

# 캡스톤 설계 최종 발표

- Fair grid -

발표자: 명세인 ([sein@pel.smuc.ac.kr](mailto:sein@pel.smuc.ac.kr))

지도교수: 이종혁([jonghyouk@smu.ac.kr](mailto:jonghyouk@smu.ac.kr))

허진영, 한석찬, 채송현

상명대학교 컴퓨터소프트웨어공학과

# 목 차

---

- 개요
- Fair-grid
- 유사 서비스 비교
- 기대효과
- 거래현황과 경매 진행

# 개요(1/3)

---

- Fair-Grid

- Fair-Grid: 공정한 전력거래 시스템
  - 경매 알고리즘
    - 판매자와 구매자는 최선의 가격 경쟁

- 개발 배경

- 기존의 전력관리 시스템
  - 공급자(발전시설)와 소비자가 명확히 구분되는 단방향 구조
- 프로슈머의 등장과 마이크로그리드
  - 생산자(Producer)와 소비자(Consumer)의 합성어로 전력을 생산하기도 하면서 소비하기도 하는 사람을 의미
    - 프로슈머는 전력이 남을때 팔고 부족할때 구입

# 개요(2/3)

---

- 개발 배경

- 전력거래 환경

- 중앙집중식 구조에서 프로슈머의 자율적인 시장 형성의 어려움
- 탈중앙화 환경 도입 시, 명확한 거래 환경제공자와 부당 거래 방지의 어려움

- 개발 목표

- 중앙관리 기관이 없는 분산네트워크에서 자율적인 거래 환경 제공
  - 블록체인 기술 활용

# 개요(3/3)

---

- 블록체인

- 분산 네트워크환경에서 신뢰성과 영속성을 보장하는 데이터공유 기술

- 1세대: 가상의 암호화폐거래를 위한 분산원장기술(DLT)

