Sunblockterminal(SBT)

Platform Architectures

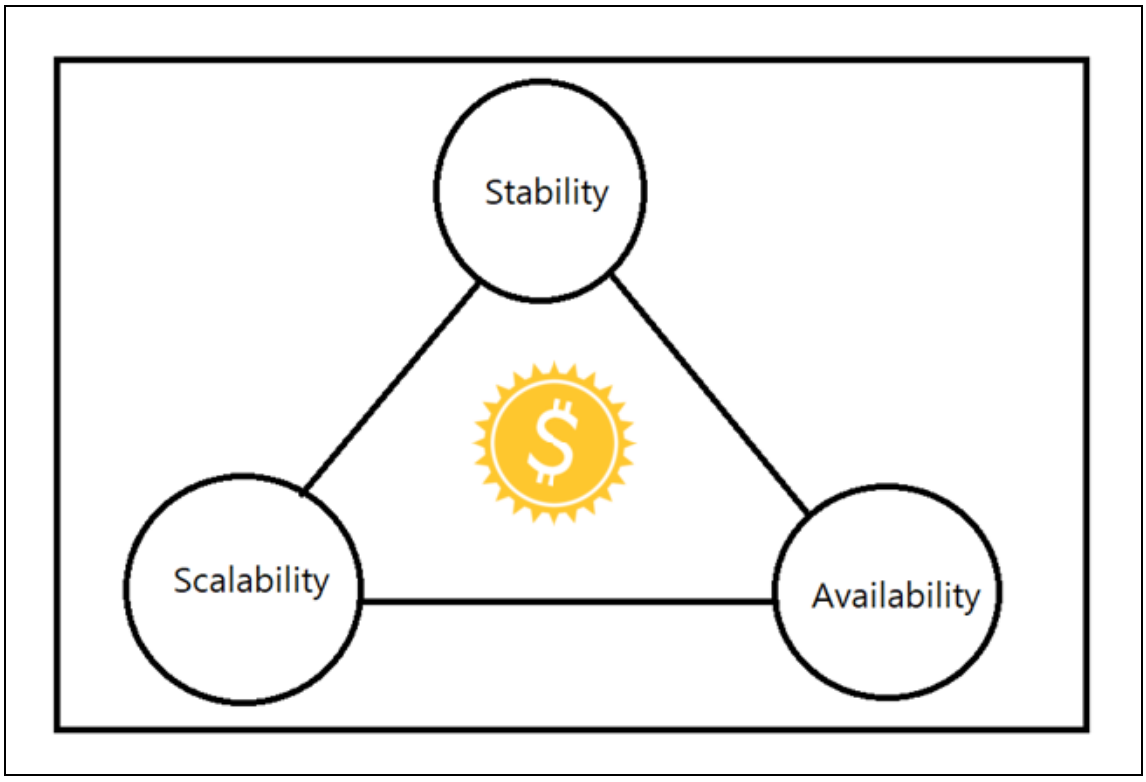
목차

Abstract

Thanks to Sunblockterminal 에 영향을 끼친이:

사토시 나가모토(비트코인의 발명자), 비탈릭 뷰테린(이더리움재단의 파운더이며 Turing-Completed 스마트컨트랙트의 개발자), 댄 라리머(비트쉐어/스팀잇/이오스의 파운더이자 개발자), 재권(이더리움재단의 개발자였으며, 코스모스의 파운더이며, 블록체인간 통신 프로토콜 개발자)에게 감사한다.,
아인슈타인(상대성이론의 발명자이자 EPR 패러독스의 제안자), 보어(양자역학의 대가), 벨(EPR 패러독스를 반박하는 벨이론의 창시자), 지생(벨이론에 입각한 양자통신을 실험하고 양자난수를 개발한 과학자)에게 감사한다.
니시모리 히데토시(양자어닐링 이론의 창시자), 디-웨이브시스템(상업용 양자컴퓨터개발사)에게 감사한다.

Sunblockterminal 은 High Availability, More Scalability, Best Stability 를 제공하는 블록체인기반 Payment Platform 이 되기를 희망한다.



1 Introduction

Sunblockterminal 의 첫 버전은 가장 큰 블록체인 플랫폼인 이더리움의 네트워크상에 론칭된다. 이더리움은 자체적인 이더리움 가상 머신(Ethereum Virtual Machin-EVM)을 사용하여 블록체인에서 구동되는 탈중앙화 응용프로그램(Decentralized App-DApp)의 개발을 용이하게 하였다. DApp 은 모바일에서도 실행 가능하며 기존의 메이저 블록체인 생태계와 완벽히 호환 가능하다. 이더리움은 수정된 비트코인 인프라스트럭처와 이더리움 가상 머신이 결합하여 신뢰할 수 있는 블록체인위에 사적계약인 스마트컨트랙트를 활용할 수 있는 생태계를 제공한다.

Sunblockterminal 의 두 번째 버전은 EVM 을 활용한 자체적인 메인넷이다. 자체적인 메인넷과 EVM 을 사용하여 EVM 과 솔리디티 표준을 준수하는 독자적인 블럭체인 생태계를 꾸려나갈 것이며, DApp Platform 에 최적화된 Alternative 가 될 것이다.

Sunblockterminal 은 Sunblockterminal 메인넷을 선택하여 첫번째 대규모 서비스인 블럭체인 기반 Payment Platform 서비스를 시작한다.

*Sunblockterminal 은 DPoS 방식의 컨센서스 알고리즘을 채택하여, 에너지 소모적인 PoW 대비 Exchange,SNS,CryptoGame 등에 최적화된 블럭체인이다. Plasma, Sharding, State Channel 과 같은 확장 솔류션과의 연결을 통해서, Mainchain 과 Sidechain 을 동시에 사용하여 Scalability 문제를 함께 해결한다.

Sunblockterminal 은 사용자의 정보 보호를 위해 Subscribe 에 기반한 이동통신 환경에서의 인증방식을 활용합니다.

전체적인 네트워크 구조에 대한 자세한 내용은 8 장에서 다룰 것이며, 새로운 개념들을 도입하려는 시도에 대해서 설명할 것입니다.

1.1 DApp 을 만들 수 있는 생태계

Decentralized Application(DApp)은 블록체인에 기록하는 실행할 수 있는 프로그램이다. 실행할 수 있는 특정 조건문을 블록체인에 기록을 해 둔 상태에서 어떤 정해진 조건이 되었을 때 자동으로 실행되는 프로그램이다. 예를 들어 다음과 같은 조건을 블록체인이 올리는 것이 가능하다.

"내 주소 A 에 코인을 보내시오. 오늘 12 시까지 가장 많은 코인을 보낸 사람에게는 내가 가진 물건을 주겠소. 그 보다 적은 코인을 보낸 사람은 자동으로 환불하겠소."

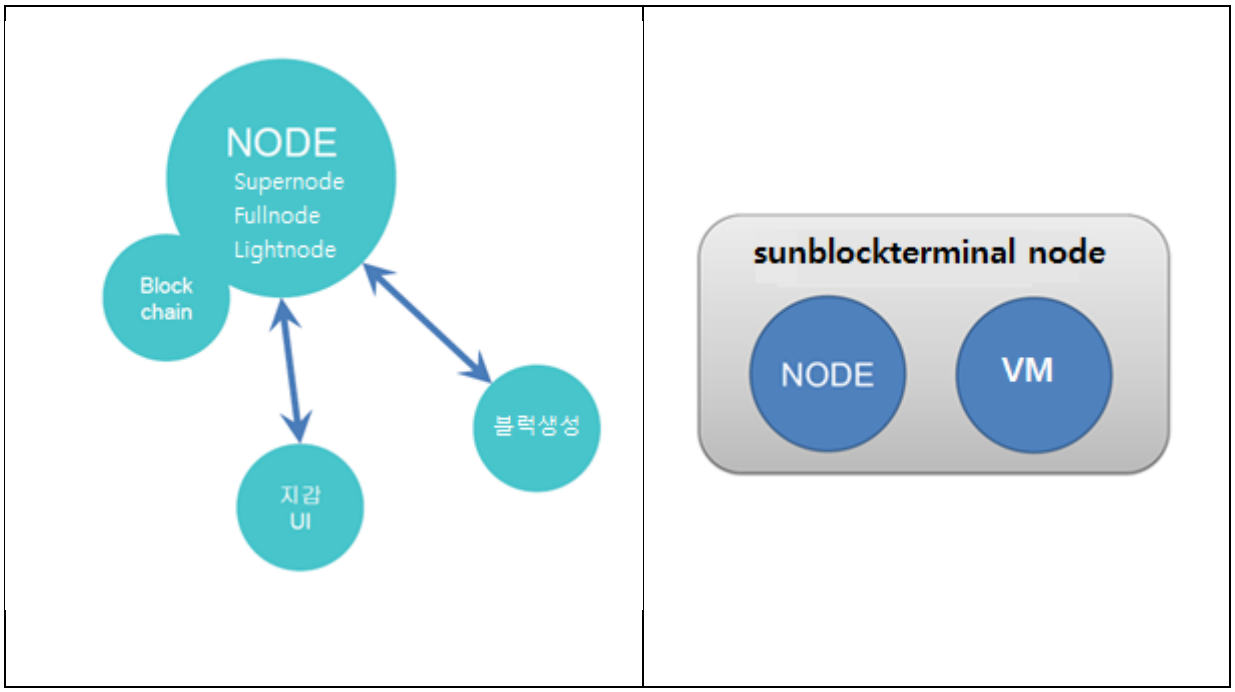
이런 조건문을 블록체인에 올리면 자동으로 가장 많이 보낸 사람을 선택하고 나머지는 모두 환불을 실행할 수 있다. 이런 식으로 작동하는 것이 DApp 입니다. 어느 누구의 통제도 받지 않고 모두가 볼 수 있는 명령문을 블록체인에 올려 두고 모두가 확인을 한 상태에서 자동으로 작동되므로 탈중앙화된 응용프로그램 (Decentralized Application-DApp)이라고 한다. 그리고 이런 계약을 스마트컨트랙트(Smart Contract)라고 한다.

Smart Contract 를 활용하면 계약당사자간 용역, 물류 등의 대금 지급등을 상세히 설정해 계약당사자간의 동의를 통해 체인위에 기록하고, 약속된 가치의 Sunblockterminal 을 계약에 귀속시키는 방법으로 계약의 집행 및 해지가 가능하다.

SBT 는 EVM 을 활용해서 스마트컨트랙트를 구현해 제 3 자의 참여없이 계약이행을 자동화하는 것이 가능하다. 누구나 SBT 에 DApp 을 탑재하여 전세계에 퍼져 있는 SBT 노드를 통해 제약없이 DApp 을 이용할 수 있다.

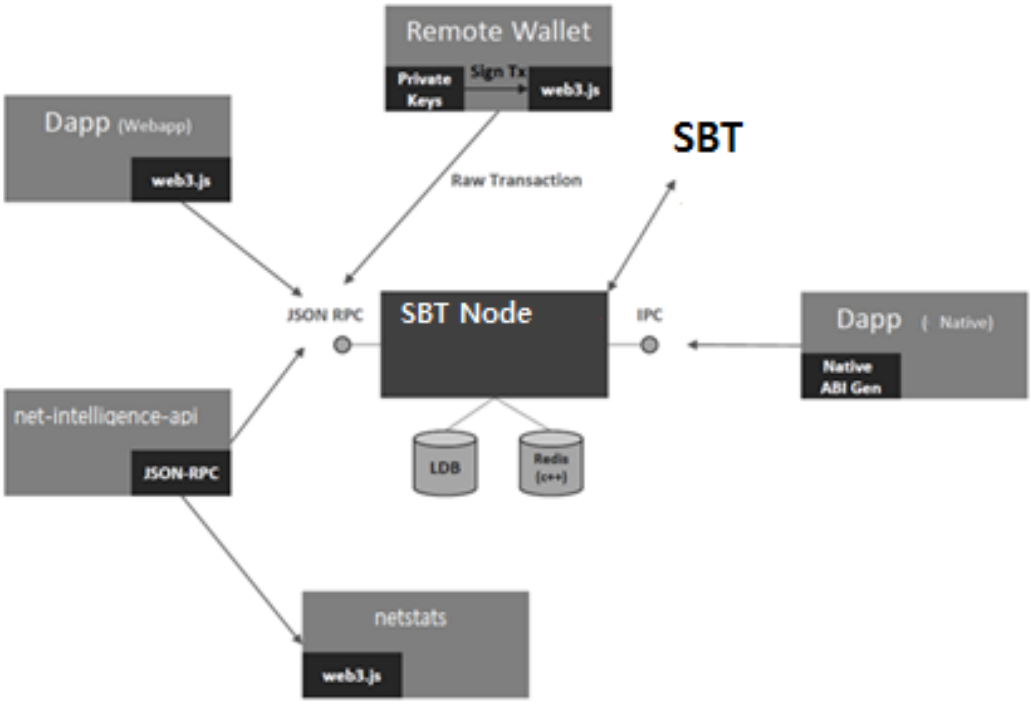
Sunblockterminal 에서는 Sunblockterminal 전용 브라우저와 전용 Metamask, My SBT Wallet(크롬 확장 프로그램)을 통해 DApp 을 이용할 수 있다.

1.1.1 Sunblockterminal.node 기본구성



- Node : Sunblockterminal 에 연결된 모든 PC 또는 서버
- Supernode : 빠른 블록생성 및 전파를 위해 광대역 네트워크로 연결된 노드. 블록생성 및 블록 전파의 역할을 한다.
- Fullnode : 모든 블록체인을 보유한 노드로 블록 검증, 동기화, 전파, 서비스 제공의 역할을 하며 블록생성에는 참여하지 않는다.
- Lightnode : 블록체인을 보유하지 않고 블록헤더와 Depth(6)의 머클정보만을 가지는 노드로서 필요할 경우 Fullnode 에 연결해서 정보를 받아오는 노드이다. 일반적으로는 서비스를 하는 홈페이지, 하드웨어지갑, 모바일 지갑 등이 Lightnode 의 역할을 담당한다.
- Lightnode 와 Fullnode 에는 Sunblockterminal node 데몬이 구동되며 이 데몬위에 VM 이 구동된다. Lightnode 에는 데몬과 VM 이 구동되지 않는다.

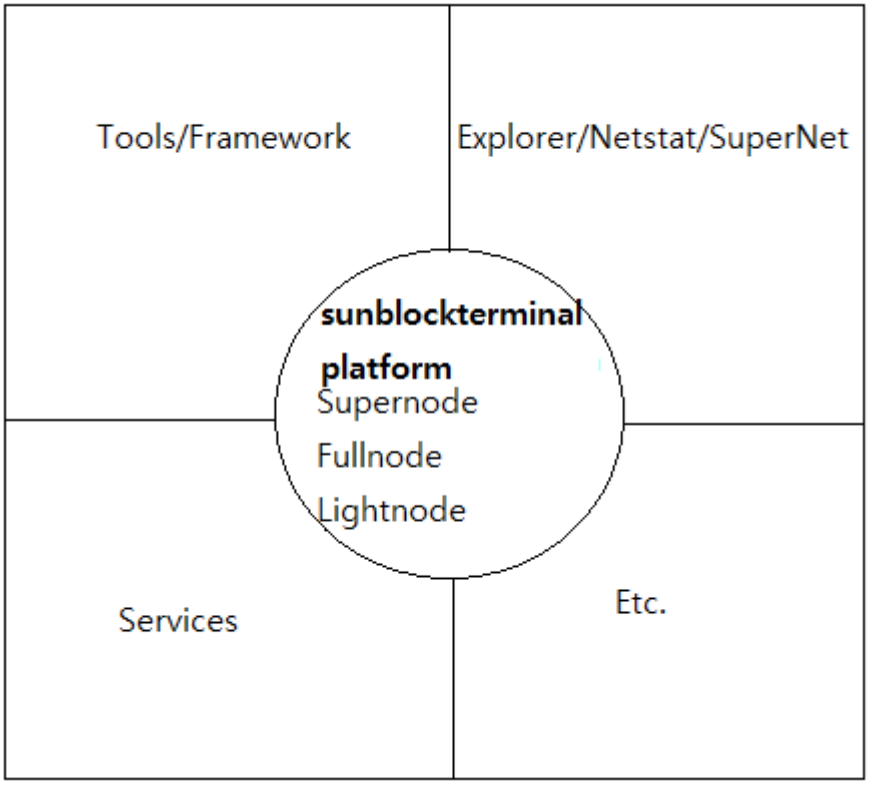
1.1.2 DApp Architecture



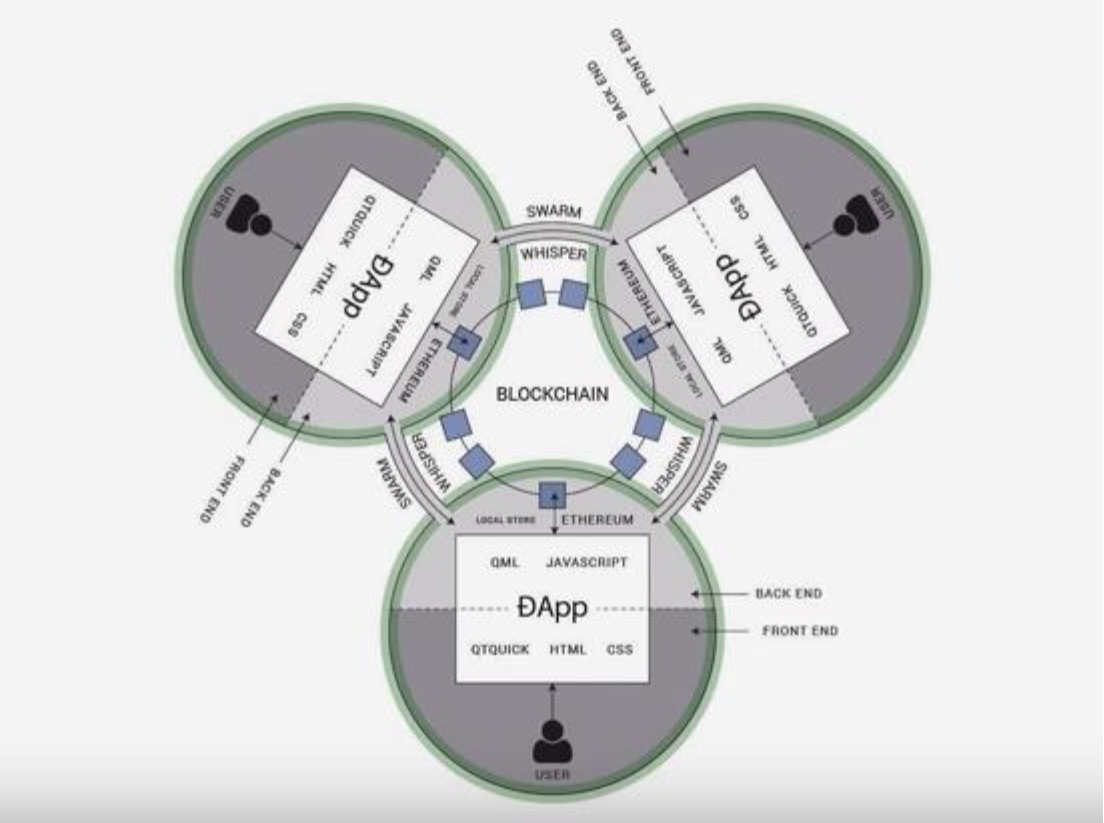
1.1.3 Key Technologies

Smart Contract	Blockchain 에 Turing Complete Code 를 배포 Transaction 을 통해 계약을 실행하면 모든 노드가 이를 실행 블럭체인 기술응용영역확장에 큰 영향
JavaScript API/JSON RPC	JSON RPC API 스펙 표준화 eth, miner, personal, db, admin 등 web3.js 를 통해 JavaScript API 제공
enode ID	Public Key 와 IP.Port 의 조합 Peer 간의 Trust 를 위해 Public Key 교환에 사용
Account/State	Externally Owned Account(EOA) Contract Account
RLP(Recursive Length Prefix)	가변의 데이터를 표현하기 위한 데이터 구조 복잡한 다차원 배열을 1 차원으로 표현 RPLx 를 통해 인증/암호화된 통신 객체를 Serialize 하기위한 프로토콜
Whisper/Swarm	Whisper - 피어간 필터링 된 빠른 메시징(shh) Swarm - 분산된 바이너리 리소스를 식별하고 교환(bzz)

