK, MID:35469, 2.05 Content (application/json)
348 0.120929865	192.168.1.120	192.168.1.172	CoAP	300	CON, MID:22042, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
351 0.122889313	192.168.1.172	192.168.1.111	CoAP	161	ACK, MID:48545, 2.05 Content (application/json)
352 0.122946553	192.168.1.172	192.168.1.118	CoAP	161	ACK, MID:35821, 2.05 Content (application/json)
353 0.124571786	192.168.1.119	192.168.1.172	CoAP	300	CON, MID:27524, PUT, /OpenADR/SERVER_EMA2/2.0b/EiEvent (application/json)
354 0.126110149	192.168.1.172	192.168.1.120	CoAP	161	ACK, MID:22042, 2.05 Content (application/json)

# 실험 결과 \_ MQTT & HTTP \_ PULL, PUSH

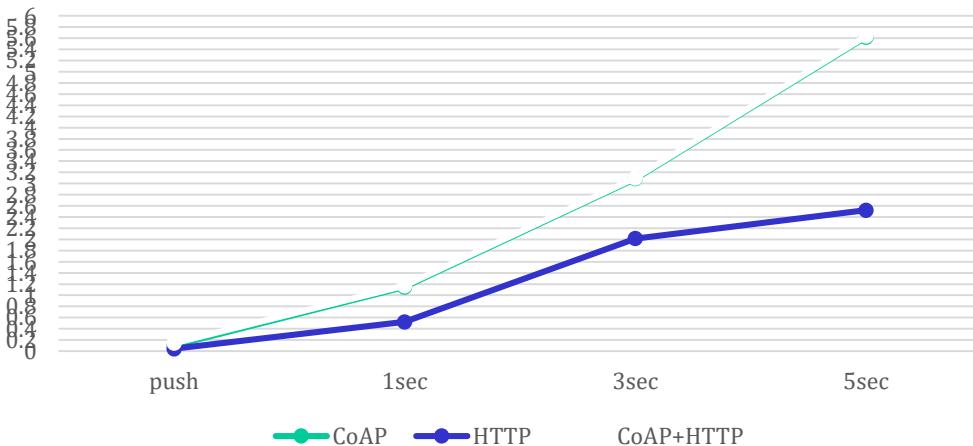
MQTT, HTTP Event Responsetime



	MQTT, HTTP			
	push	1sec	3sec	5sec
MQTT	0.0912	1.3782	1.45934	3.2558
HTTP	0.0738	0.5378	0.51486	0.5146
MQTT+HTTP	0.165	1.416	1.4742	3.2704

# 실험 결과 \_ CoAP & HTTP \_ PULL, PUSH

CoAP, HTTP Evnt Responsetime



MQTT, HTTP				
	push	1sec	3sec	5sec
CoAP	0.10068	1.146064	3.085788	5.619574
HTTP	0.04196	0.523121	2.015668	2.521368
CoAP+HTTP	0.14264	1.169185	3.101456	5.640942

# 문서용 Backup

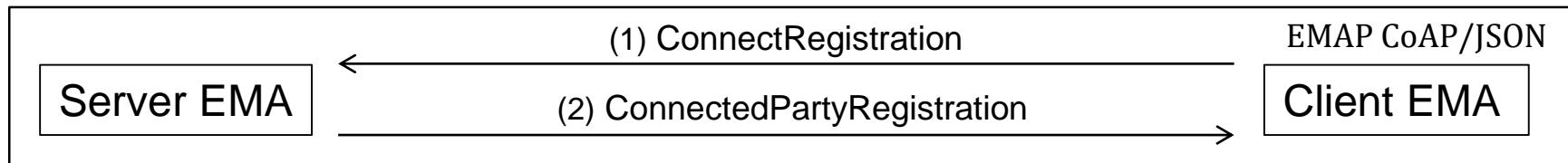
---

- BackUp 자료

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(1) ConnectRegistration

PUT : /EMAP/SEMA/1.0b/SessionSetup

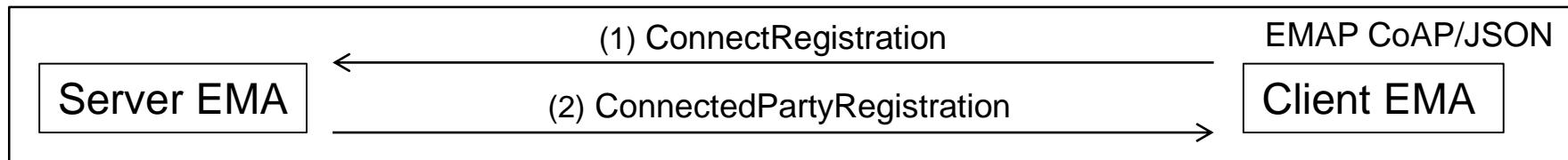
Payload : ConnectRegistration

```
ConnectRegistration Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "service": String,
    "version" : String,
    "time" : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(2) ConnectedRegistration

PUT : /EMAP/CEMA/1.0b/SessionSetup  
Payload : ConnectedRegistration

