Server Energy Management System

MIR Lab
http://mir.classlassanyang.ac.kr

Lecture Index

Base Conception

1. OpenADR
2. System Architecture

Architecture

3. EMS Overview
4. Package Explanation
6. Message Format

Practical Exercise

7. How to Execute MIR Program
   (EMS, VTN, EMA)
8. Experiment Procedure

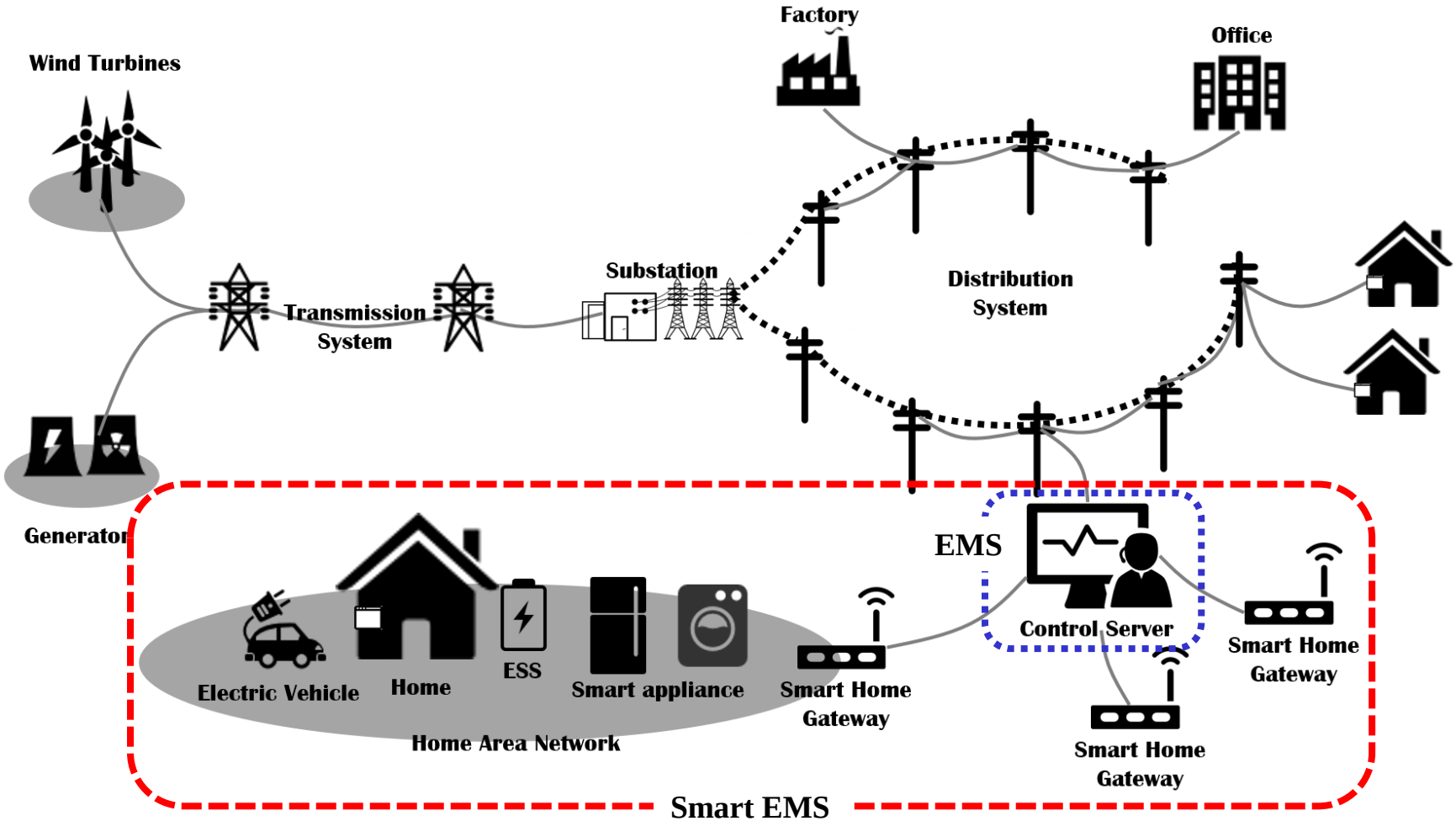
3. EMS Overview

3-1. System Architecture with Protocol

3-2. Program Overview

3-1. Optimization Overview

3-1. System Architecture



3-1. 스마트 에너지 홈 환경

스마트 에너지홈 환경

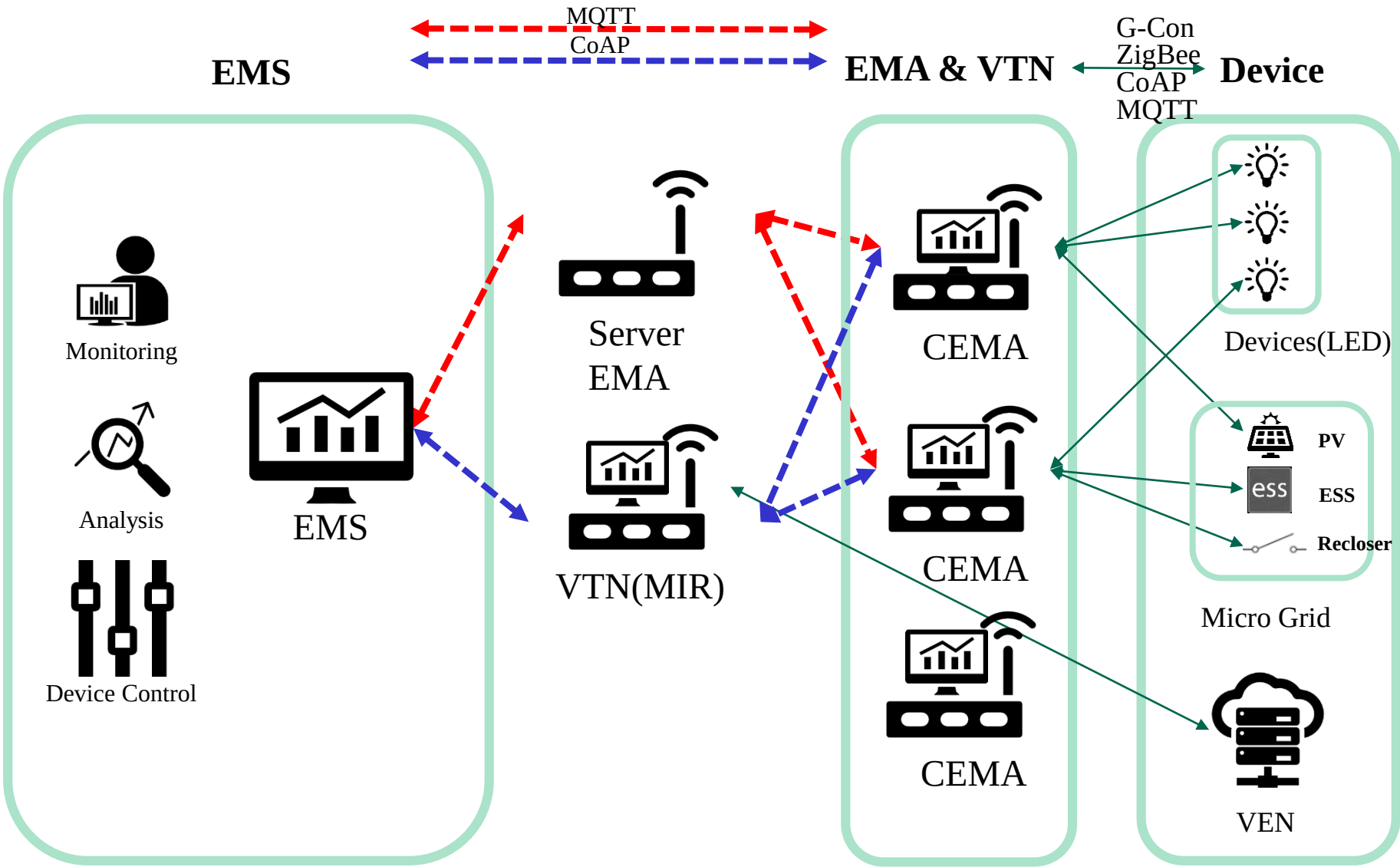
- 에너지 그리드 서비스를 제공하기 위한 스마트 에너지 홈 환경은 EMA가 관리하는 디바이스의 그룹과 상위의 서비스를 제공해 주는 서비스 제공자로 구분된다.
- EMA 하위에는 G-con, MQTT(MQ Telemetry Transport), CoAP(Constained Application Protocol)등과 같이 여러 가지 프로토콜로 구성된 디바이스의 그룹이 있고 이는 각각 아두이노와 라즈베리파이 등으로 구현되어 있다.
- 상위 서비스 제공자는 VTN, EMS, Utility등으로 구성되어 있으며 이들은 각각 에너지 소비에 대한 전략을 가지고 EMA에게 서비스를 제공해주는 역할을 한다.

3-1. 스마트 에너지 홈

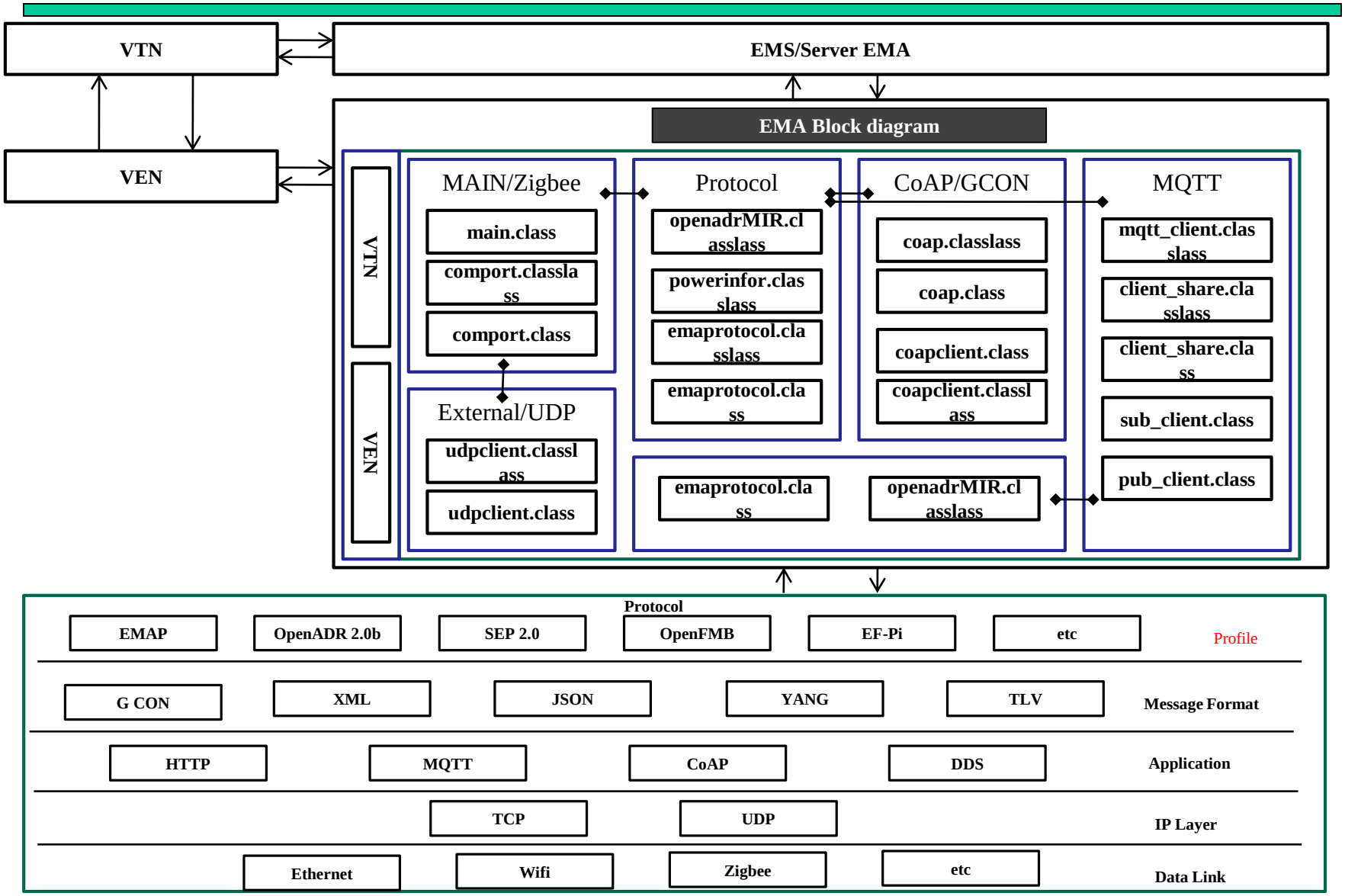
스마트 에너지 홈

- 스마트 에너지 홈란 일반적인 스마트 홈 환경에서 디바이스의 On/Off 제어 및 상태 보고 등과 같은 기능 외에도 자동적으로 지능화 된 수요반응 기능을 제공할 수 있는 환경을 의미한다.
- 앞서 이야기한 OpenADR 프로토콜을 통하여 전력 사업자와 사용자 간의 DR서비스 환경을 만들어서 지능적인 에너지 소모를 할 수 있도록 해야 한다.
- EMS는 EMA로부터 주기적으로 에너지 사용량을 보고 받아 모니터링 하게 되고, 디바이스를 직접 컨트롤 할 수 있도록 EMA에 명령을 전달 할 수 있도록 구성되어 있다.
- 다양한 모듈의 디바이스와 WiFi, Zigbee, IEEE 802.15.4통신을 위하여 해당 라이브러리에서 제공해주는 규격을 사용한다.

3-1. System Architecture with Protocol



3-2. Server EMA Overview



3-2. Server EMA Overview

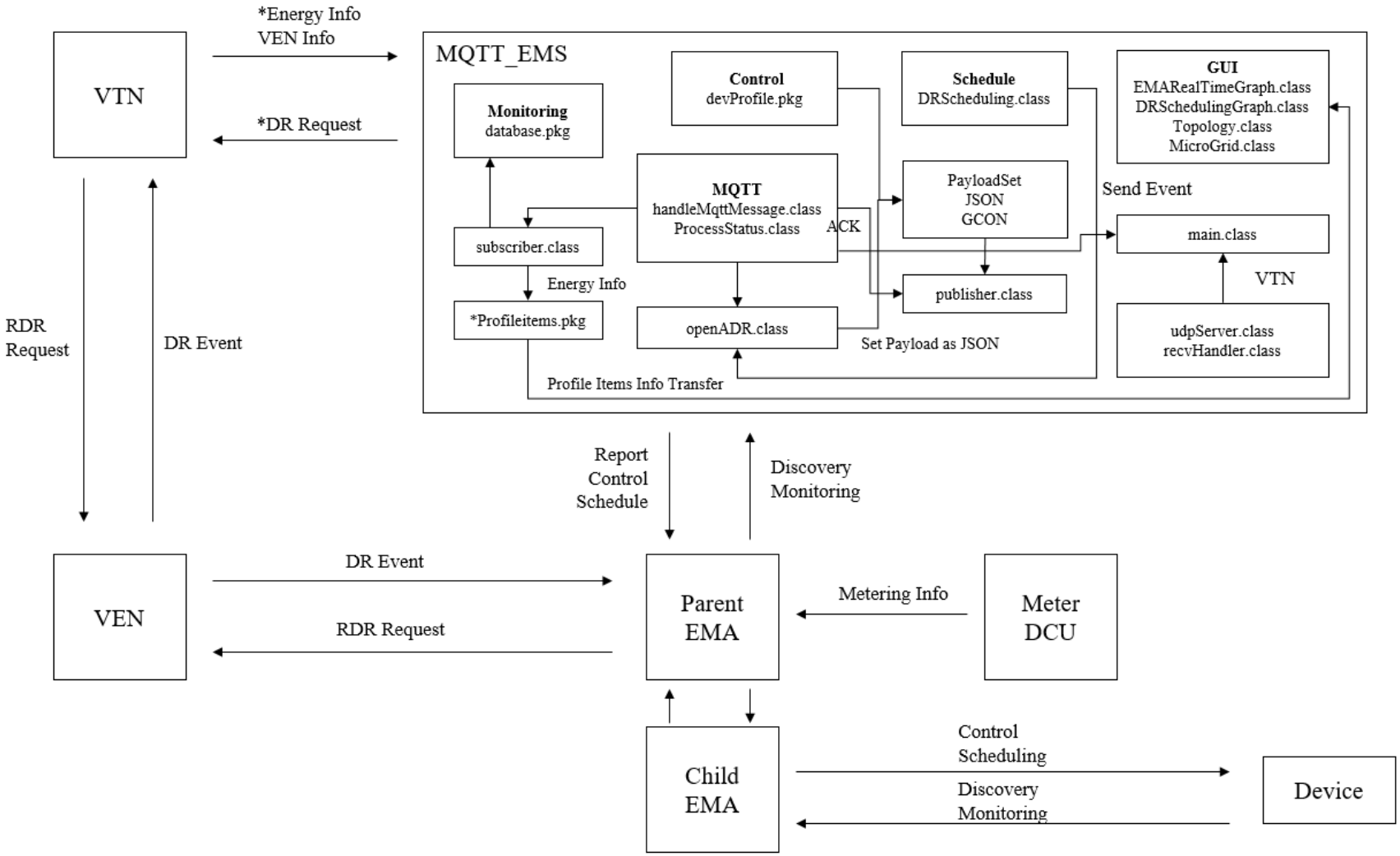
Application

•*GUI Subsystem* – To show the status and information.
•*Mobility Subsystem* – To be available for approaching anywhere.
•*Management Policy Subsystem* – Manage Policy

Energy Management System defines several primary services:

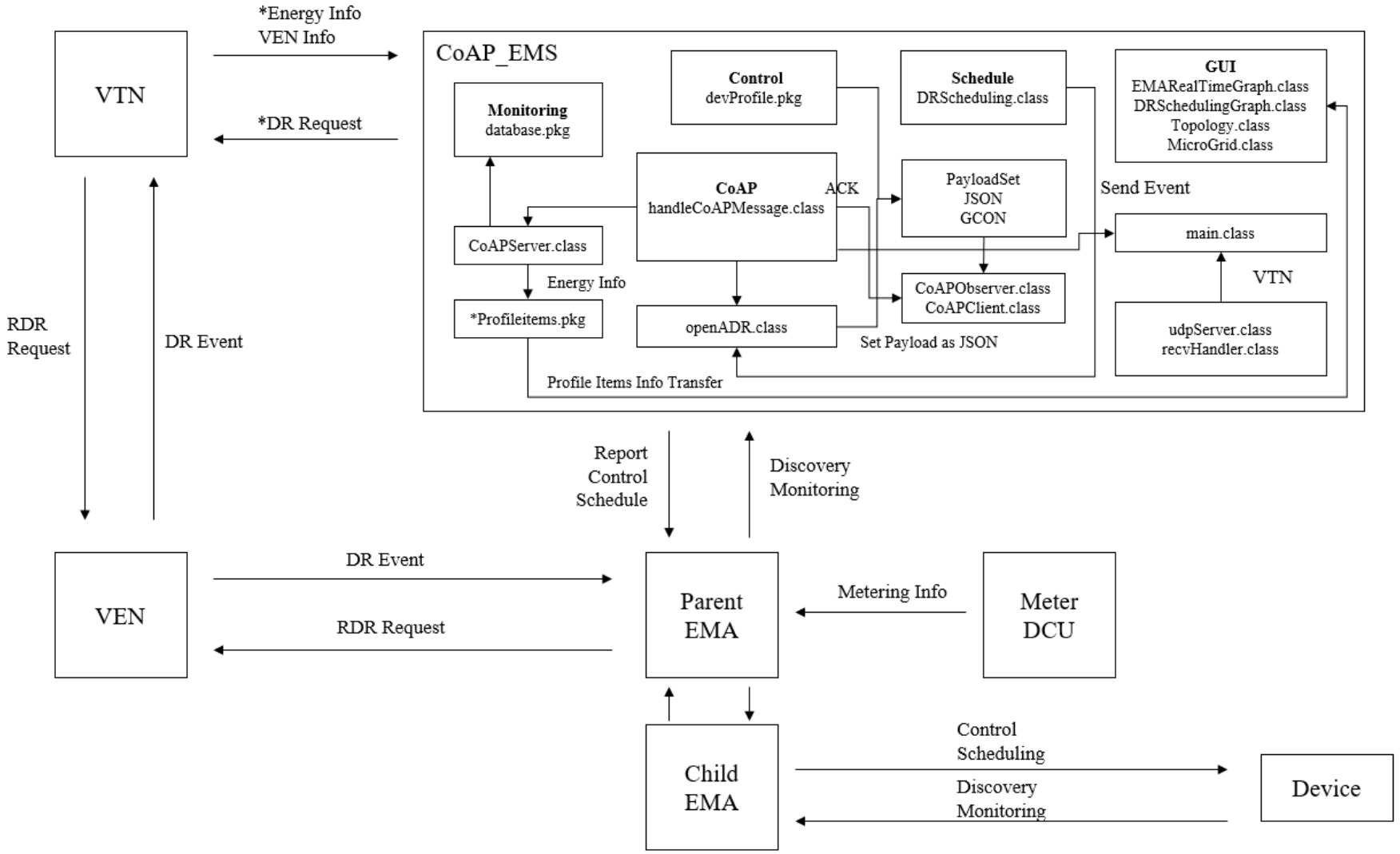
•*Scheduling Service* – Controls and manages EMA, Devices for optimization.
•*Report Subsystem* – Manage energy data to give processed report to energy customer.
•*Monitoring Subsystem* – Listen for EMA,VTN messages for manage and to make a topology and path
•*Profile Subsystem* - Manages inventory of sort of devices information and group
•*Control Subsystem* – To control child systems(EMA, OpenFMB and etc.) manually based on price policy(Incentive, Priced)

3-2. Server EMA Program Architecture



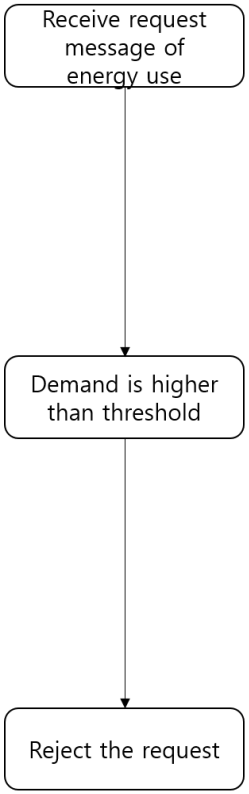
*DR Request: EMS가 VTN의 부하감축을 직접적으로 할 수는 없다. VTN Energy 정보를 바탕으로 최대 수요의 억제, 최대 부하의 이전, 기
*Energy: 소비량, 저장량, 생산량, 사용량, Threshold, Load Flow

3-2. Server EMA Program Architecture

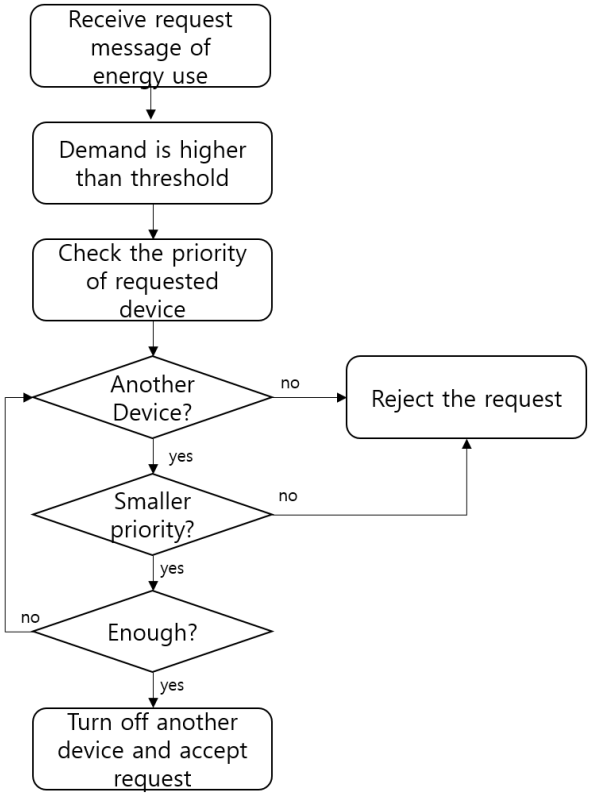


*DR Request: EMS가 VTN의 부하감축을 지정정으로 할 수는 없다. VTN Energy 정보를 바탕으로 최대 수요의 언제, 최대 부하의 이전 ...

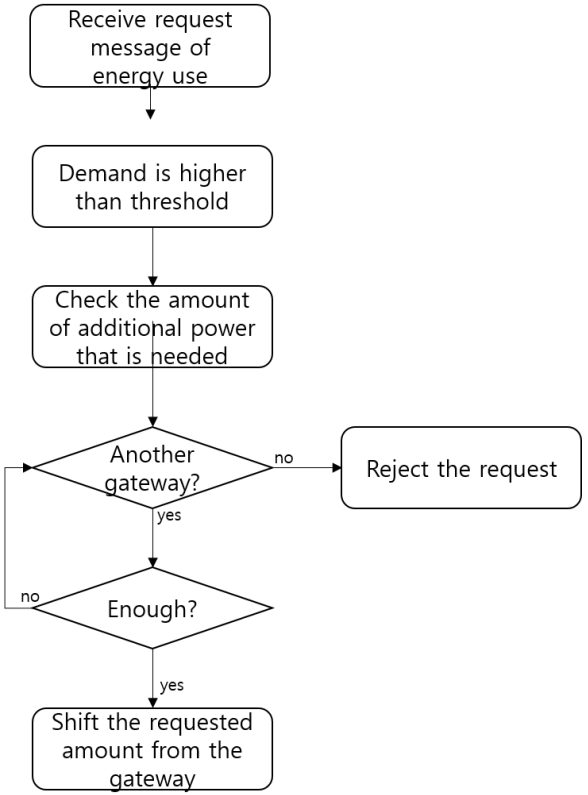
3-3. Server EMA Optimization Overview



(a) Threshold



(b) Threshold with priority



(c) Negotiation

Appendix. EMA Overview

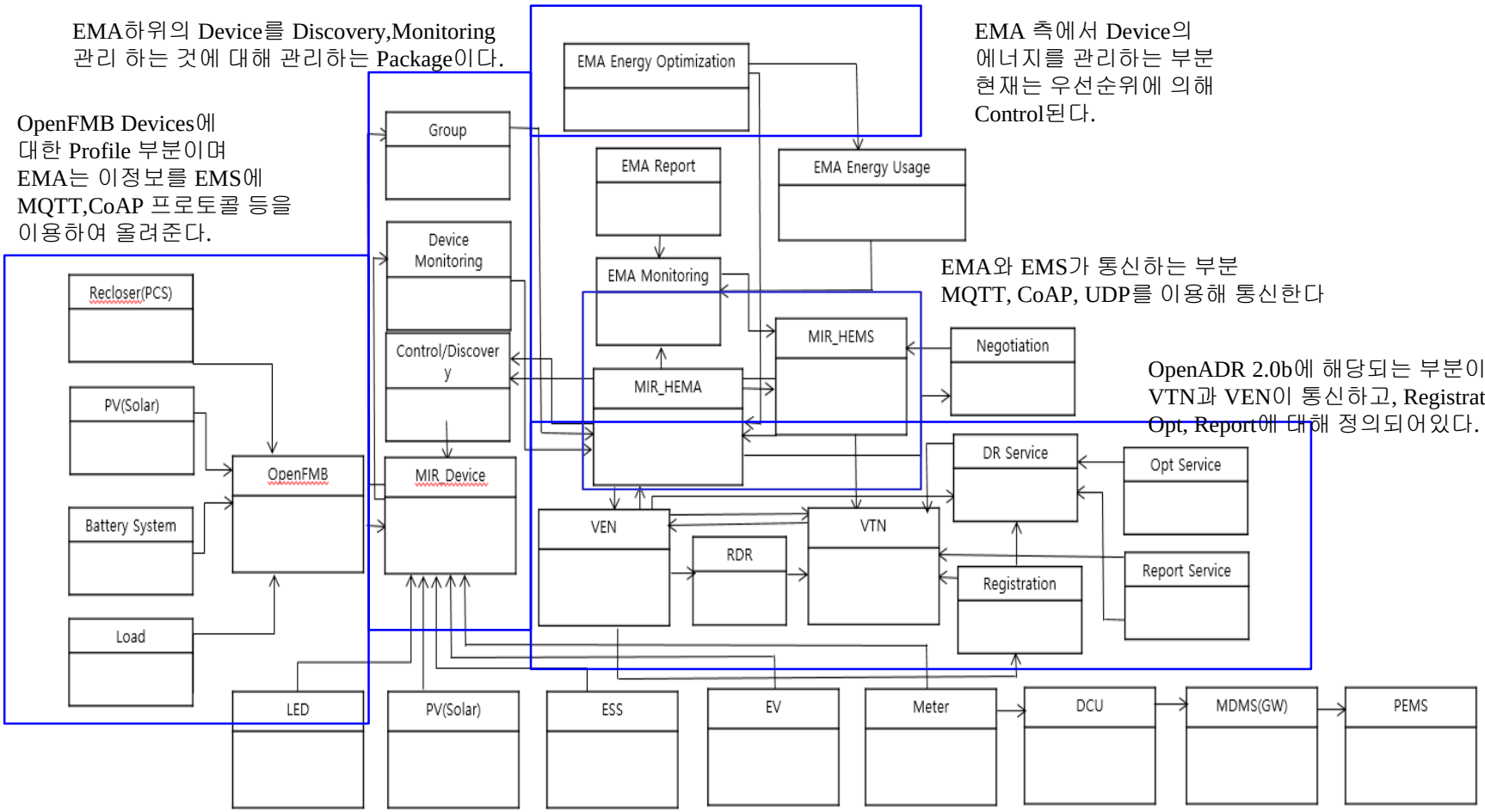
EMA하위의 Device를 Discovery,Monitoring 관리 하는 것에 대해 관리하는 Package이다.

OpenFMB Devices에 대한 Profile 부분이며 EMA는 이정보를 EMS에 MQTT,CoAP 프로토콜 등을 이용하여 올려준다.

EMA 측에서 Device의 에너지를 관리하는 부분 현재는 우선순위에 의해 Control된다.

EMA와 EMS가 통신하는 부분 MQTT, CoAP, UDP를 이용해 통신한다

OpenADR 2.0b에 해당되는 부분이다. VTN과 VEN이 통신하고, Registration, DR, Opt, Report에 대해 정의되어있다.

