자체 블록체인 네트워크 구축을 위한 블록체인 프레임워크 구현

An Implementation of Flexible Blockchain Framework (FBF) for Construct Own Blockchain Network

2018.06.21

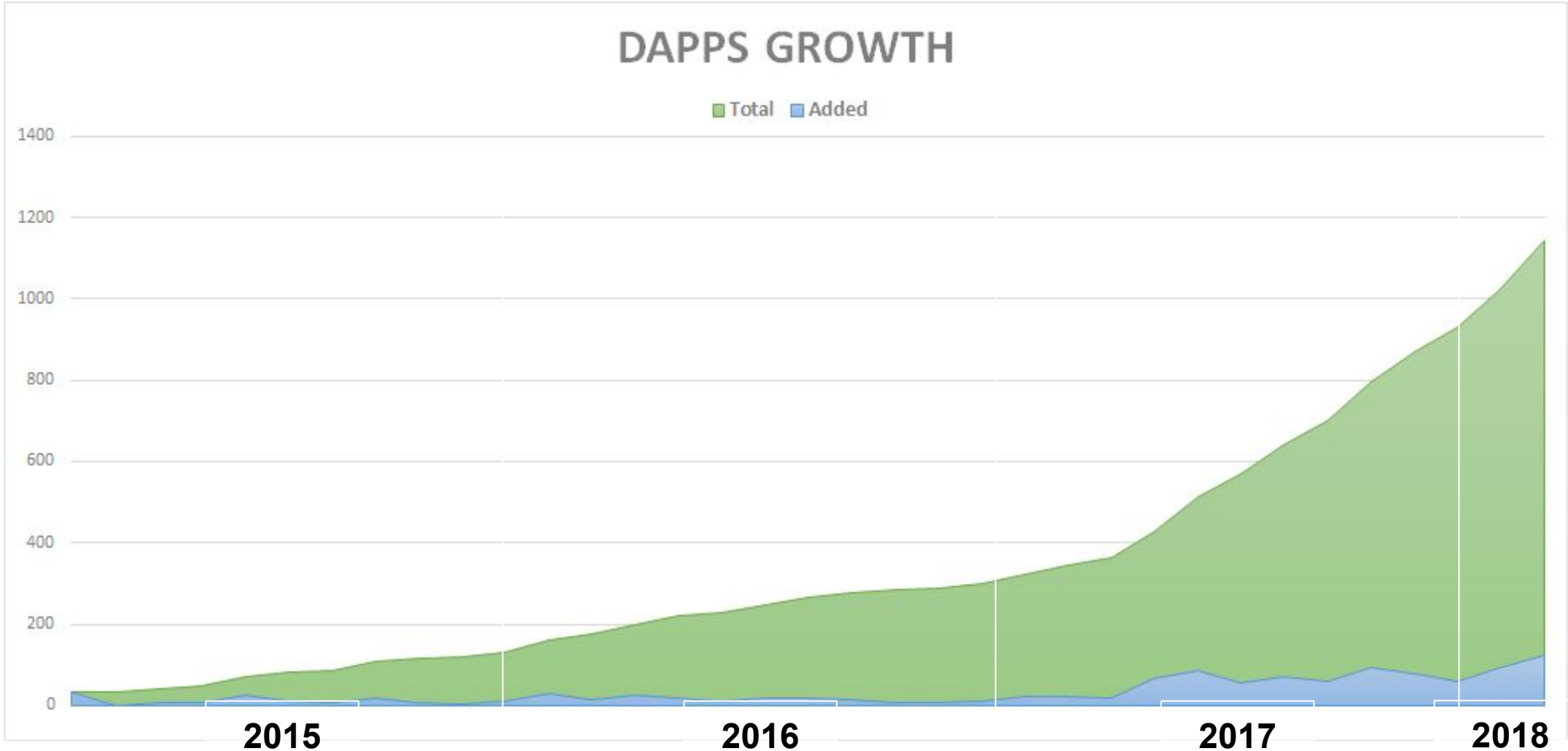
KCC 2018

발표자 : 하현수

INDEX

1. 서론

Ecosystem


DAPPS GROWTH
Total
Added
1400
1200
1000
800
600
400
200
0
2015
2016
2017
2018

동기

기존 블록체인 코어의 문제점

Too Heavy

Bitcoin Core : Approximately *100K* Lines
Bytecoin Previous Version : *600K* Lines
개발 및 수정하기에 너무 코드가 무겁다.