```
ConnectedRegistration Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "responseCode" : Integer,
    "responseDescription": String,
    "requestID": String,
    "duration" : Integer,
    "profile": Array,
    "service": String,
    "version" : String,
    "time" : Date,
    "registrationID" : String
}
```

```
profile Array{
    "transports": Array,
    "profileName": String
}

transports Array{
    "transportName": String,
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(3) CreatePartyRegistration

PUT : /EMAP/SEMA/1.0b/SessionSetup

Payload : CreatePartyRegistration

```
CreatePartyRegistration Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID" : String,
    "transportName" : String,
    "reportOnly" : Integer,
    "httpPullModel" : Boolean,
    "profileName" : String,
    "xmlSignature" : String,
    "service": String,
    "time" : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(4) CreatedPartyRegistration

PUT : /EMAP/CEMA/1.0b/SessionSetup  
Payload : CreatedPartyRegistration

**CreatedPartyRegistration Object{**  
    “SrcEMA” : String,  
    “DestEMA” : String,  
    “requestID” : String,  
    **“profile”:** Array{  
        “registrationID”: String,  
        “duration” : String,  
        “responseCode” : Integer,  
        “responseDescription”: String,  
        “service”: String,  
        **“time”** : Date  
    }

**profile** Array{  
    “transports”: Array{  
        “profileName”: String  
    }  
}

**transports** Array{  
    “transportName”: String,  
}

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(5) RegisterReport

PUT : /EMAP/SEMA/1.0b/SessionSetup

Payload : RegisterReport

**RegisterReport Object{**  
    “SrcEMA” : String,  
    “DestEMA” : String,  
    “requestID”: String,  
    “reportType” : String,  
    - “**report**” : Array,  
        “time”: Date,  
        “service” : String,  
        “type”: String  
    }  
}

**report Object{**  
    “duration” : String,  
    “reportRequestID” : Integer,  
    “reportSpecifierID” : String,  
    “reportName” : String,  
    “createdDateTime” : Date,  
    “**reportDescription**” : Array,  
    }  
}

**reportDescription Object{**  
    “rID” : String,  
    “resourceID” : String,  
    **“deviceType”** : String,  
    “reportType” : String,  
    “itemUnits” : String,  
    “siScaleCode” : String,  
    “marketContext” : String,  
    “MinPeriod” : String,  
    “MaxPeriod” : String,  
    “OnChange” : String,  
    “itemDescription” : String,  
    “**powerAttributes**” : Array,  
    “qos” : String  
    “state” : String,  
    “power” : Double,  
    “dimming” : Integer,  
    “margin” : double,  
    “generate” : double,  
    “storage” : String,  
    “maxValue” : Double,  
    “minValue” : Double,  
    “avgValue” : Double,  
    “maxTime” : Date,  
    “minTime” : Date,  
    “priority” : Integer  
    }  
}

**powerAttributes Object{**  
    “hertz” : Double,  
    “voltage” : Double,  
    “ac” : Boolean  
    }

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(6) RegisteredReport

PUT : /EMAP/CEMA/1.0b/SessionSetup

Payload : RegisteredReport

```
RegisteredReport Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode" : Integer,
    "responseDescription" : String,
    "service": String,
    "time" : Date,
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(7) Poll

PUT : /EMAP/SEMA/1.0b/SessionSetup  
Payload : Poll

```
Poll JSON Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "service" : String,
    "time" : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(8) RegisterReport

PUT : /EMAP/CEMA/1.0b/SessionSetup

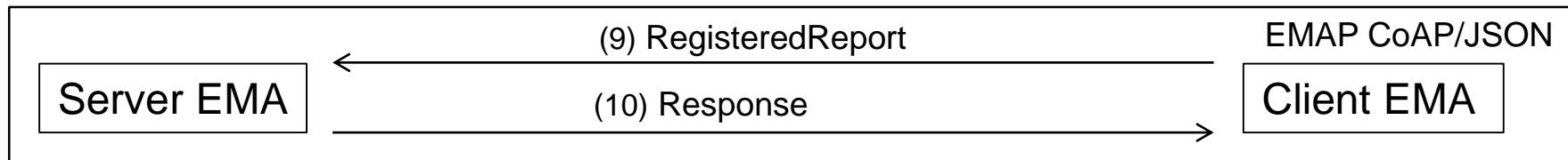
Payload : RegisterReport