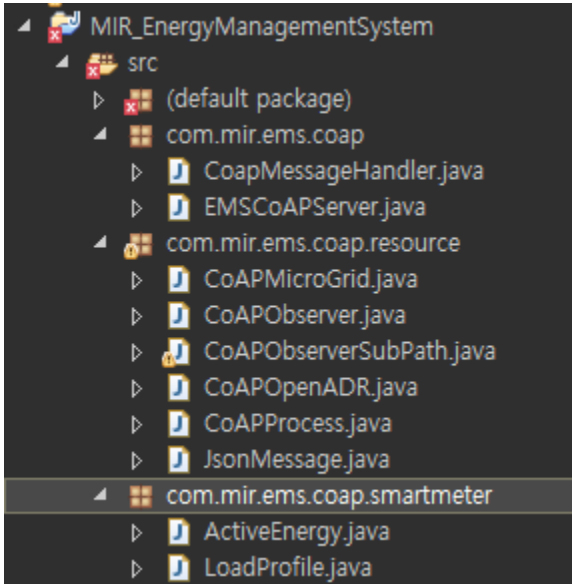


Smart Meter -> DCU
-> MDMS(Future) -> EMS

4. EMS : Package Explanation
OpenADR

In MIR Lab, We are using californium CoAP library

CoAP Package



com.mir.ems.classoap

- EMSCoAPServer :
CoAP Server
- CoAPMessageHanlder :
Handling the message that receive from CoAP Client

com.mir.ems.classoap.resource

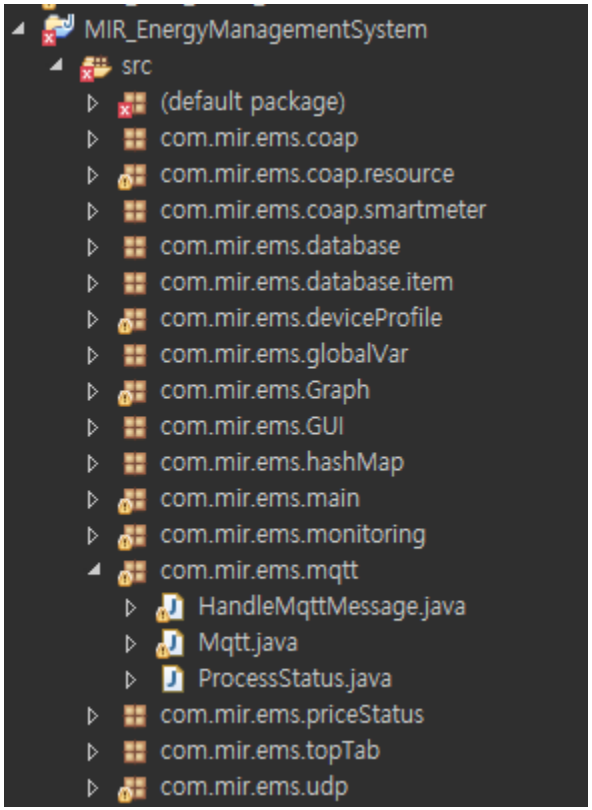
-CoAPMicroGrid:
Restful API (Only use PUT Method)

-CoAPObserver:
It is super class of CoAPObserverSubPath
-CoAPObserver:
Send Push Message when Event occur
-CoAPOpenADR:
Process of OpenADR2.0b(e.g. queryRegistration)
-JsonMessage:
Parsing JSON Type Message that receive from client

4. EMS : Package Explanation
OpenADR

In MIR Lab, We are using paho MQTT library

MQTT Package



com.mir.ems.mqtt

-Mqtt:

Start Publish, Subscribe and MQTT Client

-HandleMqttMessage:

Handling the JSON and Text message and publish Message to MQTT Broker

-ProcessStatus:

Save the Process value (Poll or Event)

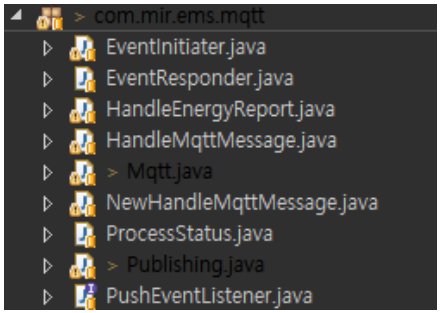
4. EMS : Package Explanation
OpenADR

Protocol

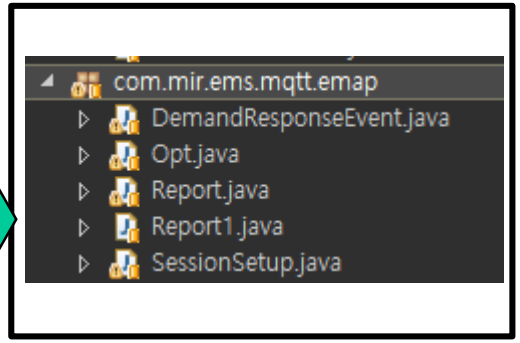
Service

OpenADR Model Package

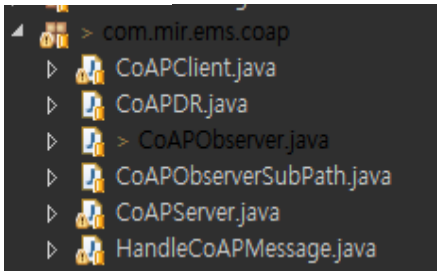
MQTT



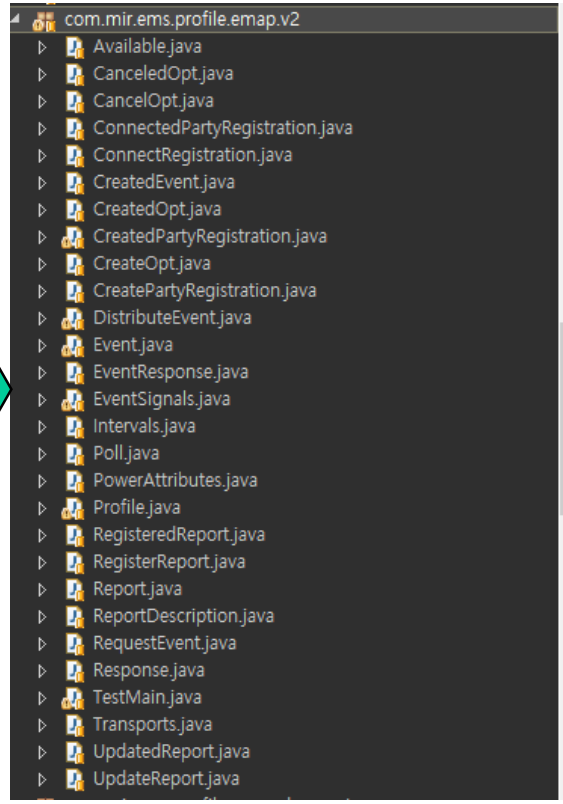
프로토콜 서비스 참조



COAP



각 서비스에 맞는
모델링 참조



4. EMS : Package Explanation
OpenADR – Server Side

CoAP Server Class : 수/발신 메시지 서비스에 따른 분류

… [Line 952]

```
                                server.add(new Emap("EMAP"));

                                server.add(new OpenADR("OpenADR"));

                                // Observe
                                server.add(new CoAPObserver("OpenADR2.0b"));
                                server.add(new CoAPObserver("EMAP1.0b"));

public class Emap extends CoapResource {

        public Emap(String name) {
                                // TODO Auto-generated constructor stub
                                super(name);

                                add(new SystemID(global.SYSTEM_ID, name));

        }
}
….
                                // OpenADR
                                add(new SessionSetup("SessionSetup"));
                                add(new Report("Report"));
                                add(new Opt("Opt"));
                                add(new DemandResponseEvent("Event"));
                                add(new DemandResponseEvent("Poll"));
```

4. EMS : Package Explanation
OpenADR - Server Side

CoAP Observe Class[Push] 별도 Class : COAP Client Observe function for Push

```
public CoAPObserverSubPath(String name, String parentPath) {
        super(name);
        this.name = name;
        setObservable(true);                                    // Observe 활성화
        setObserveType(Type.NON);
        getAttributes().setObservable();

        setParentPath(parentPath);

        Timer timer = new Timer();
        timer.schedule(new UpdateTask(), 0, 1);
}

[Line 44] : Observe 상태 체크
private class UpdateTask extends TimerTask {
        public void run() {

                if (global.getObs_emaProtocolCoAP_EventFlag().classontainsKey(name)) {
                        if (global.getObs_emaProtocolCoAP_EventFlag().get(name).isEventFlag()) {
                                changed();

                        }
                }
        }
}
```

4. EMS : Package Explanation
OpenADR - Server Side

CoAP Observe Class[Push] 별도 Class : COAP Client Observe function for Push

… [Line 63]Observe Initial

```
Response response = new Response(ResponseCode.classONTENT);
if ((!global.getObs_emaProtocolCoAP_EventFlag().classontainsKey(name))
                                || (!global.getObs_emaProtocolCoAP_EventFlag().get(name).isEventFlag())){

        response.setPayload("Initial_Success");
        exchange.respond(response);
        new Thread(new Runnable() {

                public void run() {
                        global.obs_emaProtocolCoAP_EventFlag.put(name, new EMAP_CoAP_EMA_DR());
        }}).start();

}
```

[Line 152] Event Send

```
if (getParentPath().classontains("EMAP")) {

        new Thread(new Runnable() {
                public void run() {
                global.obs_emaProtocolCoAP_EventFlag.replace(name,
                new EMAP_CoAP_EMA_DR().setEventFlag(false));
                }
        }).start();

        …. Event Send
}
```

4. EMS : Package Explanation
OpenADR - Server Side

MQTT Class : 수/발신 메시지에 따라 서비스 분류 (Session Setup/Report/Event/Opt)

… [Line 183]

```
if (topicParse[1].equals("OpenADR")) {

        String profileVersion = "OpenADR2.0b";
        String service = msg_json.getString("service");
        service = service.replaceAll("oadr", "");
        // Session Setup
        if (topicParse[4].equals("SessionSetup")) {
                if (service.matches("QueryRegistration|oadrQueryRegistration"))
                service = "CONNECTREGISTRATION";
                new SessionSetup(client, service, msg_json, profileVersion).start();
        }

        // Report
        else if (topicParse[4].equals("Report")) {
                new Report(client, service, msg_json, profileVersion).start();
        }
        // Event
        else if (topicParse[4].matches("Event|Poll")) {
                new DemandResponseEvent(client, service, msg_json, profileVersion).start();
        }
        // Opt
                else if (topicParse[4].matches("Opt")) {
                new Opt(client, service, msg_json, profileVersion).start();
        }


}
```

4. EMS : Package Explanation
OpenADR - Server Side

Session Setup/Report/Event/Opt Class : 상세 서비스 분류 ConnectRegistration, Poll…

… [Line 182]

```
switch (type) {
                case CONNECTREGISTRATION:
                        this.setPayload = acknowledgeCONNECTREGISTRATION(payload);
                        break;
                case CREATEPARTYREGISTRATION:
                        this.setPayload = acknowledgeCREATEPARTYREGISTRATION(payload);
                        break;
                case REGISTERREPORT:
                        this.setPayload = acknowledgeREGISTERREPORT(payload);
                        break;
                case POLL:
                        this.setPayload = acknowledgePOLL(payload);
                        break;
                case REGISTEREDREPORT:
                        this.setPayload = acknowledgeREGISTEREDREPORT(payload);
                        break;
                case REQUESTEVENT:
                        this.setPayload = acknowledgeREQUESTEVENT(payload);
                        break;
                case CANCELPARTYREGISTRATION:
                        this.setPayload = acknowledgeCANCELPARTYREGISTRATION(payload);

                        break;
```

4. EMS : Package Explanation
OpenADR - Server Side

각 CreatedPartyRegistration Class : 상세 서비스 CreatedPartyRegistration Message Build up

… [Line 101] JSON Message build up

```
public class CreatedPartyRegistration {

        private String srcEMA, destEMA, responseDescription, requestID, duration, service, registrationID;
        private int responseCode;
        private String profile;

        public CreatedPartyRegistration() {

        }
        @Override
        public String toString() {


                return "{\"vtnID" + "\":" + "\"" + getSrcEMA() + "\"" + ", "
                + "\"venID" + "\":" + "\"" + getDestEMA() + "\"" + ", "
                +"\"responseCode" + "\":" + "\"" + getResponseCode() + "\"" + ", "
                +"\"responseDescription" + "\":" + "\"" + getResponseDescription() + "\"" + ", "
                +"\"requestID" + "\":" + "\"" + getRequestID() + "\"" + ", "
                +"\"duration" + "\":" + "\"" + getDuration() + "\"" + ", "
                +"\"service" + "\":" + "\"" + getService() + "\"" + ", "
                +"\"registrationID" + "\":" + "\"" + getRegistrationID() + "\"" + ", "
                + "\"oadrProfile" + "\": "+ getProfile() + "}";

        }
```

4. EMS : Package Explanation
OpenADR - Client Side

MQTT Class : 수/발신 메시지에 따라 서비스 분류 (Session Setup/Report/Event/Opt)

```
… [Line 101]
if (topicParse[1].equals("OpenADR") && topicParse[2].equals(global.CHILD_ID)) {

Services services = Services.valueOf(topicParse[4]);
switch (services) {
case SessionSetup:
try {
sessionSetup(procedure, profileVersion);
} catch (JSONException e) {
e.printStackTrace();
}
break;
case Poll:
try {
poll(procedure, profileVersion);
} catch (JSONException e) {
e.printStackTrace();
}
break;
case Report:
report(procedure, profileVersion);
break;
case Opt:
opt(procedure, profileVersion);
break;
case Event:
poll(procedure, profileVersion);
break;
}
}
```

4. EMS : Package Explanation
OpenADR – Client Side

Session Setup/Report/Event/Opt Class : 상세 서비스 분류 ConnectRegistration, Poll…

```
public void sessionSetup(String procedure, String profileVersion) throws JSONException, InterruptedException {
String setPayload = "";
SessionSetup sessionSetup = SessionSetup.valueOf(procedure);
JSONObject jsonParse = new JSONObject(msgPayload);
double generate = 0, storage = 0, power = 0;

switch (sessionSetup) {
case ConnectedRegistration:

else if (profileVersion.equals("OpenADR2.0b_new")) {

com.mir.ems.profile.openadr.recent.CreatePartyRegistration cp = new com.mir.ems.profile.openadr.recent.CreatePartyRegistration();

cp.setHttpPullModel(this.connection.isPullModel());

cp.setProfileName("OpenADR2.0b");
cp.setReportOnly(false);
cp.setRequestID("requestID");
cp.setService("oadrCreatePartyRegistration");
cp.setSrcEMA(this.connection.getEmaID());
// cp.setTime(this.connection.getCurrentTime(System.currentTimeMillis()));
cp.setTransportName("MQTT");
cp.setXmlSignature(true);

String topic = "/OpenADR/" + global.getParentnNodeID() + "/2.0b/EiRegisterParty";
setPayload = cp.toString();
new Publishing().publishThread(this.client, topic, 0, setPayload.getBytes());

}
```

4. EMS : Package Explanation
OpenADR - Client Side

각 **QueryRegistration** Class : 상세 서비스 **QueryRegistration** Message Build up

```
… [Line 50] JSON Message build up
public class QueryRegistration {

int emaNum, requestID, version;
String venID;

public QueryRegistration(int emaNum, String venID, int requestID, int version){

setEmaNum(emaNum);
setRequestID(requestID);
setVenID(venID);
setVersion(version);

}

@Override
public String toString() {
return "{\"GW\":\"" + getEmaNum() + "\","
+ "\"VENID\":\"" + getVenID()+ "\","
+ "\"RequestID\":" + getRequestID() + ","
+ "\"Version\":" + getVersion()+"\""
+ "}";
}
```

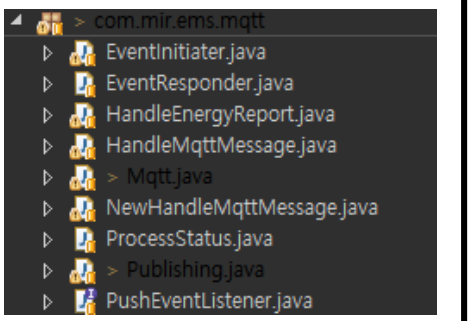
4. EMS : Package Explanation
EMAP

Protocol

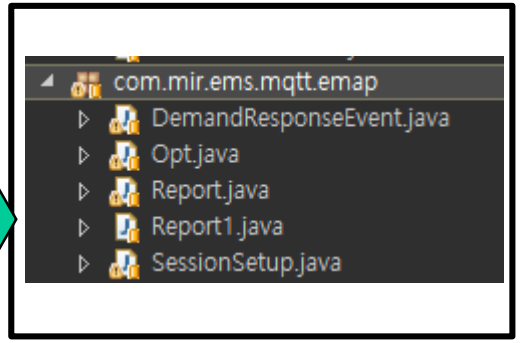
Service

EMAP Model Package

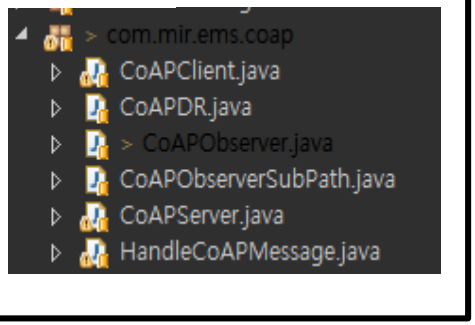
MQTT



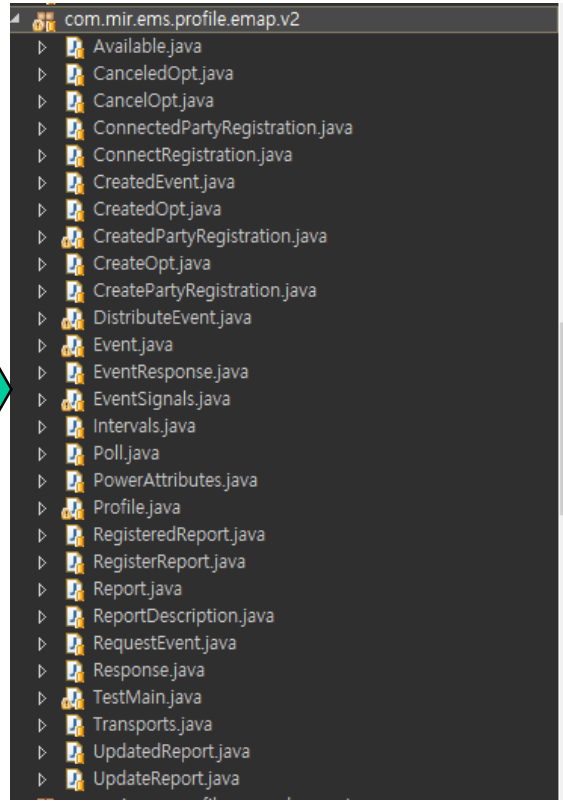
프로토콜 서비스 참조



COAP



각 서비스에 맞는
모델링 참조



4. EMS : Package Explanation
EMAP

CoAP Server Class : 수/발신 메시지 서비스에 따른 분류

… [Line 952]

```
                server.add(new Emap("EMAP"));

                server.add(new OpenADR("OpenADR"));

                // Observe
                server.add(new CoAPObserver("OpenADR2.0b"));
                server.add(new CoAPObserver("EMAP1.0b"));

public class Emap extends CoapResource {

        public Emap(String name) {
                // TODO Auto-generated constructor stub
                super(name);

                add(new SystemID(global.SYSTEM_ID, name));

        }
}
….
                // EMAP
                add(new SessionSetup("SessionSetup"));
                add(new Report("Report"));
                add(new Opt("Opt"));
                add(new DemandResponseEvent("Event"));
                add(new DemandResponseEvent("Poll"));
```

4. EMS : Package Explanation
EMAP

CoAP Observe Class[Push] 별도 Class : COAP Client Observe function for Push

```
public CoAPObserverSubPath(String name, String parentPath) {
        super(name);
        this.name = name;
        setObservable(true);                                    // Observe 활성화
        setObserveType(Type.NON);
        getAttributes().setObservable();

        setParentPath(parentPath);

        Timer timer = new Timer();
        timer.schedule(new UpdateTask(), 0, 1);
}

[Line 44] : Observe 상태 체크
private class UpdateTask extends TimerTask {
        public void run() {

                if (global.getObs_emaProtocolCoAP_EventFlag().classontainsKey(name)) {
                        if (global.getObs_emaProtocolCoAP_EventFlag().get(name).isEventFlag()) {
                                changed();

                        }
                }
        }
}
```

4. EMS : Package Explanation
EMAP

CoAP Observe Class[Push] 별도 Class : COAP Client Observe function for Push

… [Line 63]Observe Initial

```
Response response = new Response(ResponseCode.classONTENT);
if ((!global.getObs_emaProtocolCoAP_EventFlag().classontainsKey(name))
                                || (!global.getObs_emaProtocolCoAP_EventFlag().get(name).isEventFlag())){

        response.setPayload("Initial_Success");
        exchange.respond(response);
        new Thread(new Runnable() {

                public void run() {
                        global.obs_emaProtocolCoAP_EventFlag.put(name, new EMAP_CoAP_EMA_DR());
        }}).start();

}
```

[Line 152] Event Send

```
if (getParentPath().classontains("EMAP")) {

        new Thread(new Runnable() {
                public void run() {
                global.obs_emaProtocolCoAP_EventFlag.replace(name,
                new EMAP_CoAP_EMA_DR().setEventFlag(false));
                }
        }).start();

        …. Event Send
}
```

4. EMS : Package Explanation
EMAP

MQTT Class : 수/발신 메시지에 따라 서비스 분류 (Session Setup/Report/Event/Opt)

… [Line 147]

```
if (topicParse[1].equals("EMAP")) {

String profileVersion = "EMAP1.0b";

if (msg_json.getString("DestEMA").equals(global.getSYSTEM_ID())) {

String service = msg_json.getString("service");

// Session Setup
if (topicParse[4].equals("SessionSetup")) {
        new SessionSetup(client, service, msg_json, profileVersion).start();
}
// Report
else if (topicParse[4].equals("Report")) {
        new Report(client, service, msg_json, profileVersion).start();
}
// Event
else if (topicParse[4].matches("Event|Poll")) {
        new DemandResponseEvent(client, service, msg_json, profileVersion).start();
}
// Opt
else if (topicParse[4].matches("Opt")) {
        new Opt(client, service, msg_json, profileVersion).start();
}
}
}
```

4. EMS : Package Explanation
EMAP

Session Setup/Report/Event/Opt Class : 상세 서비스 분류 ConnectRegistration, Poll…

… [Line 182]

```
switch (type) {
        case CONNECTREGISTRATION:
                this.setPayload = acknowledgeCONNECTREGISTRATION(payload);
                break;
        case CREATEPARTYREGISTRATION:
                this.setPayload = acknowledgeCREATEPARTYREGISTRATION(payload);
                break;
        case REGISTERREPORT:
                this.setPayload = acknowledgeREGISTERREPORT(payload);
                break;
        case POLL:
                this.setPayload = acknowledgePOLL(payload);
                break;
        case REGISTEREDREPORT:
                this.setPayload = acknowledgeREGISTEREDREPORT(payload);
                break;
        case REQUESTEVENT:
                this.setPayload = acknowledgeREQUESTEVENT(payload);
                break;
        case CANCELPARTYREGISTRATION:
                this.setPayload = acknowledgeCANCELPARTYREGISTRATION(payload);

                break;
```

4. EMS : Package Explanation
EMAP

각 ConnectedPartyRegistration Class : 상세 서비스 ConnectedPartyRegistration Message Build up

… [Line 101] JSON Message build up

```
public class ConnectedPartyRegistration {

        private String srcEMA, destEMA, responseDescription, requestID, duration, service, version, time;
        private int responseCode;
        private String profile;

        @Override
        public String toString() {

                return "{\"SrcEMA" + "\":" + "\"" + getSrcEMA() + "\"" + ", "
                                + "\"DestEMA" + "\":" + "\"" + getDestEMA() + "\"" + ", "
                                +"\"responseCode" + "\":" + "\"" + getResponseCode() + "\"" + ", "
                                +"\"responseDescription" + "\":" + "\"" + getResponseDescription() + "\"" + ",
"
                                +"\"requestID" + "\":" + "\"" + getRequestID() + "\"" + ", "
                                +"\"duration" + "\":" + "\"" + getDuration() + "\"" + ", "
                                +"\"service" + "\":" + "\"" + getService() + "\"" + ", "
                                +"\"version" + "\":" + "\"" + getVersion() + "\"" + ", "
                                +"\"time" + "\":" + "\"" + getTime() + "\"" + ", "
                                +"\"registrationID" + "\":" + "\"" + "" + "\"" + ", "
                                + "\"profile" + "\": "+ getProfile() + "}";
        }
```

4. EMS : Package Explanation
EMAP - Client Side

MQTT Class : 수/발신 메시지에 따라 서비스 분류 (Session Setup/Report/Event/Opt)

… [Line 101] JSON Message build up

```
if (topicParse[1].equals("EMAP") && topicParse[2].equals(global.CHILD_ID)) {
String profileVersion = "EMAP1.0b“;
if (msg_json.getString("DestEMA").equals(global.CHILD_ID)) ;
String procedure = msg_json.getString("service");
Services services = Services.valueOf(topicParse[4]);
switch (services) {
case SessionSetup:
try {
sessionSetup(procedure, profileVersion);
} catch (JSONException e) {
e.printStackTrace();
}
break;
case Poll:
try {
poll(procedure, profileVersion);
} catch (JSONException e) {
e.printStackTrace();
}
break;
case Report:
report(procedure, profileVersion);
break;
case Opt:
opt(procedure, profileVersion);
break;
case Event:
poll(procedure, profileVersion);
break;
```

4. EMS : Package Explanation
EMAP – Client Side

Session Setup/Report/Event/Opt Class : 상세 서비스 분류 ConnectRegistration, Poll…

```
public void sessionSetup(String procedure, String profileVersion) throws JSONException, InterruptedException {
String setPayload = "";
SessionSetup sessionSetup = SessionSetup.valueOf(procedure);
JSONObject jsonParse = new JSONObject(msgPayload);
double generate = 0, storage = 0, power = 0;

switch (sessionSetup) {
case ConnectedRegistration:

if (profileVersion.equals("EMAP1.0b")) {

com.mir.ems.profile.emap.v2.CreatePartyRegistration cp = new com.mir.ems.profile.emap.v2.CreatePartyRegistration();

cp.setDestEMA(global.getParentnNodeID());

cp.setHttpPullModel(this.connection.isPullModel());

cp.setProfileName("EMAP1.0b");
cp.setReportOnly(false);
cp.setRequestID("requestID");
cp.setService("CreatePartyRegistration");
cp.setSrcEMA(this.connection.getEmaID());
cp.setTime(this.connection.getCurrentTime(System.currentTimeMillis()));
cp.setTransportName("MQTT");
cp.setXmlSignature(true);

String topic = "/EMAP/" + global.getParentnNodeID() + "/1.0b/SessionSetup";
setPayload = cp.toString();
new Publishing().publishThread(this.client, topic, 0, setPayload.getBytes());
```

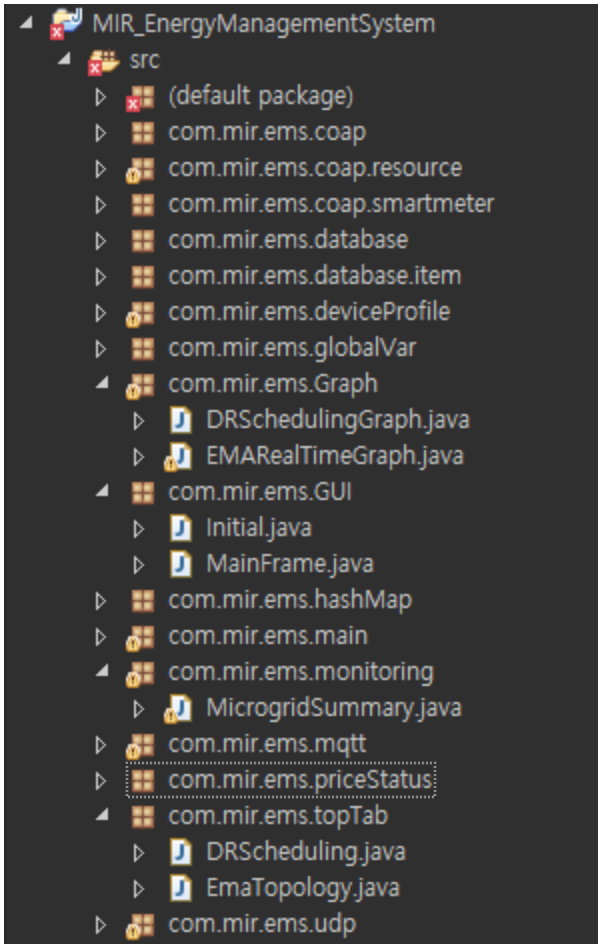
4. EMS : Package Explanation
EMAP - Client Side

각 **ConnectRegistration** Class : 상세 서비스 **ConnectRegistration** Message Build up

```
… [Line 101] JSON Message build up
public class ConnectRegistration {
// Mapping field
private String srcEMA;
private String destEMA;
private int requestID;private int version;
public ConnectRegistration(String srcEMA, String destEMA, int requestID, int version, int customerPriority,
String qoS, String service, String type, String time) {
super();
this.srcEMA = srcEMA;
this.destEMA = destEMA;
this.requestID = requestID;
this.version = version;
this.customerPriority = customerPriority;
this.qoS = qoS;
this.service = service;
this.type = type;
this.time = time;
}
@Override
public String toString() {
return "{\"SrcEMA\":\"" + srcEMA + "\","
+ "\"DestEMA\":\"" + destEMA + "\","
+ "\"requestID\":" + requestID + ","
+ "\"version\":" + version + ","
+ "\"customerPriority\":" + customerPriority + ","
+ "\"QoS\":\"" + qoS + "\","
+ "\"service\":\"" + service + "\","
+ "\"type\":\"" + type + "\","
+ "\"time\":\"" + time + "\""
+ "}";
```

4. EMS : Package Explanation
Monitoring

Monitoring



com.mir.ems.monitoring

-MicrogridSummary:

To show the microgrid status (ess, pv, resource)

com.mir.ems.Graph

- EMARealTimeGraph:

To show EMA'S Energy USE on Real Time Graph

- EnergyGraph:

To show total energy use on Real Time Graph-

-EMATopology:

To show EMA Topology

- DRSchedulingGraph:

