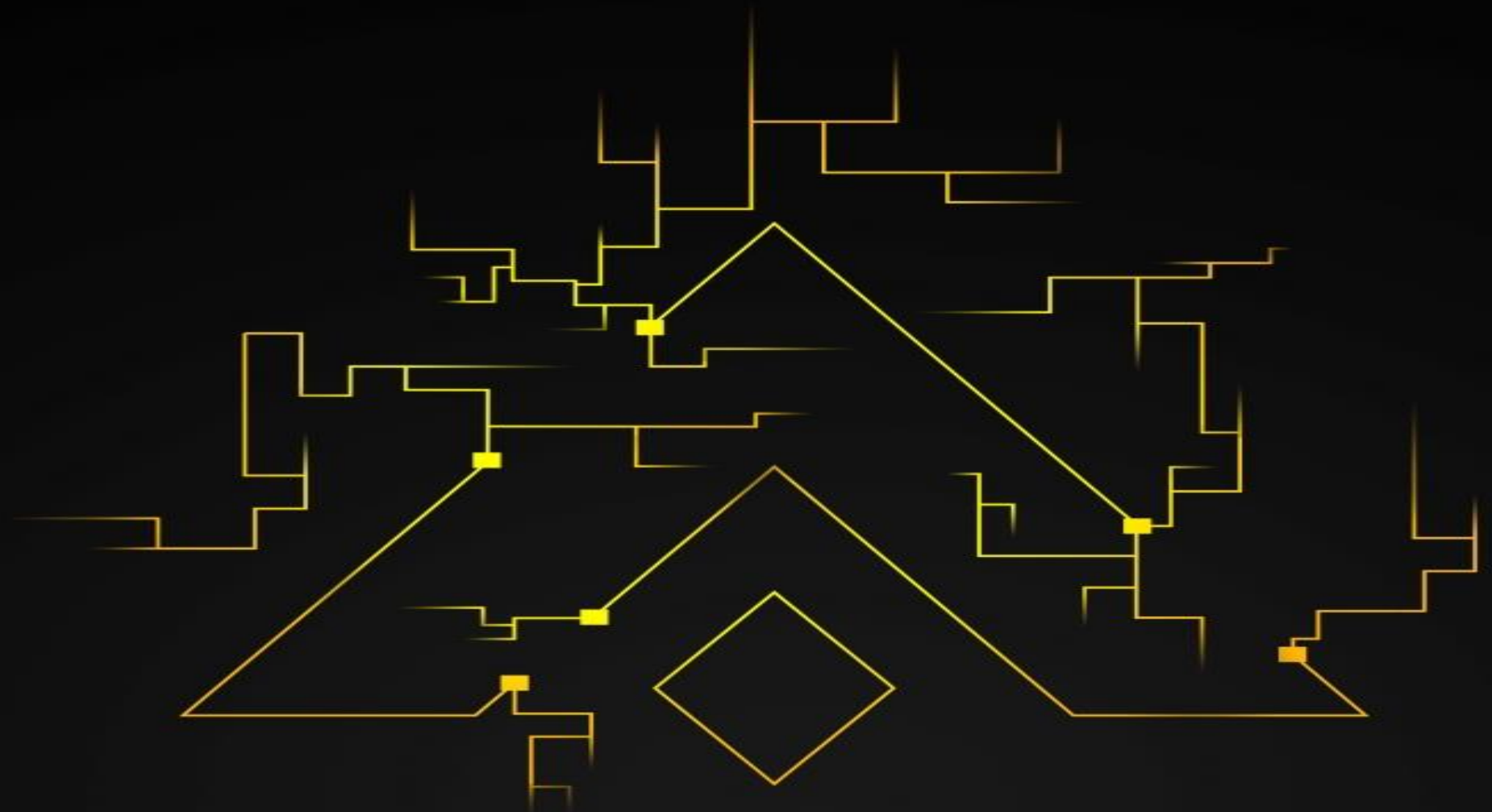
AOS

AOS White Paper

AOS Makes Encryption Assets Safer



암호화 자산을 더 안전하게 만드는 AOS
익명 운영 체제 (Anonymous Operating System)

AOS 가 필요한 이유는?