- DLT: Distribution Ledger Technology

- 2세대: 신뢰성과 무결성을 보장하는 코드실행

- Solidity: 컨트랙트 코드

# Fair-grid(1/4)

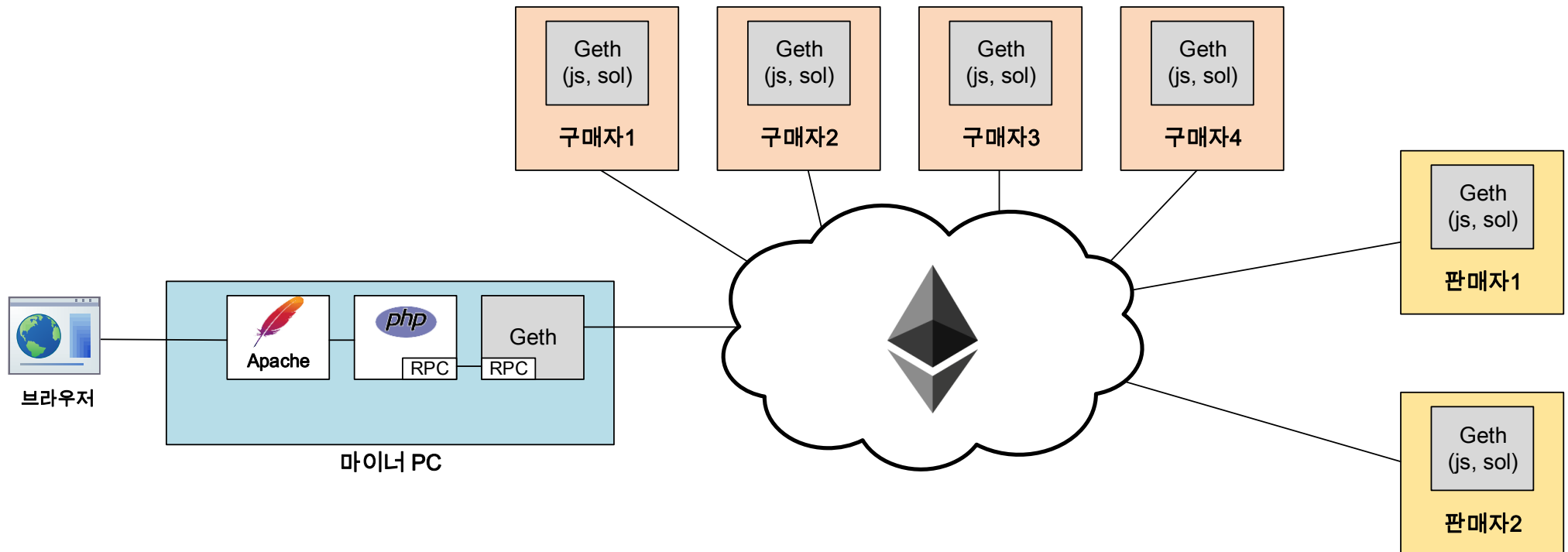
---

- 블록체인기반 마이크로그리드
  - 프로슈머는 스마트컨트랙트 기술을 활용하여 경매를 통해 전력을 거래
- 블록체인
  - 풀노드
    - 마이크로그리드의 경매 거래 환경 제공
    - 웹 기반의 거래정보 제공
  - 라이트노드
    - 프로슈머로써 스마트컨트랙트기반 자율적인 경매 진행

