Indoor fire hydrant
Obstacle Detection System

Team : Craft

TTA Sejong University oneM2M KETI Korea Electronics Technology Institute ETSI INDICO ICT standards
InDiCo is an EU-funded project

Contents

01

Project Overview

Project Overview

- Team name -
Craft

- Team member -
Information Security, 17011621 윤승구
(Individual)

Project Overview

Indoor fire hydrant

> Designed to put out fires
> early with strong water pressure.

Project Overview

소방시설 설치·유지 및 안전관리에 관한 법률

일부개정 2014. 12. 30. [법률 제12939호, 시행 2015. 7. 1.] 소방청

제10조(피난시설, 방화구획 및 방화시설의 유지·관리)

① 특정소방대상물의 관계인은 「건축법」 제49조에 따른 피난시설, 방화구획(방화구획) 및 같은 법 제50조부터 제53조까지의 규정에 따른 방화벽, 내부 마감재료 등(이하 "방화시설"이라 한다)에 대하여 다음 각 호의 행위를 하여서는 아니 된다.

1. 피난시설, 방화구획 및 방화시설을 폐쇄하거나 훼손하는 등의 행위
2. 피난시설, 방화구획 및 방화시설의 주위에 물건을 쌓아두거나 장애물을 설치하는 행위
3. 피난시설, 방화구획 및 방화시설의 용도에 장애를 주거나 「소방기본법」 제16조에 따른 소방활동에 지장을 주는 행위
4. 그 밖에 피난시설, 방화구획 및 방화시설을 변경하는 행위

② 소방본부장이나 소방서장은 특정소방대상물의 관계인이 제1항 각 호의 행위를 한 경우에는 피난시설, 방화구획 및 방화시설의 유지·관리를 위하여 필요한 조치를 명할 수 있다.

Under fire-fighting legislation, there should be no obstacles around fire-fighting facilities.

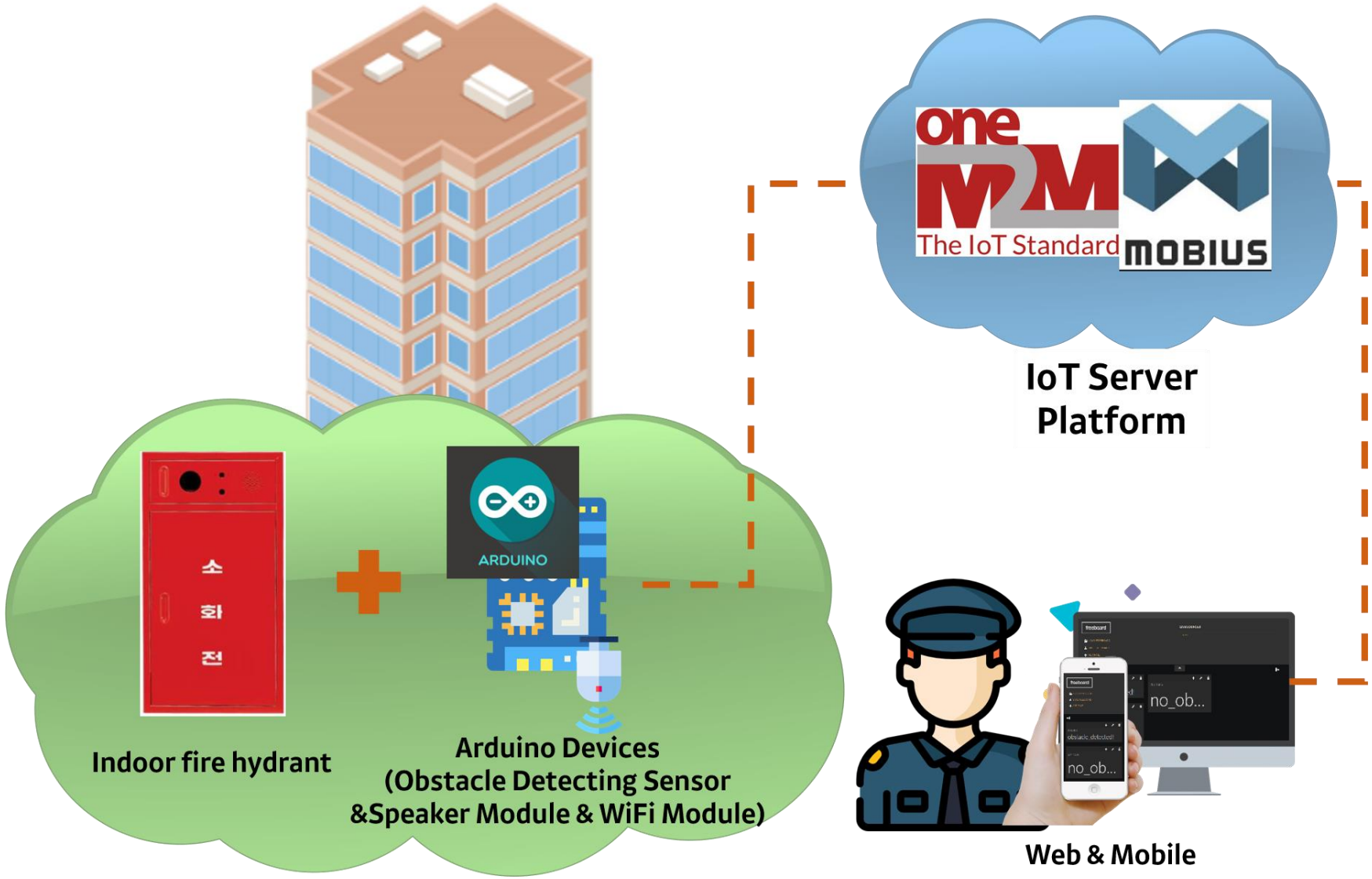
Project Overview

How about a system that can detect and manage obstacles
by attaching IoT sensors to an indoor fire hydrant?