1.1.4 Eco Systems



1.1.5 DApp 개발 플랫폼



1.2 토큰을 만들고 발행할 수 있는 시스템

컬러드 코인 접근이라는 개념은 비트코인의 블록체인 프로토콜을 사용하는 여러 프로토콜이 구현되었던 2013 년 무렵에 처음 생겼다.

*컬러드 코인 접근 : 비트코인의 블록체인 프로토콜을 통해 현물자산을 디지털형태로 표현하는, 일종의 자산 발행 레이어를 의미한다.

참고자료 : https://brunch.co.kr/@jeffpaik/13

그 외에도 커스텀 블록체인 토큰 플랫폼을 처음부터 개발하려는 시도가 여러 차례 이루어졌으며 그 중 유명한 것이 NXT 토큰입니다.

NXT 에서 개발한 방법은 전송내역(transaction)에 첨부(attachment)를 추가하는 방법으로 첨부에 기록된 내용으로 토큰 생성 및 전송을 구현한다. 이 방법은 qtum 과 zencash 에 비슷한 방법으로 적용이 된다. 이 방법은 기존에 개발된 Bitcoin transaction 구조를 무너뜨리지 않고 정보를 전달할 수 있는 이점이 있지만, 새로운 구조의 transaction 을 추가해야 하는 경우에는 모든 블록체인 프로그램이 동시에 업데이트가 되어야하는 문제가 생긴다. 만일 새로운 트랜잭션을 따르지 않는 프로그램이 계속 유지가 되는 경우 기존 블록체인과 새로운 블록체인이 분리되는 포크가 발생하게 된다.

SBT 는 확장(Extension)으로 설치되는 플러그인을 지원하고, 그것을 통해 새로운 방식의 transaction 구조가 제안되더라도 이를 도입할 수 있는 솔루션을 통해 구조가 고정됨으로써 발생하는 문제를 해결할 수 있다. 관련 플러그인이 설치되지 않은 클라이언트도 이러한 커스텀 Transaction 을 전달할 수 있으며, 이 솔루션은 3rd Party 개발자들이 새로운 transaction 을 도입하고 DApp 스토어와 같은 생태계를 만들 수 있다.

Sunblockterminal 기본 코어 수준에서는 아래와 같은 기본적인 transaction 유형을 지원합니다.

- 커스텀 토큰 생성, 삭제 및 이체

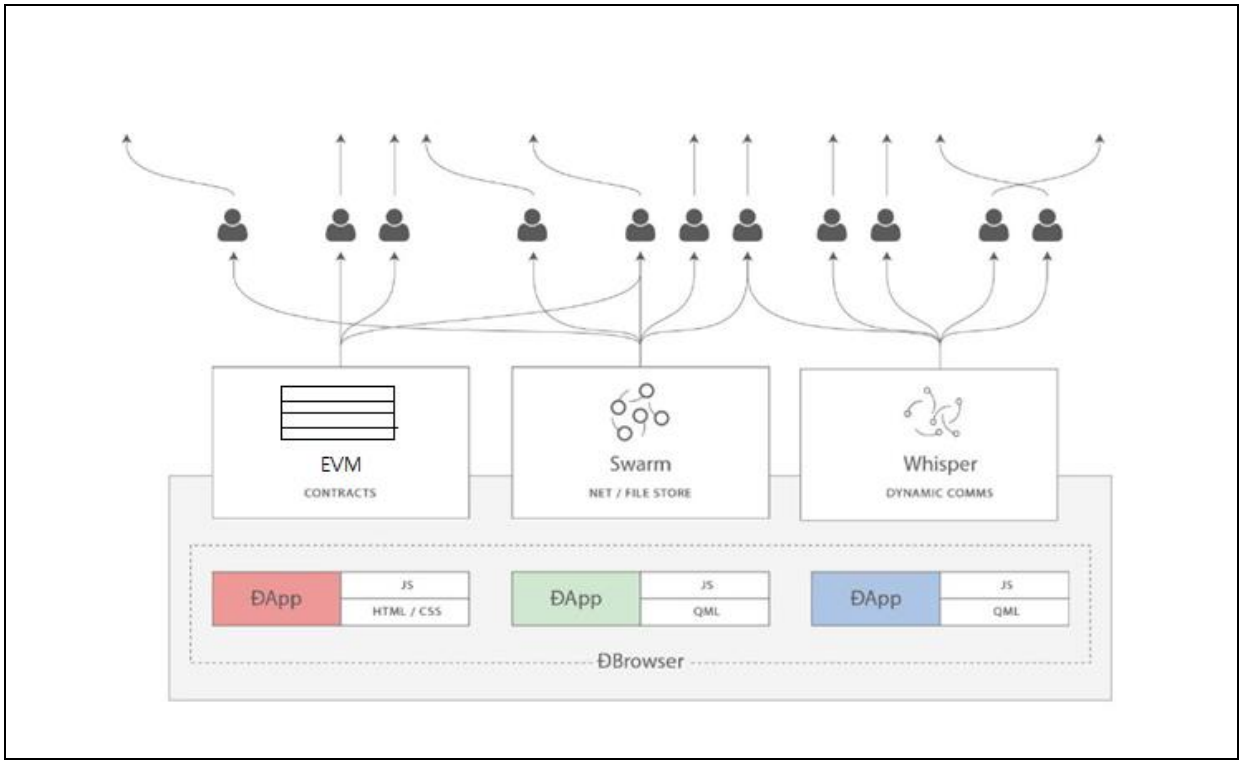
- 매수/매도(Bid and Ask) 네트워크 거래가 서로 매칭되는 주문 매칭엔진으로 구현된 분산화 토큰거래
- 익명성 기능

SBT 는 탈중앙화된 블록체인 기반의 transaction 을 통해 커스텀 토큰끼리의 거래를 제공함으로써 자산에서 자산으로 거래하는 한 발 앞선 거래방식을 지원한다.

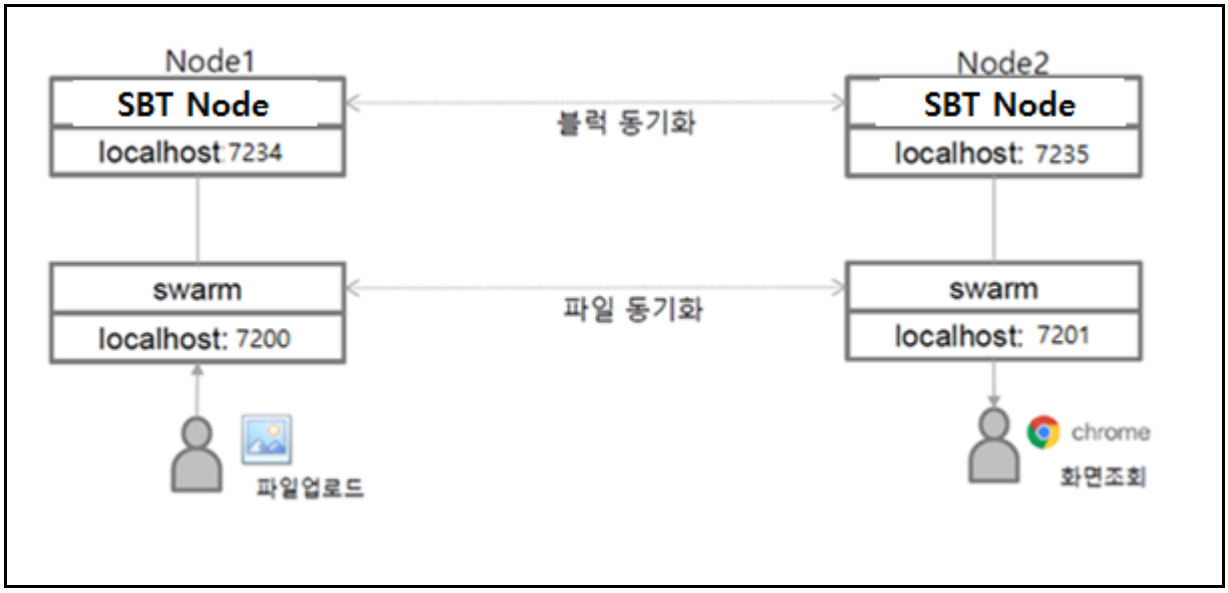
SBT 는 ERC20, ERC721 표준을 따르는 토큰발행시스템을 지원합니다. ERC20 Token 은 이더리움 블록체인 네트워크에서 발행되는 토큰의 표준 인터페이스로 디팩토 스탠다드다. 이 표준은 EVM 상에서 동작하는 스마트컨트랙트를 이용해서 생성되는 CryptoCurrency 이며 동시에 그 자체로 DApp 이다. 많은 블록체인 네트워크에서 이 표준을 수용하고 있다.

1.3 분산화된 Sunblockterminal 스토리지 DApp

Sunblockterminal.Storage PoC(Proof of Concept)는 Swarm 과 블록체인상의 NS(Name Service)를 사용하여 서버 없는 WWW 의 DNS 와 같은 컨텐츠 조회가 가능하며, 토큰과 연계되어 스마트컨트랙에 의한 거래 등의 응용확장을 제공하게 될 것이다.

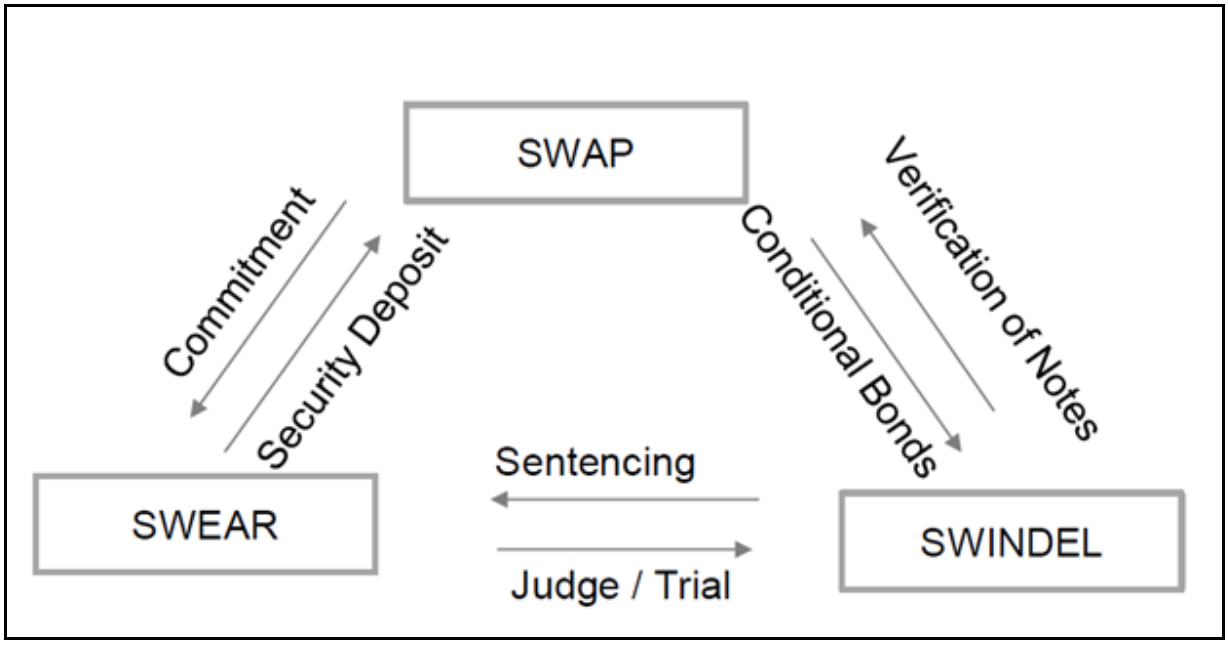


1.3.1 Sunblockterminal Storage 멀티노드 간 Swarm 동기화 및 사용



1.3.2 Swap, Swear and Swindle for Sunblockterminal Storage

Sunblockterminal.Storage network 를 유지하기 위한 Incentive System 의 기본 구성요소	
SWAP(Swarm Accounting Protocol, Secured With Automated Payments)	지연지불, 지불채널, 에스크로관리, 부채관리를 하는 스마트컨트랙트
SWEAR(Secure Ways of Ensuring ARchival or SWarm Enforcement And Registration)	멤버십 등록, 멤버십조건, 보증금 처리
SWINDLE(Secured With INsurance Deposit Litigation and Escrow)	감사(audits), 소송이관 처리



SBT.Storage 생태계의 Gadget 들은 SWAP, SWEAR, SPINDLE 을 완벽하게 지원하는 최초의 구현체기기들이 될 것이며, Sunblockterminal 의 사용자들에 더욱더 탈중앙화 된 분산 스토리지의 경험을 제공할 것이다.

2 보안 및 개인정보 보호관련

각종 서비스를 위해서는 개인정보의 저장 및 적절한 활용이 필수적이다. 하지만, 개인정보 및 빅데이터에서의 개인정보 추출과 관련한 많은 사건사고들이 발생을 하고 있는 것이 현실이다.

블록체인과 관련한 서비스를 하기 위해서는 일정부분의 개인정보를 취득할 수밖에 없다. 하지만, 블록체인은 공개적인 장부이므로 개인정보는 어떠한 방식으로도 블록체인에 저장을 할 수가 없다. 블록체인 자체의 암호화는 뛰어나지만, 개인을 특정할 수 있는 개인정보나 민감 개인정보는 저장을 할 수가 없다.

다음 표는 기존 인터넷서비스와 일반적인 블록체인, SBT 가 구현하려는 개인정보 보호를 보여준다.

