

2019 년도 2 학기 공학경진대회 보고서

블록체인을 활용한 멘토- 멘티 매칭 디앱(dApp)

MENTORS

팀장 노윤지

팀원 박주영, 정지원, 정효진

목 차

I. 서론

1. 동기 및 목표
2. 작품 설명

II. 본론

1. 블록체인
 - 가. 개념
 - 나. 구조
 - 다. 특징 및 장·단점
 - 라. 핵심 기술
 - 마. 플랫폼 종류(Public Block Chain vs Private Block Chain)
2. 디앱(dApp)
 - 가. 개념
 - 나. App vs dApp
 - 다. 전망
3. 작품의 작동 원리

III. 결론

IV. 참고 문서

I. 서론

1. 동기 및 목표

가. 블록체인

2017년 가상화폐(비트코인)가 이슈가 된 이후, 블록체인에 대한 사람들의 관심이 급증했다가 폭락하는 상황이 발생하였다. 그럼에도 불구하고, 2017년에 이어서 2019년인 지금까지도 블록체인 즉, 분산 장부 시스템은 주목 받는 신기술의 대목에 올라가 있다. 이처럼 블록체인이 계속 주목 받는 이유가 궁금해졌고, 기술에 대해 더 자세히 공부해보고 싶어졌다.

나. 주제 선정

아직 블록체인 기술은 일반인들이 사용하기에는 어려운 부분들이 많이 있는데, 이번 프로젝트에서는 블록체인의 장점을 최대한 살리면서도, 기업적 측면보단 일반인들이 어떻게 하면 블록체인 기술을 관심 있게 접해 볼 수 있을까에 초점을 맞추었다.

그래서 생각하게 된 것이 학생들이나 지식 습득을 필요로 하는 사람들이 지식을 공유할 수 있는 웹사이트이다. 다른 지식공유 커뮤니티와 다르게 서로 지식습득을 위해 소액을 지불하는 형식으로, 블록체인 가상화폐 사용을 유발하여 블록체인 기술에 대한 관심을 높이려고 한다.

2. 작품 설명

One day Mentor Dapp, 즉 소액을 지불하고 하루 동안 필요한 멘토를 구하는 블록체인 활용 웹사이트이다. 간략히 설명하자면, 누구나 멘토와 멘티가 될 수 있는 기회를 가질 수 있고, 24시간이라는 한정된 시간 동안 댓글창이 활성화 되어 둘 사이의 연결이 가능해진다. 또한 멘토와 멘티의 연결 가운데 멘티의 이더(블록체인 화폐) 소액 지불이 필요하고, 이러한 거래기록을 블록체인 장부 사용함으로써 거래에 대한 신뢰성을 높였다. 더 자세한 설명은 아래에서 계속 하겠다.

II. 본론

1. 블록체인

가. 블록체인 개념

‘블록체인’이란 데이터 분산 처리 기술이다. 이는 네트워크에 참여하는 모든 사용자가 모든 거래 내역 등의 데이터를 분산 저장하는 기술을 지칭하는 말이다. 블록에 담긴 장부와 시간 순으로 밀봉 후 체인으로 연결된 블록이기 때문에 블록체인이라는 이름이 붙었다. 검증된 공공 장부를 만들어 개인에게 분산, 공유하며 관리하는 시스템이라고 할 수 있다.

나. 블록체인 구조

1) 블록체인

가) 블록해시

블록의 헤더 정보를 모두 더하여 합을 구한 후 SHA256¹으로 변환한 결과 값이다. 쉽게 블록의 이름 정보라고 생각 할 수 있다.

나) 헤더

(1) 버전(version) : 해당 블록의 버전이다. 블록 헤더를 만든 프로그램의 버전정보를 의미한다.

(2) 이전 블록 해시(Previous Block Hash) : 이전 블록의 주소 값을 가리키는 요소다. 이 때문에 각 블록이 서로 연결되어 있는 구조가 된다.

(3) 머클루트(Merkle Root) : 바디 부분에 저장된 트랜잭션들의 해시 트리다. 블록이 유효한지 무결성을 검증하기 위한 요소다.

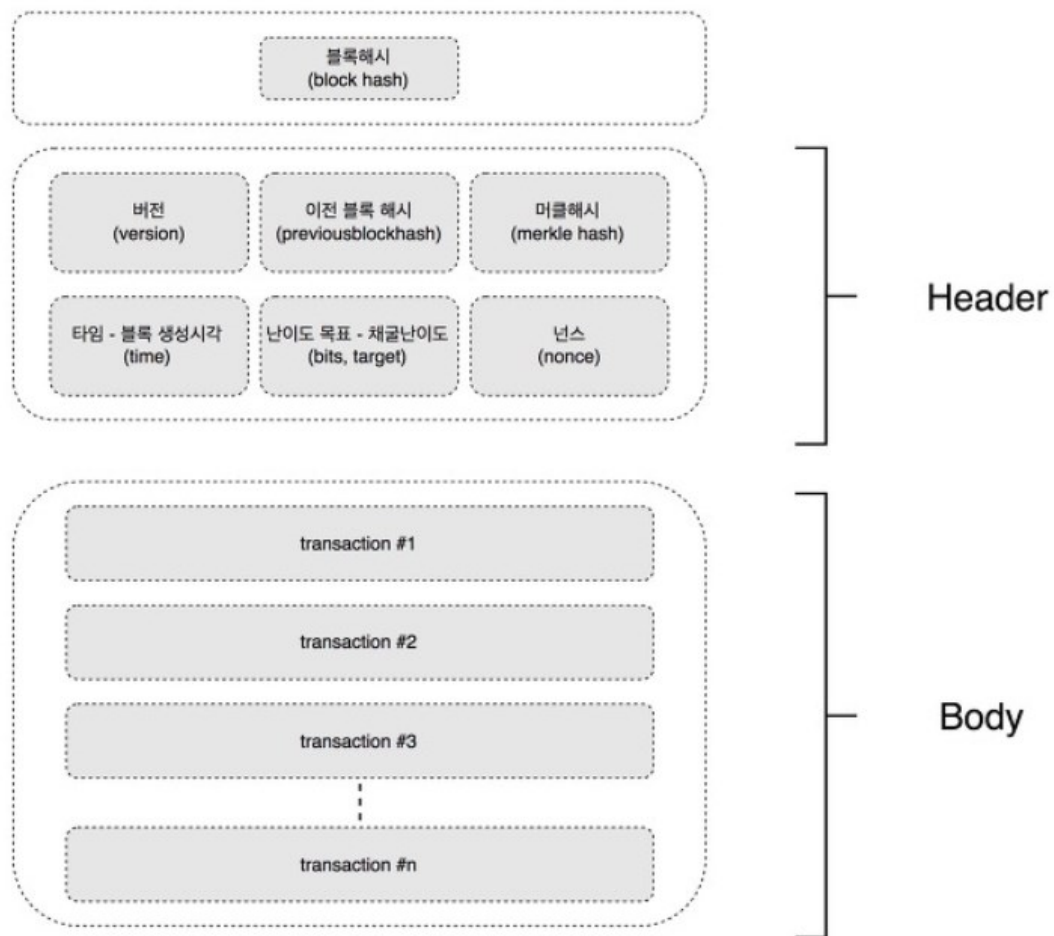
(4) 타임(Time) : 블록의 대략적인 생성 시간을 의미한다.

(5) Bits : 난이도 해시 목표 값을 의미하는 지표다.