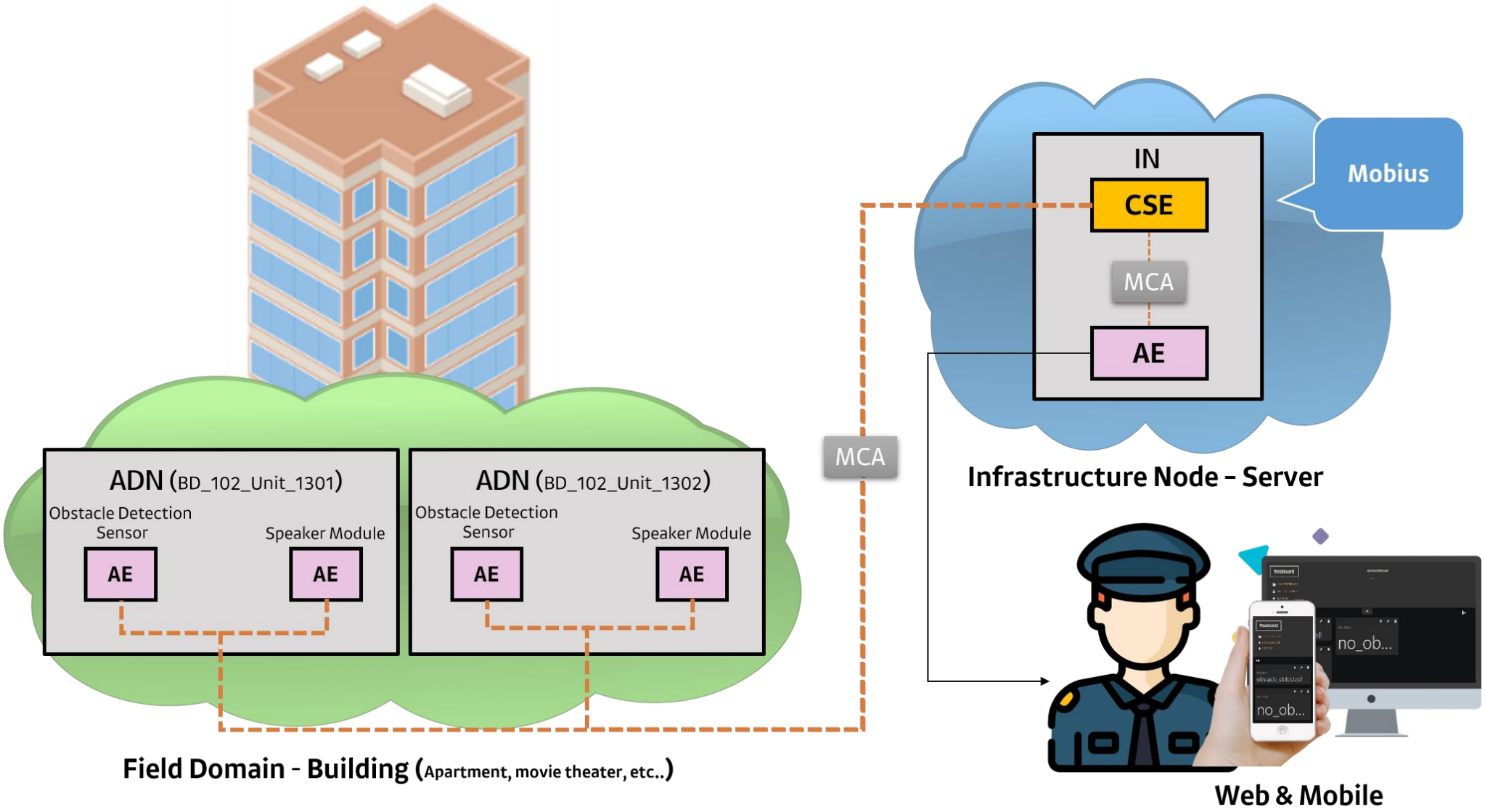
02

Use Case & Architecture

Use Case


oneM2M
The IoT Standard
MOBIUS
IoT Server Platform
ARDUINO
소
화
전
Indoor fire hydrant
Arduino Devices
(Obstacle Detecting Sensor
&Speaker Module & WiFi Module)
Web & Mobile

