

Zigbee

Interworking



Member



김상원

Computer Science 3rd Grade
Backend Developer (java)



안희찬(Leader)

Computer Science 3rd Grade
Backend Developer (java)

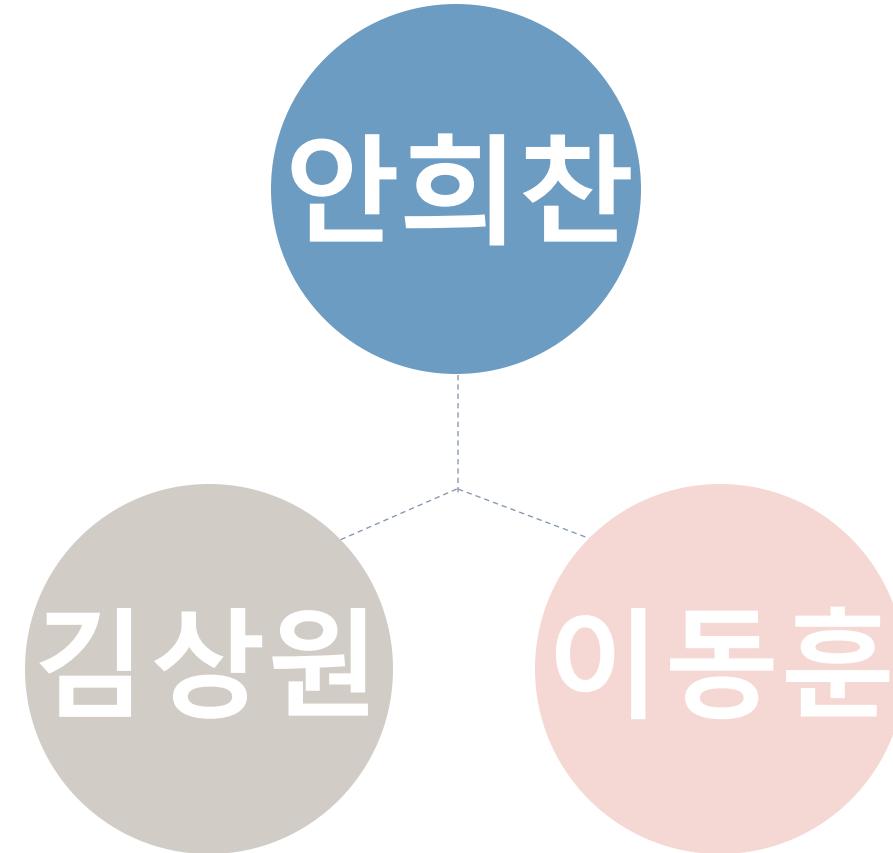


이동훈

Computer Science 3rd Grade
Backend Developer(java)

Role

1. Build oM2M project with our zigbee ipe
2. Make ipe core code such as router, LifeCycleManager and SpringLamp.
3. Refactoring our ipe source code and connect all function in our project



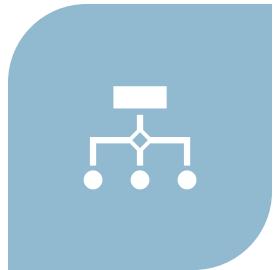
1. Analysis open source software deconz-rest-plugin
2. Code Http request/response between ipe and deconz
3. Raspberry Pi Initialization

1. Make Sensor Entity to connect with om2m
2. Make SensorModel and Controller to check the registration and Get response from URL
3. Make PPT

Part 1

Zigbee Interworking

Zigbee Interworking Overview



INTERWORKING
PROCESS BASED ON
ONEM2M



ZIGBEE DATA MAPPED
TO ONEM2M
RESOURCE



THE PROCESS TO APPLY
APPLICATIONS



IPE AND ZIGBEE
GATEWAY AND
DEVICES

Interworking Object

Produced zigbee IPE that works as an OM2M plug-in and easily communicates with Conbee through Http protocol

Register AE, Container, and ContentInstance in IPE & access in IN-CSE

Even if you don't know the details of how the inside of the oM2M works, you can manage and store zigbee data in the IN-CSE through IPE.

What are required to achieve this



STUDY OSGI



STUDY DECONZ



CODE IPE
BETWEEN DECONZ AND
OM2M

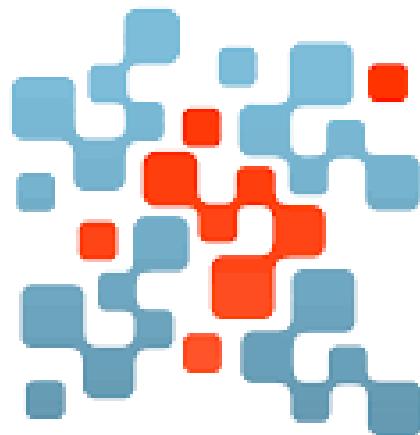


INSTALL DECONZ ON
RASPBERRY PI AND SET
UP

Part 2

Zigbee Interworking Details (Topic Analysis)