# Fair-grid(2/4)

- 네트워크 구성도

- 판매자와 구매자는 프로슈머의 스마트미터기
  - 스마트미터기
    - 정보통신 기술이 융합된 전력관리 임베디드 장비



# Fair-grid(3/4)

---

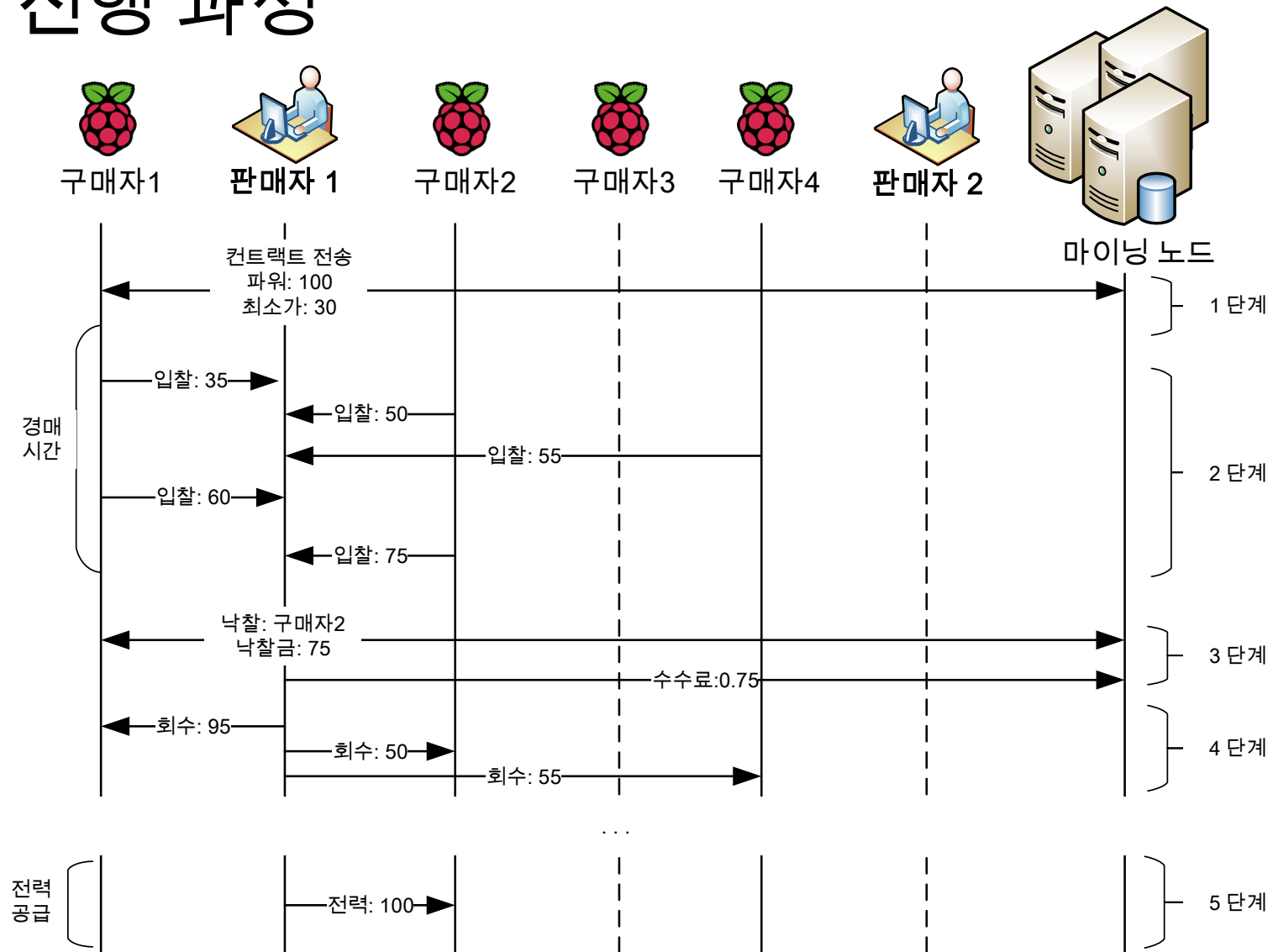
- 경매 진행 과정

1. 판매자의 경매 게시 및 구매자의 탐색
2. 구매자의 입찰 시도
3. 판매자의 경매 종료(낙찰) 및 수수료 지급
4. 구매자의 입찰 시도 금액 회수
5. 전력 공급



# Fair-grid(4/4)

## • 경매 진행 과정



# 유사 서비스 비교

## • 마이크로그리드 전력거래 기술

|                                | 탈중앙화            | 전력거래기능              | P2P 매칭 | 대상        | 전력 공급     | 매칭 간격         | 수수료    |
|--------------------------------|-----------------|---------------------|--------|-----------|-----------|---------------|--------|
| 피클로<br>(Piclo)                 | X               | O<br>(요구가격 제시)      | X      | 기업        | 전력회사      | 30분           | 지역별    |
| 반데브론<br>(Vandebron)            | X               | O<br>(공급자 제시)       | X      | 개인        | 개인        | 2분            | \$12/월 |
| 엘로하<br>(Yeloha)                | X               | △                   | O      | 개인        | 개인        | -             | 이용권    |
| 소넨커뮤니티<br>(sonnenComunity)     | X               | O<br>(25센트/kWh, 독일) | O      | 개인,<br>기업 | 개인,<br>기업 | 실시간<br>(전력 풀) | -      |
| 프래지던트 거리<br>(President Street) | O<br>(Ethereum) | △                   | △      | 개인        | 개인        | 실시간<br>(블록체인) | -      |
| Fair-grid                      | O<br>(Ethereum) | O<br>(경매 기반)        | O      | 개인,<br>기업 | 개인,<br>기업 | 실시간<br>(블록체인) | 유동     |