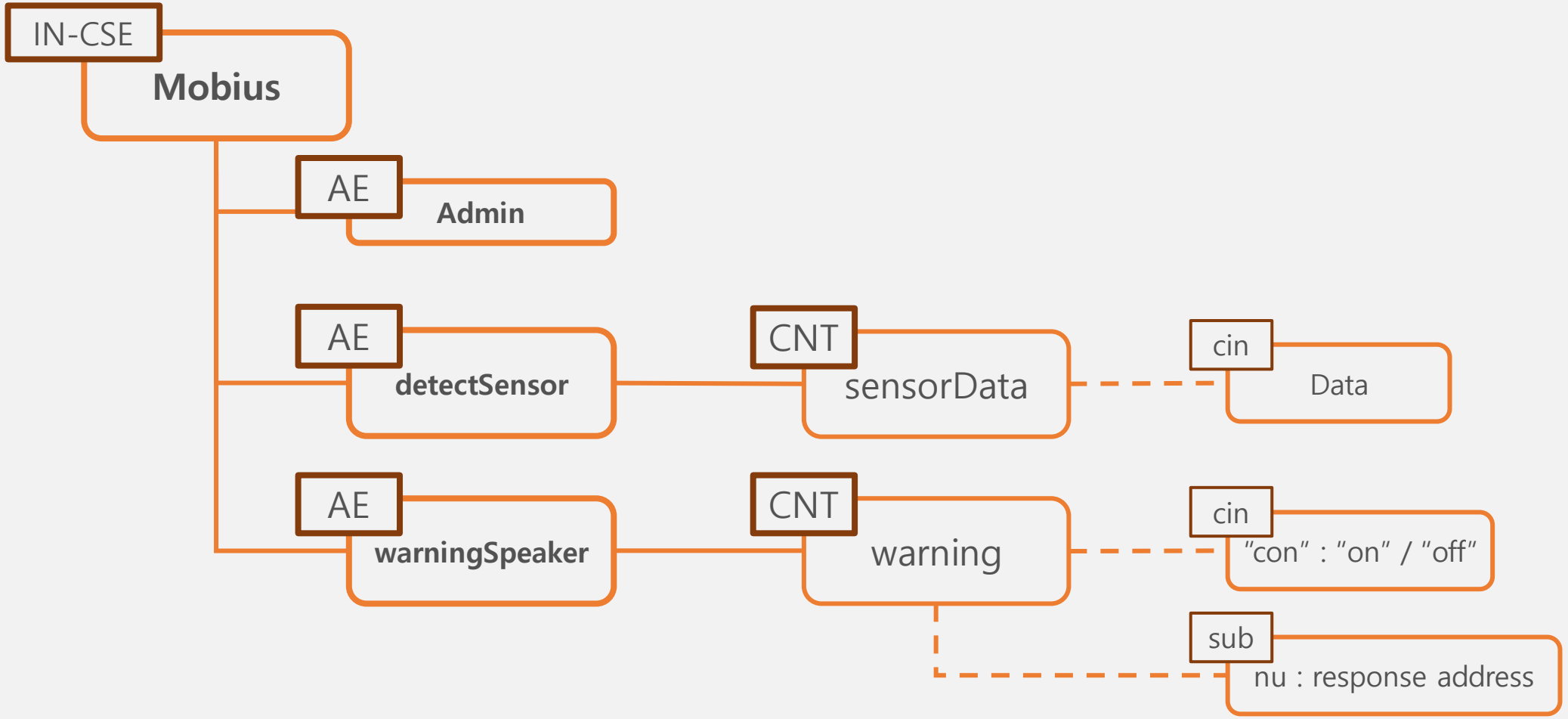
Architecture



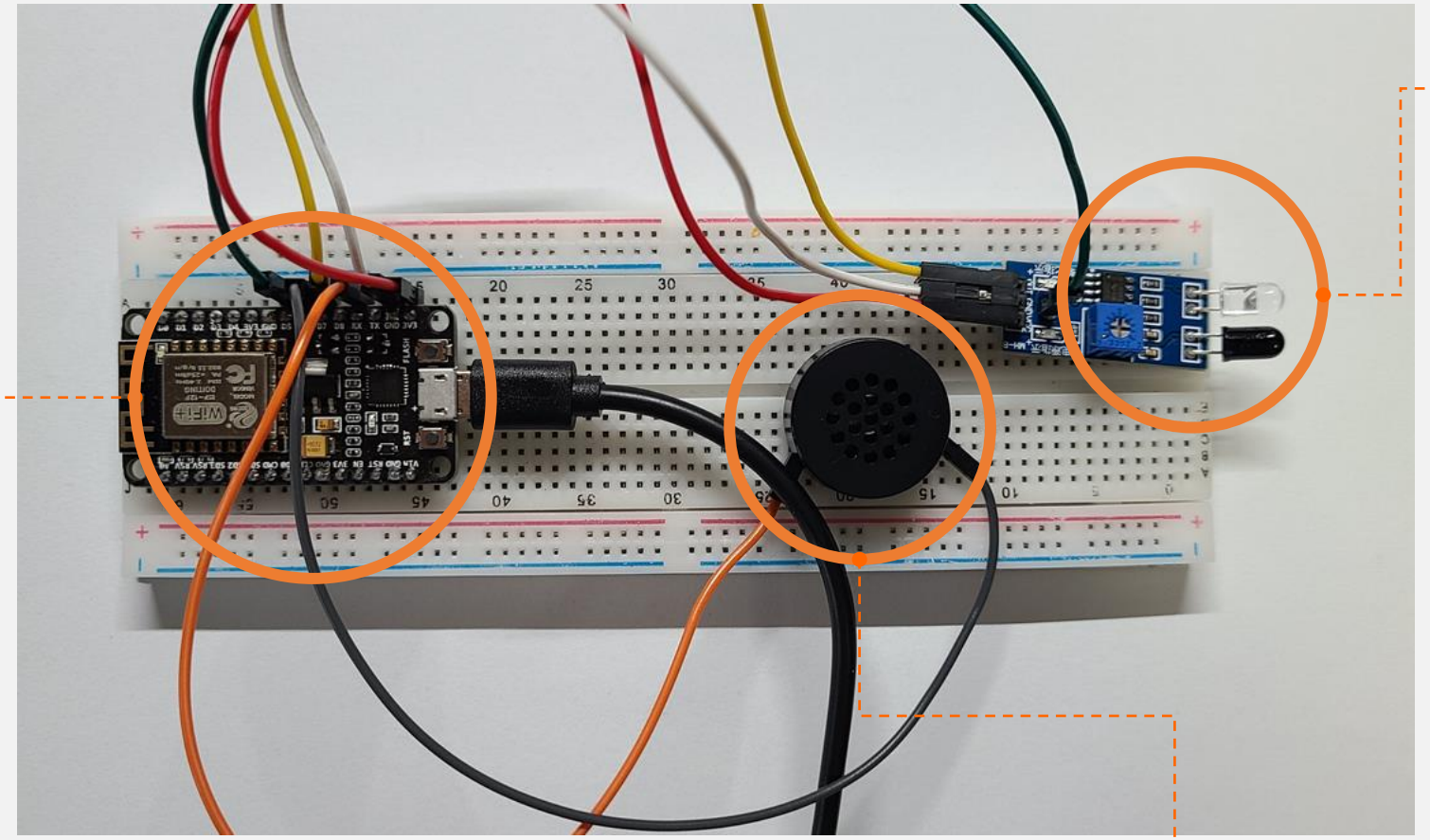
03

Resource & Device Structure

Resource Structure



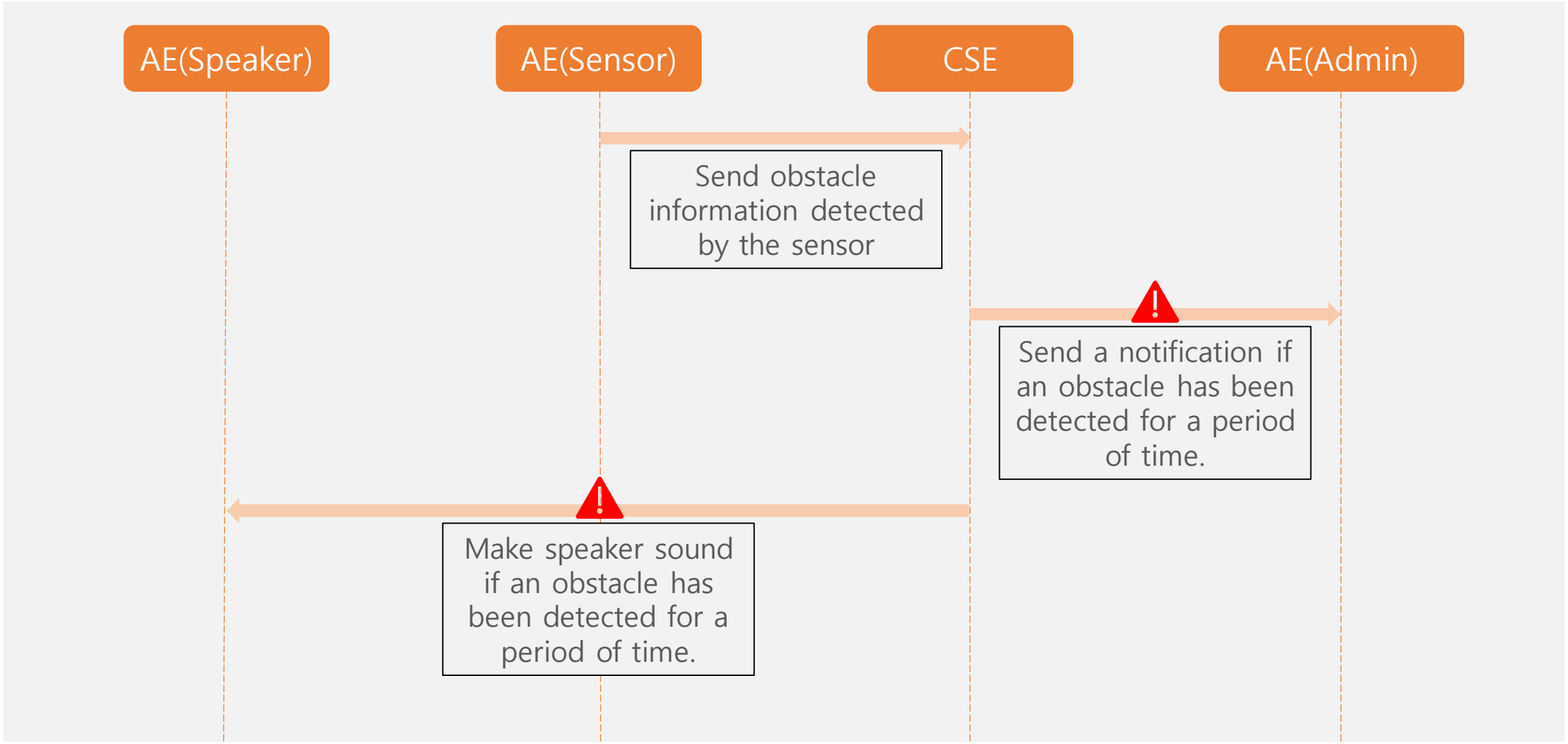
Device Structure



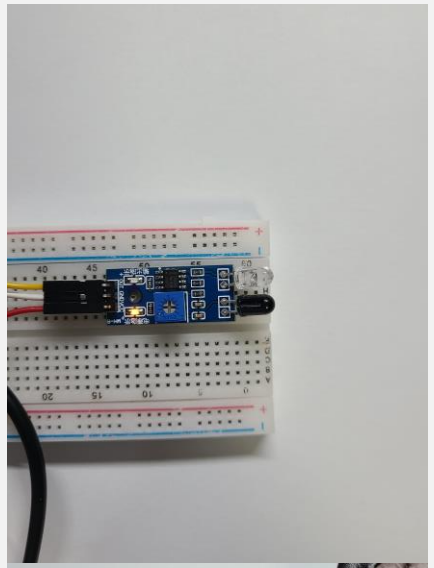
04

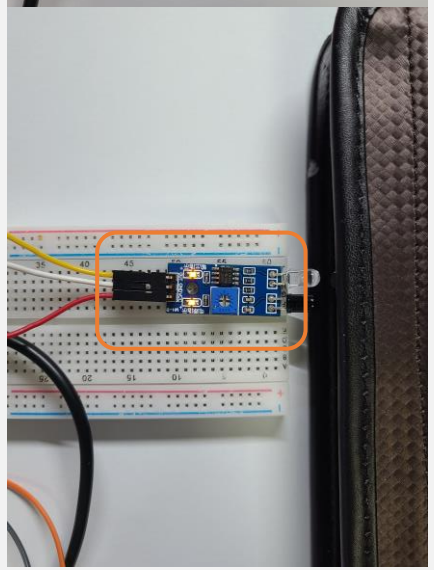
Procedure & Data Flow

Procedure


AE(Speaker)
AE(Sensor)
CSE
AE(Admin)
Send obstacle information detected by the sensor
Send a notification if an obstacle has been detected for a period of time.
Make speaker sound if an obstacle has been detected for a period of time.

Data Flow





```
http.begin("http://[address]/Mobius/ae_detectSensor/sensorData"); // path
http.addHeader("X-M2M-RI", "12345");
http.addHeader("X-M2M-Origin", "S");
http.addHeader("Content-Type", "application/vnd.onem2m-res+json;ty=4");
// header 정의

String postdata="{\"m2m:cin\":{\"con\":\"obstacle_detected!\"}}";
```

