

# **역량 포트폴리오**

**지원자 박현일**

## ■ 지원직무에 대한 보유지식/경험 및 관심/열정

-광고동아리 기획팀장 (Must Have item) -입대(통신병)	캡스톤 디자인 옷장 관리 프로젝트 (안드로이드 앱 개발 DB, 임베디드)	학부 졸업 게임아카데미 대용량 서버프로그램 구현	-한국통신학회 논문 3편 (MQTT, CoAP, HTTP) -한국통신학회 하계학술대회 논문 우수상(스마트그리드) -네이버 기술 장학생상(개발 및 연구성과 우수)	
2011	2013	2015	2017	
<b>2009</b>	<b>2012</b>	<b>2014</b>	<b>2016</b>	<b>2018-2019</b>
-대학 입학 2009~2015 -대학교 광고동아리 애드밸리 '기획' -대학교 봉사 동아리 참동이	-제대(통신병) -서점관련 -웹 프로젝트 (웹 서버)	-차세대 인터넷 연구실(2014~2015) -학부 졸업 논문 빅데이터 관련 논문 -OCP 11g 자격증(DB) -JAVA, 빅데이터 교육(2달) (한국생산성본부)	-한양대 대학원 입학 MIRLAB(2016~2018) -네트워크 대표조교 -연구실 대표(~18.3) -과제 실무담당자(2건) -프로토콜(표준화) 개발연 구 (스마트그리드 IoT, SDN)	-석사졸업논문 -표준화 기고서 1건 -학위논문우수상 -석사 졸업(18.8) -현재 알고리즘 공부 -네오위즈 블록체인 인턴 수료 3개월

활동 : Mobile Intelligence and Routing LAB(<http://mir.hanyang.ac.kr>)

논문 3건, 학술대회 논문 1건, 단기강좌 2건, 워크샵 발표 1건, 정부 과제 2건, 네이버 기술 장학생상 1건, 학위 논문상 1건

JAVA 빅데이터 교육 2달(한국생산성본부), 논문 우수상 1건, 사용언어 : C, C++, 사용가능 언어: C, C++, JAVA, PHP, MySQL

- 학사 논문: Hadoop 및 Mongo DB 기반 대용량 로그 처리 시스템 설계 및 구현
- 석사 논문: 에너지 IoT 기반 집합 건물 환경에서 효율적인 수요반응을 위한 경량 에너지관리에이전트 프로토콜 설계 및 구현
- 논문: 스마트 에너지 IoT를 위한 CoAP 기반 Lightweight OpenADR2.0b 프로토콜의 구현 및 분석. 한국통신학회논문지, 42(4), 904-914
- 논문: 에너지 IoT 환경에서 실시간 수요반응 서비스를 위한 CoAP Observe 기반 Push Mechanism 설계 및 분석. 한국통신학회논문지, 43(3), 529-540
- 논문: 에너지 IoT 환경을 위한 Multicast 방식의 Lightweight 수요반응 프로토콜 제안 및 분석. 한국통신학회논문지, 43(7), 1163-1175.
- 한국통신학회 하계 학술대회(우수논문상) : 스마트 에너지 IoT 환경에서 OpenADR 2.0 b 수요반응의 Push 메커니즘 필요성 연구". 한국통신학회 학술대회논문집(2017), 283-284
- 단기강좌 강의 : IoT 기반 OpenADR 단기 강좌(TIPS 최진식, 박현일, 박현진, 이성환, 이재조박사(한국전기연구원), OSIA), 2017. 7. 3-4
- 단기강좌 강의 : SDN 단기 강좌(TIPW, 최진식, 조호준, 이양, 천세준, 박현일, 박현진, 이성환, ATTO Research, OSIA), 2018. 2. 26-27
- 워크샵 발표 : Design and Analysis of Push mechanism based on CoAP Observe for Demand Response in Energy IoT Environment, 2017 Waseda-UKM-UMS-Hanyang IT Workshop, 2017. 12. 1-2
- 정부 과제 : 스마트그리드와 연계된 스마트가전 인터페이스 시험방법 표준개발 (2016.9.1~2018.3.31), MIRLAB 대표, 실무담당자 참여
- 정부 과제 : IoT 구축을 위한 초고신뢰성 패킷네트워크 개발의 사설 표준화 (2016.9.1~2018.2.28), 실 MIRLAB 대표, 실무담당자 참여

# 포트폴리오 목차

- 1. 스마트 에너지 홈 관련 논문
- 2. 스마트 옷장 (캡스톤 디자인)
- 3. 빅데이터 관련 논문
- 4. 게임 프로젝트(프로그램 아카데미)
- 5. 네오위즈 인턴 기술본부 블록체인팀

# 1. 스마트 에너지 홈 관련 논문

- 연구 배경
- 연구 목적
- 실험 환경
- 결과

**에너지 IoT 기반 집합 건물 환경에서  
효율적인 수요반응을 위한  
경량 에너지관리에이전트 프로토콜 설계 및 구현**

(Design and Implementation of Lightweight Energy Management Agent Protocol  
for Efficient Demand Response in Energy IoT based Community Environment)

지원자 : 박현일

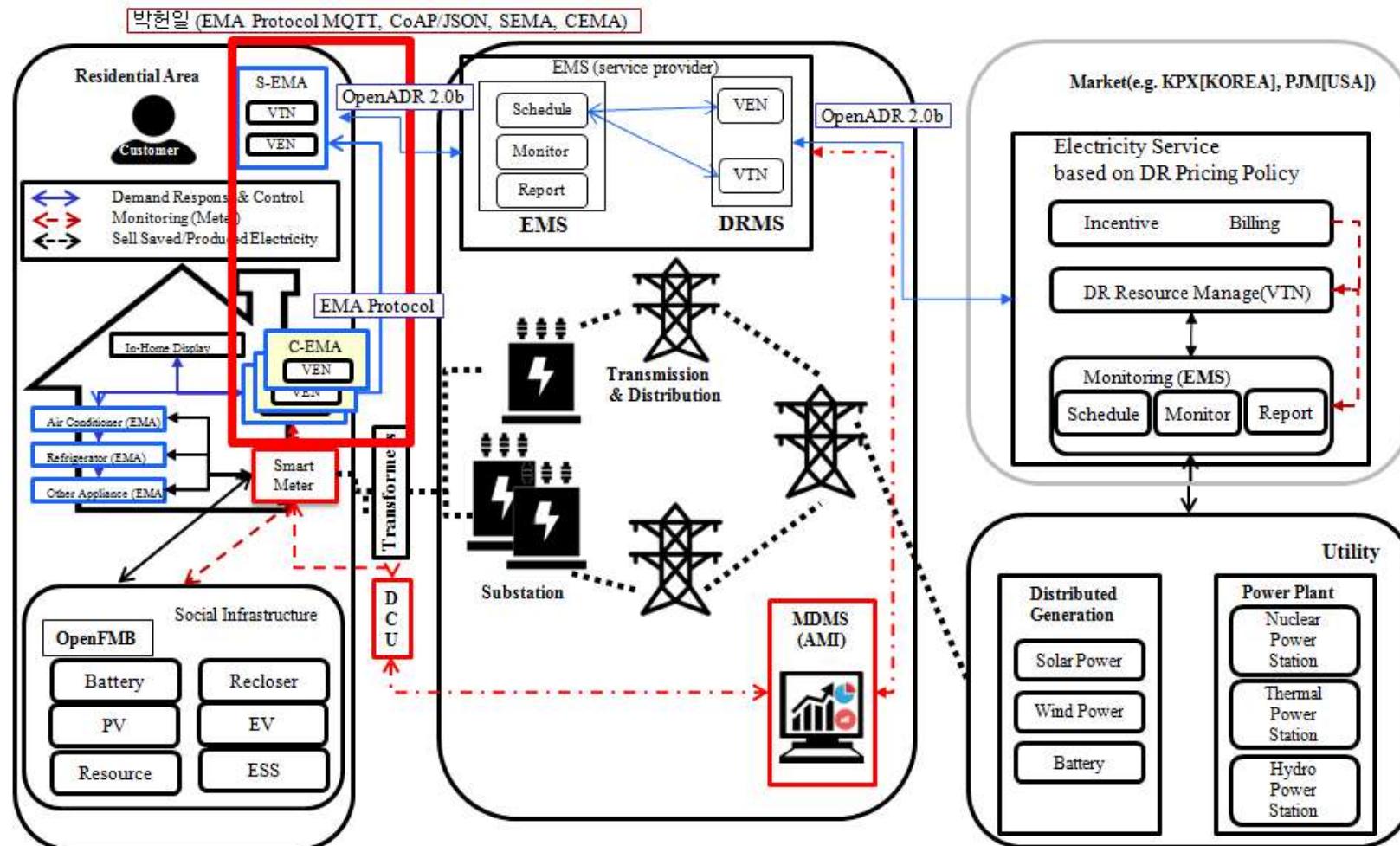
# 목차

- 학위논문 목차
- Background
- 연구 배경 및 필요성
- 연구 목적
- 문제점 및 해결책(경량 에너지관리에이전트 프로토콜 설계)
- 에너지관리에이전트 프로토콜의 수요반응 4가지 서비스 (차이)
- 경량 에너지관리에이전트 프로토콜 구현
- 경량에너지관리에이전트 프로토콜 실험 및 결과
- 결론 및 향후 과제
- 참고문헌

# 학위 논문 목차

- 제 1장. 서 론
  - 제 1절 연구 배경
  - 제 2절 연구 목적
  - 제 3절 논문의 구성
- 제 2장. 관련 연구
  - 제 1절 스마트그리드
  - 제 2절 수요반응
  - 제 3절 수요반응 및 스마트 그리드 기술에 관한 연구
  - 제 4절 에너지 IoT 환경에서의 경량 수요반응 프로토콜
  - 제 5절 에너지관리에이전트 및 표준화 동향
- 제 3장. 집합 건물 환경에서 효율적인 수요반응을 위한 경량 에너지관리에이전트 프로토콜 설계 및 구현
  - 제 1절 기존 주거 공간의 수요반응 프로토콜 제한점
  - 제 2절 집합 건물 환경의 경량 에너지관리에이전트 프로토콜 설계
  - 제 3절 집합 건물 환경의 경량 에너지관리에이전트 프로토콜 구현
- 제 4장. 집합 건물 환경의 경량 에너지관리에이전트 프로토콜 성능 분석
  - 제 1절 실험 환경
  - 제 2절 실험 구조와 시나리오 및 결과
- 제 5장. 결론 및 향후 과제

# Background



EMA = 에너지 관리 에이전트, Server EMA = SEMA , Client EMA = CEMA , Smart Appliance EMA= SEMA

# 연구 배경 및 필요성

-참고 : 수요반응 (DR - Demand Response) :증가하는 전력 수요와 피크부하에 대응하고 안정적인 전력망을 구축하기 위한 방법

## (1) 집합건물 환경의 에너지 관리의 필요성 : 블랙아웃(정전), 전력 수요 증가, 에너지 수요관리 대책 강구 집합 건물 환경의 지능화된 에너지 관리(수요반응)가 필요(주거공간의 에너지관리에이전트 집중화 필요)

- 여름철이나 겨울철 냉난방기기 판매량과 사용량 급증에 따른 블랙아웃은 심각한 문제를 발생
- 집합 건물과 일반 가정 사용량이 많게는 **피크 시간대 전력수요의 60%**를 차지함
- 수요반응, OpenADR** (Open Automated Demand Response)프로토콜 등장[2][3]

## (2) 국민 DR를 통한 수요반응 영역 확대 및 에너지 IoT[9], 경량수요반응 프로토콜의 등장

- (경량 OpenADR2.0b CoAP/JSON, MQTT/JSON [9][15] )
- 제한적인 IoT환경, (2)디바이스나 IoT기기, 게이트웨이 ->배터리와 제한된 리소스 [10][11][12]
  - 에너지관리에이전트 기술 →EIoT 기술 발전(에너지 분야와 IoT 분야와 융합)
  - Utility 회사 중심의 송배선 산업설비에서 에너지 관리 플랫폼인 에너지관리시스템(EMS)은 효율적 에너지 관리하기 위해 사용[5].