```
RegisterReport Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "time": Date,
    "service" : String
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(9) RegisteredReport

PUT : /EMAP/SEMA/1.0b/SessionSetup

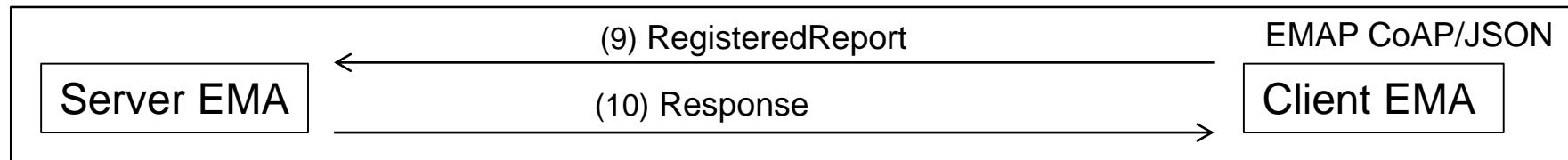
Payload : RegisteredReport

```
RegisteredReport Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “requestID”: String,
    “responseCode” : Integer,
    “responseDescription”: String,
    “service”: String,
    “time” : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(10) Response

PUT : /EMAP/CEMA/1.0b/SessionSetup

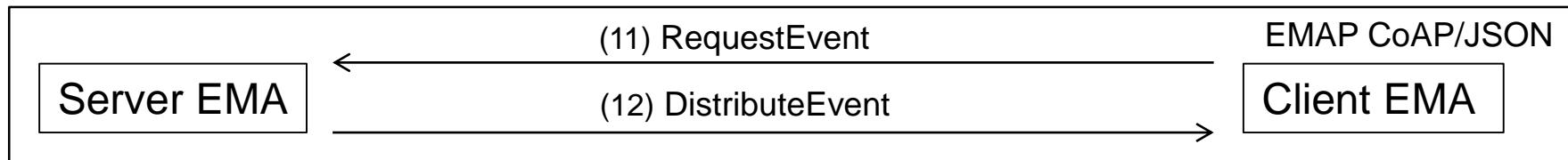
Payload : Response

```
Response Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “requestID”: String,
    “responseCode” : Integer,
    “responseDescription”: String,
    “service”: String,
    “time” : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(11) RequestEvent