Too Complex

블록체인 코어는 오랜 기간 유지보수를 거치며 다양한 기능들이 추가됨에
따라서 매우 복잡하고 난해해진다.
따라서, 본 논문에서는 불필요한 기능을 배제하여 일종의 스타트킷 역할을
하는 블록체인 프레임워크를 만들고자 한다.

Inflexible

대부분의 블록체인 코어는 비유동적인 아키텍쳐에 의해서
합의알고리즘이나 프로토콜을 변경하는것이 매우 어렵고 한정적이다.

목적

개선된 블록체인 코어 프레임워크

모듈화 및 리팩토링

- 기존 블록체인 코어 프로젝트의 코드는 모듈화가 진행되지 않아 소스코드를 분석하는데 많은 어려움이 있다.
- 위와 같은 문제를 분산되어 있는 블록체인의 속성값들을 하나의 설정 파일에 통합하는 등의 모듈화를 진행함으로서 개선한다.

코드 경량화

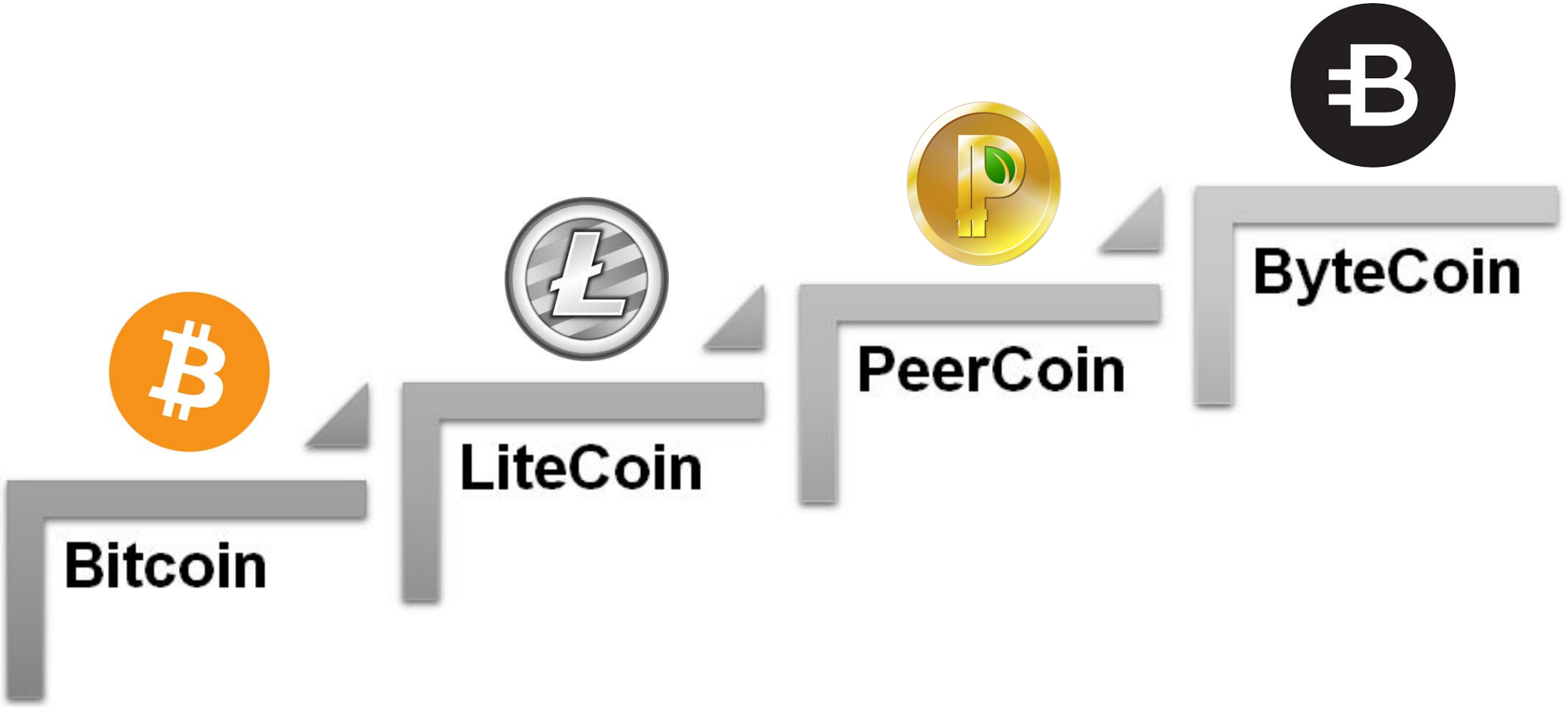
- 기존 블록체인의 소스코드는 서비스 운영과정에서 추가된 부가적인 기능이 소스 코드 내에 포함되어 있어 코드 분석을 어렵게 한다.
- 블록체인의 기본적인 기능, 즉 RPC, P2P 서버 및 컨센서스 알고리즘의 대한 소스코드만을 가지고 있는 일종의 블록체인 코어 “스타트 킷”을 개발한다.