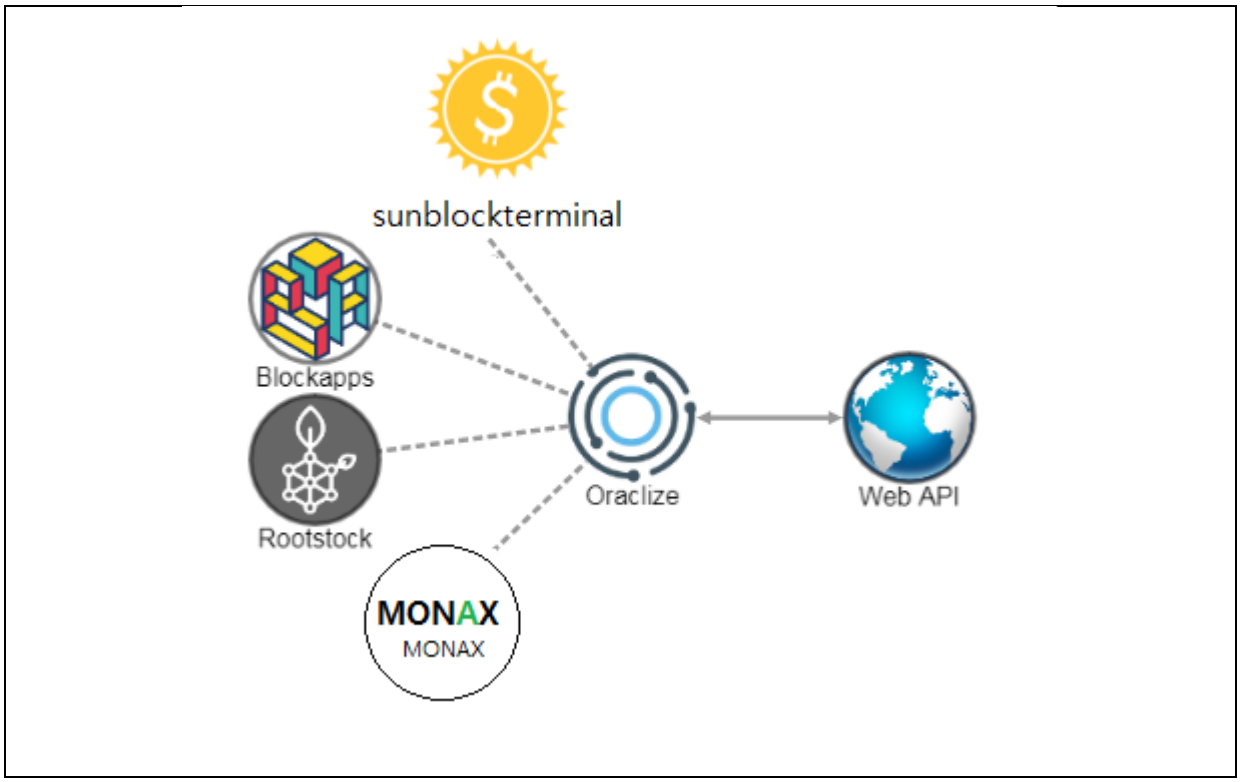
구분	인터넷	블록체인 네트워크	SBT
비밀성	X	X	○
인증	X	X	△
무결성	X	○	○
부인봉쇄	X	△	△

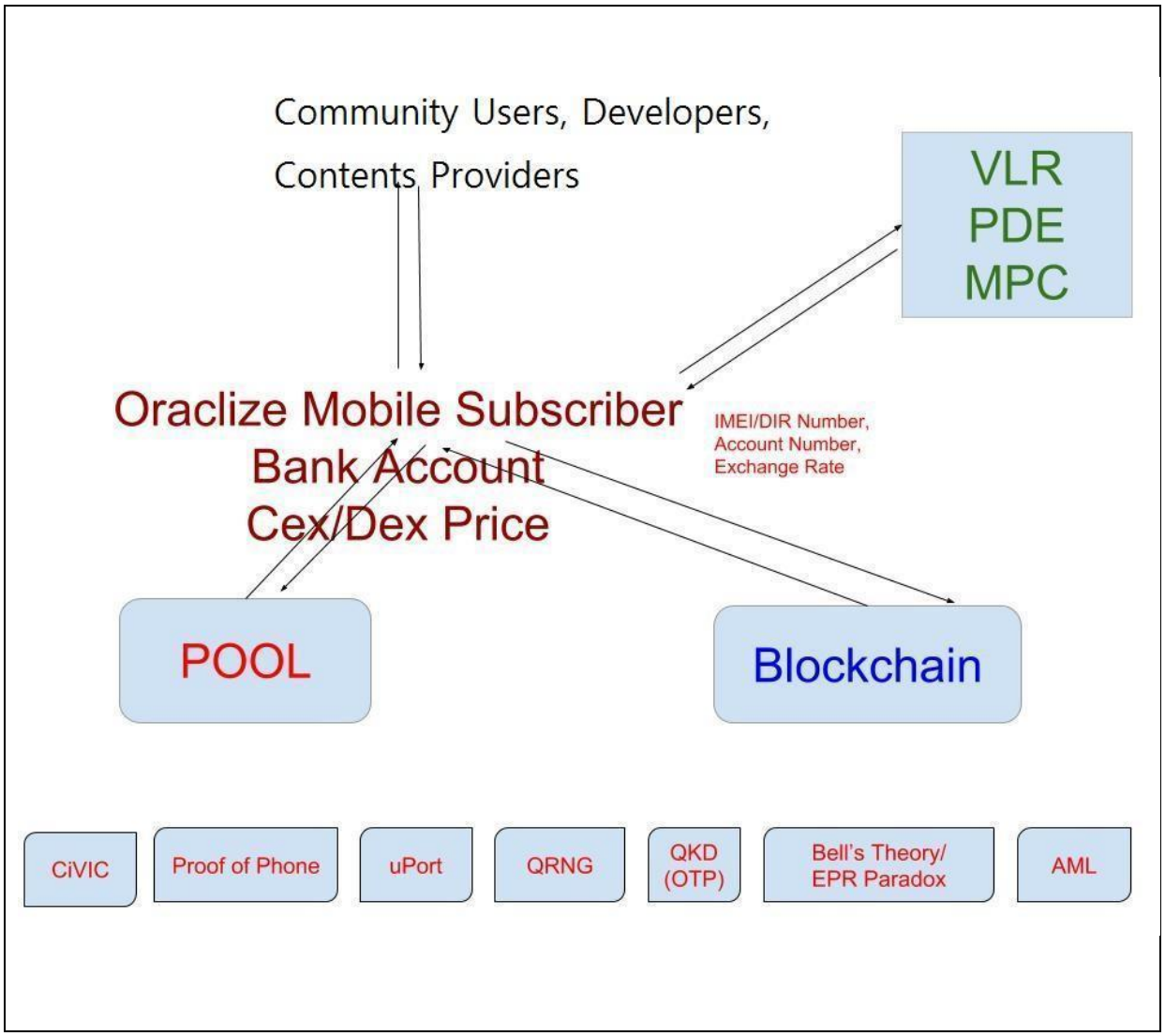
SBT 는 새로운 관점의 기술 접목 및 향후 진화 로드맵에 따라서 보안과 개인정보문제를 해결해 나가려고 한다. SBT 는 완전히 새로운 기술의 구현체이며, 블록체인 생태계에서 최초로 물리적인 현상의 원리를 사용한 근본적인 보안과 정보보호를 위한 기술을 제공할 것이다.

2.1 Oraclization

Sunblockterminal Platform 에 올라가는 계약정보의 소유권은 거래당사자에게 있다. 거래당사자는 이 정보를 작성했다는 증거를 블록체인에 기록하여 디지털자산의 소유권을 증명할 수 있고 그 데이터가 변조되지 않았다는 것을 증명할 수 있다. 또는 그 이후의 모든 수정기록에 대해서도 블록체인을 통해 입증 받을 수 있다. 하지만, 이 모든 데이터를 통째로 블록체인에 올릴 수는 없다. 모두 블록체인에 올린다면 블록체인의 데이터양이 굉장히 빠르게 증가하여 노드를 유지하기가 어렵다. 그러므로 Sunblockterminal Platform 에 올라가는 정보는 특정정보만 압축 암호화하여 블록체인에 저장된다.

이 과정에서 중요한 데이터들이 변조되지 않도록 보장하여 거래당사자와 디지털자산 데이터를 정확하게 연결하는 시스템이 필요하다. 이 시스템을 Sunblockterminal Platform 구현체라고 하며 블록체인상의 스마트컨트랙 사이에서 동작하는 계약제작자와 사용자 간의 인터랙티브 한 데이터를 주고받는 시스템이다. 이것은 데이터를 보호하는 충분하고 완전한 오라클라이재이션 해주는 시스템으로 중요한 데이터들이 변조되지 않도록 보장할 것이다.





기존의 구현체들을 깊이 있게 고찰하여 가장 많이 사용되는 기기인 모바일에 최적화된 Sunblockterminal 에 이 기능을 적용하며 이를 통해 Sunblockterminal Platform 을 사용하는 사용자 권리와 이익을 보장한다.

2.2 uPort Identity

여러가지 Exchange Service 를 사용하다 보면 여러가지 계정과 비밀번호를 사용하게 된다. 같은 아이디/비밀번호를 사용하게 되면 하나의 사이트에서만 계정 해킹이 발생해도 그 피해는 모든 사이트로 전파되어 피해가 막심해진다. 그렇다고 사이트마다 계정정보를 다르게 할 경우 계정정보를 쉽게 분실하게 된다. 이러다 보니 계정정보를 특정 계정 관리 서비스에 집중하여 관리하는 일도 생기는데 만일 이 서비스에서 해킹 피해가 발생하면 모든 관련 사이트가 동시에 해킹되는 일이 발생한다.

이를 해결하기 위해서는 기존의 계정정보 방식이 아니라 블록체인을 활용한 개인키/공개키 방식의 암호화 기술을 적용을 할 수도 있다. 하지만, 일반적인 Exchange Service 사용자가 블록체인의 개인키/공개키 시스템을 관리하는 것은 쉽지 않다. 그러므로 SBT 는 모바일환경의 일반사용자 수준에서 로그인정보를 쉽게 관리할 수 있는 솔루션을 지원한다. EVM 및 ERC20 을 지원함으로써, uPort identity 서비스를 Cryptocurrency Exchange 플랫폼에 수용할 수 있는 기반을 지원한다. 스마트컨트랙트에 관련 세부 로직을 추가하여 키의 취소, 복구와 사용자의 키 관리 부담을 줄여준다.

2.2.1 Introductionto uPort

uPort 는 EVM 을 기반으로 하는 사용자 개인식별 DApp 이다. 페이스북이나 네이버의 OpenID 와 비슷한 개념입니다. 보안성이 뛰어난 시스템으로 uPort 의 핵심기술은 스마트컨트랙트, 개발라이브러리, 모바일 앱이다. 스마트 컨트랙트에는 개인식별과 관련된 url 등의 정보화 암호화된 해시값과 개인식별정보가 변경 또는 분실되었을 때 복구할 수 있는 알고리즘 등이 포함된다. 개발자 라이브러리에는 개인식별정보와 연결되는 url, dropbox, google 등을 관리하는 정보가 포함된다. 모바일 앱에는 사용자의 키가 들어있어 본인이 해당되는 개인의 본인임을 확인할 수 있다.

uPort identities 의 발행 형태는 개인, 기관을 가리지 않는다. 생성한 identity 의 권한의 생성자에게 있으며 이를 self-sovereign identity 라고 합니다. identity 생성 및 확인이 중앙 집중화되어 있지 않다. 이 identity 는 전자서명, transaction 확인 등에 이용할 수 있다.

identity 에는 관련된 개인식별정보가 저장된 IPFS , Azure, AWS, Dropbox 등의 속성정보가 암호화된 해시로 저장되어 있어 개인정보의 파기가 필요한 경우 블록체인을 건드리지 않고서도 개인정보 원본을 쉽게 파기할 수 있으며 또한 해시정보를 이용해서 최초에 올린 식별정보에 대한 진위여부를 쉽게 확인 가능하다.

- uPort : EVM 기반. 자체적인 개인식별에 사용 가능. 보안성이 있는 시스템.
  - 3 가지 요소 : 스마트 컨트랙트, 개발 라이브러리, 모바일앱
    - 스마트 컨트렉트: 개인식별의 핵심 코어, 사용자가 기기를 분실한 경우 개인식별정보를 복구하는 로직을 포함
    - 개발 라이브러리 : 외부 저장소와 연결되는 정보 포함.
    - 모바일 앱 : 사용자 키 저장
- uPort identities
  - 형태: 개인, 기기, 객체, 기관
  - 권한 : 생성자(creator)에게 있음(self-sovereign identity). 사용자가 생성할 경우 사용자가 권한을 가짐. 생성 또는 확인을 중앙 집중적인 관리를 하는 제 3 자에 맡기지 않음
  - 핵심 기능: transaction 확인, 전자서명 등

- identity 는 off-chain data stores 와 암호로 연결
- 각각의 identity 는 속성 정보의 해시 저장 (identity 와 관련된 모든 데이터가 안전하게 저장되어 있는 IPFS , Azure, AWS, Dropbox 등 어디에서든 가능)
- Identities : 프로필변경, 친구 추가등의 자동 업데이트 가능, 특정 파일에 대한 읽기, 쓰기 허용 가능
- uPort identities 는 블록체인과 상호작용할 수 있기 때문에, 디지털 자산(암호화 화폐, 토큰 자산)을 관리할 수 있음

2.2.2 Proposed Use Cases

- self-sovereign identity 시스템의 특징으로
  - 생성자가 개인식별정보를 생성하는 시스템으로 사용자가 개인 identity, 평판, 데이터와 디지털 자산을 소유하고 통제한다.
  - 생성자가 개인식별정보를 생성하기 때문에 사용자가 선택적으로 자신의 데이터를 공개할 수 있다.
  - 키를 이용해 비밀번호 없이 디지털 서비스에 접근 가능하다.
  - 키를 이용해 디지털 트랜잭션, 디지털 문서에 서명 가능하다.

- 블록체인에서 값을 보내고 통제관리가 가능하다.
- 분산 어플리케이션과 스마트 컨트랙트와 상호작용 가능하다.
- 키를 이용해서 암호화된 메세지와 데이터전송이 가능하다.

- self-sovereign identity 시스템의 장점으로
  - 개인정보 공개에 대한 부담이 적으므로 신규 회원 가입이 쉽다.
  - 향상된 KYC(Know-Your-Customer) 프로세스를 마련할 수 있다.
  - 민감한 고객 정보를 보관하지 않아 법적 책임이 감소한다.
  - 개인정보를 취급하지 않으므로 직원의 규정 준수가 쉽다.
  - 쉽게 가입할 수 있으므로 콘텐츠 공급자 유치가 쉽다.
  - 특정 권한을 가진 세부 역할을 구분할 수 있다.
  - 개발이 완료되면 아무런 배경 지식이 없이도 쉽게 사용할 수 있다.

3 리워드 시스템

3.1 Sunblockterminal Economy

비트코인이 주목 받기 이전에도 탈중앙화된 디지털 화폐라는 아이디어를 시도한 사례는 많았지만, Digicash(1992), Cybercach(1994), e-Gold(1996)등이 있었고, 비트코인이 채용한 기술 일부를 만들어내기도 했다.

그렇지만 특정 주체가 권력을 가지지 않으면서, 동시에 모든 참여자가 신뢰할 수 있는 장부를 만들어 낸 것은 비트코인이 최초이다.

비트코인이 이전까지의 기술과 달랐던 이유는 컴퓨터공학, 암호학적 요소뿐만 아니라 참여자의 행동을 유도하는 경제학적 요소가 시스템에 녹아 들어 있었기 때문이다.

그 전까지의 시스템에선 어떤 기술을 적용했더라도, 결국에는 누군가가 책임과 권한을 가지고 네트워크 내의 규칙과 질서를 유지해야 했다. 그러나 비트코인은 이 규칙과 질서를 프로토콜로 대체함과 동시에 프로토콜의 규칙을 따르도록 유도하는 '인센티브 구조'를 만들었습니다. 비트코인 네트워크가 잘 되어야 참여자들이 이득을 보는 구조를 만든 것이다.

전 세계에서 수많은 사람이 블록체인을 활용한 새로운 서비스를 고안해 내고 있다. 이러한 서비스는 비트코인과 같이 내부에 각자의 서비스와 참여자에게 맞는 경제 체제를 탑재해야 한다.

Sunblockterminal Economy 안에서는 어떤 DApp 을 만들어도 그 위에 돌아가는 경제 시스템을 손쉽게 설계할 수 있다.

보상(토큰)은 어떤 기준으로 어떤 참여자에게 줄 것인가?

어떻게 토큰이 가치를 갖게 할 것인가?

사람들이 토큰을 보유해야 할 유인은 무엇인가?

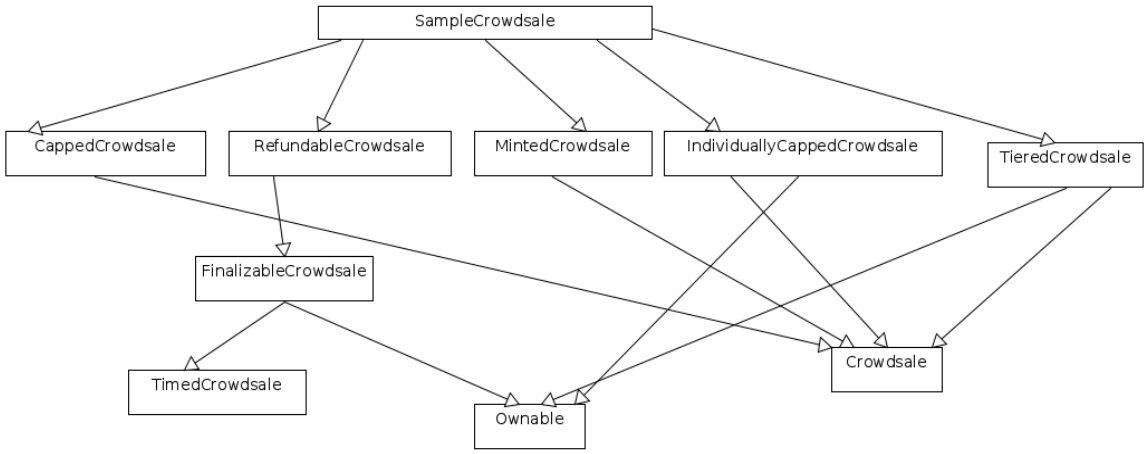
네트워크의 성장과 토큰의 가치 상승을 어떻게 연동할 것인가?

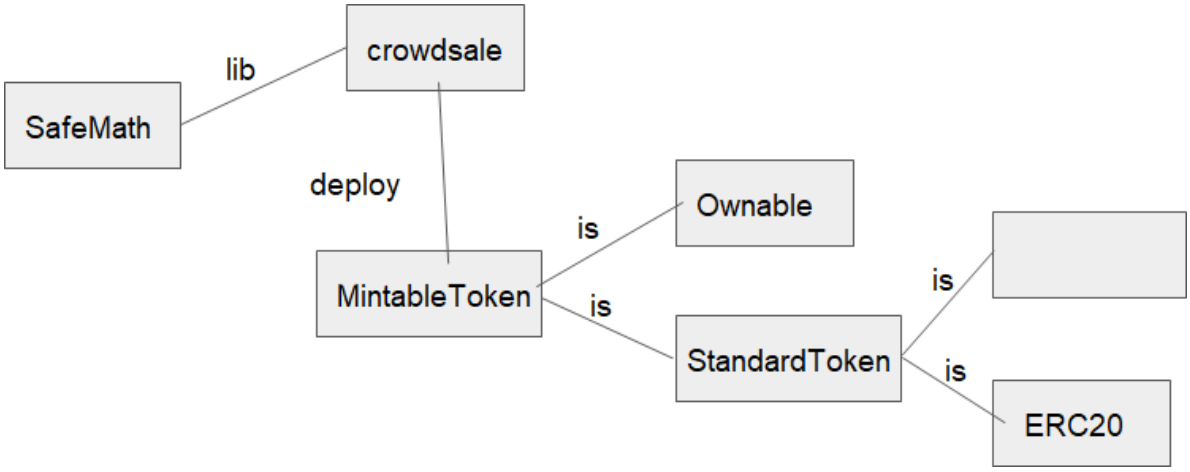
토큰의 가격 변동은 어떻게 해결할 것인가? 에 대한 토큰 이코노미가 Sunblockterminal 에 경제 구조의 인프라가 이미 구축되어 있으며, Sunblockterminal 의 성장과 연동하게 설계 및 구현될 것이다.