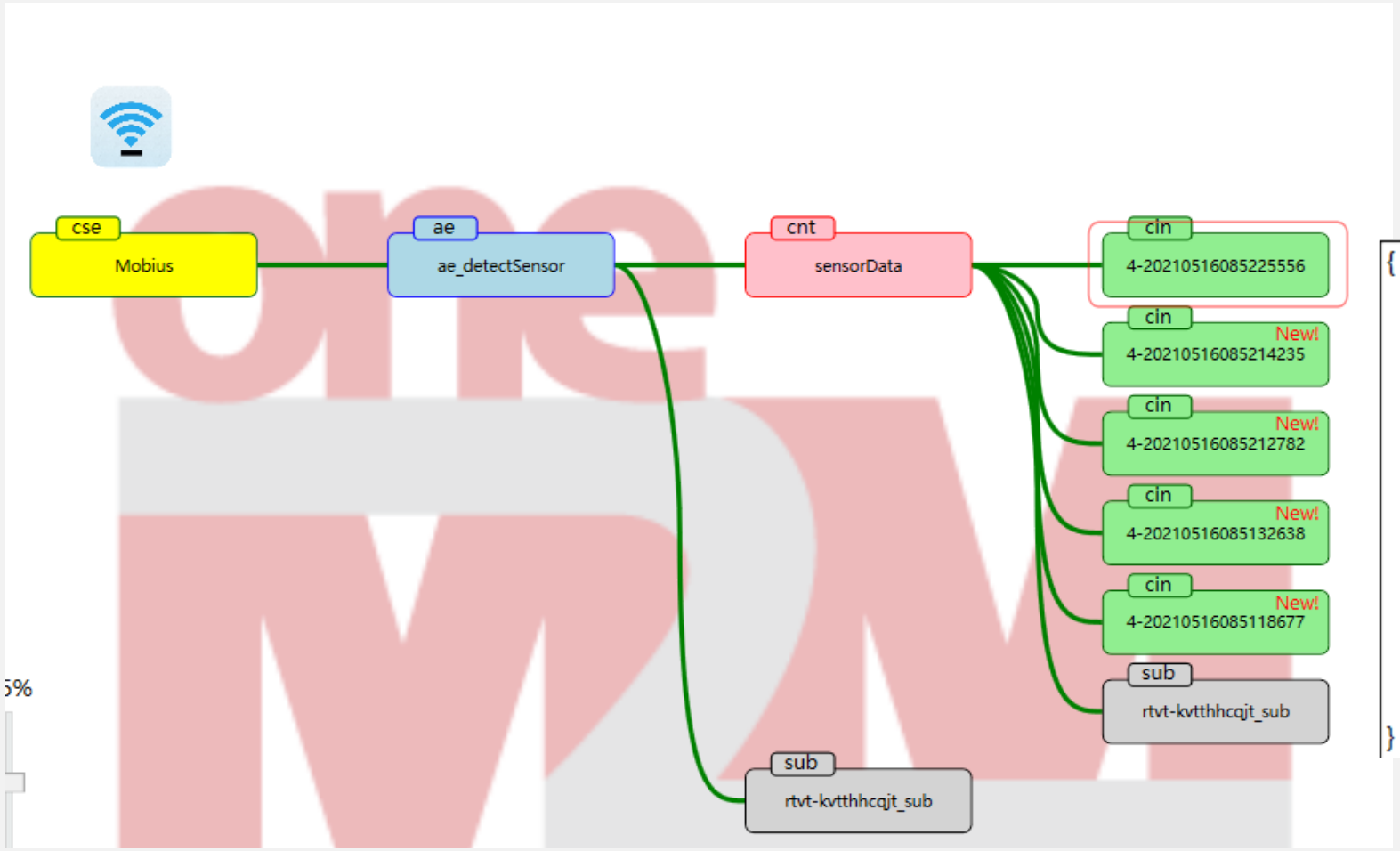
```
http.begin("http://[address]/Mobius/ae_warningSpeaker/warning"); // path
http.addHeader("X-M2M-RI", "12345");
http.addHeader("X-M2M-Origin", "S");
http.addHeader("Content-Type", "application/vnd.onem2m-res+json;ty=4");
postdata="{\"m2m:cin\":{\"con\":\"on\"}}";
```

Arduino Device -> Mobius Server

Data Flow

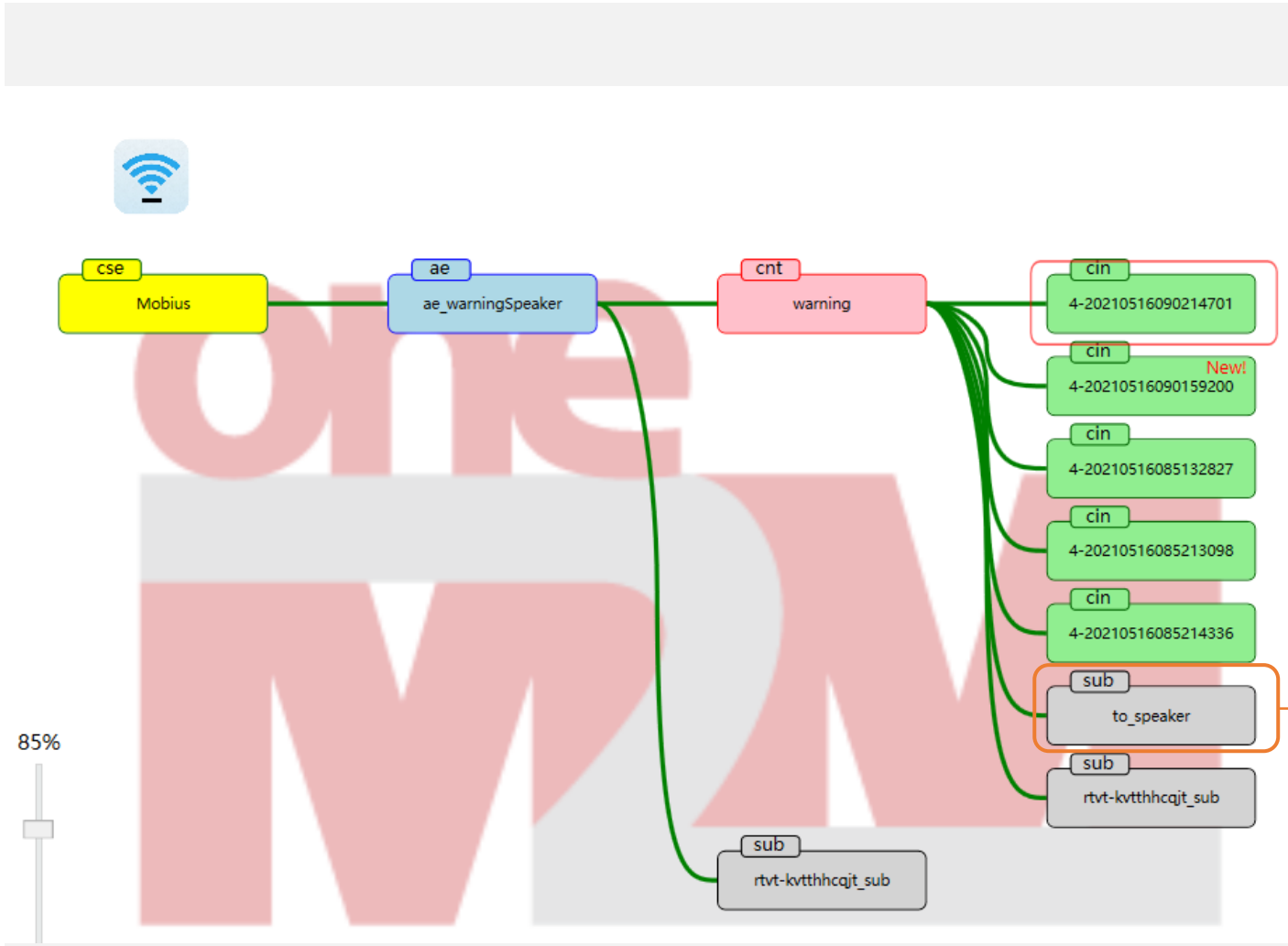
```
  '/Mobius/ae_warningSpeaker/warning': { cni: 119, cbs: 316, st: 119 }
}
[checkResponse]
=XXXXX=> [6] /Mobius/ae_warningSpeaker/warning/to_speaker
[checkResponse]
=XXXXX=> [7] /Mobius/ae_warningSpeaker/warning/to_speaker

POST : /Mobius/ae_detectSensor/sensorData
get_resource_from_url (NZOnqWbER) - /Mobius/ae_detectSensor/sensorData: 2.731ms
security.check - 1WSU8M2LHG: 0.011ms
insert_cin /Mobius/ae_detectSensor/sensorData/4-20210516085225556 - OC_bN90P7A: 1.231ms
{"m2m:cin":{"rn":"4-20210516085225556","ty":4,"pi":"3-20210515084244255780","ri":"4-202105160
18,"con":"obstacle_detected!","cr":"S"}}
""
update_parent_by_insert /Mobius/ae_detectSensor/sensorData - GtDOFCfRVT: 3.519ms
{
  '/Mobius/ae_detectSensor/sensorData': { cni: 120, cbs: 1692, st: 120 }
}
[sgn_action_send] - 200
subscribe noti_resp_topic as /oneM2M/resp/Mobius2/Crtvt-kvtthhcqjt/xml
<======= [request_noti_mqtt - /Mobius/ae_detectSensor/sensorData/rtvt-kvtthhcqjt_sub] publish
=======> [response_noti_mqtt]/Mobius/ae_detectSensor/sensorData/rtvt-kvtthhcqjt_sub

POST : /Mobius/ae_warningSpeaker/warning
get_resource_from_url (CueTmB_tKO) - /Mobius/ae_warningSpeaker/warning: 2.619ms
security.check - Bx4NkQNno6: 0.009ms
insert_cin /Mobius/ae_warningSpeaker/warning/4-20210516085225797 - uc2bZJ80AT: 1.468ms
{"m2m:cin":{"rn":"4-20210516085225797","ty":4,"pi":"3-20210515084520170139","ri":"4-202105160
2,"con":"on","cr":"S"}}
""
update_parent_by_insert /Mobius/ae_warningSpeaker/warning - usJInDH333: 3.245ms
[sgn_action_send] - 200
```

