2017 상명대학교
캡스톤 디자인

연두

컴퓨터소프트웨어공학과
김보연
이연희
지도교수 이종혁 교수님

Paramon

01

서론

Paramon 이란?

서론	본론	결론
Paramon 이란?	기능 소개	코드

Paramon

서론

01. Paramon이란 무엇인가?

중고 물품 거래 사이트에 이더리움 블록체인 개념을 도입하여
공공장부, 자동화 거래, 가상 화폐를 이용하여 사용자끼리 신뢰
를 가지고 안전하게 거래할 수 있는 중고 물품 거래 서비스

Paramon

서론

01. Paramon의 기능

- 공공 장부 / 분산 저장

사기거래에 대비할 수 있는 거래 내역을 보유.

(모든 사용자가 장부를 가지고 있으므로 거래 내역을 쉽게 변조할 수 없음)

- 스마트 컨트랙트

거래가 자동화로 이루어지므로 거래 불발로 인한 피해를 입을 확률이 낮음.

(거래 중 판매자가 돈을 받고 물건을 보내지 않을 경우,

거래자 중 한 명이라도 연락 두절일 경우)

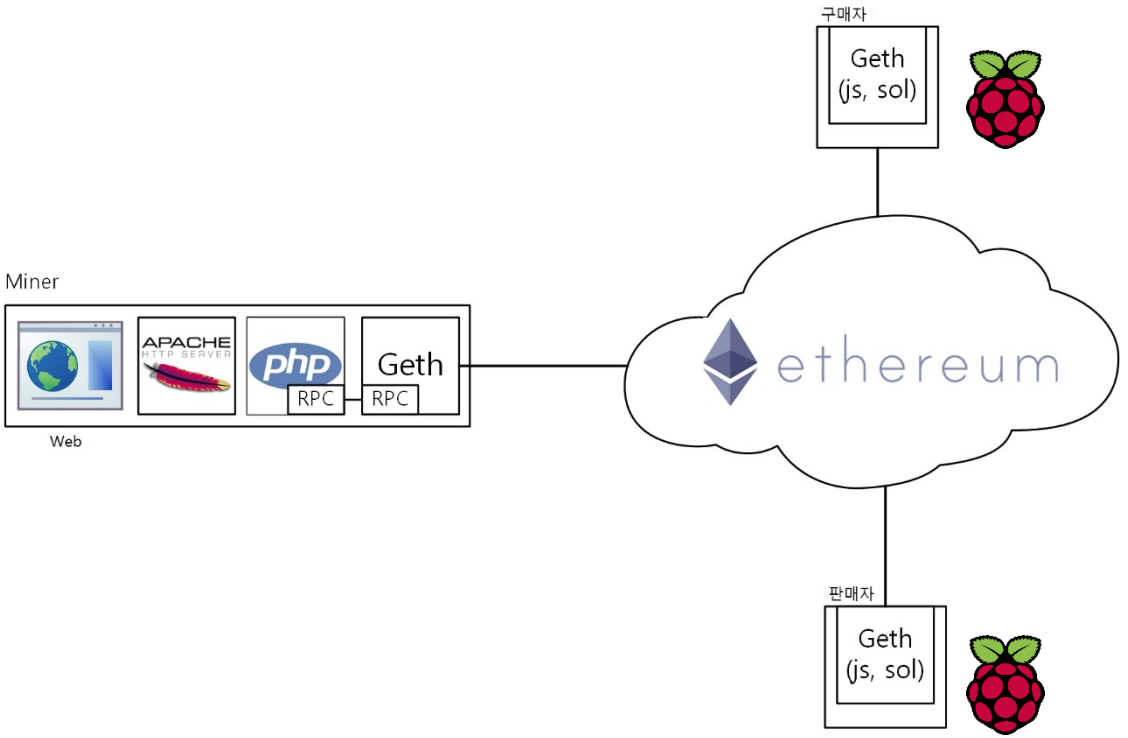
- 가상 화폐

가상의 돈으로 거래가 진행이 되어 실제 돈에 대한 피해를 막을 수 있음.