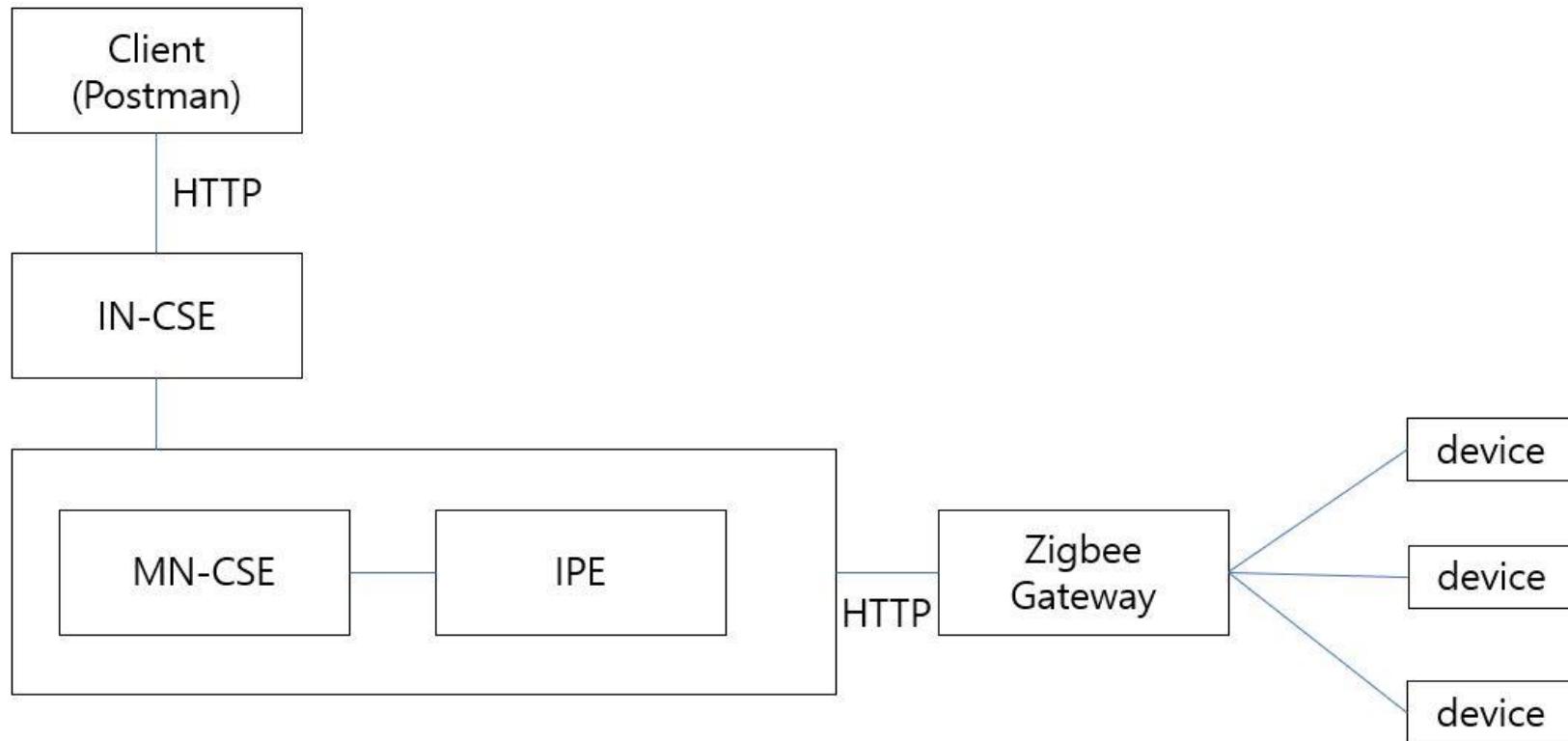
OneM2M Open Source



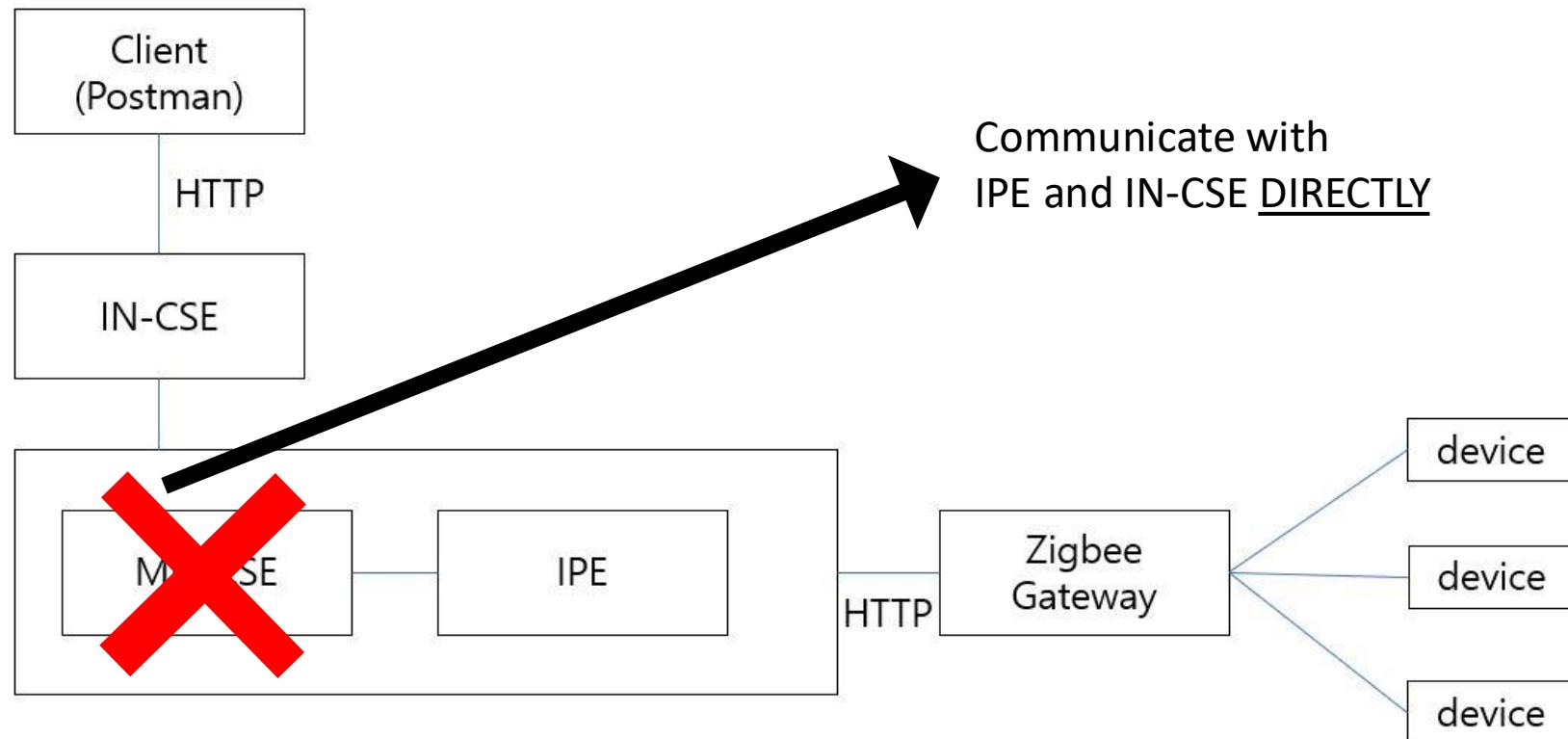
OM2M
Connecting things



Design With No MN-CSE



Design With No MN-CSE

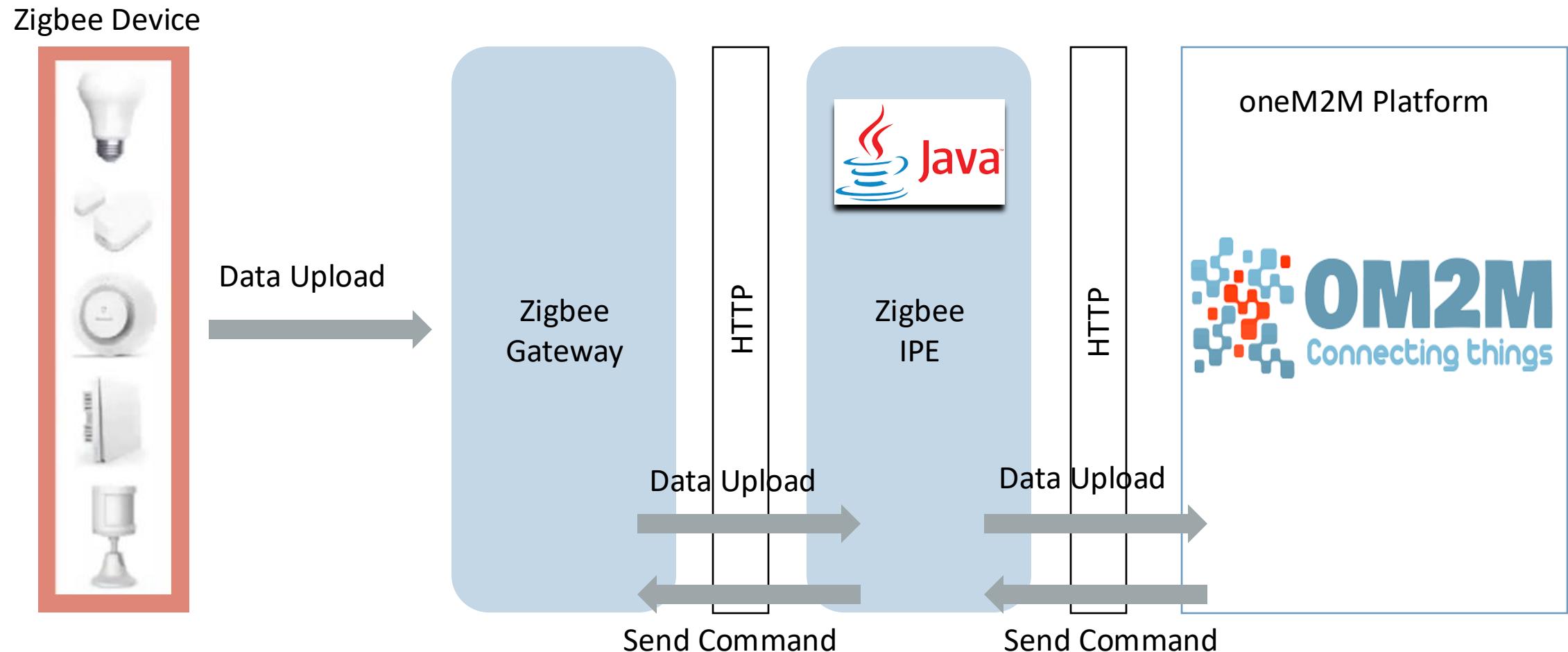


Our IPE(Interworking Proxy Entity) Design

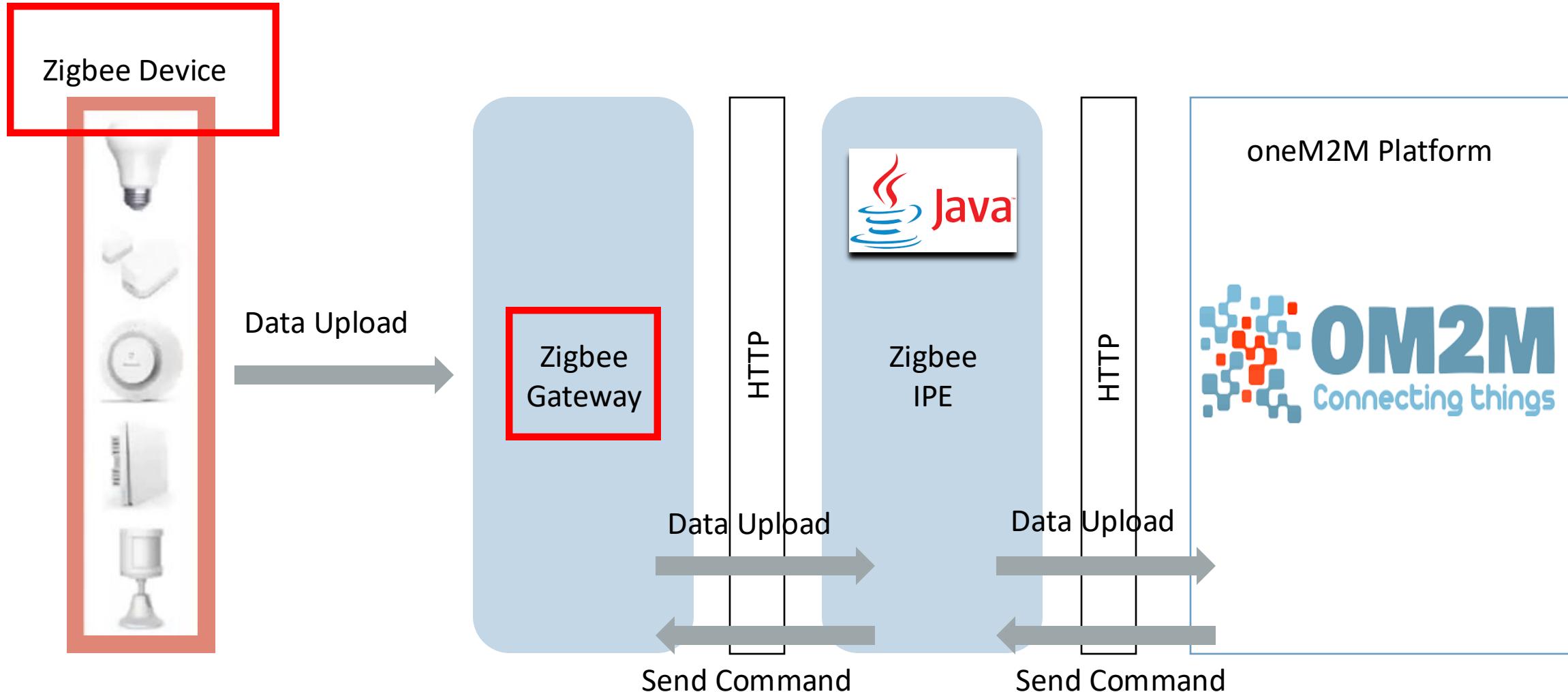
What is IPE?