## (3) 에너지관리 에이전트 표준화 및 EMA Protocol 표준화가 진행

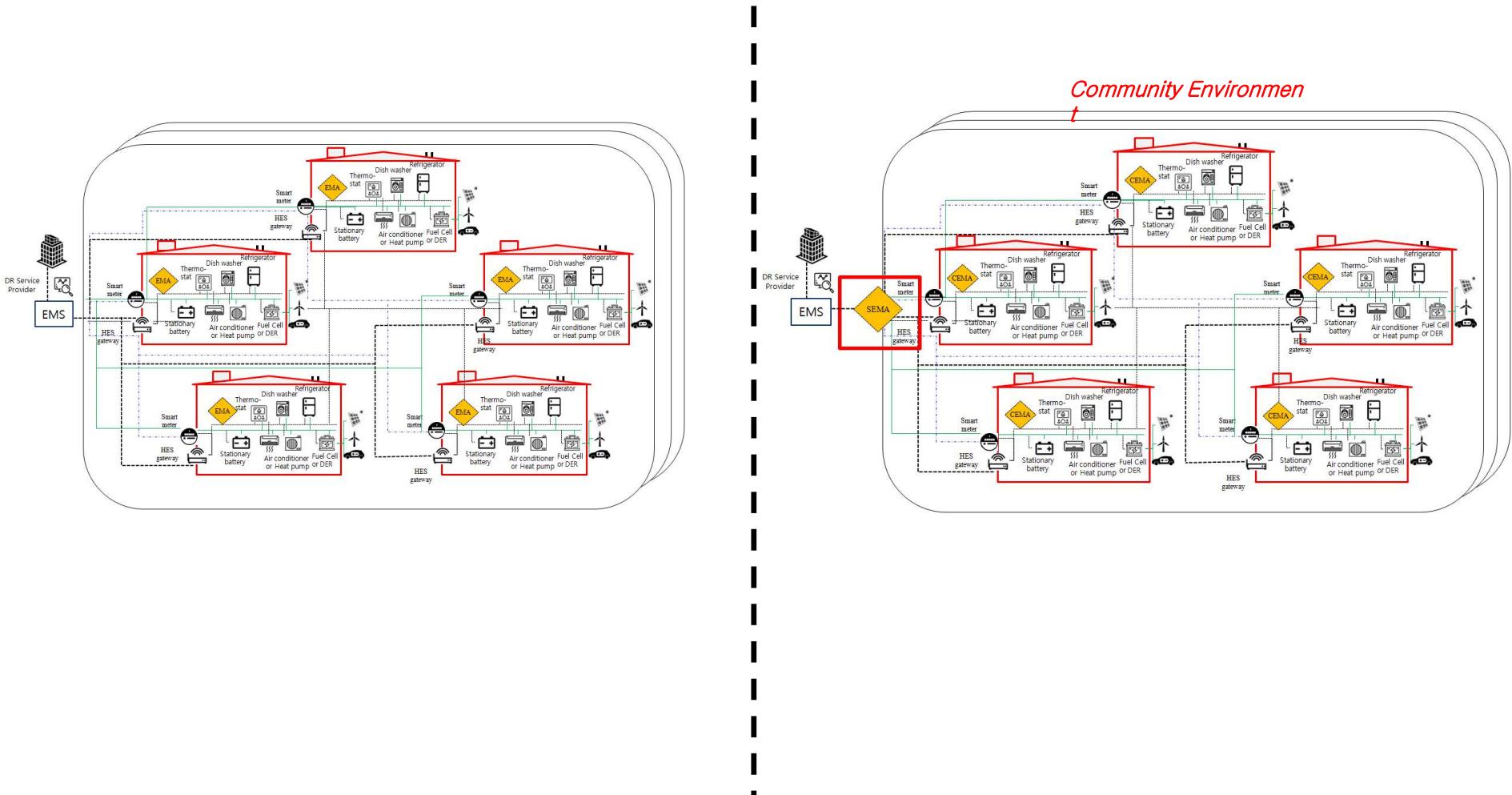
(ISO/IEC 15067-3, ISO/IEC 15067-3-3)[7][8]

- 주거지역, 집합건물, 대학캠퍼스, 빌딩 등에 Interacting EMA를 통해 에너지관리와 수요반응에 대한 표준화가 등장  
(ISO/IEC 15067-3-3의 Model of a system of interacting Energy Management Agents[8])
- 상호 작용하는 EMA는 주거지역, 집합건물, 대학캠퍼스, 빌딩 의 지역 사회의 에너지관리와 수요반응을 관리함.

## (4) 집합건물 내에 수많은 수요관리 고객을 수용할 수 있는 수요반응 프로토콜(EMAP)이 필요함

- 수요관리사업자가 모두 수요 관리해주기 위해서는 복잡성 증가(병목현상이나 서비스 지연으로 인한 블랙아웃 방지, 서버의 과부하 방지)
- 다단계 구조인 가정과 집합건물들에 대한 Aggregator가 필요(통합적 수요반응(집중화 필요))
- 기존 주거공간의 단일 에너지관리에이전트 → 다단계 서버/클라이언트 에너지관리에이전트의 개념이 필요
- 다양한 수요참여고객의 증가 및 집합건물 환경에서 다양한 서비스 필요**→ 에너지관리에이전트 간 경량 수요반응 프로토콜의 전송 메커니즘이 필요
- 수요반응 참여고객의 보안성 문제 등(기존 연구[30]는 Explicit 사용) → Implicit/Explicit 모니터링이 필요

# 연구 배경 및 필요성



# 연구 목적

## (1) 집합 건물 환경으로 수요반응을 하는 목적 (수요반응 영역 확대)

(1)집합 건물 환경으로 집중화 된 수요반응의 필요 : 블랙아웃(정전), 전력 수요 증가, 에너지 수요관리 대책 강구

집합건물 내부에도 대표적인 Aggregator(Server EMA)로 개인 영역인 EMA(Client EMA)들에게 수요 관리

(2)주거지역, 집합건물, 대학캠퍼스, 빌딩 등에 Interacting EMA를 통해 에너지관리와 수요반응에 대한 표준화가 등장

(ISO/IEC 15067-3-3의 Model of a system of interacting Energy Management Agents[7] )

- 분산형 구조, 메쉬형 구조인 주거지역, 집합건물, 등에 Interacting EMA를 통해 에너지관리에이전트 간 에너지관리와 수요 반응을 할 수 있는 구조를 기반

(3)4차 산업 혁명[16]- 인공지능, 빅데이터, IoT 기술이 등장함에 따라 집합 건물 내에서도 지능화된 에너지관리가 필요함

-집합 건물 내의 에너지관리에이전트는 게이트웨이나 스마트 IoT 디바이스(인공지능 스피커, 공유기 등)와 새롭게 결합하고 서로 간 서비스 결합

(4)국민 DR이 확산 됨에 따라서 수요반응의 영역이 확대[16] 되고 있음

에너지관리에이전트 프로토콜은 스마트 홈과 스마트 공장 등에서 스마트 시티로 확대 적용

## (2) 에너지 관리 에이전트 프로토콜(EMAP)의 목적

(1)국민 DR의 정책으로 인해 수많은 집합건물의 수요관리 고객에 대한 실시간 수요반응이 필요

하며 효율적으로 에너지를 관리하기 위해 경량 수요반응 프로토콜이 필요

(2)주거공간 내 클라이언트 에너지관리에이전트는 집합 건물의 충괄적 수요반응을 담당하는 서버 에너지관리에이전트와 통신을 통해 수요반응 및 스케줄링을 할 수 있음

(3)에너지관리에이전트 프로토콜의 확장성을 높이고 각 가정의 독립적 사용과 프라이버시를 보장하기 위한 Implicit/Explicit 방법을 제시

(4)집합 건물 내의 Client EMA는 Gateway나 IoT 디바이스에 모듈 형식으로 탑재됨

- Heavyweight인 HTTP/XML 방식의 수요반응 프로토콜은 제한이 있음(OpenADR2.0b, SEP2.0)

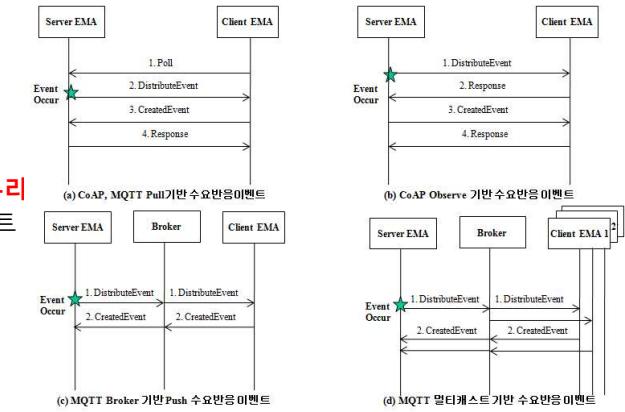
- 디바이스나 IoT기기, Gateway들에는 제한된 리소스 및 배터리 용량이 제한[10][11]

- IoT(CoAP, MQTT) 프로토콜이 lightweight, 저전력 기반[12]

(5)제한적인 IoT환경[12]에서 데이터 트래픽을 줄이기 위해서 IoT 프로토콜의 메커니즘들이 필요

- 집합건물의 수많은 개인지역으로 확대, 효율적 수요반응 관리가 필요하게 됨(다양한 집합건물의 구조에서 유리)

- 1. CoAP, MQTT 기반 lightweight, 2. CoAP Obs, 3. MQTT Broker 기반 Push 메커니즘, 4. MQTT 기반 멀티캐스트



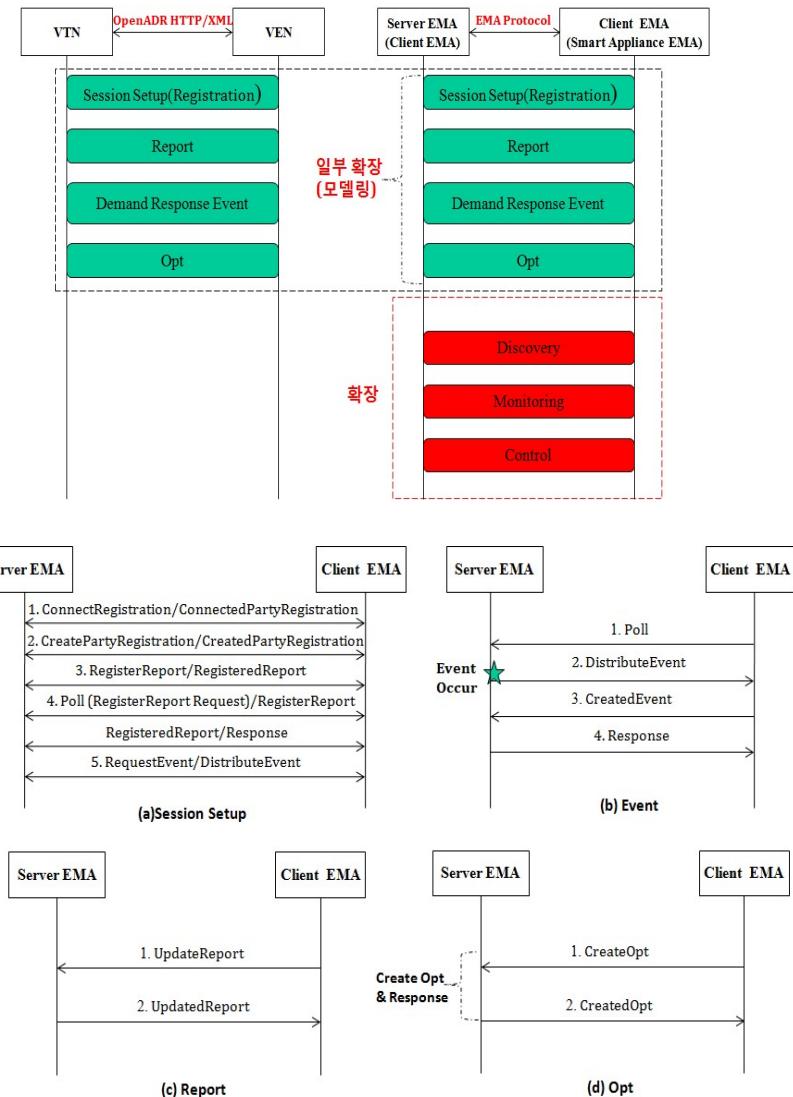
# 문제점 및 해결책(EMAP설계-표로 정리)

3장 기존 주거 공간의 수요반응 프로토콜의 제한점

→ 집합 건물 환경의 경량 에너지관리에이전트 프로토콜 설계

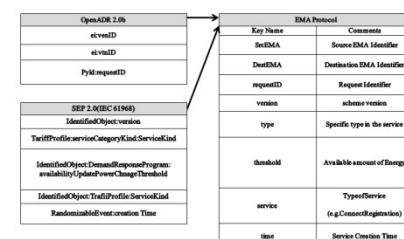
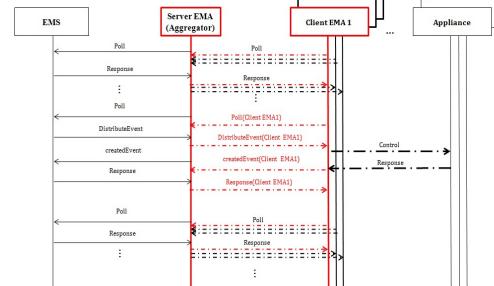
문제점 및 해결책	문제점	해결책
(1)Heavyweight 수요반응 프로토콜의 제한점 -집합 건물 환경을 위한 경량에너지관리에이전트 프로토콜	IoT 환경→저전력이고 소형화 된 네트워크 장비 <b>-HTTP 수요반응 프로토콜은 Heavyweight 해서 많은 리소스와 데이터 트래픽이 발생.</b> XML은 많은 프로세싱 오버헤드,파싱 과정이 많음. Push 메커니즘/네트워크 방화벽 문제	-경량 수요반응 프로토콜 및 JSON 사용 (CoAP, MQTT/JSON) -UpdateReport 실험 →CoAP, MQTT/JSON 경량프로토콜 데이터 트래픽 ,지연시간 감소
(2)Scalability 한계 및 수요 반응의 문제점 -통합형 에너지관리에이전트프로토콜	수요반응에 대한 효율적 수요 관리 및 분배 필요, 보안성 문제-트래픽 발생 <b>(예 : UpdateReport)</b> 수요반응사업자로 부터 받은 수요반응 이벤트를 주거공간의 에너지관리에이전트가 Control만 지시 IoT 디바이스들에 Control 지시 (SEP2.0도 다이렉트 Control)	서버 에너지관리에이전트의 Aggregator 역할 다단계 에너지관리에이전트 설계로 인한 데이터 트래픽 감소, 병목 현상이 감소
(3)에너지관리에이전트의 Explicit 모니터링 문제점 -Implicit/Explicit 모니터링	모든 디바이스의 정보를 받기 위해서는 응답시간이 오래 걸리고 디바이스를 개수가 증가하게 되면 데이터 트래픽이 늘어 병목현상이 발생 기존 연구는 디바이스 정보가 모두 보내기 때문에 보안성 문제	에너지관리에이전트 프로토콜의 Implicit/Explicit 기반 모니터링 -데이터 트래픽의 단점을 극복 Implicit 방식 : 보안성(프라이버시) 향상을 통해 증명
(4) 주거 공간 환경에서 수요반응 Event 서비스의 제한점 -수요반응 Event 메커니즘을 이용한 에너지관리에이전트 프로토콜	제한적인 IoT환경에서 집합건물 내의 고객의 요구에 맞춘 수요 관리가 필요 (보안성 문제, 네트워크 방화벽문제, 그룹화정보가 없어 유니캐스트 전송 문제 데이터 트래픽 문제)	경량 프로토콜인 CoAP, MQTT를 사용 (EMAP 프로토콜 설계 및 구현) 경량 수요반응 프로토콜의 다양한 메커니즘 사용 (다양한 시나리오 제시) 실험을 통해 증명