Paramon

서론

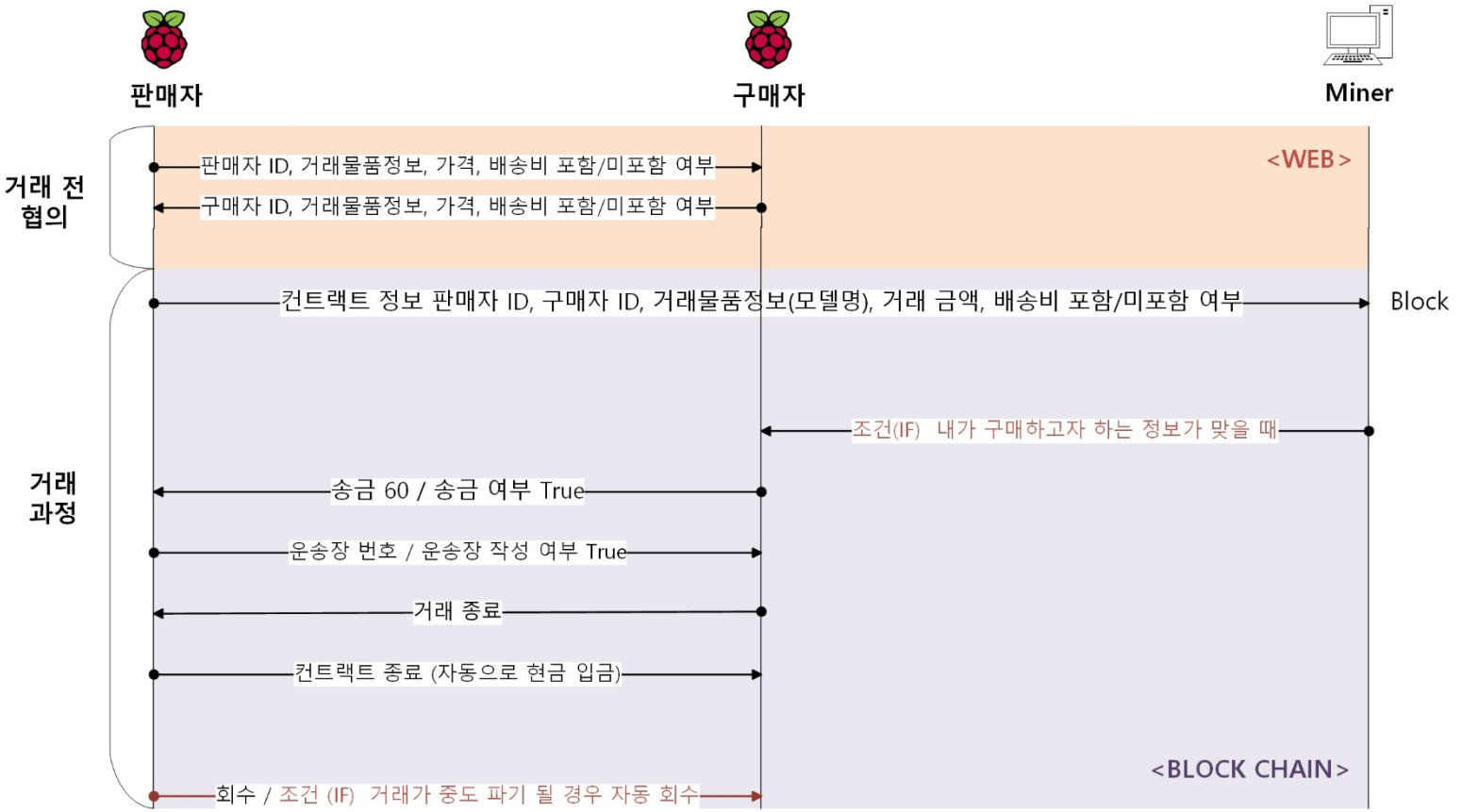
02. 네트워크 구성도



Paramon

서론

03. 동작 과정



Paramon

02

본론

기능 소개

서론

Paramon 이란?