PUT : /EMAP/SEMA/1.0b/SessionSetup

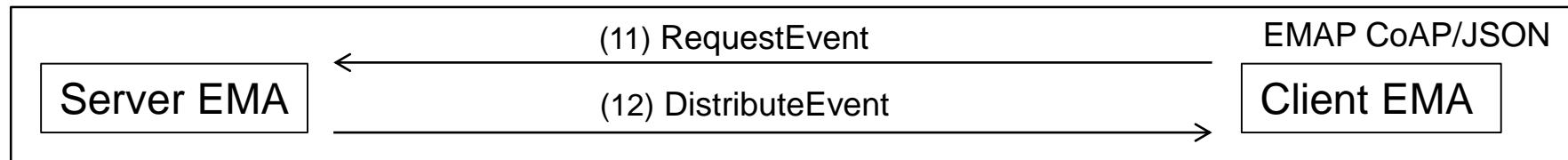
Payload : RequestEvent

```
RequestEvent Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “requestID”: String,
    “time” : Date,
    “service”: String
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(12) DistributeEvent  
PUT : /EMAP/CEMA/1.0b/SessionSetup  
Payload : DistributeEvent

**DistributeEvent Object{**  
    “SrcEMA” : String,  
    “DestEMA” : String,  
    “requestID” : String,  
    **“response” : Array,**  
    **“event” : Array,**  
    “responseRequired” : String,  
    “service” : String,  
    “time” : Date  
}

**response Array{**  
    “requestID” : String,  
    “responseCode” : Integer,  
    “responseDescription” : String  
}

**event Array{**  
    “eventID” : String,  
    **“eventSignals” : Array,**  
    “modificationNumber” : Integer,  
    “modificationReason” : String,  
    “priority” : Integer,  
    “marketContext” : String,  
    “createdDataTime” : Date,  
    “eventStatus” : String,  
    “testEvent” : Boolean,  
    “vtnComment” : String,  
    “dtstart” : Date,  
    “duration” : String,  
    “properties” : String,  
    “components” : String,  
    “specificDestEMA” : String,  
    “tolerance” : String,  
    “notification” : String,  
    “rampUp” : String,  
    “recovery” : String  
}

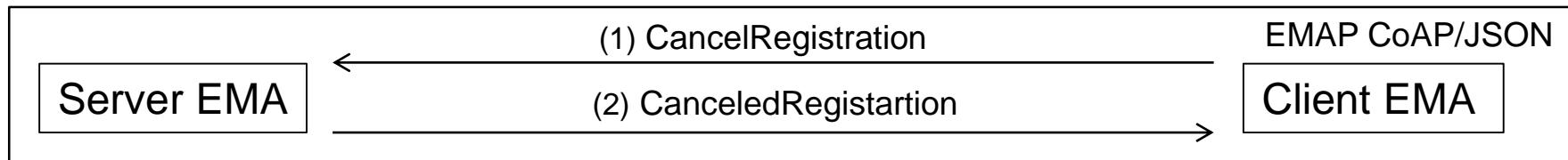
**eventSignals Object{**  
    “eventSignal” : String,  
    **“intervals” : Array,**  
    “signalName” : String,  
    **“signalType” : String,**  
    “signalID” : String,  
    “currentValue” : Double,  
    “threshold” : Double,  
    “capacity” : Double,  
    “price” : Integer,  
    “unit” : String  
}

**intervals Array{**  
    “duration” : String,  
    “uid” : Integer,  
    “value” : Double  
}

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(1) CancelRegistration

PUT : /EMAP/SEMA/1.0b/SessionSetup

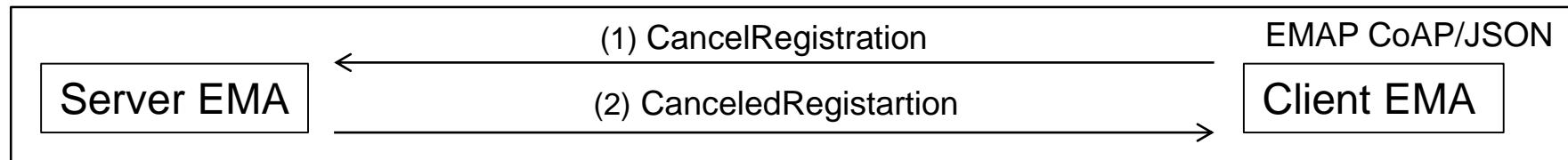
Payload : CancelRegistration

```
CancelRegistration Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “requestID”: String,
    “time” : Date,
    “service”: String
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(2) CanceledRegistration

PUT : /EMAP/CEMA/1.0b/SessionSetup

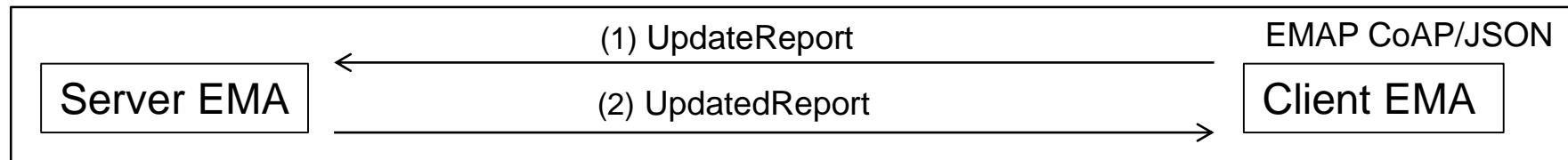
Payload : CanceledRegistration

```
CanceledRegistration Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “requestID”: String,
    “responseCode” : Integer,
    “responseDescription” : String,
    “registrationID” : String
    “time” : Date,
    “service”: String
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

# 2. Smart Home Energy Framework :

## 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(1) UpdateReport

PUT : /EMAP/SEMA/1.0b/UpdateReport

Payload : UpdateReport

```
RegisterReport Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "reportType" : String,
    - "report" : Array,
    "time" : Date,
    "service" : String,
    "type" : String
}
```