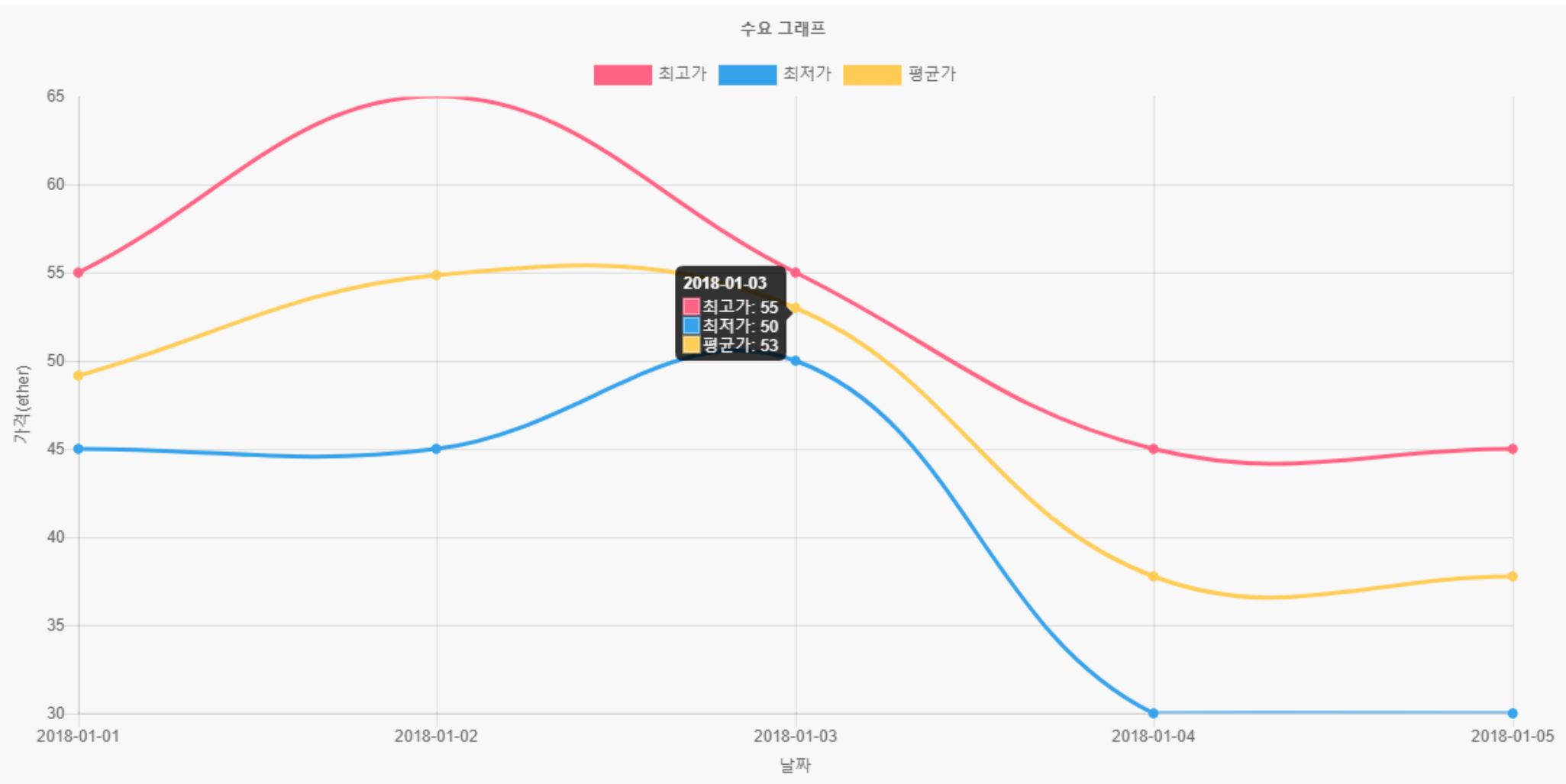
# 기대효과

---

- 분산 네트워크환경의 전력 경매 시장
  - 프로슈머간 자율적인 경매 참여 및 거래 보장 가능
- 경매 알고리즘 개선
  - 안정적인 전력 판매
  - 저렴한 전력 구매