본론

기능 소개

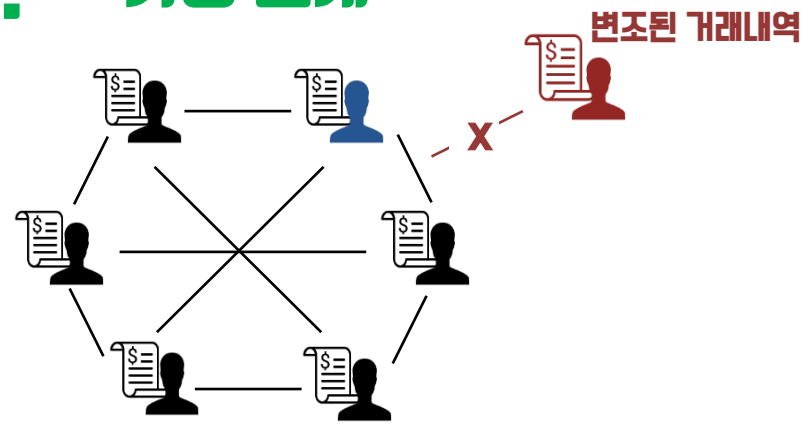
결론

코드

본론

04. 기능 소개

공공장부

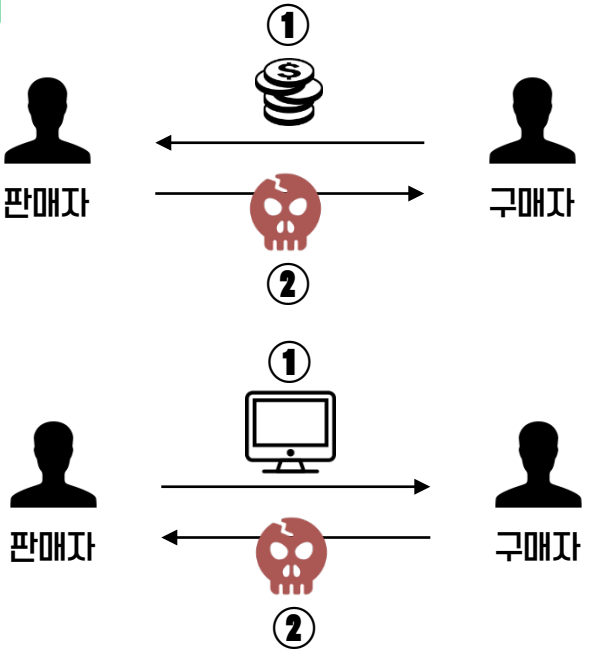


Paramon

본론

04. 기능 소개

스마트 컨트랙트



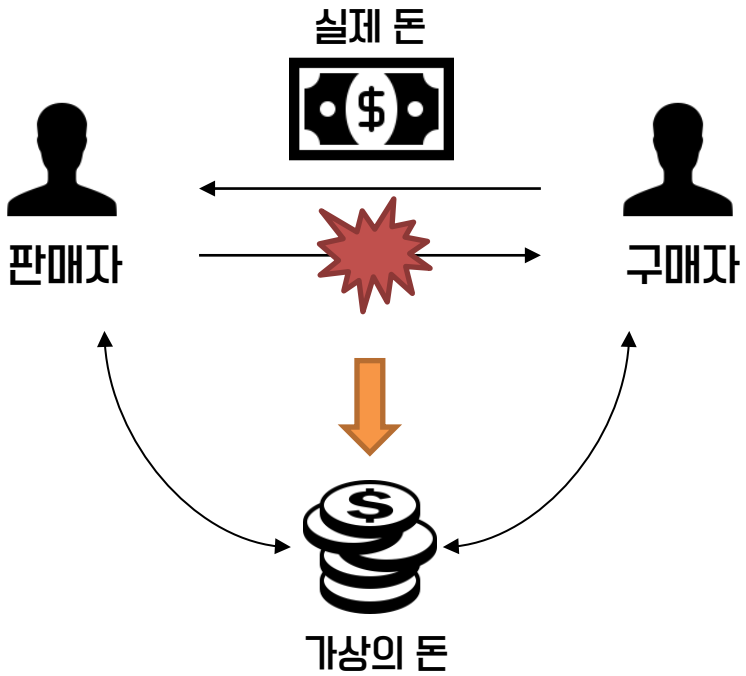
- 거래 시작 3일 이내 중간에 연락이 없을 경우, 돈이 자동으로 구매자에게 반환

자동화 거래방식

본론

04. 기능 소개

가상 화폐



Paramon

03

결론

코드

서론

Paramon 이란?