Data Flow




```
{
  "m2m:cin": {
    "pi": "3-20210515084244255780",
    "ri": "4-20210516085225556169",
    "ty": 4,
    "ct": "20210516T085225",
    "st": 120,
    "rn": "4-20210516085225556",
    "lt": "20210516T085225",
    "et": "20230516T085225",
    "cs": 18,
    "cr": "S",
    "con": "obstacle_detected!"
  }
}
```

Data Flow

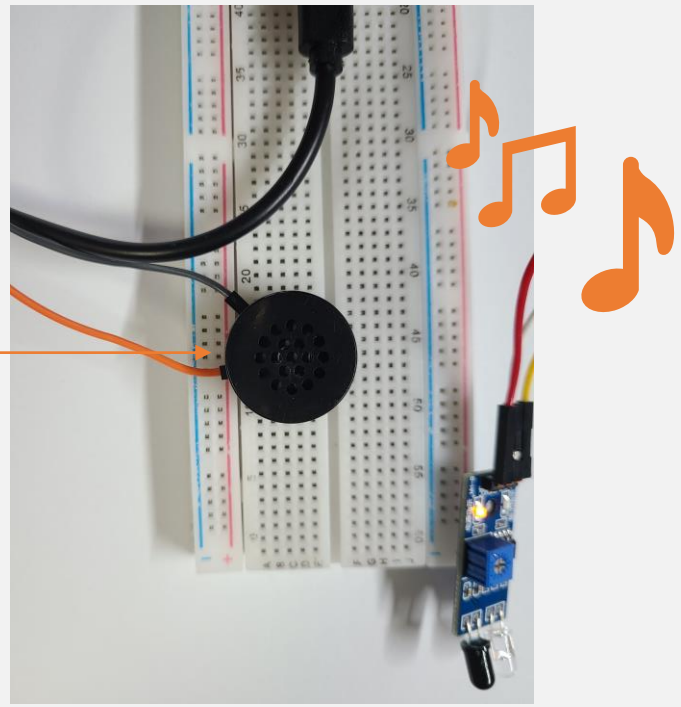



```
{
  "m2m:cin": {
    "pi": "3-20210515084520170139",
    "ri": "4-20210516090214701362",
    "ty": 4,
    "ct": "20210516T090214",
    "st": 122,
    "rn": "4-20210516090214701",
    "lt": "20210516T090214",
    "et": "20230516T090214",
    "cs": 2,
    "cr": "S",
    "con": "on"
  }
}
```

```
{
  "m2m:sub": {
    "pi": "3-20210515084520170139",
    "ri": "23-20210515112540034030",
    "ty": 23,
    "ct": "20210515T112540",
    "rn": "to_speaker",
    "lt": "20210515T112540",
    "et": "20230515T112540",
    "enc": {
      "net": [
        3
      ]
    },
    "exc": 10,
    "nu": [
      "http://address"
    ],
    "nct": 2,
    "cr": "SOrigin"
  }
}
```