Special AE that allows interact between oneM2M system and the others

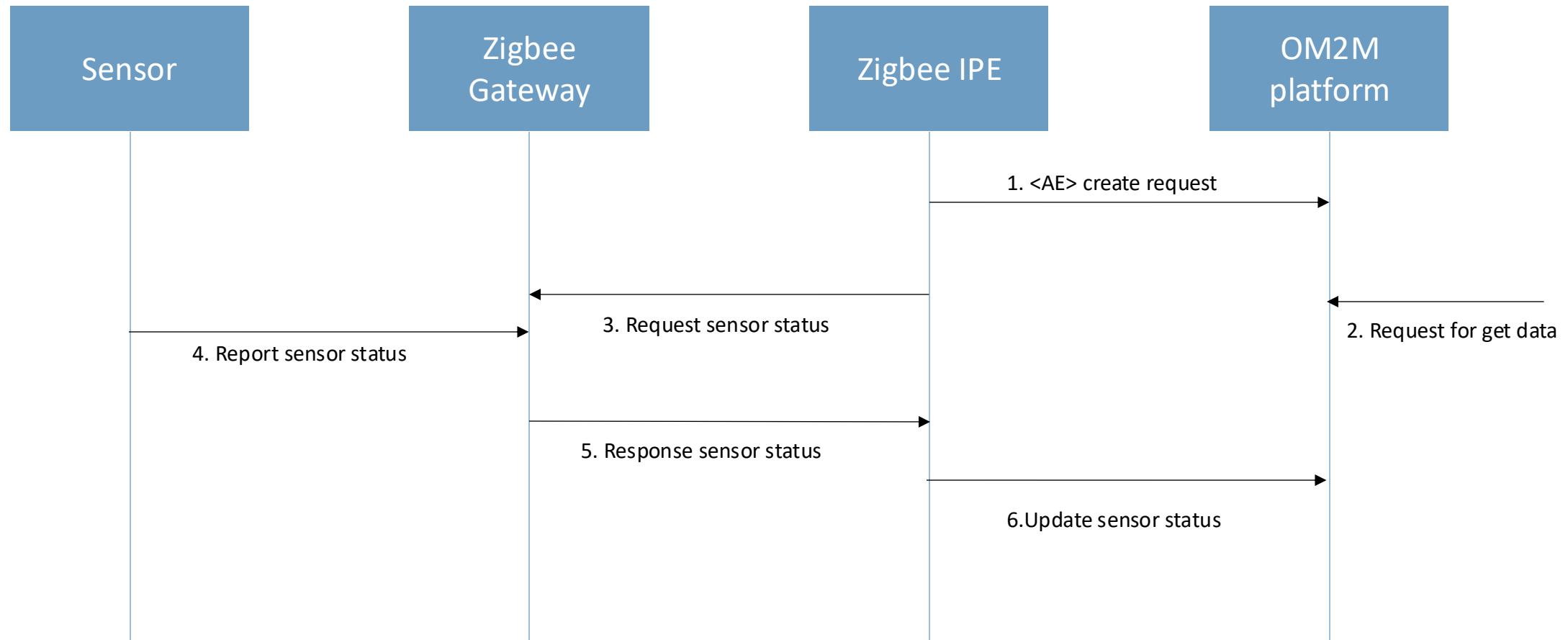
Our IPE(Interworking Proxy Entity) Design



Our IPE(Interworking Proxy Entity) Design



Interworking Scenario



{ MATERIALS }



Conbee 2

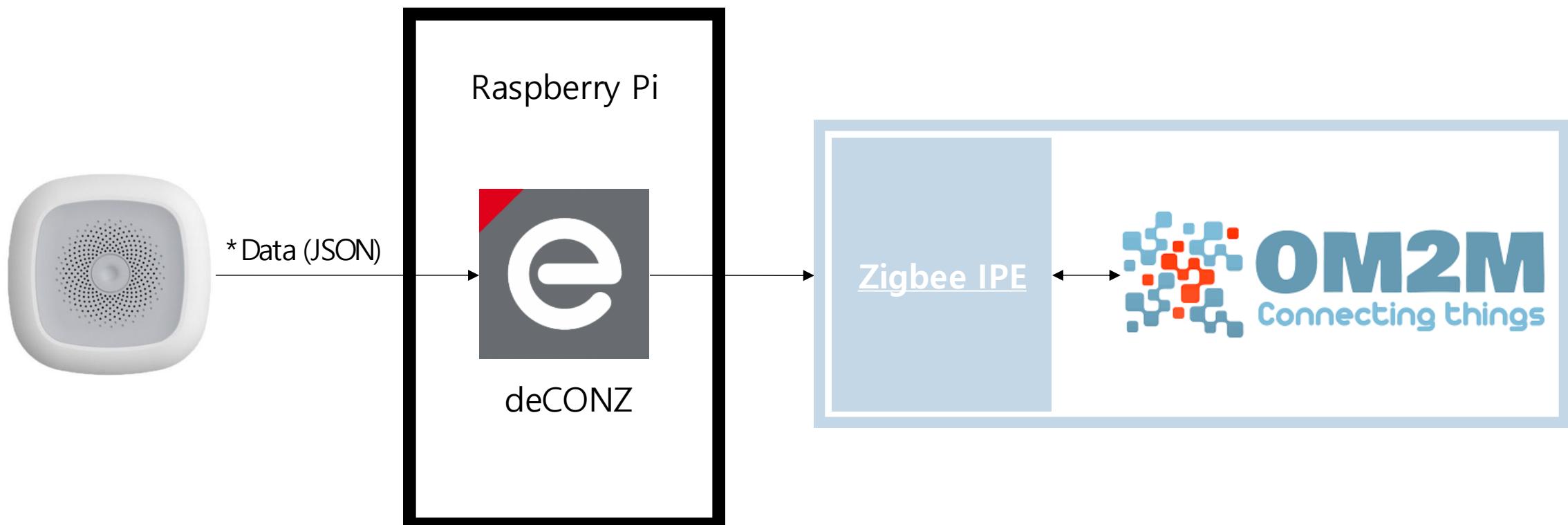


Raspberry Pi



Zigbee Sensor

Design Architecture with Materials



*Data : Temperature & Humidity

deCONZ : Gateway module of Zigbee



A software use to control and set up Zigbee network

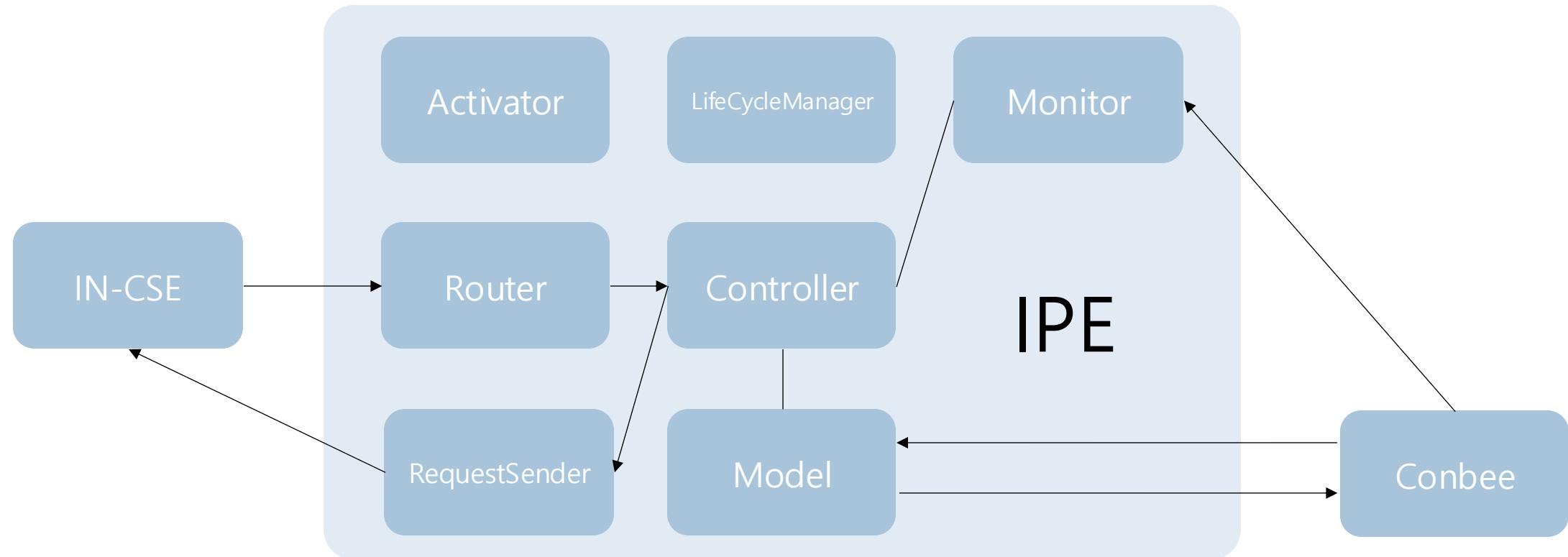


Communicate with the [ConBee/Raspberry pi](#) Zigbee gateway and expose the Zigbee devices connected to the gateway