본론

기능 소개

결론

코드

Paramon

결론 시현 동영상 / 이미지

```
contract PurchaseContract
{
    uint public value;
    address public seller;
    address public buyer;
    enum State { Inactive, Created, Locked }
    State public state;
    function Purchase()
    {
        seller = msg.sender;
        value = msg.value / 2;
        if (2 * value != msg.value) throw;
        state = State.Created;
                                created();
    }
    modifier require(bool _condition)
    {
        if (!_condition) throw;
        _
    }
    modifier onlyBuyer()
    {
        if (msg.sender != buyer) throw;
        _
    }
    modifier onlySeller()
    {
        if (msg.sender != seller) throw;
        _
    }
    modifier inState(State _state)
    {
        if (state != _state) throw;
        _
    }
                event created();
    event aborted();
    event purchaseConfirmed();
    event itemReceived();
    event refunded();
```

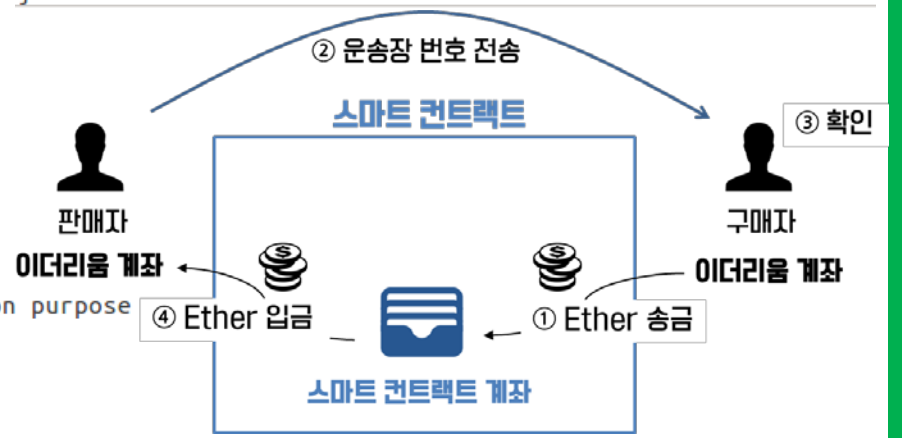
Paramon

결론

```
function abort()
    onlySeller
    inState(State.Created)
{
    seller.send(this.balance);
    state = State.Inactive;
    aborted();
}
/// Confirm the purchase as buyer.
/// Transaction has to include `2 * value` ether.
/// The ether will be locked until confirmReceived
/// is called.
function confirmPurchase()
    inState(State.Created)
    require(msg.value == 2 * value)
{
    buyer = msg.sender;
    state = State.Locked;
    purchaseConfirmed();
}
/// Confirm that you (the buyer) received the item.
/// This will release the locked ether.
function confirmReceived()
    onlyBuyer
    inState(State.Locked)
{
    buyer.send(value); // We ignore the return value on purpose
    seller.send(this.balance);
    state = State.Inactive;
    itemReceived();
}
```

```
    function refundBuyer()
        onlySeller
        inState(State.Locked)
    {
        buyer.send(2 * value);
        seller.send(this.balance);
        state = State.Inactive;
        refunded();
    }
    function getBalance() returns (uint balance) {
        return this.balance;
    }

    function() { throw; }
}
```



Paramon

결론 시현 동영상 / 이미지

```
Template.purchase.events({
  'click .create-contract': function () {

                var transactionObject = {
                        data: PurchaseContract.bytecode,
                        gasPrice: web3.eth.gasPrice,
                        gas: 300000,
                        from: web3.eth.accounts[0],
                };

                PurchaseContract.new(transactionObject, function(err, contract) {
                        if (!err) {
                                var txId = Helpers.makeId('tx', contract.transactionHash);
                                Transactions.upsert(txId, { $set: {
                                        transactionHash: contract.transactionHash,
                                        address: contract.address
                                } });

                                if (!contract.address) {
                                        Session.set('contractTxHash', contract.transactionHash);
                                } else {
                                        Session.set('contractAddress', contract.address);
                                }
                        } else {
      console.log(err);
    }
                });
```

Paramon

결론 시현 동영상 / 이미지

```
},
      'click .purchase': function() {
              var txObject = {
                      value: 20,
                      gas: 300000,
                      from: web3.eth.accounts[0]
              };
              var contractInstance = PurchaseContract.at(Session.get("contractAddress"));
              contractInstance.Purchase(null, txObject, function(err, result){
                      if (!err) {
                              var txId = Helpers.makeId('tx', result);
                              Transactions.upsert(txId, { $set: {
                                      transactionHash: result
                              } });
                      }
              });
      },
'click .abort' : function() {
              var txObject = {
                      value: 0,
                      gas: 300000,
                      from: web3.eth.accounts[0]
              };
              var contractInstance = PurchaseContract.at(Session.get("contractAddress"));
              contractInstance.abort(null, txObject, function(err, result){
                      if (!err) {
                              var txId = Helpers.makeId('tx', result);
                              Transactions.upsert(txId, { $set: {
                                      transactionHash: result
                              } });
                      }
              });
},
```