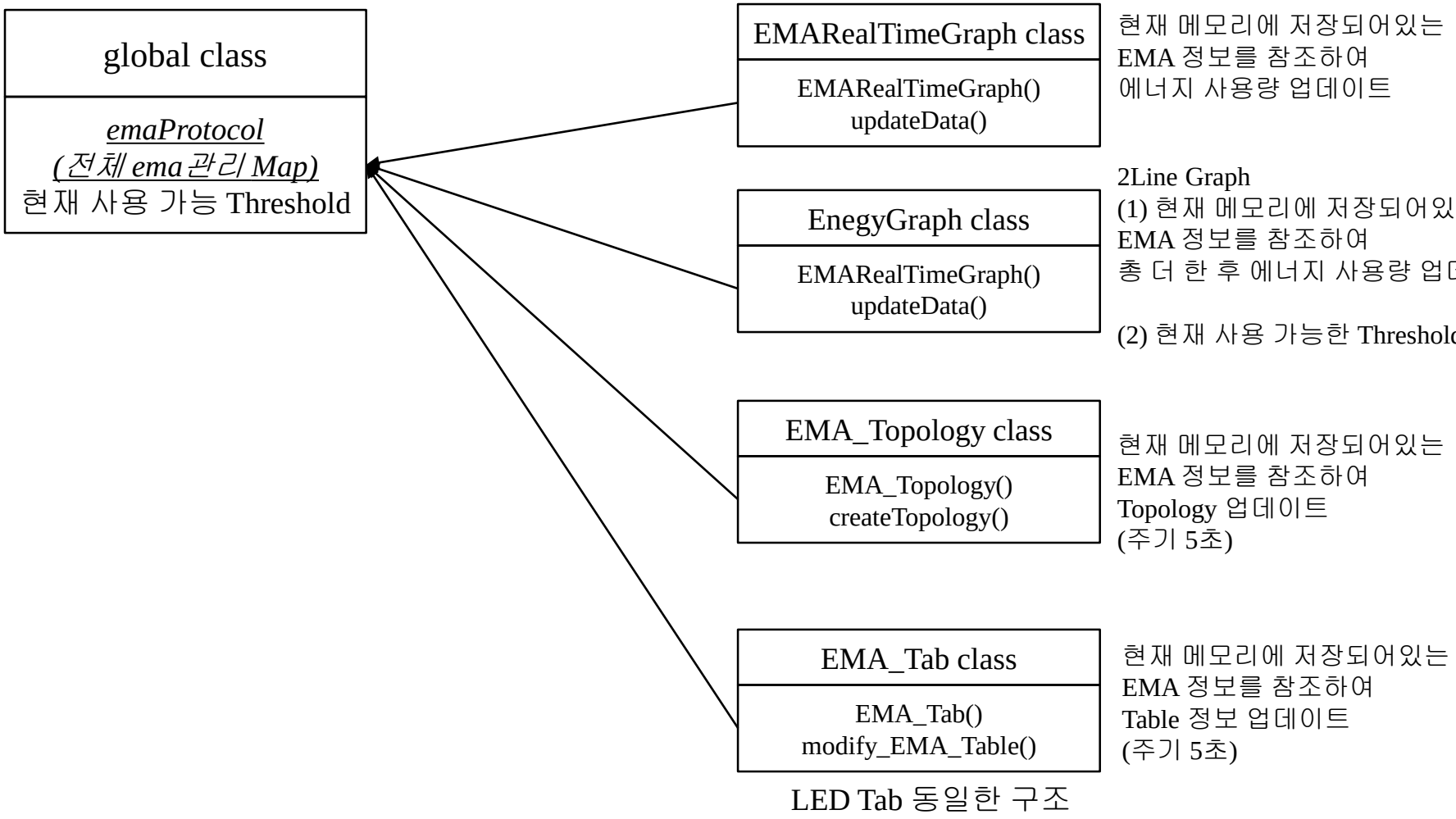
To show when you give a event to EMA

com.mir.ems.deviceProfile

: In order to store Devices profile in JVM

- **EMA_TAB**
- **LED_TAB**

4. EMS : Package Explanation
Monitoring



4. EMS : Package Explanation
Monitoring : EMARealTimeGraph

EMARealTimeGraph Class : 각각의 EMA에 대한 실시간 그래프

… [Line 49] 그래프 생성 및 그래프 업데이트 주기 설정

```
public EMARealTimeGraph() {

        setBounds(14, 60, 1467, 700);
        final XYChart chart = getChart();
        setLayout(null);
        XChartPanel chartPanel = new XChartPanel(chart);
        chartPanel.setBackground(Color.WHITE);
        chartPanel.setBounds(0, 0, 1467, 700);
        add(chartPanel);

        TimerTask chartUpdaterTask = new TimerTask() {
                @Override
                public void run() {

                        updateData();

                        javax.swing.SwingUtilities.invokeLater(new Runnable() {
                                public void run() {
                                        repaint();
                                }
                        });
                }
        };
        Timer timer = new Timer();
        timer.scheduleAtFixedRate(chartUpdaterTask, 2000, 2000); (2초 주기로 그래프 업데이트)
}
```

4. EMS : Package Explanation
Monitoring : EMARealTimeGraph

EMARealTimeGraph Class : 각각의 EMA에 대한 실시간 그래프

… [Line 109] 그래프 업데이트 함수, X축 Y축 업데이트

```
@SuppressWarnings("unchecked")
public void updateData() {
		List<Date> newXdata = getCurrentTime();
		xData.addAll(newXdata);
		int emaListSize;
		emaListSize = global.emaProtocolCoAP.size();

		for (int i = 0; i < emaListSize; i++) {
			arr[i][1] = getRandomData((List<Double>) arr[i][1], i);
		}

		for(int i=emaListSize;i<20; i++){
			arr[i][1] = getRandomData((List<Double>) arr[i][1], i);
		}

		for(int i=0; i<20; i++){
			xyChart.updateXYSeries((String)arr[i][0], xData, (List<Double>)arr[i][1], null);
		}

}
```

4. EMS : Package Explanation
Monitoring : EnergyGraph

EnergyGraph Class : 각각의 EMA에 대한 실시간 그래프

… [Line 52] 그래프 생성 및 그래프 업데이트 주기 설정

```
public EnergyGraph() {

        final XYChart chart = getChart();

        setBounds(14, 60, 1467, 700);
        setLayout(null);

        @SuppressWarnings({ "rawtypes", "unchecked" })
        XChartPanel chartPanel = new XChartPanel(chart);
        chartPanel.setBackground(Color.WHITE);
        chartPanel.setBounds(0, 0, 1467, 700);
        add(chartPanel);

        TimerTask chartUpdaterTask = new TimerTask() {
        timer.scheduleAtFixedRate(chartUpdaterTask, 2000, 2000);
}
```

Timer timer = new Timer();

(2초 주기로 그래프 업데이트)

4. EMS : Package Explanation
Monitoring : EnergyGraph

EnergyGraph Class : 각각의 EMA에 대한 실시간 그래프

… [Line 196] Y축 Sin Graph, X축 현재 시간 업데이트 함수

```
        private List<Double> getYAXIS() {

                double radians = phase + (2 * Math.PI / 100 * val);
                val += 1;
                phase += ((2 * Math.PI * 2) / 20.0) / PERIOD;

                //Sin graph
                global.THRESHOLD = (YAXIS_TRANSFERENCE * Math.sin(radians) + BASEWATT) * 1000;
                global.AVAILABLE_THRESHOLD = (global.THRESHOLD - (global.THRESHOLD /
global.RESERVE_THRESHOLD_PERCENTAGE));
                global.RESERVE_THRESHOLD = global.THRESHOLD - global.AVAILABLE_THRESHOLD;

                yData.add(YAXIS_TRANSFERENCE * Math.sin(radians) + BASEWATT);

                return yData;
        }

        private List<Date> getEMAAXIS() {

                //현재 시간
                long now = System.classurrentTimeMillis();
                Date date = new Date(now);
                totalEMAxData.add(date);

                return totalEMAxData;
        }
```

4. EMS : Package Explanation
Monitoring : EMATopology

EMATopology Class : 각각의 EMA에 대한 토폴로지 그래프

… [Line 34] 토폴로지 그래프 생성

```
public EmaTopology() {
                java.net.URL emsUrl = EmaTopology.classlass.getResource("/IMAGE/dddd.png");
                System.setProperty("gs.ui.renderer", "org.graphstream.ui.j2dviewer.J2DGraphRenderer");
                Viewer viewer = new Viewer(graph, Viewer.ThreadingModel.GRAPH_IN_GUI_THREAD);
                viewer.disableAutoLayout();

                ViewPanel topologyPanel = (ViewPanel) viewer.addDefaultView(false);
                topologyPanel.setSize(1467, 700);
                add(topologyPanel);

                setBounds(14, 60, 1467, 700);
                setLayout(new BorderLayout(0, 0));
                setVisible(true);

                Node a = graph.addNode("EMS");
                a.addAttribute("ui.label", a.getId());

                int sum = 0;
                for (int i = 0; i < 20; i++) {
                                sum += (i * 20);
                }
                a.setAttribute("x", (sum / 40));
                a.setAttribute("y", 10);

                a.addAttribute("ui.style","text-alignment: above; size: 65px, 65px; shape: rounded-box; size-mode: fit; fill-
mode: image-scaled; fill-image: url('"+ emsUrl + "');");
                createTopology();
```

4. EMS : Package Explanation
Monitoring : EMATopology

EMATopology Class : 각각의 EMA에 대한 토폴로지 정보

… [Line 34] 그래프 생성 및 그래프 업데이트 주기 설정

```
TimerTask chartUpdaterTask = new TimerTask() {
        Node emaGroup = null;
        Node deviceGroup = null;
        @Override
        public void run() {
                int cnt = 0;
                int devCnt = 0;
                for (int i = 0; i < emaList.length; i++) {
                String key = emaList[i];
                if (!strSet.classontains(key.toString())) {
                        cnt += 1;
                try {
                        // NODE 추가
                        emaGroup = graph.addNode(key);
                        // NODE 생성 위치
                        emaGroup.setAttribute(＂x＂, (cnt * 10));
                        // NODE ID 설정
                        emaGroup.addAttribute("ui.label", emaGroup.getId());
                        emaGroup.setAttribute("y", 0);

                        //NODE 사이즈 설정
                                emaGroup.addAttribute(＂ui.style＂,
                         ＂text-alignment: under; size: 65px, 65px; shape: rounded-box; size-mode: fit; fill-mode:
image-scaled; fill-image: url(＇＂+ gatewayUrl +＂＇);＂);
                        // EDGE 설정
                        graph.addEdge(emsEdge + key, emsEdge, key);

        …..
```

4. EMS : Package Explanation
Monitoring : EMA_Tab

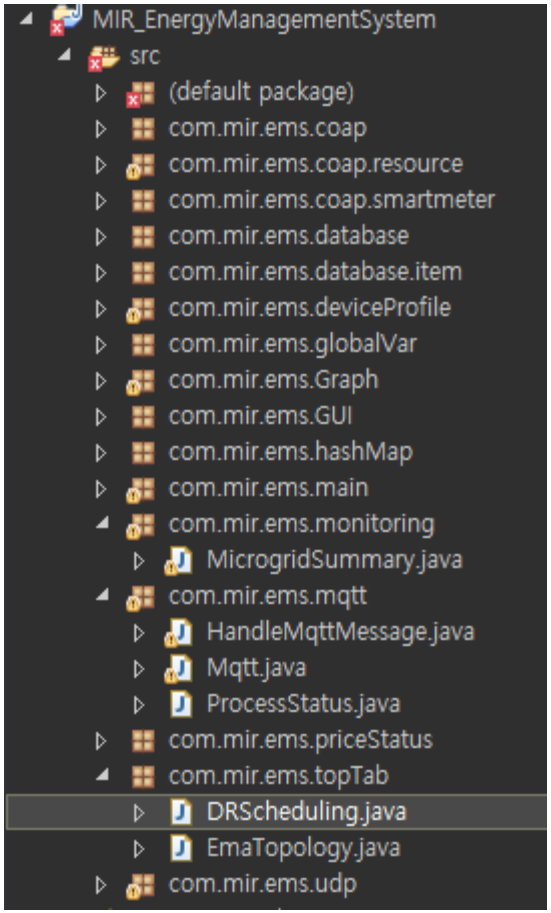
EMA_tab Class : 각각의 EMA에 대한 상세 정보 테이블

… [Line 23] 테이블 생성 및 테이블 업데이트 주기 설정

```
public void modify_EMA_table() {

                int ema_rows_num = EMA_tab_temp.ema_table_model.getRowCount();
                for (int i = ema_rows_num - 1; i >= 0; i--) {
                        EMA_tab_temp.ema_table_model.removeRow(i);
                }

                Iterator<String> keys = global.emaProtocolCoAP.keySet().iterator();

                while (keys.classlassasNext()) {
                        String key = keys.next();

                        // 업데이트 항목을 global ema관리 Map에서 참조하여 업데이트
                        EMA_tab_temp.ema_table_model
                                        .addRow(new Object[] { false, key,
global.emaProtocolCoAP.get(key).getProtocol(),
        global.emaProtocolCoAP.get(key).getqOs(), global.emaProtocolCoAP.get(key).getEmaCNT(),
        global.emaProtocolCoAP.get(key).getPower(), global.emaProtocolCoAP.get(key).getMaxValue(),
        global.emaProtocolCoAP.get(key).getMinValue(), global.emaProtocolCoAP.get(key).getMargin(),
        global.emaProtocolCoAP.get(key).getCustomerPriority() });
        }
```

4. EMS : Package Explanation
Control

Control



com.mir.ems.topTab

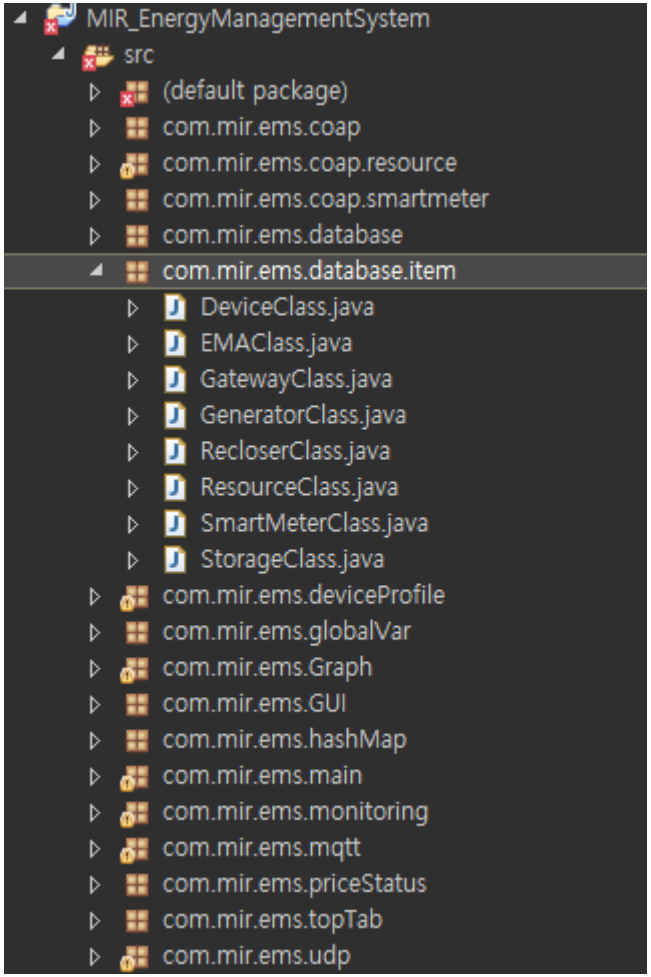
- DRScheduling:

Send DR Message to EMA

It is possible to send Push and Multicast Message here

4. EMS : Package Explanation
Profile

Device Profile



com.mir.ems.database.item

<u>Below all classes are Generic Class</u>

:To make object type

DeviceClass
EMAClass
SmartMeterClass
…

com.mir.ems.classlassashMap

:Give key value each devices for easy to handle and search

ESS_values
PV_values
Recloser_values
Resource_values
VTN_values

4. EMS : Package Explanation
Monitoring : global

gloal Class : 각각의 EMA에 대한 정보를 저장하는 Map

… [Line 23] 테이블 생성 및 테이블 업데이트 주기 설정

```
public static ConcurrentHashMap<String, EMA> emaProtocol = new ConcurrentHashMap<String, EMA>();

// EMA 정보 저장
public static void putEmaProtocol (String emaID, EMA) {
        try {
                Thread.sleep(20);
                emaProtocol.put(emaID, EMA);

        } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }

}

// Set EMA 정보
public static void setEmaProtocol (ConcurrentHashMap<String, EMA> emaProtocol) {
        global.emaProtocol = emaProtocol;
}
```

*** TIP: ConcurrentHashMap** 은 비 동기 방식의 **HashMap**으로 빠른 응답이 필요하거나 **Map**내에 **Sorting**이 필요하지 않은 경우 사용한다. 멀티 스레딩 방식에서 주로 사용하는 방식이다.

4. EMS : Package Explanation
Profile : EMA

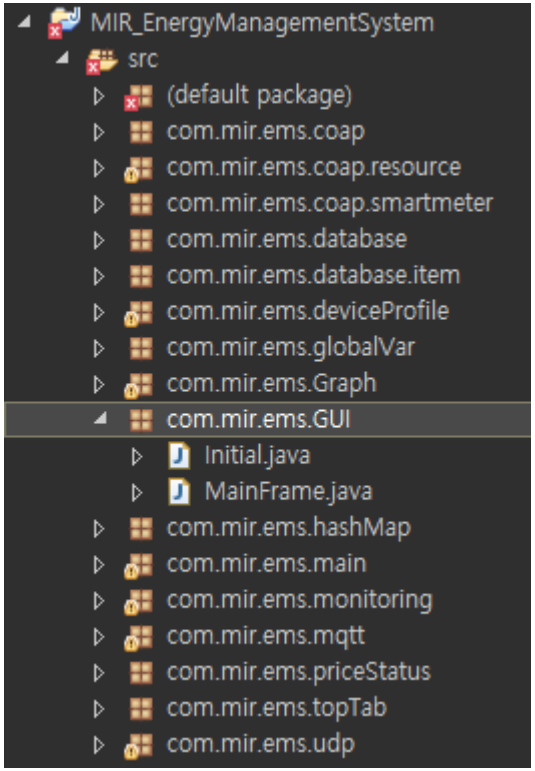
EMA Class : 각각의 EMA에 대한 상세 정보 테이블

… [Line 23] 테이블 생성 및 테이블 업데이트 주기 설정

```
public class EMA {

        private String emaID, qOs, type, registrationID, transportName, transportAddress, reportName, reportType, state,
                                        profileName, requestID, version;
        private String time, maxTime, minTime, connect, protocol;
        private int customerPriority, reportOnly, httpPullModel, xmlSignature, emaCNT, priority, dimming;
        private double margin, minValue, maxValue, avgValue, power, generate, storage;
        private boolean pullModel, tableChanged, realTimetableChanged;
        private JSONObject EMARegisteredInfo, EMARegisteredMgnInfo;

        // EMA 정보를 저장하는 구조체
        public Emap_Cema_Profile(String protocol, String emaID, String registrationID, String qos, String state, double power, int
dimming, double margin, double generate, double storage, double maxValue, double minValue, double avgValue, String maxTime, String
minTime, int priority, boolean pullModel, boolean tableChanged, boolean realTimetableChanged, String connect) {

                setProtocol(protocol);
                setRegistrationID(registrationID);
                setEmaID(emaID);
                setqOs(qos);
                setState(state);
                setPower(power);
                …..
        }
```

4. EMS : Package Explanation
GUI

GUI



com.mir.ems.GUI

Initial	**- First Page**
MainFrame	**- Main Frame that will be shown after first**

4. EMS : Package Explanation
GUI : Initial

Initial Class : IP 설정 및 프로토콜, 프로파일 설정 GUI

GUI 빌드 함수

```
public Initial() {
        // setting
        setTitle("MIREnergy Management System");
        setSize(326, 614);
        setResizable(false);
        setLocation(800, 450);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        // panel
        JPanel panel = new JPanel();
        placeLoginPanel(panel);

        // add
        getContentPane().add(panel);

        JPanel panel_1 = new JPanel();
        panel_1.setBorder(new EtchedBorder(EtchedBorder.LOWERED, null, null));
        panel_1.setBounds(12, 66, 298, 118);
        panel.add(panel_1);
        panel_1.setLayout(null);
        ..
}
```

4. EMS : Package Explanation
GUI : Initial

Initial Class : IP 설정 및 프로토콜 설정 GUI

```
GUI 빌드 함수
public Initial() {

                    //IP 설정
                    final Jlabel lblNewLabel = new Jlabel( " IP Address " );
                    lblNewLabel.setFont(new Font( " Arial " , Font.BOLD, 13));
                    lblNewLabel.setBounds(39, 75, 76, 25);
                    panel_2.add(lblNewLabel);
                    //PORT 설정
                    final JLabel lblNewLabel_1 = new JLabel("Port");
                    lblNewLabel_1.setFont(new Font("Arial", Font.BOLD, 13));
                    lblNewLabel_1.setBounds(39, 110, 76, 25);
                    panel_2.add(lblNewLabel_1);

                    //Protocol 설정
                    final JComboBox<String> comboBox = new JComboBox<String>();
                    …
                    comboBox.addItem("MQTT");
                    comboBox.addItem("CoAP");
                    comboBox.addItem("UDP");
                    comboBox.addItem("BOTH");                    ..
}
```

4. EMS : Package Explanation
GUI : MainFrame

MainFrame Class : 가격 정보, 등록된 EMA 정보 CFG 파일을 가져오는 함수

```
public MainFrame() {

		// 가격정보를 가져오는 함수
		rdbtnmntmNewRadioItem_1.addActionListener(new ActionListener() {

		@Override
		public void actionPerformed(ActionEvent arg0) {
				// TODO Auto-generated method stub
				if (rdbtnmntmNewRadioItem_1.isSelected()) {
						rdbtnmntmNewRadioItem.setSelected(false);

						JFileChooser jfc = new
JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory());
						jfc.setDialogTitle("Select an configuration file");
						jfc.setAcceptAllFileFilterUsed(false);
						FileNameExtensionFilter filter = new FileNameExtensionFilter(".classfg files", "cfg",
"CFG");
						jfc.addChoosableFileFilter(filter);

						int returnValue = jfc.showOpenDialog(null);

						if (returnValue == JFileChooser.APPROVE_OPTION)
						new RealTimePriceFileReader(jfc.getSelectedFile().getPath());

								}
						}
				});
				..
}
```

4. EMS : Package Explanation
GUI : MainFrame

Initial Class : IP 설정 및 프로토콜, 프로파일 설정 GUI

GUI 빌드 함수

```
public Initial() {
        // setting
        setTitle("MIREnergy Management System");
        setSize(326, 614);
        setResizable(false);
        setLocation(800, 450);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        // panel
        JPanel panel = new JPanel();
        placeLoginPanel(panel);

        // add
        getContentPane().add(panel);

        JPanel panel_1 = new JPanel();
        panel_1.setBorder(new EtchedBorder(EtchedBorder.LOWERED, null, null));
        panel_1.setBounds(12, 66, 298, 118);
        panel.add(panel_1);
        panel_1.setLayout(null);
        ..
}
```

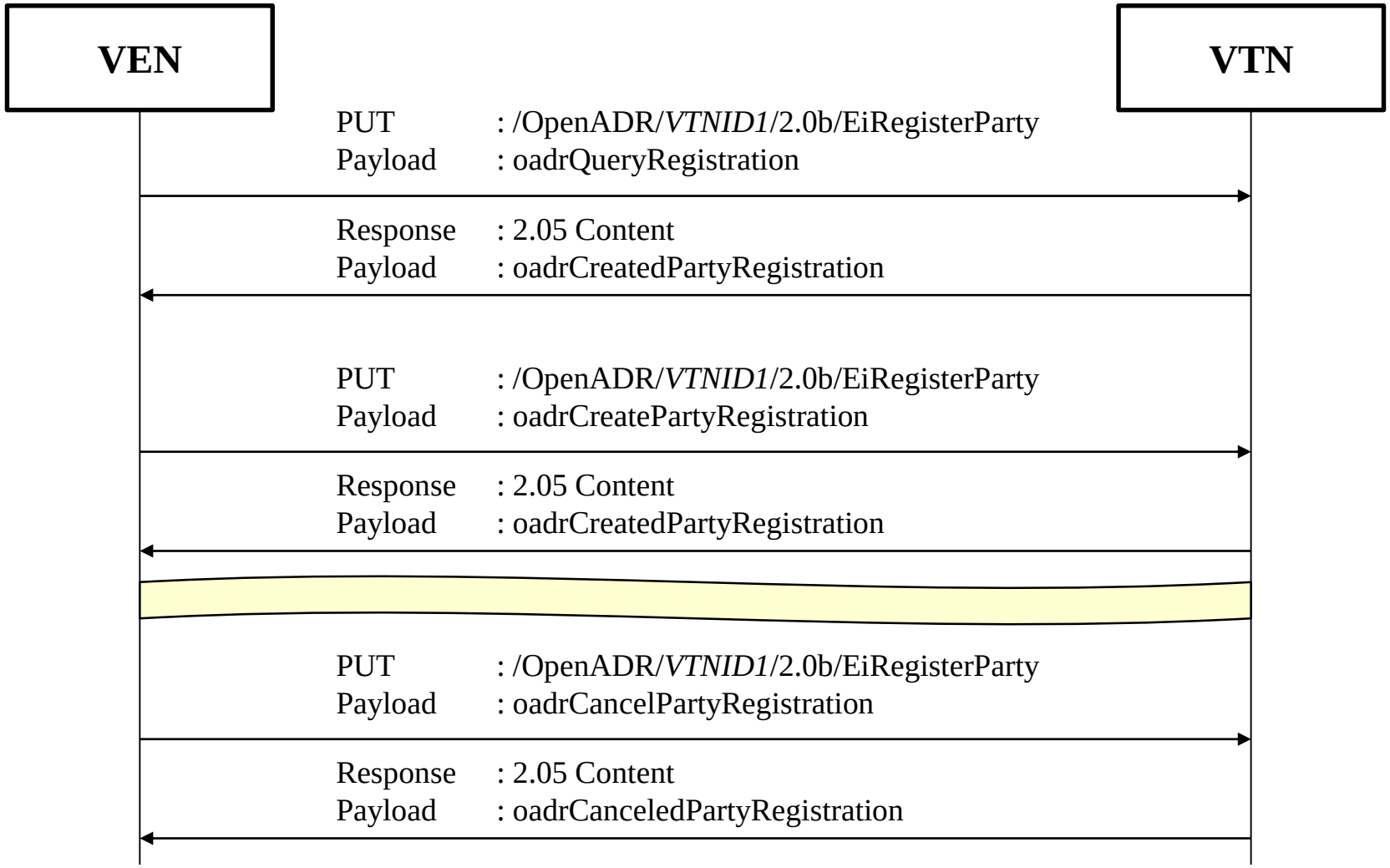
OpenADR 2.0b

(1) EiRegistrationParty

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

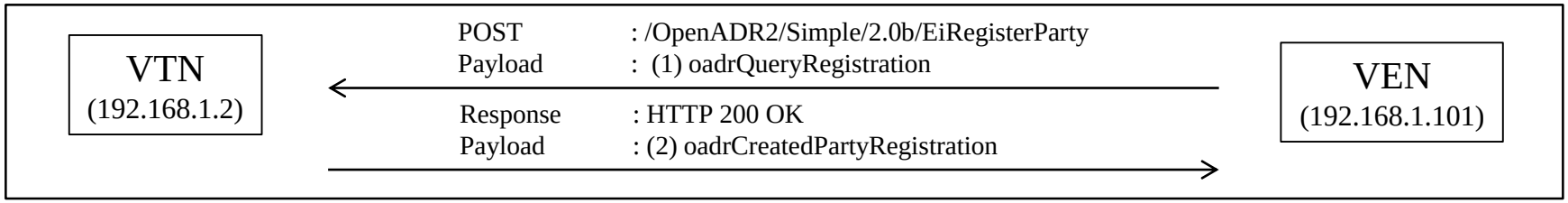
2. Profile : OpenADR 2.0b

2.1 Services : EiRegistrationParty (CoAP / JSON)



2. Profile : OpenADR 2.0b

2.1 Services : EiRegistrationParty (CoAP / JSON)



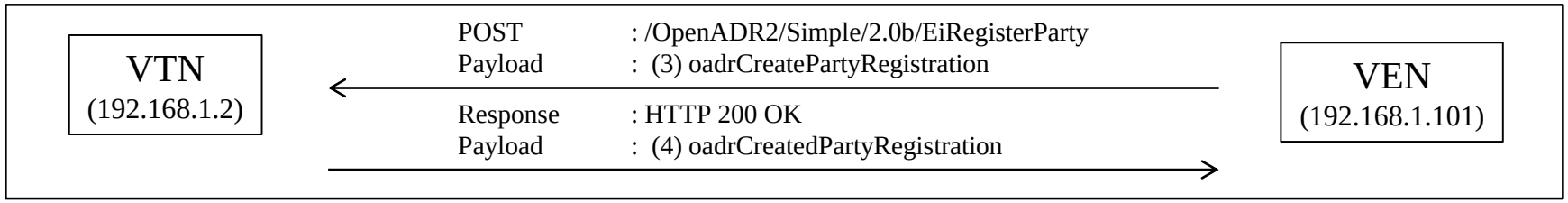
(1) oadrQueryRegistration

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

(2) oadrCreatedPartyRegistration

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

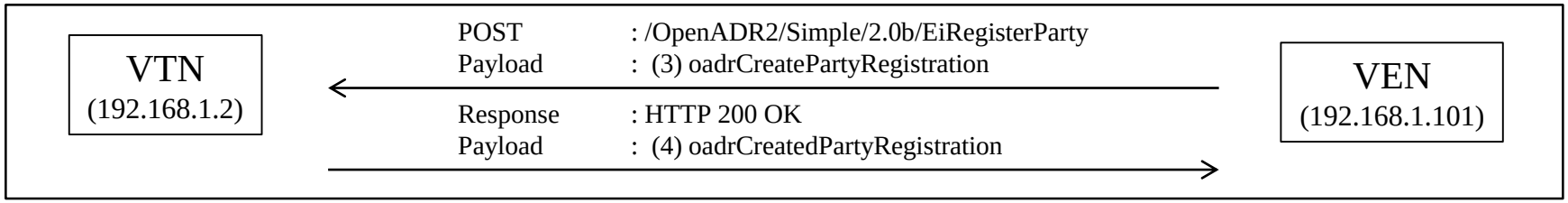
2. Profile : OpenADR 2.0b
2.1 Services : EiRegistrationParty (CoAP / JSON)



(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

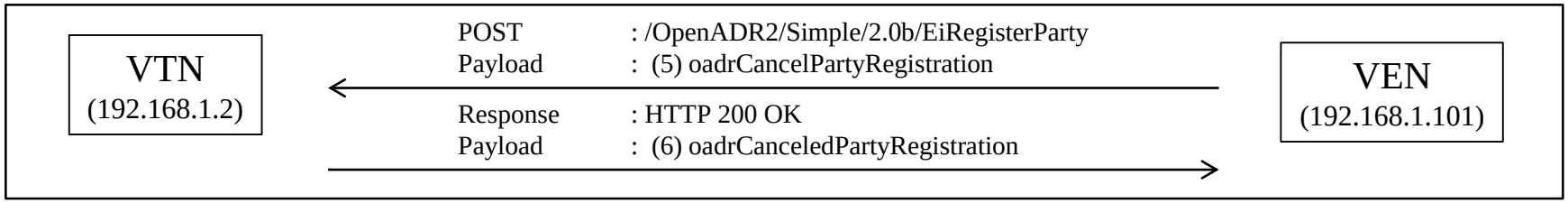
2. Profile : OpenADR 2.0b
2.1 Services : EiRegistrationParty (CoAP / JSON)



(4) oadrCreatedPartyRegistration

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

2. Profile : OpenADR 2.0b
2.1 Services : EiRegistrationParty (CoAP / JSON)



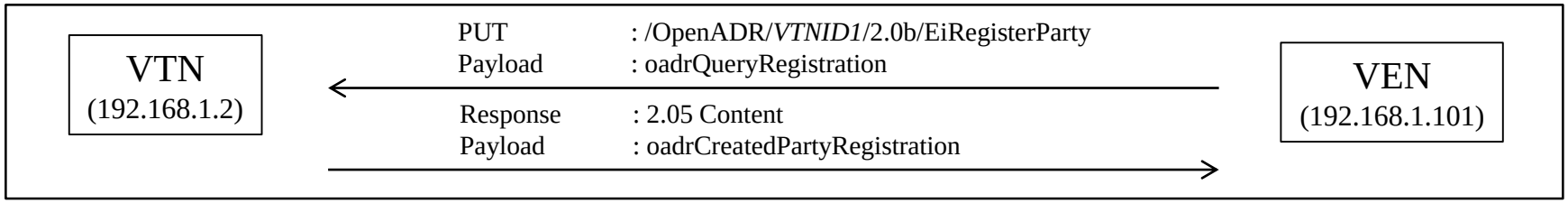
(5) oadrCancelRegistration

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 (CoAP / JSON)



(1) QueryRegistration
(2) CreatedPartyRegistration

CoAP	192.168.1.101	192.168.1.127	CON, MID:17505, PUT, /QueryRegistration
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
}

```
QueryRegistratio n{ "Service": " QueryRegistratio n", "GW": "gw\/1 ",
"VENID": "VEN _MIR1", "Request ID": 1, "Version ": 2 }
```