블록체인의 발전을 살펴보면 몇 가지 중요한 이정표를 발견할 수 있고 발전을 가능하게 한 과학 기술을 살펴보면 미래 트렌드도 예측할 수 있습니다. 2009 년 탄생한 비트코인은 모든 거래를 보장하고 기록하기 위해 전체 P2P 네트워크에 수많은 노드로 이뤄진 분산 데이터베이스를 사용해 창의적인 방식으로 탈중앙화 결제 시스템을 제시했으며 통화 순환의 안전을 보장하기 위해 암호화된 디자인을 사용했습니다. 비트코인은 암호화폐 시대를 열어갔으며 비트코인의 가격은 2017 년 $19,850 를 기록하며 정점에 도달했습니다.

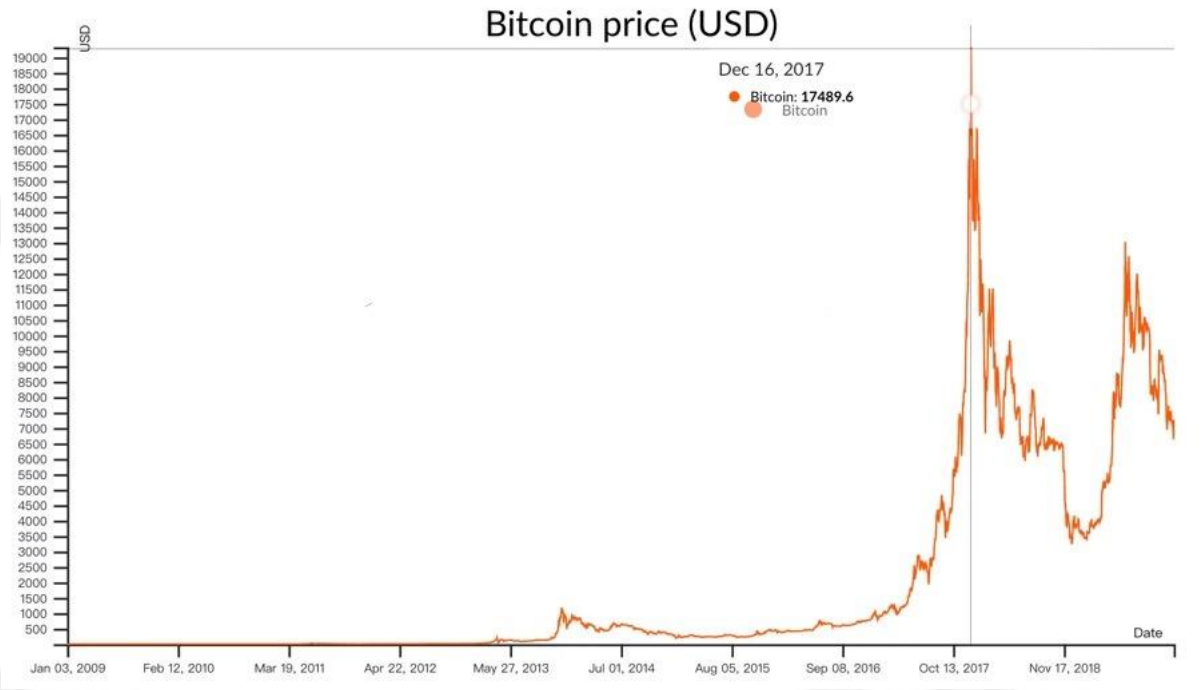


그림 1. 비트코인 가격 차트

(출처: https://blockchair.com/bitcoin/price)

하지만 비트코인은 완벽하지 않으며 그 결함 중 하나는 프로토콜 확장성입니다. 예를 들어 비트코인 네트워크에는 bitcoin 이라는 하나의 기호만 있습니다. 유저들은 다른 기호로 커스터마이징을 할 수 없습니다. 만약 다른 기호를 사용할 수 있다면 기업의 주식, 부채 증명서 등을 해당 기호를 통해 나타낼 수 있는데 이런 기능이 없습니다. 또한 비트코인 프로토콜에는 스택 기반 스크립팅 언어가 사용되어 어느 정도의 유연성을 제공하고 다중 서명 기능을 실현할 수 있지만 분산 트랜잭션과 같은 고급 애플리케이션을 구축하기에는 부족합니다. 2013 년 발행된 백서에

이어 이더리움은 비트코인이 가진 한계점을 해결하도록 설계되어 출시되었습니다. 이더리움은 내부 튜링 컴플리트(turing complete) 스크립팅 언어를 사용해 유저들이 정확하게 본인이 정의할 수 있는 스마트 컨트랙트 또는 거래 유형을 구축할 수 있도록 하였습니다. 스마트 컨트랙트의 개념을 창시한 이더리움은 시장에서 널리 알려졌고 ETH 의 가격은 $1,428 에 달했습니다.