# (1)구현 : 에너지관리에이전트 프로토콜의 수요반응 4가지 서비스 (차) OpenADR2.0과 비교



- (a)Session Setup(Registration 과정)
  - 에너지관리에이전트 간 서로 연결을 수립하고 Report을 교환할 때 기존의 Report에 대한 에너지 가격 정보나 클라이언트 에너지관리에이전트의 에너지 관리 정보를 얻음.
- (b)Event(Demand Response Event)
  - 에너지관리에이전트 프로토콜은 다양한 이벤트 종류가 있다. 예로 가격기반 수요반응 이벤트에서 Initial Price, Incentive Price, Negotiation Price가 있다. 서버 에너지관리에이전트는 실시간적으로 가격에 대한 정보를 이벤트로 발생
- (c)Report(updateReport)
  - 에너지관리에이전트 간 서로 연결을 수립 할 때 Report을 교환할 때 실시간 에너지에 대한 가격 정보나 클라이언트 에너지관리에이전트의 디바이스 정보 등을 얻음.
- (d)Opt(수요반응 가용상태 및 스케줄링)
  - 에너지관리에이전트 프로토콜의 Opt는 클라이언트 에너지관리에이전트가 상위 서버 에너지관리에이전트에게 수요반응 이벤트의 가용상태 또는 수요반응 이벤트 프로그램 변경, 수요반응 스케줄링을 요청을 알려주는 서비스

# 경량에너지관리에이전트 프로토콜 구현

구현 내용	그림	구현
(1) 경량 수요반응 프로토콜 기반 OpenADR 2.0b와 에너지관리에이전트 프로토콜의 4가지 수요반응 서비스 구현		(+4가지 설명) 에너지 관리 에이전트 프로토콜(EMAP)가 다른 수요반응 프로토콜( <b>OpenADR2.0b</b> ) <b>SEP 2.0 의 데이터모델링</b> 을 따랐기 때문에 <b>수요반응 프로토콜과 상호운용성</b> 이 있음 OpenADR 2.0b의 수요반응 명령어들과 Smart Energy Profile 2.0 (SEP2.0)의 과금, 제어 및 상태 정보에 대한 데이터모델링을 사용
(2) 서버 에너지관리에이전트 (Aggregator)의 기능적 중계 기능		1. 서버 에너지관리에이전트는 에너지관리시스템으로부터 <b>수요반응 이벤트를 중계</b> 해서 클라이언트 에너지관리에이전트에게 바로 전달 2. <b>서버 에너지관리에이전트의 자체적인 수요반응 스케줄링</b> 통해 클라이언트 에너지관리에이전트에게 수요반응 이벤트를 내림 (외부 그리드와 통합적 수요반응 가능)
(3) Implicit/Explicit 모니터링 통신 기능	<p>(a) Implicit</p> <pre>Implicit Object{     "SrcEMA": String,     "DestEMA": String,     "Type": String,     "EMANUM": Integer,     "EMAID": String,     "Power": double,     "Margin": double,     "Generate": double,     "Storage": double,     "Time": Date }</pre> <p>(b) Explicit</p> <pre>Explicit Object{     "SrcEMA": String,     "DestEMA": String,     "Type": String,     "EMANUM": Integer,     "EMAID": String,     "Power": double,     "Margin": double,     "Generate": double,     "Storage": double,     "Time": Date,     "Explicit": [         {"DeviceEMID": Integer, "Sort": "LED", "Power": double, "State": String, "Dimming": Integer, "Priority": Integer},         {"DeviceEMID": Integer, "Sort": "PV", "State": String, "Power": double, "Priority": Integer},         {"DeviceEMID": Integer, "Sort": "ESS", "Mode": Integer, "Power": double, "State": On, "Capacity": double, "Soc": double, "Vol": double, "Hz": double, "ChargedEnergy": double, "Priority": Integer},         {"DeviceEMID": Integer, "Sort": "Recloser", "State": String, "Power": double, "Vol": double, "Hz": double, "Priority": Integer},         {"DeviceEMID": Integer, "Sort": "Resource", "State": String, "Power": double, "Priority": Integer}     ] }</pre>	<b>-Implicit 메시지 방식</b> (1) 모니터링에 있어 사용자가 누군지 모르고 디바이스 제어 같은 의사 결정이 어려움 (2) 광범위한 스마트 홈 네트워크 환경에서 사용자가 얼마나 많을지 모름 (3) 데이터 트래픽 양이 적고 보안성은 높기 때문에 공격자의 침입을 막을 수 있음  <b>-Explicit 메시지 방식</b> (1) 상위 서버 관리에이전트나 에너지관리시스템에서는 <b>모든 디바이스 데이터가 있음</b> - 통계적 분석과 빅데이터 분석을 할 수 있는 장점이 있음.

# 실험 구조 및 환경

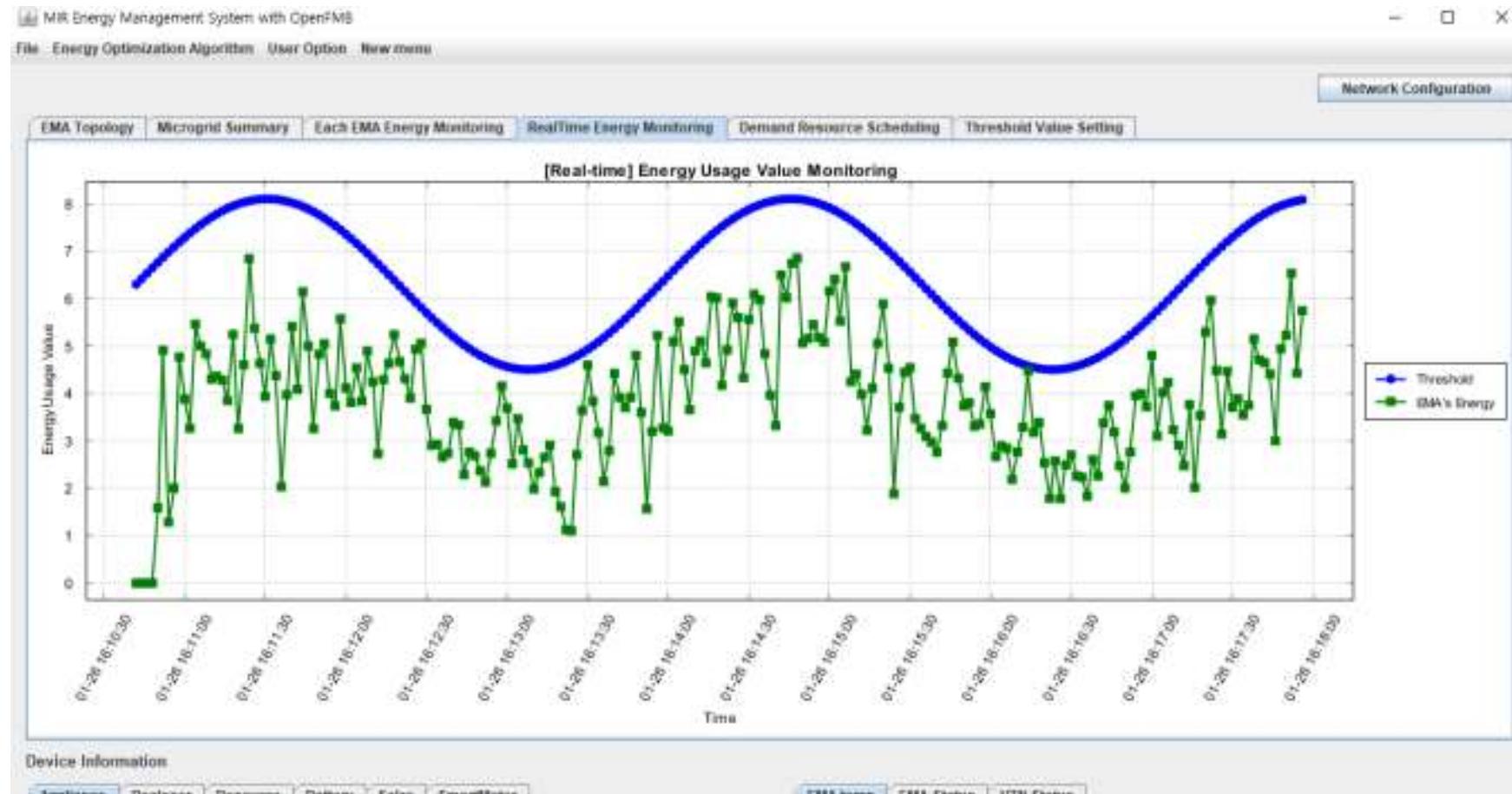


Client EMA(OpenWRT)



Smart Home(Appliances)

# 실험 구조 및 환경



Server EMA

# 경량에너지관리에이전트 프로토콜 실험

실험 내용	그래프	실험 결과
(1) 경량 수요반응 프로토콜의 데이터 트래픽 및 자연 시간 분석 (HTTP과 비교)		<p>초경량 수요반응 프로토콜인 MQTT, CoAP보다 20대일 때 2~3배 트래픽 양이 많음 지연시간은 HTTP가 가장 많으며 CoAP가 가장 적음 <b>IoT 기반의 경량 수요반응 프로토콜(MQTT, CoAP)이 기존 HTTP/XML의 수요반응 프로토콜보다 지연시간과 트래픽 측면에서 우수</b></p>
(2) 집합 건물 내 이벤트 수요반응 전송 메커니즘 비교 실험		<p>PPT : P.12 설명</p>
(3) 에너지관리에이전트의 Implicit/Explicit 통신 비교 실험		<p>Interval Time이 매우 짧은 250ms에서 Packet 량은 제안하는 보고 방법이 약 1/2인 것을 확인하는 EMA 1( Device 5대) – EMA 20 ( Device 100대)에서 측정한 결과이며 <b>Device의 개수가 늘어 날 수록 제안하는 방식은 기존방식과 비교하였을 때 크게 감소할 것으로 예상</b></p>

# 경량에너지관리에이전트 프로토콜 실험

\*집단 건물 내 다양한 수요반응 전송 메커니즘 필요성 정리 및 실험표

실험 시나리오	CoAP/JSON	MQTT/JSON	실험 결과	결론
1. 평소 Pull 기반 수요반응 서비스 트래픽 비교 <b>(UpdateReport 실험)</b>	Pull 기반 CoAP/JSON	Pull 기반 MQTT/JSON	(SEMA에서 트래픽측정) CoAP와 MQTT와 트래픽 양은 비슷했음	리소스가 제한적인 집합건물에 경량 수요반응 프로토콜MQTT/JSON, CoAP/JSON이 필요
2. 수요반응 Event 을 한대에 내릴 때(Pull)	Pull 기반 CoAP/JSON	Pull 기반 CoAP/JSON	Event 응답 시간은 HTTP/XML<MQTT/JSON < <b>CoAP/JSON</b> 순서로 빠름(CoAP가 MQTT보다 1.5배 빠름)	리소스가 제한적인 집합건물에 경량 수요반응 프로토콜 <b>MQTT/JSON, CoAP/JSON</b> 이 필요
3. 수요반응 Event 을 한대에 내릴 때(Push)	CoAP Observe	Broker 기반 MQTT Push	20대 일때 MQTT Push Un보다 <b>CoAP Observe(Push)</b> 응답속도가 81배 빠름	집단 건물 내의 많은 EMA가 있고 <b>긴급한 실시간 수요반응</b> 프로토콜은 <b>CoAP/JSON</b> 이 적합.
4.수요반응 Event 을 모든 EMA에 내릴때	CoAP Observe 유니캐스트	Broker 기반 MQTT Push 유니캐스트 / Broker 기반 MQTT Push 멀티캐스트 메커니즘	20대 일때 <b>MQTT Push Unicast</b> 보다 <b>CoAP Observe(Push) Unicast</b> 가 응답속도가 3.4배 빠름 20대 일때 <b>MQTT/JSON Push Multicast</b> 보다 <b>CoAP Observe(Push) Unicast</b> 가 응답속도가 2.5배 빠름	-집단 건물 내의 많은 CEMA가 있고 <b>여러 개의 종복 메시지를 전달 할 때 CoAP/JSON이 적합.</b> (하지만, 집합 건물의 SEMA가 부하가 많을 때 MQTT Broker 기반의 MQTT/JSON Push Multicast을 쓰면 보내는 쪽에서 1/N(CEMA)로 트래픽이 적음 )