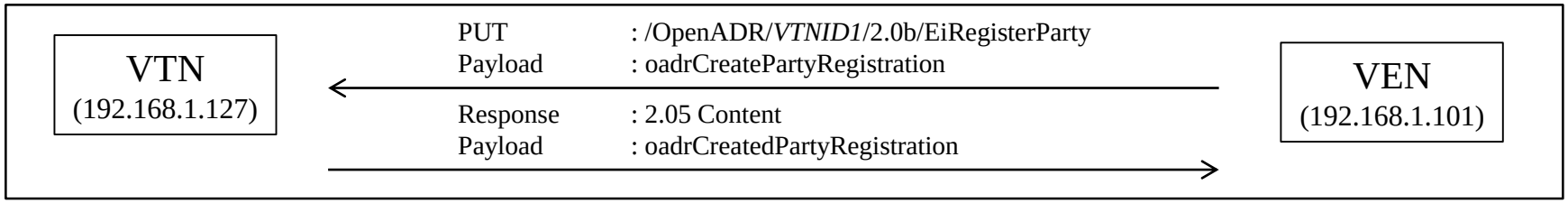
oadrCreatedPartyRegistration JSON{
"venID": String,
"vtnID": String,
"requestID": String,
"duration" : Integer,
"responseCode" : Integer,
"responseDescription": String,
"registrationID" : String,
"**oadrProfile**" : Array,
"service" : String
}

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
}

```
CreatePartyRegis trationÿ{ "Servi ce": "CreatePart yRegistrationReq uest",
"GW": "gw \/1", "RequestID ": 1, "Version": 2, "TansportNam e": "CoAP", "Rep
ortOnly": 0, "oa drVenName": "VEN _MIR1", "LastPol lPushGet": 3, "c
adrProfileName": 2, "oadrXmlSign ature ": 0, "oad rTransportAddres s ":
"192.168.1. 2" }
```

oadrCreatedPartyRegistration JSON{
"venID": String,
"vtnID": String,
"requestID": String,
"duration" : String,
"responseCode" : Integer,
"responseDescription": String,
"registrationID": String,
"**oadrProfile**" : Array,
"service" : String
}

oadrTransports Array{
"oadrTransportName": String,
}

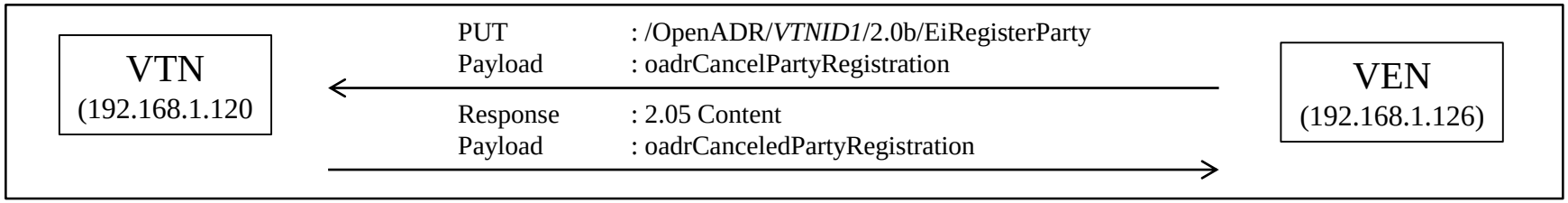
oadrProfile Array{
"**oadrTransports**": Array,
"oadrProfileName": String
}

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

implementation yet

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

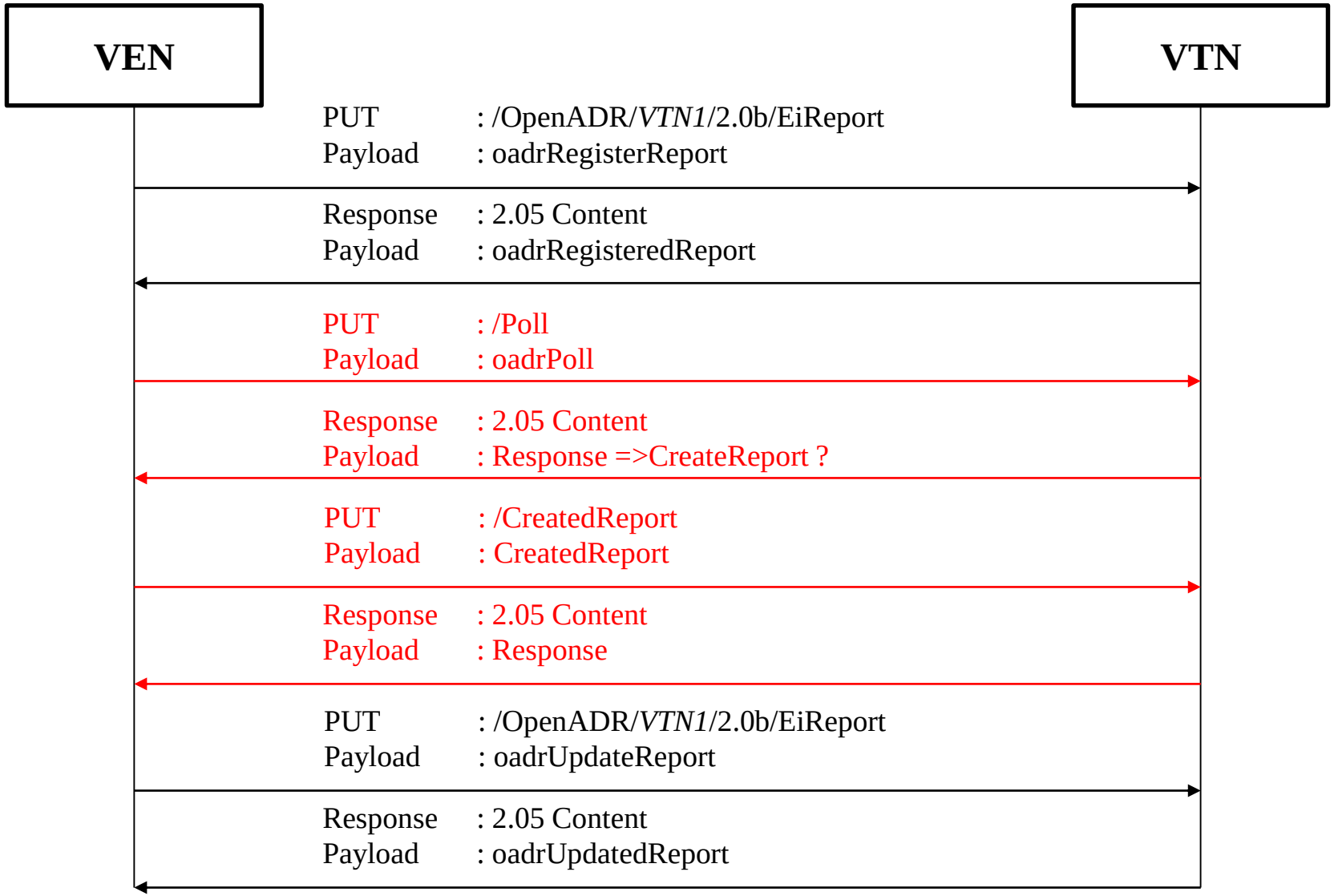
(2)EiReport

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

빨간색 생략

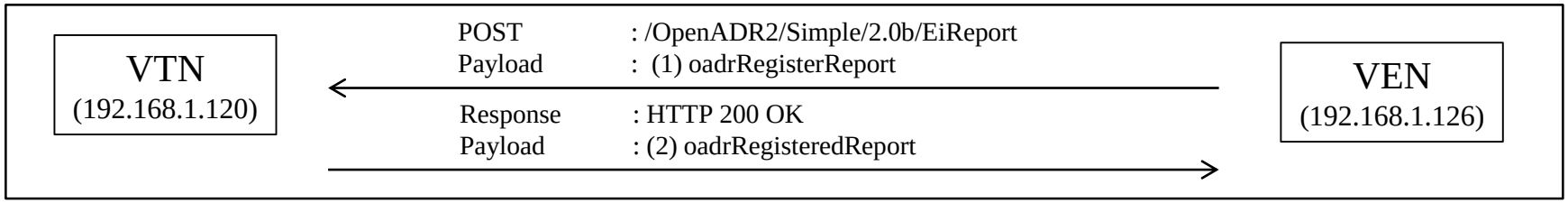
2. Profile : OpenADR 2.0b

2.2 Services : EiReport (CoAP / JSON)



2. Profile : OpenADR 2.0b

2.2 Services : EiReport (CoAP / JSON)

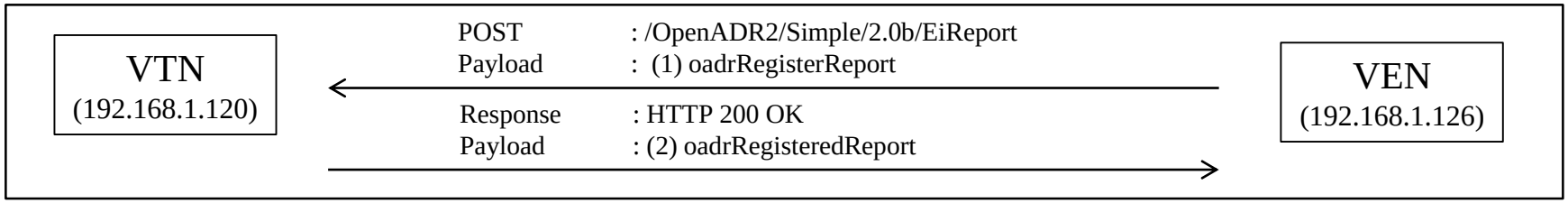


(1) oadrRegisterReport

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
	reportDescription	rID		
		resourceID		resource identifier
		reportType		report type
		itemUnits		unit of item that report
		siScaleCode		
		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 (CoAP / JSON)

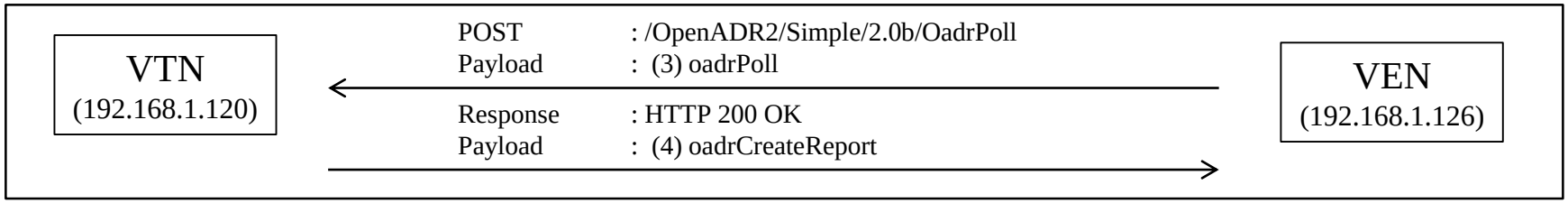


(2) oadrRegisteredReport

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 (CoAP / JSON)



(3) oadrPoll

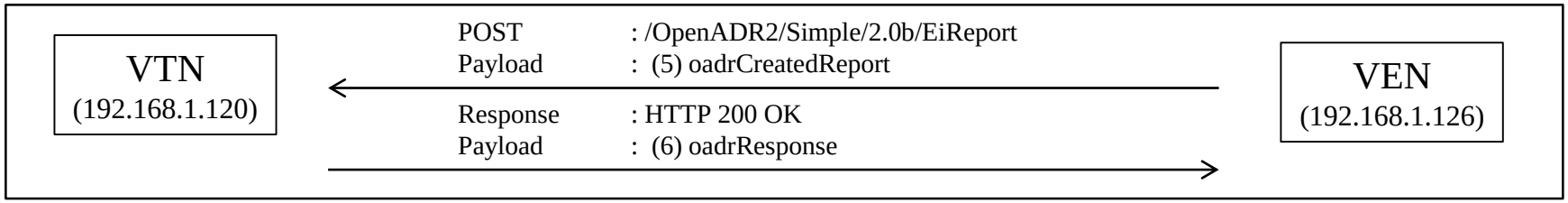
Key name	Comments
venID	requested VEN ID
service	message type

(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 (CoAP / JSON)



(5) oadrCreatedReport

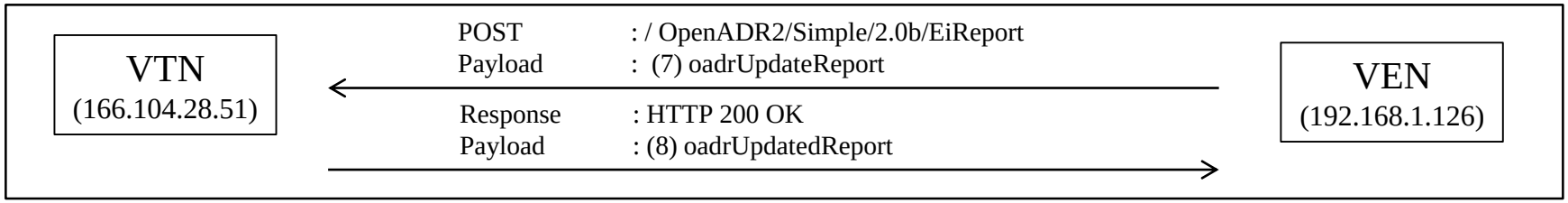
Key name	Comments
venID	requested VEN ID
requestID	request identifier
responseCode	response code
reportRequestID	report request identifier
service	message type

(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 (CoAP / JSON)

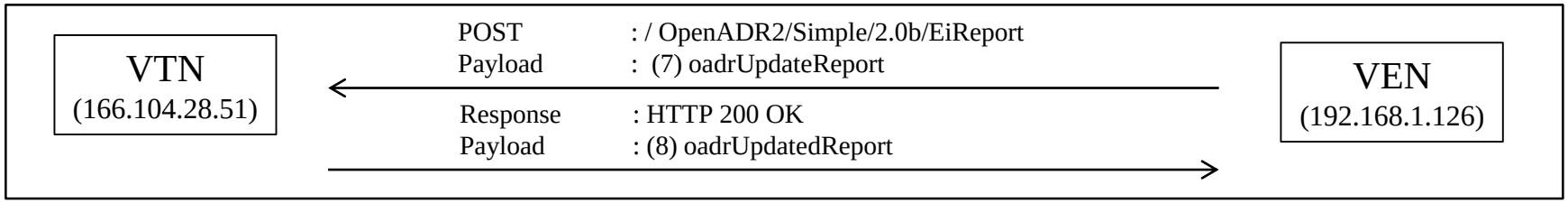


(7) oadrUpdateReport

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
	reportDescription	rID		
		resourceID		resource identifier
		reportType		report type
		itemUnits		unit of item that report
		siScaleCode		
		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 (CoAP / JSON)

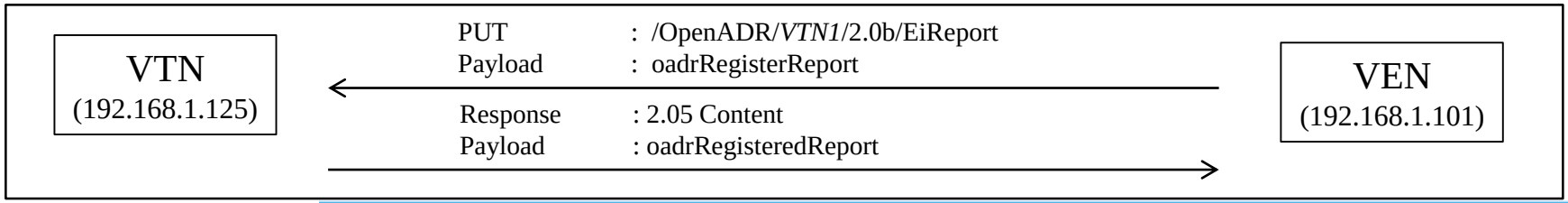


(8) oadrUpdatedReport

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 (CoAP / JSON)



(1) RegisterReport
(2) RegisteredReport

CoAP	192.168.1.101	192.168.1.125	CON, MID:46948, PUT, /RegisterReport
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
}

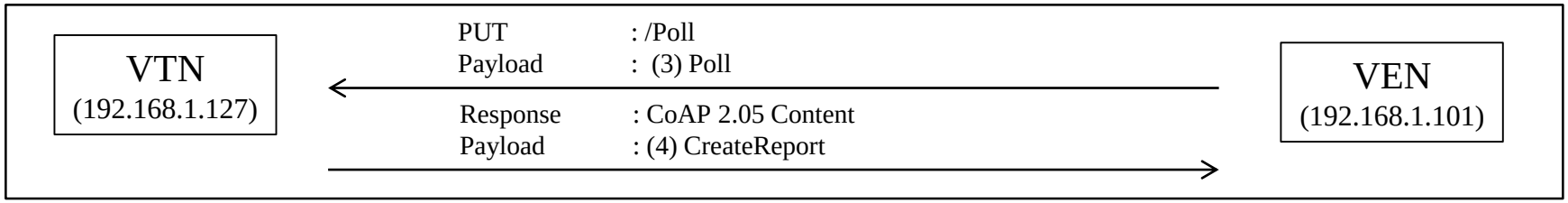
oadrReportDescription Array{
"rID" : String
"resourceID" : String,
"reportType" : String,
"itemUnits" : String,
"siScaleCode" : String,
"marketContxt" : String,
"oadrMinPeriod" : String,
"oadrMaxPeriod" : String,
"oadrOnChange" : Boolean,
"itemDescription" : String.
"powerAttributes" : Array
}

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

2. Profile : OpenADR 2.0b

Not implement yet

2.2 Services : EiReport (CoAP / JSON)



(3) Poll
(4) CreateReport

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

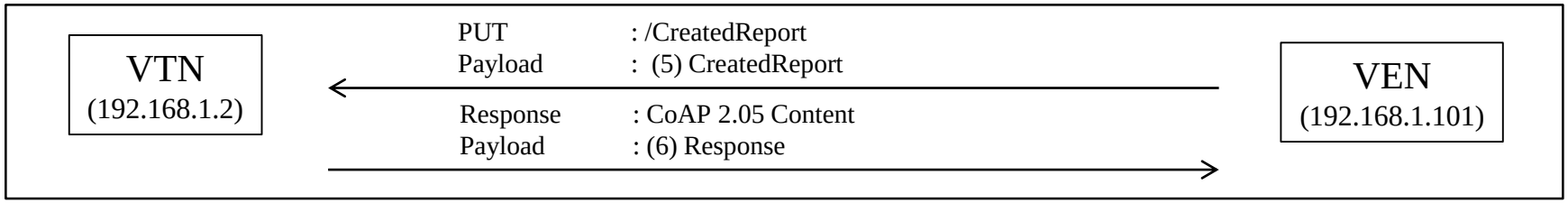
```
Poll{ "Service": "Poll", "GW": " gw\/1", "VENID": "VEN_MIR1", "Re questID":
1, "Ve rsion": 2 }
```

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

2. Profile : OpenADR 2.0b

Not implement yet

2.2 Services : EiReport (CoAP / JSON)



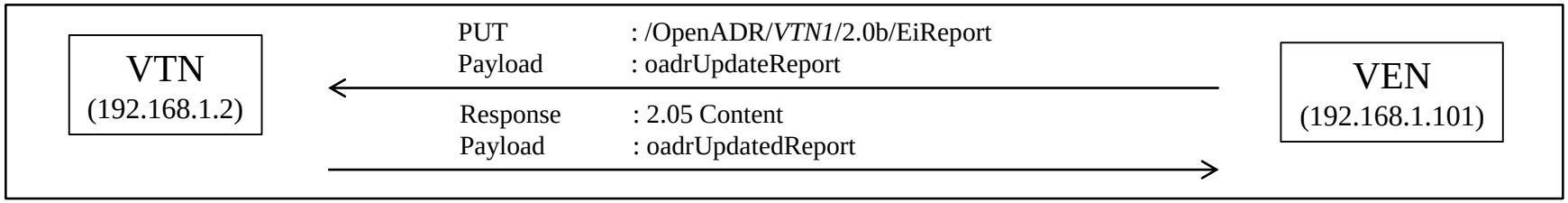
(5) CreatedReport
(6) Response

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/pl

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

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

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
}

oadrUpdatedReport JSON{
"reponseCode" : Integer,
"responseDescription" : String,
"requestID" : String,
"venID" : String,
"service" : String
}

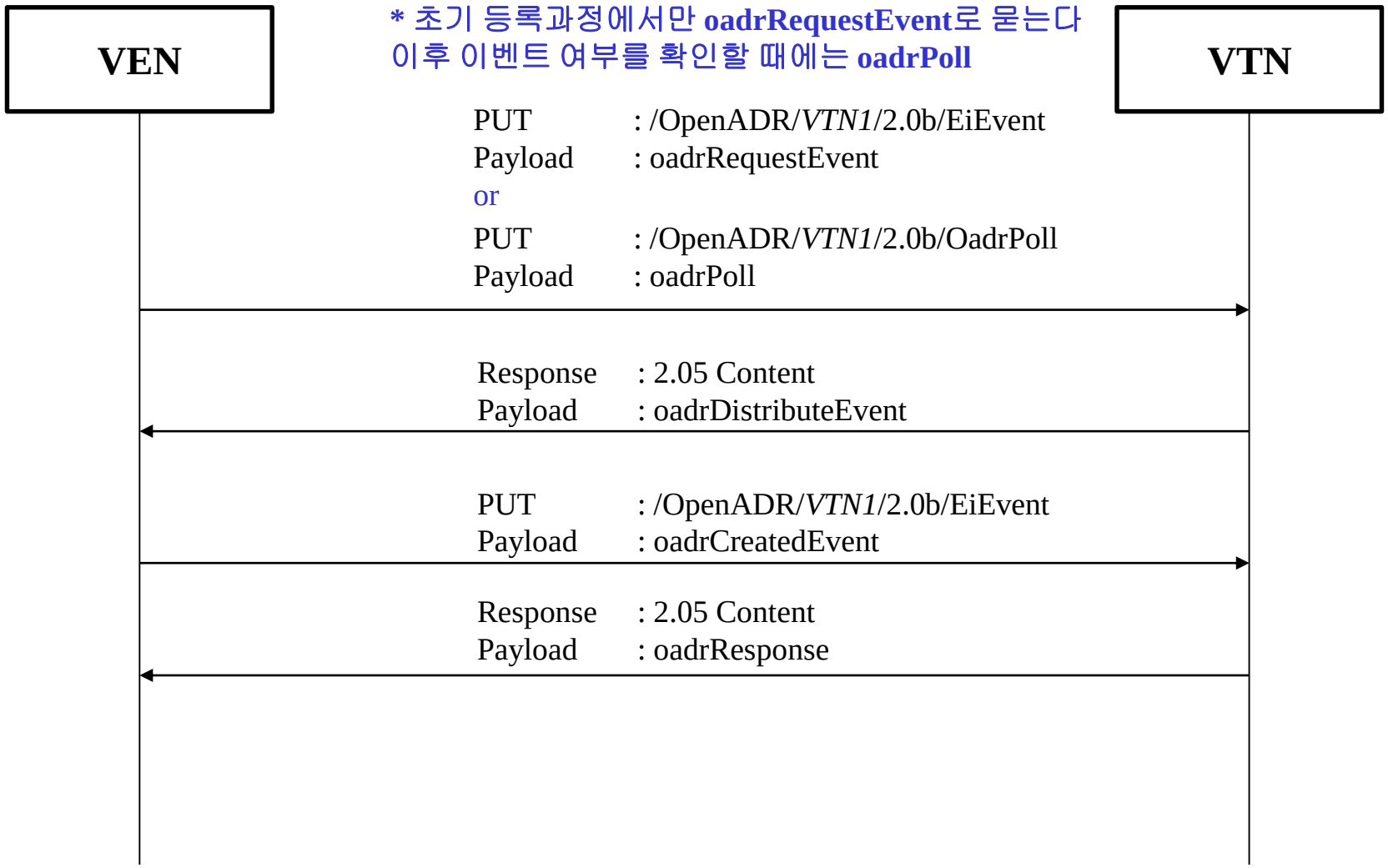
OpenADR 2.0b

(3) EiEvent

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

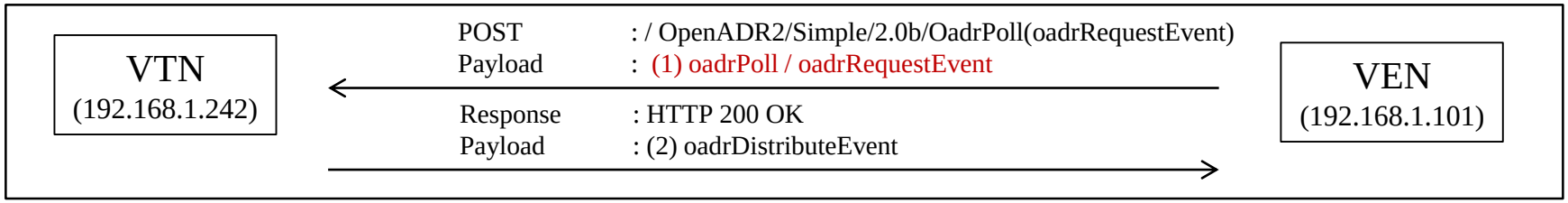
2. Profile : OpenADR 2.0b

2.3 Services : EiEvent (CoAP / JSON)



2. Profile : OpenADR 2.0b

2.3 Services : EiEvent (CoAP / JSON)



(1) oadrPoll

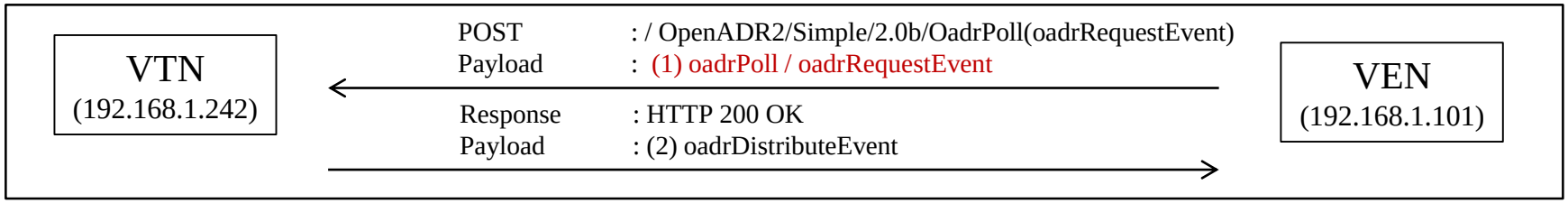
Key name	Comments
venID	requested VEN ID
service	message type

oadrRequestEvent

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

2. Profile : OpenADR 2.0b

2.3 Services : EiEvent (CoAP / JSON)

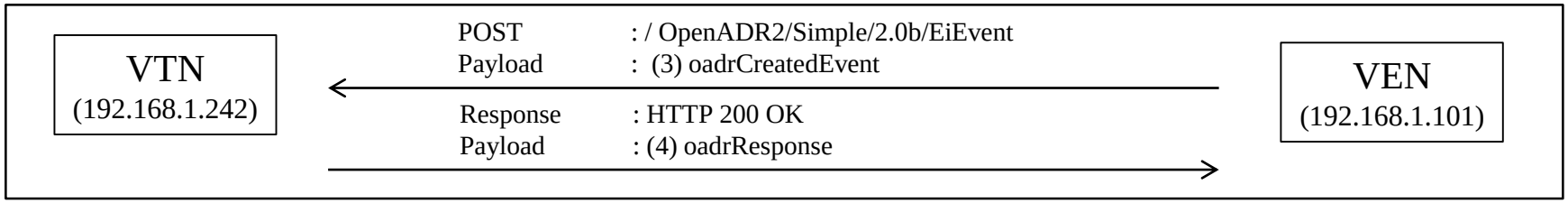


(2) oadrDistributeEvent

Key name				Comments
venID				requested VEN ID
response	requestID			request identifier
	responseCode			response code
	responseDescription			description of response code
event	eventID			event identifier
	eventSiganls	intervals	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
	modificationNumber			modification Number(count)
	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 (CoAP / JSON)



(3) oadrCreatedEvent

Key name	Comments
vtnID	responsed 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 paticipate event or not
service	message type

(4) oadrResponse

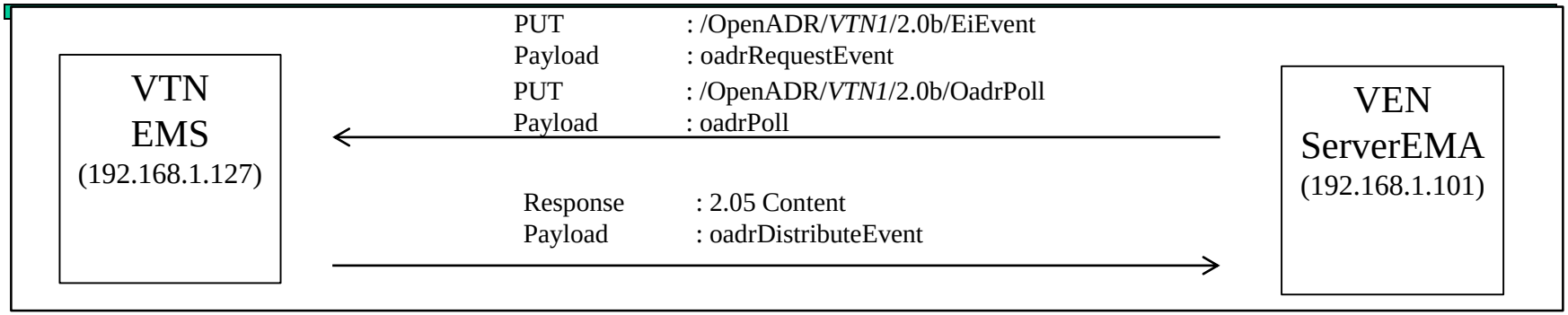
Key name	Comments
venID	requested VEN ID
requestID	request identifier
responseCode	response code
responseDescription	description of response code
service	message type

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
(2) DistributeEvent

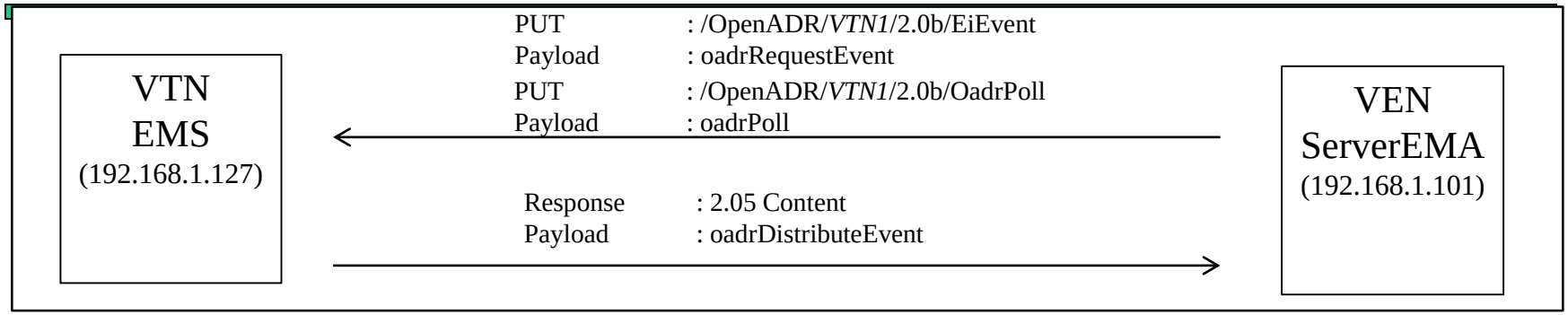
CoAP	192.168.1.101	192.168.1.127	CON, MID:35224, PUT, /Poll
CoAP	192.168.1.127	192.168.1.101	ACK, MID:35224, 2.05 Content (text/plain)