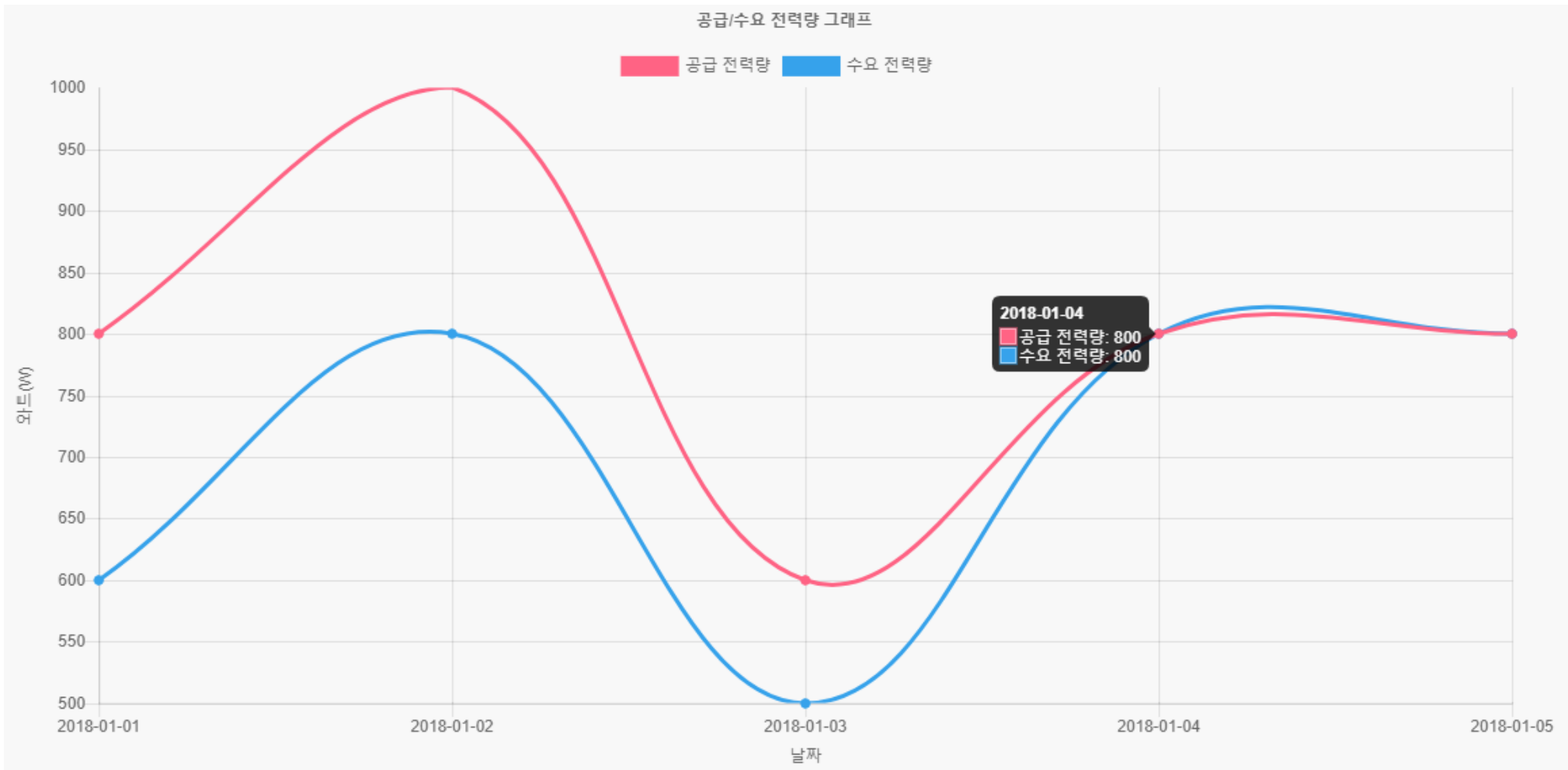
# 거래 현황과 경매 진행(1/4)

- 수요 시세 그래프



# 거래 현황과 경매 진행(2/4)

- 공급, 수요 전력량



# 거래 현황과 경매 진행(3/4)

## • 사용자 거래정보

### 판매완료

| 경매 시작               | 경매 종료               | 전력량 | 공급 시작               | 공급 종료               | 낙찰자  |
|---------------------|---------------------|-----|---------------------|---------------------|--|
| 2017-11-04 16:41:th | 2017-11-04 16:43:37 | 100 | 2018-01-01 15:00:00 | 2018-01-01 16:00:00 | 0xdae1c89f1e0d1c060a8a0ba94bbb7857a314f9a4 |
| 2017-11-04 16:43:th | 2017-11-04 16:45:07 | 100 | 2018-01-01 15:00:00 | 2018-01-01 16:00:00 | 낙찰자없음                                      |
| 2017-11-04 16:44:th | 2017-11-04 16:46:47 | 100 | 2018-01-01 15:00:00 | 2018-01-01 16:00:00 | 0xdae1c89f1e0d1c060a8a0ba94bbb7857a314f9a4 |
| 2017-11-04 16:46:th | 2017-11-04 16:48:27 | 100 | 2018-01-01 15:00:00 | 2018-01-01 16:00:00 | 0x7451b07fc1f2316d4c3be8b993742aa7bf755a35 |
| 2017-11-04 16:48:th | 2017-11-04 16:50:06 | 100 | 2018-01-01 15:00:00 | 2018-01-01 16:00:00 | 0xdae1c89f1e0d1c060a8a0ba94bbb7857a314f9a4 |