# 결론 및 향후 과제

## - 본 논문의 결론

- (1) EMA를 통해 집합 건물 환경에서도 지능화된 에너지관리인 수요반응을 함
- (2) 에너지 관리 에이전트 프로토콜을 통해 수많은 집합건물의 수요관리 고객에 대한 실시간 수요반응을 할 수 있음(Push 실험을 통해 증명)
- (3) 집합 건물 환경에서 에너지관리에이전트 간 통신을 통해 수요반응 및 통합적 스케줄링을 할 수 있음
- (4) 집합 건물 환경에서 EMA 프로토콜을 경량 IoT 프로토콜인 CoAP, MQTT 이용
  - Heavyweight인 HTTP/XML 방식의 수요반응 프로토콜은 제한이 있음(OpenADR2.0b, SEP2.0)
  - 디바이스나 IoT기기, Gateway들에는 제한된 리소스 및 배터리 용량이 제한[10][11], IoT(CoAP, MQTT) 프로토콜이 lightweight, 저전력 기반[12]
- (5) 제한적인 IoT환경[12]에서 데이터 트래픽, 수요반응 전송 속도를 줄이기 위해서 IoT 프로토콜의 메커니즘 비교를 통해 적합한 프로토콜인 CoAP/JSON 이용- (이유 : 집합건물의 수많은 개인지역으로 확대(수많은 노드), 효율적 수요반응 관리가 필요)
  - 1. CoAP, MQTT 기반 lightweight, 2. CoAP Obs, 3. MQTT Broker 기반 Push 메커니즘, 4. MQTT 기반 멀티캐스트
  - CoAP 가 실시간 수요반응의 효율성이 있음을 증명
- (6) 기존연구의 Explicit모니터링과 EMAP의 Implicit/Explicit 모니터링 비교를 통해 데이터 추상화 기능 의 필요성을 증명

## -본 논문의 향후 과제

- 에너지관리에이전트 프로토콜은 스마트 홈과 스마트 공장 등에서 **스마트 시티**로 확대 적용  
(스마트시티 기반의 에너지관리에이전트를 분산전원 및 Advanced Metering Infrastructure(AMI)과 결합 하는 추가적 연구가 필요)
- 다양한 IoT 기술 및 플랫폼과 결합하여 IoT 기기들과 MQTT, CoAP를 이용하여 통합 서비스 프레임워크를 구축
- 4차 산업 혁명이 등장함에 따라 집합 건물 내의 에너지관리에이전트는 **홈 게이트웨이나 스마트 IoT 디바이스**(인공지능 스피커, 공유기 등)와 새롭게 결합하고 서로 간 서비스 결합이 활발히 발생

# 참고문헌

- [1] Paulson Institute, Demand Response, 2015
- [2] OpenADR Demand Response Program Implementation Guide, OpenADR Alliance.
- [3] OpenADR, <http://www.openadr.org/>
- [4] Korea Power Exchange(KPX), Demand Resource Trading Market 'National DR Conference'('국민DR 컨퍼런스')
- [5] Wei, Min, Seung Ho Hong, and Musharraf Alam. "An IoT-based energy-management platform for industrial facilities." *Applied energy* 164 (2016): 607–619.
- [6] D. Delphine, B. W. Jang, Y. S. Shin, S. T. Kang, and J. S. Choi, "Design and implementation of building energy management system with quality of experience power scheduling model to prevent the blackout in smart grid network," 16th Int. Conf. Advanced Commun. Technol., pp. 208–211, Pyeongchang, Feb. 2014.
- [7] ISO/IEC 15067-3, "Model of a demand response energy management system for HES", 2012. 7.
- [8] Information technology – Interconnection of information technology equipment – Home Electronic System – Application models – Part 3-3: Model of a system of interacting Energy Management Agents (EMAs) for demand response energy management, ISO/IEC 15067-3-3, 2017
- [9] 박현일, 김세영, 강성철, 박현진, 김일연, 최진식. (2017). 스마트 에너지 IoT 를 위한 CoAP 기반 Lightweight OpenADR2.0b 프로토콜의 구현 및 분석. *한국통신학회논문지*, 42(4), 904–914
- [10] Bandyopadhyay, Soma, and Abhijan Bhattacharyya. "Lightweight Internet protocols for web enablement of sensors using constrained gateway devices." *Computing, Networking and Communications (ICNC), 2013 International Conference on*. IEEE, 2013.
- [11] De Caro, Niccolò, et al. "Comparison of two lightweight protocols for smartphone-based sensing." *Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium on*. IEEE, 2013.
- [12] Al-Fuqaha, Ala, et al. "Internet of things: A survey on enabling technologies, protocols, and applications." *IEEE Communications Surveys & Tutorials* 17.4 (2015): 2347–2376.
- [13] ISO/IEC 20922:2016, Information technology -- Message Queuing Telemetry Transport (MQTT) v3.1.1
- [14] Shelby, Zach, Klaus Hartke, and Carsten Bormann. "The constrained application protocol (CoAP)." (2014).
- [15] 박현일, 박현진, 이성환, 최진식. (2018). 에너지 IoT 환경을 위한 MQTT 기반의 Lightweight 수요반응 프로토콜 구현 및 분석, *한국통신학회논문지*
- [16] 혁신성장을 위한 사람 중심의 4차 산업혁명 대응계획(최종), 과학기술정보통신부, 2017. 11. 30
- [17] 최동배, 스마트그리드의 기본개념과 최근동향 스마트리드, 인포더북스, 2016.6.24.
- [18] 전력거래소, <http://www.kpx.or.kr/>, 2018.5.8
- [19] 분산전원, <http://terms.tta.or.kr/>, TTA정보통신용어사전, 2018.5.8
- [20] OpenFMB, <https://openfmb.github.io>, 2018.5.8
- [21] Ghatikar, Girish, and Rolf Bienert. "Smart grid standards and systems interoperability: a precedent with OpenADR." *proceedings of the Grid Interop Forum*. 2011.
- [22] 전력거래소, 수요반응자원 전력거래시스템(OpenADR 2.0b VEN Guide), 2015.11.6
- [23] Zigbee alliance, [www.zigbee.org](http://www.zigbee.org)
- [24] Homeplug alliance, [www.homeplug.org](http://www.homeplug.org)

# 참고문헌

- [25]Smart Energy Profile 2.0 Application Protocol Specification, Zigbee Alliance and Homeplug Alliance, 2013.
- [26]Smart Energy Profile Marketing Requirements Document (MRD), Zigbee Alliance and Homeplug Alliance, 2009.
- [27]Smart Energy Profile Technical Requirements Document (TRD), Zigbee Alliance and Homeplug Alliance, 2012.
- [29]텔핀, 최진식.(2015), 스마트 그리드에서 블랙아웃을 방지하기 위한 에너지 스케줄링 스키마에 의한 성능분석, 2015.2, 한양대학교 일반대학원 컴퓨터소프트웨어학과 석사학위논문
- [29]장병욱, 최진식.(2016), OpenADR 기반의 실시간 VEN 스마트 게이트웨이 프로토콜 개발과 에너지 최적화, 2016.8, 한양대학교 일반대학원 컴퓨터소프트웨어학과 석사학위논문
- [30]김세영,최진식.(2017), 에너지 IoT 환경에서 효율적인 에너지 관리를 위한 분산형 스마트 홈 에너지관리시스템 구현 및 분석에 대한 연구, 2017.8, 한양대학교 일반대학원 컴퓨터소프트웨어학과 석사학위논문
- [31]강성철,최진식.(2018), 주거 공간 에너지 수요관리를 위한 MQTT기반의 수요반응 프로토콜 설계 및 구현, 2018.2, 한양대학교 일반대학원 컴퓨터소프트웨어학과 석사학위논문
- [32]Yassine, Abdulsalam. "Implementation challenges of automatic demand response for households in smart grids." Renewable Energies for Developing Countries (REDEC), 2016 3rd International Conference on. IEEE, 2016.
- [33]Koh, Jason, Steven Ray, and Jack Hodges. "Information Mediator for Demand Response in Electrical Grids and Buildings." Semantic Computing (ICSC), 2017 IEEE 11th International Conference on. IEEE, 2017.
- [34]Vyas, D., and H. Pandya. "Advance metering infrastructure and dlms/cosem standards for smart grid." International Journal of Engineering Research & 1.1 (2012).
- [35]Katipamula, Srinivas, et al. "VOLTTRON: An open-source software platform of the future." IEEE Electrification Magazine 4.4 (2016): 15–22.
- [36]Son, Young-Sung, et al. "Home energy management system based on power line communication." IEEE Transactions on Consumer Electronics 56.3 (2010).
- [37]Pipattanasomporn, Manisa, Murat Kuzlu, and Saifur Rahman. "An algorithm for intelligent home energy management and demand response analysis." IEEE Transactions on Smart Grid 3.4 (2012): 2166–2173.
- [38]Chang, Sheng-Fa, et al. "Application and Development of Zigbee Technology for Smart Grid Environment." Journal of Power and Energy Engineering 3.04 (2015): 356.
- [39]박현진, 박현일, 강성철, 김세영, 이성환, & 최진식. (2017). 스마트 에너지 IoT 환경에서 OpenADR 2.0 b 수요반응의 Push 메카니즘 필요성 연구. 한국통신학회 학술대회논문집, 283–284.
- [40]박현진, 박현일, 이성환, 최진식.(2018) 에너지 IoT 환경에서 실시간 수요반응 서비스를 위한 CoAP Observe 기반 Push Mechanism 설계 및 분석. 한국통신학회논문지, 43(3), 529–540
- [41]분산형 에너지관리시스템 모델 정의 및 인터페이스에 대한 통신구현에 대한 단체표준 제정(SGSF-043-1:2017, 에너지관리에이전트 통신 요구사항)
- [42]Wilson, Charlie, Tom Hargreaves, and Richard Hauxwell-Baldwin. "Smart homes and their users: a systematic analysis and key challenges." Personal and Ubiquitous Computing 19.2 (2015): 463–476.
- [43]Jawurek, Marek, and Felix C. Freiling. "Privacy threat analysis of smart metering." Proceedings of the 41th Annual Conference of the Gesellschaft fur Informatik eV (GI), GI-Edition. Vol. 267. 2011

## 2. 스마트 옷장 목차

- 개발 배경
- 소개



# 개발 배경

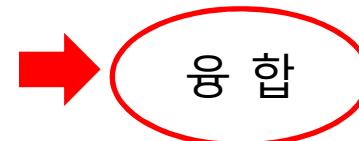
현대인들의 바쁜 생활로  
부가적인 **자기관리를 소홀히** 하게 됨

현대인들은  
자신을 꾸미는 욕구가 높아짐  
**의류에 대한 관심이 많아짐**

하지만, 의류에 대한 관리소홀  
옷장 안의 버리는 옷이 많아짐  
개인이 의류를 관리 못해 (기억력)  
한 해 버려지는 의류만 **6만 톤** 이 넘음