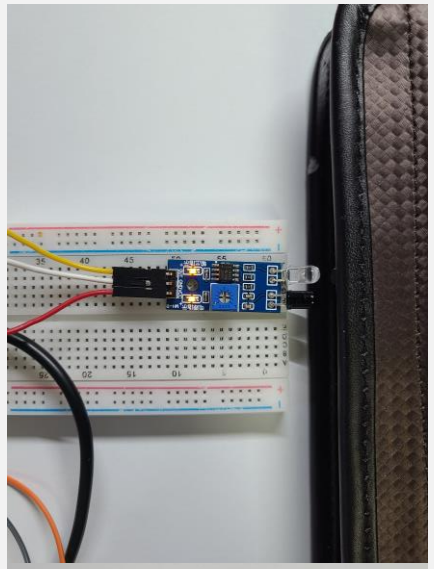
Data Flow

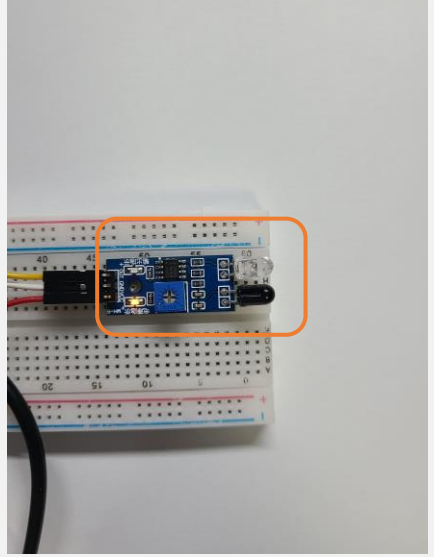
```
WiFiClient client = server.available();
if(client){
  StaticJsonBuffer<400> JSONBuffer;
  for(int i=0;i<9;i++){
    String im=client.readStringUntil('\r');
    //Serial.println(response);
  }
  for(int i=0;i<46;i++){
    String im=client.readStringUntil('"');
    //Serial.println(response);
    response=im;
  }
  Serial.println(response);
}
if(response=="on"){
    rickroll();
}
```



Mobius Server -> Arduino Device

Data Flow



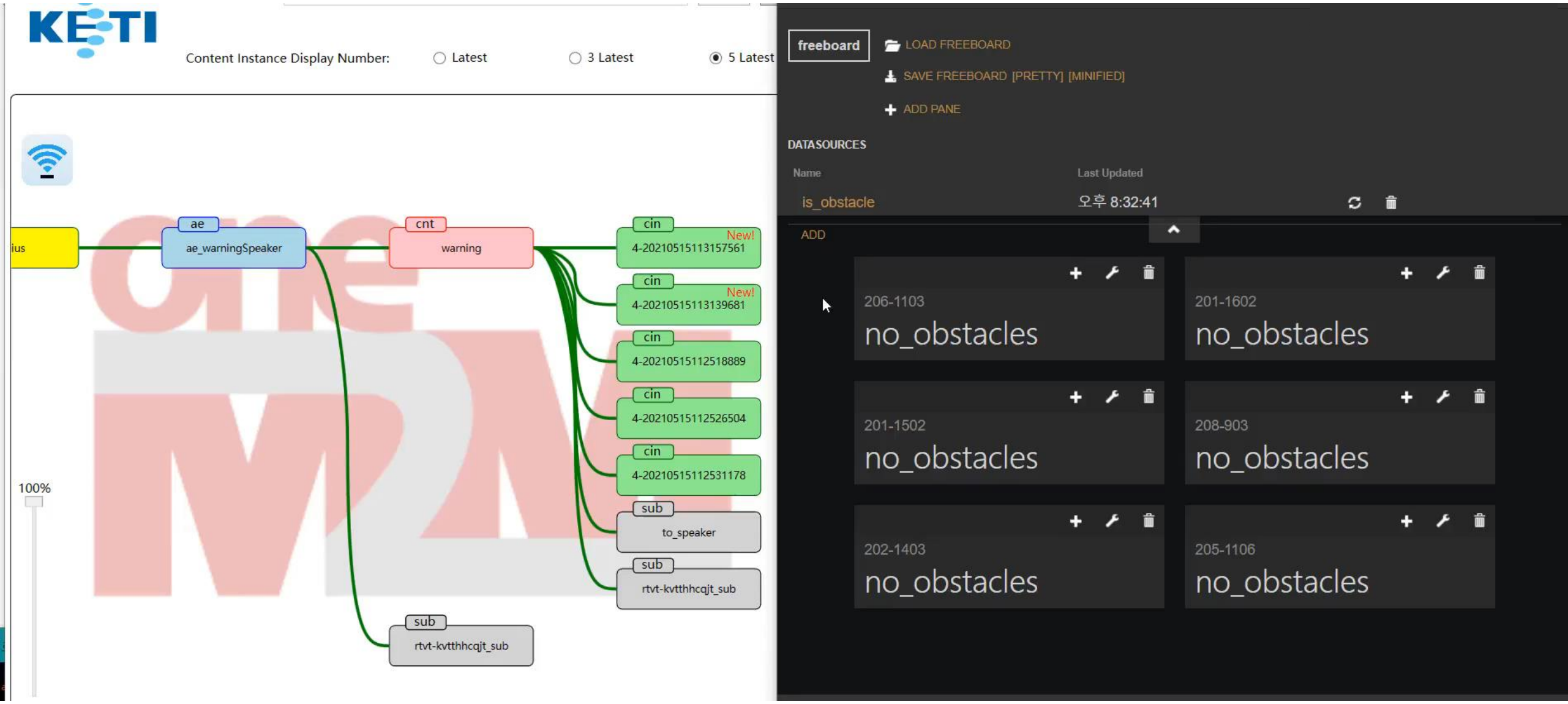


```
http.begin("http://219.240.212.226:7579/Mobius/ae_detectSensor/sensorData"); // path
http.addHeader("X-M2M-RI", "12345");
http.addHeader("X-M2M-Origin", "S");
http.addHeader("Content-Type", "application/vnd.onem2m-res+json;ty=4");
// header 정의

String postdata="{\"m2m:cin\":{\"con\":\"no_obstacles\"}}";
//body 정의
```

```
http.begin("http://219.240.212.226:7579/Mobius/ae_warningSpeaker/warning"); // path
http.addHeader("X-M2M-RI", "12345");
http.addHeader("X-M2M-Origin", "S");
http.addHeader("Content-Type", "application/vnd.onem2m-res+json;ty=4");

postdata="{\"m2m:cin\":{\"con\":\"off\"}}";
```

05

Demonstration


KETI
Content Instance Display Number:
Latest
3 Latest
5 Latest
ae
ae_warningSpeaker
cnt
warning
cin
New!
4-20210515113157561
cin
New!
4-20210515113139681
cin
4-20210515112518889
cin
4-20210515112526504
cin
4-20210515112531178
sub
to_speaker
sub
rtvt-kvtthhcqjt_sub
sub
rtvt-kvtthhcqjt_sub
100%
freeboard
LOAD FREEBOARD
SAVE FREEBOARD [PRETTY] [MINIFIED]
ADD PANE
DATASOURCES
Name
Last Updated
is_obstacle
오후 8:32:41
ADD
206-1103
no_obstacles
201-1602
no_obstacles
201-1502
no_obstacles
208-903
no_obstacles
202-1403
no_obstacles
205-1106
no_obstacles