Easy to use and requires no programming

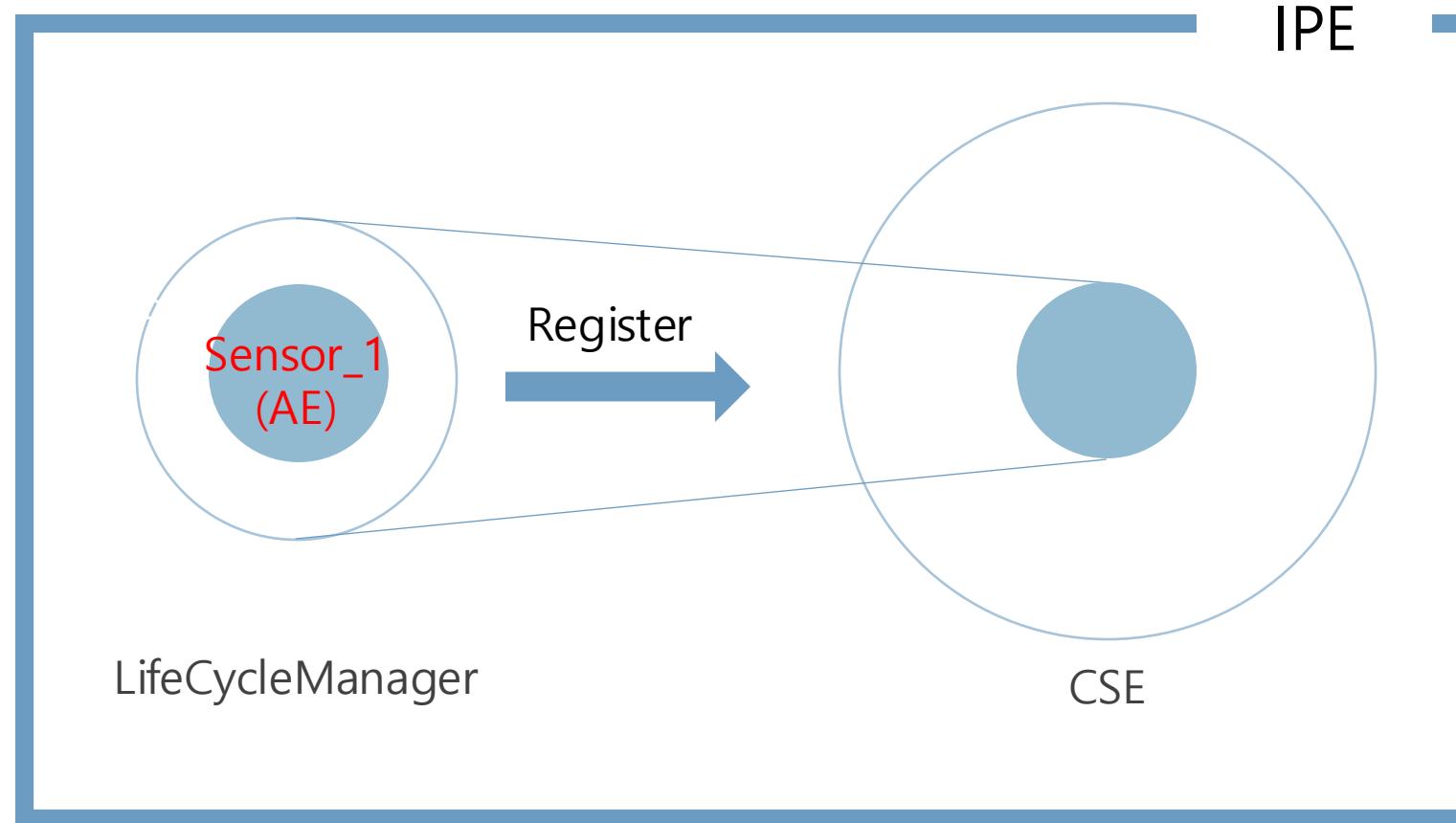
Device & OM2M Operation Process



Part 3

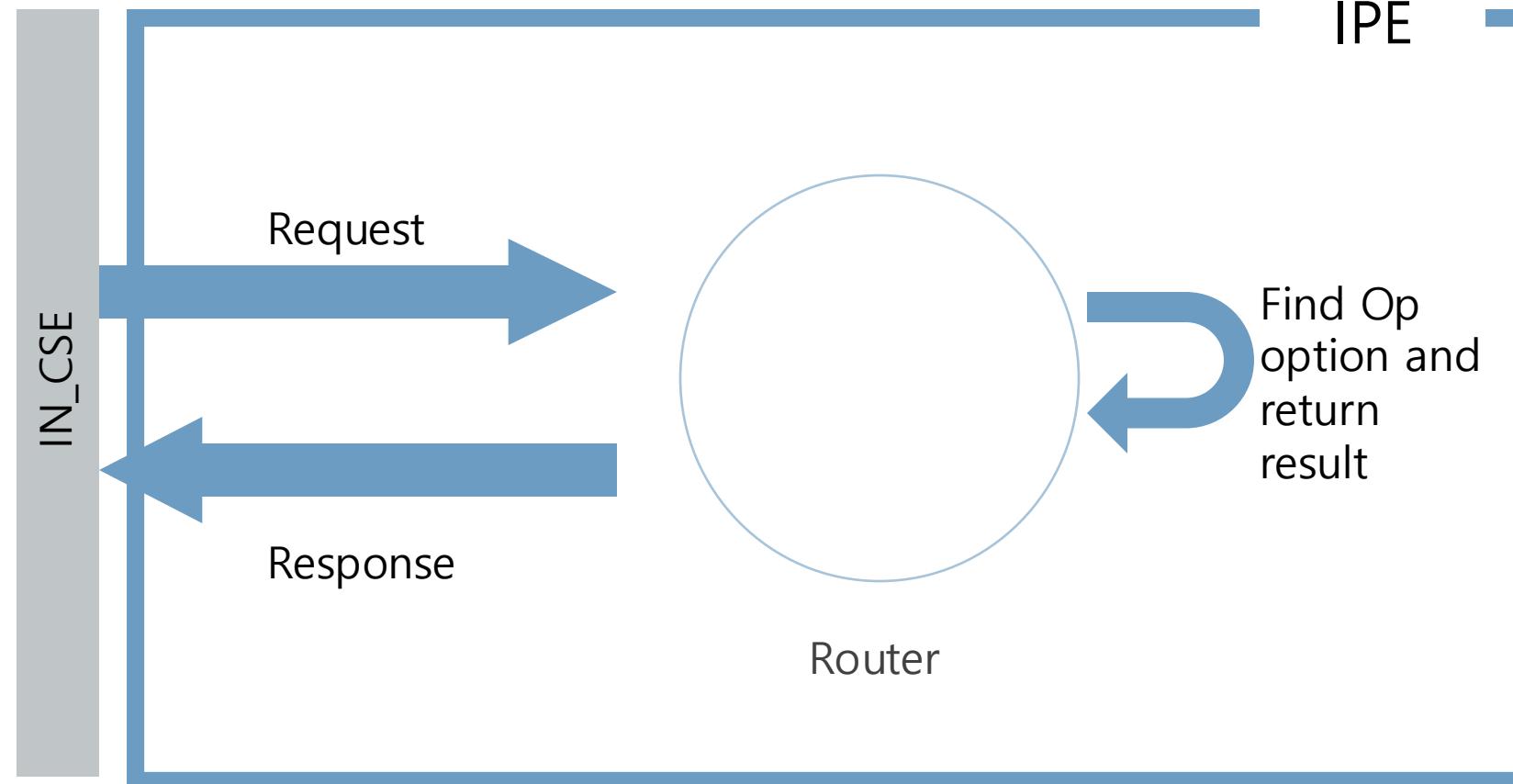
Interworking Process Implementation (Solution)

Process Sequence



1. LifeCycleManager activates and make a model of sensor
2. Register the AE to CSE

Process Sequence



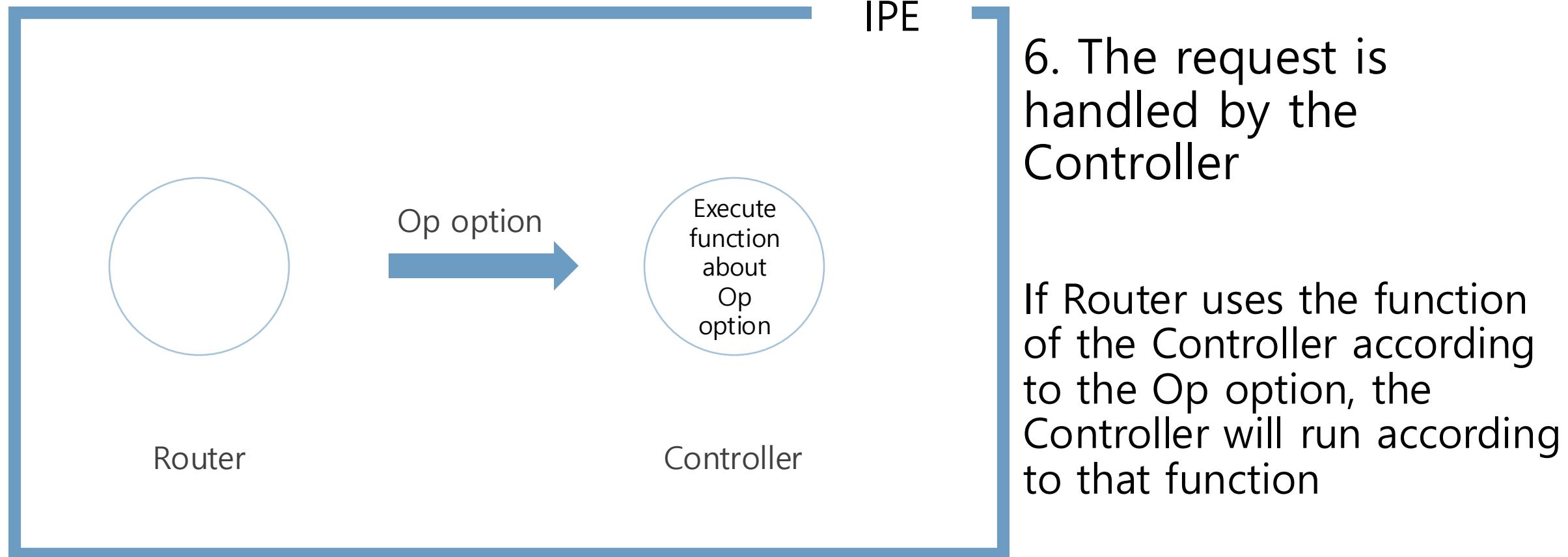
3. Router get the Request from IN-CSE
4. Find the proper Op option and process
5. After process, send response