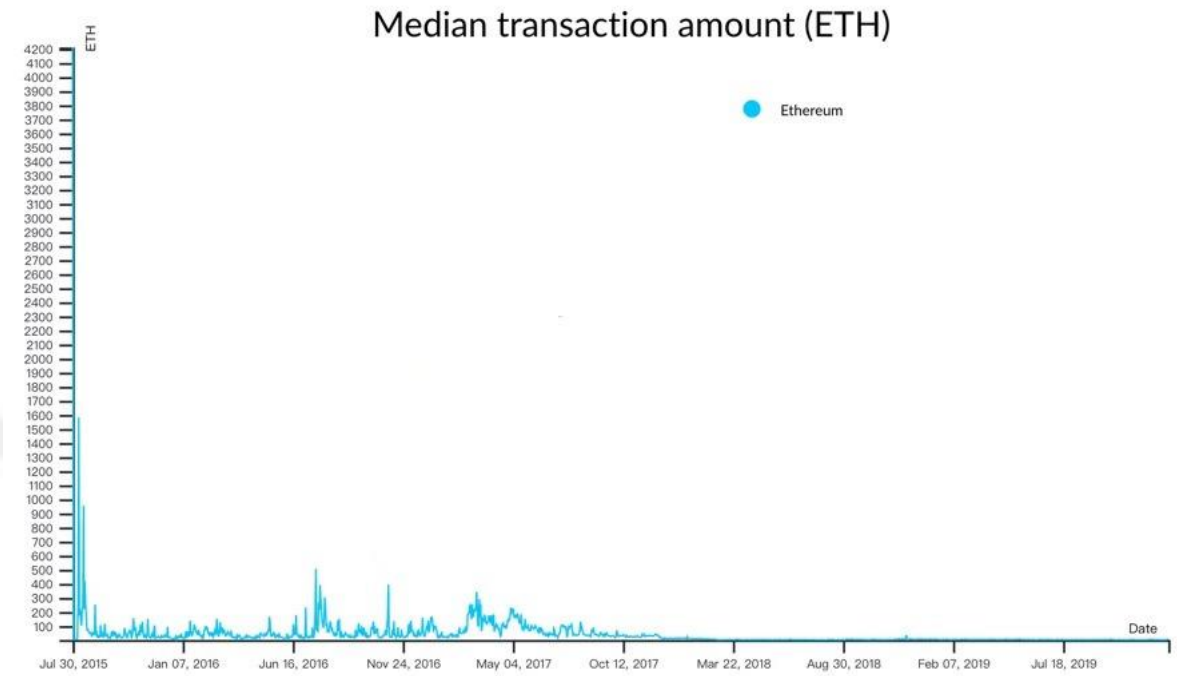


그림 2. 이더리움 거래량 차트

(출처: https://blockchair.com/ethereum/charts/average-transaction-amount-eth)

이더리움의 시초는 디앱(DApp) 생태를 구축하기 위한 세계를 위한 범용 컴퓨터를 만들기 위함이었습니다. 이런 탈중앙화를 우선시하는 설계는 확장성과 기능을 어느정도 희생해야 합니다. 그러나 고병행성 애플리케이션을 운영하기엔 이더리움의 성능이 부족해 본 역할을 수행하기 어렵습니다. 이더리움의 성능 문제를 고려해 EOS 는 처음부터 성능과 확장성을 우선시하면서 탈중앙화 수준을 어느정도 타협했습니다. EOS 설계 원칙은 퍼블릭 블록체인 플랫폼의 비즈니스 운영 로직과 매우 일치하며 핵심 기술 메커니즘은 이미 실제로 널리 입증이 되었고 이는 블록체인 기술의 근본적인 발전으로 이어지고 있습니다. EOS 성능이 디앱 구축 단계에서 개발자의 요구를 충족시킬 수 있는 경우에만 이를 능가할 터닝 포인트가 될 것입니다. EOS 의 가격은 $22 로 정점에 도달했습니다.