Paramon

결론 시현 동영상 / 이미지

```
'click .confirmPurchase' : function() {
        var txObject = {
                value: 20,
                gas: 300000,
                from: web3.eth.accounts[1]
        };
        var contractInstance = PurchaseContract.at(Session.get("contractAddress"));
        contractInstance.confirmPurchase(null, txObject, function(err, result){
                if (!err) {
                        var txId = Helpers.makeId('tx', result);
                        Transactions.upsert(txId, { $set: {
                                transactionHash: result
                        } });
                }
        });
},
'click .confirmReceived' : function() {
        var txObject = {
                value: 0,
                gas: 300000,
                from: web3.eth.accounts[1]
        };
        var contractInstance = PurchaseContract.at(Session.get("contractAddress"));
        contractInstance.confirmReceived(null, txObject, function(err, result){
                if (!err) {
                        var txId = Helpers.makeId('tx', result);
                        Transactions.upsert(txId, { $set: {
                                transactionHash: result
                        } });
                }
        });
},
```

Paramon

결론 시현 동영상 / 이미지

```
'click .refundBuyer' : function() {
                var txObject = {
                        value: 0,
                        gas: 300000,
                        from: web3.eth.accounts[0]
                };
                var contractInstance = PurchaseContract.at(Session.get("contractAddress"));
                contractInstance.refundBuyer(null, txObject, function(err, result){
                        if (!err) {
                                var txId = Helpers.makeId('tx', result);
                                Transactions.upsert(txId, { $set: {
                                        transactionHash: result
                                } });
                        }
                });
},
```

Paramon

감사합니다!

×

연두

Paramon

2017 한국정보보호학회 하계학술대회

블록체인을 도입한 중고 물품 거래 서비스 설계

김보연, 이연희, 이부형, 이종혁*

*상명대학교 컴퓨터소프트웨어공학과

A Design of Used Product Transaction Services Using a Block Chain

Boyeon KIM, Yeonhui LEE, Boohyung Lee, Jong-Hyouk Lee*

*Dep. of Computer Software, Sangmyung Univ.

요 약

기존 중고 물품 거래 사이트에서 일어나는 불미스러운 사건들이 일어날 가능성을 낮추기 위하여 본 논문에서는 중고 물품 거래 사이트에 블록체인이라는 개념을 도입하였다. 본 논문에서는 블록체인의 장점인 탈 중앙성, 확장성, 투명성, 무결성 등을 이용하여 사용자끼리 신뢰성이 있고 안전한 거래가 가능한 Paramon 이라는 중고 물품 거래 서비스를 제안한다.

I. 서론

기존에 많은 사람들이 이용하는 중고 물품 거래 사이트와 애플리케이션의 시스템은 사용자들의 개인 정보가 정확하지 않고 신상 정보에 대한 보안성이 낮다. 그로 인해 계정을 조작하여 사기거래를 하려는 범죄가 만연히 나타나고 있다. 또한, 사기 거래가 발생했을 때 거래 내역을 증명하는 시스템이 없으므로 사용자들의 신뢰가 보장되지 않는 문제점이 있다. 따라서 개인 정보의 안전이 보장되고 거래 내역의 무결성이 증명되는 시스템이 필요하다.

본 논문에서 제안하는 Paramon은 중고 물품 거래에서 사용자들의 거래내역을 투명하게 공개하여 사용자 간 신뢰성이 높고, 거래 완료 후 거래 내역 수정이 불가능하게 만들어 거래 내역 데이터의 보안성을 높인 중고 물품 거래 서비스이다.

Paramon은 블록체인을 사용하여 거래 내역을 공공장부에 저장함으로써 그 내역을 투명하게 관리할 수 있고 데이터의 무결성도 보장할 수 있다.

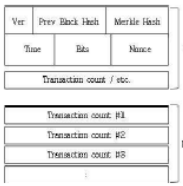
본 논문에서는 블록체인 플랫폼 중 이더리움을 이용하여 데이터의 무결성과 사용자들 간의 신뢰성을 높인 중고 물품 거래 서비스 Paramon을 설계한 내용을 다룬다.