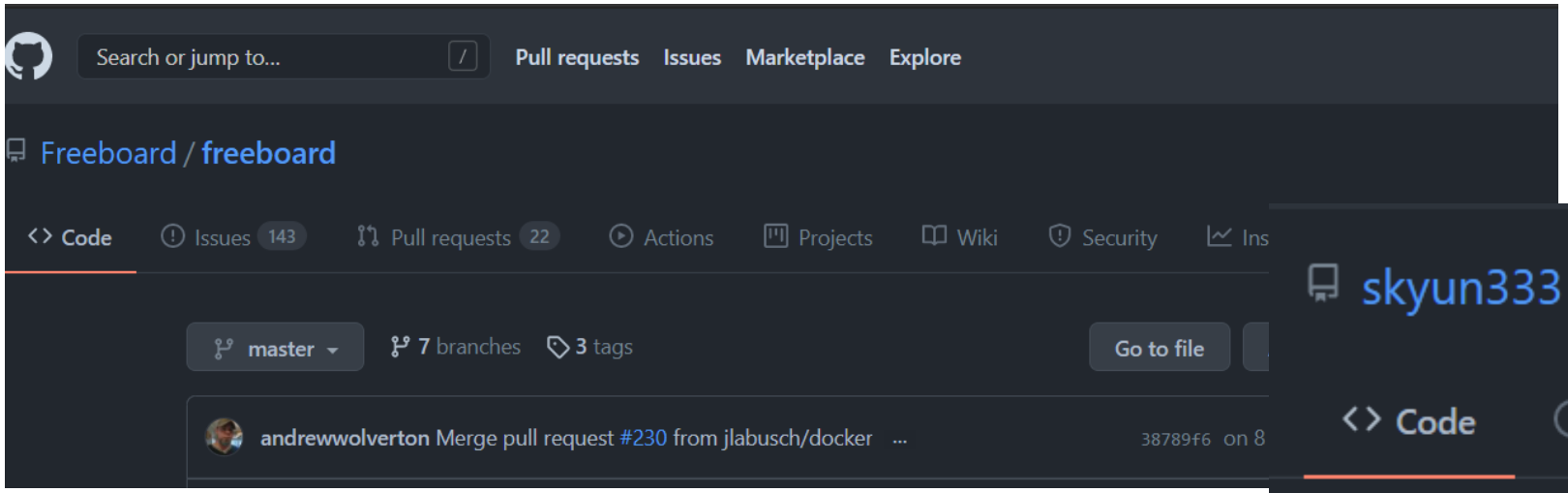
06

Benefits

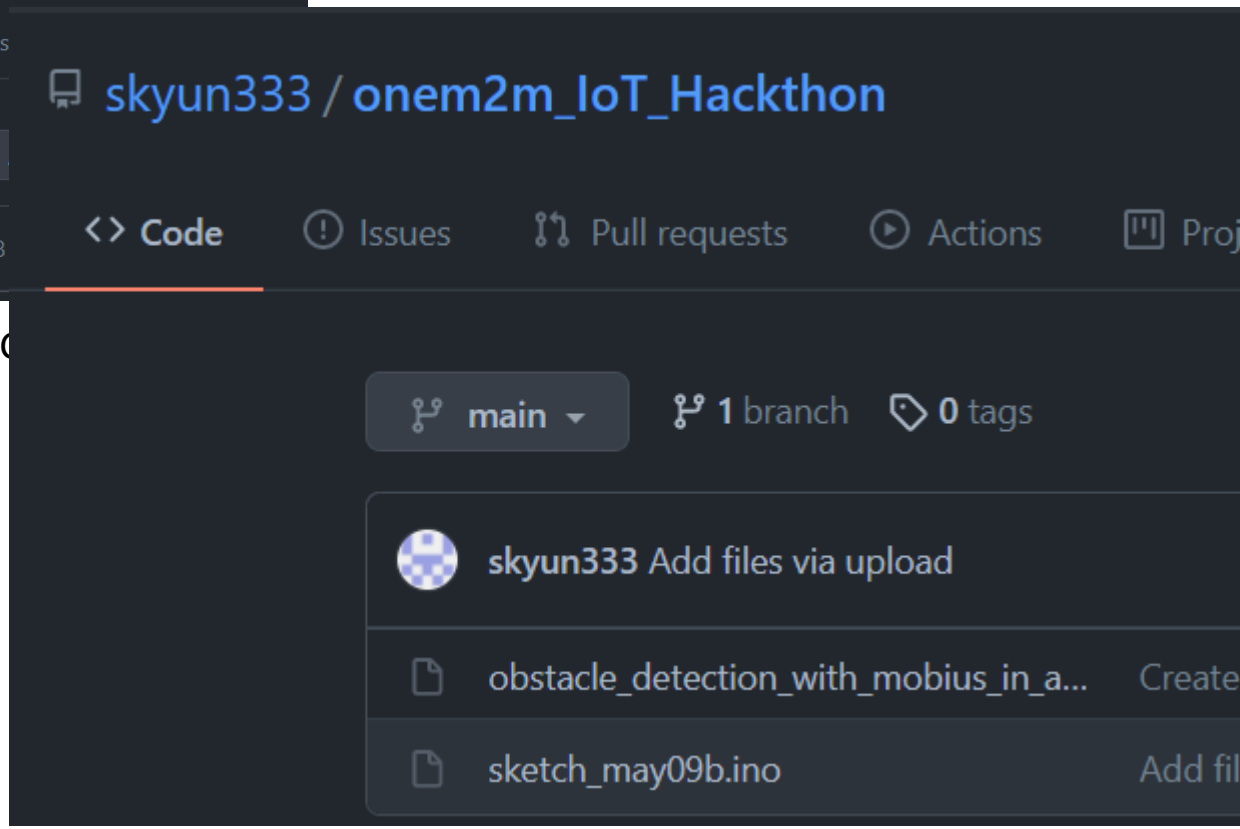
Benefits

- Speaker & Web
- No obstacles around fire hydrant
- Effective fire patrol

"Can be used at additional ideas"



dashboard source : https://github.com/Freeboard/freeboard



detailed code : https://github.com/skyun333/on

Thank you!

detailed code : https://github.com/skyun333/onem2m_IoT_Hackthon

dashboard source : https://github.com/Freeboard/freeboard.git