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
(2) DistributeEvent

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


```
{
  "Responsed escription":"MIR ",
  "EndTime":"194 1",
  "RequestID":1,
  "StartTime":"19 40",
  "Service":"DistributeEvent",
  "EndYMD":"201702 09",
  "Response":1,
  "Value":1,
  "OptT ype":"optIn",
  "St artYMD":"2017020 9",
  "EventID":1,
  " ModificationNumb er":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
}

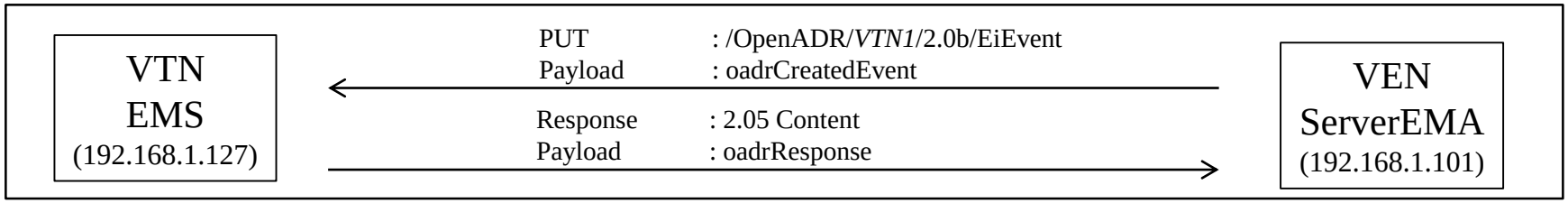
eventSignals Array{
"intervals" : Array,
"signalName" : String,
"signalType" : String,
"signalID" : String,
"currentValue" : Double
}

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, "Req uestID": 1, "Value": 1 }
```

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

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

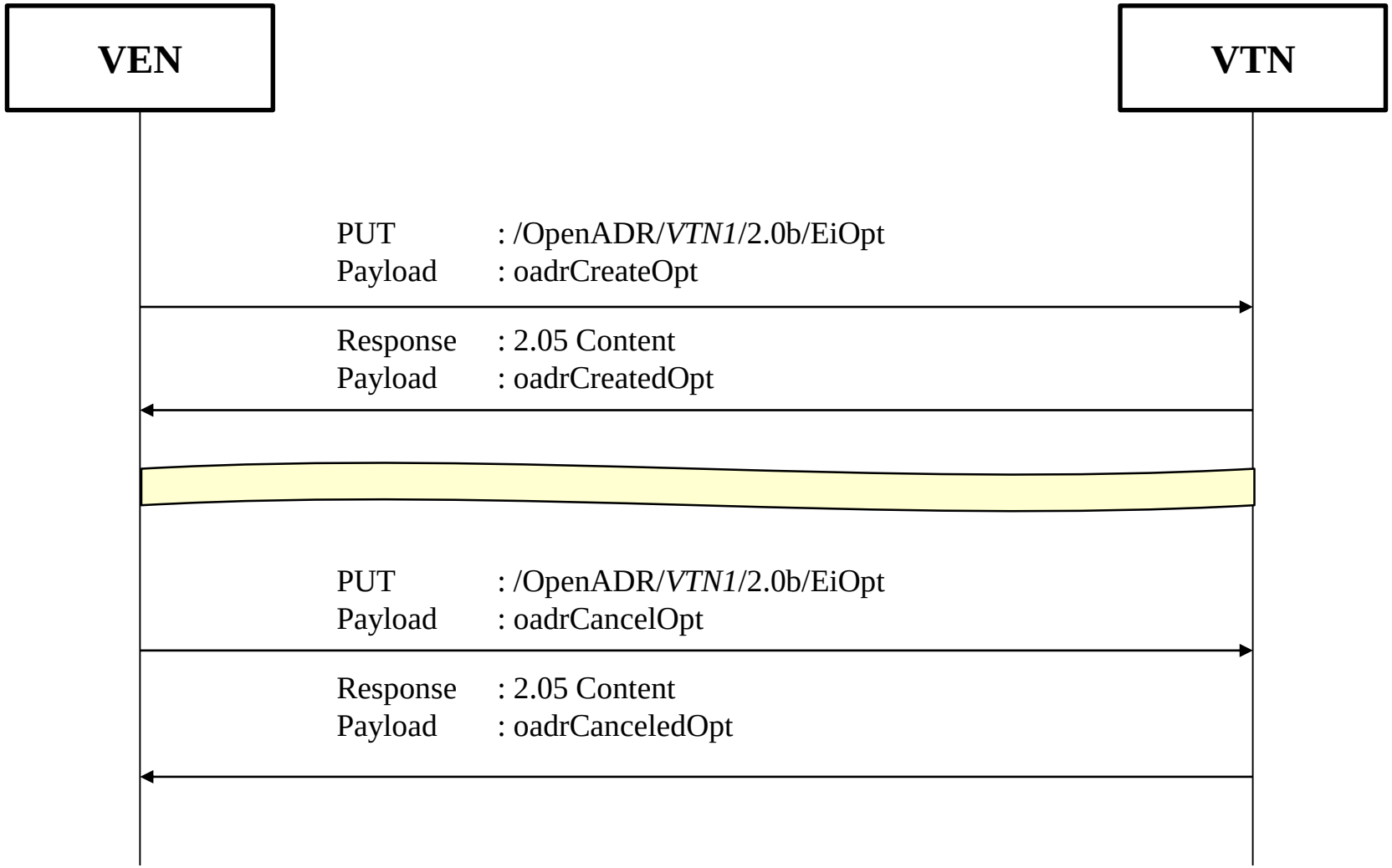
OpenADR 2.0b

(4) EiOpt

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

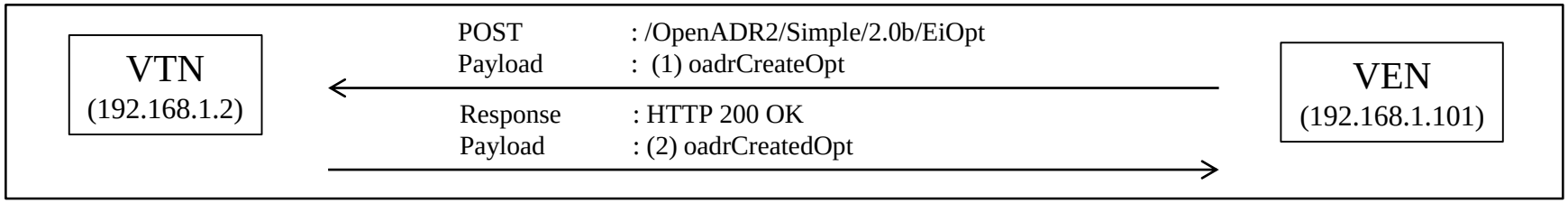
2. Profile : OpenADR 2.0b

2.3 Services : EiEvent (CoAP / JSON)



2. Profile : OpenADR 2.0b

2.4 Services : EiOpt (CoAP / JSON)



(1) oadrCreateOpt

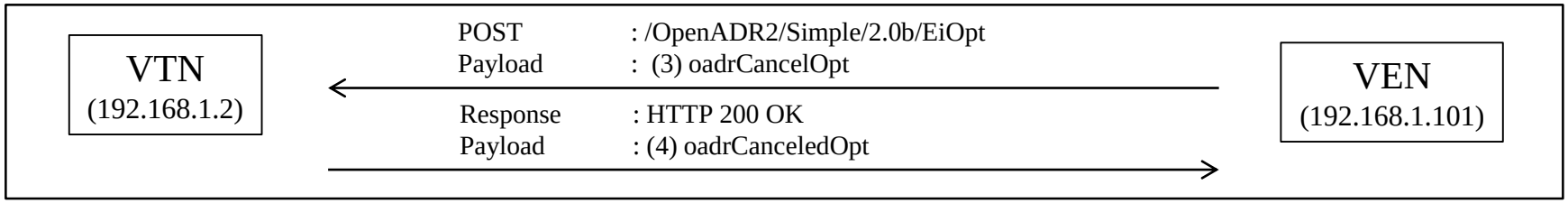
Key name		Comments
venID		requested VEN ID
optID		opt identifier
optType		type of opt
optReason		opt reason(e.g. emergency)
marketContext		refer market address
available	dtstart	opt start time
	duration	opt duration
createdDateTime		created time of this message
requestID		request identifier
service	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



(3) oadrCancelOpt

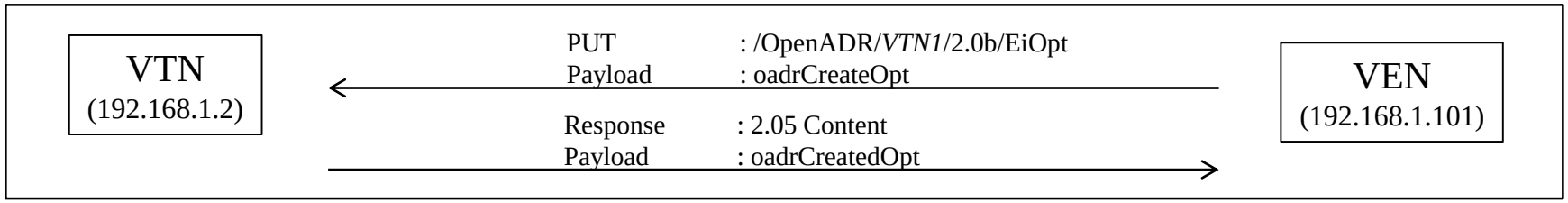
Key name	Comments
requestID	request identifier
optID	opt identifier
venID	requested VEN ID
service	message type

(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 (CoAP / JSON)



(1) CreateOpt
(2) CreatedOpt

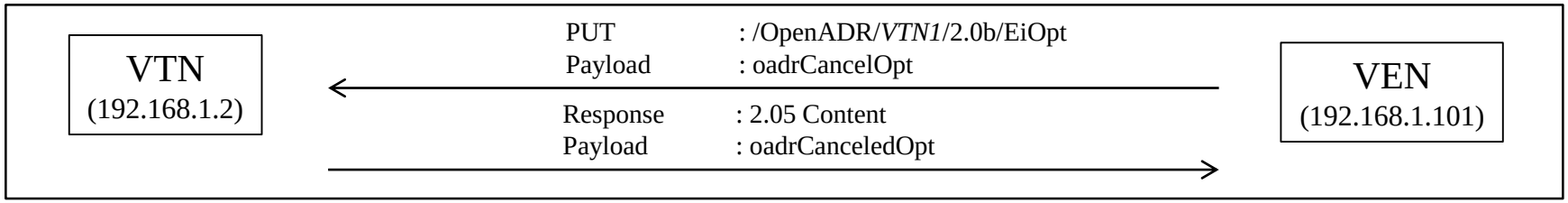
oadrCreateOpt JSON{
"optID": String,
"optType" : String,
"optReason" : String,
"venID" : String,
"marketContext" : String,
"available" : Array,
"createdDateTime" : Date,
"requestID" : String,
"service" : String
}

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

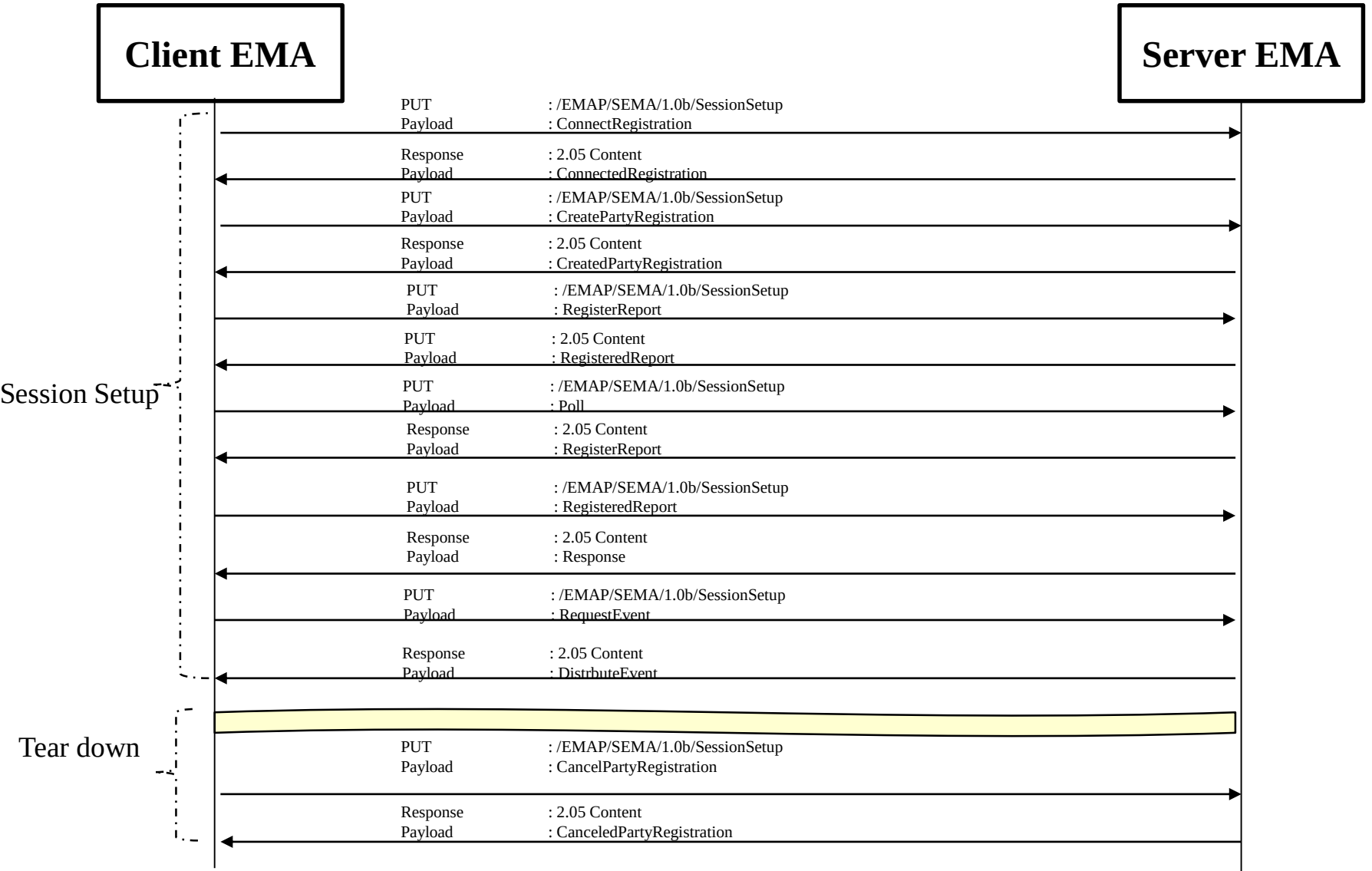
oadrCancelOpt JSON{
"requestID" : String,
"optID" : String,
"venID" : String,
"service" : String
}

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

EMAP
(1) Session Setup (Registration)

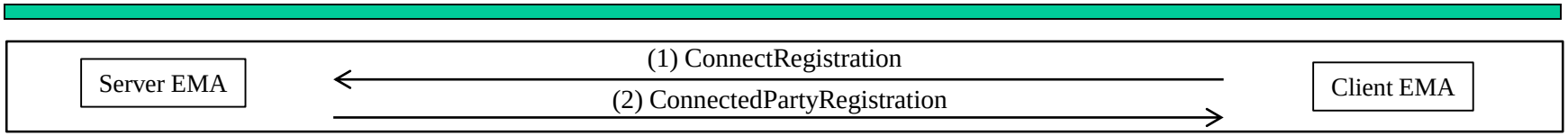
- **CoAP / JSON**
- **MQTT / JSON**

2.2 EMAP(CoAP/JSON)
Service : Session Setup



파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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



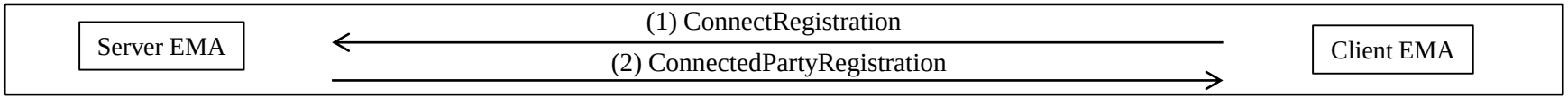
(1) ConnectRegistration

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:eiResponse	ei:responseCode	
responseDescription			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	profileName		type of profile
	transports	transportName	type of transport protocol
version			EMAP protocol version
service			type of service
time			service creation time

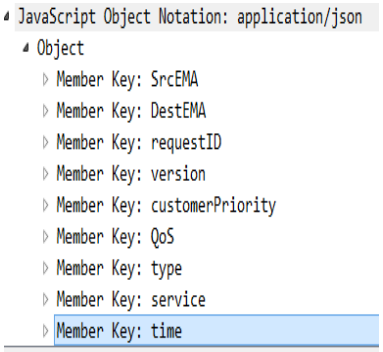
파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

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)

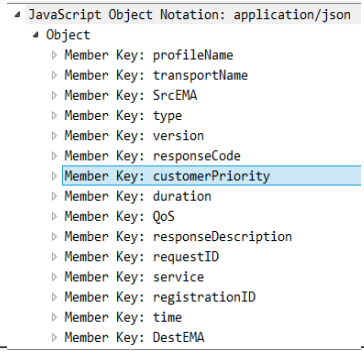
ConnectRegistration Object{
"SrcEMA" : String,
"DestEMA" : String,
"requestID": String,
"service": String,
"version" : String,
"time" : Date,
~~"type": String => RegisteredReport~~ ~~로 변경~~
~~"QoS" : String => RegisterReport로 변경~~
}



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 => RegisteredReport 로 변경~~
~~"QoS" : String => RegisterReport 로 변경,~~
"registrationID" : String
}

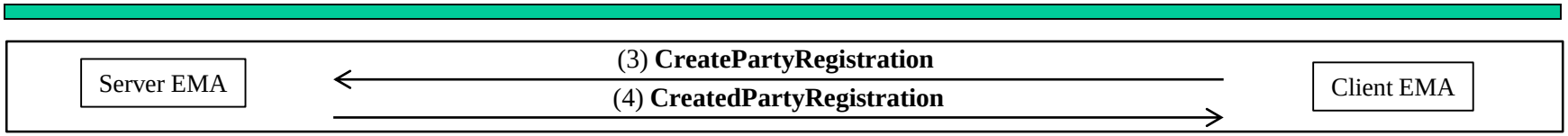
oadrProfile Array{
"oadrTransports": Array,
"oadrProfileName": String
}

oadrTransports Array{
"oadrTransportName": String,
}



파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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:requsetID	
profileName	oadr:oadrProfileName	
transportName	oadr:oadrTranportName	
reportOnly	oadr:oadrReportOnly	
xmlSignature	oadr:oadrXMLSignature	
httpPullModel	oadr:oadrHttpPullModel	
service	(Tag 이름으로 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:eiResponse	Ei:responseCode	
responseDescription			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:randomizeDuration		
service		(Tag 이름으로 존재)		
time				RandomizableEvent:creation Time

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

(3) CreatePartyRegistration

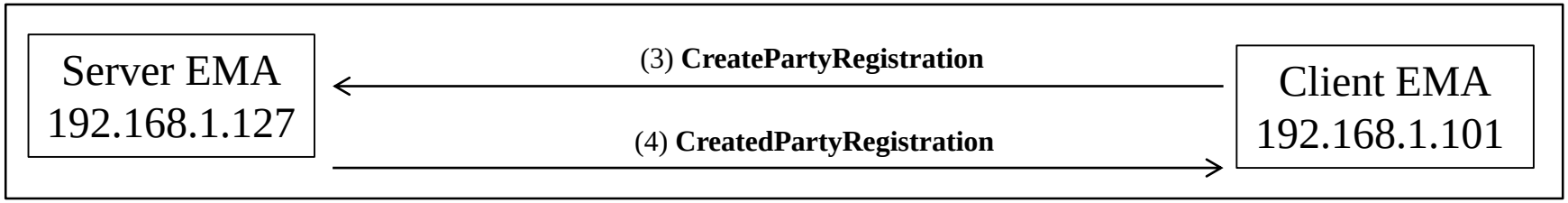
Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
profileName	profile name used by client EMA
transportName	transport name used by client EMA
reportOnly	EMA type (report only or full functional)
xmlSignature	xml 사용여부 true/false
httpPullMode	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		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 : OpenADR확장된 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,
}

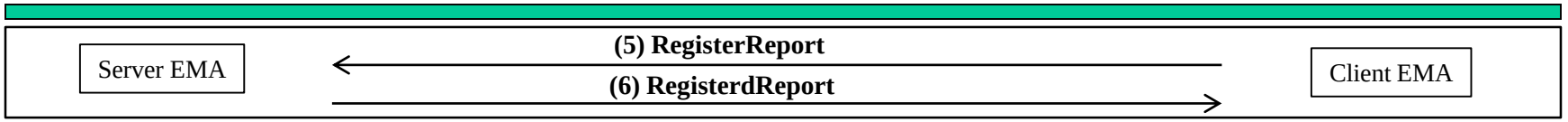
oadrProfile Array{
"**oadrTransports**": Array,
"oadrProfileName": String
}

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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

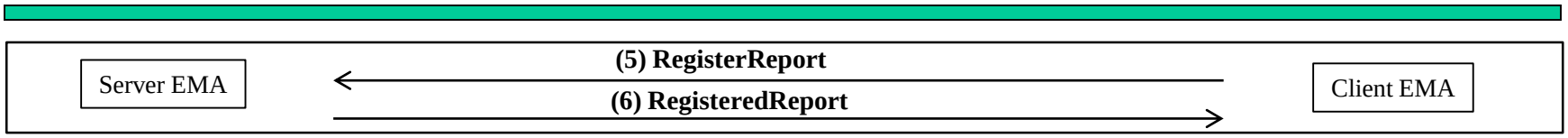


(5) RegisterReport

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			
type(Implicit, Explicit)				TariffProfile:serviceCategoryKind:ServiceKind	
report	duration	oadrReport	duration		
	reportDescription		oadrReportDescription		
	reportRequestID		reportRequestID		
	reportSpecifierID		reportSpecifierID		
	reportName		reportName		
	createdDateTime		createdDateTime		
report:reportDescription	rID	oadrReport:oadrReportDescription	rID		
	resourceID		resourceID		
	deviceType				EndDeviceControlType:type
	reportType		reportType		
	itemUnits		itemUnits		
	siScaleCode		siScaleCode		
	marketContext		marketContext		
	MinPeriod		oadrMinPeriod		
	MaxPeriod		oadrMaxPeriod		
	OnChange		oadrOnChange		
	itemDescription		itemDescription		
	powerAttributes		powerAttributes		
	qos			EndDeviceControl:loadShiftForward	
	state			DeviceStatus:opState	
	power				Readings: value
	dimming			Subscription:Level	
	margin			IdentifiedObject:DemandResponseProgram:availabilityUpdatePowerChnageThreshold	
	generate				SolarEventProfile:SolarInvetterStatus:value
	storage				BatteryEventProfile:BatteryStatus:value

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

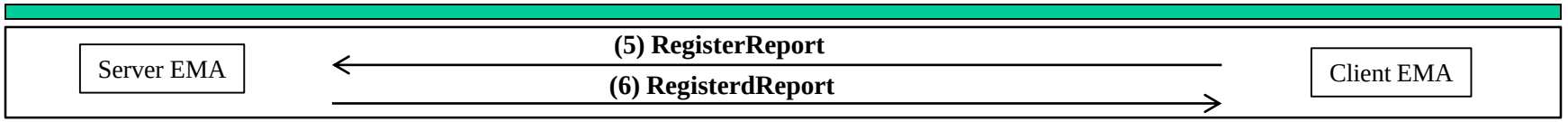
2.2 EMAP(MQTT, CoAP/JSON)
Service : Session Setup



(6) RegisteredReport

Key Name	Reference		
	OpenADR 2.0b		SEP 2.0(IEC 61968)
SrcEMA		ei:venID	
DestEMA		ei:vtnID	
responseCode	ei:eiResponse	ei:responseCode	
responseDescription		ei:responseDescription	
requestID		pyld:requestID	
service	(tag이름으로 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
	reportDescription	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
		powerAttributes	hertz	pulse frequency of power
			voltage	voltage of power
			ac	Is this AC power? (True or False)
service				type of service
time				service creation time
type				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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

2. Smart Home Energy Framework :
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup

(5) RegisterReport
(6) RegisteredReport

192.168.1.101	192.168.1.127	CoAP	CON, MID:36645, PUT, /RegisterReport (application/json)
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,
~~"EMAregisteredDRinformation" : Object =>변경~~
~~"EMAregisteredMgnInformation" : Object => 변경,~~
- **"report" : Array,**
"time": Date,
"service" : String,
"type": String (Explicit, Implict인지 구분)
}

JavaScript Object Notation: application/json
- Object
  - Member Key: SrcEMA
  - Member Key: DestEMA
  - Member Key: requestID
  - Member Key: reportName
  - Member Key: reportType
  - Member Key: EMAregisteredDRInformation
  - Member Key: EMAregisteredMgnInformation
  - 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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

2. Smart Home Energy Framework : 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup

(5) RegisterReport
(6) RegisteredReport

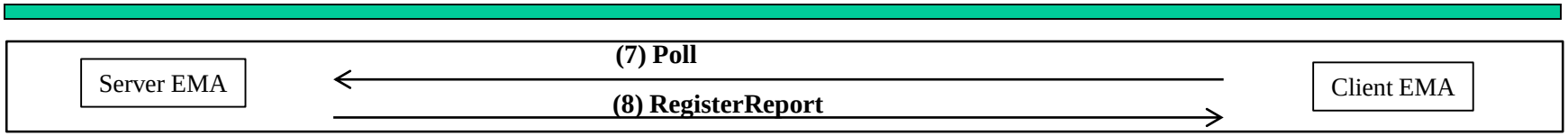
192.168.1.101	192.168.1.127	CoAP	CON, MID:36645, PUT, /RegisterReport (application/json)
192.168.1.127	192.168.1.101	CoAP	ACK, MID:36645, 2.05 Content (application/json)

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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



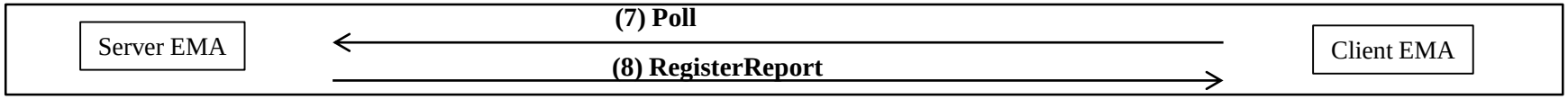
(7) Poll

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

(8) RegisterReport

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:creationTime

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



(7) Poll

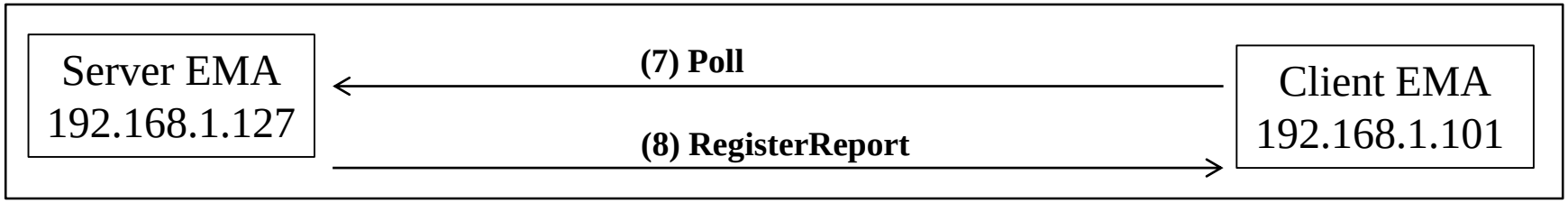
Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
service	message type
time	service creation time

(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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

2. Smart Home Energy Framework : 2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(7) Poll
(8) RegisterReport

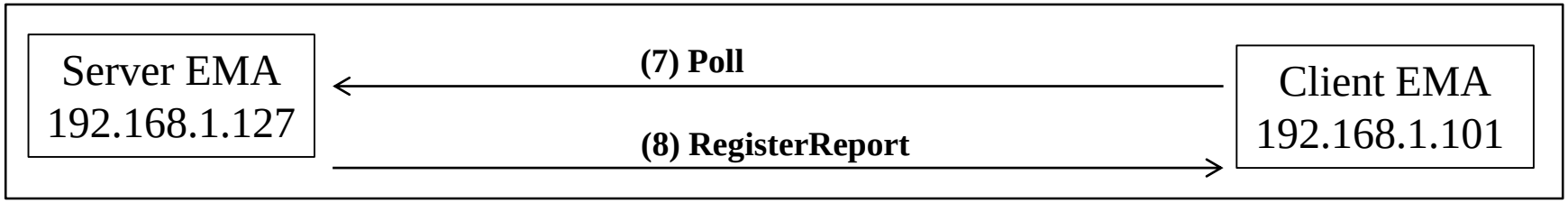
CoAP	192.168.1.101	192.168.1.127	CON, MID:22170, PUT, /Poll (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:22170, 2.05 Content (application/json)

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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

2. Smart Home Energy Framework :
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(7) Poll
(8) RegisterReport

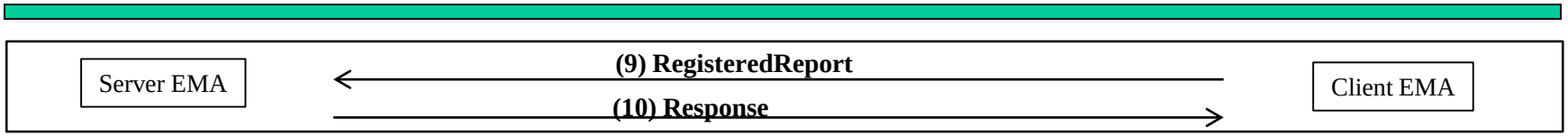
CoAP	192.168.1.101	192.168.1.127	CON, MID:22170, PUT, /Poll (application/json)
CoAP	192.168.1.127	192.168.1.101	ACK, MID:22170, 2.05 Content (application/json)

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: EMAregisteredDRInformation
    - Member Key: EMAregisteredMgnInformation
    - Member Key: service
    - Member Key: time

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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



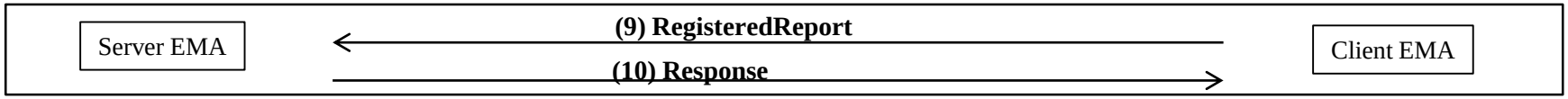
(9) RegisteredReport

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

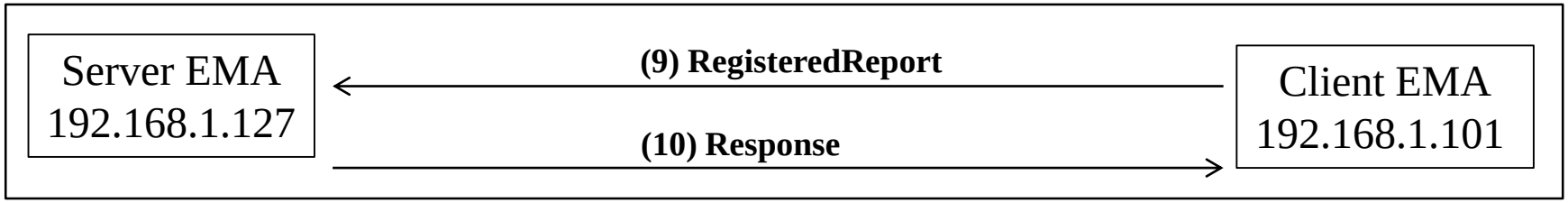
(10) Response

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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