(6) 논스(Nonce) : 블록을 만드는 과정에서 해시 값을 구할 때 필요한 재료 역할이다.

다) 바디 : 여러 트랜잭션으로 이루어져 있다.

¹ SHA256 : 미국 표준 기술 연구소에 의해 공표된 표준 해시 알고리즘 중 가장 많이 채택되어 사용되는 알고리즘



<그림 1> 블록체인

(1) 노드(Node)

블록체인은 중앙 서버에 거래 기록을 보관하지 않고 거래에 참여하는 개개인의 서버들이 모여 네트워크를 유지 및 관리하는데 이 개개인의 서버 즉 참여자를 노드라고 한다. 이는 노드에 블록체인이 저장된다는 것을 의미한다.

① 풀 노드(Full Node)

모든 블록체인 원장을 가진 노드이며 블록체인 데이터를 동기화 하기 위해서 메모리를 사용해야 한다. 모든 거래를 독립적으로 검증하고 실시간으로 데이터를 업데이트 할 수 있다. 제네시스 블록²부터 시작해서 현재 블록까지 모든 블록체인 정보를 수집, 저장한다. 자신의 머신에 모든 블록체인의 내용을 가지고 있기 때문에 다른 노드의 도움없이 스스로 거래 검증이 가능하다. 하지

² 제네시스 블록(Genesis Block) : 어떤 블록체인 네트워크에서 생성된 첫 번째 블록.

만 용량이 너무 크다는 단점이 있다.

② 라이트 노드(Light Node)

풀노드의 단점을 해결하기 위해 나온 노드다. 모든 블록 정보의 원본을 가지고 있지 않고 블록 헤더에 있는 중요한 데이터만 보유하고 있다. 그렇기 때문에 어떤 거래 정보를 수신 받았을 경우 이 거래가 정상적인지 검증할 수 없다.

다. 블록체인 특징 및 장단점

1) 특징

가) 탈중앙성

블록체인 시스템은 제 3자가 없고 자율적으로 참여자들에 의해서 운영되므로 탈중앙성격을 가진다. 제 3자 없이 네트워크 환경에서 거래가 가능하고 거래의 각 당사자간에 데이터를 분산해서 저장하는 구조다. 따라서 일부 사용자나 시스템에 대한 오류가 성능저하 등과는 무관하게 전체 네트워크에서 영향을 주는 범위가 미미하다.

나) 안정성

해킹 방법 중 '디도스³' 공격을 하게 되면 블록체인 시스템은 서버-클라이언트의 구조가 아니기에 특정 노드들을 멈추게 한다 하여도 전체 블록체인 시스템을 마비시킬 수 없다. 또 다른 해킹방법인 '랜섬웨어⁴'를 사용하게 된다면 기존에는 서버-클라이언트 구조에서 서버를 감염시켜 클라이언트의 접속을 제한할 수 있었다. 하지만 블록체인 시스템에서는 특정 노드들을 감염시킨다 하더라도 동시에 마비시킬 수 없다면 문제없다.

다) 신뢰성

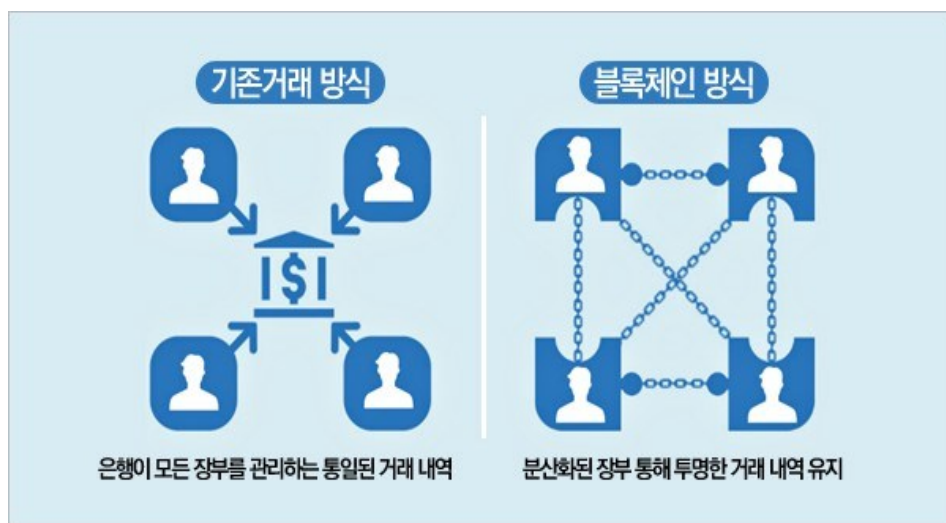
블록체인에 기록된 내용은 해시함수를 통해서 변환되어 암호화된 상태로 저장된다. 단 한 글자만 바꾸더라도 모든 해시 값이 변경되므로 특정 데이터를 변조하는 방식으로서는 해킹이 불가능하므로 신뢰성이 확보된다.

라) 기존 거래와의 차이점

³ 디도스(D-Dos) : 다수의 일반 pc를 이용해 특정 시스템으로 대량의 유해 트래픽을 전송함으로써 시스템 상에 과부하를 발생시켜 해당 시스템의 정상적인 서비스를 방해하는 해킹방법.

⁴ 랜섬웨어 : 컴퓨터 시스템을 감염시켜 접근을 제한하고 몸값을 요구하는 악성 소프트웨어의 한 종류.

기존 거래 방식은 은행이 모든 거래 내역을 가지고 있다. 두 사람이 안전하게 거래할 수 있도록 은행이 중간자 역할을 해준다. 블록체인은 거래 내역을 저장하고 증명한다. 전과 다른 점은 거래 내역을 은행이 아닌 여러 명이 나눠서 저장한다는 점이다. 만약 한 네트워크에 4명이 참여하고 있다면 A와 B의 거래 내역을 4개의 블록으로 생성해 모두에게 전송, 저장한다. 나중에 거래 내역을 확인할 때는 블록으로 나눠 저장한 데이터들을 연결해서 확인한다. 만약 변조된 거래가 일어난다면, 모든 공공장부들을 비교 검증을 하고 과반수 이상이 일치해야 거래가 인정된다. 이 때 필요한 것은 블록체인에 연결된 참여자들의 장부가 사용되기 때문에 중앙 서버가 필요 없다.



<그림 2> 기존 거래와의 차이점

2) 장점

- 가) 개인 정보를 요구하지 않는다. 기존 지급 수단에 비해 높은 익명성을 제공한다.
- 나) 은행 없이 P2P 방식으로 거래하기 때문에 불필요한 수수료를 절감할 수 있다.
- 다) 공개된 소스에 의해 쉽게 구축, 연결, 확장이 가능하다. IT 구축 비용 절감 가능하다.
- 라) 거래 내역이 공개되어 있어 원칙적으로 모든 거래에 공개적 접근이 가능하다.
- 마) 장부를 공동 소유하기 때문에 보안관련 비용을 절감할 수 있다.
- 바) 일부 참가 시스템에 오류 또는 성능저하 발생시 전체 네트워크가 받는 영향이 미미하다.

3) 단점

- 가) 51% 공격

블록체인의 전체 노드 중 50%를 초과하는 해시 연산력을 확보한 뒤, 거래 정보를 조작함으로써 이익을 얻으려는 해킹 공격을 말한다. 하지만 실제로는 사실상 불가능하다.

나) 결제 및 처리 가능한 거래건수가 실제 거래 규모 대비 미미하다.

다) 문제 발생시 책임소재가 모호하다.