자체 블록체인 네트워크

- 기존 블록체인을 포크하여 사용할 경우 SeedNode의 주소값이 고정되어 있어 자체 블록체인 네트워크로 분리 하기 어렵다.
- SeedNode의 주소값의 수정을 가능하게 하여 블록체인 네트워크를 보다 쉽게 분리시킬 수 있도록 한다.

2. 관련 연구

Blockchain Core 발전 흐름



Bitcoin

비트코인은 최초의 블록체인 기반 암호화폐이며 오픈소스로 공개되어 있다. 하지만 비트코인의 소스코드에서 상수와 변수의 값들을 변경하는 것은 아주 어렵다. 해싱 값들이 단순 하드코딩되어 있거나, **Assert**문으로 고정되어 있기 때문이다.

또한 비트코인의 소스코드는 모듈화가 전혀 되어있지 않기 때문에 소스코드의 의존성 분석에 시간이 오래 소요된다.

Litecoin

라이트 코인은 비트코인 기반의 암호화폐로 해싱함수를 **SHA-256** 에서 **Scrypt**로 바꾸고, 블록사이즈와 최대 발행 수량을 늘린 블록체인 프로젝트이다. 블록체인 속성값 수정을 위해 **Bitcoin**의 **Assert**문장을 제거하여 비교적 플렉서블하게 수정 할 수 있게 되었다는 의의가 있다.

하지만 모듈화가 되어 있지 않고 컨센서스 방식과 같은 큰 틀에서의 변화를 위해서는 의존성을 검사하며 수정해야 한다는 한계점을 벗어나지 못한다.

Peercoin

피어 코인은 비트코인을 모체로 하며 기존 비트코인의 합의 알고리즘인 작업증명(POW)을 지분증명(POS) 방식으로 변경한 프로젝트이다. 수 많은 지분증명 방식의 블록체인들의 모체가 되고 있으며 비트코인 기반의 프로젝트 중 최초로 가장 많은 부분에서 변화가 나타난 프로젝트이다.

비트코인을 Fork한 후 합의 알고리즘을 Dependency 에러를 직접 해결하며 새롭게 구현하였지만 기존 비트코인의 코드와 Peercoin의 새로운 코드가 난잡하게 섞여있어 블록체인 구축에 사용하기에는 많은 어려움이 있다.

Bytecoin

비트코인 엔진이 아닌 자체 블록체인 코어 엔진을 사용한 최초의 익명성 코인이다. **Cryptonote**라고 하는 자체 블록체인 코어 기술을 통해 만들어졌으며, 비교적 쉽게 블록체인 속성값을 수정할 수있다.

하지만 소스코드 길이가 **Bitcoin**보다 훨씬 방대하고 모듈화가 진행되지 않아 소스 분석이 아주 복잡하며, 블록체인의 속성값 변경은 간단하더라도 자체 블록체인의 **P2P** 네트워크를분리해내는것이 어렵다는 단점이 있다.