Sunblockterminal 참여자의 특성을 고려하여 그에 맞는 토큰 이코노미를 설계가 되어있으며, 모든 네트워크 참여자에 맞는 이코노미를 설계가 가능하다.

3.1.1 Truffle Framework

토큰을 발행을 위한 개발 프레임웍은 Truffle Framework 을 사용하며, ERC20 등 표준 인터페이스를 따르는 OpenZeppelin Contract Library 가 준비되어 있으며, 테스트에서 Deploy 까지 스마트컨트랙트를 개발하는 일련의 과정을 자동적으로 핸들링 할 수 있다.





elopio style: use the max-len solidity rule (#944) Latest commit 746673a 9 days ago

..		
BasicToken.sol	update constructor syntax for solidity 0.4.23 in numerous contracts (#...	15 days ago
BurnableToken.sol	update constructor syntax for solidity 0.4.23 in numerous contracts (#...	15 days ago
CappedToken.sol	style: use the max-len solidity rule (#944)	9 days ago
DetailedERC20.sol	update constructor syntax for solidity 0.4.23 in numerous contracts (#...	15 days ago
ERC20.sol	style: use the max-len solidity rule (#944)	9 days ago
ERC20Basic.sol	update constructor syntax for solidity 0.4.23 in numerous contracts (#...	15 days ago
MintableToken.sol	style: use the max-len solidity rule (#944)	9 days ago
PausableToken.sol	style: use the max-len solidity rule (#944)	9 days ago
RBACMintableToken.sol	Adding RBAC Mintable token (#923)	14 days ago
SafeERC20.sol	update constructor syntax for solidity 0.4.23 in numerous contracts (#...	15 days ago
StandardBurnableToken.sol	style: use the max-len solidity rule (#944)	9 days ago
StandardToken.sol	style: use the max-len solidity rule (#944)	9 days ago
TokenTimelock.sol	style: use the max-len solidity rule (#944)	9 days ago
TokenVesting.sol	update constructor syntax for solidity 0.4.23 in numerous contracts (#...	15 days ago

다음의 4,5,6 장에서는 이와 같은 Sunblockterminal Economy 를 준수하는 Token Design Pattern 에 따라 이미 검증된 OpenZeppelin ERC20 스마트컨트랙트 라이브러리와 트러플 프레임워크로 구현될 것이다.

4 보팅 시스템

블록체인 서비스에서 의사결정 과정에서 서비스 이용자들이 특정사안에 대해 자신의 의견을 피력하려고 하게 하려면, 네크워크가 이용자들에게 투표권을 부여해서 효율적으로 의견을 취합할 수 있다. Sunblockterminal 보팅 시스템에서는 투표권과 토큰을 결합하여 의견 취합에 적합한 패턴으로 작용한다.

Voting Token 에서 투표권은 토큰 자체가 될 수도, 네트워크 상의 수치로 주어질 수도 있다.

Sunblockterminal 에서는 Sunblockterminal 토큰에 비례하는 네트워크상의 수치로서 SBT CREDIT 이 주어진다.

SBT CREDIT 를 통한 의사결정 대상은 Sunblockterminal Platform 에대한 네트워크의 이용자들의 의견을 반영한다. 네트워크의 의견을 잘 반영하는 결과를 위해선 악의적인 행동에 대해선 제약을 두어야 하며 voting 에 많은 이용자들을 참여시키는 것이 중요하다. 이용자들의 투표참여를 격려하기 위해 의사 결정 참여 이외에 SBT CREDIT 에 비례하는 보상수익을 분배 받는다. 또한, SBT CREDIT 와 Sunblockterminal 직/간접적으로 연결되어, 네트워크 내 토큰의 가치상승과도 연결된다.

4.1 CREDIT 도입

Sunblockterminal 이용자는 SBT CREDIT 이라는 수치를 가진다. 이 CREDIT 는 네트웍에 올바른 참여를 할수록 그 수치가 올라간다. 반대인 경우, 그 수치가 떨어진다. 이 수치는 블록생성자 선출 등에 영향을 미치며, 이용자들 스스로의 자정 작용에 중요한 도구로서 작용한다.

4.2 부가적인 혜택

Sunblockterminal 서비스에서 사용되는 Sunblockterminal 을 네트워크에 예치하면 SBT CREDIT 을 확보할 수 있다. 네트워크에서 사용되는 Sunblockterminal 과 SBT CREDIT 가 간접적으로 연결되어, SBT CREDIT 의 수요는 Sunblockterminal 의 수요로 이어질 수 있다.

4.2.1 이자

Sunblockterminal 에서 SBT CREDIT 가 높을 수록 이자가 높아진다. SBT CREDIT 보유량에 따라 이자를 나누어 주며, 유일한 수단이다.

4.2.2 큐레이션 보상

큐레이션 보상이란 voting 을 한 사람에게 주어지는 보상이다. 이 보상의 크기를 결정하는 요소 중 하나는 voter 의 SBT CREDIT 이다. SBT CREDIT 가 높은 voter 가 더 많은 큐레이션 보상을 가져간다.

5 Sunblockterminal CREDIT

Sunblockterminal CREDIT 은 Sunblockterminal Platform 서비스 이용을 위한 유저의 신용평가 기반이 되는 신용등급을 나타냄과 동시에 '스마트 계약(Smart Contract)으로 묶여 있는 Sunblockterminal 비례하는 토큰의 기능 및 네트워크 전체에서 해당 이용자가 가지는 영향력의 레벨을 나타낸다.

6 Sunblockterminal Lightnode/ LightNet Architectures

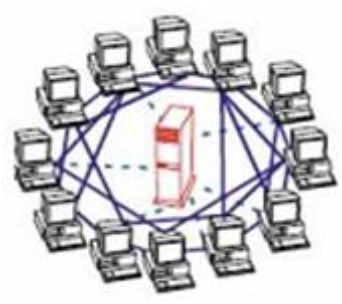
6.1 Nodes 들

- Node: 블록체인에 연결되어 있는 모든 서버들과 PC 들 각각을 이르는 말
- LightNode: 블록체인 P2P 시스템에서 데이터를 전달하고 분산시켜주는 역할의 중심에 있는 노드이다. 일반적으로 슈퍼노드 상호 사이에서는 광대역의 네트워크망으로 고속으로 통신한다. 빠른 CPU 와 Storage 장비를 이용하여 작업처리속도가 초고속으로 빠르며 많은 양의 업무를 처리한다. Sunblockterminal 의 Lightnode 는 일반적인 Lightnode 의 역할에 더해 블록생성의 역할을 함께 가진다. Lightnode 사이에서 광대역 네트워크를 형성하여 블록의 생성과 전달이 빠르게 이루어진다. 이를 초고속으로 수행하기 위해 Lightnode 는 Sunblockterminal 테스트넷에서 부하테스트를 거친 후에 Sunblockterminal 자체기준을 통과하는 노드에 한해 Sunblockterminal 의 자격이 주어진다. 빠른 블록생성과 전달을 위해 Lightnode 의 숫자는 적절한 수로 관리되며 그 수는 16~128 개의 범위내에서 생성 관리된다. 노드를 유지하는 보상으로 PoS 방식의 블록체인에서 지분(stake)을 보유하고 coin age 에 비례해서 cd 금리 concept 에 따라 이자로서 코인을 지급받는다. 자체적인 Lightnode 및 Sunblockterminal 에서 위임받는 Lightnode 가 존재할 수 있다.
- Fullnode: Lightnode 에서 생성된 블록체인을 전달받아 검증, 저장, 전달하는 노드이다. 다양한 서비스를 위해 데이타베이스를 구축하는 역할을 수행한다. 많은 경우 모바일 앱에 서비스를 제공하기 위한 경로로써 사용된다. 블록 생성권한은 주어지지 않는다.
- Lightnode: 블록체인을 저장하지 않은 노드이다. 머클트리와 블록체인의 일부 Depth(6)만을 유지함으로써 블록체인 네트워크에 트랜잭션을 보내고 Fullnode 와 통신하여 블록체인원장을 조회할 수 있는 모바일 등을 위한 경량화 노드이다.

6.2 LightNode 의 출현

- P2P Network 는 다음과 같은 세가지 형태가 있다.
  - Centralized P2P
    - 중앙서버가 있음
    - 정적 중앙집중
    - 검색비용 발생

- 노드가 많을수록 복잡도 증가

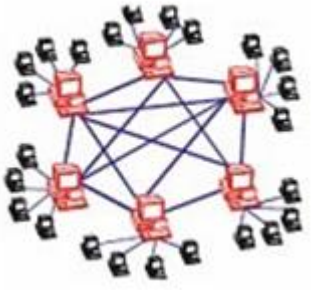


○ Pure P2P

- 중앙서버가 없음.
- 완전한 분산
- 검색비용 발생
- 노드가 많을수록 복잡도가 증가

○ Hybrid P2P

- 인덱스서버가 있음.
- 동적 중앙집중.
- 검색비용 적음
- 노드가 많아져도 복잡도 증가가 적음.



● 비트코인과 이더리움은 Centralized p2p network 모델을 사용하고 있다. 이더리움을 예로 들어보면 다음과 같이 진행된다.

○ 이더리움 프로그램(이하 데몬) 실행

○ 데몬이 이더리움 재단이 운영하는 중앙서버에 접속함.

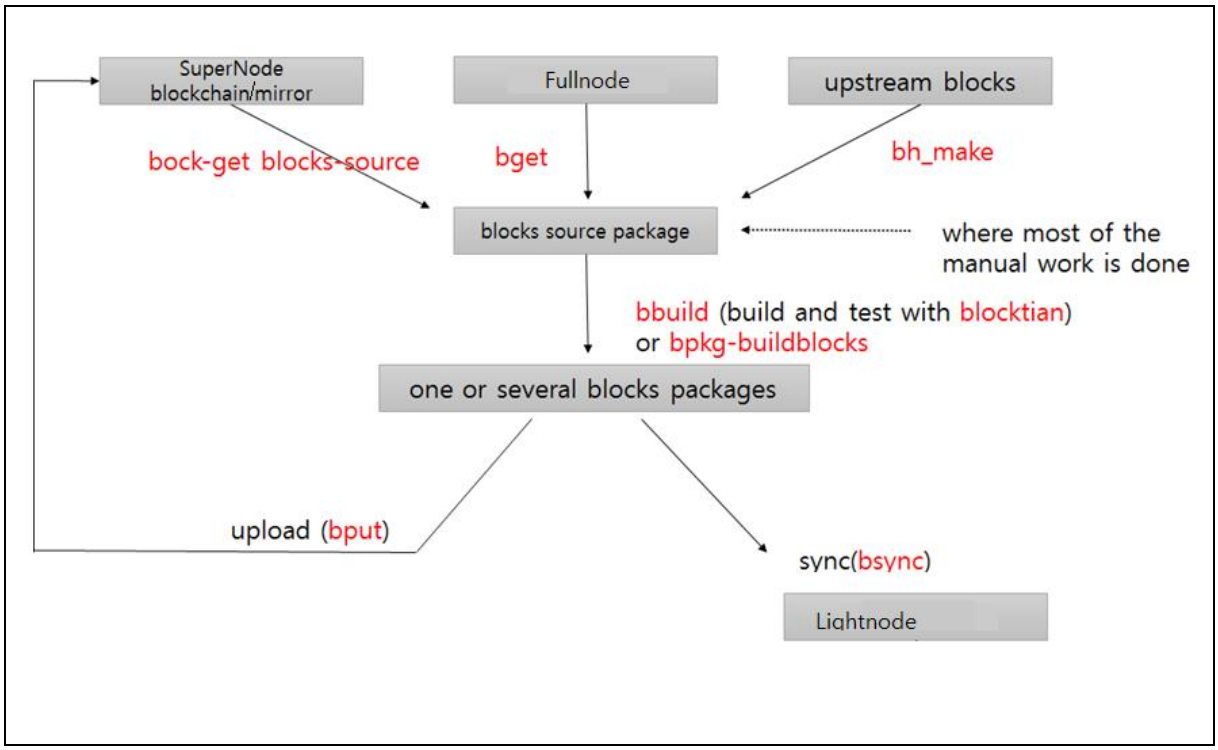
○ 중앙서버에서 운영되고 있는 다른 일반 노드 A, B, C 등에 대한 정보를 받아옴.
○ 데몬에서 A 에 접속시도(핸드쉐이크 Handshake)를 함.
○ 핸드쉐이크에 실패하면 B 에 핸드쉐이크 시도함.
○ 핸드쉐이크 성공하면 상대방 노드에 대한 정보를 받아옴.
○ 데몬 종류 및 버전 교환.
○ genesis block hash 교환.
○ block height 교환.
○ 정보가 일치하는 경우 동기화 시작.

- 위와 같은 순서로 블록의 전달이 시작된다. 이런 무작위 방법을 사용하기 때문에 노드 사이의 물리적인 거리가 멀고 네트워크 속도가 느리고 노드의 블록생성속도가 느리면 전체 네트워크에서의 블록 생성 속도와 블록 전달 속도에도 영향을 미쳐서 1 초당 transaction 처리속도(이하 tps)가 느려지게 된다.
- 비트코인이나 이더리움의 느린 블록생성속도를 해결하기 위해 Hybrid P2P 네크워크의 슈퍼노드를 사용하는 암호화폐가 점차 증가하는 추세이다.
- EOS 의 경우에는 Nxt→Waves, Bitshares→Stemmit→EOS, COSMOS 등의 개발 히스토리를 거쳐 노드를 집약적으로 운용하여 블록체인의 속도/용량문제를 해결하려고 시도하고 있다. 이 블록체인에서 슈퍼노드는 고용량의 트랜잭션과 블록생성을 고속으로 처리하기위한 처리능력을 가지고 있으며, 일반 node 들로부터 지분을 위임과 투표를 받아서 선출되는 소수(21 개~64 개)의 스페셜한 노드들이다.
- Sunblockterminal Platform 에서는 트랜잭션처리를 고속으로 처리하기 위해 Hybrid P2P network 구조를 가지며 16~128 개의 집약적인 Lightnode 를 구축할 계획을 가지고 있다.

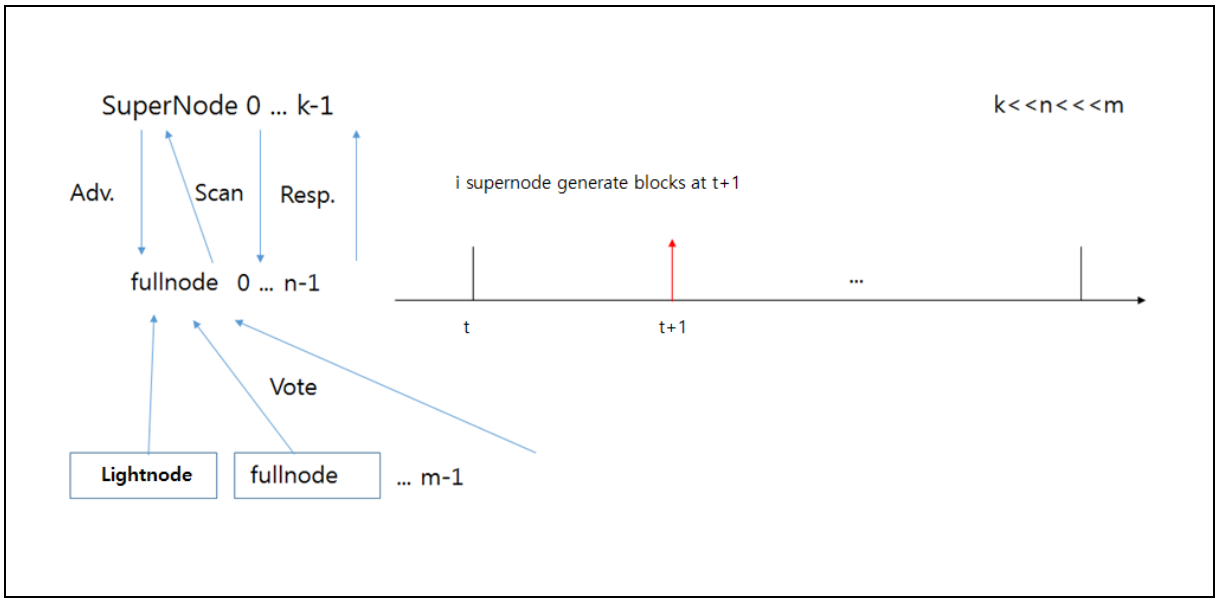
6.3 LightNet

Lightnode 의 출현으로 블록체인 네트워크망은 2 계층으로 분리되며, 본래 목적인 대용량의 트랜잭션처리와 고속의 블록생성을 위해서는 Lightnode 들을 연결하는 LightNet 이 필요하며, 비밀성과 인증을 수행할 수 있는 프로토콜과 물리적인 레이어를 갖추어야한다.