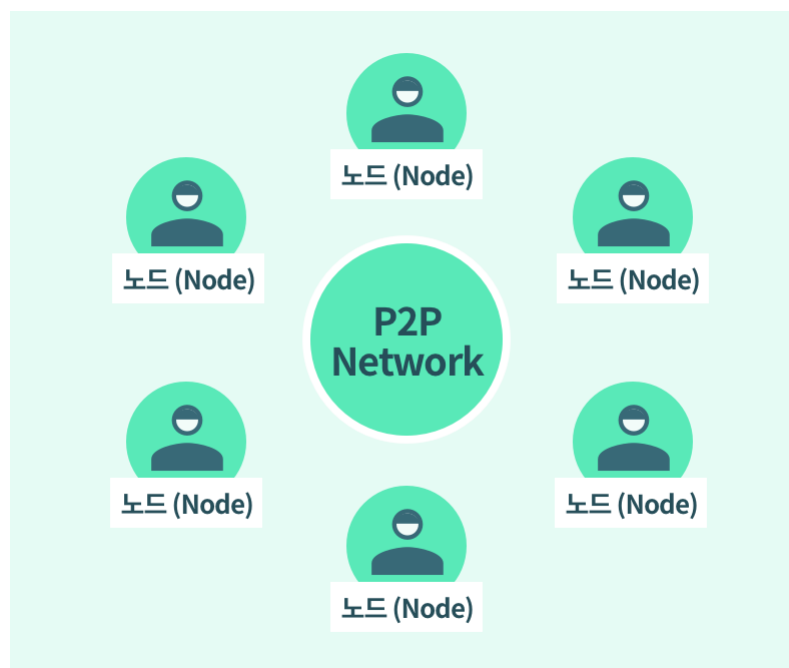
라) 개인키의 해킹, 분실 등의 경우 일반적으로 해결방법이 없다.

마) 실시간, 대용량 처리가 어렵다

라. 블록체인 핵심 기술

1) P2P(Peer-to-Peer)

P2P는 중앙 서버없이 각 단말들이 서로 동등한 입장에서 통신을 하는 네트워크다. 각 단말은 서버이기도 하면서 동시에 클라이언트가 된다. 블록체인은 p2p 형태의 이루어진 탈중앙화 네트워크다.



<그림 3> P2P

2) 해시(Hash)

해시함수는 입력 값 상관없이 비슷한 길이의 난수가 결과로 출력이 된다. 출력 값으로 입력 값을 예측할 수 없다. 그 때문에 장부가 많을 때도 변조 사실을 쉽게 알 수 있다. 원본의 장부를 해싱 했을 때 나오는 결과와 변조된 장부의 해싱 결과는 다르게 나오므로 이상하다는 것을 알 수 있다.

3) 채굴(Mining)

퍼블릭 블록체인에서 보상과 사용자에게 참여를 유도시키는 기술이다. 작업증명을 통해 블록에 거래 내역을 정리해주고 그 보상으로 코인과 거래 수수료를 받게 한다.

4) 스마트계약(Smart Contract)

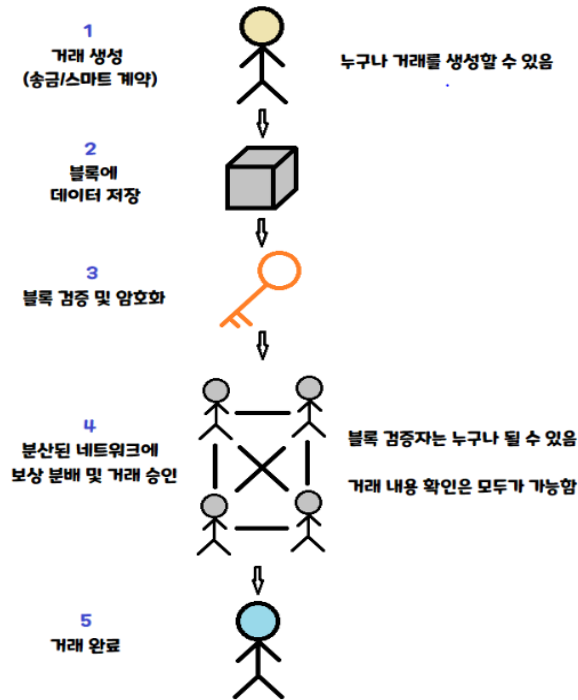
중개자없이 p2p로 쉽고 편리하게 계약을 체결하고 수정할 수 있는 기술이다. 블록체인에서 스마트계약은 계약 이행 및 검증 과정이 자동화되고 신뢰를 바탕으로 안전하게 계약을 실행할 수 있게 한다.

마. 플랫폼의 종류(Public Block-Chain vs Private Block-Chain)

1) Public Block Chain

가) 개념

퍼블릭 블록체인의 또 다른 명칭은 '무허가형 원장(Permissionless Ledger)'이다. 따라서 누구든지 허가없이 블록체인의 데이터를 읽고, 쓰고, 검증할 수 있다. 이로 인해 높은 보안성과 투명성을 보장된다. 또한 네트워크에 참여한 노드들은 채굴이라는 방식을 통하여 보상을 얻음으로써 네트워크를 지속적으로 유지하고 있다. 그러나 많은 참여자들의 합의가 진행되고 전체네트워크에 전파하여 동기화 하기 때문에 속도가 느리다.



<그림 4> 퍼블릭 블록체인(Public Block Chain)

나) 비트코인 : 블록체인 기술을 기반으로 만들어진 디지털 가상화폐이다.

다) 이더리움(Ethereum)

이더리움은 블록체인 기술을 기반으로 스마트 계약 기능을 구현하기 위한 분산 컴퓨팅 플랫폼으로, public block chain의 한 종류이다. 비트코인은 가상암호 화폐의 기능만 했다면 이더리움은 스마트컨트랙트를 가능하게 함으로써 블록체인 플랫폼이 되었다. 또한 이더리움은 오픈소스로 dApp개발도 유용하게 쓰인다.

라) Public Block Chain에서의 합의 알고리즘⁵

(1) 비트코인 – 작업증명 알고리즘(Proof of Work, PoW)

컴퓨팅 파워 사용하여 Nonce 값을 계산한 뒤 검증을 통해 합의 과정이 이루어진다.

⁵ 합의 알고리즘(consensus algorithm) : 다수의 참여자들이 통일된 의사결정을 하기 위해 사용하는 알고리즘을 말함.

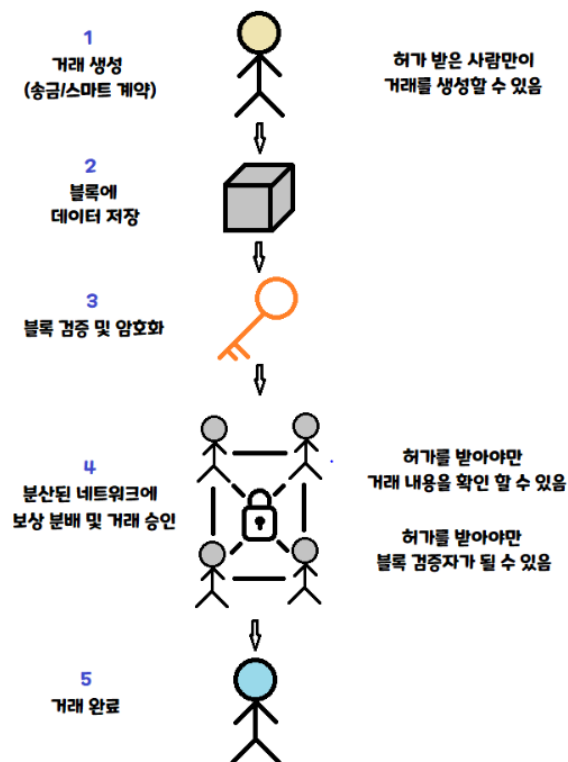
(2) 이더리움 – 지분증명 알고리즘(Proof of Stake, PoS)

노드가 보유한 자산을 기준으로 권한이 주어진다. PoW의 컴퓨팅 파워 낭비 문제를 해결할 수 있다.