```
report Object{
    "duration" : String,
    "reportRequestID" : Integer,
    "reportSpecifierID" : String,
    "reportName" : String,
    "createdDateTime" : Date,
    "reportDescription" : Array,
}
```

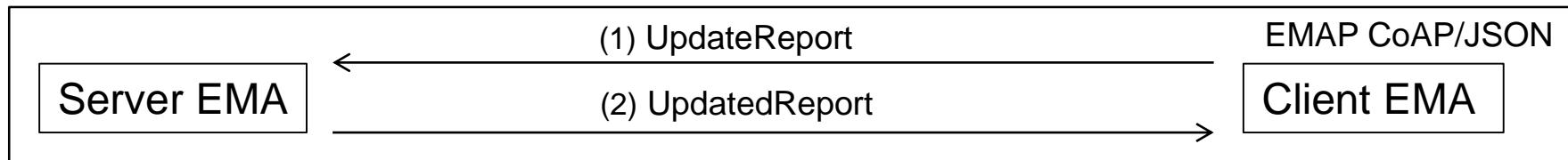
```
reportDescription Object{
    "rID" : String,
    "resourceID" : String,
    "deviceType" : String,
    "reportType" : String,
    "itemUnits" : String,
    "siScaleCode" : String,
    "marketContext" : String,
    "MinPeriod" : String,
    "MaxPeriod" : String,
    "OnChange" : String,
    "itemDescription" : String,
    "powerAttributes" : Array,
    "qos" : String,
    "state" : String,
    "power" : Double,
    "dimming" : Integer,
    "margin" : double,
    "generate" : double,
    "storage" : String,
    "maxValue" : Double,
    "minValue" : Double,
    "avgValue" : Double,
    "maxTime" : Date,
    "minTime" : Date,
    "priority" : Integer
}
```

```
powerAttributes Object{
    "hertz" : Double,
    "voltage" : Double,
    "ac" : Boolean
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport



(2) UpdatedReport

PUT : /EMAP/CEMA/1.0b/UpdateReport

Payload : UpdatedReport

```
RegisteredReport Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “requestID”: String,
    “responseCode” : Integer,
    “responseDescription” : String,
    “service”: String,
    “time” : Date,
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event



(1) Poll

PUT : /EMAP/SEMA/1.0b/Poll

Payload : Poll

```
Poll JSON Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "service" : String,
    "time" : Date
}
```

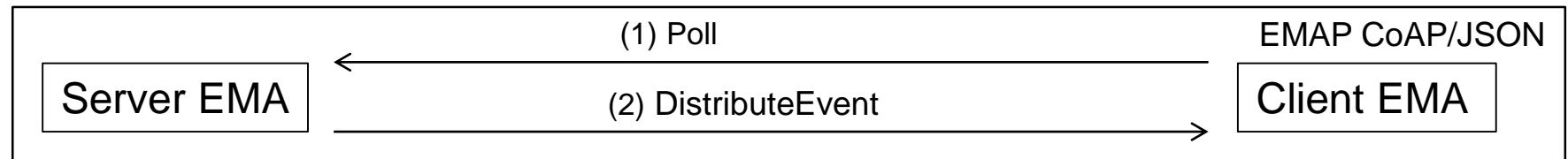
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event



(2) DistributeEvent

PUT : /EMAP/CEMA/1.0b/Event

Payload : DistributeEvent

DistributeEvent Object{

    “SrcEMA” : String,  
    “DestEMA” : String,  
    “requestID” : String,  
    **“response” : Array,**  
    **“event” : Array,**  
    “responseRequired” : String,  
    “service” : String,  
    “time” : Date  
}

event Array{

    “eventID” : String,  
    **“eventSignals” : Array,**  
    “modificationNumber” : Integer,  
    “modificationReason” : String,  
    “priority” : Integer,  
    “marketContext” : String,  
    “createdDataTime” : Date,  
    “eventStatus” : String,  
    “testEvent” : Boolean,  
    “vtnComment” : String,  
    “dtstart” : Date,  
    “duration” : String,  
    “properties” : String,  
    “components” : String,  
    “specificDestEMA” : String,  
    “tolerance” : String,  
    “notification” : String,  
    “rampUp” : String,  
    “recovery” : String  
}

eventSignals Object{

    “eventSignal” : String,  
    **“intervals” : Array,**  
    “signalName” : String,  
    **“signalType” : String,**  
    “signalID” : String,  
    “currentValue” : Double,  
    “threshold” : Double,  
    “capacity” : Double,  
    “price” : Integer,  
    “unit” : String  
}

response Array{

    “requestID” : String,  
    “responseCode” : Integer,  
    “responseDescription” : String  
}

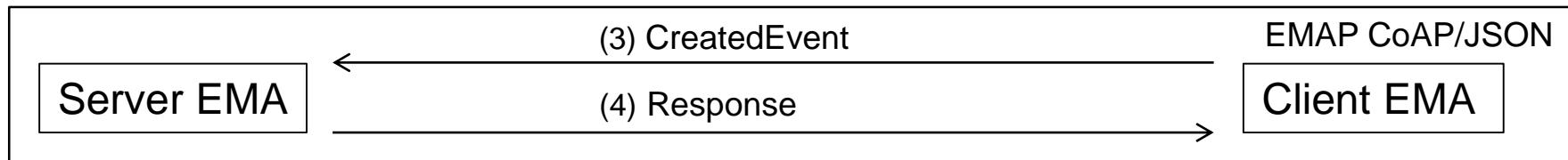
intervals Array{

    “duration” : String,  
    “uid” : Integer,  
    “value” : Double  
}

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event



(3) CreatedEvent

PUT : /EMAP/SEMA/1.0b/Event

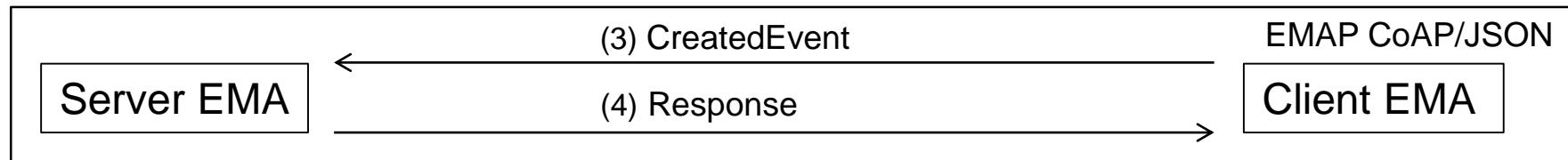
Payload : CreatedEvent