II. 관련연구

2.1 Blockchain(블록체인)

블록체인 [1]은 2008년 사토시 나카모토가 처음 제안한 것으로, 거래 내역이 담긴 여러 개의 블록들을 서로 연결하여 블록 안에 저장된 데이터의 무결성을 보장하는 기술이다. 그의 논문에서 디지털 통화를 안전하게 주고받으면서 선한 이용자를 지키고 부적절하게 사용하려는 이용자를 막기 위한 방법으로 블록체인을 사용하였다.

블록의 구조는 [그림2-1]와 같다. 블록은 블록 헤더와 블록 바디로 이루어져 있는데, 블록 헤더 부분은 난스 값, 비트, 타임스탬프 등이 들어가 있으며, 이전 블록의 해시 값을 가지고 있기 때문에 블록들이 서로 연결되어 블록 안의 데이터를 보호한다. 블록 바디는 사용자들의 거래 내역들로 이루어져 있다.



[그림 2-1] 블록의 구조

블록체인은 3가지로 분류할 수 있는데 누구나 블록체인 네트워크에 참여할 수 있는 퍼블릭 블록체인과 하나의 기관에서 독자적으로 사용하는 프라이빗 블록체인, 여러 기관들이 컨소시엄을 이뤄 구성하는 컨소시엄 블록체인으로 분류 할 수 있다.

P2P 방식의 네트워크를 사용하며 블록체인을 구성하는 노드에는 풀 노드와 라이트 노드가 있다. 풀 노드는 로컬에 모든 거래 내역을 가지고 있으므로 거래 사실을 확인하고자 할 때 혼자서 모든 거래 내역을 찾을 수 있고, 라이트 노드는 이웃 노드들과의 통신으로 자신이 찾고자 하는 거래 내역을 찾는다.

2.2 Ethereum(이더리움)

Ethereum(이더리움) [2]은 러시아 이민자 출신 캐나다인 비탈리크 부테린이 2014년에 제안한 애플리케이션 개발을 위한 블록체인 플랫폼이다. 개발자들의 편의성을 위해서 C++, Java, Python, GO언어로 구현된 다양한 API를 제공한다. 전 세계 수많은 사용자들이 보유하고 있는 컴퓨팅 자원을 활용해 분산 네트워크를 구성하고, 이 플랫폼을 이용하여 개발 초기에 SNS, 이메일, 전자투표 등 다양한 정보를 기록하는 시스템을 창안하였다.

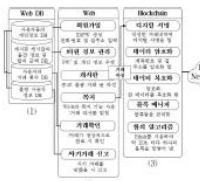
이더리움은 콘트랙트(Contracts)라고 하는 일종의 프로그램을 사용자가 작성하고, 블록체인에서 일정한 조건을 만족하면 동작하도록 코드 형태로 거래 내역을 작성할 수 있기 때문에 '스마트 콘트랙트 플랫폼(Smart Contracts Platform)' 이라고 불리기도 한다. 상대방의 확인 및 상호신뢰가 어려운 디지털 환경에서 파기할 수 없는 강력한 계약을 만들어 낼 수 있다는 점이 장점이다.

2.3 디지털 서명

디지털 서명 [3]이란 암호화 알고리즘을 통해 컴퓨터상에서 만들어진 문서나 전자 메일, 메시지 및 매크로와 같은 디지털 정보에 해당 정보를 만든 사람의 신원을 인증 및 보증하는데 사용된다.

디지털 서명은 사용자 인증과 무결성, 부인방지 기능이 있다. 사용자 인증의 경우 사용의 신원을 보증해주는 기능이며, 디지털 서명의 무결성은 서명 이후 문서에 변동이 없다는 것을 보증하는 것을 말한다. 마지막으로 부인 방지 기능은 서명자가 해당 문서에 서명한 후 서명한 사실을 부인하지 못하게 하는 기능이다. 제안하는 시스템에서는 블록체인에 데이터를 등록할 때 디지털 서명을 통해 사용자의 신원을 보증하고 거래를 부인하는 것을 방지하기 위해 사용한다.