2) Private Block Chain

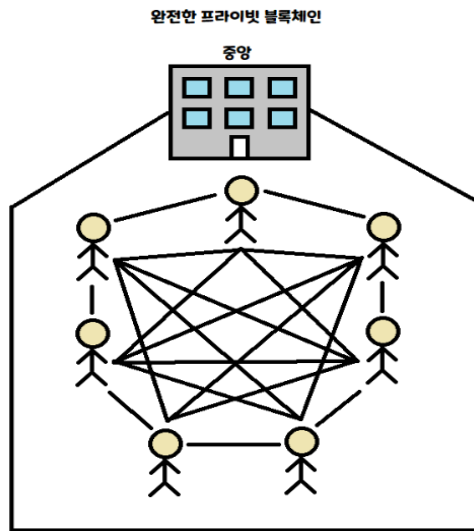
가) 개념

프라이빗 블록체인은 '허가형 원장(Permissioned Ledger)'으로도 불린다. 읽기,쓰기, 합의 과정에 참여할 수 있는 참여자가 미리 지정되어 있으며, 필요에 따라 특정 주체가 새로 추가되거나 제거될 수 있다. 또한 public blockchain과 가장 큰 차이점은 프라이빗 블록체인은 참여자와 관리자가 분리되어 있어 화폐 발행이 필수 조건은 아니다.



<그림 5> Private Block Chain

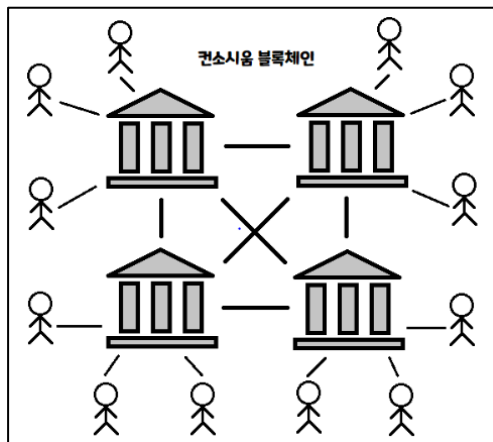
나) 완전한 프라이빗 블록체인(Fully-Private-Blockchain)



<그림 6> 완전한 프라이빗 블록체인

완전한 프라이빗 블록체인은 하나의 기관 안에서 독자적으로 사용하는 블록체인이다. 데이터 검증, 거래내역 접근은 허가를 받은 사람만 가능해진다. 블록체인이라는 이름을 갖고 있지만 완전한 탈 중앙화가 이루어지지 않고 분산 데이터 베이스만 갖고 있는 형태이다.

다) 컨소시움 블록체인(Consortium Blockchain)



<그림 7> 컨소시움 블록체인

컨소시움 블록체인은 하나의 목적이나 가치를 가지고 있는 여러 기관들이 하나의 그룹을 이루어 블록체인 네트워크를 구성한다. 분류는 프라이빗 블록체인으로 되지만 사실상 퍼블릭과 프라이빗의 두 가지 면을 다 갖고 있다.

라) 하이퍼레저(Hyperledger)

이더리움이나 비트코인 같은 public blockchain은 금융기관,기업 등의 기밀 유지가 필요한 환경에서는 적합하지 않은 경우가 있었다. 그의 해결방법은 private blockchain 인데, 기업들이 이를 발저시키기 위하여 연합한 것을 하이퍼레저라고 한다. 퍼블릭 블록체인과 다르게 특정 허가를 받아야지만 네트워크 접근이 가능해지기 때문에 더 빠른 속도의 네트워크로 구성되어 있다.

마) Private Block Chain 에서의 합의 알고리즘

PBFT(Practical Byzantine Fault Tolerance)

PoW 와 PoW 의 알고리즘을 보완하기 위해 나온 알고리즘이다. 기존에는 전체 노드의 참여와 동의를 필요했지만, 이 방식은 2/3 이상의 노드만 합의가 가능하면 된다. 즉, 속도를 더 높일 수 있는 방법이다. 하지만 일부 중앙집권으로 인해 블록체인의 완벽한 탈 중앙화를 이루진 못했다.

2. 디앱(dApp)

가. 디앱의 개념

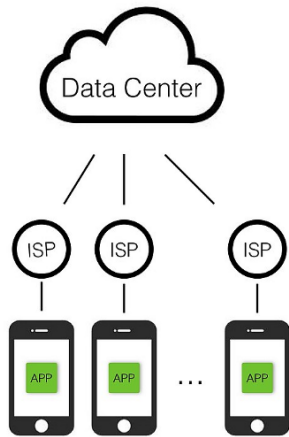
기존의 거래만 가능했던 블록체인 1.0을 벗어나서 블록체인 2.0인 이더리움이 등장하며 다양한 분야로 확장이 가능하게 된다. 이는 스마트 계약을 기반으로 각 서비스의 성격에 맞는 탈중앙화된 어플리케이션, 디앱을 개발할 수 있다. 디앱은 블록체인 기술을 활용하여 중앙 서버 없이 네트워크 상에 정보를 분산하여 저장 및 구동하는 앱이다.

나. 앱과 디앱(dApp)의 차이점

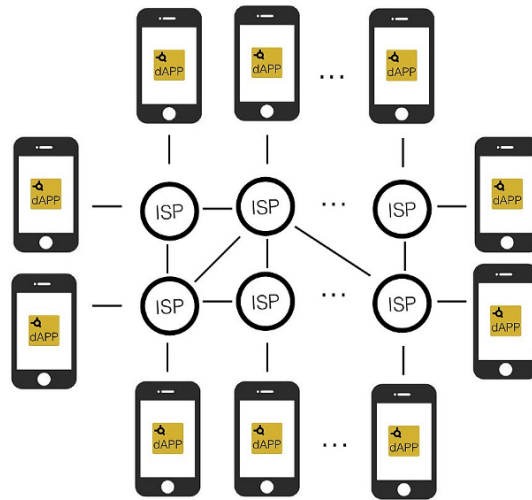
일반 앱은 회원가입을 할 때 개인정보를 입력해서 계정을 만들고 로그인 후 즉시 서비스 이용 가능하다. 하지만 디앱은 계정을 만들 때, 블록체인 지갑을 보유해야 한다. 이는 저장 또는 거래에 사용될 암호화폐를 위한 지갑을 의미한다. 가장 큰 차이점은 디앱은 제품 공개 전 엄격한 테스트 후 공개한다는 점이다. 왜냐하면 스마트 계약이 메인넷⁶에서 한번 실행되면 절대 변경이 안 되기 때문이다.

⁶ 메인넷(Main-Network) : 블록체인 프로젝트를 실세 출시하여 운영하는 네트워크.