**스마트 홈**의 발전 - 옷장에 대한 발전이 필요

어플리케이션 상품 개발  
+ 패션 의류 분야

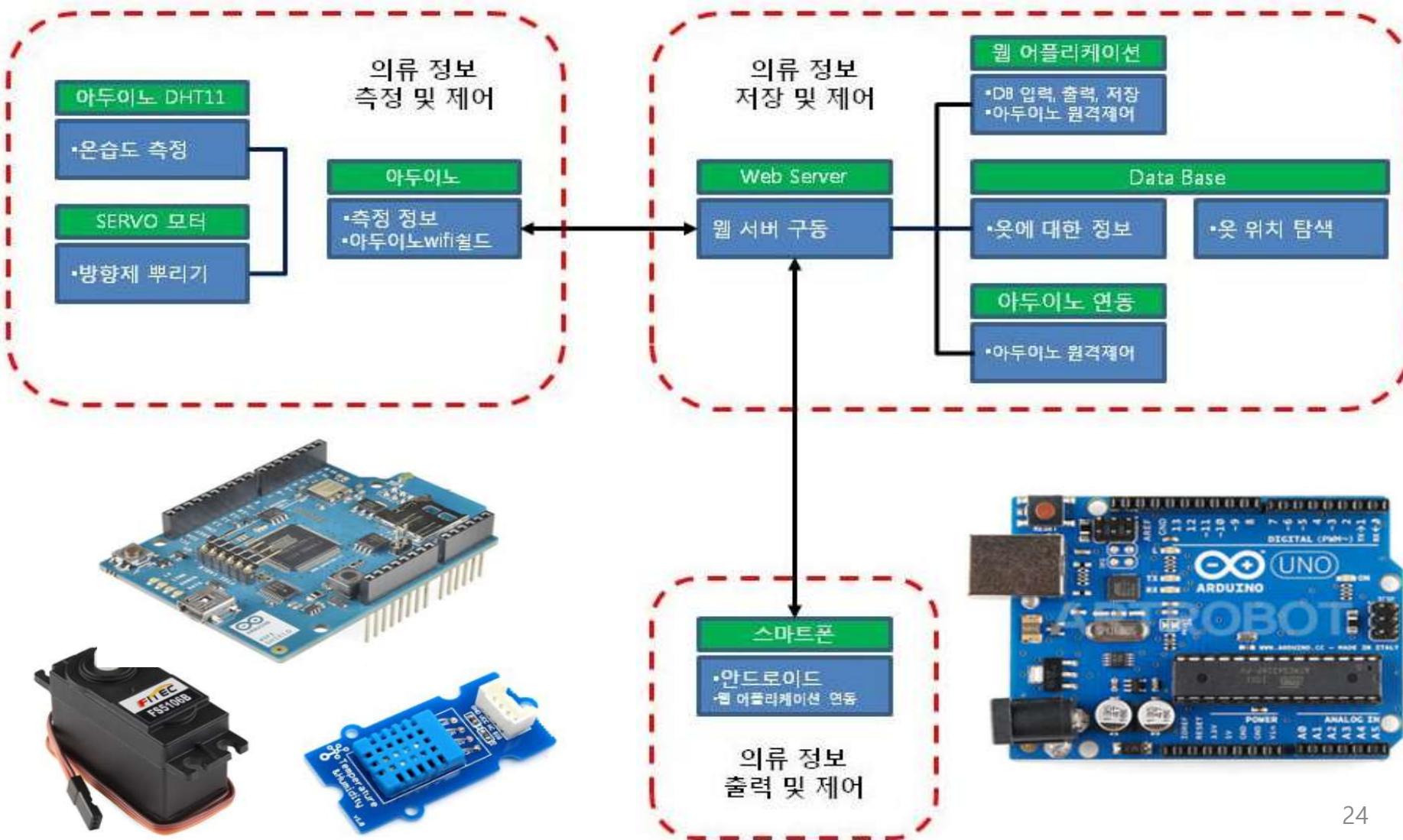


# 소개

스마트 옷장 소개

스마트 옷장 시스템 구성

스마트 옷장 어플리케이션 기능



# 소개

스마트 옷장 소개

스마트 옷장 시스템 구성

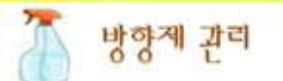
스마트 옷장 어플리케이션 기능



의류 관리 정보

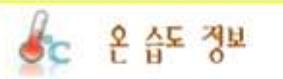
의류 검색 기능

의류를 데이터베이스에 사진과 의류의 특성을 저장하고 검색할 수 있으며 의류를 체계적으로 관리하는 기능



방향제 관리

옷의 냄새를 제거하기 위해서 스마트 폰 어플리케이션으로 원격 제어하는 기능

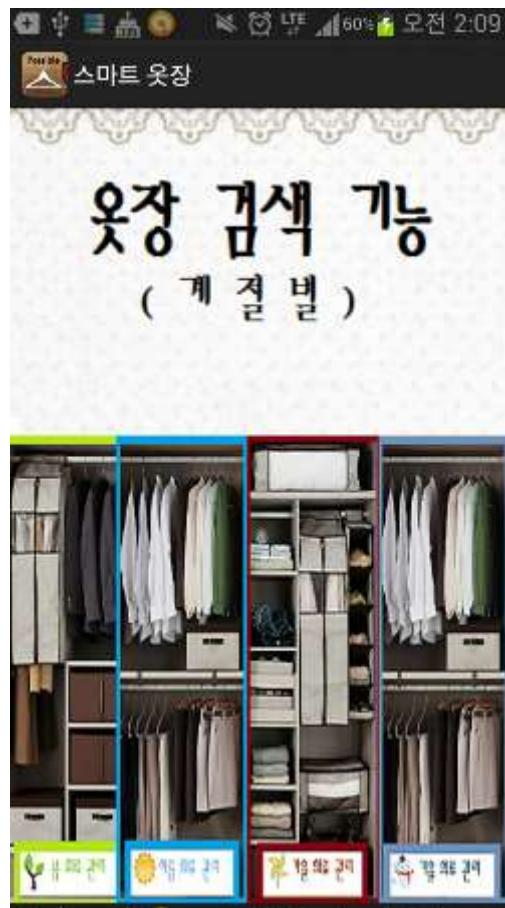


온습도 정보

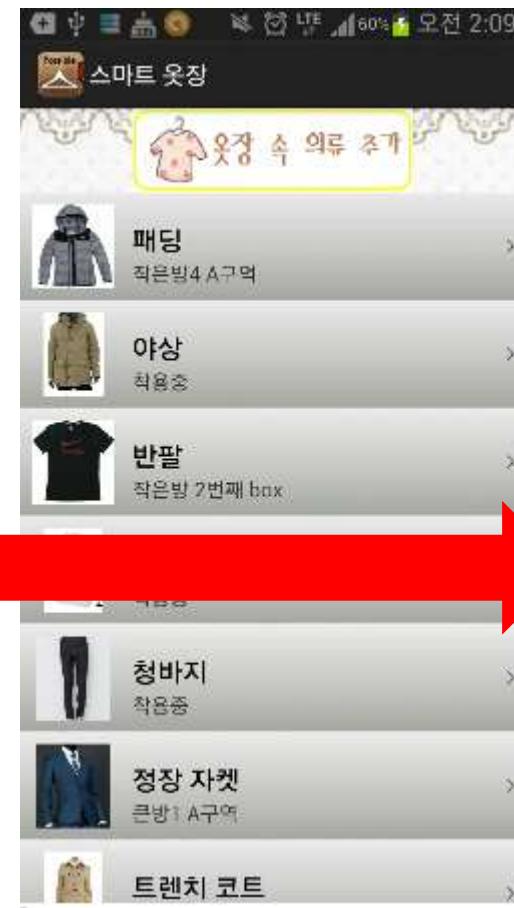
옷장의 온습도 관리 및 제습제 관리 곰팡이의 피해를 막을 수 있는 기능

# 소개

스마트 옷장 소개



스마트 옷장 시스템 구성



스마트 옷장 어플리케이션 기능



옷장에서  
여름 의류 관리  
버튼을 클릭하면  
전체 목록에서  
여름옷만 검색됨  
(다른 계절도)

# 소개

스마트 옷장 소개

스마트 옷장 시스템 구성

스마트 옷장 어플리케이션 기능

The image displays three screenshots of the Smart Closet system interface:

- Smart Closet Database Management (Left):** Shows the SQLite Database Browser with a table named "smarts". The columns are id, name, location, season, number, day, brand, and color. Data rows include items like Padding, Zipper Trouser, etc. A sidebar on the left lists categories: 의류명 (highlighted with a red box), 의류 위치, 입을 계절, 입은 횟수, 구매일자, 브랜드명, and 의류 색상.
- Smart Closet System Configuration (Middle):** Shows a list of clothing items with their details. An arrow points from the "Edit" button in the sidebar of the first screenshot to this screen, indicating the connection between database management and configuration.
- Smart Closet Application Function (Right):** Shows a user interface for viewing and managing closet items. It includes sections for 정장 자켓, 의류 위치, 큰방1 A구역, 입는 계절, and spring. A sidebar on the right shows item details with buttons for Edit and Delete (highlighted with a red box).

**7가지 항목과  
의류 사진이  
데이터베이스  
저장 (DDMS)  
에뮬레이터로**

**DB파일  
관리 가능**

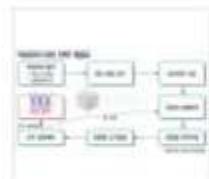
**의류 관련 정보보기와  
의류수정과 삭제도 가능**

### 3. 빅데이터 관련 논문(학사)

- 연구 배경
- 소개
- 결론 및 시사점

## 연구 배경

1. 기존 관계형 데이터베이스 시스템(RDBMS)기반 로그 처리 시스템이 Hadoop 기반 BigData(대용량)플랫폼인 시스템으로 넘어가고 있음
2. 빠르게 대용량의 웹 로그를 처리/분석해야 되는 이유는 선거기간이나 짧은 시간 내에 통계를 내야 되는 마케팅처럼 대량으로 들어오는 유저의 성향을 파악하기 위해서 속도가 빠른 BigData(대용량)시스템이 필요  
(미래의 선거는 빅데이터가 당락이 좌우- 오바마 캠프 빅데이터 선거 전략)

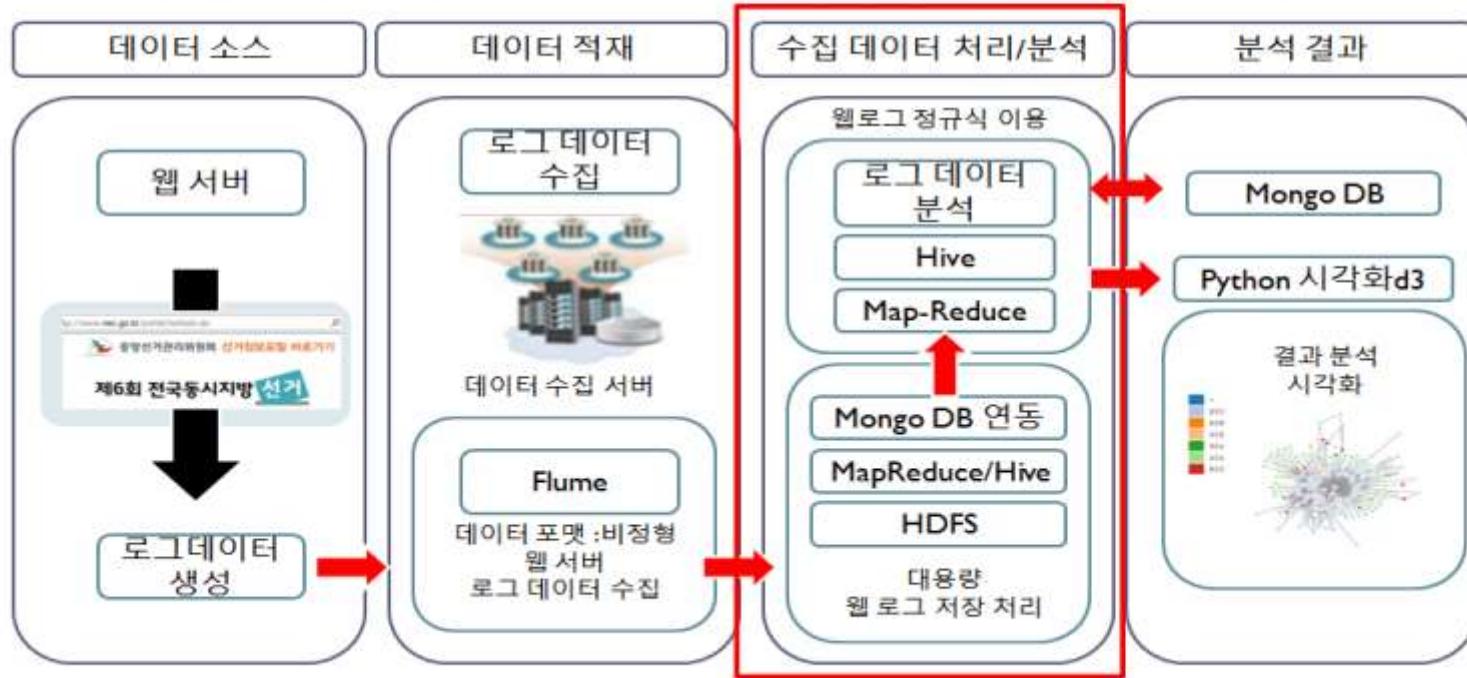


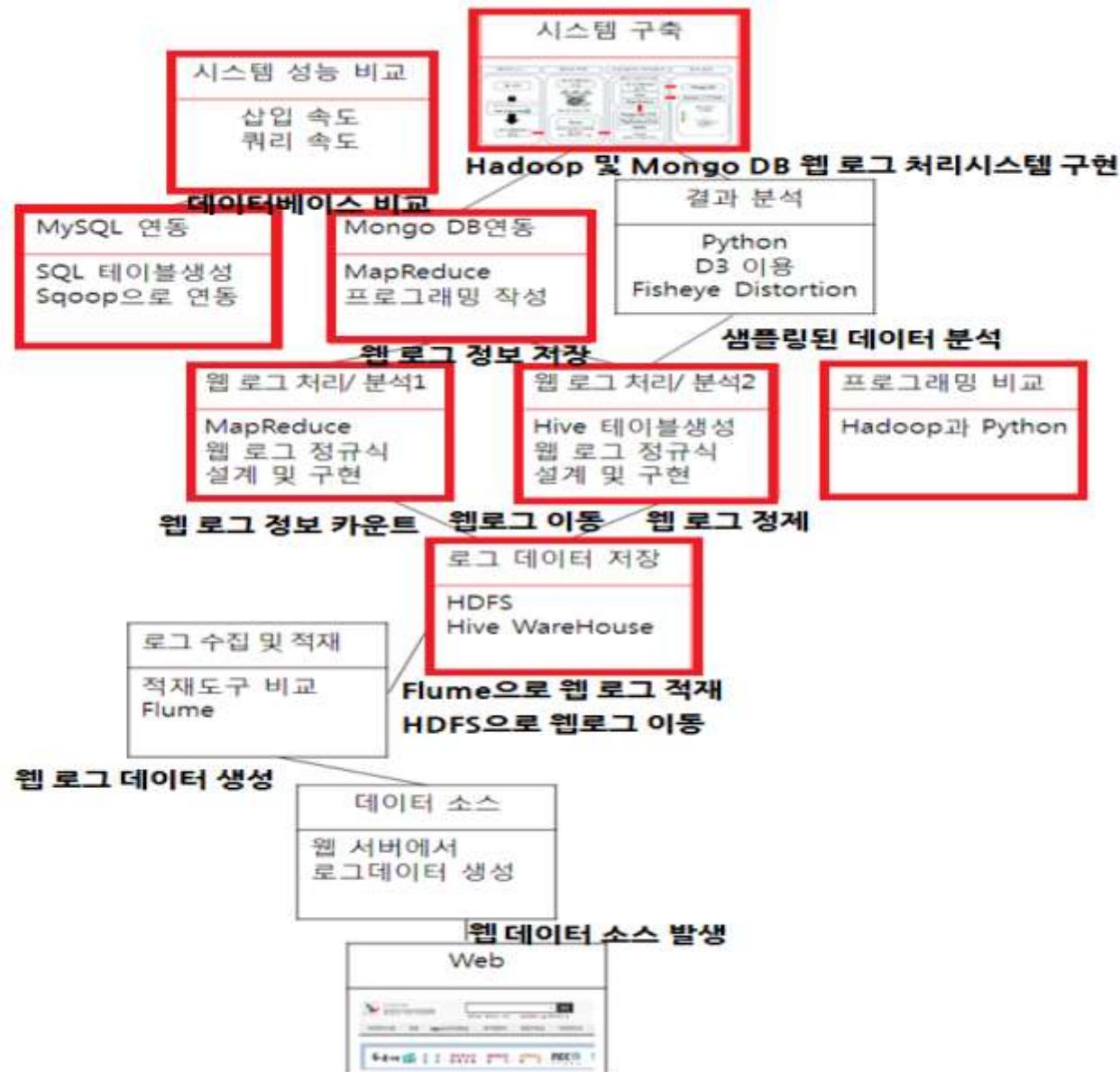
[6.4 지방선거 빅데이터 활용 '주목'](#) | 2014.03.17 | 디지털타임스 | 미디어다음

데이터 분석에 기초한 '빅데이터 선거'는 2012년 미국 대통령 선거에서 관심을 받기 시작했다. 당시 미국 언론과 전세계는 바락 오바마 미국 대통령이 재...