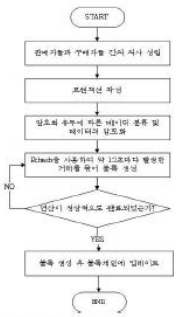
3.2 시스템 구성도



[그림 3-3] 시스템 구성도

시스템 구성도는 [그림 3-3]와 같다. (1) Web 데이터베이스에는 회원가입에 필요한 사용자들의 기본적인 정보와 게시판 기능에 쓰이는 사용자들의 ID, 상품 정보 및 합의한 거래 가격이 들어간다. 그리고 거래될수 데이터정보를 저장하는데 이는 사용자가 여러 번 안전한 거래를 완료했을 경우, 해당 사용자는 그간의 안전한 거래 횟수 데이터로 신뢰할 수 있는 사용자임을 알려줌으로써 다른 보통의 사용자들 보다 좀 더 믿을 수 있는 사용자임을 나타내기 위해서 이다. 사기 거래 신고가 들어온 사용자의 ID는 신고가 들어온 즉시 다른 사람들과 거래를 하지 못하게 공지나 제재를 하기 위해 사기 신고가 들어온 사용자들의 ID를 저장한다. (2) 웹 사이트의 기능으로는 크게 회원가입, 게시판, 사기거래신고, 거래확인 기능이 있다. 회원가입 기능에서는 ID/PW, 전화번호, 집 주소와 같이 개인정보를 입력하여 본 시스템의 사용자가 된다. 게시판 기능에서는 판매자가 상품을 게시하고 구매자는 원하는 상품에 대해 쪽지 기능으로 가격을 제시해 판매자에게 구매를 요청한다. 판매자와 구매자는 상품의 거래 가격 및 거래 순서를 합의 한다. 이 과정 후 사용자들은 거래를 진행하게 되는데 이때, 사기를 당해 사기거래 신고를 하게 되면 그간 거래 내역을 조회하여 사기를 당했음을 쉽게 입증할 수 있다. (3) 합의 완료 후 대다수 사용자에게 공개(사용자들의 ID, 상품의 기본 정보) / 공개 불가(판매자의 계좌 번호, 구매자의 집 주소) 라는 기준으로 분류한 뒤, 공개할 수 없는 데이터들은 데이터 암호화 시스템으로 암호화 되어 블록체인으로 보내진다. 작성한 거래내역에 디지털 서명을 하고, 데이터 암호화 시스템에서는 넘어온 데이터들을 암호화하고, 필요에 따라 복호화가 필요한 데이터들은 데이터 복호화 시스템에서 복호화 시킨다. 마지막으로 거래에 사용된 데이터들은 합의 알고리즘을 거쳐 P2P Network에 등록된다.