Apps



dApps



<그림 8> 앱과 디앱의 차이점

다. 디앱의 전망

2018년 이래 디앱의 전망이 두드러진다. 현금화가 용이한 디앱의 특성상, 게임 분야가 지속해서 강세를 유지할 예정이다. 지금 대부분의 디앱은 게임 분야에 한정되어 있지만, 점차 다른 분야에서 더욱 발전하고 있다. 이에 디앱의 성공이 블록체인 성공에도 서로 기여할 수 있을 것으로 전망된다. 이미 진행되고 있는 사물 인터넷 등 다른 기술과 연계된 디앱의 성공을 기대해 볼 수 있다.

3. 작품의 작동원리

가. 사용 Language

1) Solidity

- 다양한 블록체인 플랫폼의 스마트 컨트랙트 작성 및 구현에 사용되는 계약 지향 프로그래밍 언어.
- Solidity를 통해 서비스로 실행되는 비즈니스 로직을 스마트 컨트랙트에 담아 디앱을 구현할 수 있음.
- Solidity로 작성된 코드는 EVM⁷에서 작동 가능한 바이트코드로 컴파일 됨.

2) HTML/CSS/JavaScript

- 웹 프론트엔드 코딩을 위한 언어
- HTML : 웹 페이지의 구조와 내용을 작성하기 위한 언어
- CSS : 웹 페이지가 브라우저에 출력되는 모양을 표현하기 위한 언어
- JavaScript : 웹 페이지의 행동이나 응용프로그램을 처리하기 위한 언어

3) JSP

- HTML 내에 자바 코드를 삽입하여 웹 서버에서 동적으로 웹 페이지를 생성하여 웹 브라우저에 돌려주는 언어

나. 사용 Tools

1) 인퓨라(Infura)

- 클라우드로 제공되는 풀 노드 HTTP API 서비스
- 사용자가 자신의 Ethereum 노드를 설정하지 않고 디앱을 실행할 수 있는 호스팅 된

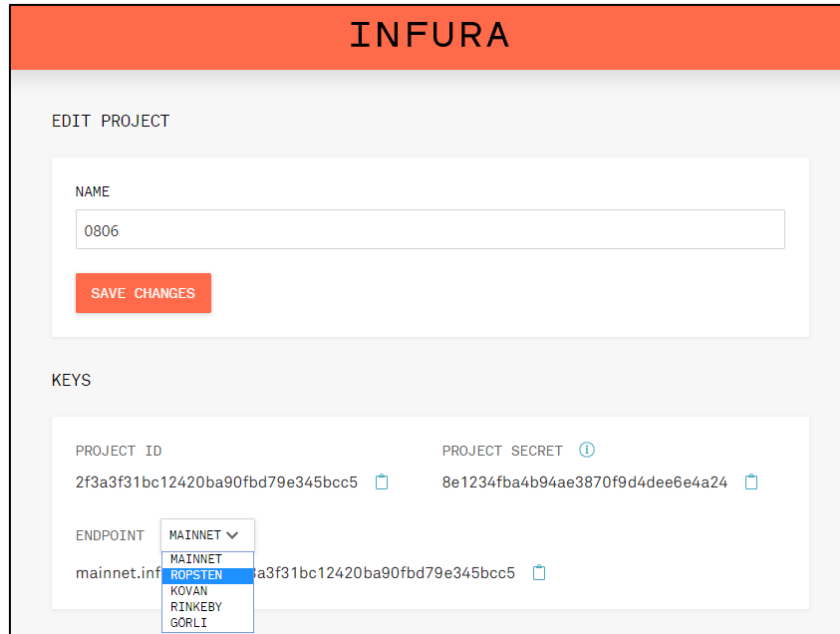
⁷ EVM(Ethereum Virtual Machine) : 이더리움 블록체인 네트워크의 노드들이 공유하는 하나의 가상 머신, 하나의 분산 컴퓨터

Ethereum 노드 클러스터

- 보안 상의 이유로 개인 키를 관리하지 않기 때문에 HD Wallet Provider 필요

가) <https://infura.io>에 접속 후 프로젝트 생성

나) ENDPOINT에서 ROPSTEN 선택 후 주소 복사하여 사용



<그림 9> INFURA 프로젝트

2) HD Wallet Provider

- Infura를 통해 이더리움 네트워크에 스마트 컨트랙트를 배포할 때 대신 서명을 해주는 공급자

가) 명령 프롬프트 관리자 권한으로 실행

나) Windows 빌드 도구 설치

```
C:\Project\0806>npm install -g windows-build-tools
```

<그림 10> Windows 빌드 도구 설치

다) HD Wallet Provider 설치

3) 트러플(Truffle)

- npm에서 제공하는 스마트 컨트랙트 컴파일, 배포, 관리, 테스트를 돕는 프레임워크

가) Truffle 설치

```
C:\Project\0806>npm install -g truffle
```





<그림 11> Truffle 설치

나) Truffle 초기화

```
C:\Project\0806>truffle init
```

<그림 12> Truffle 초기화

나)-1 생성된 폴더

	contracts	2019-08-06 오후...	파일 폴더
	migrations	2019-08-06 오후...	파일 폴더
	test	2019-08-06 오후...	파일 폴더
	truffle-config	2019-08-06 오후...	JavaScript 파일

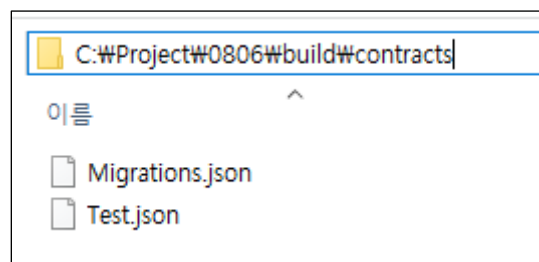
<그림 13> 생성된 폴더의 모습

다) 스마트 컨트랙트 컴파일

```
C:\Project\0806\contracts>truffle compile
```

<그림 14> Smart Contract 컴파일

다)-1 ABI 형식의 JSON 파일 생성



<그림 15> JSON 파일 생성

라) migrations 폴더에 2_deploy_contracts.js 파일 생성

```
1  const Test = artifacts.require("../contracts/Test");
2
3  module.exports = function(deployer) {
4    deployer.deploy(Test);
5  };
```

2_deploy_contracts.js

<그림 16> 2_deploy_contracts.js 생성

마) truffle-config.js 파일 수정

마)-1 HDWalletProvider 주석 해제, mnemonic 변수에 배포자의 메타마스크 계정 시드 삽입

```
21 const HDWalletProvider = require('truffle-hdwallet-provider');
22 // const infuraKey = "fj4jll3k.....";
23 //
24 // const fs = require('fs');
25 const mnemonic = 'trophy emerge crowd never fiction luggage large price member carbon length license';
```

truffle-config.js

<그림 17> 배포자의 메타마스크 계정 Seed 삽입

마)-2 network 중 ropsten 주석 해제, new HDWalletProvider(mnemonic, Infura Project ENDPOINT)

```
63 ropsten: {
64   provider: () => new HDWalletProvider(mnemonic, 'https://ropsten.infura.io/v3/2f3a3f31bc12420ba90fbd79e345bcc5'),
65   network_id: 3, // Ropsten's id
66   gas: 5500000, // Ropsten has a lower block limit than mainnet
67   confirmations: 2, // # of confs to wait between deployments. (default: 0)
68   timeoutBlocks: 200, // # of blocks before a deployment times out (minimum/default: 50)
69   skipDryRun: true // Skip dry run before migrations? (default: false for public nets )
70 },
```

truffle-config.js

<그림 18> Network 중 Ropsten 주석 해제

바) 스마트 컨트랙트 배포

```
C:\WPProject\W0806\contracts>truffle migrate --network ropsten
```