2. Smart Home Energy Framework :

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



(9) RegisteredReport
(10) Response

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)

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,
}

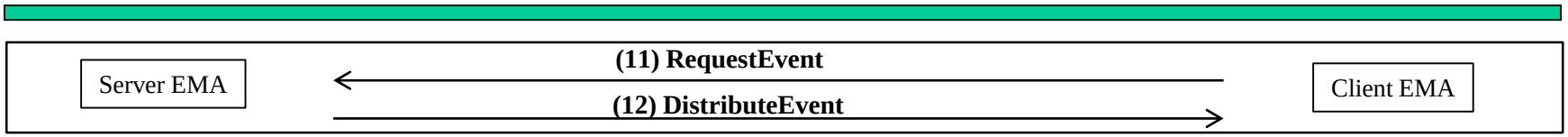
JavaScript Object Notation: application/json
Object
Member Key: SrcEMA
Member Key: DestEMA
Member Key: transportName
Member Key: requestID
Member Key: responseCode
Member Key: version
Member Key: responseDescription
Member Key: threshold
Member Key: service
Member Key: time

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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

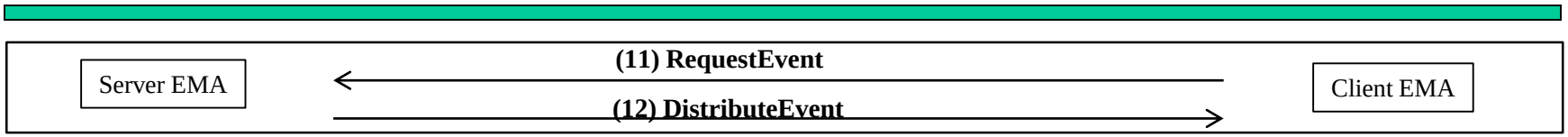


(11) RequestEvent

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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



(12) 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	ei:Response	pyld:requestID	
	responseCode		ei:responseCode	
	responseDescription		ei:responseDescription	
event	eventID	oadrEvent:eiActivePeriod:eventDescriptor	eventID	
	eventSignals		eventSignals	
	modificationNumber		modificationNumber	
	modificationReason		modificationReason	
	priority		priority	
	marketContext		eiMarketContext	
	createdDateTime		createdDateTime	
	eventStatus		eventStatus	
	testEvent		testEvent	
	vtnComment		vtnComment	
	properties	oadrEvent:eiActivePeriod	properties	
	components		components	
	specificDestEMA	oadrEvent:eiTarget	venID	
	dtStart	oadrEvent:eiActivePeriod:properties	dtstart	
	Duration		duration	
	Tolerance		tolerance	
	notification		x-eiNotification	
	rampUp		x-eiRampUp	
	Recovery		x-eiRecovery	
event:eventSignals	eventSignal	oadrEvent:eiEventSignals	eiEventSignal	
signalType으로 Price Event인지 Control Event, Reserve Mode, RealtimeDR인지 구분한다	**Intervals**	oadrEvent:eiEventSignals:eiEventSignal	intervals	
	signalName		signalName	
	signalType (Price Event, Control Event, Reserve Mode, RealtimeDR)		signalType	
	signalID		signalID	
	currentValue		currentValue	

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

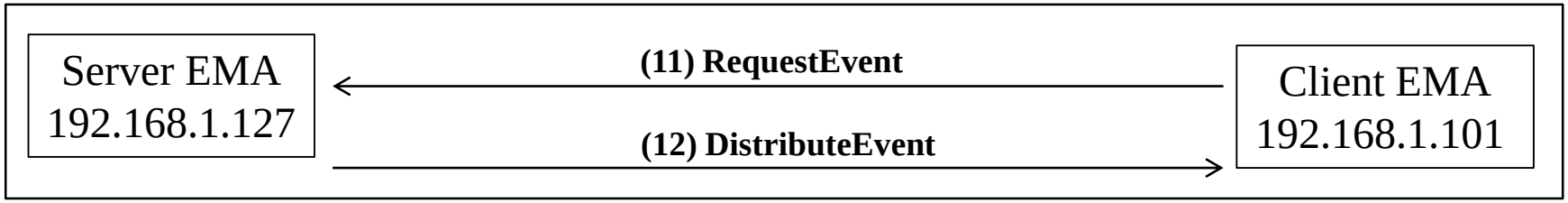
(12) DistributeEvent

Key name				Comments
SrcEMA				source EMA identifier
DestEMA				destination EMA identifier
response	requestID			request identifier
	responseCode			response code
	responseDescription			description of response code
event	eventID			event identifier
	eventSiganls	intervals	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

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

2. Smart Home Energy Framework :

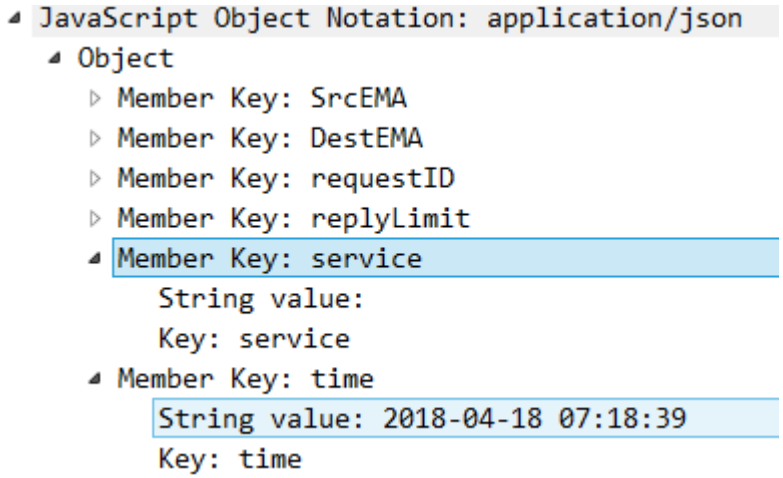
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Session Setup



(11) RequestEvent
(12) DistributeEvent

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

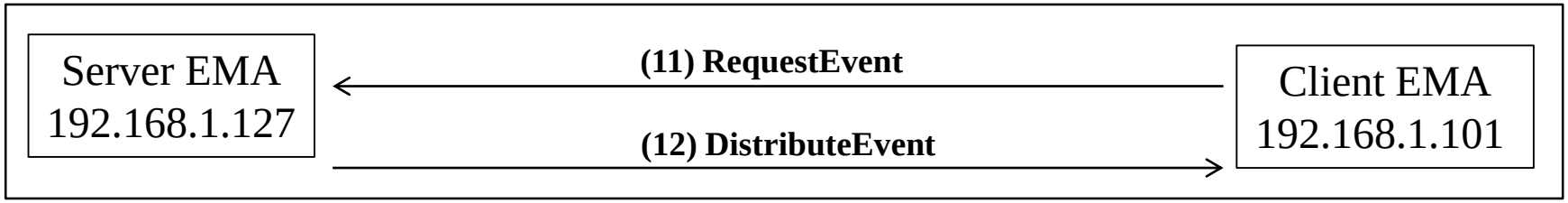
RequestEvent Object{
"SrcEMA" : String,
"DestEMA" : String,
"requestID": String,
~~"replyLimit" : Integer => 삭제,~~
"time" : Date,
"service": String
}



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

2. Smart Home Energy Framework :

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

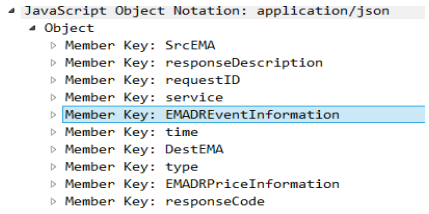


(11) RequestEvent
(12) DistributeEvent

CoAP	192.168.1.101	192.168.1.127	CON, MID:44189, PUT, /RequestEvent (application/json)
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
}



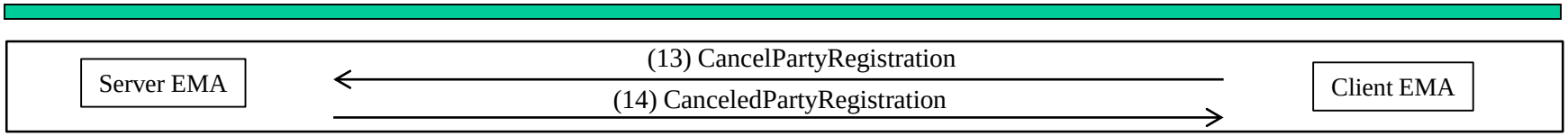
event Array{
"eventID" : String,
"eventSignals" : Array,
"modificationNumber" : Integer,
"modificationReason" : String,
"priority" : Integer,
"marketContext" : String,
"createdDataTime" : Date,
"eventStatus" : String,
"testEvent" : Boolean,
"vtnComment" : String,
"dtstart" : Date,
"duration" : String,
"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 : OpenADR확장된 Profile

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



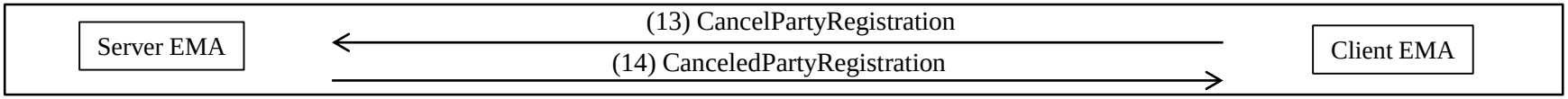
(13) CancelPartyRegistration

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

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



(13) CancelPartyRegistration

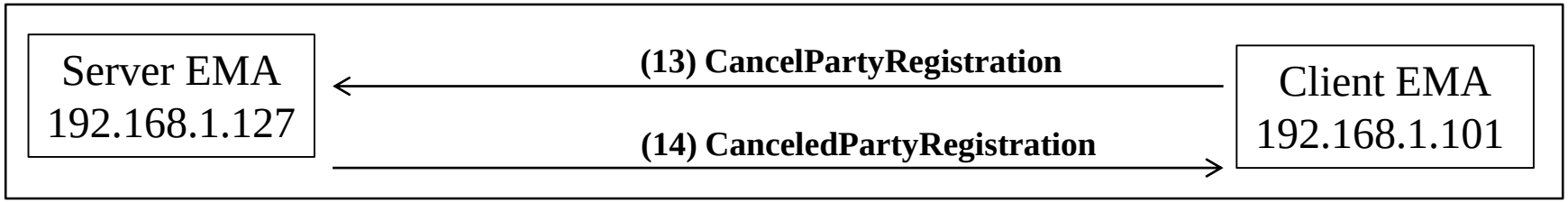
Key name	Comments
SrcEMA	source EMA identifier
DestEMA	destination EMA identifier
requestID	request identifier
registrationID	registration identifier
service	type of service
time	service creation time

(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 : OpenADR확장된 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
}

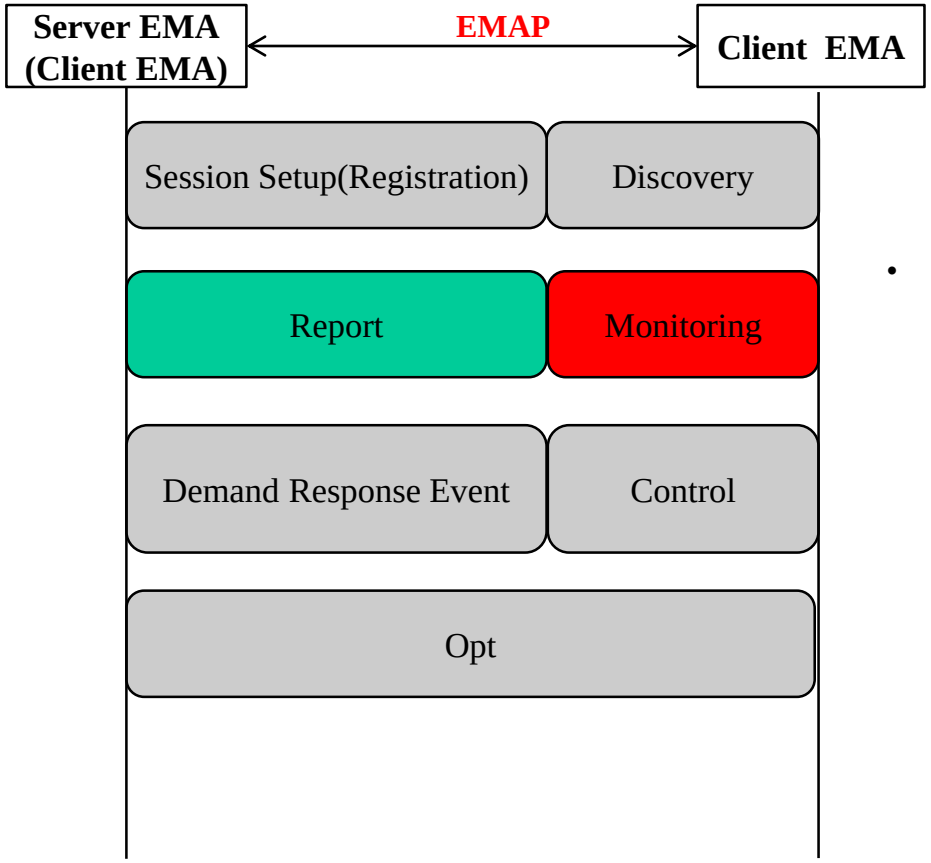
EMAP (CoAP & MQTT/JSON)

UpdateReport

2.2 EMAP(MQTT, CoAP/JSON)
Service : Update Report

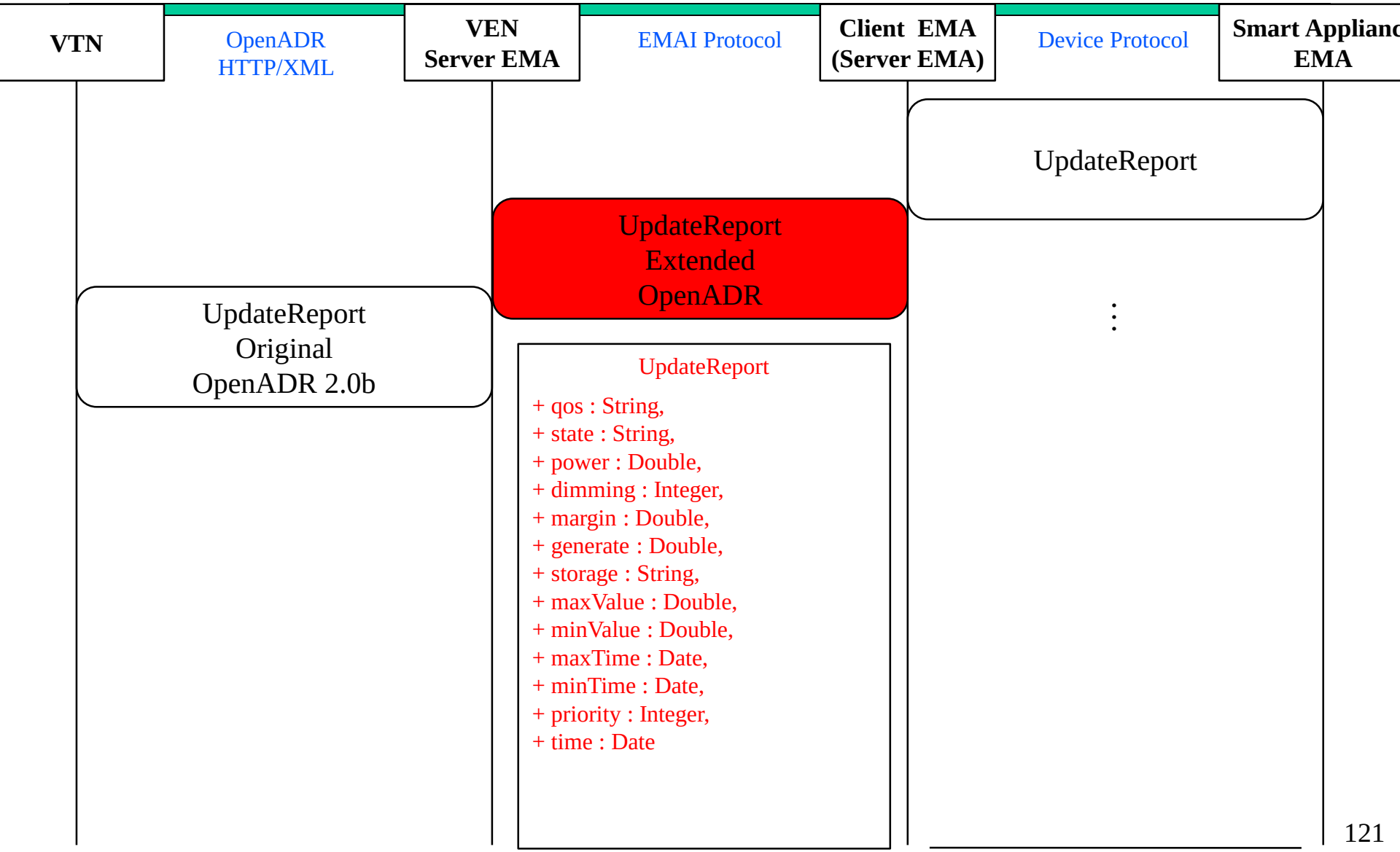
EMAP

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

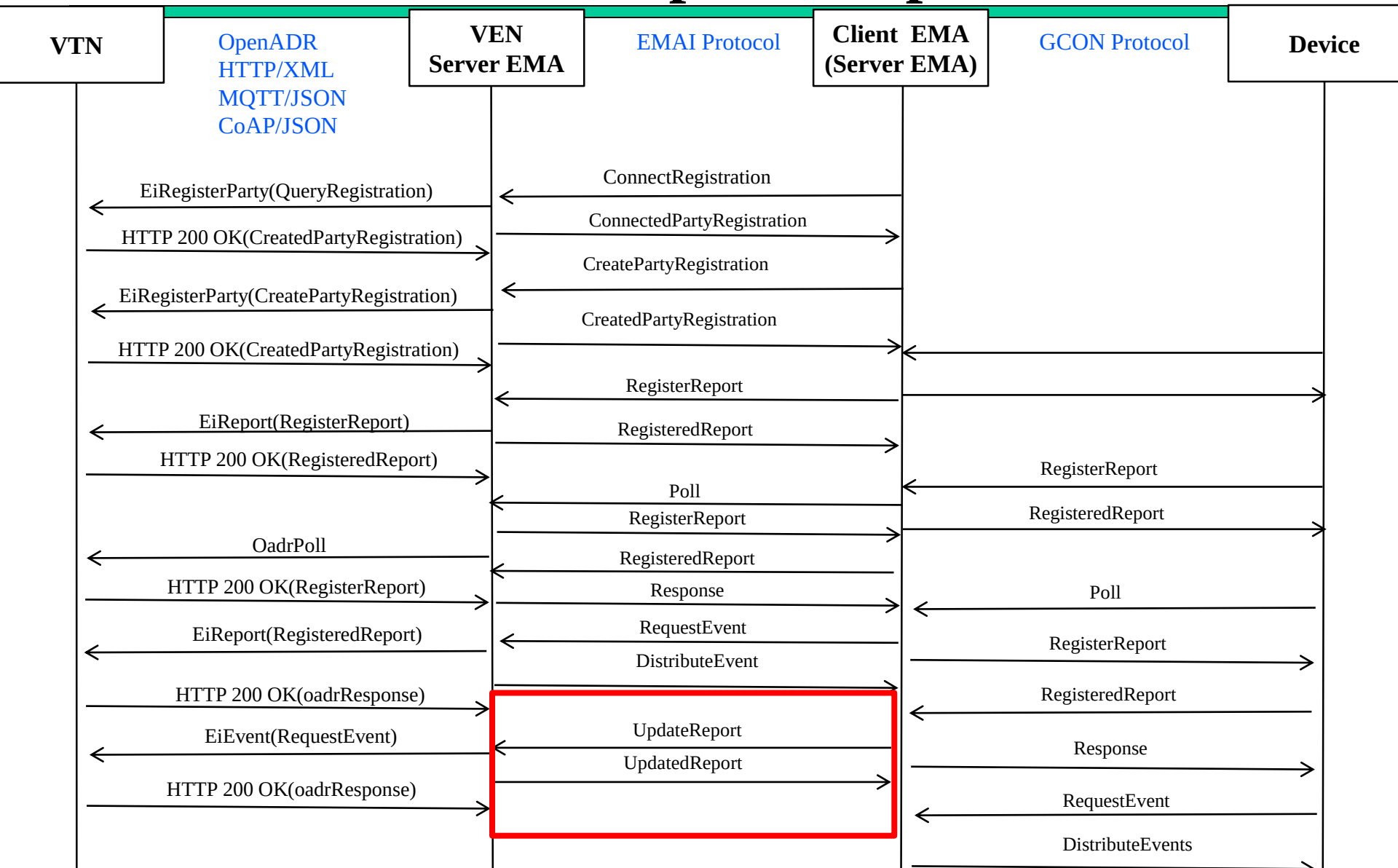


- Report(updateReport)
  - 에너지관리에이전트 간 서로 연결을 수립 할 때 Report을 교환할 때 실시간 에너지에 대한 가격 정보나 클라이언트 에너지관리에이전트의 디바이스 정보 등을 얻음.
  - 하위의 EMA의 정보를 모니터링 하는 단계, Explicit/Abstract 방식으로 Monitoring.
  - Explicit에는 Device Type (LED, ESS, Recloser, Resource, PV)에 대한 정보가 포함
  - Implicit에는 Device Type을 제외한 정보가 포함되어 있어 필요로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

2.2 EMAP(MQTT, CoAP/JSON)
Service : Update Report



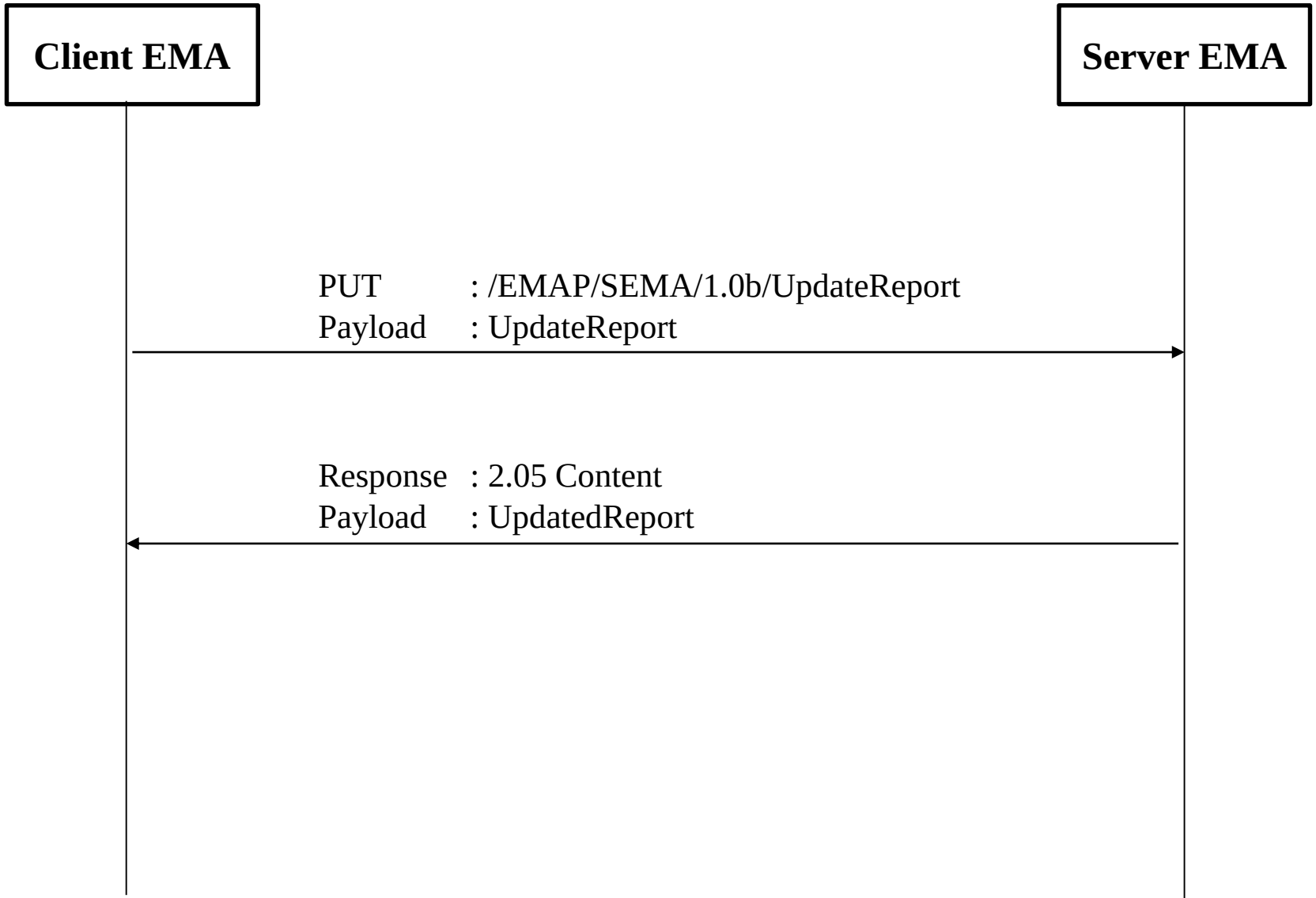
2.2 EMAP(MQTT, CoAP/JSON)
Service : Update Report



EMAP
(2) UpdateReport

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

2.2 EMAP(CoAP/JSON)
Service : Update Report



파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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 이름으로 존재)			
time				RandomizableEvent:creation Time	
requestID		requestID			
type(Implicit, Explicit)					
report	duration	oadrReport	duration		
	reportDescription		oadrReportDescription		
	reportRequestID		reportRequestID		
	reportSpecifierID		reportSpecifierID		
	reportName		reportName		
	createdDateTime		createdDateTime		
report:reportDescription	rID	oadrReport:oadrReportDescription	rID		
	resourceID		resourceID		
	deviceType				EndDeviceControlType:type
	reportType		reportType		
	itemUnits		itemUnits		
	siScaleCode		siScaleCode		
	marketContext		marketContext		
	MinPeriod		oadrMinPeriod		
	MaxPeriod		oadrMaxPeriod		
	OnChange		oadrOnChange		
	itemDescription		itemDescription		
	powerAttributes		powerAttributes		
	qos			EndDeviceControl:loadShiftForward	
	state			DeviceStatus:opState	
	power				Readings: value
	dimming			Subscription:Level	
	margin			IdentifiedObject:DemandResponseProgram:availabilityUpdatePowerChnageThreshold	
	generate				SolarEventProfile:SolarInvetterStatus:value
	storage				BatteryEventProfile:BatteryStatus:value
	maxValue				solarModule:SolarCapability:MaxVal
	minValue				solarModule:SolarCapability:MinVal
					ResourceReading_MMTR:A

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

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	ei:eiResponse	pyld:requestID		
responseCode		ei:responseCode		
responseDescription		ei:responseDescription		
service	(Tag 이름으로 존재)			
type			IdentifiedObject:TrafiiProfile:ServiceKind	
time			RandomizableEvent:creation Time	

type: Explicit, Implicit

2. Smart Home Energy Framework :

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

(1) UpdateReport

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 ven)
	reportName			report name
	createdDateTime			created time of this report
	reportDescription	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
		powerAttributes	hertz	pulse frequency of power
			voltage	voltage of power
			ac	Is this AC power? (True or False)
service				type of service
time				service creation time
type				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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 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,
~~"EMAregisteredDRinformation" : Object =>변경~~
~~"EMAregisteredMgnInformation" : Object => 변경,~~
- **"report" : Array,**
"time": Date,
"service" : String,
"type": String (Explicit, Implict인지 구분)
}

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
}

JavaScript Object Notation: application/json
Object
Member Key: "SrcEMA"
Member Key: "DestEMA"
Member Key: "requestID"
Member Key: "reportName"
Member Key: "reportType"
Member Key: "EMAupdatedDRInfo"
Member Key: "EMAupdatedMgnInfo"
Member Key: "service"
Member Key: "time"

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

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
}

```
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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

2. Smart Home Energy Framework :

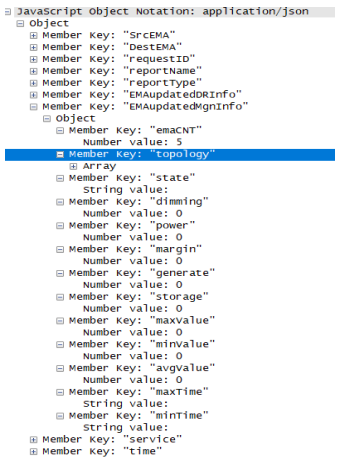
2.2 EMAP(CoAP/JSON, MQTT/JSON) : UpdateReport (Implicit, Explicit)

(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,
~~"EMAregisteredDRinformation" : Object =>변경~~
~~"EMAregisteredMgnInformation" : Object => 변경,~~
- **"report" : Array,**
"time": Date,
"service" : String,
"type": String (Explicit, Implict인지 구분)
}

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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

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)

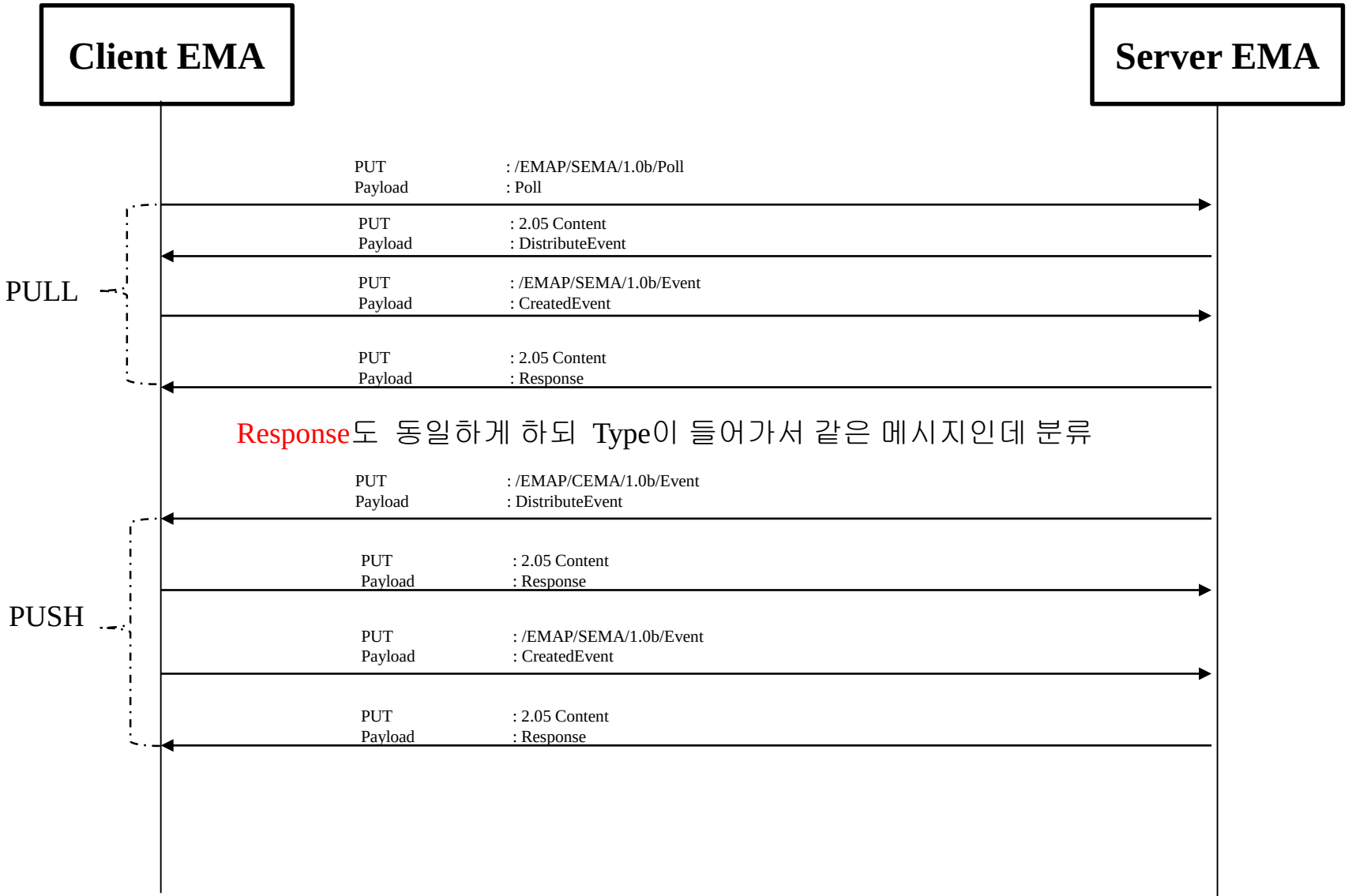
UpdatedReport Object{
"SrcEMA" : String,
"DestEMA" : String,
"requestID": String,
"responseCode": Integer,
"responseDescription" : String,
~~"EMAEnergyinfo" : Object =>삭제~~
"service": String,
"time": Date
}