3. 구현 내용

FBF

Flexible Blockchain Framework

FBF(Flexible Blockchain Framework)는 기존 블록체인의 문제점을

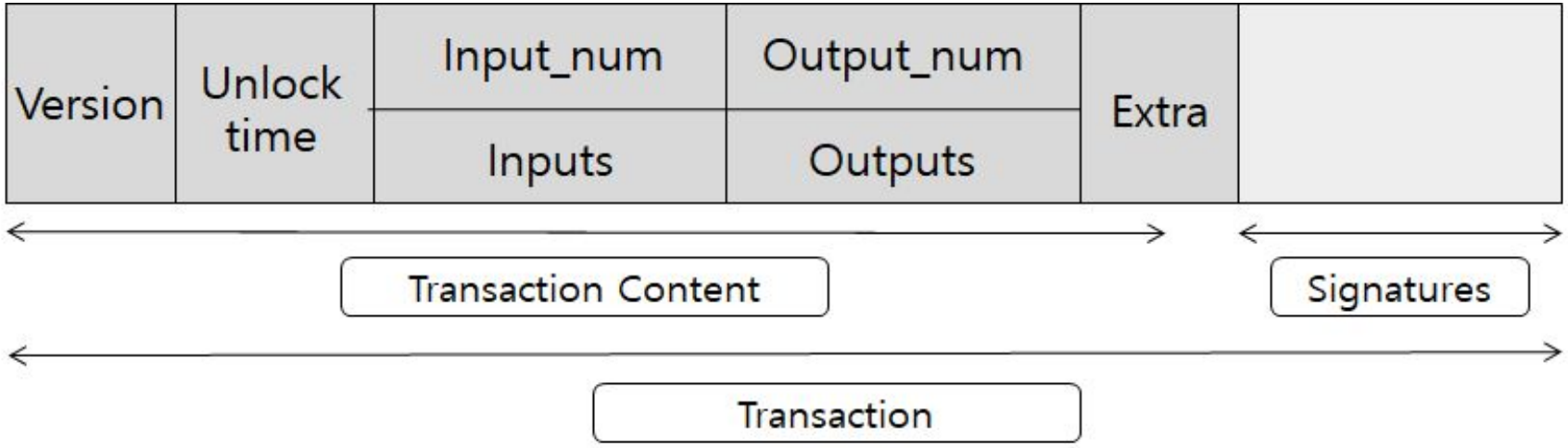
해결하기 위해 기존 블록체인 코드를 기능 단위로 모듈화하고,

블록체인 속성값들을 설정 파일 한곳으로 통합시켜 보다 효율적인 수정을 할 수

있도록 하였으며, **Seednode**의 주소값 또한 수정 가능할 수 있도록 하여

블록체인 네트워크를 보다 쉽게 분리시킬 수 있도록 하였다.