<그림 19> Smart Contract 배포

4) 메타마스크(MetaMask)

- 사용자가 이더리움 블록체인 네트워크와 통신할 수 있게 하는 이더리움 지갑(Chrome 확장 프로그램)

가)

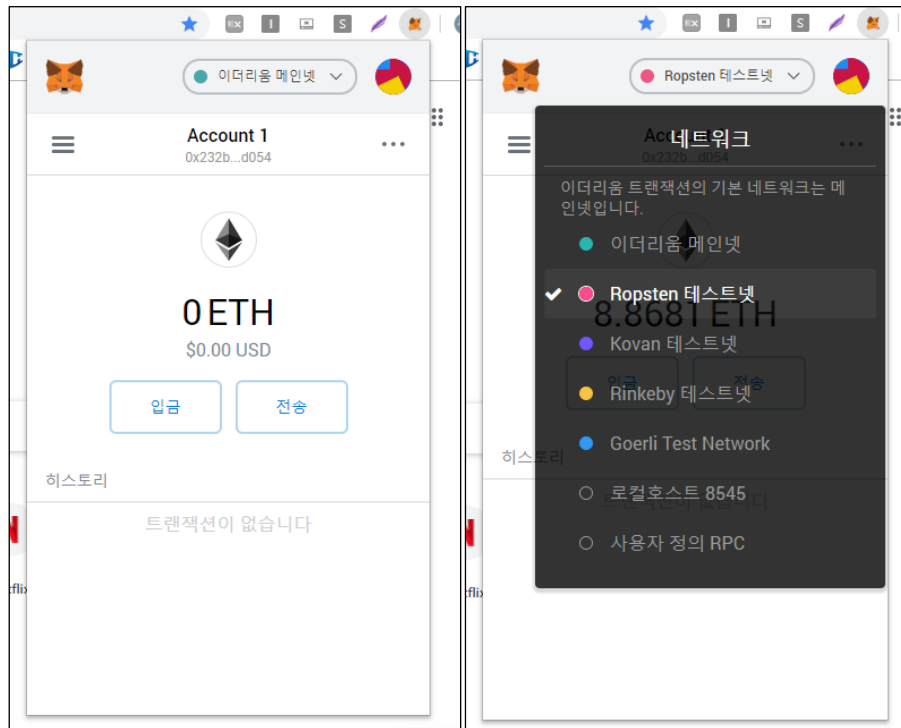
<https://chrome.google.com/webstore/detail/metamask/nkbihfbeogaeaoehlefnkodbefgpgknn?hl=ko> 접속 후 Chrome에 추가



<그림 20> 메타마스크 Chrome에 추가

나) 시드 구문 생성 (12개의 단어로 구성된 구문, 백업 필요)

다) 로그인 후 네트워크를 Ropsten 테스트넷으로 설정



<그림 21> 메타마스크 화면

<그림 22> 메타마스크 네트워크 설정

5) 이클립스(Eclipse)

- JAVA를 비롯한 다양한 언어를 지원하는 프로그래밍 통합 개발 환경(IDE)

6) Web3.js

- 웹에서 이더리움 네트워크를 접근하게 해주는 JavaScript용 API
- JSON-RPC:8545의 통신을 통해 직접적으로 노드에 접근 가능

7) 부트스트랩(Bootstrap)

- 웹사이트를 쉽게 만들 수 있게 도와주는 HTML/CSS/JavaScript 프레임워크

8) Apache / Tomcat

- Apache : 아파치 소프트웨어 재단에서 관리하는 HTTP 웹 서버(Web Server)

- Tomcat⁸ : 아파치 소프트웨어 재단에서 개발한 서블릿 컨테이너(또는 웹 컨테이너)만 있는 웹 애플리케이션 서버(WAS, Web Application Server)

- Tomcat에 내장된 웹 서버로만 웹 시스템을 구성할 수 있지만, 대규모의 사용자가 사용하는 시스템을 구축하려면 Apache 웹 서버와 연동하는 안정적인 시스템을 구축해야 한다. mod_jk 모듈을 사용하여 Apache와 Tomcat을 연동해 웹 서버를 구축한다.

9) MySQL

- 구조 질의어 형식의 관계형 데이터베이스 관리 시스템(RDBMS)

10) Visual Studio Code

- 다양한 프로그래밍 언어를 지원하는 소스 코드 편집기
- 스마트 컨트랙트를 작성하는 편집 툴

⁸ 여기에서는 Tomcat 8.5 버전을 사용함.

다. 작품 전체 작동 원리

1) 네트워크

이더리움은 다양한 네트워크를 보유하고 있다. 개발자들이 개발한 이더리움 애플리케이션을 사용자들에게 정식 서비스를 하기 위한 '메인 네트워크(Main Network)'가 있다. 이 때, 메인 네트워크 구축 및 배포하기 전에 다양한 조건에서 개발자들이 애플리케이션을 테스트할 수 있는 '테스트 네트워크(Test Network)'가 존재한다.

가) 메인 네트워크(Main-Network)

메인넷(Main-net)이란, 기존 사용 플랫폼(이더리움, 쿼텀, 네오 등)에서 벗어나 블록체인 프로젝트를 실제 출시하여 운영하는 네트워크를 말한다. 즉, 독립적인 생태계를 구성하는 것을 말한다. 일반적으로 메인넷이 구축되기 까지는 절차들이 존재한다.

- (1) 기존 플랫폼(이더리움, 쿼텀 등)을 기반으로 토큰을 제작하고, ICO⁹를 진행한다.
- (2) 독자적인 플랫폼으로 자리잡을 수 있는지 테스트넷에서 테스트를 진행한다.
- (3) 테스트넷이 성공하면, 메인넷 릴리즈를 하여, 독립적인 플랫폼으로 트랜잭션과 생태계를 구성하며 독립적인 지갑을 생성한다.
- (4) 토큰이 코인으로 전환된다.

나) 테스트 네트워크(Test-Network) - Ropsten

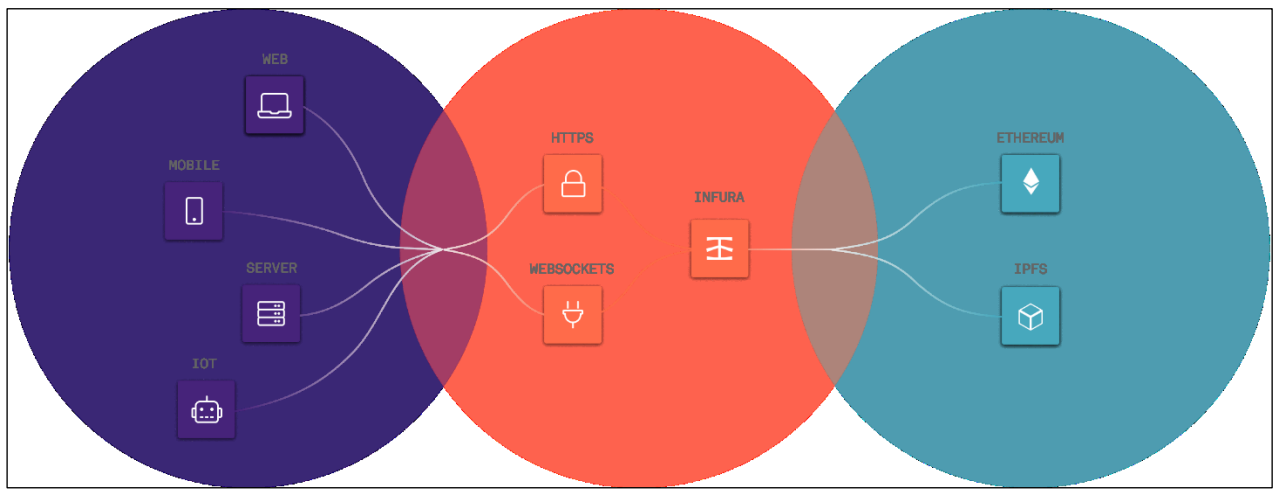
Ropsten 네트워크는 PoW¹⁰ 기반의 테스트 네트워크이다. Ropsten 테스트 네트워크는 테스트용 이더(Ether)를 Ropsten Faucet에서 개발자의 해당 네트워크 계정 주소로 직접 받아서 개발자가 디앱(dApp)을 테스트한다. 이 네트워크에서는 테스트용 이더를 실제로 채굴도 할 수 있다. 메인넷과 가장 유사한 테스트 환경을 가지고 있기에 다양한 테스트 네트워크 중에 Ropsten을 테스트 네트워크로 지정하게 되었다.