3.3 프로그램 순서도

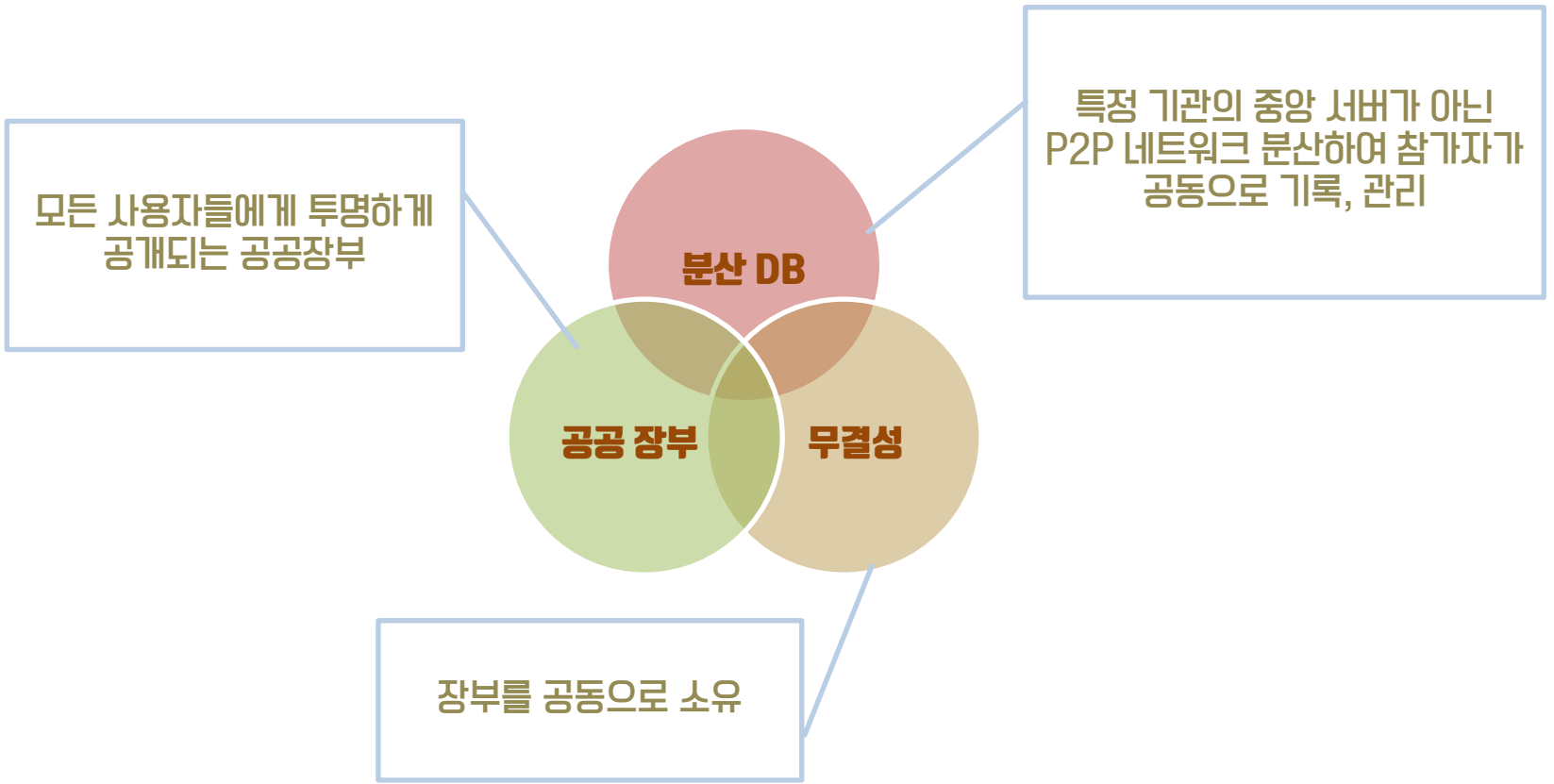


[그림 3-4] 프로그램 순서도

프로그램의 순서도는 [그림 3-4]와 같다. 판매자와 거래자는 Web을 통해 상품을 사거나 판

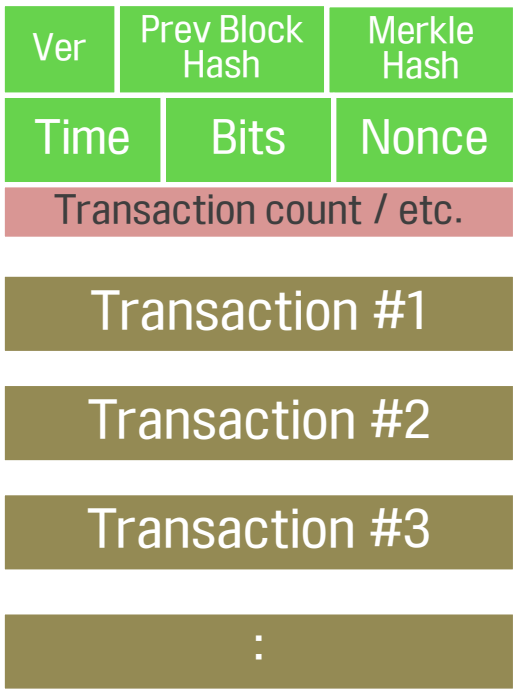
Paramon

참고자료 _ 블록체인



Paramon

참고자료 _ 블록체인



<Header>

- 현재 프로그램의 버전
  : 소프트웨어 / 프로토콜 업그레이드를 추적하는 버전 번호
- 이전 블록을 해싱한 해시 값
  : 체인에서 이전 (상위) 블록의 해시 값
- 머클 해시
  : 이 블록의 트랜잭션의 Merkle 트리의 루트 해시
- 현재 블록의 타임 스탬프
  : 이 블록의 대략적인 생성 시간
- 난이도 Bits
  : 이 블록에 대한 작업 증명 알고리즘 난이도 목표
- Nonce
  : 작업 증명 알고리즘에 사용되는 카운터

<Body>

- 트랜잭션
  : 거래 정보