Process Sequence

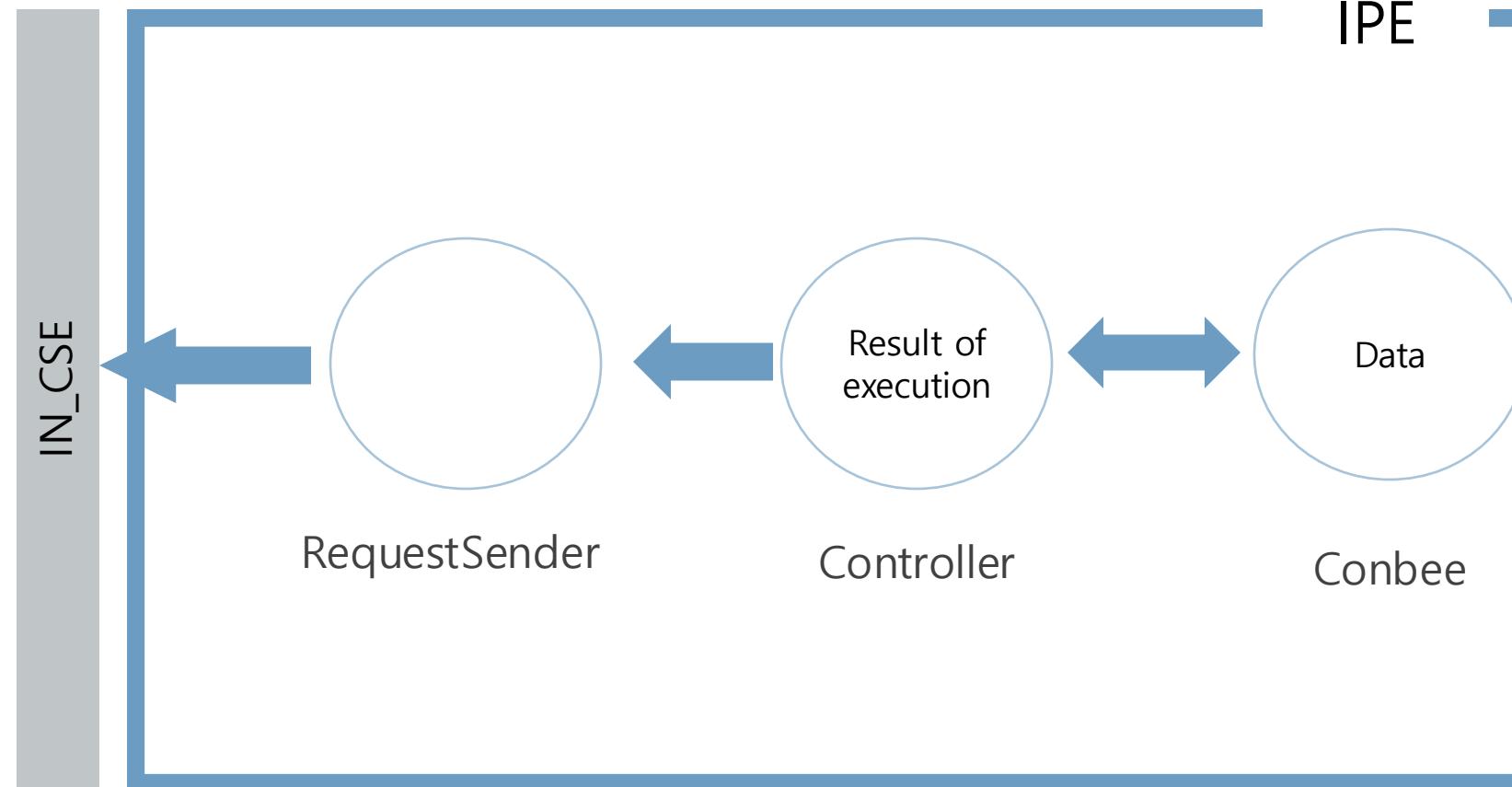
- Op option can be customized

```
public enum Operations {  
    GET_RECENT_STATE( value: "getRecentState"),  
    SET_ON( value: "setOn"),  
    SET_OFF( value: "setOff"),  
    GET_STATE_DIRECT( value: "getStateDirect"),  
    GET_SENSOR_STATE( value: "getSensorState");  
}
```

Process Sequence



Process Sequence



7. Controller gets the data from Conbee, the result of function executed
8. Reflect the result to IN-CSE

OM2M Modification

Get Conbee Address



```
public class ipeConfig {  
    6 usages  
    private static Map<String, String> ipeConfigs = new HashMap<~>();  
    static {  
        ipeConfigs.put("ipe", "127.0.0.1");  
        ipeConfigs.put("apiKey", "D9DF3B2018");  
        ipeConfigs.put("sensingInterval", "1000");  
        ipeConfigs.put("address", "172.19.23.232");  
        ipeConfigs.put("port", "8080");  
    }  
    2 usages  ↘ LeeDongHoon  
    public static String getConfig(String key) { return ipeConfigs.get(key); }  
}
```

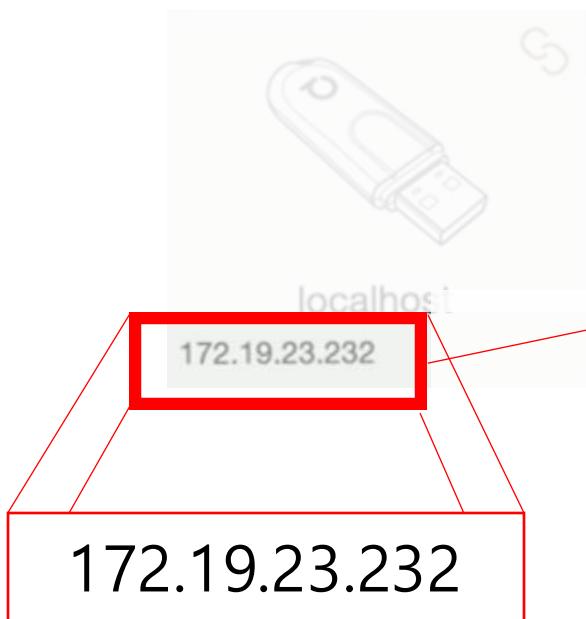
Interworking Process Implementation

Get Conbee Address



```
public class ipeConfig {  
    6 usages  
    private static Map<String, String> ipeConfigs = new HashMap<~>();  
    static {  
        ipeConfigs.put("ipe", "127.0.0.1");  
        ipeConfigs.put("apiKey", "D9DF3B2018");  
        ipeConfigs.put("sensingInterval", "1000");  
        ipeConfigs.put("address", "172.19.23.232");  
        ipeConfigs.put("port", "8080");  
    }  
    2 usages ↗ LeeDongHoon  
    public static String getConfig(String key) { return ipeConfigs.get(key); }  
}
```

Get Conbee Address



```
public class ipeConfig {  
    6 usages  
    private static Map<String, String> ipeConfigs = new HashMap<~>();  
    static {  
        ipeConfigs.put("ipe", "127.0.0.1");  
        ipeConfigs.put("apiKey", "D9DF3B2018");  
        ipeConfigs.put("sensingInterval", "1000");  
        ipeConfigs.put("address", "172.19.23.232")  
        ipeConfigs.put("port", "8080");  
    }  
    2 usages  ↳ LeeDongHoon  
    public static String getConfig(String key) { return ipeConfigs.get(key); }  
}
```