- ▶ 연구 주제
  - MongoDB 및 Hadoop 기반 대용량 웹 로그 처리 시스템 설계 및 구현

- ▶ 연구 내용
  - 웹 로그 처리 / 분석 중심 시스템 설계 및 구현
  - Hive, MySQL, MongoDB 기반 시스템 성능 평가(Insert, 질의문)





---

## 결론 도출

1. 기존 관계 형 데이터베이스 시스템(RDBMS)의 단점을 극복할  
로그 처리 최적화된 방안으로 Hadoop 및 Mongo DB기반으로 한  
대용량 웹 로그 처리 시스템을 설계 및 구현을 하였음
2. Hadoop 및 MySQL 기반 시스템보다 Hadoop 및 MongoDB 기반 시스템이  
삽입과 질의문의 속도가 저용량 일 때는 차이가 안 나지만 대용량일 때  
성능차이가 많이 나는 것을 알 수 있음  
(이후에 쿼리 문을 여러 번 돌려서 신뢰성 있는 성능평가와  
D3으로 시각화 하는 것에 부족하여 초점을 맞추어 더 연구 예정 )

## 시사점

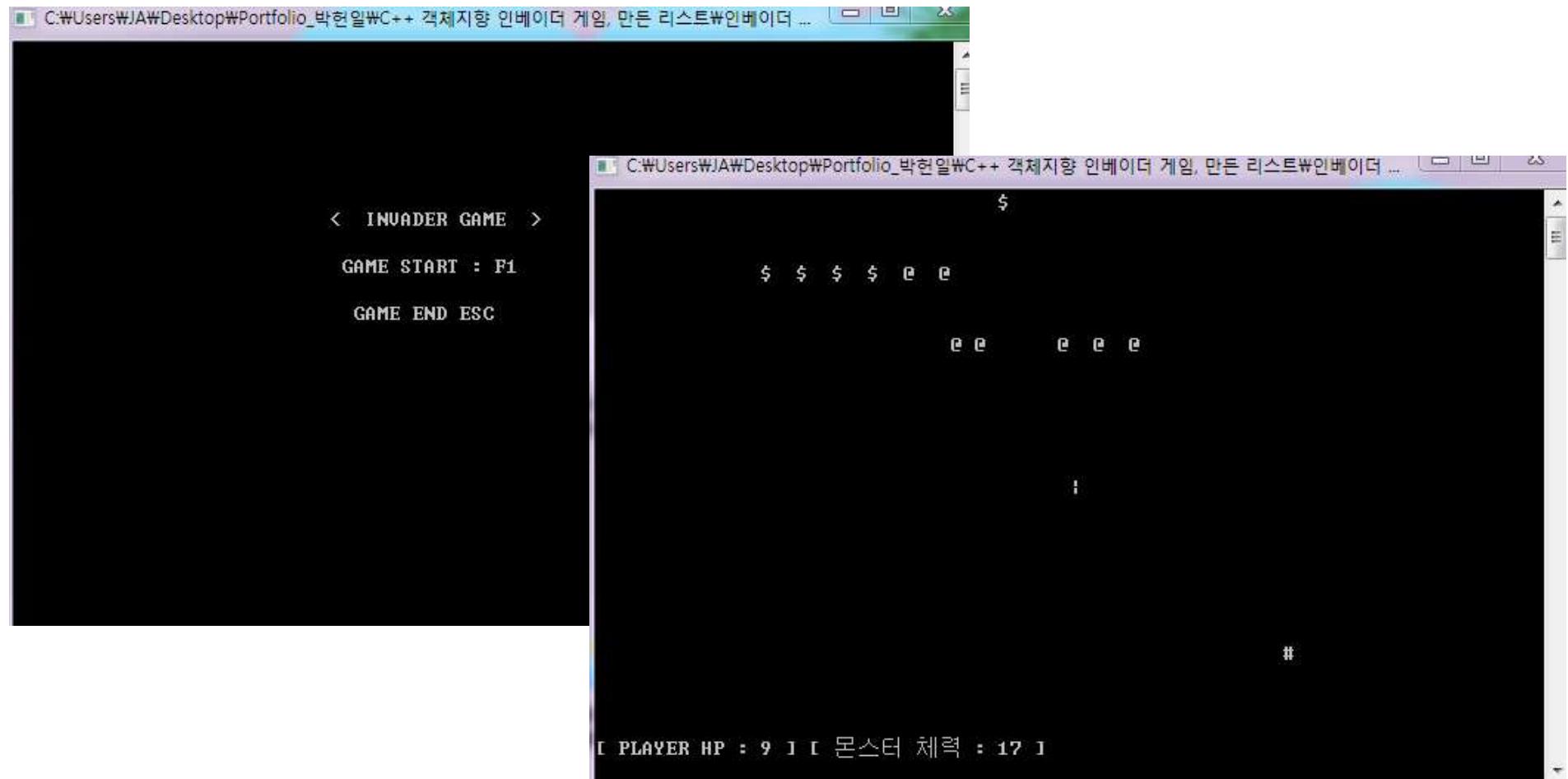
1. Hadoop 및 MongoDB 기반으로 한 웹 로그 처리 시스템에서 처리 중심이 아닌  
로그 분석 중심(시각화)으로 발전  
컴퓨터 한대로 연구를 하였는데 컴퓨터나 서버 여러 대로 병렬 분산 처리
2. MongoDB 뿐만 아니라 Hadoop과 다른 SQL-on-Hadoop과 NoSQL로 연동해서  
시스템 설계 및 구현, 성능 분석 연구
3. 웹 로그 처리/분석 뿐만 아니라 다양한 분야 로그 처리 분석으로 연구  
(시스템, 네트워크, 게임 로그 등)