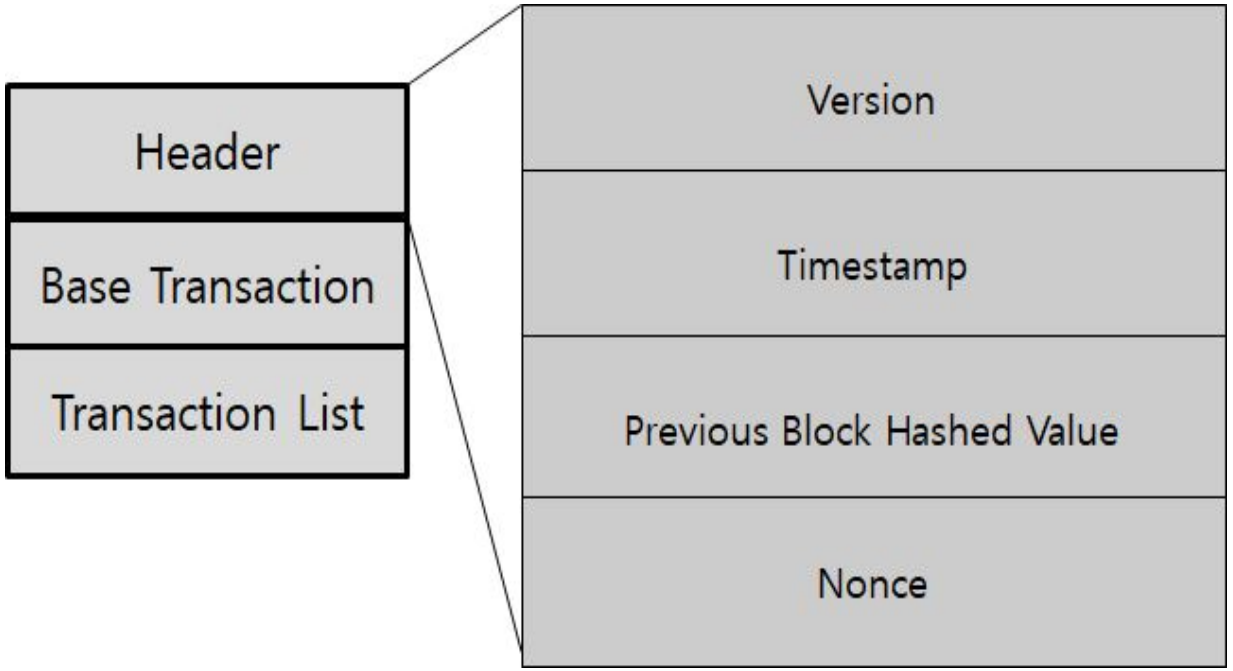
Transaction



< Transaction Structure >

비트코인의 트랜잭션 구조를 참고하여 가장 일반적으로 사용되는 구조로 설계하여 호환성을 높였다. **UnlockTime**에는 **UNIX**의 시간정보인 **Timestamp**를 기록하며, **Inputs**에는 트랜잭션 소유권에 대한 키 정보와 **Block Height**가, **Outputs**에는 전송될 코인의 수량과 송금 주소정보가 저장된다.

Block

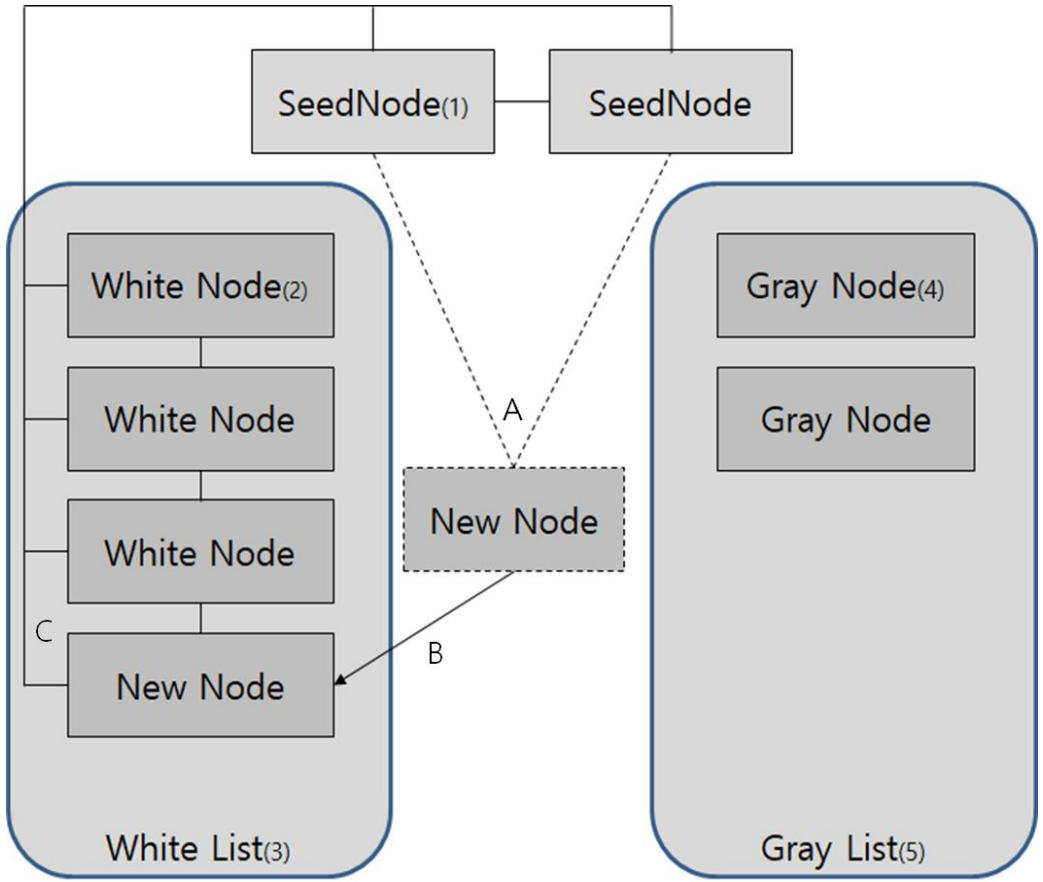


< Block Structure >

블록헤더는 **Version**과 **UNIX**기반의 **Timestamp,** 이전 블록의 해시값**,** 작업증명을 위한 **Nonce**로 구성되어 있으며 이러한 값들을 통해 위·변조를 파악할 수 있다**.**