6.3.1 Lightnode block workflow



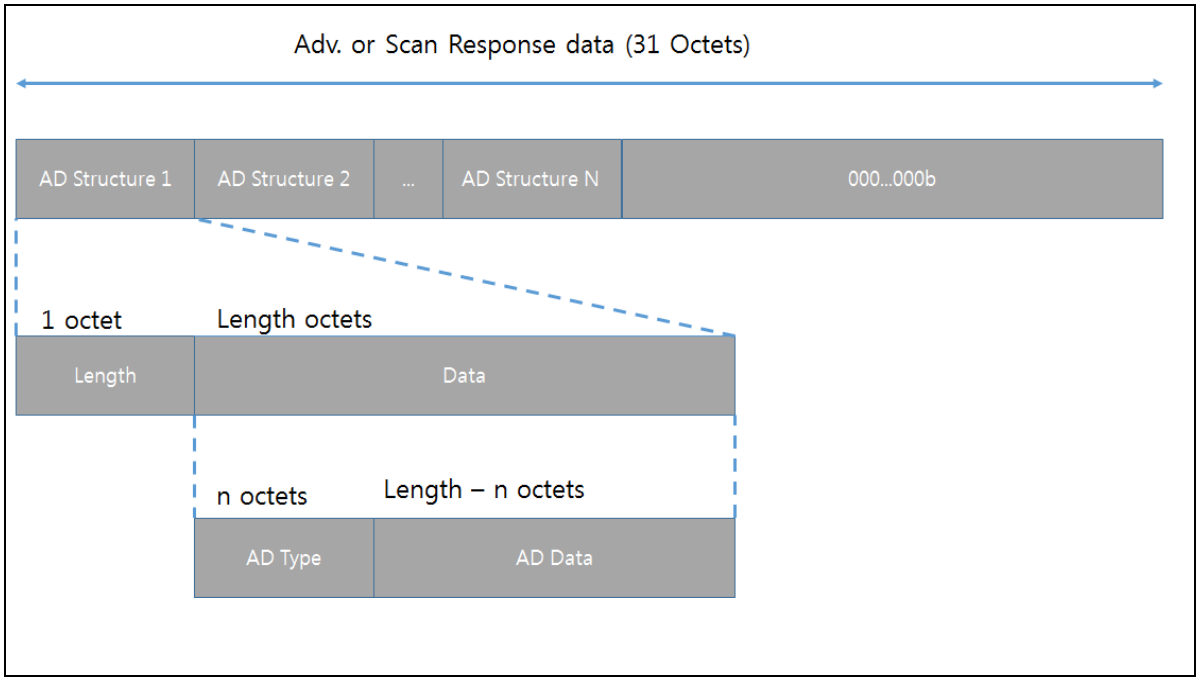
6.4 Lightnode block generating process



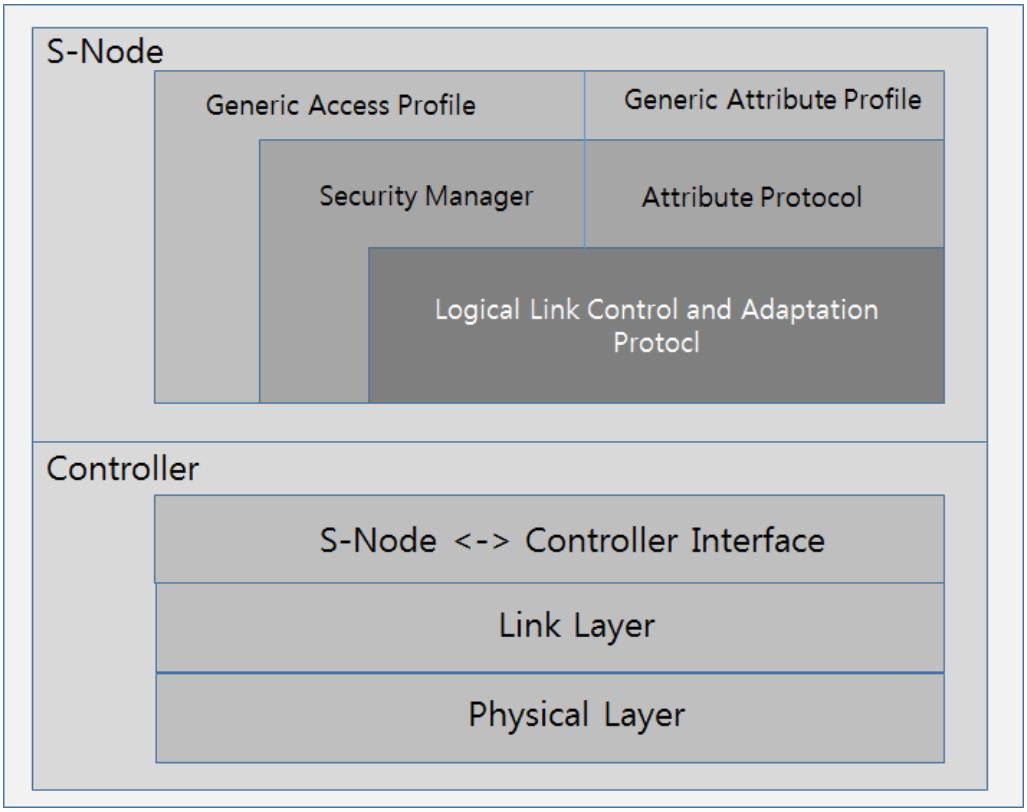
Sunblockterminal 는 Ethereum 의 Casper project 에서 시도되고 있는 PoS 합의 알고리즘을 사용한다. 일반적인 PoS 알고리즘에 의한 느린 블록생성 속도를 올리기 위해 Sunblockterminal 의

스펙기준과 네트워크 기준을 만족하는 node 를 Lightnode 로 참여시켜 블록생성 속도를 빠르게 처리하고 tps 를 극대화시킨다. 수퍼노드들을 연결하는 수퍼넷의 프로토콜은 테스트넷을 통하여 검증하고, 업데이트를 할 것이다.

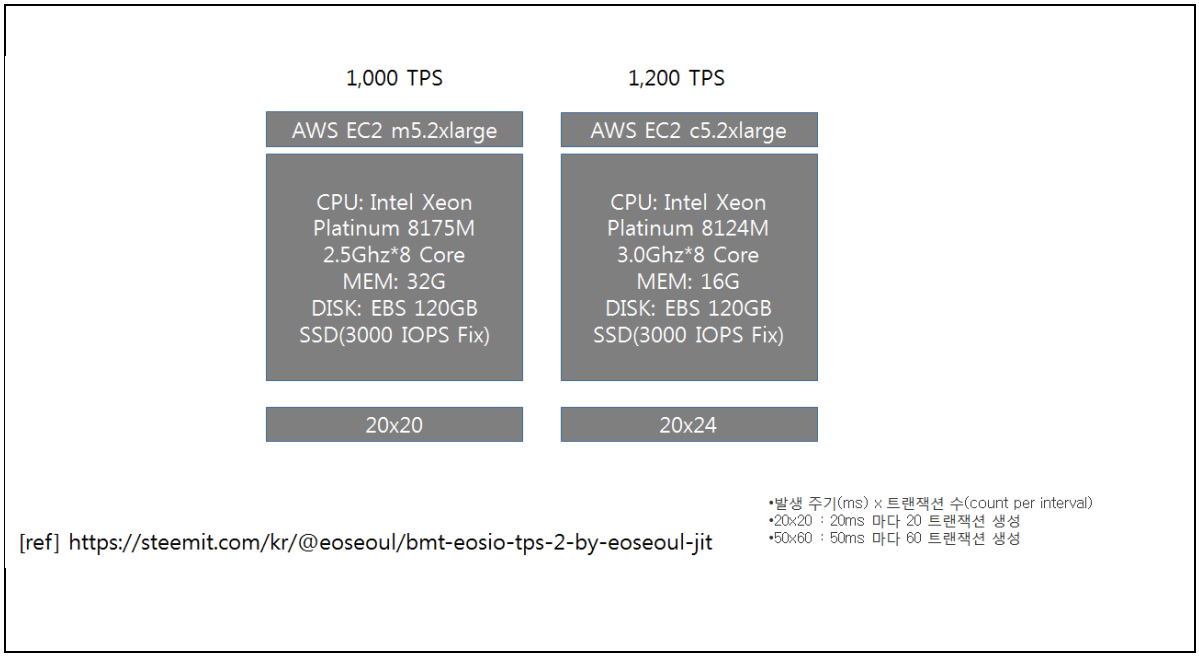
6.5 Lightnode discovery packet



6.6 Lightnode protocol stack

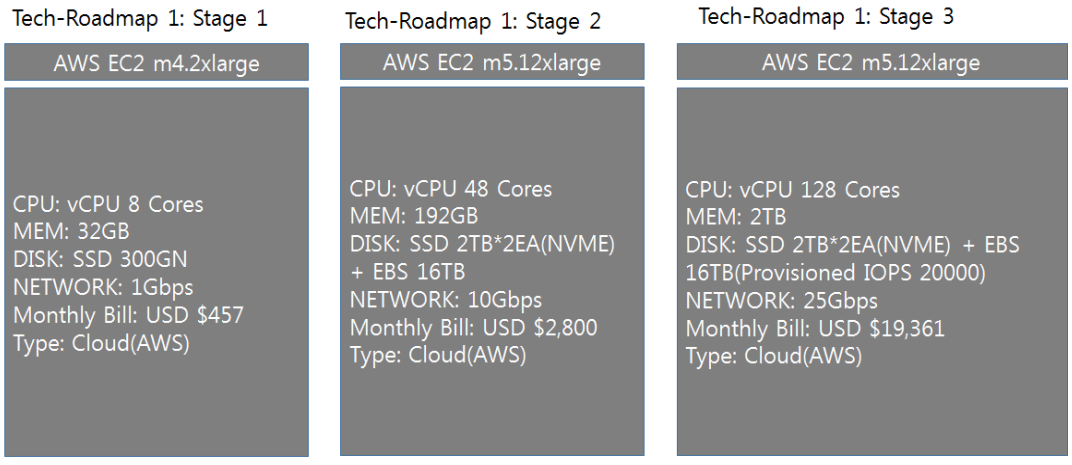


6.7 Lightnode dimension estimation



[ref] https://steemit.com/kr/@eoseoul/bmt-eosio-tps-2-by-eoseoul-jit

6.8 Lightnode testnode of Tech-Roadmap-1



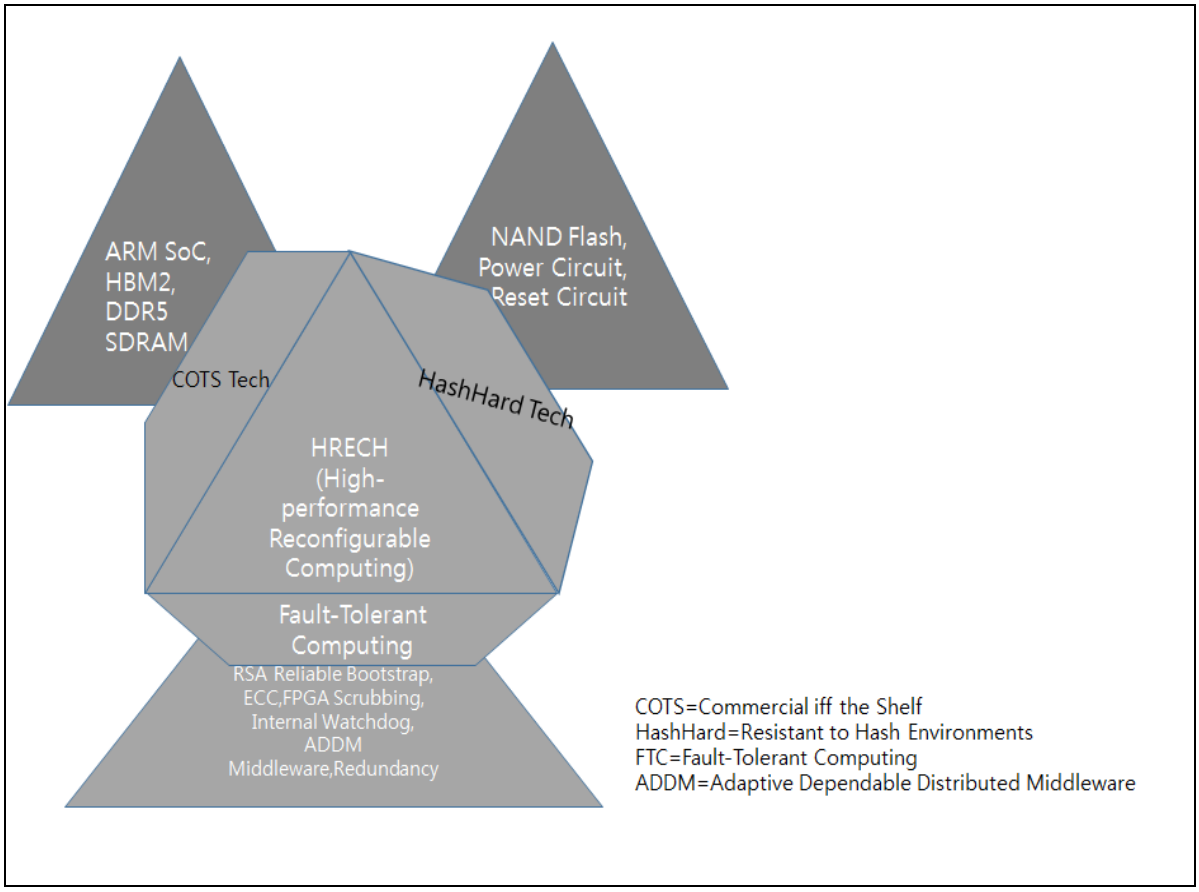
[ref] http://koreos.io/29589

6.9 HERC Lightnode

ASIC, GPU Miner 의 경우처럼 특정 제조업체에 블럭생성을 의존하는 경우의 다양한 문제점들이 TechRoadmap 1 단계에서처럼 Cloud Service 에 의존할 경우에도 반드시 발생할 것이다. 그러므로, Lightnode 를 설계 및 제작하여, LightNet 에 참여할 수 있는 방법이 OpenSource 로 제공되어야 한다. TechRoadmap 1 단계에서처럼 AWS 를 사용하는 경우 필연적으로 탈중앙화에 역행한다. 기존 PoW miner 들의 LightNet 에 참여를 유도하고, Mining Factory 환경에서도 AWS 정도의 신뢰할 수 있는 Lightnode 를 운영할 수 있도록 Guide 를 주어야한다.

*HREC=High-performance Reconfigurable Computing

6.10 Lightnode HREC for Tech-Roadmap 2,3



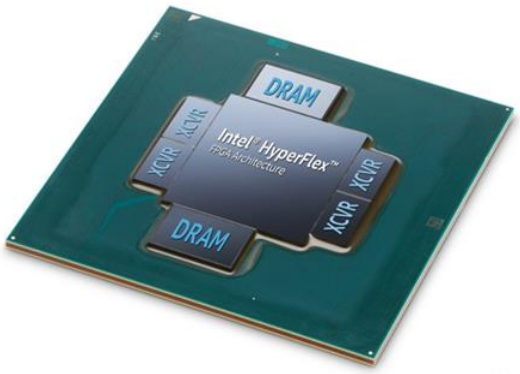
6.11탈중앙화 유지방안

누구나 LightNode 가 되기 위해서 자동화된 Protocol 과 Election 메커니즘에 의해서 자격을 갖춘(w/인증서) 노드는 LightNet 에 경쟁을 통하여 참여할 수 있는 기회가 부여된다.

