

01. About me

Research Field (Master Degree)

Network(Protocol Design &Impl)
Energy Management (Server)
SDN

Current

SK (2019.05 ~)
: SKT Billing System 개발/운영



Github @ <https://github.com/hyyyjin>

02. My Experience – *Language Experience*

Network(Protocol Design & Impl)

Energy Management System(Server)

Protocol: HTTP, CoAP, MQTT, Web Socket

Multi Thread – Over 1,000 Clients

(Spring boot Basic experience)

Server Performance simulator

Protocol: HTTP, CoAP, MQTT

Latency, Data Traffic, Event Response

Thread Pool – Over 1,000 Threads

Protocol Testing Server

(HTTP, UDP, TCP, OpenFlow, ARQ and etc.)

Energy Management Agent Interface design(ISO/IEC)

Open Source : EPRI OpenADR

Language : Java, Oracle SQL

Tools : Bit bucket, GitHub(Source version 관리)

OS : Linux

Server : Window Server 2012, Google Cloud

Teaching Assistant @ Hanyang Univ

SK Telecom Billing System

SK Telecom Billing System(Swing)

(1) 자동납부 신청 Batch Program

(2) 자동납부 실시간 신청 API 개발

SK Telecom Billing System UI(Swing)

(1) 자동납부 Promotion 관리 UI 개발

(2) 자동납부 신청 오류 Monitoring UI 개발

Language : Pro c, JavaScript, Oracle SQL

Tools : Harvest(Source version 관리)

Oracle SQL(Certification : SQLD)

- SK Telecom Billing System 운영

Side Project

Ongoing

(1) 전국 아파트 Ranking Backend 서버 개발

(2) 공공기관 API Crawling Batch

- 전국 아파트 실 거래가 Daily 배치

(3) 아파트 실 거래가 테이블 설계 및 관리

Language : Python, SpringBoot, Oracle SQL

Tools : GitHub(Source version 관리)

02. My Experience – *Project & Paper (1/2)*

SK Telecom Billing System

1) SKT – 외부기관(금융결제원) 카드 원장 등록 프로세스 구축

: 고객 Migration 배치 프로그램 개발
SKT 카드 자동납부 결과 조회 화면 개발
카드 원장 신청/해지 Daily 배치 프로그램 개발

2) SKT - 간편결제(SK Pay, Kakao, Payco) 자동납부 방법 추가 프로세스 구축

: 외부기관(간편결제 App) 자동납부 신청 API 개발

3) SKT – SK VAN – 금융기관 실시간 자동납부 신청 인터페이스 설계 및 구현

: 이동전화 → SKT 고유번호 조회 API 개발
가입 가능 여부 조회 및 신청 API 개발

4) SKT –금융기관(카드사) 자동납부 신청 배치 프로그램 개발

: KB카드 → SKT (Monthly) 자동납부 신청 파일 송수신 배치프로그램 개발
파일 송수신 Layout 설계

5) SKT 자동납부 전환 프로모션 자동화 설계 및 구현

: 프로모션 관리 화면 개발
프로모션 대상 집계 자동화 On demand 배치 프로그램 개발

6) (Ongoing) 114 ARS 자동납부 신청&결과 api 개발

: 자동납부 신청 가능여부 조회 API
실시간 자동납부 신청/변경/해지 API
자동납부 신청/변경 결과 조회 API

SK
(2019.05 ~

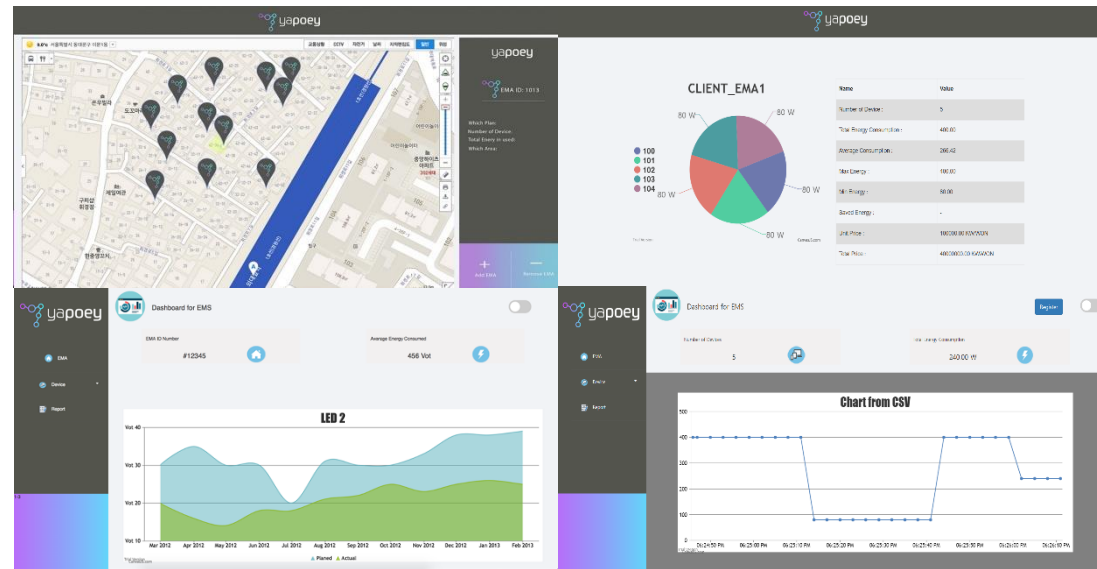
02. My Experience – *Project & Paper (2/2)*

