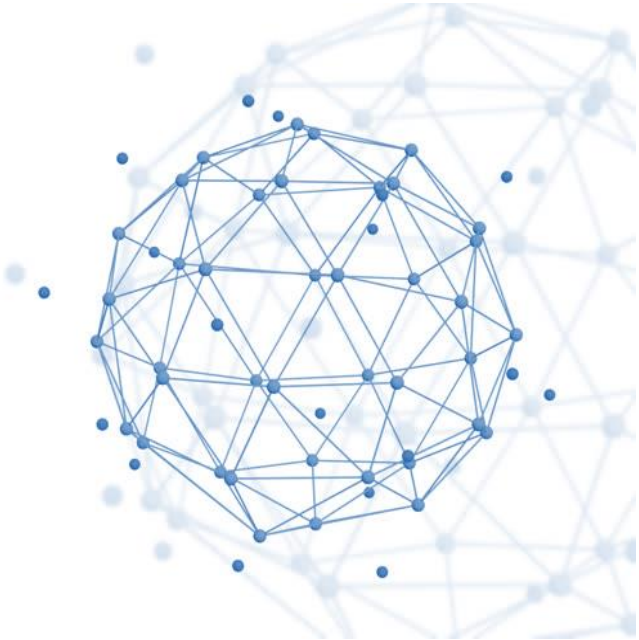
HALLYM UNIVERSITY

# MQTT Programming 5

2020.05.27

**Sang-woo Lee**

**glutton.leesw@gmail.com**



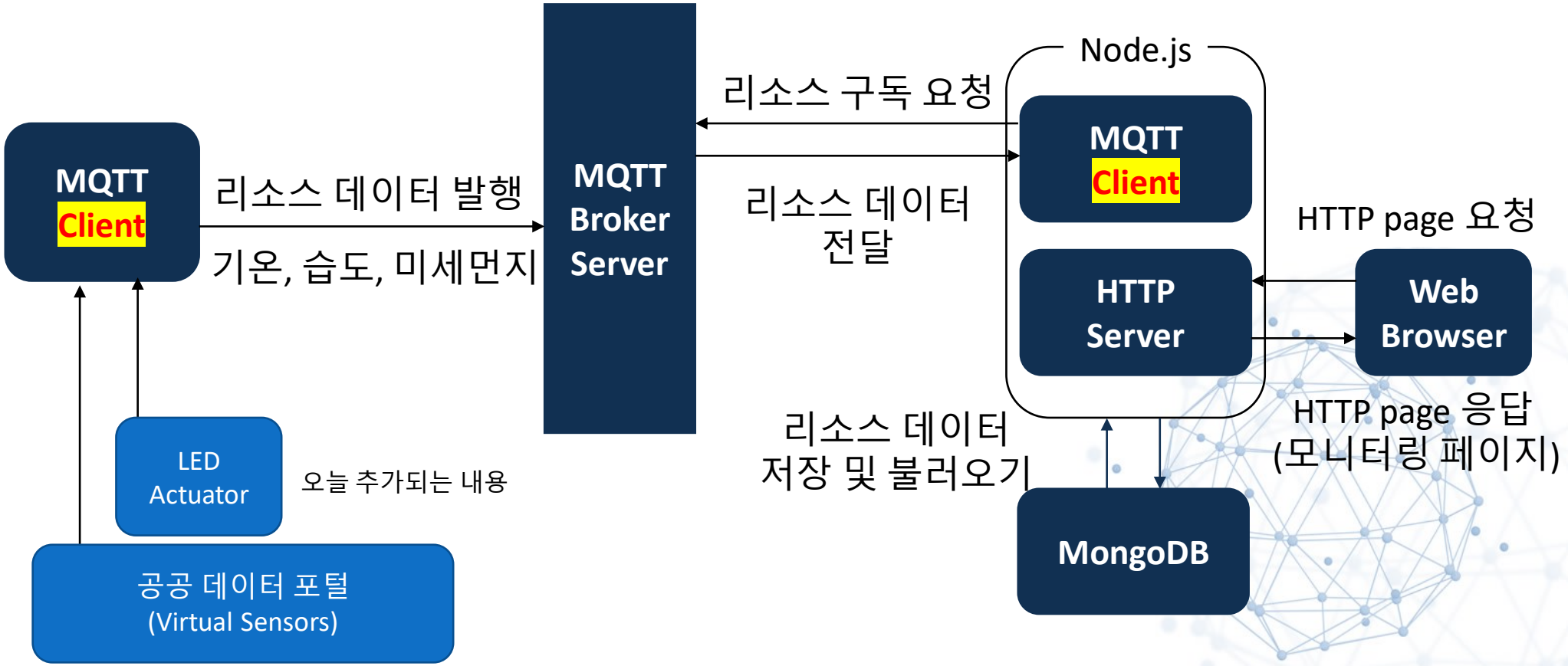
CIC Convergence Information & Communications

# Contents

- Virtual Actuator 구현 및 Virtual Actuator 제어

# Virtual Actuator 구현 및 Virtual Actuator 제어

- 복습 및 실습진도 확인
  - Mini project 실습이 끝난 상태에서 이번 실습을 진행해주세요.