## 4. 게임 프로젝트(프로그램 아카데미)

- 프로폴리오 작업 내용
- 코딩 스타일

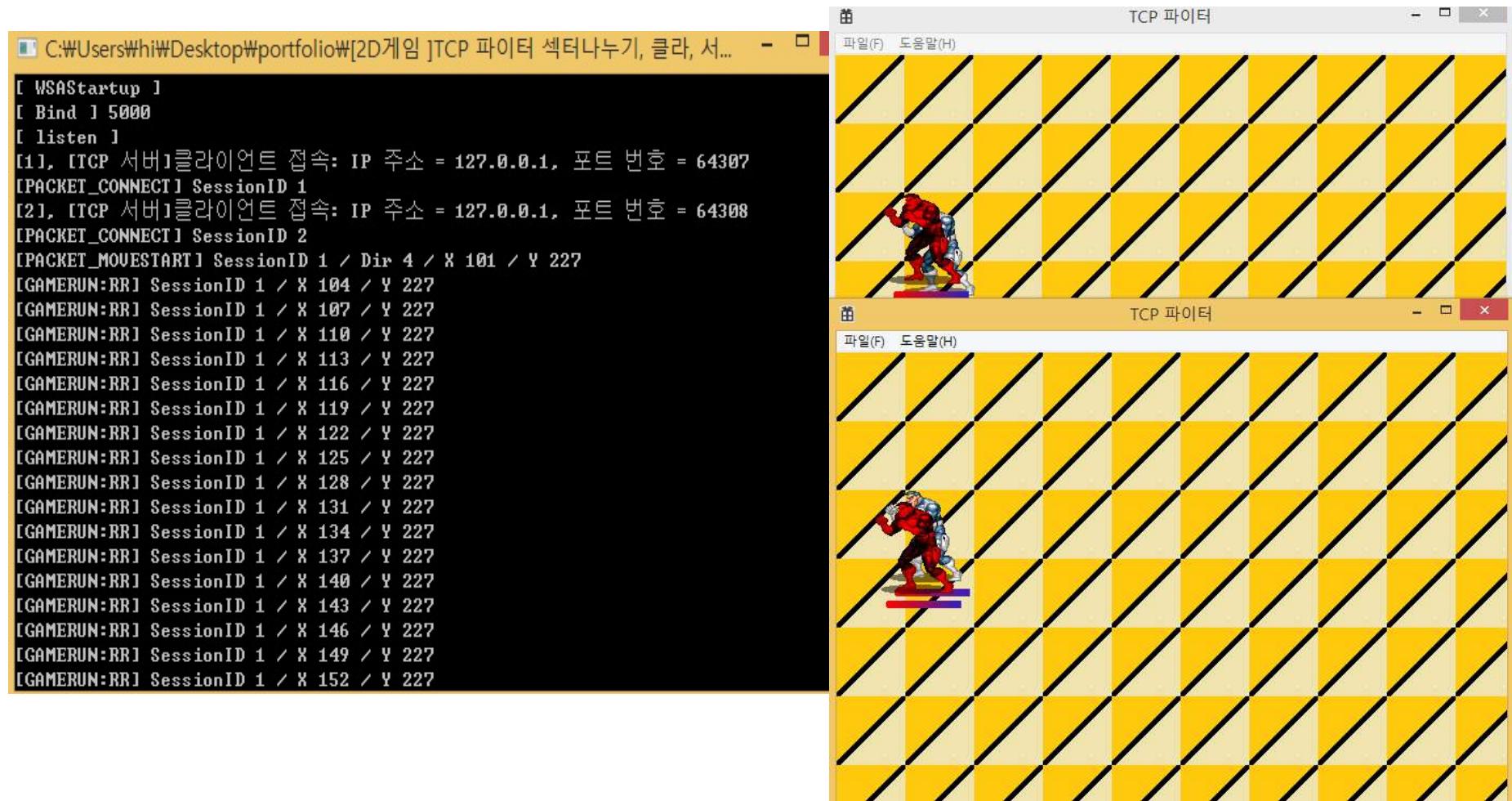
# 포트폴리오 작업 내용

## C++ 객체 지향을 이용한 인베이더 게임 만들기



# 포트폴리오 작업 내용

## 2D게임 - TCP 파이터 서버와 클라이언트



# 포트폴리오 작업 내용

## 서버 모니터링 뷰어차트

### 서버 모니터링 뷰어차트

#### 1. 윈도우 API 이용

#### 2. 큐 이용

#### 3. 서버 모니터링을 하기 위한 그래프 작업

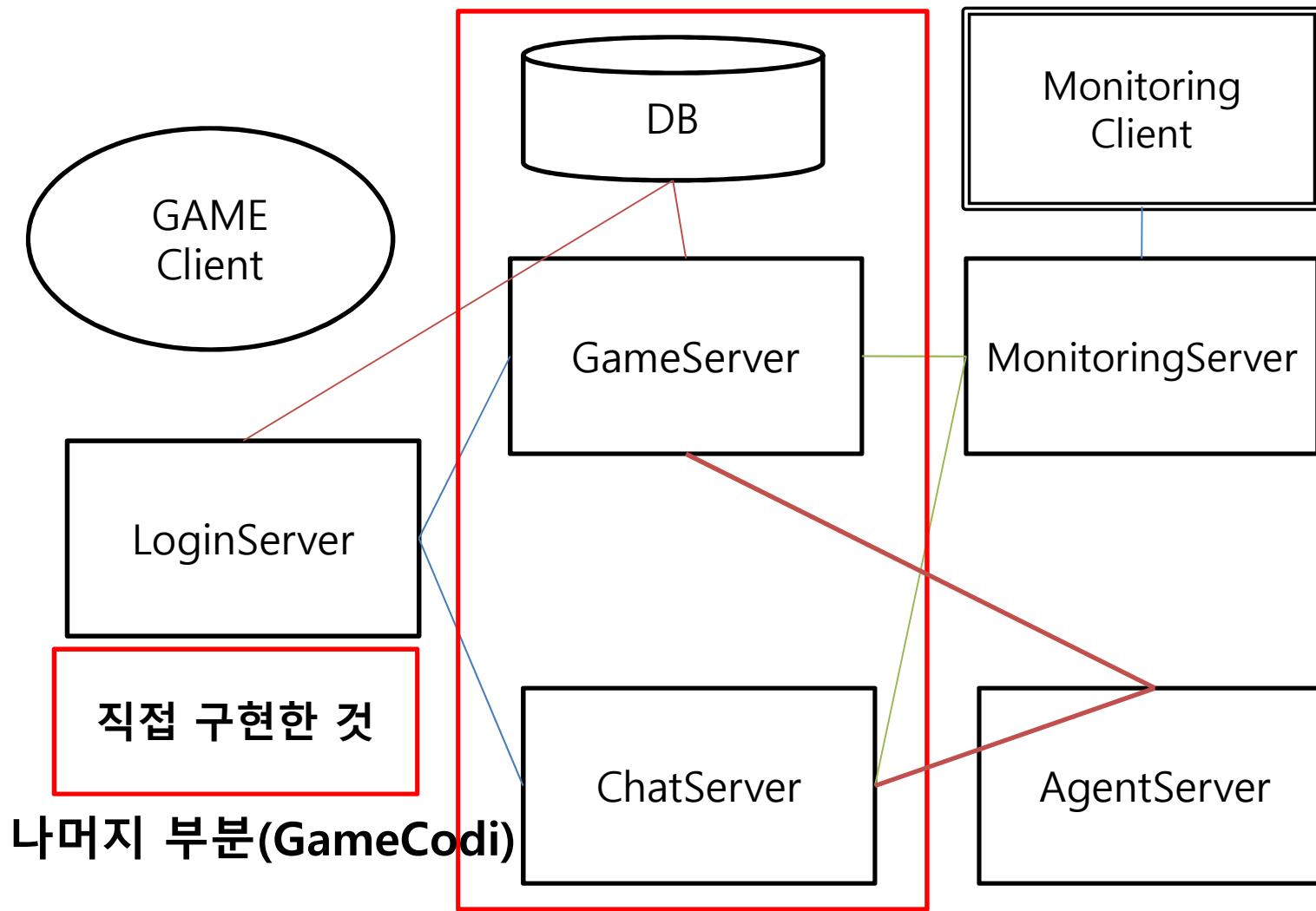
#### 4. 일정 수준이 넘으면 소리나게 함

```
void gs_Graph::RectPaint(HWND hWnd){  
    HPEN MyPen, OldPen;  
    FillRect(hMemDC, &crt, (HBRUSH)CreateSolidBrush(m_color));  
  
    MyPen = CreatePen(PS_SOLID, 1, RGB(255, 0, 0));  
    OldPen = (HPEN)SelectObject(hMemDC, MyPen);  
    Rectangle(hMemDC, 100, 162, 200, QUE.TailGet());  
    SelectObject(hMemDC, OldPen);  
    DeleteObject(MyPen);  
  
    InvalidateRect(hWnd, NULL, FALSE);  
}
```

```
void aram(HWND hWnd, WPARAM wParam)  
{  
    //-----  
    //알림함수 - 경보음과 붉은색 등 각종 색깔 메시지 보냄  
    //-----  
    if (p1->GetintData() >= 99)  
    {  
        SendMessage(hWnd, WM_ALERT1, wParam, 0);  
        PlaySound(L".\WavFiles\test.wav", NULL, SND_ASYNC);  
    }  
}
```

# 포트폴리오 작업 내용

유니티 액션 게임 게임서버 , 채팅서버 IOCP 이용,DB



# 포트폴리오 작업 내용

유니티 액션 게임 게임서버 , 채팅서버 IOCP 이용,DB

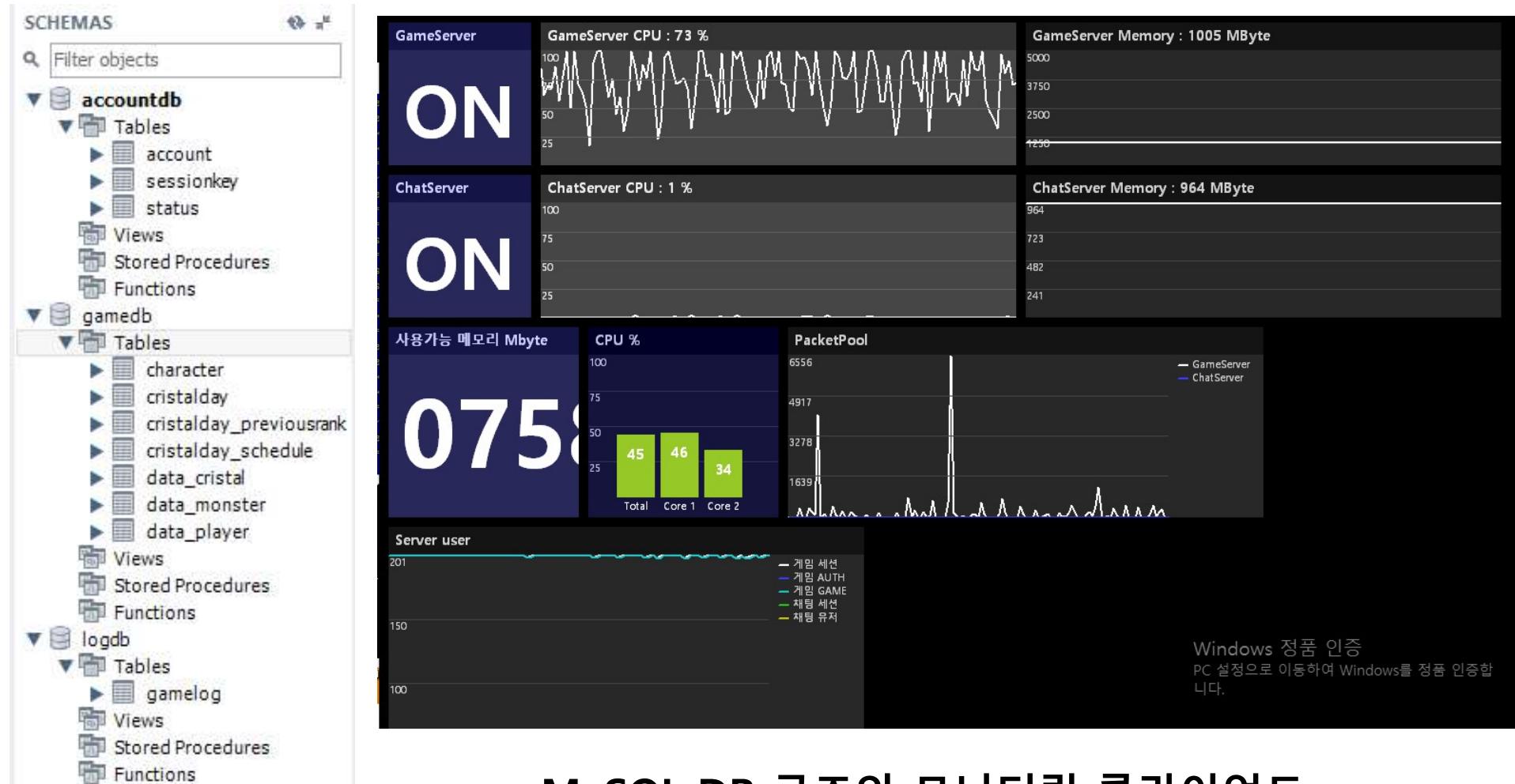


[게임 동영상 보기 클릭](#)

유니티 액션 게임 클라이언트

# 포트폴리오 작업 내용

유니티 액션 게임 게임서버 , 채팅서버 IOCP 이용,DB



MySQL DB 구조와 모니터링 클라이언트

# 포트폴리오 작업 내용

유니티 액션 게임 게임서버 , 채팅서버 ,DB

The image shows two separate terminal windows side-by-side. Both windows have a yellow header bar with the text "ChatServer.exe" or "GameServer.exe". The left window displays the log for the ChatServer, and the right window displays the log for the GameServer. Both logs show various performance metrics such as TPS (Transactions Per Second) for different threads and session counts.

**ChatServer.exe Log:**

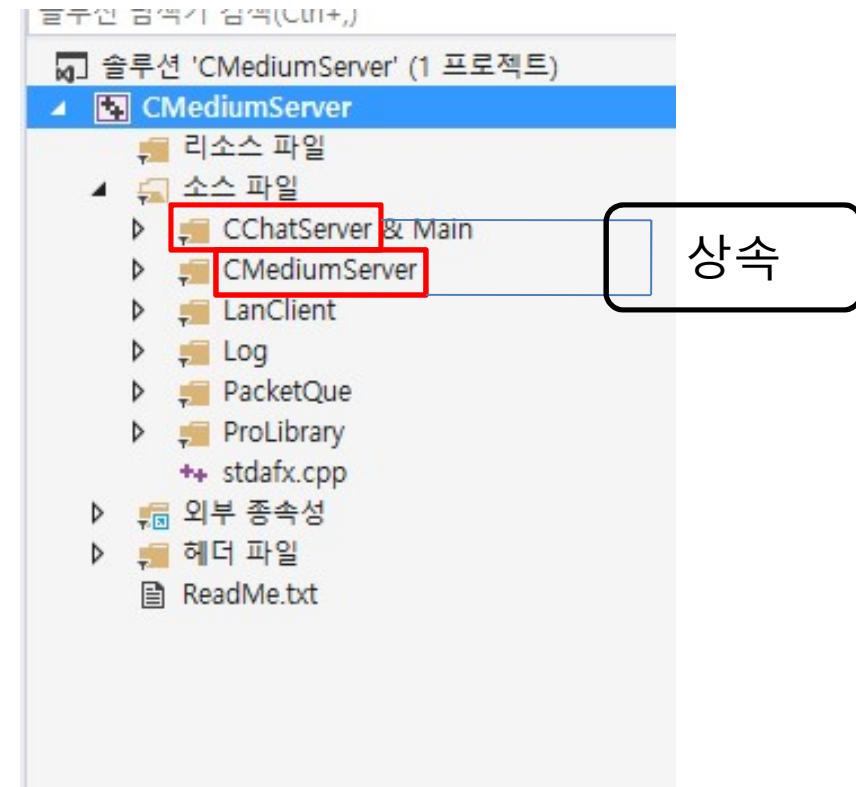
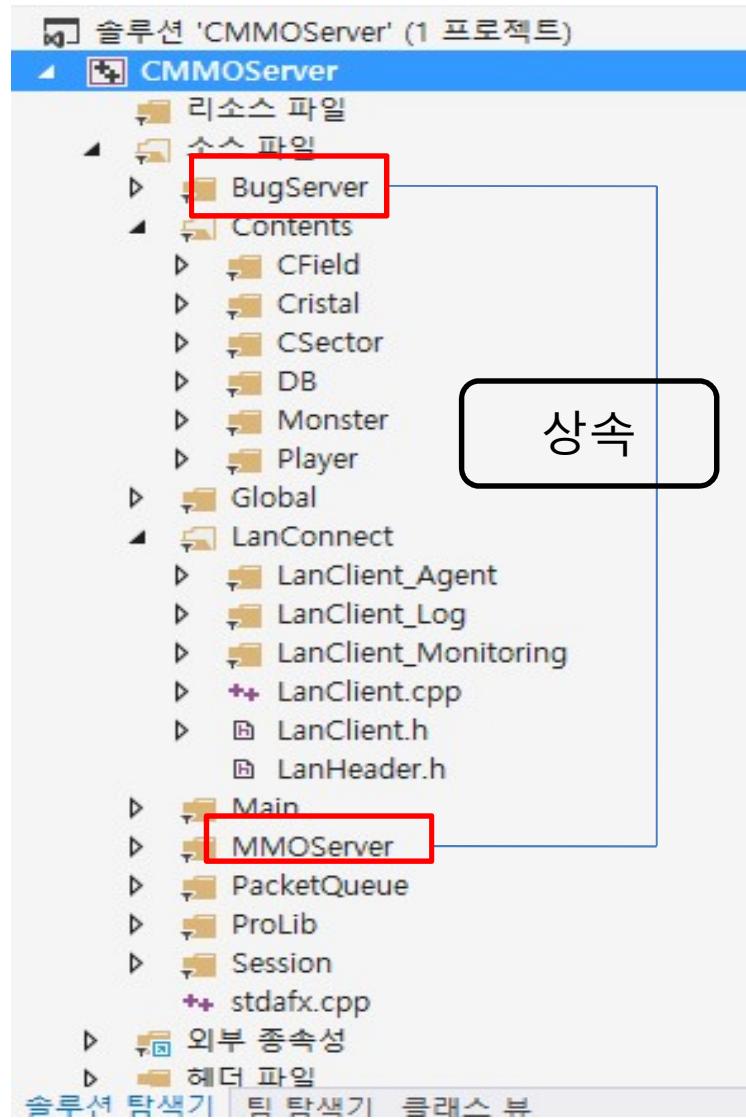
```
ChatServer Monitoring
Monitor_SessionAllMode : 1
```

**GameServer.exe Log:**

```
Accept SOCKET : 0
Accept TPS : 0
PacketProc TPS : 60099
PacketSend TPS : 15626
=====
AuthThread TPS : 539
GameThread TPS : 304
=====
=====
Session : 201
=====
Session_AuthMode : 0
Session_GameMode : 201
=====
PacketPool : 512 use
=====
Accept SOCKET : 0
Accept TPS : 0
PacketProc TPS : 63315
PacketSend TPS : 25865
=====
AuthThread TPS : 551
GameThread TPS : 321
=====
```