### 구매완료

| 경매 시작               | 경매 종료               | 전력량 | 공급 시작               | 공급 종료               | 낙찰자  |
|---------------------|---------------------|-----|---------------------|---------------------|--|
| 2017-11-04 16:27:th | 2017-11-04 16:29:36 | 100 | 2017-12-31 15:00:00 | 2017-12-31 16:00:00 | 0xac99606225b4668ce909cd0df110076ad7771f2c |
| 2017-11-04 16:27:th | 2017-11-04 16:29:36 | 100 | 2017-12-31 15:00:00 | 2017-12-31 16:00:00 | 0xbd9922f05984404c3a6f5d78249abce63caa5b6c |
| 2017-11-04 16:56:th | 2017-11-04 16:58:04 | 100 | 2018-01-02 15:00:00 | 2018-01-02 16:00:00 | 0xbd9922f05984404c3a6f5d78249abce63caa5b6c |
| 2017-11-04 16:56:th | 2017-11-04 16:58:04 | 100 | 2018-01-02 15:00:00 | 2018-01-02 16:00:00 | 0xac99606225b4668ce909cd0df110076ad7771f2c |
| 2017-11-04 16:57:th | 2017-11-04 16:59:17 | 100 | 2018-01-02 15:00:00 | 2018-01-02 16:00:00 | 0xbd9922f05984404c3a6f5d78249abce63caa5b6c |

# 거래 현황과 경매 진행(4/4)

---

- 경매 진행

- 테스트 환경

- 블록체인 플랫폼: 이더리움(Ethereum)
- 스마트컨트랙트: 솔리디티(Solidity)
- 경매 알고리즘 동작: 자바스크립트(Javascript)
- 마이너 OS: Ubuntu
- 프로슈머 OS: Raspbian

- 테스트 영상

- 전력판매 최저가격: 30
- 구매자 1: 최대 45
- 구매자 2: 최대 55

---

감사합니다!



# 백업1: 프로젝트 진행 과정

## • 프로젝트 진행 계획

| 9월 3일 ~ 11월 4일           | 주 단위 |   |   |   |   |   |   |   |   |    |
|--------------------------|------|---|---|---|---|---|---|---|---|----|
|                          | 1    | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 마이크로그리드 환경 정의            |      |   |   |   |   |   |   |   |   |    |
| 엔티티 역할 정의                |      |   |   |   |   |   |   |   |   |    |
| 경매 설계<br>(스마트컨트랙트, 알고리즘) |      |   |   |   |   |   |   |   |   |    |
| 블록체인 네트워크 구현             |      |   |   |   |   |   |   |   |   |    |
| 경매 알고리즘 구현               |      |   |   |   |   |   |   |   |   |    |
| 웹 인터페이스 구현               |      |   |   |   |   |   |   |   |   |    |
| 웹 연동 구현                  |      |   |   |   |   |   |   |   |   |    |
| 테스트                      |      |   |   |   |   |   |   |   |   |    |

# 백업1: 프로젝트 진행 과정

## • 프로젝트 참여도

|                          | 명세인          | 한석찬 | 허진영 | 채송현 |
|--------------------------|--------------|-----|-----|-----|
| 블록체인 분석                  | 2017년 상반기 진행 |     |     |     |
| 마이크로그리드 환경 정의            |              | ○   | ○   | ○   |
| 엔티티 역할 정의                |              | ○   | ○   | ○   |
| 경매 설계<br>(스마트컨트랙트, 알고리즘) | ○            |     |     | ○   |
| 블록체인 네트워크 구현             | ○            |     |     |     |
| 경매 알고리즘 구현               | ○            |     |     |     |
| 웹 인터페이스 구현               |              | ○   | ○   | ○   |
| 웹 연동 구현                  | ○            | ○   | ○   |     |