Base Transaction은 채굴 보상을 받을 사람을 식별하기 위한 블록에 생성된 최초 트랜잭션인 블록 생성 트랜잭션 정보를 저장하고 있으며**,** **Transaction List**는 해당 블록에 포함된 트랜잭션들의 **TxHash**값이 리스트 형태로 각각 저장된다**.**

P2P Network

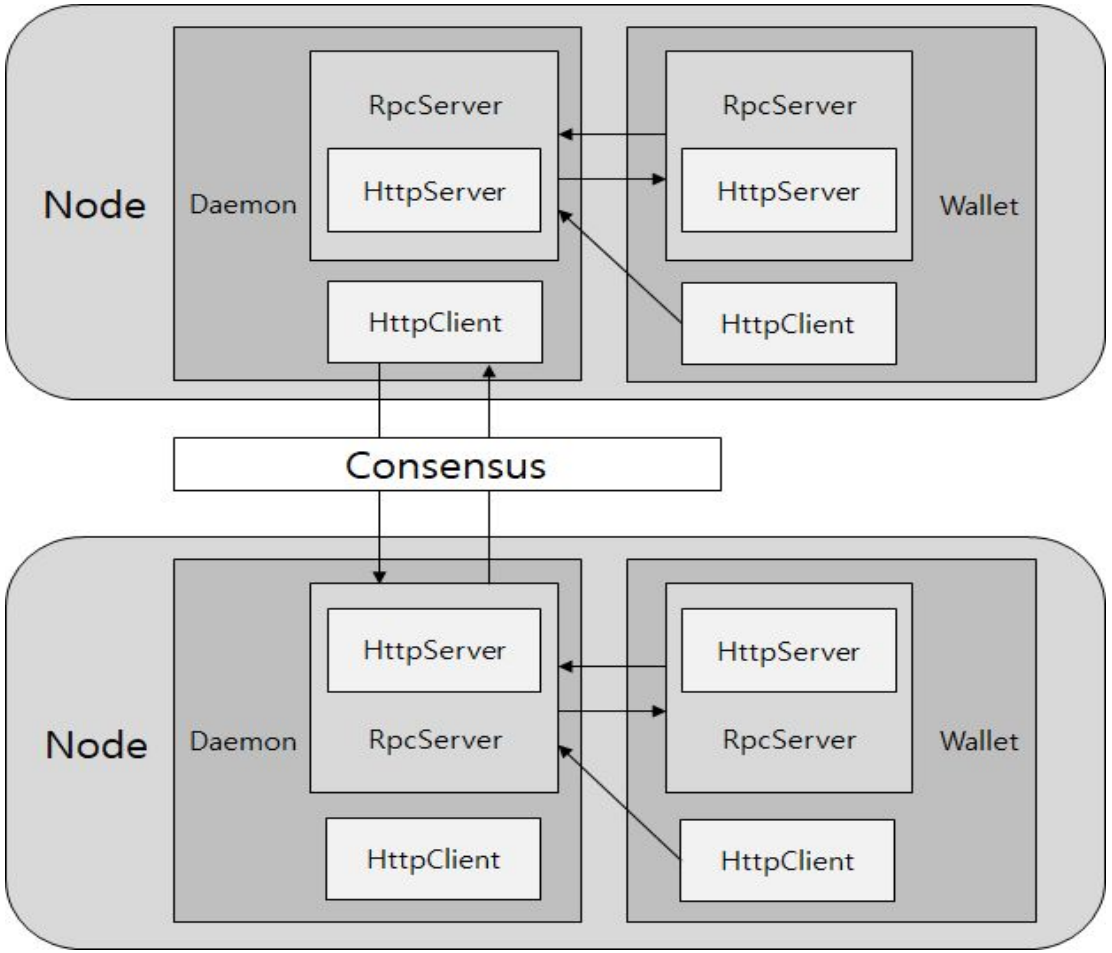


< P2P Network Structure >

P2P 네트워크는 **SeedNode**와 현재 **P2P**네트워크에 연결되어 동작중인 노드인 **White Node**와 이들의 집합인 **White List,** 연결이 종료된 노드인 **Gray Node**와 이들의 집합인 **Gray List**로 이루어져 있다.

동작 **A, B, C**는 새로운 노드가 **P2P Network**에 참여하고자 할 때 진행되는 순차적인 동작들이다.