Interworking Process Implementation

Get Conbee API key

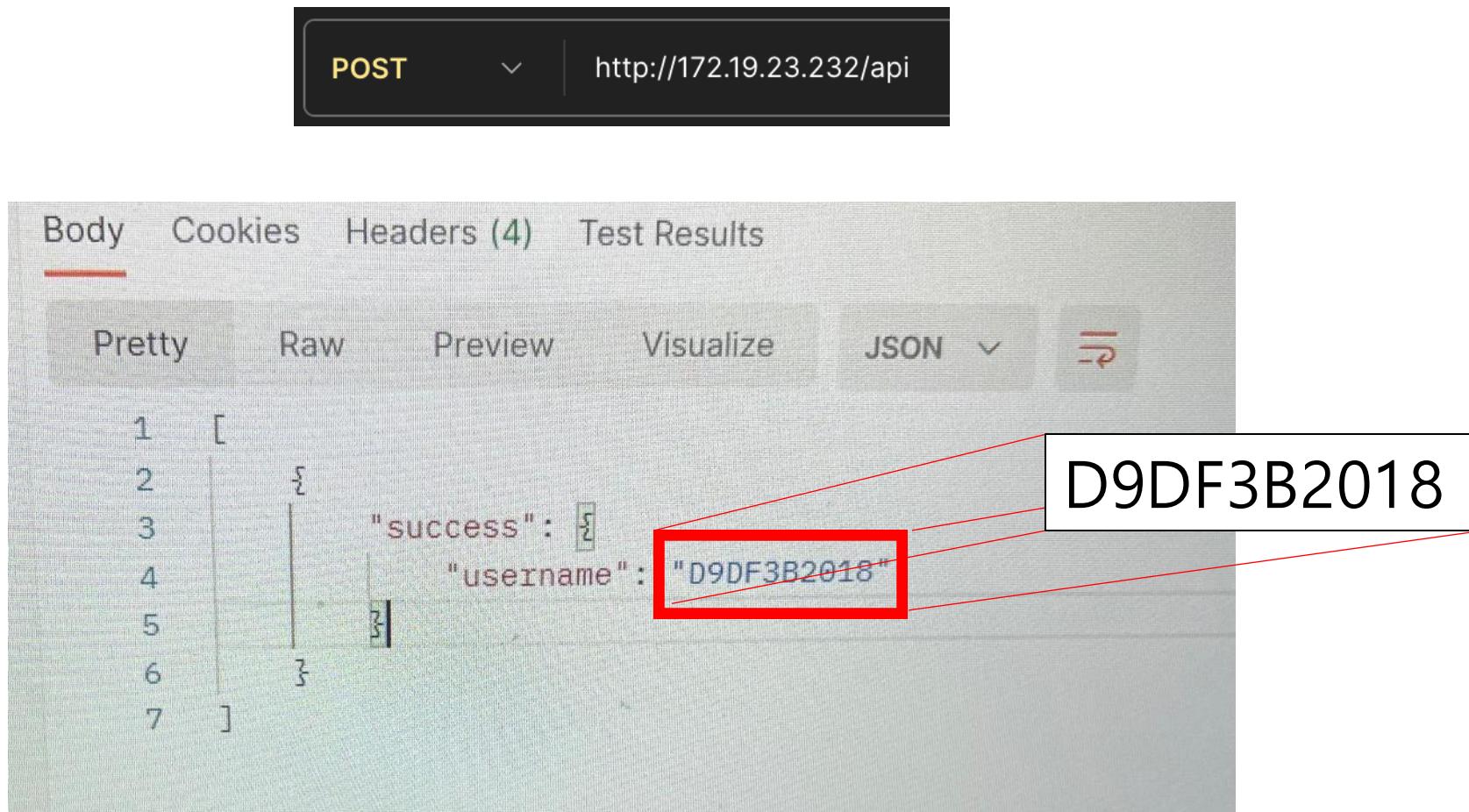
POST | http://172.19.23.232/api

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON ↻

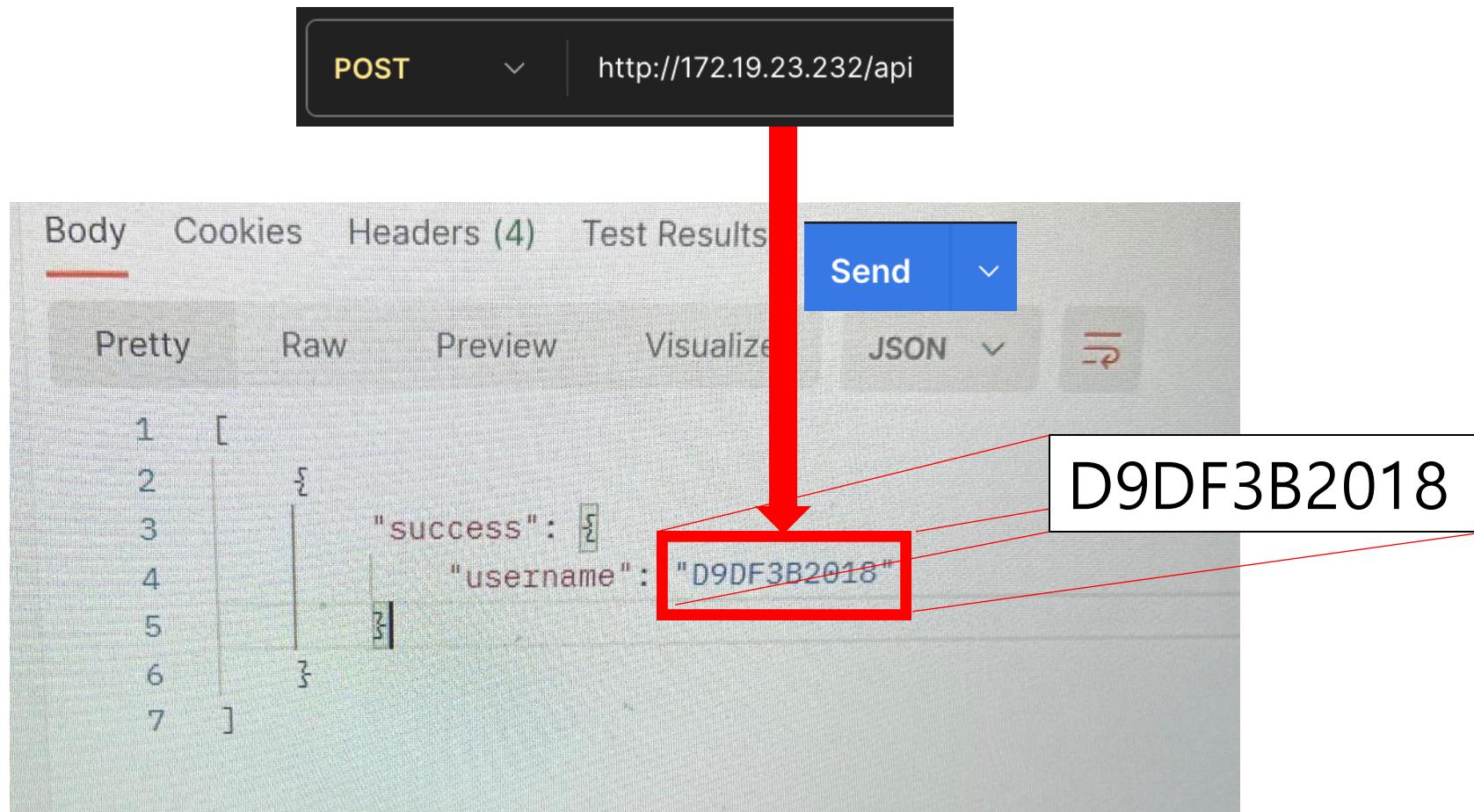
```
1 [  
2 {  
3   "success": {  
4     "username": "D9DF3B2018"  
5   }  
6 }  
7 ]
```

D9DF3B2018



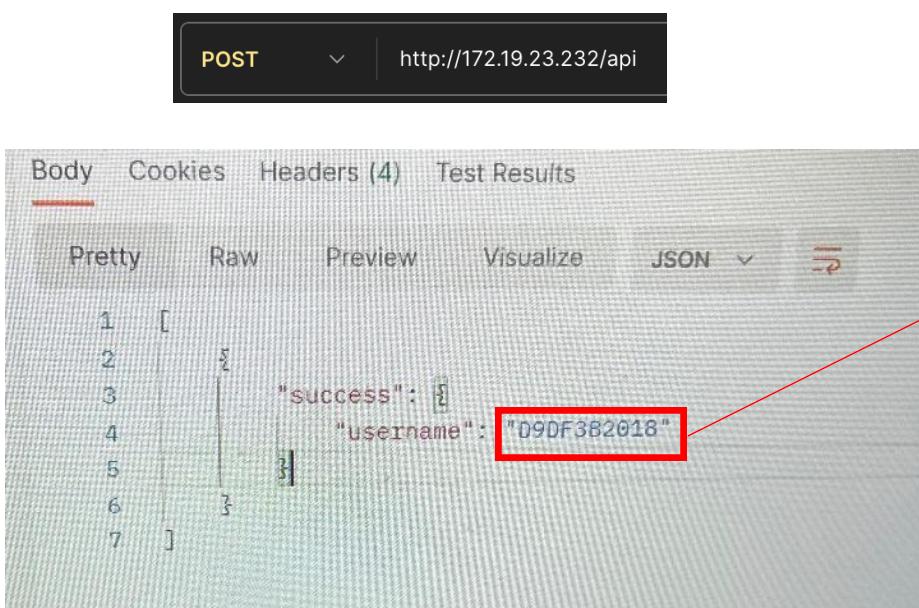
Interworking Process Implementation

Get Conbee API key



Interworking Process Implementation

Get Conbee API key



```
public class ipeConfig {  
    6 usages  
    private static Map<String, String> ipeConfigs = new HashMap<~>();  
    static {  
        ipeConfigs.put("ipe", "127.0.0.1");  
        ipeConfigs.put("apiKey", "D9DF3B2018");  
        ipeConfigs.put("sensingInterval", "1000");  
        ipeConfigs.put("address", "172.19.23.232");  
        ipeConfigs.put("port", "8080");  
    }  
    2 usages  ↳ LeeDongHoon  
    public static String getConfig(String key) { return ipeConfigs.get(key); }  
}
```

Interworking Process Implementation

HTTP Proxy

Connect OM2M
and device with
HTTP

```
public JsonNode connect(String DeviceName, String id, String state, String method, String json) throws Exception {
    this.id = id;
    this.state = state;
    this.method = method;
    this.DeviceName = DeviceName;

    my_url = my_url + "/" + DeviceName;
    if (this.id != null)
        my_url = my_url + "/" + id;
    if (this.state != null)
        my_url = my_url + "/" + state;

    URL url = new URL(my_url);

    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    this.method = method;
    connection.setRequestMethod(this.method);
    if (this.method.equals("POST") || this.method.equals("PUT")) {
        connection.setRequestProperty("Content-Type", "application/json");
        connection.setDoOutput(true);
        // POST 데이터 설정 (JSON 형태의 데이터 예시)
        // String postData = "{\"key1\":\"value1\", \"key2\":\"value2\"}";
        String postData = json;
        try (DataOutputStream wr = new DataOutputStream(connection.getOutputStream())) {
            wr.write(postData.getBytes(charsetName: "UTF-8"));
        }
    }
    int responseCode = connection.getResponseCode();

    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(connection.getInputStream()));
    StringBuffer stringBuffer = new StringBuffer();
    String line;
    while ((line = bufferedReader.readLine()) != null) {
        stringBuffer.append(line);
    }
    bufferedReader.close();
    return parseJson(stringBuffer.toString());
}
```

Interworking Process Implementation

Design IPE (LifeCycleManager)

Register AE, Container, and ContentInstance in IPE and access them in IN-CSE

```
private static void createDeviceResources(String appId, String poa) {  
  
    Container container = new Container();  
    container.getLabels().add("sensor");  
    container.setMaxNrOfInstances(BigInteger.valueOf(0));  
  
    AE ae = new AE();  
    ae.setRequestReachability(true);  
    ae.getPointOfAccess().add(poa);  
    ae.setAppID(appId);  
    ae.setName(appId);  
  
    // CSE에 AE 등록되어 있는지 확인  
    ResponsePrimitive response = RequestSender.createAE(ae);  
  
    // CSE 최초 등록이면 아래 실행  
    if(response.getResponseStatusCode().equals(ResponseStatusCode.CREATED)) {  
        container = new Container();  
        container.setMaxNrOfInstances(BigInteger.valueOf(10));  
  
        container.setName(SampleConstants.DESC);  
        RequestSender.createContainer(response.getLocation(), container);  
  
        container.setName(SampleConstants.DATA);  
        RequestSender.createContainer(response.getLocation(), container);  
  
        String content;
```