```
CreatedEvent Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode": Integer,
    "reponseDescription": String,
    "optType": String,
    "eventID" : String,
    "modificationNumber" : Integer,
    "service" : String,
    "time" : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event



(4) Response

PUT : /EMAP/CEMA/1.0b/Event

Payload : Response

```
Response Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode" : Integer,
    "responseDescription": String,
    "service": String,
    "time" : Date
}
```

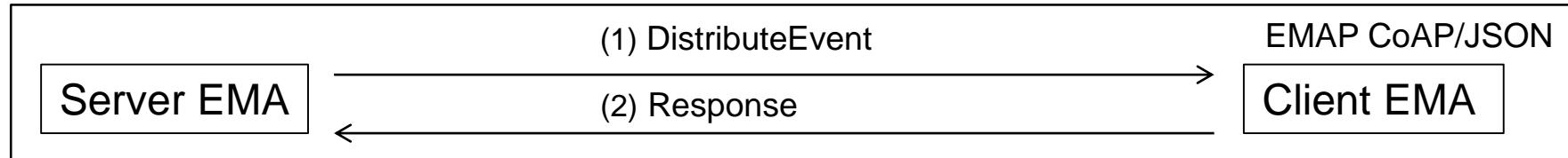
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Push



#### (1) DistributeEvent

PUT : /EMAP/CEMA/1.0b/Event

Payload : DistributeEvent

```
DistributeEvent Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID" : String,
    "response" : Array,
    "event" : Array,
    "responseRequired" : String,
    "service" : String,
    "time" : Date
}
```

```
response Array{
    "requestID" : String,
    "responseCode" : Integer,
    "responseDescription" : String
}
```

#### (1) DistributeEvent

EMAP CoAP/JSON

#### (2) Response

Client EMA

```
event Array{
    "eventID" : String,
    "eventSignals" : Array,
    "modificationNumber" : Integer,
    "modificationReason" : String,
    "priority" : Integer,
    "marketContext" : String,
    "createdDataTime" : Date,
    "eventStatus" : String,
    "testEvent" : Boolean,
    "vtnComment" : String,
    "dtstart" : Date,
    "duration" : String,
    "properties" : String,
    "components" : String,
    "specificDestEMA" : String,
    "tolerance" : String,
    "notification" : String,
    "rampUp" : String,
    "recovery" : String
}
```