Blockchain



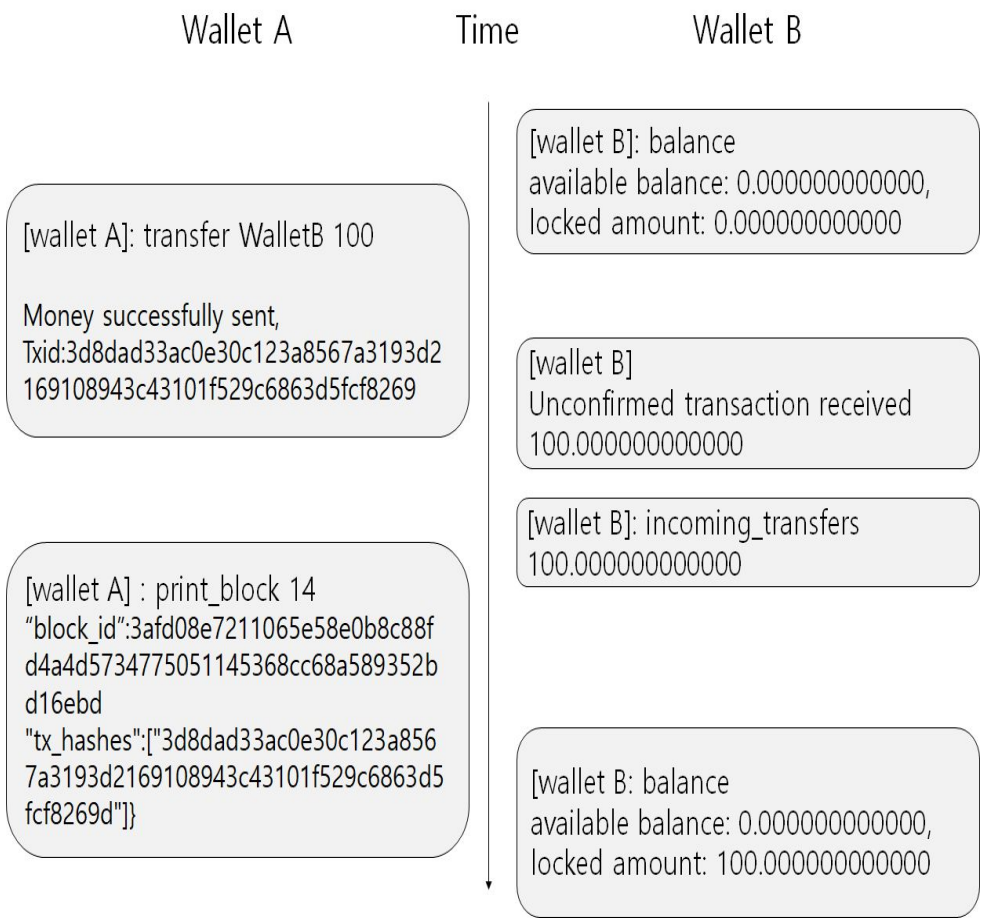
< Blockchain Network Structure >

P2P 네트워크를 구성하는 각 노드내에는 데몬과 지갑 프로세스가 동작하고 있으며 **RPC** 서버와 **HTTP** 서버를 사용하여 외부 노드 및 다른 프로세스와 통신을 진행한다.

위와 같은 통신 방식을 통해 동기화 과정 및 작업증명 **(POW)** 방식에 의한 합의를 통한 블록 생성 등의 과정을 진행하게 된다.

4. 결론 및 향후 연구

정상 구현 여부 실험



< Transaction Test >

블록체인이 가져야 할 가장 기본적인 기능인 **Coin** 송금 기능을 테스트해보기 위해, 서로 다른 컴퓨터(노드)에 **Wallet A**와 **Wallet B**를 각각 생성한 후, 사전에 미리 채굴해놓은 **Wallet A**의 코인 **100**개를 **Wallet B**로 송금하는 과정이다.
해당 트랜잭션은 **14**번째 블록에 기록되어 있음을 확인할 수 있다.

최종 **Balance**값이 성공적으로 증가한 것을 확인할 수 있다.