# 백업2: 경매 알고리즘 주요 함수

## • Auction.sol: 스마트컨트랙트

| 타입       | 함수 명       | 파라미터  | 기능                                    |
|----------|------------|---|---------------------------------------|
| function | Auction    | uint32 type<br>address beneficiary<br>uint32 auctionStart<br>uint32 biddingTime<br>uint min<br>uint max<br>uint32 power<br>uint32 powerStart<br>uint32 powerEnd | 컨트랙트를 전송하며 최초 필드 세팅(생성자)              |
| function | Bid        | (payable)   | 함수를 호출하며 화폐전송, 변수필드를 확인하여 임시 낙찰 여부 판단 |
| function | auctionEnd | -   | biddingTime을 확인하여 경매 종료               |
| function | withdraw   | -   | 경매 종료 후 낙찰되지 못한 금액을 회수                |
| function | Auction    | uint32 type<br>address beneficiary<br>uint32 auctionStart<br>uint32 biddingTime<br>uint min<br>uint max<br>uint32 power<br>uint32 powerStart<br>uint32 powerEnd | 컨트랙트를 전송하며 최초 필드 세팅(생성자)              |
| function | Bid        | (payable)   | 함수를 호출하며 화폐전송, 변수필드를 확인하여 임시 낙찰 여부 판단 |

# 백업2: 경매 알고리즘 주요 함수

## • Seller.js: 판매자

| 타입       | 함수 명    | 파라미터  | 기능  |
|----------|---------|---|---|
| function | supply  | -   | 판매자가 원하는 판매 조건을 입력하는 함수   |
| function | sell    | Number sec<br>Number min<br>Number power<br>Number powerStart<br>Number powerEnd<br>Number interval<br>Number num<br>String miner | 판매자가 입력하는 조건(파라미터)에 따라 컨트랙트 코드 생성해 경매를 시작 및 경매 종료시간에 맞춰 종료하는 함수 |
| function | sendcon | Number sec<br>Number min<br>Number power<br>Number powerStart<br>Number powerEnd  | 조건에 맞는 컨트랙트를 생성해 트랜잭션 작성(경매 시작)                                 |
| function | conEnd  | Object con  | 경매 종료시간을 확인해 만족 시 종료하는 함수                                       |
| function | getcon  | Number addr<br>Array abi  | 입력한 컨트랙트의 주소에 해당하는 컨트랙트 객체를 반환하는 함수                             |
| function | coninfo | Object con  | 종료된 경매에 대한 결과 출력  |

# 백업2: 경매 알고리즘 주요 함수

## • Buyer.js: 구매자

| 타입       | 함수 명             | 파라미터   | 기능  |
|----------|------------------|--|---|
| function | buy              | -  | purchase함수를 호출하여 경매 시작  |
| function | purchase         | Number add<br>Number power<br>Number rate<br>Number powerStart<br>Number powerEnd  | 최신 블록 중 구매자 조건에 맞는 컨트랙트를 선별해 비딩 및 경매 종료 후 낙찰되지 못한 금액을 회수하는 함수         |
| function | get_candidate    | Array arr_con<br>Number block_start<br>Number max<br>Number power<br>Number rate<br>Number powerStart<br>Number powerEnd | 컨트랙트 여부와 설정한 컨트랙트 조건에 맞는 컨트랙트를 선별해 arr_con 배열에 저장해주는 함수               |
| function | get_lowest_con   | Array candidate  | 후보 경매 중 비딩 금액이 제일 낮은 컨트랜트를 반환하는 함수                                    |
| function | select_candidate | Object con<br>Number max<br>Number Start<br>Number End   | 설정한 비딩 조건(경매 종료 여부, 구매자의 잔금, 설정한 최대 값, 전력 공급 시작 및 종료시간)에 맞는지 확인해주는 함수 |

# 백업2: 경매 알고리즘 주요 함수

## • Buyer.js: 구매자

| 타입       | 함수 명        | 파라미터                     | 기능                                     |
|----------|-------------|--------------------------|--|
| function | ended_check | Object con               | ended와 경매 시간을 이용해 컨트랙트의 종료 여부 확인해주는 함수 |
| function | assort_con  | Array arr_tx<br>Number i | 블록에 존재하는 트랜잭션들 중 컨트랙트인지 아닌지 분류하는 함수    |
| function | print_con   | Object con<br>Number i   | 입력 받은 컨트랙트 객체의 정보를 출력해주는 함수            |
| function | bal         | Number a                 | 지갑 주소를 통해 잔액을 반환하는 함수                  |
| function | getcon      | Number addr<br>Array abi | 컨트랙트의 정보를 저장해 해당 컨트랙트의 주소를 반환          |
| function | conBid      | Number b<br>Object con   | 해당 컨트랙트(con)에 b원(ether)을 비딩하는 함수       |

# 백업3: 웹페이지