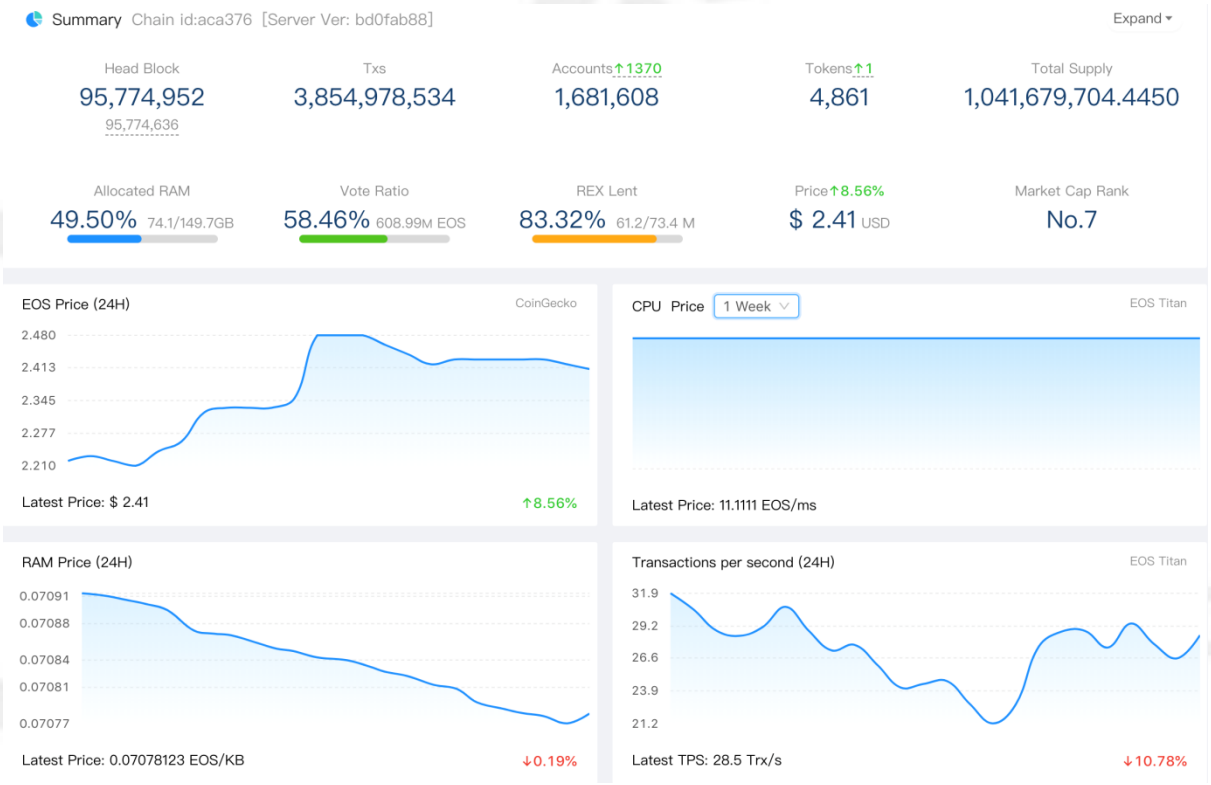


그림 3. EOS 써머리 차트

(출처: https://eospark.com/)

최근 몇 년 간 비트코인, ETH 및 EOS 는 각자 다른 기간에 걸쳐 업계의 발전을 이끌었습니다. 블록체인 산업의 발전과 함께 시장은 단순한 암호화 결제에서부터 스마트 컨트랙트 및 고성능 스마트 컨트랙트 플랫폼을 필요로 하게 되었습니다. 전세계 블록 기술에 대한 인식이 높아짐에 따라 유저 커뮤니티도 기존의 기술 괴짜(tech