**LightNode 인증서를 얻기 위해서는 Specification 이 정의된 testnet 에서 블럭생성 처리능력과 Interoperability 에대한 규격을 만족했을 때에만 취득하고 사용할 수 있다.

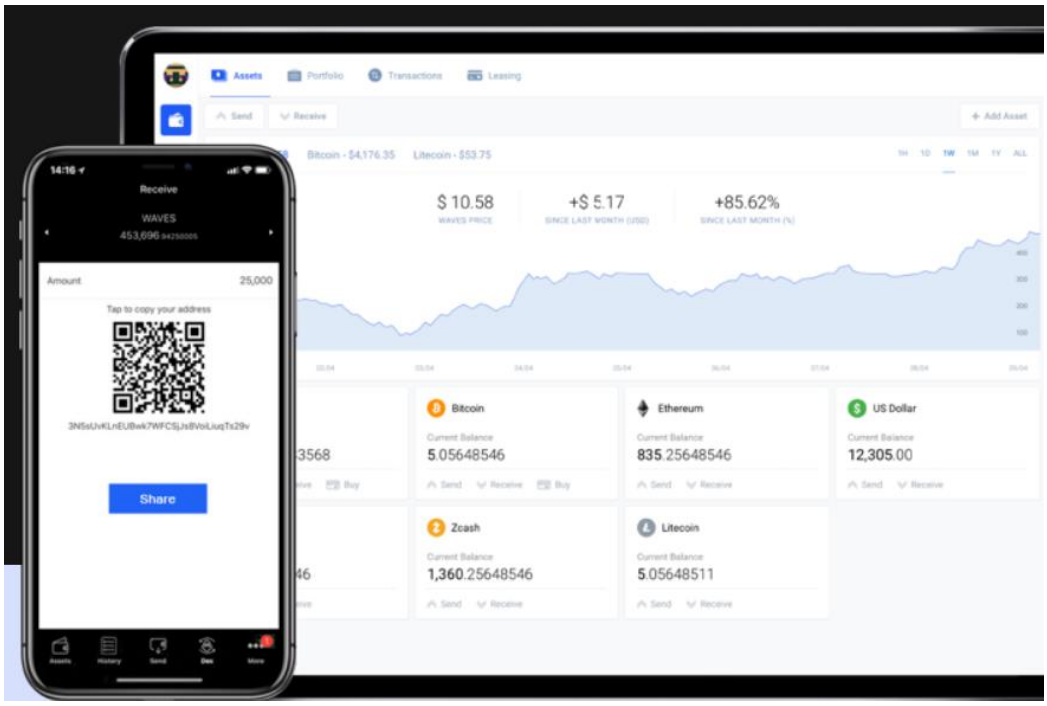
6.12Lightnode platform customization recommendation

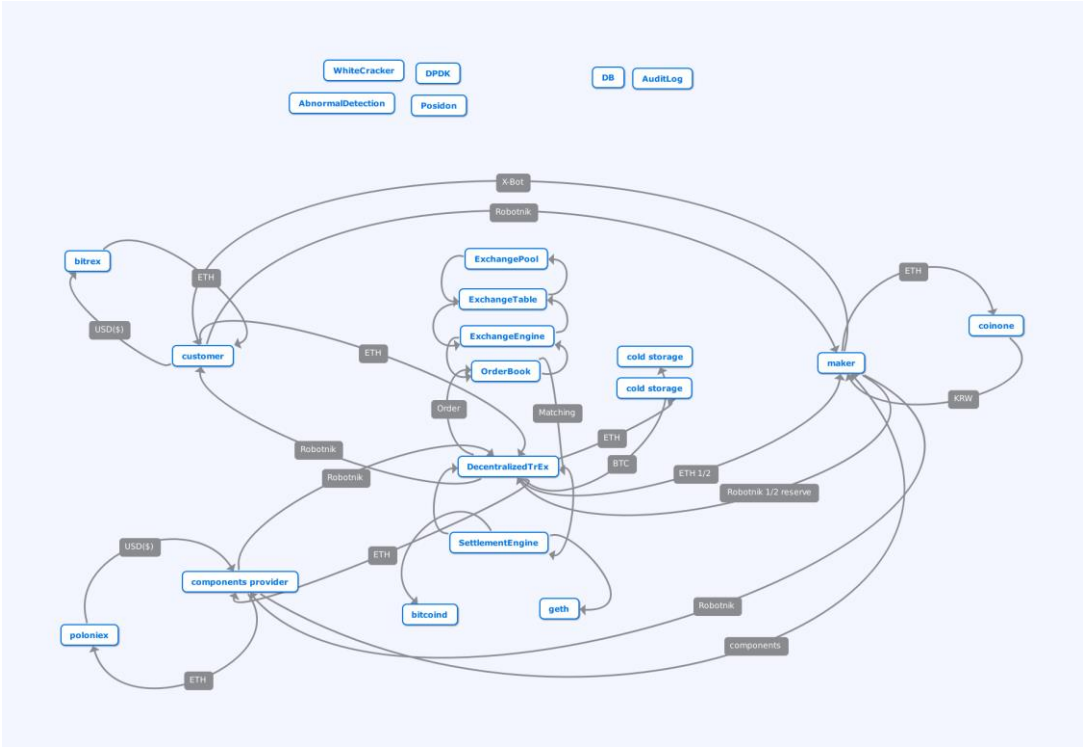
- Xilinx Zynq UltraScale+ MPSoC ZCU106
- 인텔, HBM2 메모리 통합한 Stratix 10 MX FPGA



7 Exchange Protocol(거래소연결)

SBT 는 글로벌 서비스를 지향함과 동시에 Sunblockterminal 을 이용하는 모든 유저들이 보다 편하게 법정화폐(Fiat)로 쉽게 환전할 수 있기를 원한다. 이를 위해 세계 각국 메인 거래소와 연계를 통해 앱 상에서 현재 유저가 보유한 코인의 시세를 실시간으로 나타낸다. 또한, StableCoin 과 Dex 와 연동을 통해서 가치 안정화된 토큰거래를 제공한다.


Assets
Portfolio
Transactions
Leasing
Send
Receive
Add Asset
Bitcoin - $4,176.35
Litecoin - $53.75
$ 10.58
+85.62%
Bitcoin
Current Balance
5.05648546
Ethereum
835.25648546
US Dollar
12,305.00
Zcash
1,360.25648546
Litecoin
5.05648511
14:16
Receive
WAVES
453,696
Amount
25,000
Tap to copy your address
Share


WhiteCracker
DPDK
AbnormalDetection
Posidon
DB
AuditLog
X-Bot
Robotnik
bitrex
ETH
USD($)
customer
ExchangePool
ExchangeTable
ExchangeEngine
OrderBook
cold storage
cold storage
Order
Matching
maker
coinone
KRW
DecentralizedTrEx
BTC
ETH 1/2
Robotnik 1/2 reserve
SettlementEngine
components provider
bitcoind
geth
poloniex
components

Sunblockterminal Platform 은 우선 Blockchain 기반의 개발 프로젝트를 대상으로 한다. 그러한 목적을 달성하기 위해 탈중앙화 투표 및 메시징이 구현된다. 이에 따라 커뮤니티 프로젝트를 관리하는데 있어 DAO 와 같은 경험을 허용하는 동시에 기술적인 관점에서는 단순하게 유지된다.

Sunblockterminal 은 커스텀 토큰(자산)으로 네트워크 거래 수수료를 지급하도록 허용한다. 그러한 거래와 함께 자산을 주 네트워크 토큰으로 교환하라는 주문이 탈중앙화된 거래소로 전송되고, 주문이 처리되어야 다음 블럭에 거래가 추가될 수 있다.

7.1 Cryptocurrency 에 최적화된 P2P Exchange based on Masternode(Node Service)

기본적으로 사용자당 최소 10 개의 코인을 오픈해주며, 오픈 후 업데이트가 가능하다. 블록체인 별 세팅이 가능하며, 코인 별 블럭체인 노드 컴파일 및 설치는 자동화된다. 코인 별 액세스를 위하여 RPC 모듈을 제공한다. 노드운영을 위한 기본작업 및 마스터노드를 제공해주며, 일부 코인의 경우 경량의 월렛수준으로 제공할 수도 있다. 이 마스터노드 또는 월렛들은 Sunblockterminal Platform 과 Payment channel, State Channel 등으로 연결될 수 있다.

Sunblockterminal 의 Dex 를 통하여 Sunblockterminal 토큰을 매도/매수할 수 있으며 일반, 분할, 입찰, 분할입찰 등이 가능하며, 폴링이 없는 시세정보 수집시스템을 구축한다. 거래체결, 공지 등의 알림서비스 및 푸시, SNS, 이메일 등의 수단을 사용한다. 거래내역 및 통계조회를 다양한 종류의 수단을 사용할 수 있도록 한다.

zk-SNARKs 라이브러리를 사용한 단체채팅방을 사용할 수 있게 해주며, 사용자간 거래관련 문의 및 협의를 위한 메신저서비스를 제공한다.

7.1.1 Ring Signature

A ring signature is a type of digital signature in which a group of possible signers are merged together to produce a distinctive signature that can authorize a transaction.

Ring signature 로 서명된 메시지는 특정 그룹의 사용자가 보증

링 서명의 보안 속성 중 하나는 그룹 구성원의 키 중 어느 것이 서명을 생성하는 데 사용되었는지 결정하는 것이 계산 상으로 불가능하다는 점

The actual signer is a one-time spend key that corresponds with an output being sent from the sender's wallet.

ring size: ring signature 에 서명해야 하는 서명자의 수, 많을 수록 보안이 강화됨, ring size 가 4 면 3 개의 foreign output 과 하나의 "real" output 이 있음

트랜잭션이 어디로 전송되는지 알아차리지 못하게 하는 것

Monero 는 링 서명 기술을 사용하기 때문에 double spending 문제를 해결하기 위해 링 서명 트랜잭션에서 소비되는 출력을 확인할 수 있는 기능을 포함해야 함

Monero 의 key image 사용으로 해결

- Bob 이 Alice 에게 Monero 를 보내려고 하면 ring size 가 5 이고 5 개의 입력 중 하나는 Bob 의 지갑에서 가져와 링 서명 트랜잭션에 추가된 것
- 나머지 네 개는 Monero 블록체인에서 가져온 과거 거래 내역
- 이 4 개의 인풋은 디코이이라고 하며, Bob 의 트랜잭션과 융합되어 5 명의 서명자 그룹을 형성
- 제 3 자는 Bob 의 one-time spend key 에 의해 실제로 어떤 Tx 가 서명되었는지를 확인할 수 없음
- key image 를 사용하면 Alice 의 계정으로 전송되는 Monero 가 이전에 지출되지 않았음을 확인할 수 있음

7.1.2 Stealth address

모네로 보안의 핵심 중 하나, 그들은 송신자가 수신자를 대신하여 모든 트랜잭션에 대해 임의의 일회성 주소를 생성하도록 허용하고 요구한다.

When you create a Monero account you'll have a private view key, a private spend key, and a Public Address.

view key - 해당 계정으로 들어온 트랜잭션을 볼 수 있는 키, 해당 계정에서 보낸 트랜잭션은 볼 수 없음

spend key - 해당 계정의 트랜잭션을 발생시킬 수 있는 키, 다른 키들을 파생시킬 수 있다.

view key 를 통해 모네로는 optionally semi-transparent

7.1.3 One-Time Account System

Account Generation Algorithm

1. Main account generation

앨리스의 오리지날 계정을 (A, a), one-time account system 에서의 메인 계정을 (B, b)라고 하면 앨리스의 메인 계정의 프라이빗 키는 (a, b), 퍼블릭 키를 (A, B) 라고 함.

앨리스는 메인 계정의 주소로 private key (a, b), scan key (A, b), 퍼블릭키 (A, B)를 가짐

2. Sub-account generation

밥이 앨리스에게 트랜잭션을 보내려고 하면 앨리스의 메인 계정 (A, B)는 sub-account (A1, S1)를 생성, (A1, S1)는 one time account

밥이 random number s 를 생성해서 S1 = [s] G 와 A1 = A + [Hash_p([s]B)] G 를 계산, (A1, S1)은 one-time account.

7.1.4 ECDSA G

Signature generation algorithm [edit]

Suppose Alice wants to send a signed message to Bob. Initially, they must agree on the curve paramete
multiplicative order of the point G.

Parameter	
CURVE	the elliptic curve field and equation used
G	elliptic curve base point, a generator of the elliptic curve with large prime order *n*
n	integer order of *G*, means that $n \times G = 0$

7.1.5 Stamp System

어떤 계정에서 트랜잭션 비용을 지불하는지 결정하는 시스템, one-time account system 에서는 트랜잭션을 추적할 수 없기 때문에 어떤 트랜잭션의 비용을 지불해야 하는지 알 수 없음

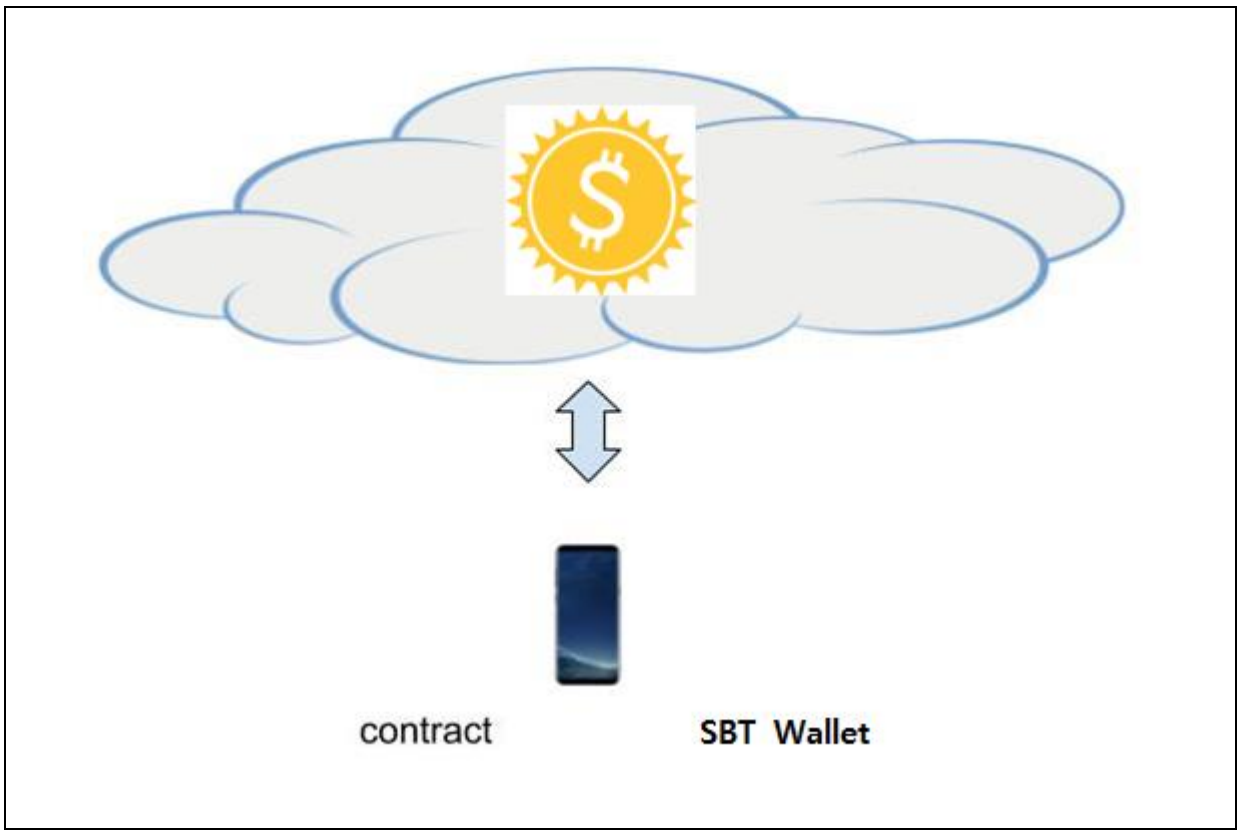
기술적으로 one-time account 와 one-time stamp 는 같음

유저는 트랜잭션을 수행하기 전에 스탬프를 구매, 스탬프는 트랜잭션으로 복제되어 한 번만 사용

8 Sunblockterminal.phone

우분투 OS 는 전세계인이 사용하게 된 데비안 계열의 리눅스 OS 이다. 2017 년에 야심차게 추진되었던 스마트폰에 포팅되어 작동하는 것을 목표로 했던 "Ubuntu for Phone", "UbuntuTouch" 프로젝트는 안타깝게도 좌초되었으나, Sunblockterminal 에서는 모바일환경에서 블럭체인 Full node 를 확산하기 위해서는 embedded linux 기반의 phone 위에서 동작하는 Sunblockterminal 블럭체인을 고려하게 되었으며, 진정한 Worldwide 한 Billion full-node 를 가지는 블록체인 프로젝트가 될 것을 목표로 한다. SBT.Storage, SBT.Messenger 가 SBT.Phone 의 대표적인 DApp 이 될 것으로 예상한다.

우선적인 Target phone 으로는 Galaxy S8, Oneplus5T, Mate10 이 될 것이며, Custom Phone 도 미래에는 등장할 것을 희망한다.

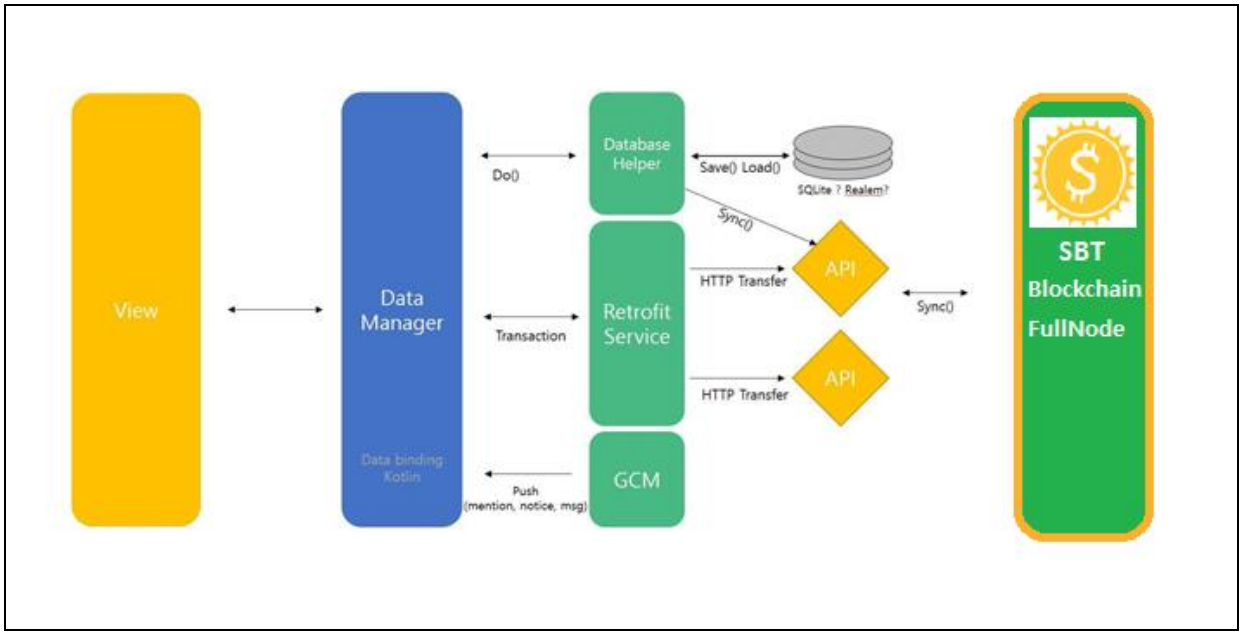


SBT.Storage SBT.Messenger SBT.Wallet(SBT,DApp Contract)

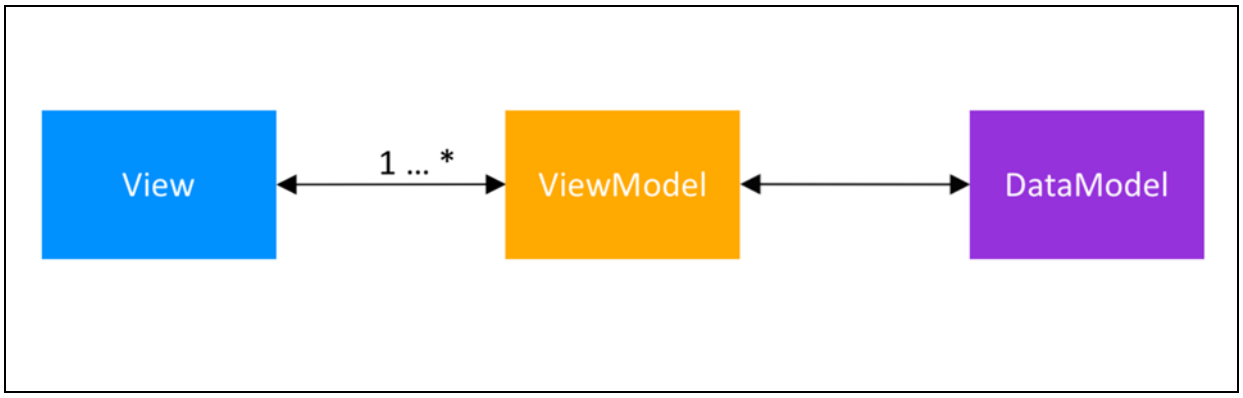
SBT for android 소프트웨어는 Sunblockterminal 블록체인 네트워크에서 사용되는 SBT 를 내 손안에서 사용하기 위한 모바일 플랫폼입니다.