```
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"
        Object
          Member Key: "emaCNT"
              Number value: 5
          Member Key: "topology"
            Array
      Member Key: "time"
      Member Key: "DestEMA"
      Member Key: "responseCode"
```

EMAP
(3) Event_PULL

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

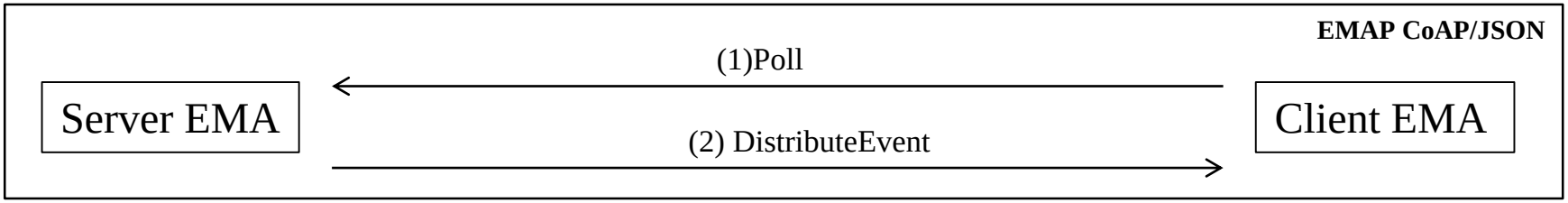
2.2 EMAP(CoAP/JSON)
Service : Event



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

2. Smart Home Energy Framework :

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



(1)Poll

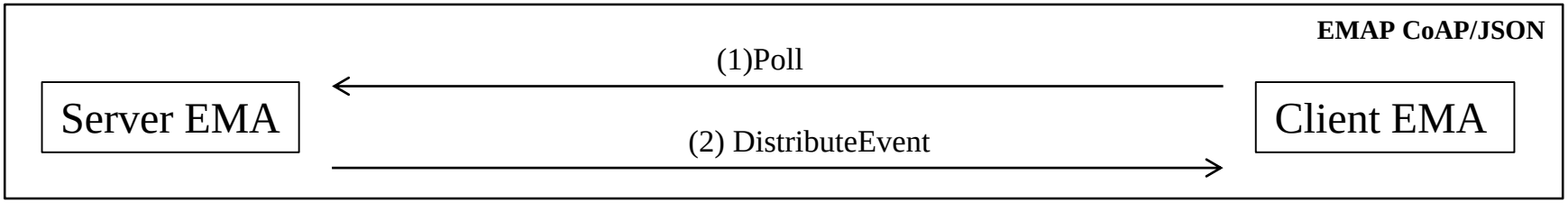
(2) DistributeEvent

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

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

2. Smart Home Energy Framework :

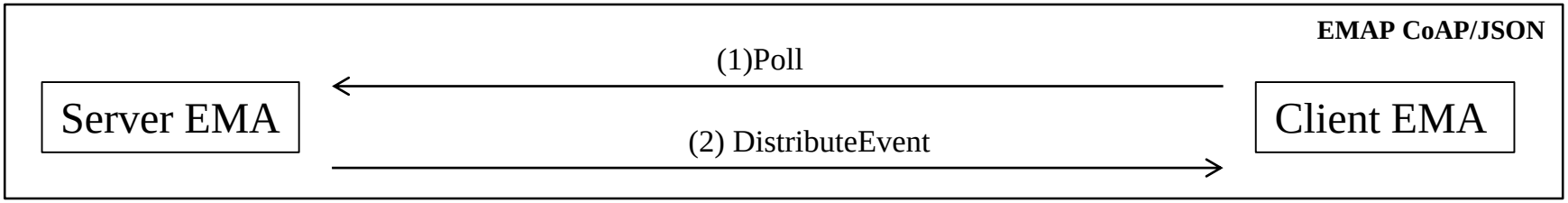
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(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	ei:Response	pyld:requestID
	responseCode		ei:responseCode
	responseDescription		ei:responseDescription
event	eventID	oadrEvent:eiActivePeriod:eventDescriptor	eventID
	eventSignals		eventSignals
	modificationNumber		modificationNumber
	modificationReason		modificationReason
	priority		priority
	marketContext		eiMarketContext
	createdDateTime		createdDateTime
	eventStatus		eventStatus
	testEvent		testEvent
	vtnComment		vtnComment
	properties	oadrEvent:eiActivePeriod	properties
	components		components
	specificDestEMA	oadrEvent:eiTarget	venID
	dtStart	oadrEvent:eiActivePeriod:properties	dtstart
	Duration		duration
	Tolerance		tolerance
	notification		x-eiNotification
	rampUp		x-eiRampUp
	Recovery		x-eiRecovery
signalType으로 Price Event인지 Control Event, Reserve Mode	eventSignal	oadrEvent:eiEventSignals	eiEventSignal
	Intervals	oadrEvent:eiEventSignals:eiEventSignal	intervals
	signalName		signalName
	signalType (Price Event, Control Event, Reserve M		signalType

2. Smart Home Energy Framework :
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH

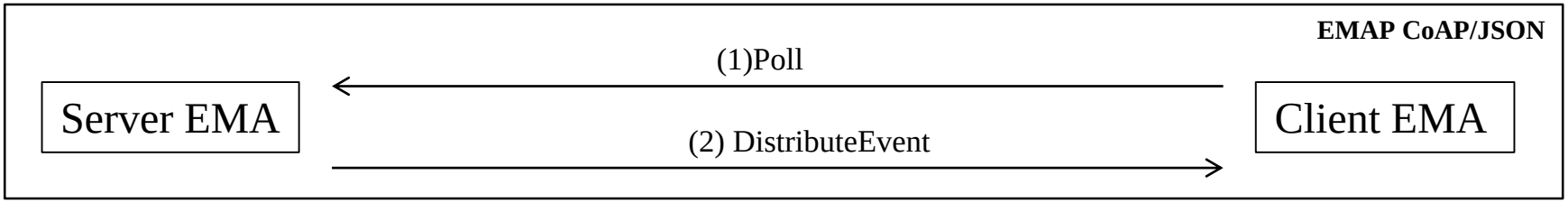


(1)Poll

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



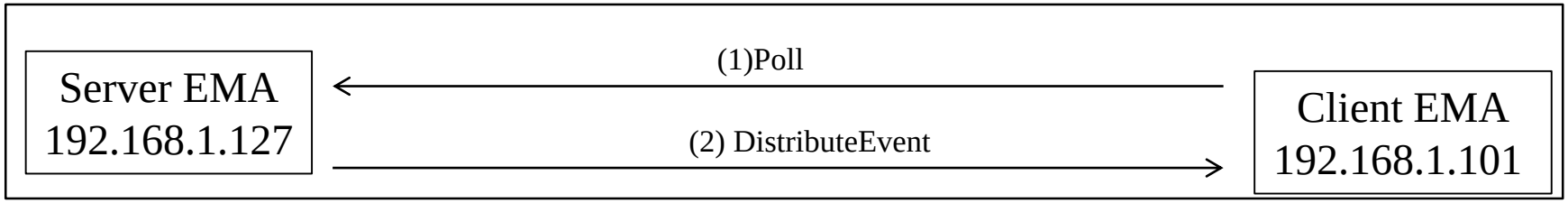
(2) DistributeEvent

Key name				Comments
SrcEMA				source EMA identifier
DestEMA				destination EMA identifier
response	requestID			request identifier
	responseCode			response code
	responseDescription			description of response code
event	eventID			event identifier
	eventSiganls	intervals	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

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

2. Smart Home Energy Framework :

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



(1) Poll
(2) DistributeEvent

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

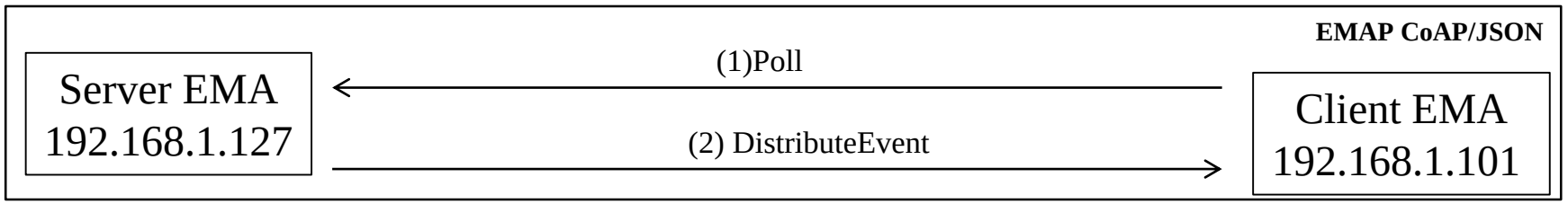
Poll JSONObject{
"SrcEMA" : String,
"DestEMA" : String,
~~"version": Integer=>삭제,~~
~~"type" : String => 삭제(RegisteredReport로 변경)~~
"service" : String,
"time" : Date
}

- 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

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

2. Smart Home Energy Framework :

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

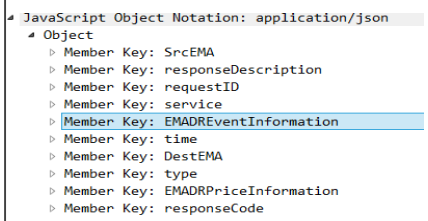


(1) Poll
(2) DistributeEvent

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,
"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
}

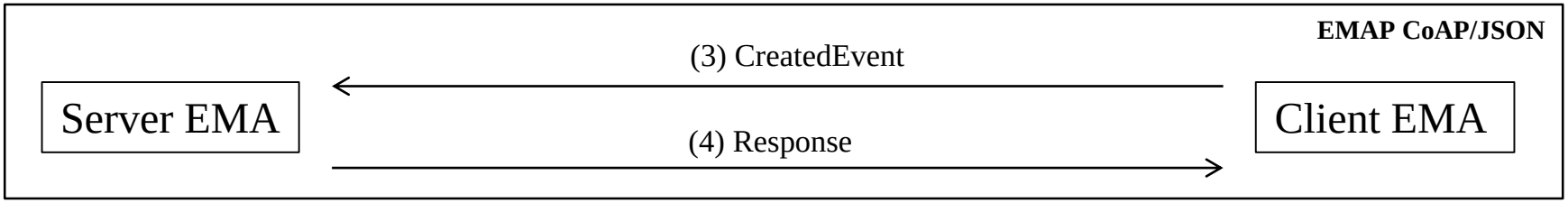
eventSignals Object{
"eventSignal" : String,
"intervals" : Array,
"signalName" : String,
"signalType" : String, (Price Event, Control Event, Reserve Mode, RealtimeDR인지 구분)
"signalID" : String,
"currentValue" : Double,
"threshold" : Double,
"capacity" : Double,
"price" : Integer,
"unit" : String,
}

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

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



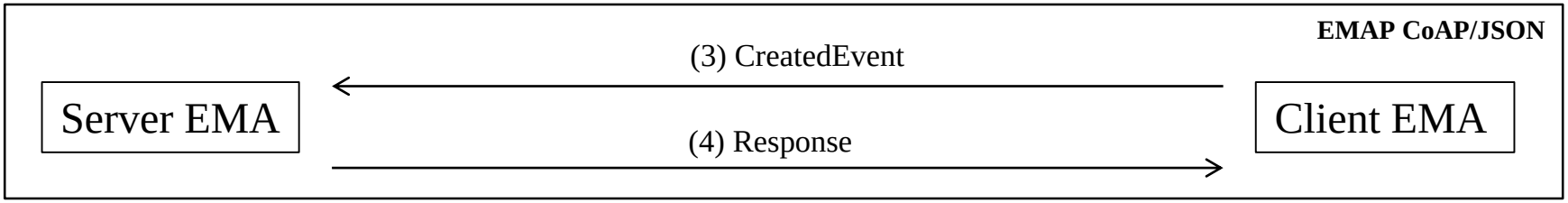
(3) CreatedEvent

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

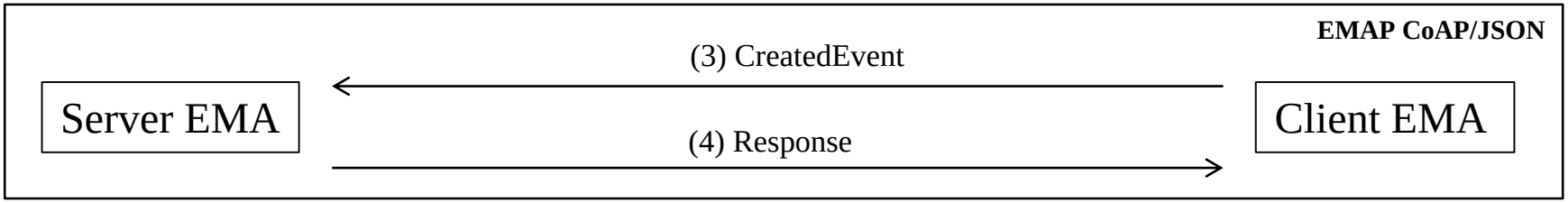
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(4) Response

Key Name	Reference		
	OpenADR 2.0b		SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID		
DestEMA	ei:venID		
responseCode	ei:eiResponse	ei:responseCode	
responseDescription		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 paticipate event or not
service	type of service
time	service creation time

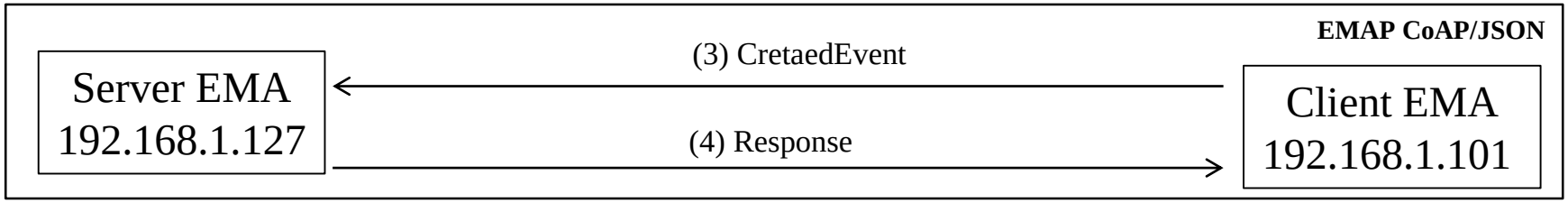
(4) Response

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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

2. Smart Home Energy Framework :

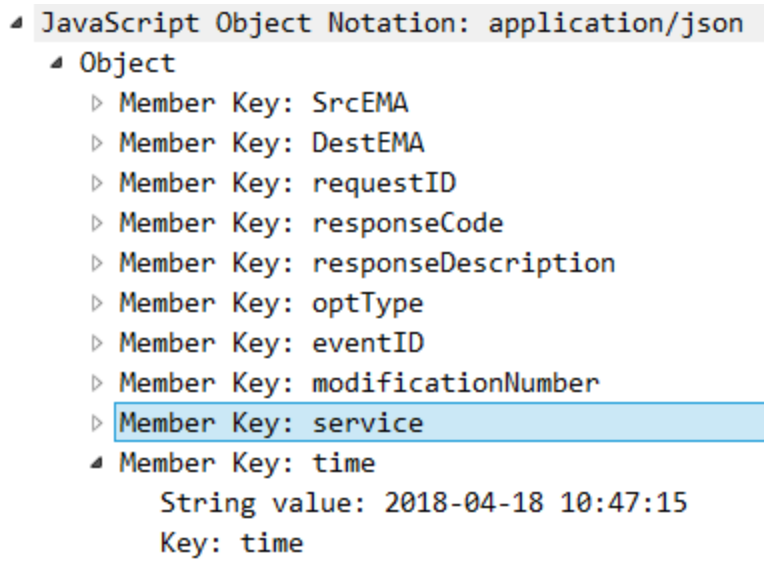
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL,PUSH



(3) CreatedEvent
(4) Response

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

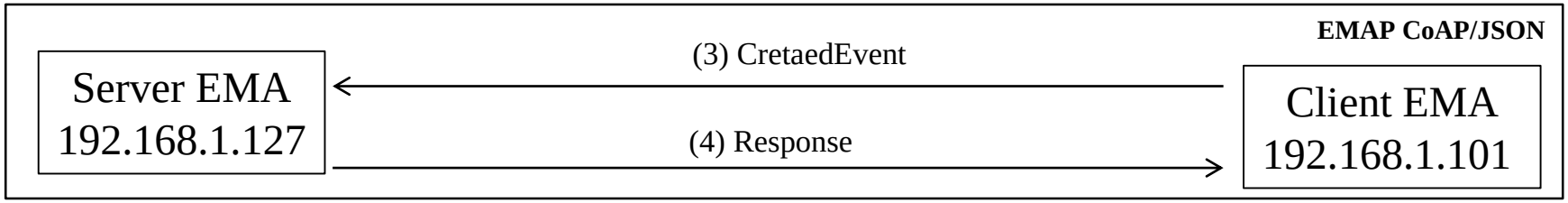
CreatedEvent Object{
"SrcEMA" : String,
"DestEMA" : String,
"requestID": String,
"responseCode": Integer,
"reponseDescription": String,
"optType": String,
"eventID" : String,
"modificationNumber" : Integer,
"service" : String,
~~"type" : String,~~
"time" : Date
}



파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 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:8526, PUT, /CreatedEvent (application/json)
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

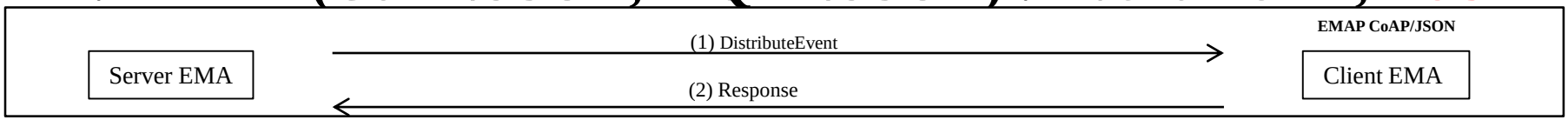
EMAP
(3) Event_PUSH

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

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



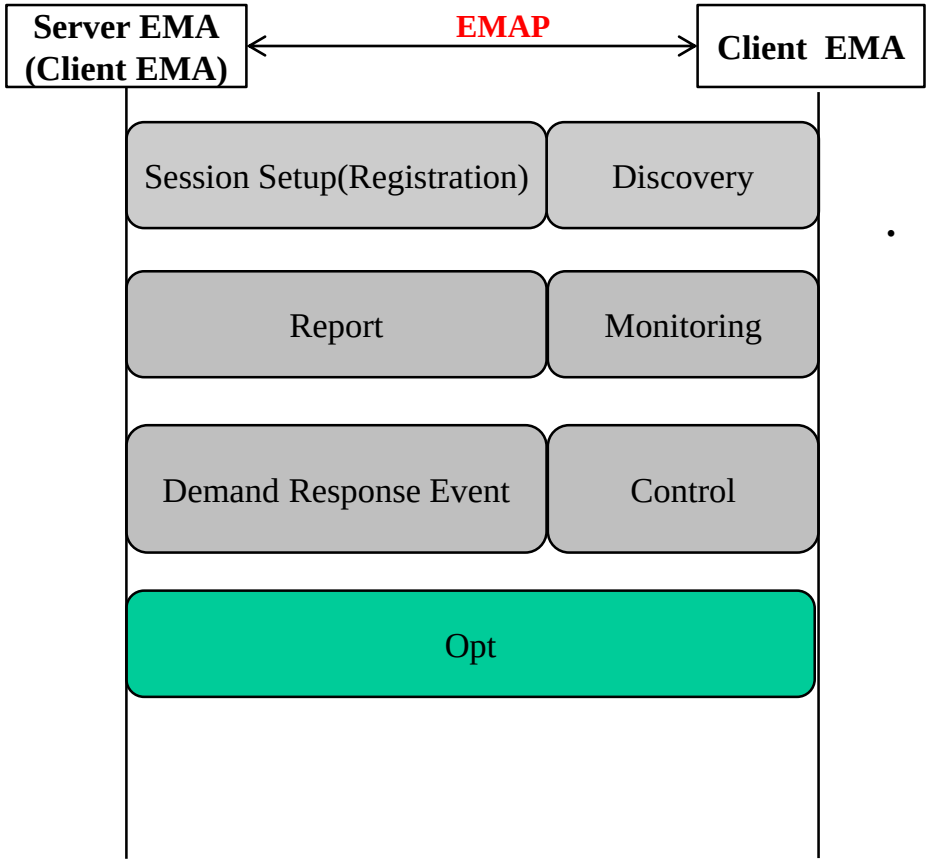
(1) DistributeEvent

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

2.2 EMAP(MQTT, CoAP/JSON)
Service : EiEvent

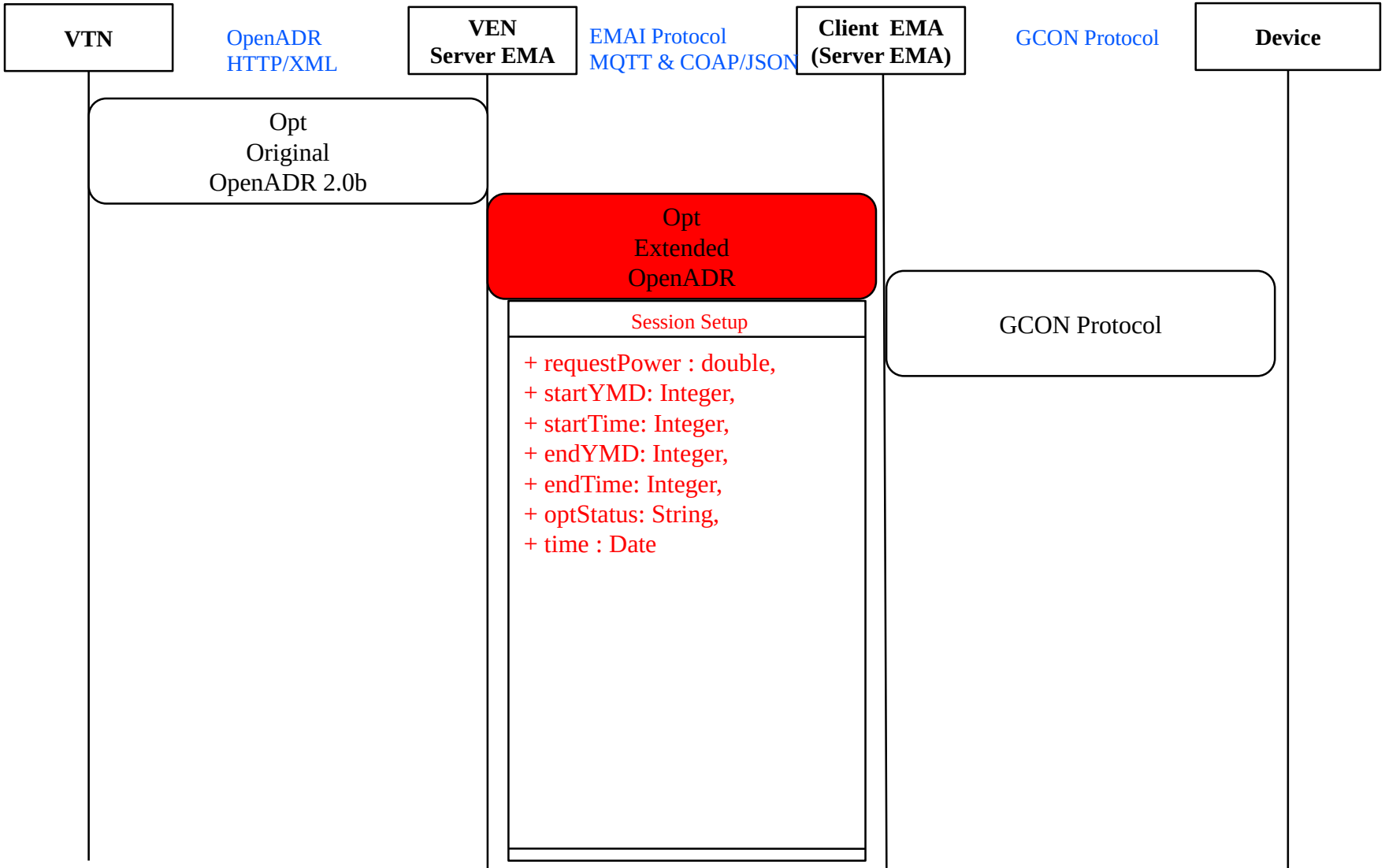
EMAP

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



- 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)
Service : Opt

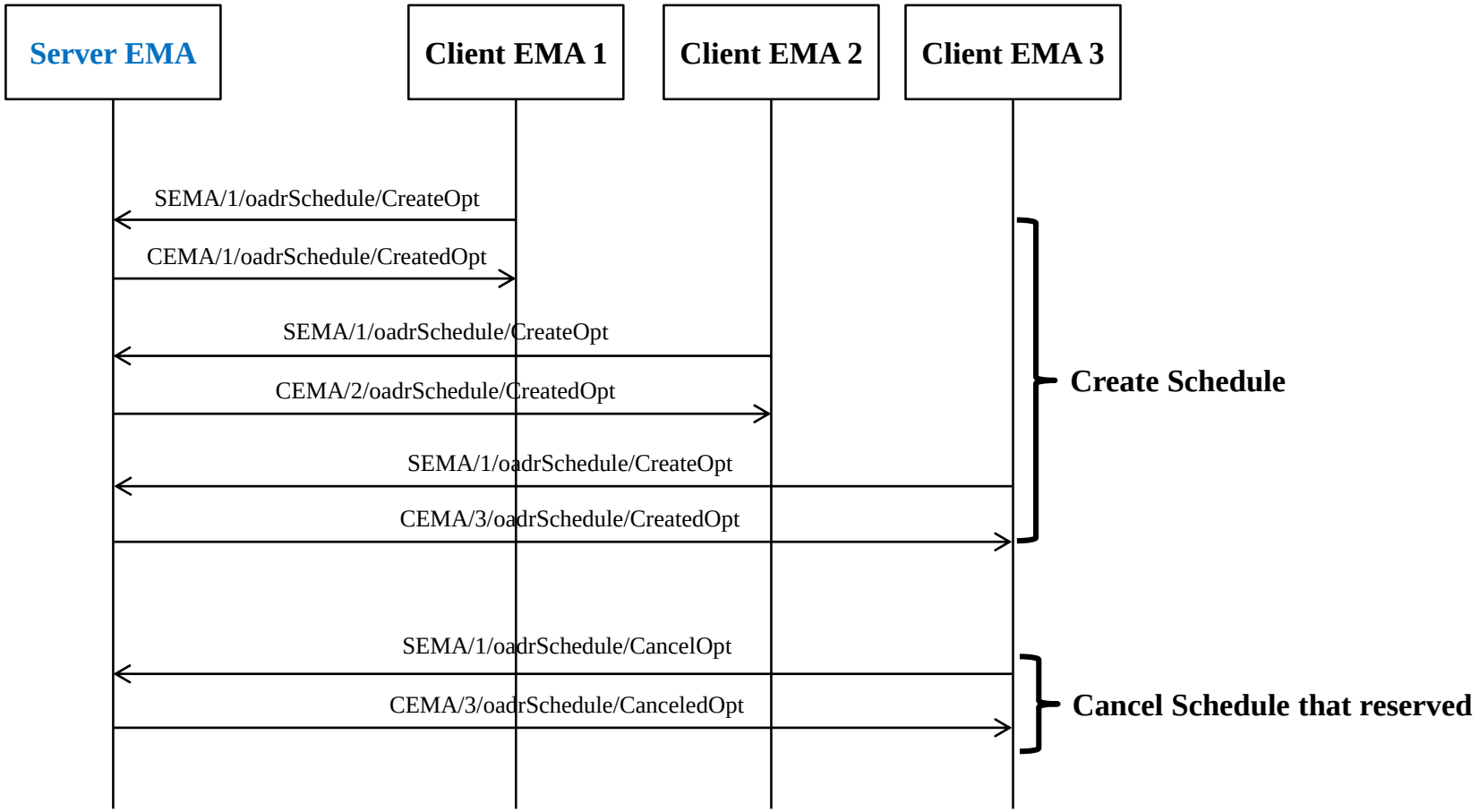


EMAP
(4) Opt

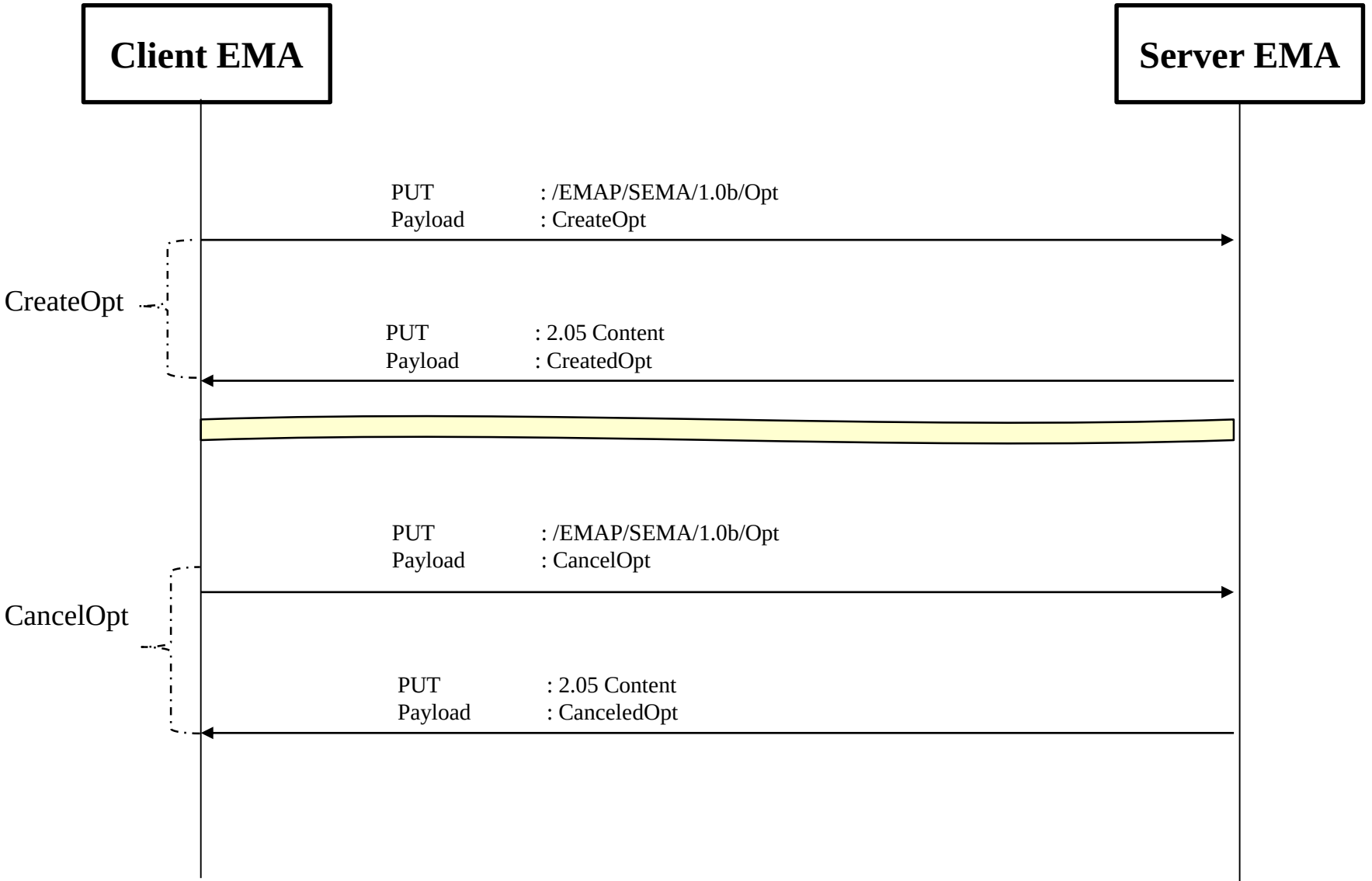
- **CoAP/JSON**
- **MQTT / JSON**

2. Smart Home Energy Framework :

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



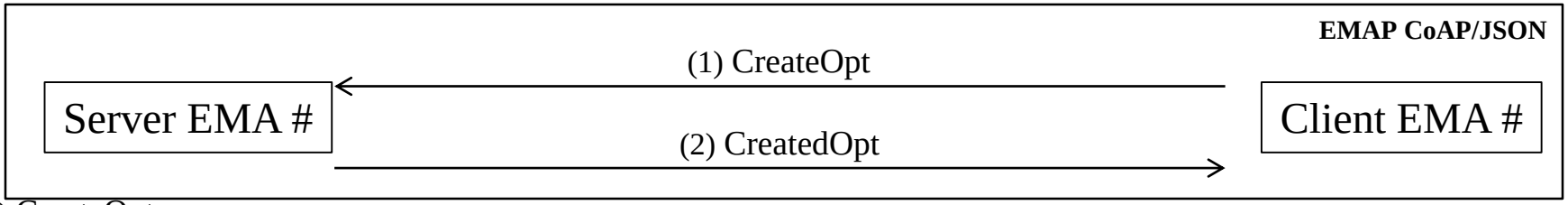
2.2 EMAP(CoAP/JSON)
Service : Opt



파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

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



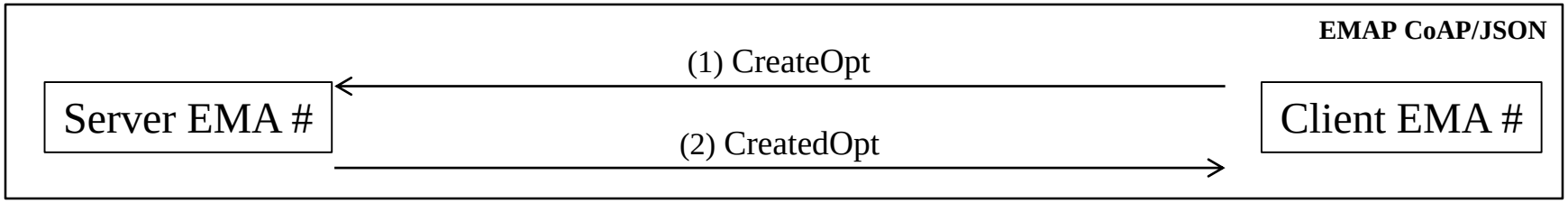
(1) CreateOpt

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

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

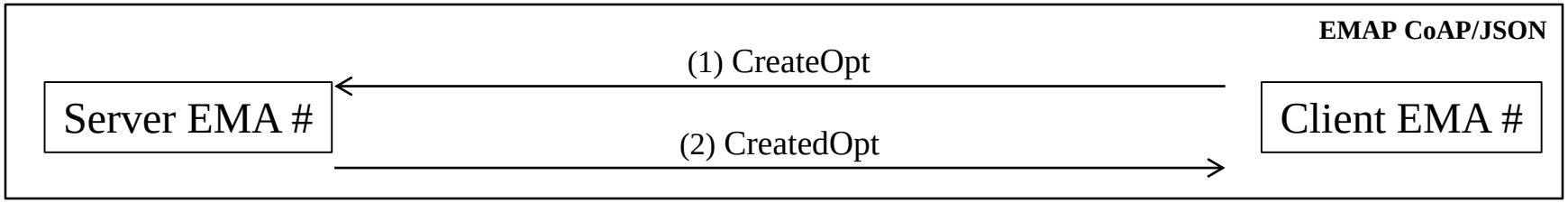


(2) CreatedOpt

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:TrafiiProfile: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
	endYMD	opt end date
	endTime	opt end time
requestID		request identifier
service		type of service

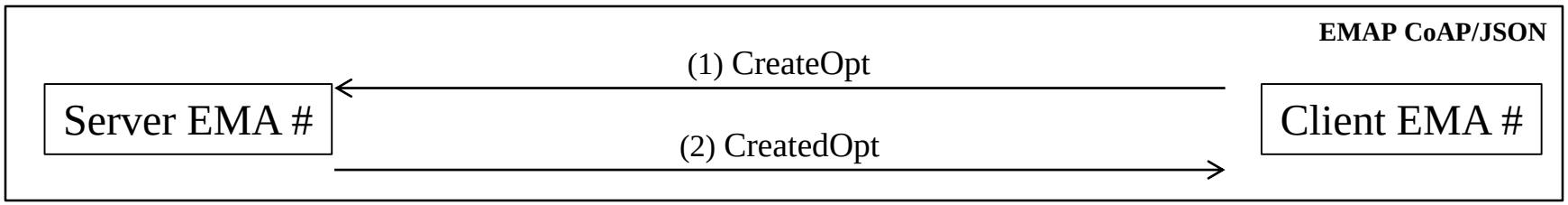
(2) CreatedOpt

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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

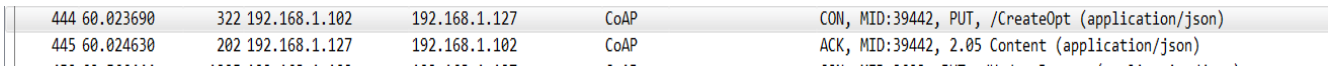
2. Smart Home Energy Framework :

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



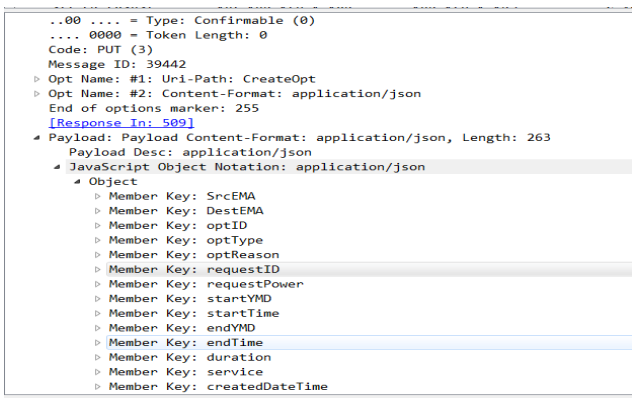
(1) CreateOpt

(2) CreatedOpt

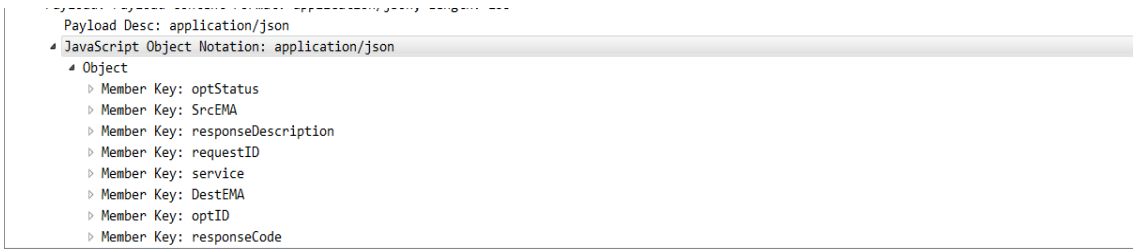


CreateOpt Object{
“SrcEMA” : String,
“DestEMA” : String,
“optID”: String,
“optType”: String,
“optReason”: String,
“requestID” : String,
“marketContext” : String,
“createdDateTime”: Date,
“service” : String,
“available” : Array,
}

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



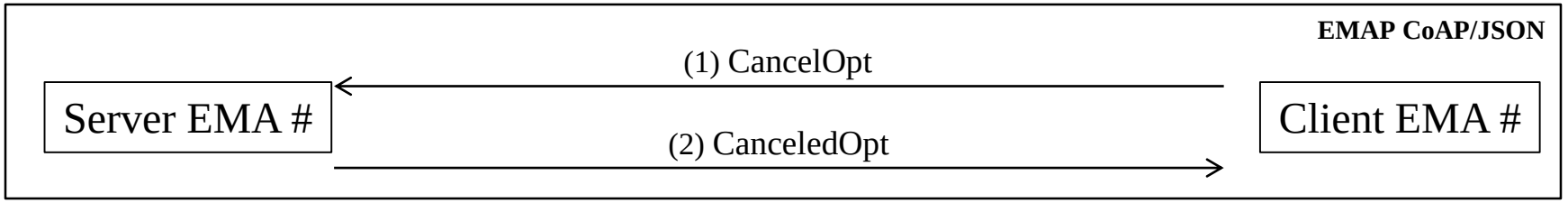
CreatedOpt Object{
“SrcEMA” : String,
“DestEMA” : String,
“optID”: String,
“requestID”: String,
“responseCode”: Integer,
“responseDescription” : String,
“service” : String,
“optStatus”: String
}



파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

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



(1) CancelOpt

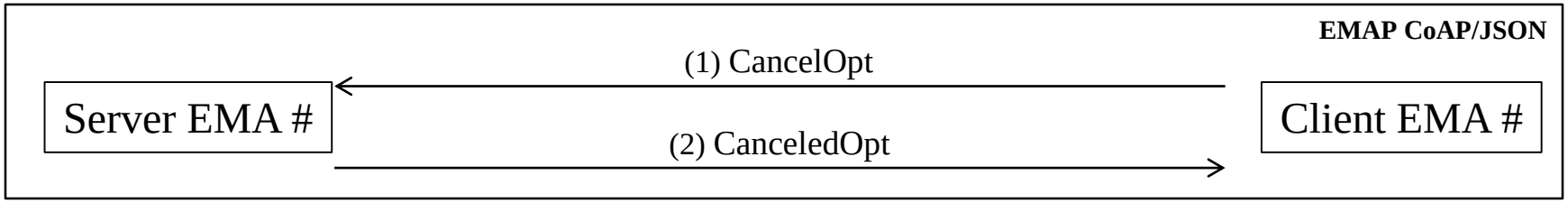
Key Name	Reference	
	OpenADR 2.0b	SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID	
DestEMA	ei:venID	
requestID	pyld:requestID	
optID	ei:OptID	
time		RandomizableEvent:creation Time
service	(Tag 이름으로 존재)	

(2) CanceledOpt

Key Name	Reference	
	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

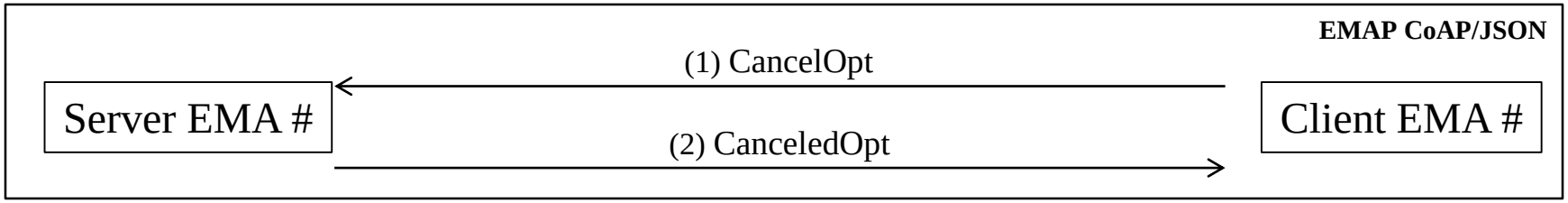
(2) CanceledOpt

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

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

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
```