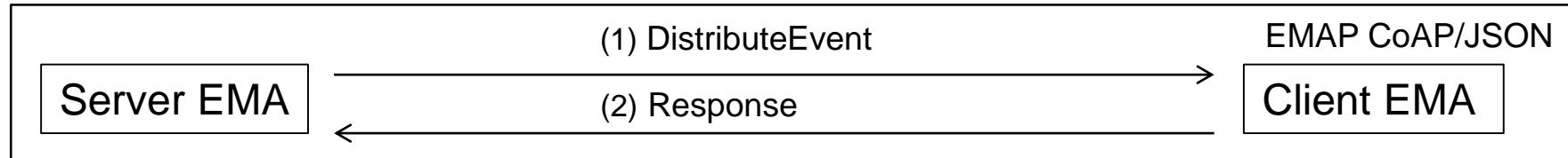
```
eventSignals Object{
    "eventSignal" : String,
    "intervals" : Array,
    "signalName" : String,
    "signalType" : String,
    "signalID" : String,
    "currentValue" : Double,
    "threshold" : Double,
    "capacity" : Double,
    "price" : Integer,
    "unit" : String
}
```

```
intervals Array{
    "duration" : String,
    "uid" : Integer,
    "value" : Double
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Push



(2) Response

PUT : /EMAP/SEMA/1.0b/Event

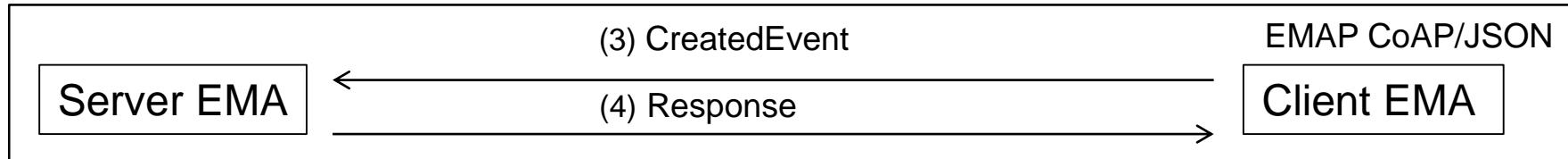
Payload : Response

```
Response Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode" : Integer,
    "responseDescription": String,
    "service" : String,
    "time" : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : push



(3) CreatedEvent

PUT : /EMAP/SEMA/1.0b/Event

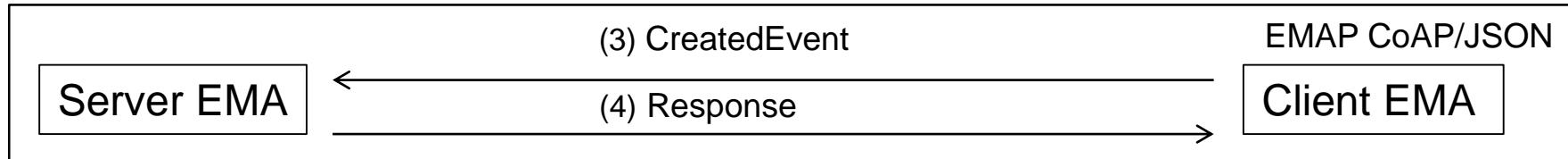
Payload : CreatedEvent

```
CreatedEvent Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode": Integer,
    "reponseDescription": String,
    "optType": String,
    "eventID" : String,
    "modificationNumber" : Integer,
    "service" : String,
    "time" : Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : push



(4) Response

PUT : /EMAP/CEMA/1.0b/Event

Payload : Response

```
Response Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “requestID”: String,
    “responseCode” : Integer,
    “responseDescription”: String,
    “service”: String,
    “time” : Date
}
```

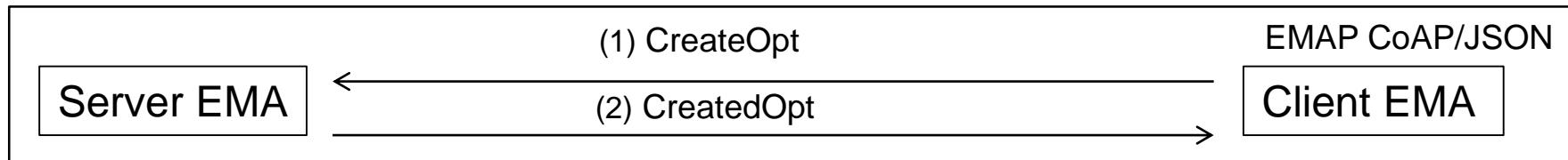
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : opt



(1) CreateOpt

PUT : /EMAP/SEMA/1.0b/Opt

Payload : CreateOpt