⁹ ICO(Initial Coin Offering) : 새로운 암호화폐를 만들기 위해 불특정 다수의 투자자들로부터 초기 개발 자금을 모집하고, 그 대가로 코인을 나눠주는 행위를 말한다.

¹⁰ Pow(Proof of Work) : 컴퓨터가 연산을 수행해 블록 생성 및 검증에 기여한 대가로 보상을 받는 채굴 방식.

2) 인퓨라 (INFURA)

디앱(dApp)을 로컬 환경에서 개발하기 위해서는 geth와 같은 이더리움 클라이언트를 구축하거나 Truffle에서 제공하는 Ganache¹¹를 사용하여 개발을 하게 된다. 개발이 완료되었다고 판단이 들면 테스트넷 환경에서 실행을 해보아야 하는데 이더리움 클라이언트를 사용하여 테스트 환경을 구축하게 되면 완전한 동기화(full sync)까지 최소 하루 이상이 걸리게 되고, 저장공간 또한 많이 들게 된다. 이러한 불편함을 해소시킬 방법을 찾다가 INFURA API를 알게 되었다.



<그림 23> 출처 : INFURA 공식 홈페이지(infura.io)

Infura는 HTTPS나 웹소켓을 통해 이더리움 및 IPFS 네트워크에 즉시 액세스 할 수 있게 해주는 Backend API이다. Infura를 사용하면 아래 장점들이 있기에 이 API를 쓰게 되었다.

(1) Accessibility(접근성) : 최소한의 하드웨어만 요구하기 때문에 접근성이 보다 용이해진다.

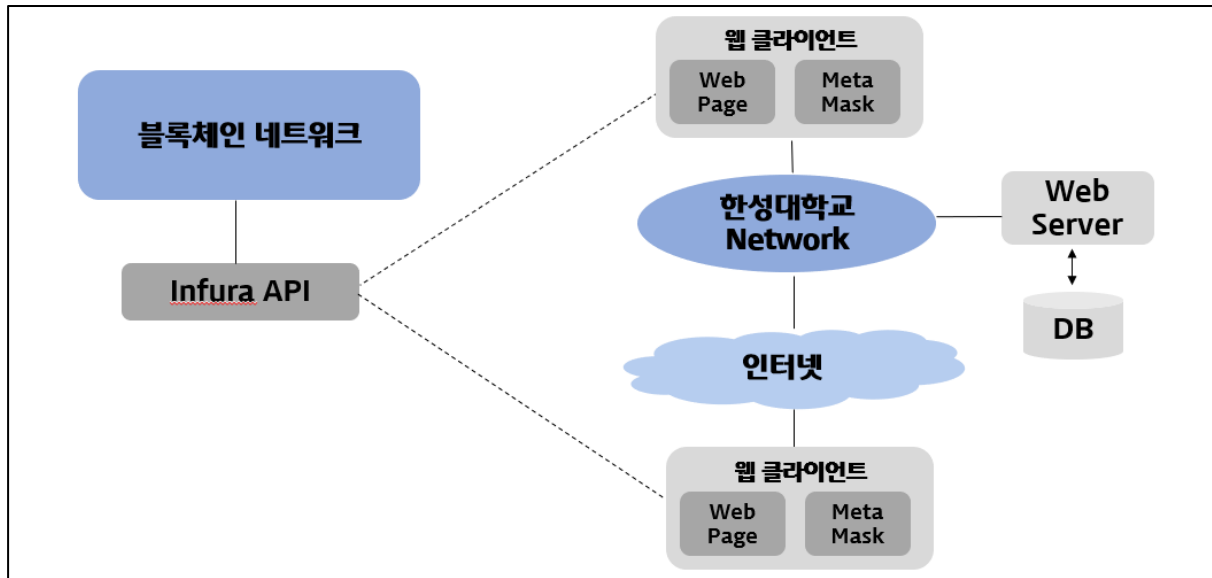
(2) Convenience(용이성)

: 직접 노드를 만들게 되면 노드를 설치하고, 동기화하는 작업이 필요하지만 Infura을 사용함으로써 사용자의 수고가 줄어든다.

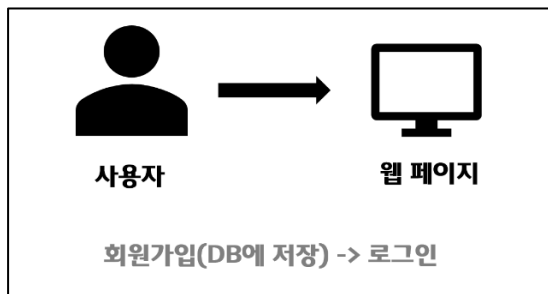
(3) Security(보안성) : 웹 접속 없이 서명할 수 있다.

¹¹ Ganache(가나슈) : 로컬환경에서 가상의 계좌를 제공하여 블록체인 테스트를 도와주는 프로그램.