geek)로부터 일반 인터넷 유저로 확대되었습니다. 인터넷은 정보 전송을 크게 가속화했고 사용자는 많은 양의 데이터를 생산했으며 인터넷 기업은 수많은 유저 데이터를 축적했습니다. 그러나 데이터는 쉽게 훔칠 수 있어 많은 문제가 발생하고 심지어 재산 피해까지 발생하기도 했습니다. 플랫폼에 충분한 데이터 보안 관련 보호 메커니즘이 없어서 인터넷 응답 시나리오에서 개인 데이터가 도난당하는 경우도 많았습니다. 실제 기업의 경제 활동에서도 개인정보 보호가 부족해 탈중앙화 스마트 컨트랙트의 보급에 큰 장애물이 되었습니다. 사용자들은 기업 공급망 관리, 금융 거래, 비밀 경매, 소셜 또는 온라인 쇼핑, 기밀정보와 같은 것에 대한 프로세스를 기록해야 한다고 생각했습니다. 일반 사용자의 개인 정보 보호가 특히나 중요해졌습니다. 코인베이스(Coinbase)의 CEO 인 브라이언 암스트롱(Brian Armstrong)은 트위터에서 "기본적으로 개인 거래를 지원하는 확장 가능하고 충분히 탈중앙화된 체인(프라이버시 코인)이 게임 체인저가 될 것"이라고 언급했습니다.

이러한 이유 때문에 **높은 성능과 높은 확장성을 유지하면서 결제 프라이버시 기능을 추가한 스마트 컨트랙트 플랫폼인 AOS 를 제안합니다. 차세대 프라이버시 퍼블릭 체인으로서 익명 EOS 라고 알려진 AOS 는 스마트 컨트랙트를 지원하고 익명 자산을 발행할 수 있습니다.** AOS 는 비트코인, ETH, EOS 에 이어 프라이버시 컴퓨팅 및 자산 교환 분야에 있어 블록체인 산업의 다음 발전 단계를 대표할 것입니다.

AOS 의 비전

익명 운영 체제인 AOS(anonymous operating system)는 유저 프라이버시에 중점을 둔 디앱을 위해 설계된 블록체인 운영 체제입니다.

AOS 는 높은 성능 및 확장성을 유지하면서 유저 프라이버시를 효과적으로 보호하는 블록체인 운영 체제가 되길 희망합니다.

인터넷이 처음 생겨난 이후 세계화, 자율성, 분배, 개방성, 투명성, 동등성, 익명성, 공유 등 이를 수식하는 수많은 구호들이 나왔고 마치 유토피아를 연상시키게 하는 이 구호들 덕분에 인터넷이 더 급속도로 발전할 수 있었습니다. 하지만 인터넷의 현재 패턴을 살펴보면 인터넷은 과두적(oligarchic)이며 폐쇄되고 있으며 유저들은 권리를 박탈당하고 있습니다. 유저들은 인터넷의 거버넌스에 참여하지도 못하며 그들의 프라이버시도 효과적으로 보호하지 못하고 있습니다. 왜 이런 상황이 초래된 것일까요?

인터넷은 발전 초기 단계에서 배포되었습니다. 하지만 그 당시 기술적 역량이 제한 되어있었으며 모든 사람들이 서버의 역할을 수행할 수 있는 것이 아니기 때문에 결과적으로 중앙화 된 서비스 공급자가 나타나게 되었습니다. AOS 는 이런 중앙 집중화된 서비스 제공자들이 경제적 이익을 추구하는 과정에서 인터넷 창시 의도에 반하는 방식으로 변모했다고 생각합니다. 그들은 이익을 얻는 대가로 경쟁자들을 공격하고 업계를 독점하며 사용자들의 개인 데이터를 팔았습니다. 클라우드 컴퓨팅, 빅데이터 및 인공지능과 같은 새로운 기술이 자유로운 인터넷을

파괴하고 있었습니다. 사용자는 더 이상 유저가 아니라 분석된 데이터의 일부분일 뿐이며 인터넷 거인들이 이끄는 대로 따라가는 순한 양이 되어버렸습니다.

비트코인이 대표하는 블록체인 기술은 세계화, 자율성, 분배, 개방성, 투명성, 동등성, 익명성, 공유 등과 같은 구호의 중요성을 이어갑니다. 하지만 한가지 의문이 들지 않을 수 없습니다. 바로 블록체인이 인터넷에서 있었던 실수를 반복하지 못하게 하려면 어떻게 해야 할까요? 블록체인이 중앙 관리자가 되고 소수의 집권자들이 일반 유저들을 노예화 시키는 것을 어떻게 방지할 수 있을까요?