```
CreateOpt Object{
    “SrcEMA” : String,
    “DestEMA” : String,
    “optID”: String,
    “optType”: String,
    “optReason”: String,
    “requestID” : String,
    “marketContext” : String,
    “createdDateTime”: Date,
    “service” : String,
    “available” : Array,
}
```

```
available Array{
    “dtstart” : Date,
    “duration” : String,
    “requestPower” : double,
    “startYMD”: Integer,
    “startTime”: Integer,
    “endYMD”: Integer,
    “endTime”: Integer,
}
```

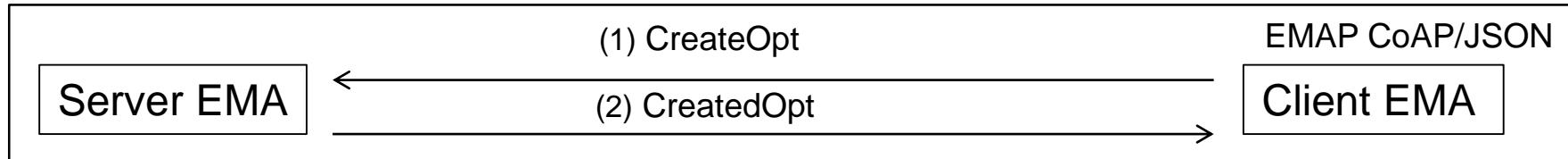
파란색 : 기존 OpenADR Tag 부분

빨간색 : OpenADR 확장된 Profile

초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) :opt



(2) CreatedOpt

PUT : /EMAP/CEMA/1.0b/Opt

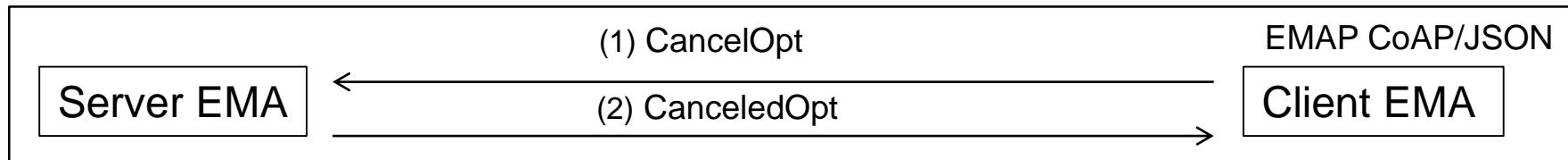
Payload : CreatedOpt

```
CreatedOpt Object{
    "SrcEMA" : String,
    "DestEMA" : String,
    "optID": String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription" : String,
    "service" : String,
    "optStatus": String
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : opt



(1) CancelOpt

PUT : /EMAP/SEMA/1.0b/Opt

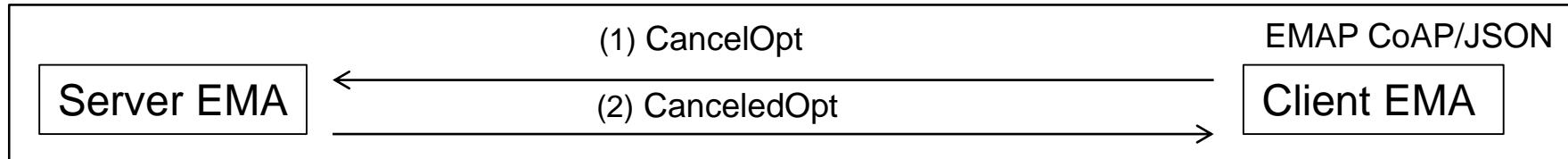
Payload : CancelOpt

```
CancelOpt JSONObject{
    "optID": String,
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "service" : String
    "time": Date
}
```

파란색 : 기존 OpenADR Tag 부분  
빨간색 : OpenADR 확장된 Profile  
초록색 : 삭제 또는 변경

## 2. Smart Home Energy Framework :

### 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Opt



(2) CanceledOpt

PUT : /EMAP/CEMA/1.0b/Opt

Payload : CanceledOpt

```
CanceledOpt Object{
    "optID": String,
    "SrcEMA" : String,
    "DestEMA" : String,
    "requestID": String,
    "responseCode": Integer,
    "responseDescription" : String,
    "service": String
}
```