*주의사항: 해당 문서에서 언급되는 사용은 Sunblockterminal Platform 을 활용해 SBT 가 이용되는 모든 상황을 말한다. (적립, 사용, 거래, 판매 등)

8.1 MVVM 아키텍처



기본적으로 MVVM(Model – View – ViewModel) 아키텍쳐를 기반으로 설계한다. MVVM 아키텍쳐는 뷰에 대한 의존성을 최대한 없애는 방향으로 설계하고 Unit Test 와 모듈화를 용이하게 만들도록 도와준다.



MVVM 에서는, ViewModel 이 이벤트들의 스트림을 뷰가 그 스트림에 바인드 할 수 있도록 노출시킨다. ViewModel 은 MVP 아키텍처의 Presenter 와는 다르게 View 의 참조자를 가지고 있을 필요가 없다. 이것은 MVP 가 필요로 했던 모든 인터페이스가 없어져도 된다는 것을 의미한다.

뷰들은 또한 ViewModel 에게 다른 액션들이 발생했음을 알려준다. 그래서, MVVM 패턴은 View 와 ViewModel 의 양방향 데이터 바인딩을 지원한다. 그리고 View 와 ViewModel 은 many-to-one

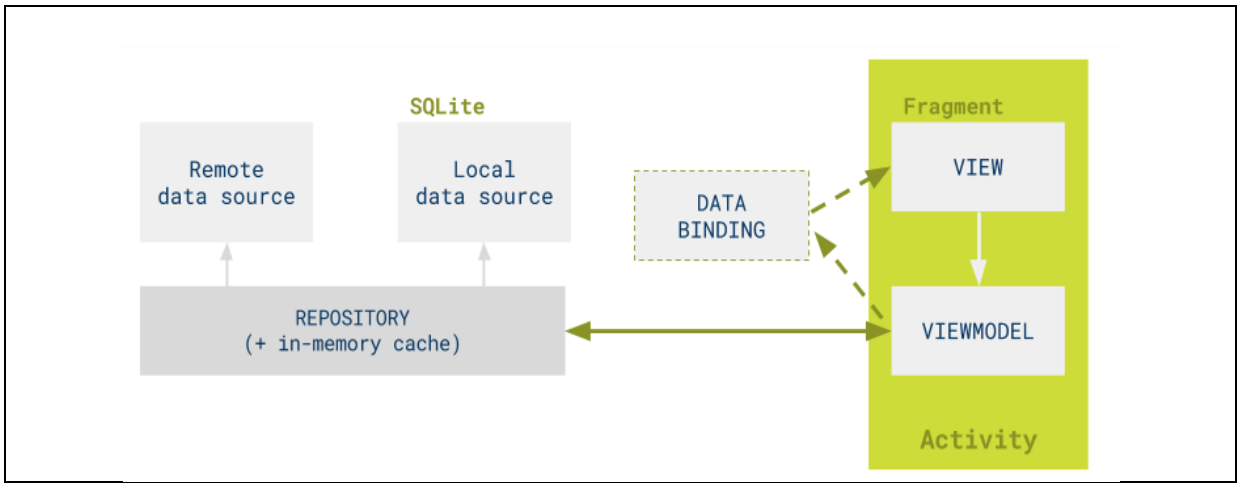
관계가 있다고 할 수 있다. View 는 ViewModel 의 참조자를 가지게 되지만, ViewModel 은 View 에대한 정보가 전혀 없게 된다. 데이터 소비자(consumer)는 데이터 공급자(producer)에 대해서 알아야 하지만 데이터 공급자(이 경우엔 ViewModel)는 데이터 소비자가 누구인지 알지도, 신경 쓰지도 않는다.

Data binding

데이터 바인딩을 이용하여 안드로이드의 각 View 에서 발생하는 이벤트들을 바인딩할 수 있습니다. 이로 인해 코드 생산성이 더 좋아지고, 데이터를 따로 관리해야 하는 로직들이 사라져 더욱 견고한 비즈니스 로직에만 집중할 수 있는 애플리케이션을 설계할 수 있습니다.

http://gun0912.tistory.com/71

https://developer.android.com/topic/libraries/data-binding/



Kotlin

IntelliJ IDEA 및 Android Studio 의 개발사 JetBrains 에서 개발한 언어로 Google.io 에서 안드로이드의 공식 언어로 지정되었다. 간결한 문법을 가지고 있으며, JVM 기반의 환경에서 동작이 가능하며, Java 와의 상호 운용이 100% 지원된다. JVM 바이트코드가 기본이지만,

Kotlin/Native 컴파일러를 사용하여 기계어 또는 LLVM 으로 최종컴파일이 가능하다. 안드로이드, 톰캣, JavaScript, Java EE, HTML5, iOS, 라즈베리 파이 등을 개발할 때 사용할 수 있다.

아래 링크는 Kotlin 과 Data binding 을 활용한 구글의 예제 프로젝트이다.

https://github.com/googlesamples/android-architecture/tree/todo-mvvm-databinding/

8.2 Data Model

HTTP 전송에 사용되는 모든 Data 는 JSON-RPC 모델을 차용한다. 아래 Gson 라이브러리를 활용하여 모든 Object 를 json 형태로 전송한다.

Gson (Google + JSON)

https://github.com/google/gson

Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. Gson can work with arbitrary Java objects including pre-existing objects that you do not have source-code of.

There are a few open-source projects that can convert Java objects to JSON. However, most of them require that you place Java annotations in your classes; something that you can not do if you do not have access to the source-code. Most also do not fully support the use of Java Generics. Gson considers both of these as very important design goals.

8.3 Network Transfer

Square 사의 Retrofit Library 를 이용하여 HTTP 통신을 진행하며 Blockchain Main network 와 연동되는 API 서버와 함께 통신한다. Retrofit Library 는 Rest API 에서 사용하는 CURD 기능을 모두 제공하고 있다. (POST (create), PUT (update), GET (read), DELETE (delete))

Retrofit 은 다른 HTTP Library 보다 월등히 빠른 속도를 제공한다. 간결한 네트워크 로직을 제공하며 재사용성을 증가시켜 빠르고 좋은 품질의 코드를 작성할 수 있다.

	One Discussion	Dashboard (7 requests)	25 Discussions
AsyncTask	941 ms	4,539 ms	13,957 ms
Volley	560 ms	2,202 ms	4,275 ms
Retrofit	312 ms	889 ms	1,059 ms

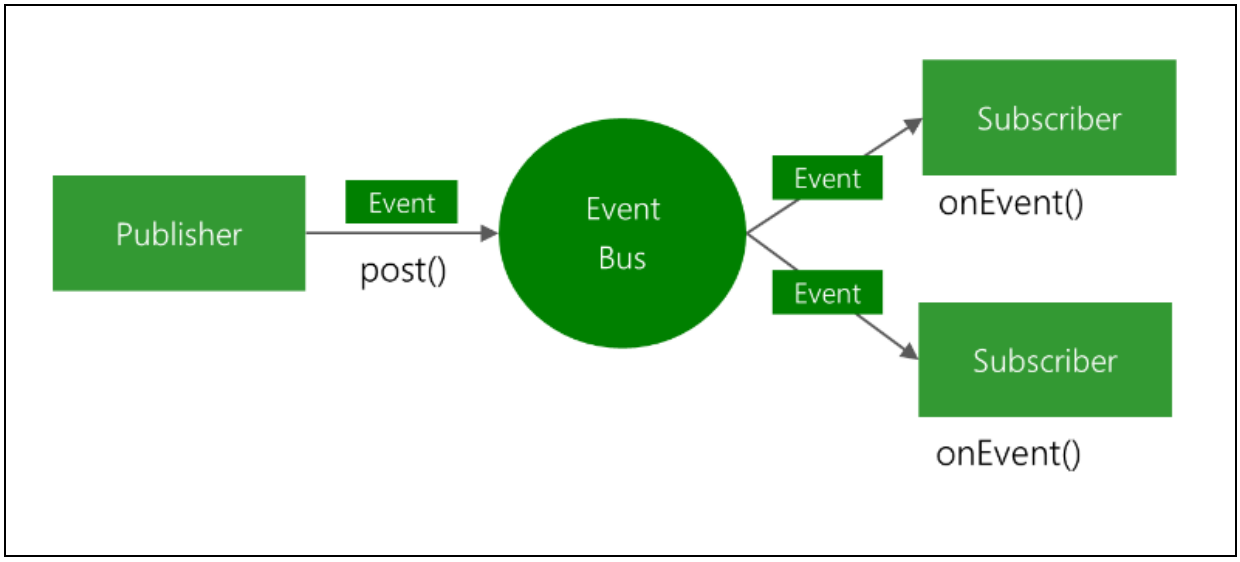
http://instructure.github.io/blog/2013/12/09/volley-vs-retrofit/

Square Retrofit

A type-safe HTTP client for Android and Java

http://square.github.io/retrofit/

EventBus 라는 Java 라이브러리를 활용해 콘텐츠 크리에이터의 구독/발행 모델에 대한 Event 를 전달받아 사용자에게 알려줄 수 있도록 한다. 독자와 크리에이터 간의 상호 연결을 향상 킬 수 있다.



Event Bus

EventBus is a publish/subscribe event bus for Android and Java.

https://github.com/greenrobot/EventBus

9 QRNG/OTP

암호학적으로 True Random Number 의 생성은 정말로 중요한 요소이다. 하지만, 실제 경제적인 트랜잭션의 시간의 제약에 따라서 Pseudo Random Number 를 이용하고 있는 실정이다.

2014 년 8 월 17 일 산동대 정보연구소의 여성암호학자 왕샤오윈이 "Collision for Hash" 발견의 연구발표후에 더욱더 True Random Number 의 중요성이 커지고 있다.

Sunblockterminal 은 탈중앙화된 분산화 된 커뮤니트를 구축하고 서비스하려는 시점에 깊이 고려해보아야 하는 문제점을 인식을 했고, 기준의 암호화폐 주소 생성과 코인 또는 토큰전송에 있어서 근본적인 안전을 추가하려, QRNG 의 사용을 실험하고, 그 새로운 이용방법을 제안을 할 것이다.

비트코인이 공개키를 사용하여, 서명알고리즘을 만든 이후 모든 알트코인들은 이방법을 사용하여 암호화폐의 전송에 사용하는 것은 널리 알려지고 적용되었다.

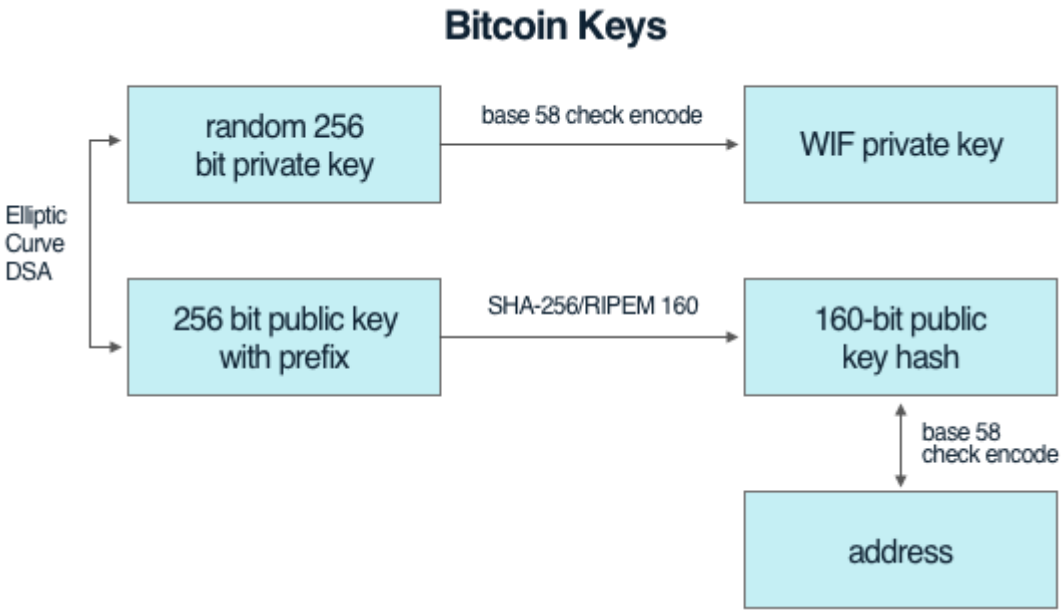
비트코인은 비대칭키 서명 알고리즘인 ECDSA(Elliptic Curve Digital Signature Algorithm)계열의 secp256k1 표준 타원곡선함수(elliptic curve function)를 사용하여 키 쌍을 생성한다. 공개키는 타원곡선함수의 좌표 (x,y)로 표현되는데 과거 비트코인 클라이언트에서는 (x,y)를 사용하였으나 사실 함수(y=function(x))이므로 x 값만으로 공개키를 표현할 수 있다. 이를 압축형 공개키라 부르고 최신 버전은 이를 이용해 주소를 만든다. 이를 pubkey 라고 한다면 비트코인 주소 해시를 구하는 공식은 다음과 같다.

pubkey hash<20bytes/160bits> = RIPEMD160(SHA256(pubkey))

bitcoin address = Base58CheckEncode(pubkey hash)

여기서 RIPEMD160 과 SHA256 은 해시함수이다.

따라서 비트코인 주소는 공개키를 바탕으로 만든다.



그러므로 비트코인 이하 모든 블럭체인의 주소를 생성하는 방법은 다음 같은 절차를 따른다.

256 비트의 난수 개인키를 생성한다. 개인키는 트랜잭션에 서명하고 비트코인을 전송하는데 필요하다. 개인키는 안전하게 보관되지 않으면, 비트코인 도난의 우려가 있다.

타원형 DSA 알고리즘은 개인키로부터 256-비트 공개키를 생성한다. (타원형 암호는 나중에 논의하기로 한다.) 이 공개키는 트랜잭션상의 서명을 확인하는데 사용된다. 불편하게도, 비트코인 프로토콜은 공개키 앞에 04 를 덧붙인다. 공개키는 트랜잭션이 서명되기전까지는 공개되지 않는데, 대부분의 다른 시스템들이 공개키를 공개하는 것과는 다르다.

다음 단계는 비트코인 주소를 생성해서 다른 사람과 공유하는 것이다. 256-비트 공개키는 커서 불편하기 때문에, SHA-256 과 RIPEMD 해시 알고리즘을 사용해서 160 비트로 다운그레이드 한다. 키는 비트코인 커스텀 Base58Check encoding 을 사용해서 아스키로 인코딩된다. 최종결과는 1KKKK6N21XKo48zWKuQKXdvSsCf95ibHFa 와 같은 값이 되고 비트코인을 받기 위해서 사람들이 알려주는 주소가 된다. 그 주소로부터는 공개키와 개인키를 알아낼 수가 없다. 개인키를 잃어버린다면, 비트코인을 잃어버려서 찾을 수 없을 것이다.

마지막으로, Wallet Interchange Format 키(WIF)는 클라이언트 지갑 소프트웨어에 추가하기 위하여 사용된다. 이것은 단순히 개인키를 아스키로 Base58Check 인코딩한 것이고, 쉽게 256-비트 개인키를 추출해낼 수 있다.

요약을 하면, 세가지 타입의 키가 있다: 개인키, 공개키, 공개키의 해쉬, 그리고 Base58Check 인코딩을 사용하여 아스키로 외부에 제공된다. 개인키는 아주 중요한 키이고, 비트코인 사용과 다른 키를 생성하는데 필요하기 때문이다. 공개키 해쉬는 비트코인 주소이다.

WIF 포맷과 주소를 생성하기위해 다음과 같은 코드 manipulation 을 사용하곤 한다. 개인키는 256-비트 난수일 뿐이다. ECDSA 암호 라이브러리가 개인키로부터 공개키를 생성한다. 비트코인 주소는 SHA-256 해싱, RIPEMD-160 해싱, 그리고 Base58 인코딩을 하고 체크섬을 더해서 만들어진다. 최종적으로, 개인키는 Base58Check 로 인코딩해서 개인키를 비트코인 클라이언트 소프트웨어에 넣기 위해서 WIF 인코딩을 생성한다.