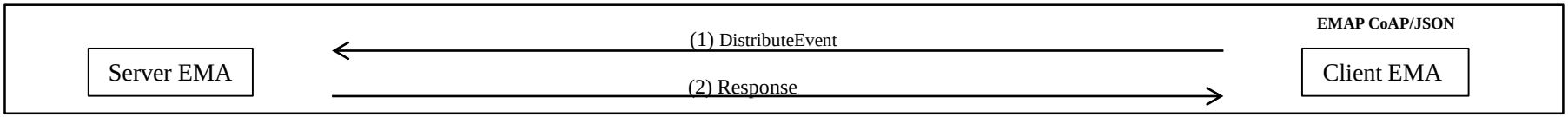
CanceledOpt Object{
"optID": String,
"SrcEMA" : String,
"DestEMA" : String,
"requestID": String,
"responseCode": Integer,
"responseDescription" : Strin
"service": String
}

```
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
```

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

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

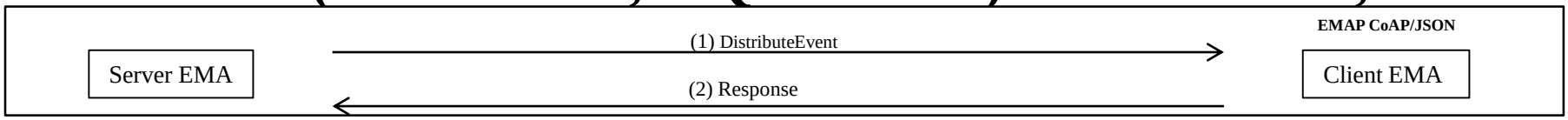


(2) Response

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

2. Smart Home Energy Framework :

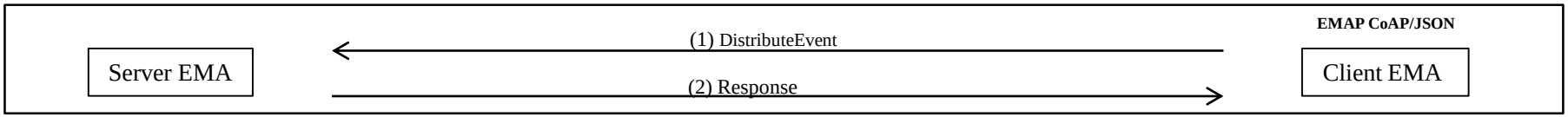
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(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
event	eventID			event identifier
	eventSiganls	intervals	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



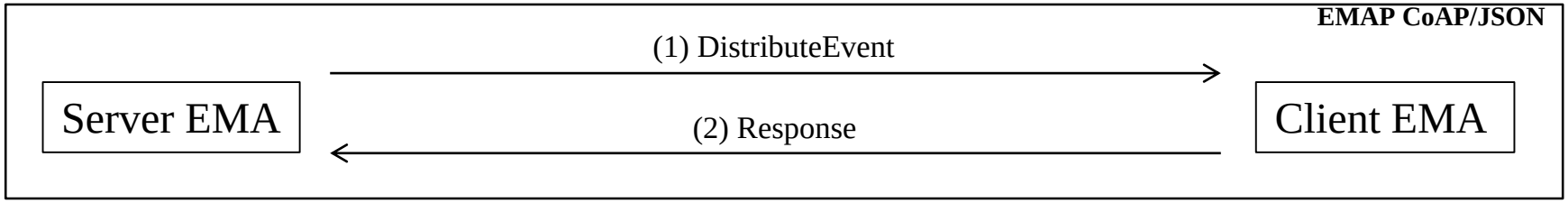
(2) Response

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 : OpenADR확장된 Profile
초록색 : 삭제 또는 변경

2. Smart Home Energy Framework :

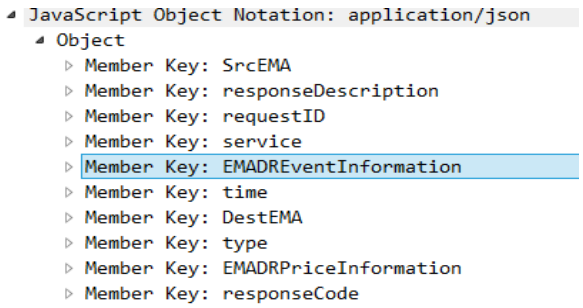
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(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
}



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
}

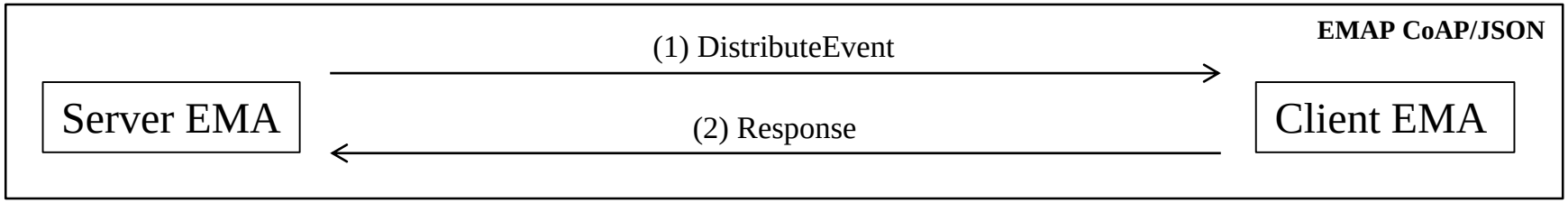
eventSignals Object{
"eventSignal" : String,
"intervals" : Array,
"signalName" : String,
"signalType" : String, (Price Event, Control Event, Reserve Mode, RealtimeDR인지 구분)
"signalID" : String,
"currentValue" : Double,
"threshold" : Double,
"capacity" : Double,
"price" : Integer,
"unit" : String,
}

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

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

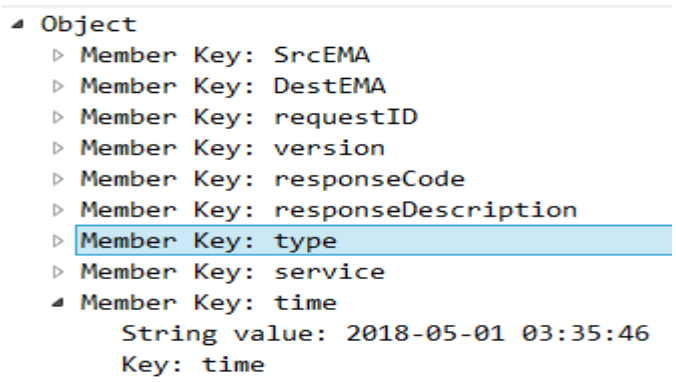
2. Smart Home Energy Framework :

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,
"type": String,
"time" : Date
}

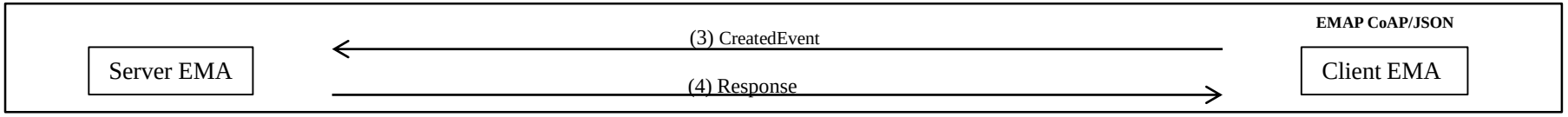


type : Registration ,Periodic , report, Event, Price인지 구분

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

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



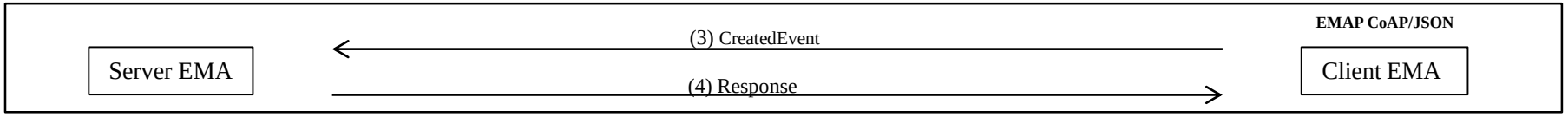
(3) CreatedEvent

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

파란색 : 기존 OpenADR Tag 부분
빨간색 : OpenADR확장된 Profile

2. Smart Home Energy Framework :

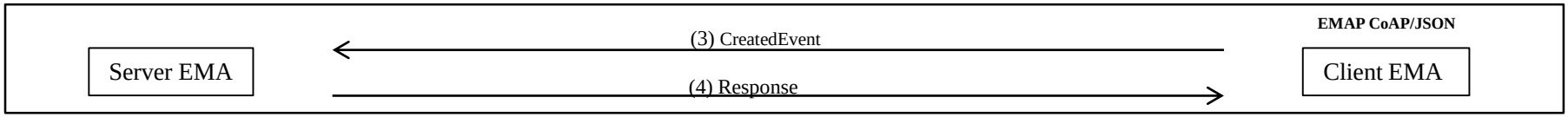
2.2 EMAP(CoAP/JSON, MQTT/JSON) : Event-PULL, PUSH



(4)Response

Key Name	Reference		
	OpenADR 2.0b		SEP 2.0(IEC 61968)
SrcEMA	ei:vtnID		
DestEMA	ei:venID		
responseCode	ei:eiResponse	ei:responseCode	
responseDescription		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



(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 paticipate event or not
service	type of service
time	service creation time

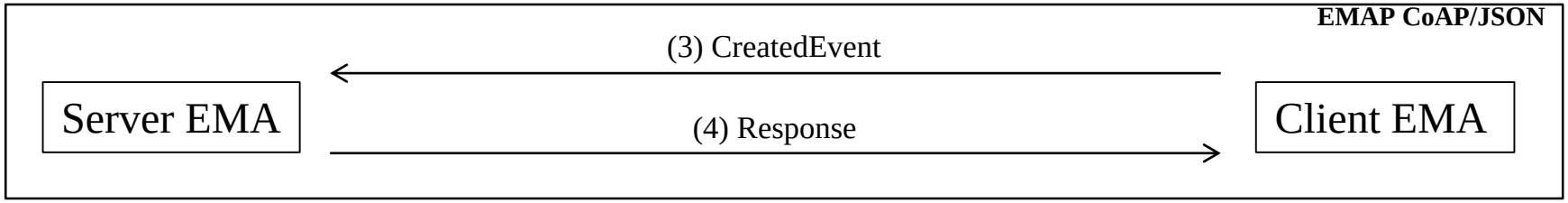
(4)Response

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 : OpenADR확장된 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)

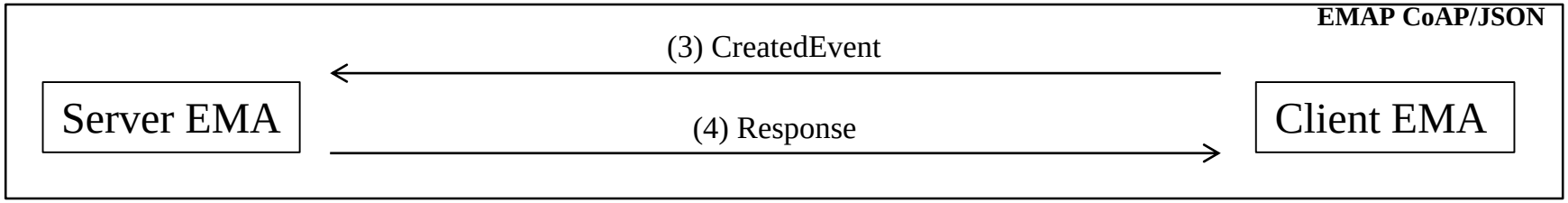
CreatedEvent Object{
"SrcEMA" : String,
"DestEMA" : String,
"requestID": String,
"responseCode": Integer,
"reponseDescription": String,
"optType": String,
"eventID" : String,
"modificationNumber" : Integer,
"service" : String,
"time" : Date
}

```
▾ 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: responseDescription
    ▸ 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 : OpenADR확장된 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,
"type": 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
```

Appendix

- Java Thread Management
  - Smart Meter : Periodical On Demand Request

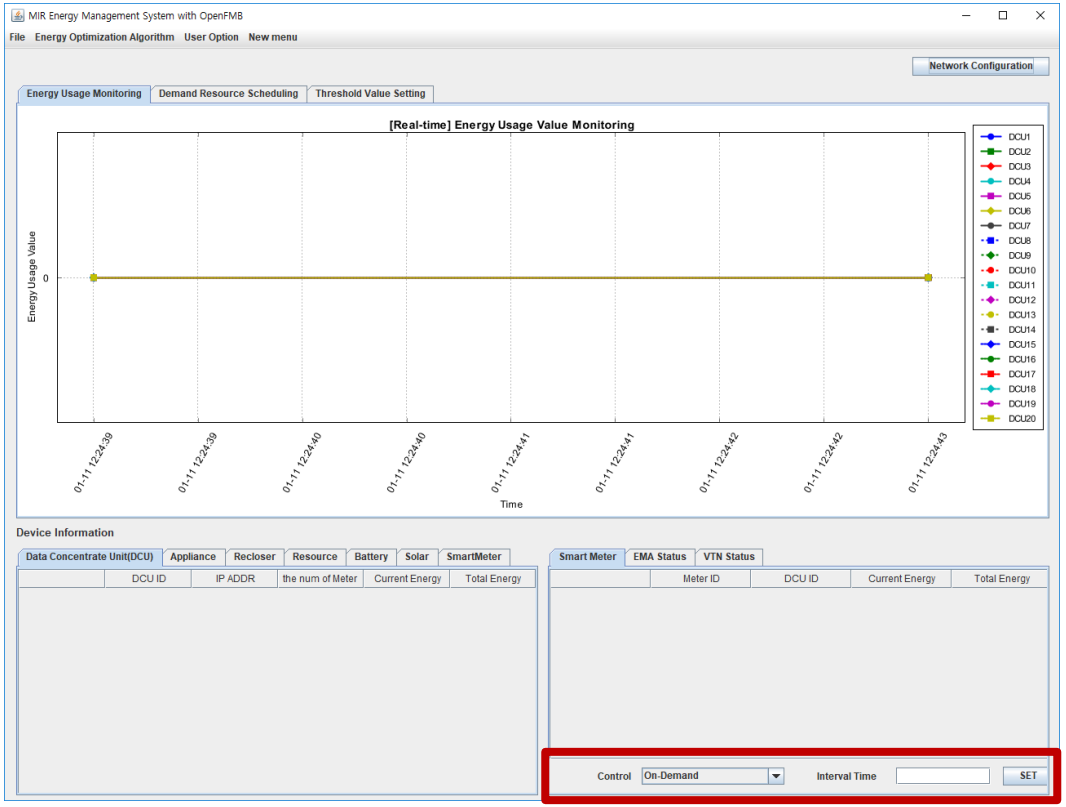
Java Thread Management

```
JButton btnNewButton = new JButton("SET");
btnNewButton.setBounds(556, 242, 62, 23);
add(btnNewButton);

btnNewButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
                if (global.onDemandCNT > 0) {
                        Thread[] ee = new Thread[10000];
                        Thread.enumerate(ee);                          현재 실행되고 있는 Thread List를 Thread 배열에 저장한다

                        for (int i = 0; i < ee.length; i++) {
                                if (ee[i].getName().equals("onDemand")) {   Thread 배열에 'onDemand' 라는 이름인 Thread를 검색한다.
                                        ee[i].interrupt();                  해당 Thread가 있을 경우 Interrupt 명령어를 통해 종료한다.
                                        break;                              검색 시간을 줄이기 위해 break;
                                } else {                                    Stop 이라는 명령어를 쓰는 것 보다 수행하는 Thread Class 에
                                }                                           Interrupt Option 을 걸어 주는 것이 효율 적인 Thread 관리 방법
                        }
                }
                int sec = Integer.parseInt(textField.getText()) * 1000;
                RunnableJob onDemandInterval = new RunnableJob(sec);
                Thread onDemandRequest = new Thread(onDemandInterval);
                onDemandRequest.setName("onDemand");
                onDemandRequest.start();
                global.onDemandCNT += 1;

        }
});
```

Java Thread management

```
public class RunnableJob implements Runnable {
        public int intervalTime = 1000;
        public RunnableJob(int interval) {
                this.intervalTime = interval;
        }
        @Override
        public void run() {
                try {
                        while (!Thread.classurrentThread().isInterrupted()) {          Thread.isInterrupt() 이 아니라 현재 수행 중인 Thread 를 종료하는 코드
                                long sTime = System.classurrentTimeMillis();
                                Thread.sleep(intervalTime);
                                long cTime = System.classurrentTimeMillis();
                                long rTime = cTime - sTime;
                                System.out.println(rTime / 1000 + "초");
                                Object[] dcuList = global.dcuHashMap.keySet().toArray();
                                for (int i = 0; i < dcuList.length; i++) {
                                        System.out.println(global.dcuHashMap.get(dcuList[i]).toString());
                                        String[] parseRemoteIp = global.dcuHashMap.get(dcuList[i].toString()).toString().split("/");
                                        String remoteIp = parseRemoteIp[0];
                                        String dcuId = parseRemoteIp[1];
                                        String meterId = parseRemoteIp[4];
                                        JSONArray meterList = global.dcuHashMap.get(dcuList[i]).getMeterInfo();

                                        for (int j = 0; j < meterList.size(); j++) {
                                                JSONArray meterIDarr = new JSONArray();
                                                meterIDarr.add(meterList.get(j));
                                                TcpClient tcpClient = new TcpClient(remoteIp, dcuId, meterIDarr, "ondemand");
                                                tcpClient.start();
                                        }
                                }
                        }
                } catch (InterruptedException e) {
                        e.printStackTrace();
                } finally {
                        System.out.println("OnDemand Thread is Dead");
                }
        }

}
```

수정 된 사항



목적:
Data Traffic & Event Response 실험을 위해
On-Demand Interval Time을 설정 할 수 있다.

기존 문제점:
기존에는 'SET' 버튼을 누를 때 마다
Thread가 추가적으로 생성된다.

해결방법:
Thread Interrupt 함수를 이용하여
보다 효과적으로 Thread를 관리한다.