채팅 서버와 게임 서버

# 코딩 스타일



# 코딩 스타일

```
bool CBugServer::DB_writer_Update(void)
{
    st_DBWRITER_MSG *pMessage;
    DWORD dwResult = 0;
    static ULONGLONG uTick = 0;

    while (1)//!_bDBShutDown)/// || !_DBWriterQueue.isEmpty())
    {
        dwResult = WaitForSingleObject(_hDBWriterEvent, INFINITE);

        if (dwResult != WAIT_OBJECT_0) //setevent가 시그널
            break;
        _LanClient_Login.LogServer_PacketProc_Heartbeat(dfTHREAD_TYPE_DB);

        while (_DBWriterQueue.Get(&pMessage))
        {
            switch (pMessage->Type_DB)
            {
            case dfACCOUNT_DB:
                _pDBAccount_DBwriter->WriteDB(pMessage->Type_Message, (LPVOID)pMessage->Message);
                break;
            case dfGAME_DB:
                _pDBGame_DBwriter->WriteDB(pMessage->Type_Message, (LPVOID)pMessage->Message);
                break;
            case dfLOG_DB:
                _pDBLog_DBwriter->WriteDB(pMessage->Type_Message, (LPVOID)pMessage->Message);

            default:
                break;
            }
            _DBwriterMessagePool.Free(pMessage);
        }
        CloseHandle(_hDBWriterEvent);
    }

    return false;
}
```

DB 스레드 Event 방식으로 구현

# 코딩 스타일

```
PacketSZ * CPlayer::MakePacket_Game_MOVE_Player()
{
    INT64 ClientID = this->_AccountNo;
    float X = this->fPosX;
    float Y = this->fPosY;
    USHORT Rotation = this->iRotation;
    BYTE VKey = this->VKey;
    BYTE HKey = this->HKey;

    PacketSZ *pSendPacket = NULL;

    pSendPacket = (PacketSZ *)PacketMemoryPool.Alloc();

    if (pSendPacket == NULL)
    {
        return NULL;
    }
    *pSendPacket << (WORD)en_PACKET_CS_GAME_RES_MOVE_CHARACTER;

    *pSendPacket << ClientID;
    *pSendPacket << X;
    *pSendPacket << Y;
    *pSendPacket << Rotation;
    *pSendPacket << VKey;
    *pSendPacket << HKey;

    return pSendPacket;
}
```

각각 함수를 패킷화

```
PacketSZ* CMonster::PakcetMake_Monster_DIE()
{
    //-----
    // 몬스터 사망
    //
    // 몬스터가 죽으면 클라에게 이 패킷을 보냄
    //
    // {
    //     WORD      Type
    //     //
    //     INT64    MonsterClientID
    // }
    //

    //-----

    PacketSZ *pPacket = NULL;

    pPacket = (PacketSZ *)PacketMemoryPool.Alloc();

    if (pPacket == NULL)
    {
        return NULL;
    }

    WORD      Type = en_PACKET_CS_GAME_RES_MONSTER_DIE;
    *pPacket << Type;
    *pPacket << this->_ClientID;

    return pPacket;
}
```

# 코딩 스타일

```
int _tmain(int argc, _TCHAR* argv[])
{
    setlocale(LC_ALL, "");

    LanConnect();

    g_Server.Start(L"127.0.0.1", 40002, true);

    wprintf(L"=====#\n");
    wprintf(L"      Server Start#\n");
    wprintf(L"=====#\n");

    while (g_Server.ServiceState())
    {
        if (g_Server._LanClient_Agent._bServerShutdown == true)
        {
            g_Server.Stop();
        }

        ConsoleProcess();
        ConsoleMonitor(1000);
        Sleep(10);

    }

    Sleep(1000);

    wprintf(L"#n#nbye#\n");
    return 0;
}
```

```
class CPlayer :public CNetSession
{
public:

    CPlayer();
    virtual ~CPlayer();

    void PlayerInitial(void);
    void PacketProc_Auth_Heartbeat(PacketSZ* pPacket);
    virtual bool OnAuth_ClientJoin(void);
    virtual bool OnAuth_ClientLeave(void);
    virtual int OnAuth_Packet(PacketSZ *pPacket);
    virtual PacketSZ* OnGame_ClientJoin(void);
    virtual PacketSZ * OnGame_ClientLeave(void);

    virtual PacketSZ * OnGame_OtherClientJoin(void);
    void CPlayer::PacketProc_Auth_Ch(PacketSZ *pPacket);
    int OnGame_Packet(PacketSZ *pPacket);
    virtual bool OnGame_ClientRelease(void);

    virtual void CPlayer::NoChoice_Gamelogin_char(int AccountNo);
    virtual void CPlayer::Choice_Gamelogin_char(int Account);
    int PacketProc_Auth_Login(PacketSZ *pPacket);

    bool PacketProc_Game_Echo(PacketSZ *pPacket);

    int PacketProc_Game_MOVE_Player(PacketSZ *pPacket);
```

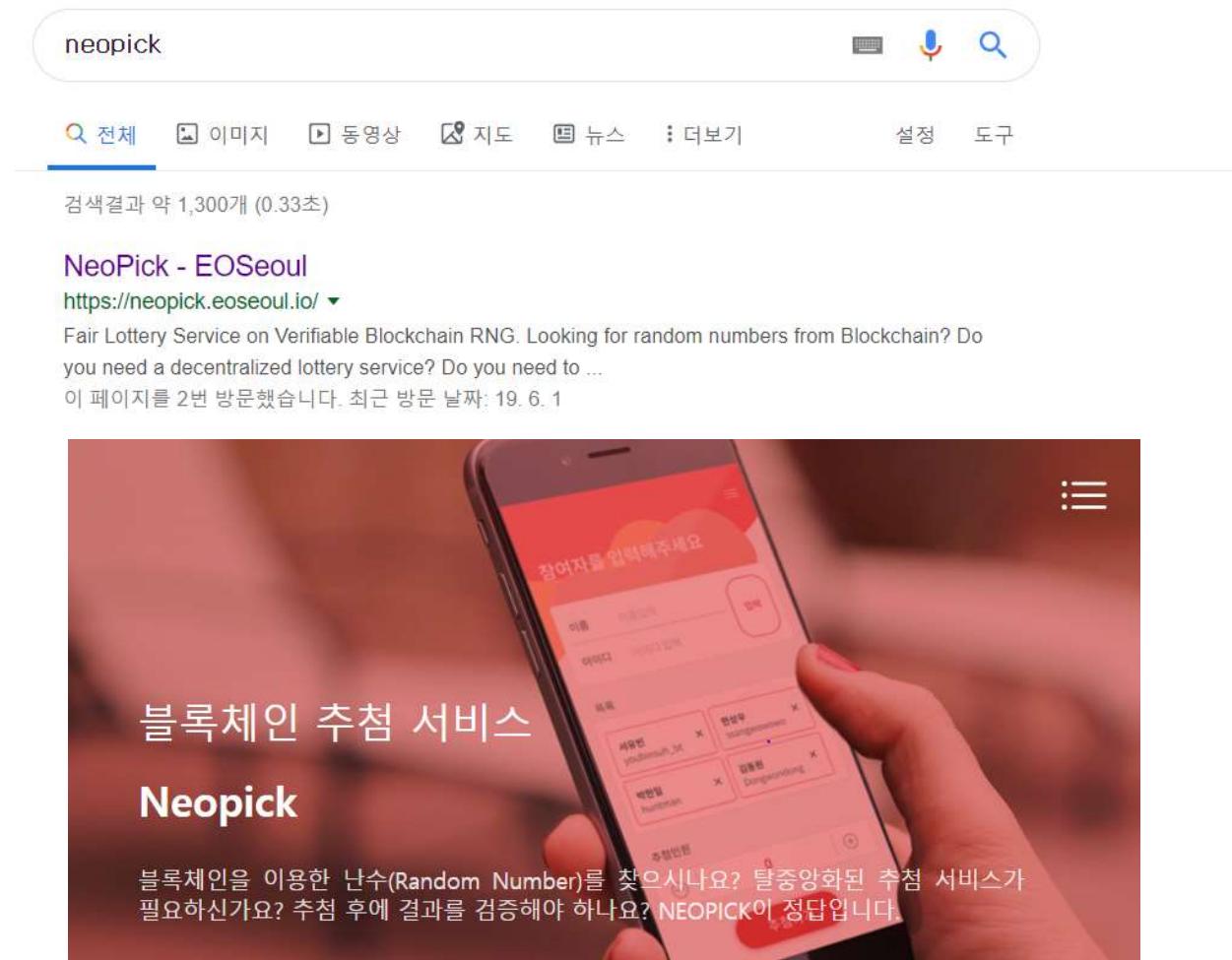
클래스 상속 및 가상함수 이용

## 5. 네오위즈 기술본부 블록체인 팀

- 인턴 직무 작업 내용
  - C++ 기반 서버, 스마트 컨트랙트 개발

# 업무 프로젝트 내용

네오픽(Neopick) : 블록체인 기반 추첨 뽑기 서비스(4명 작업 진행)  
<https://neopick.eoseoul.io/>



neopick

전체 이미지 동영상 지도 뉴스 더보기 설정 도구

검색결과 약 1,300개 (0.33초)

**NeoPick - EOSeoul**  
<https://neopick.eoseoul.io/>

Fair Lottery Service on Verifiable Blockchain RNG. Looking for random numbers from Blockchain? Do you need a decentralized lottery service? Do you need to ...  
이 페이지를 2번 방문했습니다. 최근 방문 날짜: 19. 6. 1

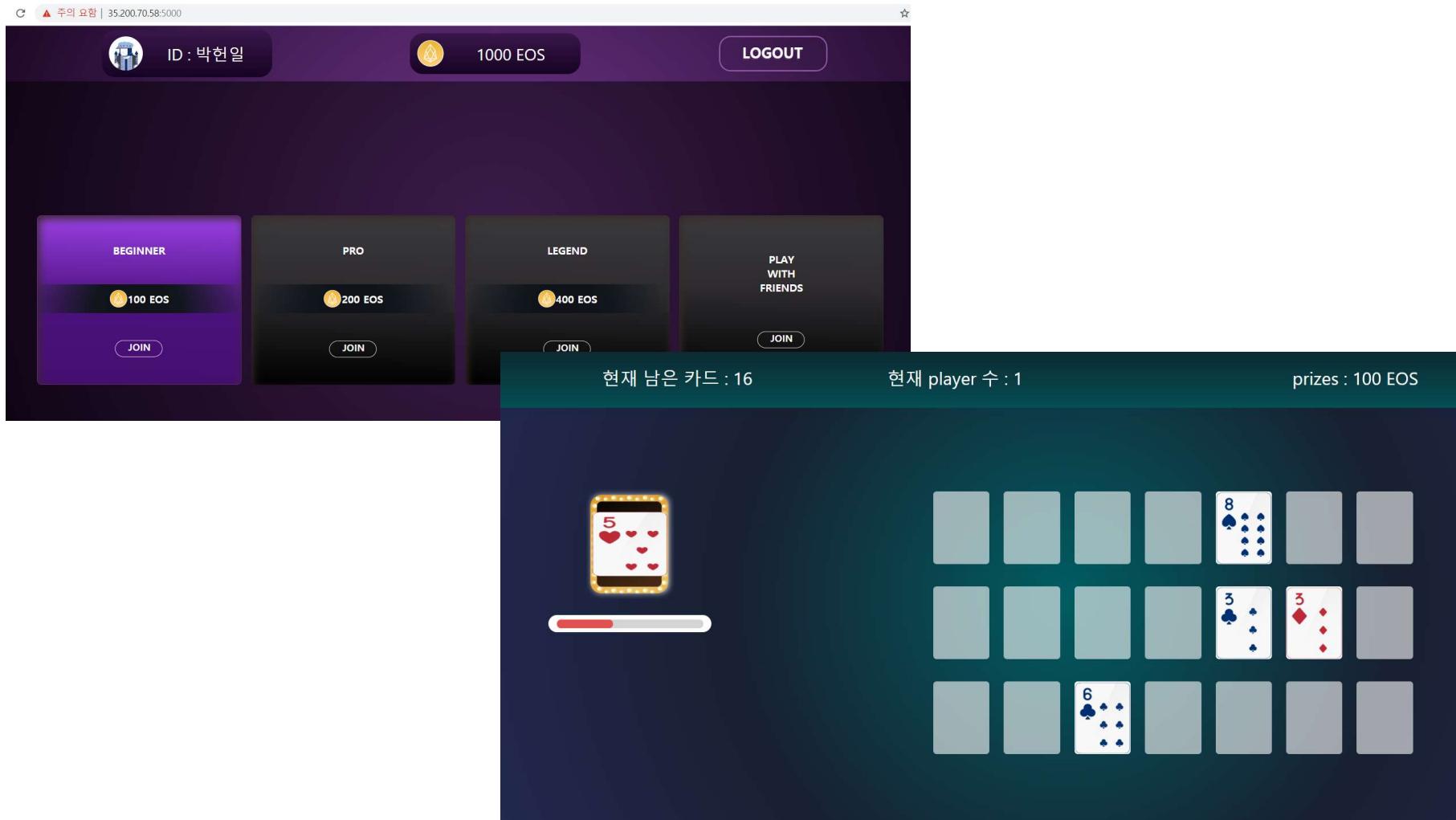
블록체인 추첨 서비스  
**Neopick**

블록체인을 이용한 난수(Random Number)를 찾으시나요? 탈중앙화된 추첨 서비스가 필요하신가요? 추첨 후에 결과를 검증해야 하나요? NEOPICK이 정답입니다.

# 업무 프로젝트 내용

Roper in the card : 블록체인 기반 카드 게임 (4명 작업 진행)

<http://35.200.70.58:5000/>



Q&A  
감사합니다.