Interworking Process Implementation

Design IPE (Sensor Entity)

```
public class Sensor {  
  
    /** Device ID */  
    3 usages  
    private String deviceId;  
  
    /** Default Device location */  
    2 usages  
    public final static String LOCATION = "Home";  
    /** Default Device type */  
    3 usages  
    public final static String TYPE = "SENSOR";  
  
    /** init state of temperature & humidity */  
    2 usages  
    public static String temperature = "36.5" ;  
    2 usages  
    public static String humidity = "60";
```

Interworking Process Implementation

Design IPE (Model)

1. Check if sensor is registered in CSE
2. Set data from the sensor to sensor entity



```
public SensorModel() {  
}  
5 usages  
private static Sensor DEVICE = null;  
  
/**  
 * Sets the device state.  
 * @param deviceId - Application ID  
 */  
1 usage new *  
public static void setDeviceState(final String deviceId, String temperature, String humidity) {  
    checkSensorIdValue(deviceId);  
    DEVICE.setTemperature(temperature);  
    DEVICE.setHumidity(humidity);  
}  
  
1 usage new *  
public static void checkSensorIdValue(String lampId){  
    if(lampId == null || !DEVICE.getDeviceId().equals(lampId)){  
        throw new BadRequestException(lampId+" error");  
    }  
}  
  
no usages new *  
public static Sensor getDEVICE() { return DEVICE; }  
1 usage new *  
public static void setSensor(Sensor sensor) { DEVICE = sensor; }
```

Interworking Process Implementation

Design IPE(Router)

Read the query and choose the service what client wants

- GET_SENSOR_STATE
- GET_RECENT_STATE

```
@Override
public ResponsePrimitive doExecute(RequestPrimitive request) {
    ResponsePrimitive response = new ResponsePrimitive(request);
    if(request.getQueryStrings().containsKey("op")){
        String operation = request.getQueryStrings().get("op").get(0);
        Operations op = Operations.getOperationFromString(operation);
        String deviceId= null;
        if(request.getQueryStrings().containsKey("deviceId")){
            deviceId = request.getQueryStrings().get("deviceId").get(0);
        }
        try{
            switch(op){
                case GET_STATE_DIRECT:
                    String content1 = Controller.getFormatedLampState(deviceId);
                    response.setContent(content1);
                    request.setReturnContentType(MimeMediaType.OBIX);
                    response.setResponseStatusCode(ResponseStatusCode.OK);
                    break;
                case GET_SENSOR_STATE:
                    Controller.setSensorState(deviceId);
                    response.setResponseStatusCode(ResponseStatusCode.OK);
                    break;
                case GET_RECENT_STATE:
                    String content2 = Controller.getFormatedDeviceState(deviceId);
                    response.setContent(content2);
                    request.setReturnContentType(MimeMediaType.OBIX);
                    response.setResponseStatusCode(ResponseStatusCode.OK);
                    break;
                default:
                    throw new BadRequestException();
            }
        } catch (Exception e) {
            response.setError(e.getMessage());
            response.setResponseStatusCode(ResponseStatusCode.INTERNAL_ERROR);
        }
    }
}
```

Interworking Process Implementation

Design IPE(Controller)

Get response from
`http://{conbee_ip_adress}/api/{conbee_api}/sensors`

Set sensor entity with
data from sensors

Make a ContentInstance
and request to
RequestSender

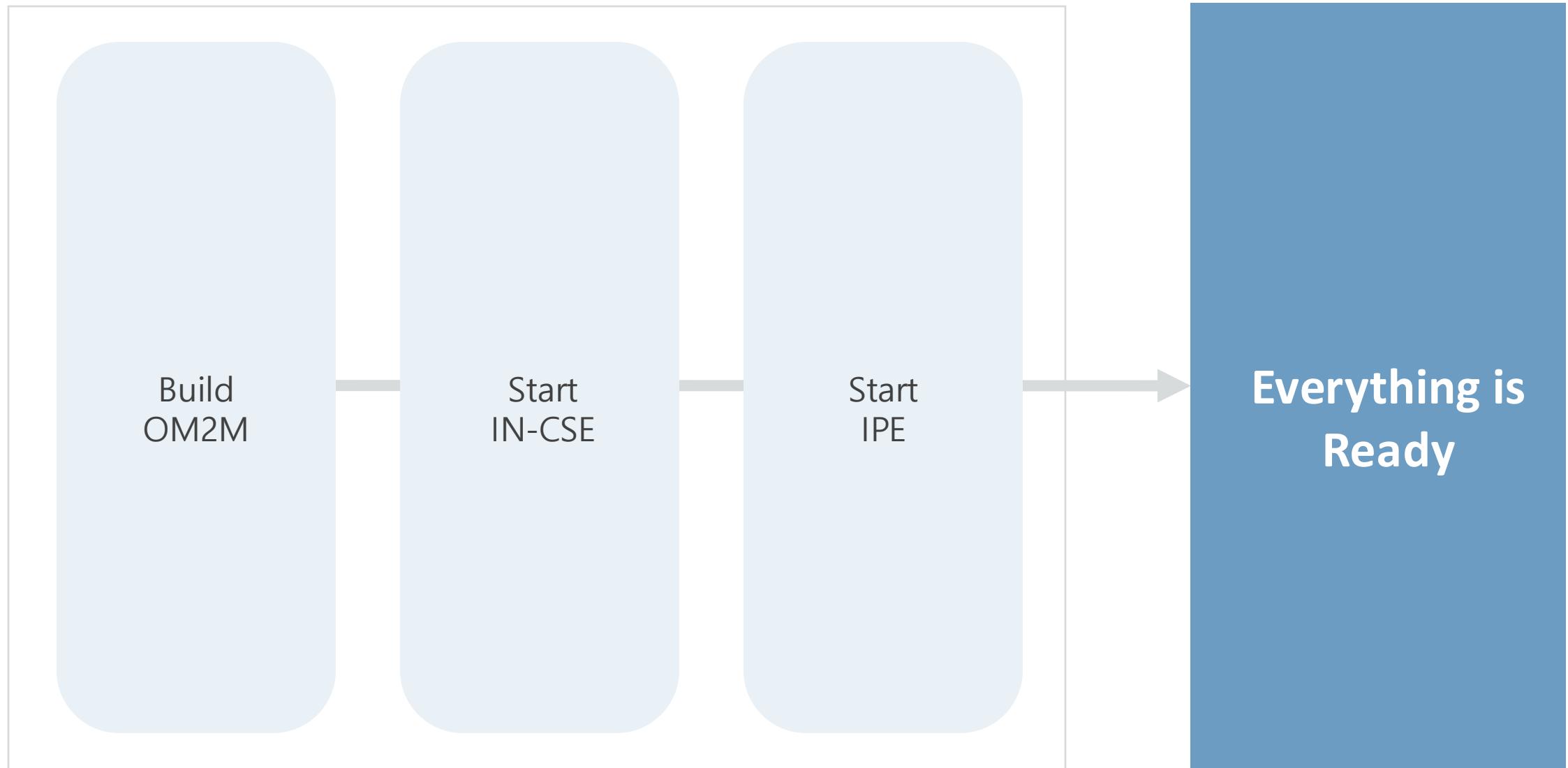
```
public static void setSensorState(String deviceId) throws Exception{
    HttpProxy proxy = new HttpProxy();
    // 1. 데이터 받아오기
    JsonNode response = proxy.connect(DeviceName: "sensors", id: null, state: null, method: "GET", json: null);
    Double rowTemp = response.get("2").get("state").get("temperature").doubleValue() / 100;
    Double rowHum = response.get("3").get("state").get("humidity").doubleValue() / 100;

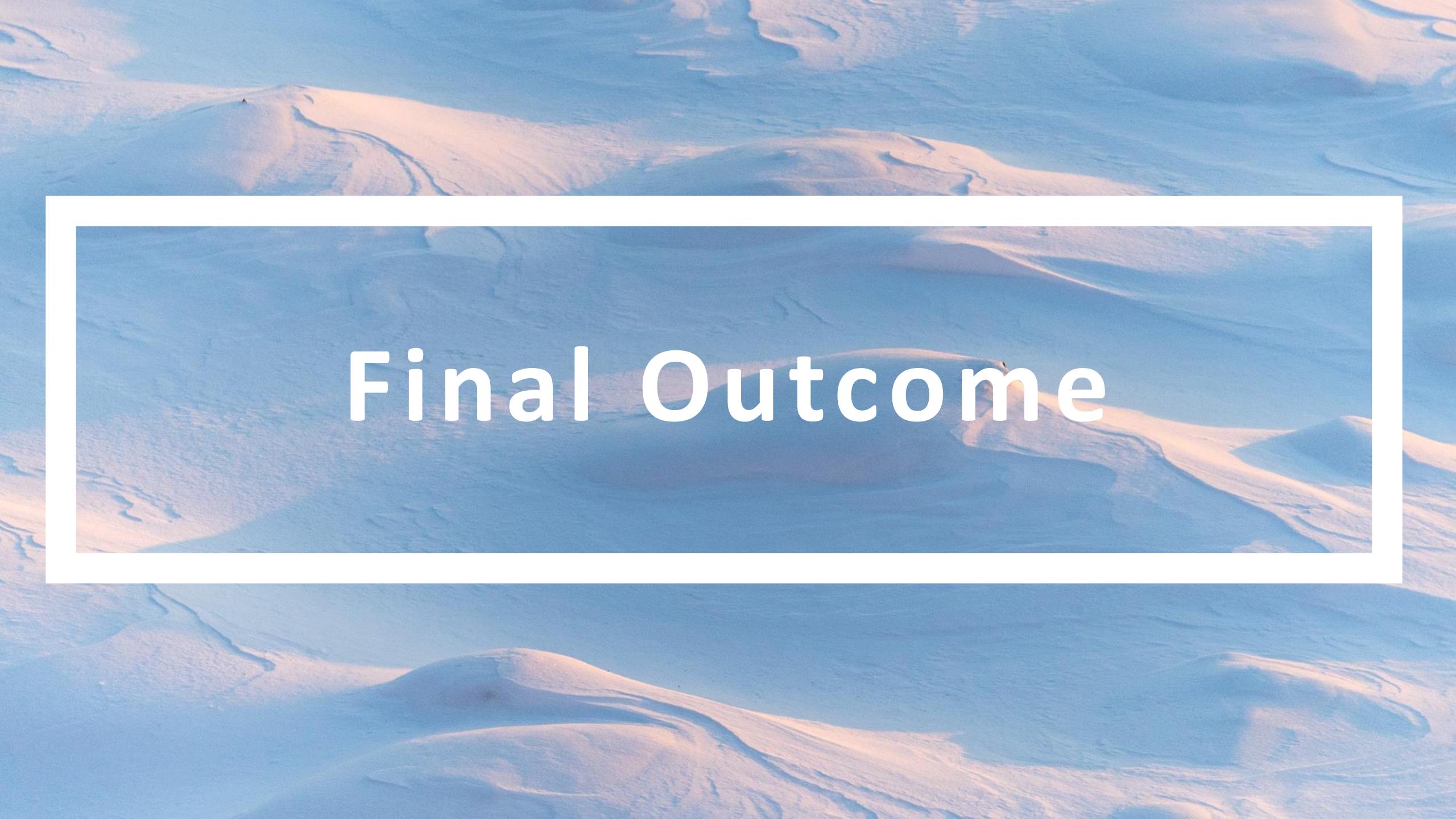
    // 2. SensorModel에 저장하기
    SensorModel.setDeviceState(deviceId, String.valueOf(rowTemp), String.valueOf(rowHum));

    // 3. ContentInstance 만들고 RequestSender로 IN-CSE에 요청보내기
    String targetID = SampleConstants.CSE_PREFIX + "/" + deviceId + "/" + SampleConstants.DATA;
    ContentInstance cin = new ContentInstance();
    cin.setContent(ObixUtil.getSensorState(deviceId)); // 받아온 정보 이용해야 함
    cin.setContentInfo(MimeMediaType.OBIX + ":" + MimeMediaType.ENCOD_PLAIN);
    RequestSender.createContentInstance(targetID, cin);

    1 usage  new *
    public static String getFormatedDeviceState(String deviceId) { return ObixUtil.getSensorState(deviceId); }
```