첫째, 이를 위해서는 블록체인의 고성능 및 높은 확장성이라는 기술적 특성은 유지해야 합니다. 만약 블록체인이 고성능을 유지할 수 없다면 초창기 인터넷 시대처럼 중앙 서비스가 되어 소수의 몇 명이 통제하게 될 것입니다. 기본 기술의 발전으로 인해 네트워크, 컴퓨팅 및 저장소는 더 이상 소수의 집권자만을 위한 독점적인 자원이 아닙니다. 점점 더 많은 사람들이 소수의 집권자들과 동일한 생산 및 생산성을 갖추게 될 것입니다. 확장성 및 성능이 뛰어난 블록체인 시스템은 이러한 사람들을 연결할 것이며 전세계 유저들에게 서비스를 제공할 것입니다. 따라서 생산 및 생산성을 위한 수단은 더 이상 소수를 위한 특권이 아닙니다. 전세계의 현명한 사람들은 독점을 하는 것이 아닌 사용자에게 더 나은 서비스를 제공하는 방식에 대해 생각하기 시작할 것입니다.

둘째, 유저의 프라이버시가 우선시되어야 합니다. 개인 정보에 대한 통제를 직접 할 수 없기 때문에 유저들은 인터넷의 노예가 되고 무의식적으로 소수의 집권자들에게 임의로 통제를 받게 됩니다. 소수의 집권자들은 유저들의 개인 정보를 보호해주겠다는 몇 가지 약속을 했지만 프라이버시 보호를 위해 노력했다는 증거도 없고 개인 정보가 어떤 용도로 사용되는지 사용자들은 모르기 때문에 약속은 지켜지지 않았습니다.

따라서 AOS 는 세계화, 자율성, 분배, 개방성, 투명성, 동등성, 익명성, 공유의 개념을 실현하기 위해 높은 성능 및 확장성을 가지고 사용자 개인 정보를 실질적으로 보호할 수 있는 블록체인 시스템을 구축하기를 희망합니다. 이런 플랫폼만이 인터넷 거인들의 통제를 원하지 않는 전세계 사용자들 편에서 함께 할 수 있습니다.

AOS 기술 구조 설계 및 원리 구현

AOS 는 사용자 및 디앱의 익명성과 개인 정보에 더 많은 관심을 기울일 것입니다. 사용자와 디앱의 익명성 또는 개인정보는 ETH 및 EOS 에서는 보호되지 않습니다. AOS 는 이 사실이 사회 발전의 객관적인 사실에 위배된다고 생각하며 이는 AOS 가 출시된 이유이기도 합니다. 블록체인의 미래는 사용자의 자산 및 개인 정보 보호에 달려 있다고 생각합니다.

알려진 바로는 Monero, Zcash, Grin 등을 포함한 익명 기능을 제공하는 모든 블록체인은 UTXO 모델을 도입했다고 합니다. 이러한 블록체인은 스마트 컨트랙트 기능을 지원하지 않으므로 디앱의 발전에 한계가 있을 수밖에 없고 심지어 익명 자산을 발행하는 기본적인 기능 또한 제공하지 않습니다.

AOS 의 비전은 디앱을 위한 운영 체제 기반을 제공하고 개인 정보 보호 기능을 도입하는 것입니다. 개발자들은 AOS 에서 개인 정보 보호 기능에 중점을 두면서 디앱을 빠르게 개발할 수 있을 것이다. 동시에 AOS 는 익명 자산 발행 기능을 제공해 모든 사용자가 자신의 익명 자산을 발행할 수 있도록 합니다. 후속 자산 거래에서는 참여자 외에는 자산의 규모를 알 수 없을 것입니다.

AOS 는 계정 모델, 동종 암호화(homomorphic encryption) 및 영지식증명(zero knowledge proof)기술을 사용해 익명성에 대한 기본 암호화 기능을 서포트 할 것입니다. 이런 암호화 기술은 추후 스마트 컨트랙트 레이어에 사용될 수 있으며 디앱이 익명 기능을 보다 쉽게 제공할 수 있도록 합니다.

a) Account model

AOS will select the account model to manage users.

AOS 가 사용자들을 관리하는 방식으로 채택한 방법은 계정 모델입니다. 비트코인에서 사용하는 UTXO 모델은 자산의 분할성과 추적성을 제공하며 중복 검사를 통해 "이중 지출"을 방지합니다. 또한 블록 처리 프로세스 동안 각 트랜잭션 ("이중 지출"이 없는 트랜잭션)을 동시에 처리하므로 블록체인의 효율성이 크게 향상됩니다. 따라서 주로 자산과 거래를 기반으로 하는 여러 블록체인은 UTXO 모델을 채택했습니다.