Project 1 (2016)	ICT Program under Ministry of Employment and Labor in Korea, one-man Project(Excellence Award) Subject : Smart home System based Web & Mobile App
Paper 1 (2017)	The Korean Institute of Communications and Information Sciences Paper(Excellence Award, 1ST Author) Title : Research on The Necessity of Push Mechanism on Smart Energy IoT Environment of OpenADR 2.0b Demand Response Desc : 한국통신학회 논문 등재, 제시한 CoAP/JSON 통신 방식은 HTTP/XML 통신 방식과 비교하였을 때 약 4.39배 적은 데이터 트래픽 유발
Paper 2 (2017)	The Korean Institute of Communications and Information Sciences Paper(1st Author) Title : Design and Analysis of Push Mechanism Based on CoAP Observe for Demand Response in Energy IoT Environment Desc: 제한된 환경에서 Pull mechanism 64.36% 더 많은 데이터 트래픽 유발(네트워크 병목현상, Scalability 해결 방안 제시), Polling 주기가 5초일 때 Push mechanism은 약 74.62배(약 2.983초 차이) 향상된 속도
Presentation1 (Speaker, 2017)	Hanyang Univ - Waseda Univ(JAPAN) - UKM – UMS Subject : Design and Analysis of Push mechanism based on CoAP Observe for Demand Response in Energy IoT Environment
Project (2018)	Server Performance simulator - HTTP, MQTT, CoAP Dummy Client로 서버와 통신하여 원하는 성능 측정 실험 수행을 요청하고 종료까지 자동화 결과 : 기존에 수작업으로 진행돼 7일 이상 걸리던 실험을 자동화하여 3일 이내로 줄이는 효과 - 실험 결과 Data Traffic, Latency 결과 추출 자동화 결과 : 오차율이 높고 최소 2일 이상 소요되던 결과 분석 과정을 스크립트를 통해 분석하여 단 몇시간 내로 정확한 분석
Presentation 2 (Speaker, 2018)	Hanyang Univ – Keio Univ(JAPAN) Subject : Demonstration Energy management system
Master Course (~2019.02)	Implementation of Orchestration for Energy Management (Excellence Award in Hanyang) - Energy Management System(JAVA) - Protocol Interface Design and Implementation (Restful, MQTT, CoAP) - Server Performance simulator(Latency, Data Traffic, Event Response) - Network Protocol Auto Testing System(JAVA)

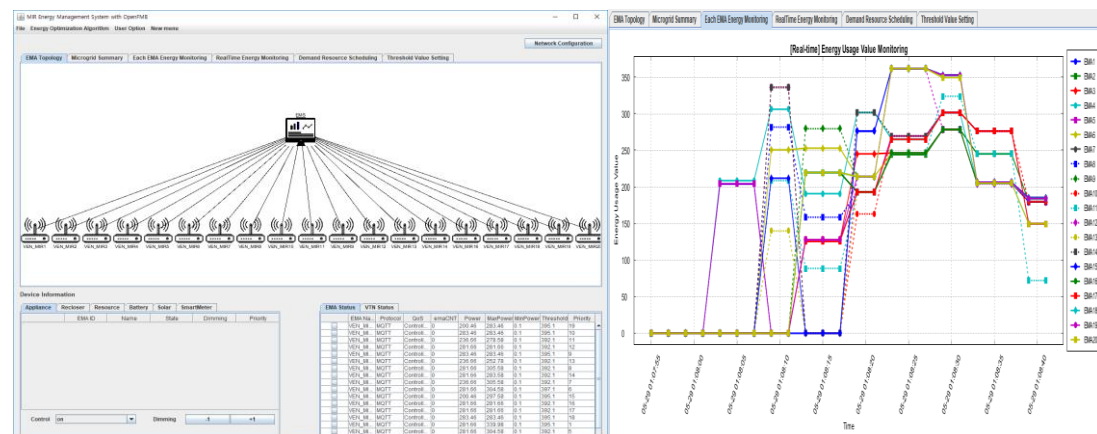
02. My Experience – *On Lab*

What I did

- 1) Multi Thread(MAX: 1,00Clients Experiment, CPU: 1-Core, RAM: 4GB)
- 2) Network Topology Discovery
- 3) Each sub node(EMA) energy consumption monitor
- 4) Total energy consumption monitor
 - Goal consumption(Threshold)
 - Current usage consumption
- 5) Demand Response Event(Manually)
- 6) Micro grid simulator (OpenFMB)
- 7) Cloud Service (VM : Hyper-V, Google Cloud)