# Virtual Actuator 구현 및 Virtual Actuator 제어

- IoT_server

**www.js**

```
// Mongo DB에서 최근 데이터 불러와서, HTML 페이지에 업데이트
var io = require("socket.io")(server);
io.on("connection", function(socket){
  socket.on("socket_evt_update", function(data){
    var temp = dbObj.collection("temperature"); // temperature 라는 이름의 collection 선택
    var humi = dbObj.collection("humidity"); // humidity 라는 이름의 collection 선택
    var pm_value = dbObj.collection("pm_value"); // pm_value 라는 이름의 collection 선택
    // 온도 데이터
    temp.find({}).sort({_id:-1}).limit(1).toArray(function(err, results){
      // collection에서 가장 최근 데이터 정렬-> 하나의 데이터만 불러옴 -> 배열로 만듬
      if(!err){
        console.log(results[0]);
        socket.emit("socket_up_temp", JSON.stringify(results[0]));
      }
    });
    // 습도 데이터
    humi.find({}).sort({_id:-1}).limit(1).toArray(function(err, results){
      // collection에서 가장 최근 데이터 정렬-> 하나의 데이터만 불러옴 -> 배열로 만듬
      if(!err){
        console.log(results[0]);
        socket.emit("socket_up_humi", JSON.stringify(results[0]));
      }
    });
    // 미세먼지 데이터
    pm_value.find({}).sort({_id:-1}).limit(1).toArray(function(err, results){
      // collection에서 가장 최근 데이터 정렬-> 하나의 데이터만 불러옴 -> 배열로 만듬
      if(!err){
        console.log(results[0]);
        socket.emit("socket_up_pm", JSON.stringify(results[0]));
      }
    });
  });
  // HTML 페이지에서 버튼이 눌리면 LCD 제어를 위해 MQTT publish
  socket.on("socket_evt_bnt", function(data){
    console.log("publishing led");
    console.log(data);
    client.publish('led', data);
  });
});
```

이 부분 추가

# Virtual Actuator 구현 및 Virtual Actuator 제어

- IoT_server

**MQTT.html**

```
    function timer_1(){
        socket.emit("socket_evt_update", JSON.stringify({}));
    }
    function button_on() {
        socket.emit("socket_evt_bnt", "ON");
    }
    function button_off() {
        socket.emit("socket_evt_bnt", "OFF");
    }
</script>
</head>
<body>
MQTT Mornitoring Service
    <div id="msg">
        <div id="mqtt_logs">
            <ul class="mqttlist_temp"></ul>
            <ul class="mqttlist_humi"></ul>
            <ul class="mqttlist_pm"></ul>
            <button id="button1" onclick="button_on();"><b>LED on</b></button>
            <button id="button1" onclick="button_off();"><b>LED off</b></button>
        </div>
    </div>
</body>
</html>
```

이 부분 추가

이 부분 추가

# Virtual Actuator 구현 및 Virtual Actuator 제어

- MqttPublisher_API.java (전반적인 수정)
  - Class에 “implements MqttCallback” 추가 → 필요한 3개의 method 정의
    - ✓ 특히, messageArrived() 함수 중요
  - 기존에 static 속성으로 정의했던 method들 static 속성 제거
  - 기존 main문에 정의했던 기능 run() 함수으로 따로 작성
    - ✓ main문에서 클래스 객체 생성, run 실행
  - run() 함수에 ‘led’ subscribe 추가
  - connectBroker() 함수에 setCallback() 라인 추가

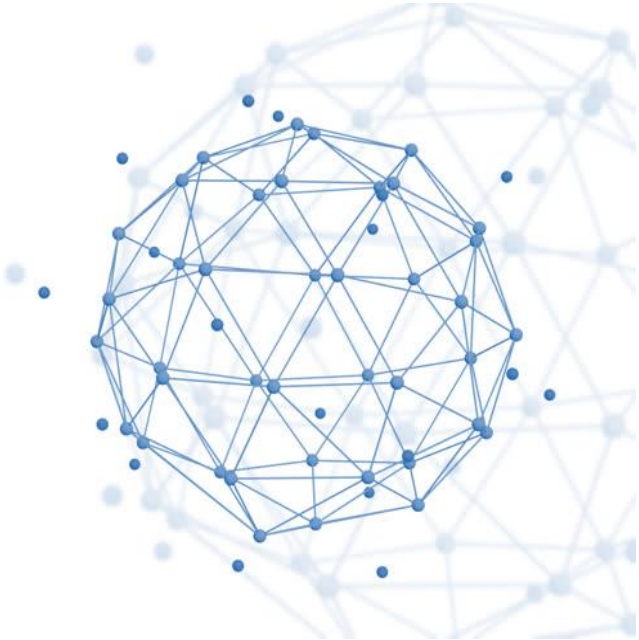
**영상 꼭 참고!**

# 프로젝트 참고

- 아이디어의 독창성 필요성을 구체적으로 생각해봐야 함.
- API & 크롤링할 데이터를 탐색해서, 자신만의 virtual sensor 구현.
- Actuator의 경우 GUI 혹은 다른 방식으로 표현할 수 있음.
- UI (HTML page)의 기능 확장.

HALLYM UNIVERSITY

# Thank you



CIC Convergence Information & Communications