트랜잭션 JSON


```
'jsonrpc' : '2.0',
  'id' : 'transfer' ,
  'result' : {
      'transaction' : {
        'fee' : 10,
        'extra' : '0178a684d7d2e802c3a15854ab011bd78...',
        'timestamp' : 0,
        'blockIndex' : 13,
        'state' : 0,
        'transactionHash' : '3d8dad33ac0...',
        'unlockTime' : 0,
        'transfers' :
{
        'amount' : 100,
        'type' : 0,
        'address' : 'FUACdVN272wipNYR..."
}
        'paymentId' :,
        'isBase' : False
      }
```


<Transaction Result >

송금 실험에서 발생한 트랜잭션은 결과로 그림과 같은 **JSON**값을 반환한다**.** 즉**,** 성공적으로 트랜잭션이 생성되었고 **14**번 블록에 기록되었음을 확인할 수 있다**.**

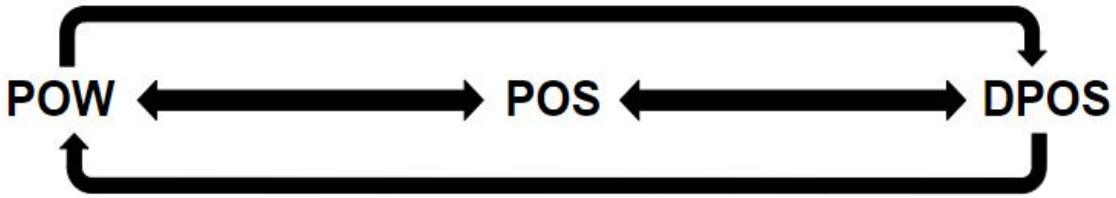
결론

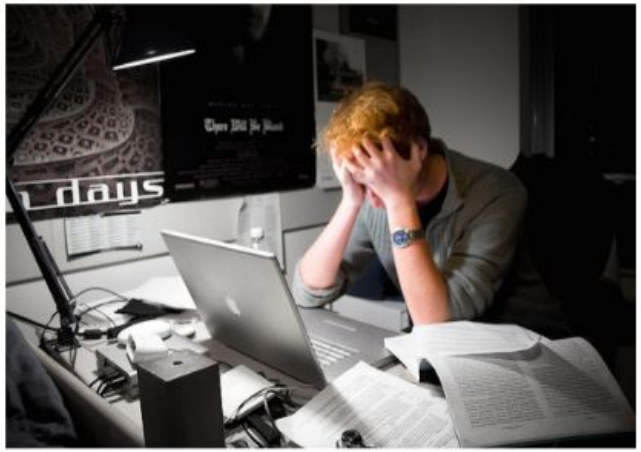
```
EMISSION_SPEED_FACTOR=18
DIFFICULTY_TARGET=120
CRYPTONOTE_DISPLAY_DECIMAL_POINT=12
MONEY_SUPPLY=18446744073709551615
GENESIS_BLOCK_REWARD=0
DEFAULT_DUST_THRESHOLD=1000000
MINIMUM_FEE=1000000
MAX_TRANSACTION_SIZE_LIMIT=100000
FBF_PUBLIC_ADDRESS_BASE58_PREFIX=86
DIFFICULTY_CUT=60
...
p2p-bind-port=29437
rpc-bind-port=29438
FBF_NAME=KCCOIN
GENESIS_COINBASE_TX_HEX=010a01ff0001ffffffffffff...
MAX_BLOCK_SIZE_INITIAL=100000
UPGRADE_HEIGHT_V2=1
UPGRADE_HEIGHT_V3=30
seed-node=35.200.161.200:29437
seed-node=35.200.147.252:29437
```

<Blockchain Config Parameter>

본 논문에서는 블록체인의 코인 총 발행량, 채굴난이도, **P2P Port, RPC Port, Seed Node**의 **IP**정보등 각종 블록체인의 **Parameter**를 담고 있는 **Config**파일을 따로 분리하고 이를 유동적으로 수정함으로써 간단히 블록체인 네트워크를 구성할 수 있는 **FBF**를 구현하였으며 실제 **Coin** 송금을 통해서 정상적으로 작동함을 확인하였다.

향후 연구 계획





<Changing Consensus Algorithm>

대다수의 블록체인 네트워크는 필요에 따라 컨센서스 알고리즘을 변경하곤 한다. 기존의 블록체인 프로젝트는 코드 의존성 문제가 심각하고 모듈화가 되어 있지 않아 컨센서스 알고리즘의 변경이 쉽지 않다. 따라서 보다 유동적인 **FBF**를 구현하기 위해서 **POW, POS, DPOS**의 다양한 컨센서스 알고리즘을 단순히 **Config** 파일의 타입값을 변경하는 것만으로 변경할 수 있도록 추가 구현할 것이다.

Q&A

감사합니다