Interworking Process Implementation



The background of the image is a wide-angle photograph of a desert landscape. The terrain is covered in light-colored sand dunes that stretch across the frame. The sky above is a clear, pale blue. In the center-left area of the image, there is a white rectangular box containing the text.

Final Outcome

```
[INFO] org.eclipse.om2m.sdt.home.mocked.devices ..... SUCCESS [ 0.236 s]
[INFO] org.eclipse.om2m.sdt.home.utils ..... SUCCESS [ 0.242 s]
[INFO] org.eclipse.om2m.sdt.home.applications ..... SUCCESS [ 0.003 s]
[INFO] Authentication service for Home Monitoring ..... SUCCESS [ 0.110 s]
[INFO] org.eclipse.om2m.sdt.home.monitoring ..... SUCCESS [ 0.264 s]
[INFO] org.eclipse.om2m :: IPE SDT ..... SUCCESS [ 0.101 s]
[INFO] org.eclipse.om2m :: SDT IPE sample ..... SUCCESS [ 0.109 s]
[INFO] org.eclipse.om2m :: IPE SDT Test Suite ..... SUCCESS [ 0.134 s]
[INFO] org.eclipse.om2m :: sdt comparator xml ..... SUCCESS [ 0.215 s]
[INFO] org.eclipse.om2m :: asn product ..... SUCCESS [ 3.157 s]
[INFO] org.eclipse.om2m :: dal ..... SUCCESS [ 0.017 s]
[INFO] org.eclipse.om2m :: dal driver sample ..... SUCCESS [ 0.139 s]
[INFO] org.eclipse.om2m :: ipe dal ..... SUCCESS [ 0.081 s]
[INFO] org.eclipse.om2m :: in product ..... SUCCESS [ 2.872 s]
[INFO] org.eclipse.om2m :: mn product ..... SUCCESS [ 2.688 s]
[INFO] org.eclipse.om2m :: Orange's p2 repo ..... SUCCESS [ 1.890 s]
```

```
[INFO] -----
```

```
[INFO] BUILD SUCCESS
```

```
[INFO] -----
```

```
[INFO] Total time: 56.393 s
```

```
[INFO] Finished at: 2023-12-06T09:44:46+09:00
```

```
[INFO] -----
```

```
Leedonghoon@idonghun-ui-MacBookAir org.eclipse.om2m % org.eclipse.om2m.site.in-cse/target/products/in-cse
```

```
zsh: permission denied: org.eclipse.om2m.site.in-cse/target/products/in-cse
```

```
Leedonghoon@idonghun-ui-MacBookAir org.eclipse.om2m % cd org.eclipse.om2m.site.in-cse/target/products/in-cse
```

```
Leedonghoon@idonghun-ui-MacBookAir in-cse % cd macosx
```

```
Leedonghoon@idonghun-ui-MacBookAir macosx % cd carbon
```

```
Leedonghoon@idonghun-ui-MacBookAir carbon % cd aarch64
```

```
Leedonghoon@idonghun-ui-MacBookAir aarch64 % sh start.sh
```

Part 4

Git & Future Plan

The background of the image is a wide-angle photograph of a desolate, snow-covered terrain. The landscape features numerous low, rounded hills covered in white snow, with dark, rocky outcrops visible in some areas. The sky above is a clear, pale blue, suggesting a cold, possibly arctic environment. A thin white rectangular border frames the central text area.

Git

<https://github.com/oM2M-Zigbee-IPE>

| |
|---|
| Readme 사용법 정리 |
| HeeChanN committed 3 days ago |
| [FIX] 디렉토리명 수정 및 빌드 정보 수정 |
| HeeChanN committed 3 days ago |
| [ADD] Conbee에서 센서 데이터 값 가져오기 및 오류 수정 |
| HeeChanN committed 3 days ago |
| Json 노드 response 파악 전까지 완료 |
| HeeChanN committed 3 days ago |
| -o Commits on Dec 3, 2023 |
| Conbee 통신 전 코드 |
| HeeChanN committed 4 days ago |
| Router 설정 및 센서 경로 설정 |
| HeeChanN committed 4 days ago |
| -o Commits on Dec 2, 2023 |
| readme 수정 |
| HeeChanN committed 5 days ago |
| almost done |
| daydream-er committed 5 days ago |
| -o Commits on Nov 29, 2023 |
| ObixUtil getStateRep 수정 |
| LEEDONGHOON committed last week |

| |
|---|
| connect -> static |
| levelmax14 committed last week |
| HttpProxy, RequestMapping, ipeConfig |
| levelmax14 committed last week |
| feat : Controller toggleDevice 추가 |
| LEEDONGHOON committed last week |
| [ADD] Spring Lamp test |
| HeeChanN committed last week |
| Commits on Nov 27, 2023 |
| 이동훈 수정-2 |
| LEEDONGHOON committed last week |
| model device 수정-1 |
| LEEDONGHOON committed last week |
| Commits on Nov 22, 2023 |
| Delete src/main/resources/images directory |
| HeeChanN committed 2 weeks ago |
| [ADD]:Basic IPE source code |
| levelmax14 committed last week |

| |
|---|
| ObixUtil getStateRep 수정 |
| LEEDONGHOON committed last week |
| Device 설정 |
| LEEDONGHOON committed last week |
| build용 IPE |
| LEEDONGHOON committed last week |
| feat : Lamp용 함수 추가 |
| LEEDONGHOON committed last week |
| [Fix] HttpProxy 문법 수정 및 Lamp 데이터 연결 |
| HeeChanN committed last week |
| Commits on Nov 28, 2023 |
| 범용적인 connect.... |
| levelmax14 committed last week |
| [ADD] Make Router and ObixUtil |
| HeeChanN committed last week |
| HttpProxy Error fixed |
| levelmax14 committed last week |
| connect -> static |
| levelmax14 committed last week |
| HttpProxy, RequestMapping, ipeConfig |
| levelmax14 committed last week |

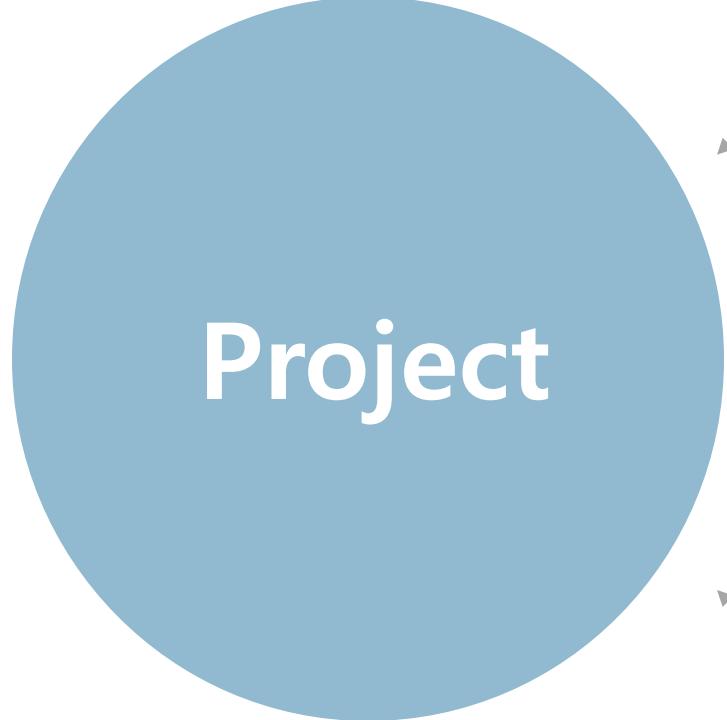


NEXT STEP

Git & Future Plan

Use other devices
and connect to our project

make our project
more concise



Project

Deep Dive into OM2M

Communicate with
other Protocol (MQTT)



THANK YOU