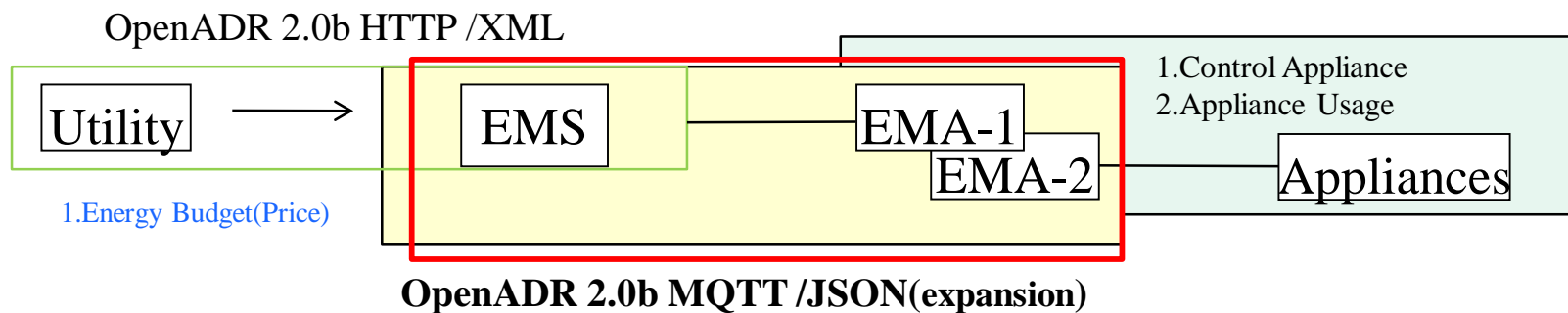
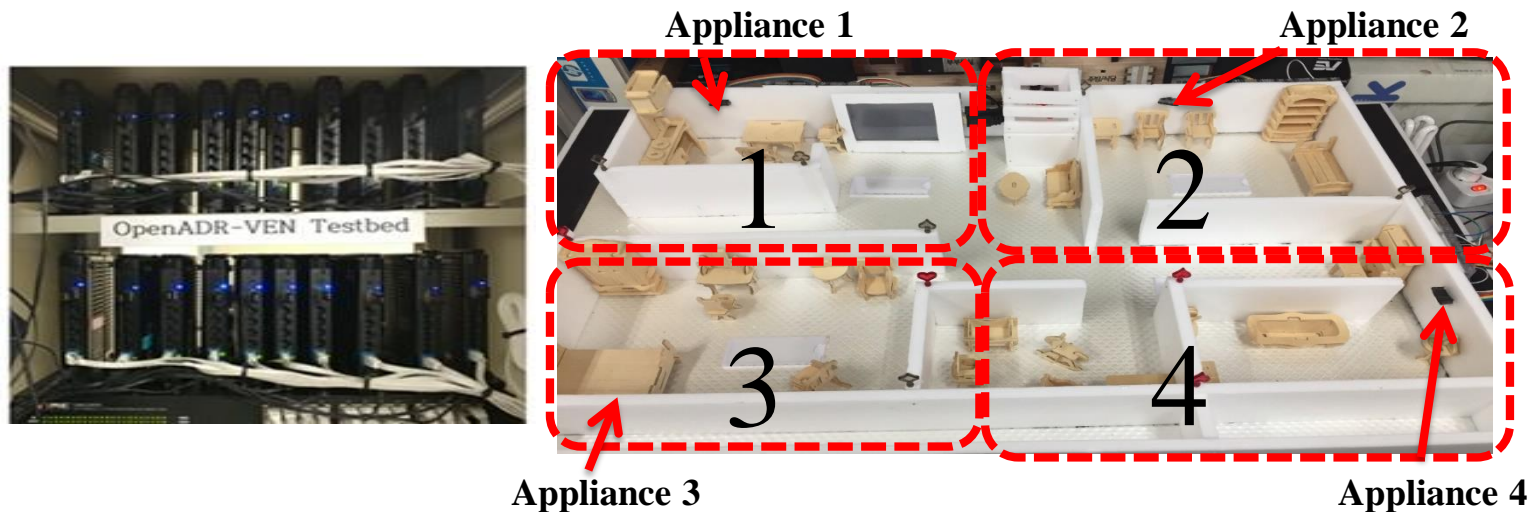


▲ Energy Management Orchestrator (Spring Boot)



▲ Energy Management System(Java Environment)

02. My Experience – *On Lab*



What I did

- 1) EMS – EMA Communication
- 2) Data Modeling
- 3) Event Distribute Algorithm (Priority, Greedy, Knapsack)

Github @ <https://github.com/92phantom>

Demonstration Video @

https://www.youtube.com/watch?v=6_pg-Md-EKk