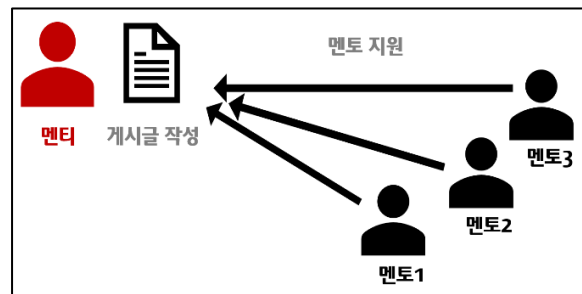
3) 시스템 작동 원리



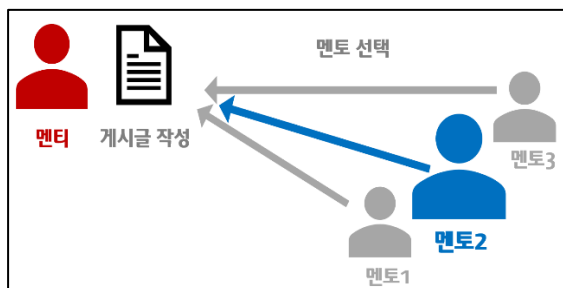
<그림 24> 시스템 설계



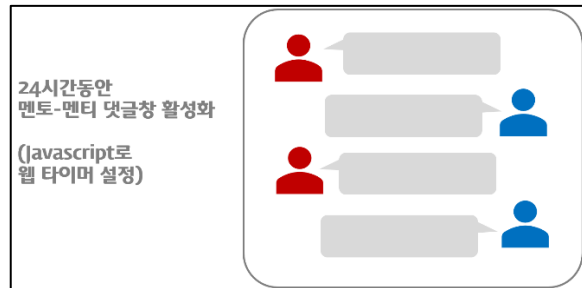
<그림 25> 웹 작동원리 1



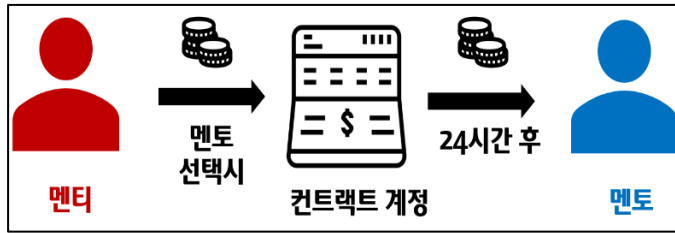
<그림 26> 웹 작동원리 2



<그림 27> 웹 작동원리 3



<그림 28> 웹 작동원리 4



<그림 29> 웹 작동원리 5



<그림 30> 웹 작동원리6

[조건1 : 특정 조직(학교 혹은 회사 등)에서도 작품을 사용한다고 가정. 그 중, 한성대학교로 한정함.]

[조건2 : 웹페이지를 사용하는 사용자들은 '메타마스크' 확장 프로그램을 깔았다고 가정함.]

작품을 만들기 전에 작품의 시스템 설계를 먼저 하였다. (<그림 2> 참고) 서론에서 말했다시피, 이 시스템을 사용하게 되는 사람들은 모두 멘티가 될 수 있으며, 또한 모든 이가 멘토가 될 수 있다. 아래에서 작품의 작동 원리에 대해서 자세히 설명하겠다.

<그림 2>와 같이 웹서버 및 데이터 베이스(DB)를 한성대학교 네트워크 내부에 구축한다. 이 네트워크의 내 · 외부에서 사용자들이 웹서버에 접속하게 되면 디앱 페이지를 서비스 받게 된다.

페이지를 처음 이용할 시, 회원가입 절차를 진행하게 된다. 이 때, 입력된 회원 정보는 웹서버와 연결된 DB에 저장된다. 가입자의 성명, 아이디, 비밀번호, 성별, 이메일, 메타마스크 계정 주소가 저장된다. (<그림 3> 참고) 가입 시 적은 아이디와 비밀번호로 로그인을 하게 된 순간부터 모든 사람들은 멘토임과 동시에 멘티가 된다.

사용자들이 각 분야 별로 질문을 올리거나 멘토를 구한다는 게시물을 작성하게 되면 모든 사용자들은 그것을 볼 수 있고, 자신을 제외한 사람들은 그 게시물에 멘토 지원이 가능하다. (<그림 4> 참고) 게시물 작성자인 멘티는 자신이 작성한 글의 지원자의 지원글을 보고 한 명의 멘토를 선택할 수 있다. 그 순간부터 멘토 - 멘티만의 24시간 댓글창이 활성화가 된다. 이 댓글창은 선택된 멘토와 게시글을 작성한 멘토 이외에는 그 누구도 볼 수가 없다. (<그림 5> 참고)

이 과정에서 블록체인에 배포된 스마트 컨트랙트가 실행되면서 송금 트랜잭션이 Infura API를 통해 생성되고, 블록체인 네트워크에 블록의 형태로 저장된다. 멘티가 가입 시 적었던 메타마스크 계정에서 컨트랙트 계정¹²으로 가상화폐가 빠져나간다. 웹 타이머로 24시간이 지나면 멘티 계정에서 빠져나갔던 가상 화폐가 멘토의 메타마스크 계정으로 송금이 된다. (<그림 7> 참고)

송금이 완료된 이후에도 계속 멘토-멘티 관계를 연장하고 싶을 경우 양 측에서 모두 수락하

¹² 컨트랙트 계정(Contract Account) : Contract Code 실행 시, 자동으로 생기는 계정.

게 되면 위에서 설명한 과정이 계속 반복되고, 한 명이라도 거절하게 되면 그 순간 관계는 종료된다.

IV. 참고 문서

[블록체인]

<https://banksalad.com/contents/%EB%B8%94%EB%A1%9D%EC%B2%B4%EC%9D%B8-%EA%B0%9C%EB%85%90-%EC%99%84%EB%B2%BD-%EC%A0%95%EB%A6%AC-dh1do>

<http://wiki.hash.kr/index.php/%EB%85%B8%EB%93%9C#.ED.92.80.EB.85.B8.EB.93.9C>

[dApp]

<https://medium.com/grabityio/dapp-%EB%94%94%EC%95%B1-%EB%8C%91-%EC%97%90-%EA%B4%80%ED%95%98%EC%97%AC-%EC%95%84%EC%A3%BC-%EA%B0%84%EB%8B%A8%ED%9E%88-%EC%95%8C%EC%95%84%EB%B3%B4%EC%9E%90-f852c6b60a75>

<http://wiki.hash.kr/index.php/%EB%94%94%EC%95%B1#.ED.8A.B9.EC.A7.95>

[이미지]

<https://towardsdatascience.com/what-is-a-dapp-a455ac5f7def>

<https://medium.com/@elamachain/%EB%B8%94%EB%A1%9D%EC%B2%B4%EC%9D%B8-blockchain-%EC%9D%B4%EB%9E%80-69eada649cd4>

<https://banksalad.com/contents/%EC%89%BD%EA%B2%8C-%EC%84%A4%EB%AA%85%ED%95%98%EB%8A%94-%EB%B8%94%EB%A1%9D%EC%B2%B4%EC%9D%B8-%EB%B8%94%EB%A1%9D%EC%B2%B4%EC%9D%B8%EC%9D%98-%EC%9B%90%EB%A6%AC-%EC%B1%84%EA%B5%B4-%ED%95%B4%EC%8B%9C-%EA%B7%B8%EB%A6%AC%EA%B3%A0-%EC%9E%91%EC%97%85%EC%A6%9D%EB%AA%85-qvCud>

<https://brownbears.tistory.com/371>

<https://cryptochain.tistory.com/42>

<https://pusiu.tistory.com/15>

<https://medium.com/@drhot552/%EC%89%BD%EA%B2%8C-web3-js-%EB%A1%9C-%EA%B0%9C%EB%B0%9C%ED%95%98%EA%B8%B0-1-b2035fab95e6>

<http://blog.naver.com/PostView.nhn?blogId=kss9409&logNo=221546048337>

<https://www.trufflesuite.com/docs/truffle/reference/configuration#general-options>

<https://www.trufflesuite.com/tutorials/using-infura-custom-provider>

<https://ko.wikipedia.org/wiki/%EC%86%94%EB%A6%AC%EB%944%ED%8B%B0>

- <https://wjt21952.tistory.com/67>

- <https://www.ethereum.org/developers>

- <http://wiki.hash.kr/index.php/ICO>

- <https://medium.com/bitfwd/get-ropsten-ethereum-the-easy-way-f2d6ece21763>

-

<https://medium.com/hexlant/%EC%9D%B4%EB%8D%94%EB%A6%AC%EC%9B%80-%ED%85%8C%EC%8A%A4%ED%8A%B8%EB%84%B7-%EC%86%8C%EA%B0%9C-eff82e237cde>

- <https://infura.io>

- <https://steemit.com/kr/@yahweh87/eoa>