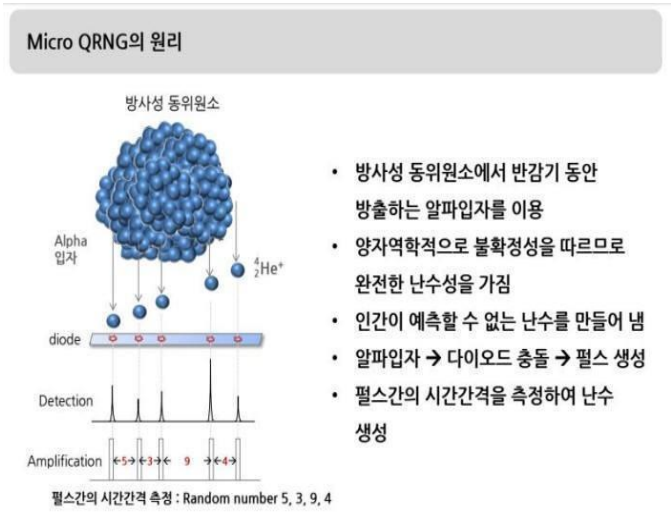
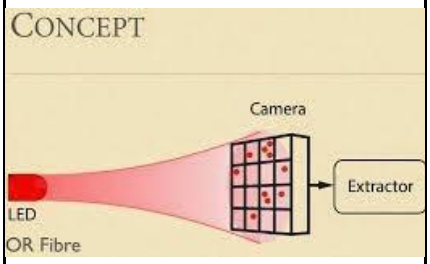
우리는 직관과 어긋나는 원자와 광자세상의 기술을 알고 있다. 개개의 입자단계에서 완전히 새로운 가능성을 가진 양자적 우연성(진정한 난수)의 발견을 알고 있다. 양자적 비국소성은 안전한 비밀키의 분배의 새로운 가능성을 제시하고 있으며, 그것은 비국소적 상관관계에 근거하고 있다. 이것은 얽힘과 양자 비국소성을 이용한 기술이며, 그 원리는 20 세기초에 밝혀진 불확정성의 원리에 기반한다.

$$\Delta x \, \Delta p \geq \frac{\hbar}{2}$$

우리는 다행스럽게도 이기술을 우리의 핵심에 사용할 기회를 가지게 되었으며, SBT 를 지탱하는 Lightnode 와 "SBT for Phone"의 H/W HD Wallet 의 안전한 개인키를 생성하는데 고속의 QRNG 를 적용할 수 있을 것이다.

사용가능한 제품

EYL QRNG

<table>
<tr>
<td> 
</td>
<td>Lightnode 에 적합</td>
</tr>
<tr>
<td colspan="2">SKTELECOM QRNG</td>
</tr>
<tr>
<td> 
</td>
<td> 
</td>
<td>Mobile 에 적합</td>
</tr>
<tr>
<td colspan="2">SWISS IDQ</td>
</tr>
</table>

	Quantis QRNG: USB • 4Mbps of true quantum randomness • Certified by Swiss National Laboratory • USB 2.0 interface • OS Support: Windows, Linux, Solaris, FreeBSD, MAC OS X • Demo application	Lightnode 에 적합
	Quantis QRNG: PCIe 4Mb • 4Mbps of true quantum randomness • PCI Express interface • Certified by Swiss National Laboratory • OS Support: Windows, Linux, Solaris, FreeBSD • Demo application	

	Quantis QRNG: PCIe 16Mb • 16Mbps of true quantum randomness • PCI Express interface • Certified by Swiss National Laboratory • OS Support: Windows, Linux, Solaris, FreeBSD • Demo application	

9.1 게임 DApp 에서의 사용

QRNG 양자난수 생성기를 이용한 공평한 랜덤게임에 사용하기 위한 방법으로 게임 앱이나 타 사이트에서 연동 가능한 api 를 제공한다.

단, 게임 앱을 연동하는 것은 가능하나 게임에 필요한 난수를 타사이트에서 직접 넘겨 받는 것은 불가하게 해야 한다. (직접 넘겨받아 확률 조작 및 게임복제 방지)

연동 게임 앱을 통해서 게임결과 값을 넘겨받아 처리하는 것은 가능하게 제공한다.

9.2 블록체인 OTP 사용

EVM 스마트컨트랙트 코드를 view 에서 QRNG 를 읽어서 OTP 를 제공할 수 있게 한다.

otp 등록을 위한 secret key 는 스마트 컨트랙트의 mapping 스토리지 변수(블록체인내의 저장)에 저장을 하여야 한다.

OTP 스마트컨트랙트 코드를 라이브러리 코드로 임포트하여 스마트 컨트랙트를 작성하면, 출금이나, 입금 시, 기타 보안이 중요시되는 함수 호출 등에 OTP 코드를 입력 받고 해당 OTP 코드가 검증이 맞을 때만 출금, 입금이 가능하게 활용한다.

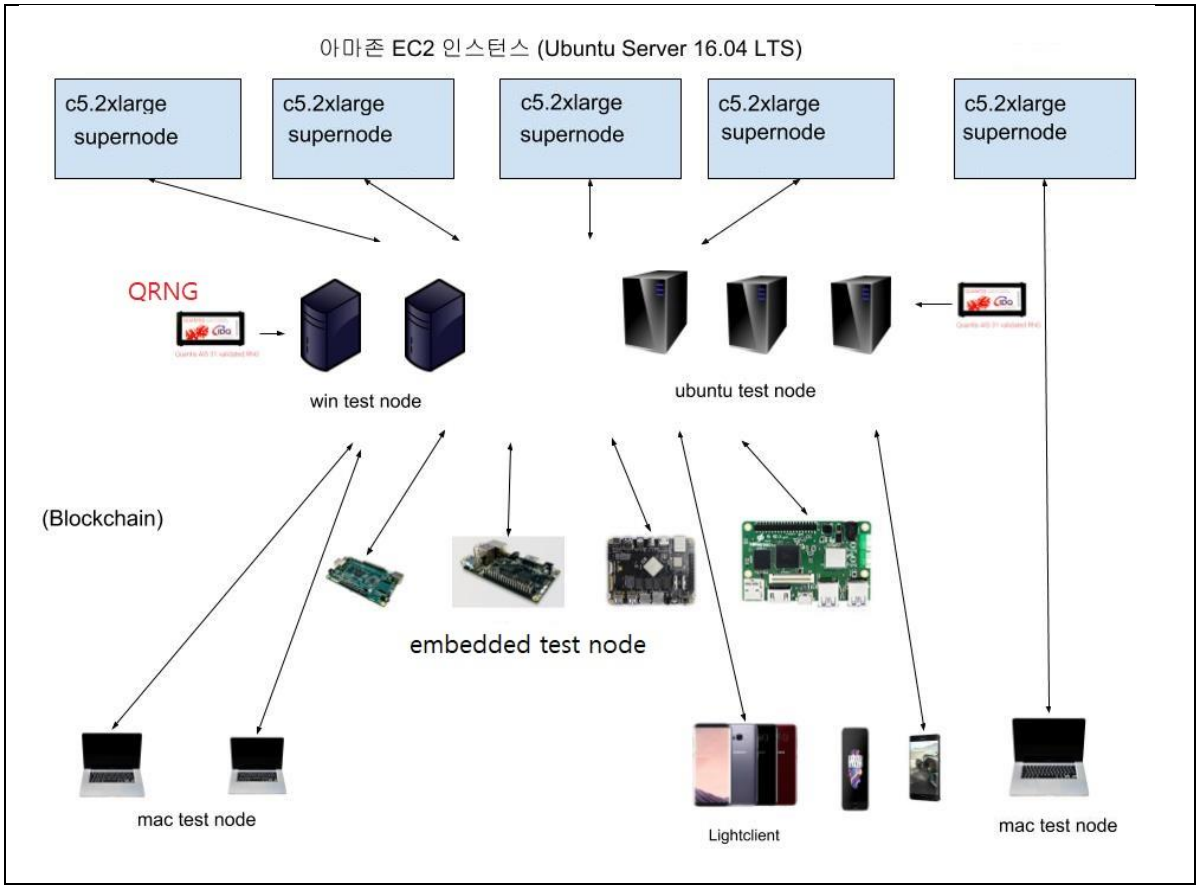
9.3 Randomness Beacon

QRNG 로부터 추출된 True Random number 를 효율적으로 사용하기 위해서는 별도의 비컨체인에서 난수를 JSON-RPC API 와 Schema 로 broadcast 해주는 것이 더욱 효율적일 것으로 판단되어 충분한 실험과 검증을 통해서 구현될 것이다.

10 Conclusion

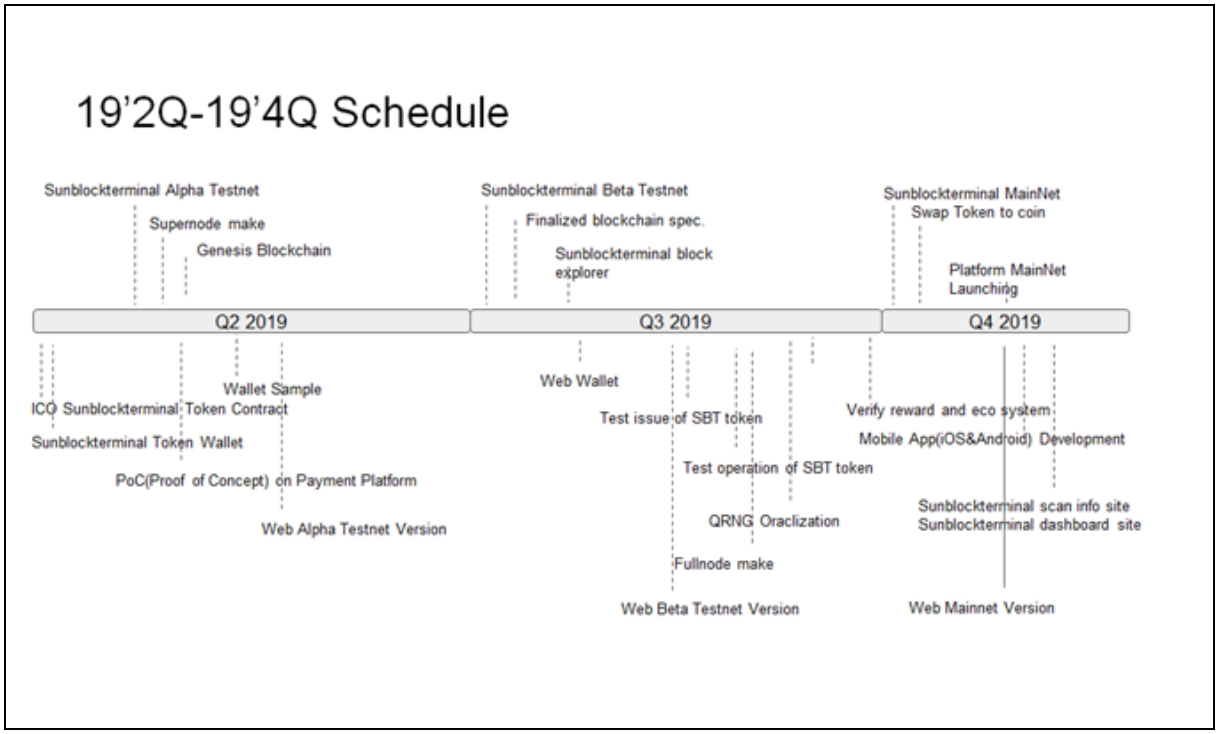
10.1Sunblockterminal.Testnet 구성도

Testnet Alpha

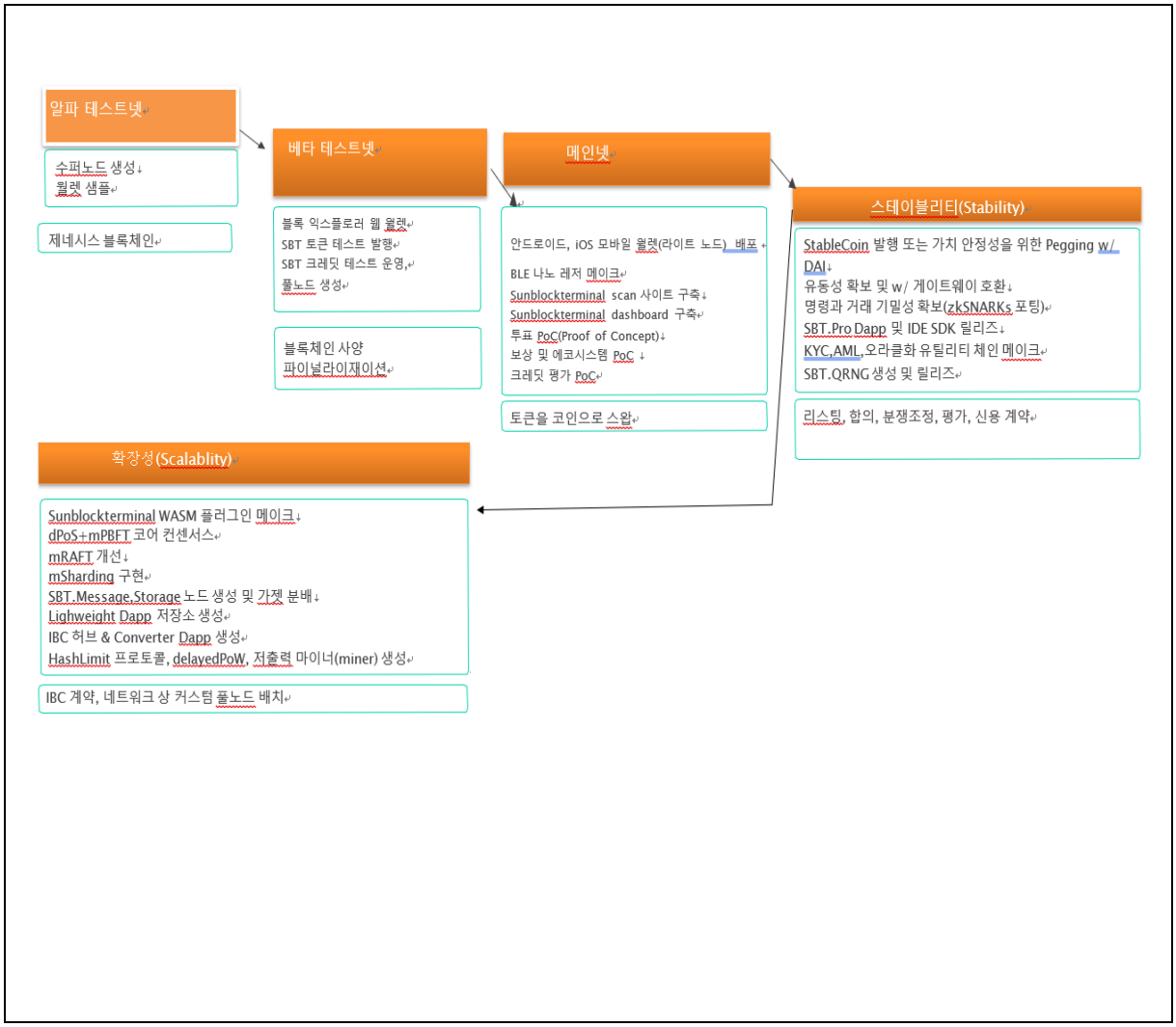


	Lightnode	Fullnode	Lightnode
Testnet Beta	~1	3	10
Lightnet	~2	5	20

10.2Schedule



10.3Sunblockterminal.Roadmap



10.4Grand Lightnet Launching

Sunblockterminal 은 그랜드 메인넷 론칭 시 블록체인 기반 DApp Platform 서비스를 목표로 하고 있으며, 이에 최적화된 네트웍을 구축할 것이다.

블록체인상에 한 명의 유저를 위한 계정의 RAM 사용량은 4KBytes 이다.

4Kbytes * 1 * 10^8 ≒ 4TBytes

1 억명의 사용자 후보에 대한 DB 로 계산했을 때이다. 향후 사용자가 늘어남에 따라서 4TB,16TB 의 용량을 갖는 최적의 Lightnode 를 개발할 계획을 가지고 있다.

Reference Books

[1] https://github.com/bitcoinbook/bitcoinbook Andreas M. Antonopoulos. Mastering Bitcoin. O'Reilly Media, Inc., 2010.

[2] https://forum.ethereum.org/discussion/46/total-supply-of-eth

[3] Sunny King and Scott Nadal. 2012. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. (August 2012).

http://peercoin.net/assets/paper/peercoin-paper.pdf.

[4] Jae Kwon. 2014. Tendermint: Consensus without Mining. (2014). http://tendermint.com/docs/tendermint.pdf.

[5] https://www.stellar.org/papers/stellar-consensus-protocol.pdf

[6] https://github.com/input-output-hk/Scorex

[7] https://theethereum.wiki/w/index.php/ERC20_Token_Standard#Sample_Fixed_Supply_Token_Contract

[8] https://github.com/Giveth/minime/blob/master/contracts/MiniMeToken.sol

[9] https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md EOS.IO Technical White Paper v2

[10] Nicolas Gising. Quantum Chance: Nonlocality, Teleportation and Other Quantum Marvels. Copernicus, 2014.

[11] https://ubports.com/ UbuntuTouch: A Mobile Version of the Ubuntu Operating System

[12] david J. Stang and Sylvia Moon, Network Security Secrets, IDG Books Worldwide, Inc., 1993.

[13] S. Bellovin and M. Merritt, "Limitations of the Kerberos Authentication System," Computer Communications Review, October 1997.

[14] Ray Bird et al., "Systematic Design of a Family of Attack Resistant Authentication Protocols," IEEE Journal on Selected Areas in Communications, Vol. 11, No. 5, June 1993.

[15] eric-maxwell-mvc-mvp-and-mvvm-on-android

https://academy.realm.io/kr/posts/eric-maxwell-mvc-mvp-and-mvvm-on-android/

[16] gson

https://github.com/google/gson

[17] Retrofit

http://square.github.io/retrofit/

[18] Event Bus

https://github.com/greenrobot/EventBus

[19] **Learning Solidity Part 2: Commit-Reveal Voting**

https://karl.tech/learning-solidity-part-2-voting/

[20] Partial Lock Commit Reveal Voting System that utilizes ERC20 Tokens

https://github.com/ConsenSys/PLCRVoting