하지만 UTXO 모델은 굉장히 잘 알려진 단점이 있다. 스마트 컨트랙트를 서포트 하는 기능이 충분히 사용자 친화적이지는 않다는 점입니다. 그러나 자산을 이전하는 트랜잭션과는 다르게 스마트 컨트랙트는 사용자 간의 계약이고 계정을 본체로 사용하는 것이 컨트랙트 개발 및 유지에 더 도움이 됩니다.

AOS 계정 모델을 설계할 때 다음과 같은 사항을 고려했습니다. 순수 자산 거래는 현대 비즈니스 시나리오의 니즈를 충족시키지 못합니다. 또한 단순한 비즈니스를 위한 니즈 이상을 충족시키면서 개인정보 및 익명성에 대한 요구사항도 있습니다. 스마트 컨트랙트는 이 격차를 메꿀 수 있으며 컨트랙트에서 다양한 요구사항을 실행할 수 있습니다.

해당 설계에서 적용된 원리는 다음과 같이 실행될 수 있습니다.

표 1. 계정 모델 설계 법

Permission	**Weight : Keys**	**Threshold**
Owner	1 : EOS2abcd... 1 : EOS1efgh...	2
Active	1 : EOS3 abcd... 1 : EOS4 efgh...	2

계정을 생성하면 기본적으로 계정에 @owner와 @active 두가지 권한이 포함됩니다. @owner은 주로 @active에 해당하는 키 수정, @publish 권한 만들기 또는 삭제 등 다른 권한 변경을 관리하는데 사용됩니다. @active 권한은 주로 전송과 같은 컨트랙트 운영을 수행하는데 사용됩니다. 즉, EOS를 전송하려면 @active에서 키(Keys)로 거래에 서명을 해야 합니다.

권한과 키 외에도 weight 과 임계치(threshold)가 있습니다. 이체를 예시로 둔다면 위 표의 구성에 따라 Alice 의 계정으로 이체를 하려면 키의 weight 를 추가해야 합니다. 결과가 도메인 값보다 작지 않으면 트랜잭션은 합법적인 것으로 간주됩니다.
eos3 과 eos4 에 해당하는 개인 키가 트랜잭션에 서명해야 하는 경우 해당 weight 합계는 도메인 값 2 보다 작아야 조건을 충족시킬 수 있습니다. 마찬가지로 @active 키를 수정하려면 eos2 및 eos1 에 해당하는 키를 사용해 동시에 작업해서 서명해야 합니다. 위의 예시를 보면 계정을 관리하기 위해 충분히 안전하게 계정을 관리할 수 있는 숫자인 4 쌍의 키를 사용했습니다.

다수의 블록체인 프로젝트는 UTXO+계정 모델이라는 이중 구조를 도입해 다음과 같은 각 모델의 장점을 잘 활용하길 원합니다. UTXO 차원에는 자산의 익명성을 실현시키고 계정 모델 차원에서는 스마트 컨트랙트를 구현하기를 원합니다. 이런 훌륭한 아이디어에도 불구하고 사실 실현하기 불가능 한 부분이 있습니다. 바로 모든 것이 멀티 기능으로 출시될 시 모든 기능이 각자 다 제대로 작동할 수 없다는 것입니다. 예를 들어 스위스 세이버(Swiss Sabre)를 생각해보면 많은 기능을 갖춘 것 같지만 그 중 제대로 쓸 수 있는 기능은 없습니다.

AOS 는 계정 모델에 집중하기로 결정했습니다. 한 편에서는 AOS 기술은 자산의 익명화를 포함한 스마트 컨트랙트 레이어를 위한 익명성 기능을 제공하고 있습니다. 다른 편에서 계정 모델을 적용하는 장점은 실제 비즈니스 시나리오에서 증명이 되었다는 점입니다.

b) 동형암호(Homomorphic encryption)

동형암호는 암호화된 상태에서 연산을 할 수 있는 차세대 암호화 기술입니다.

데이터 암호화는 데이터 개인 정보를 보호하는 일반적인 방법입니다. 계좌의 정보를 보호하는 것과 같은 예시가 있습니다. 그러나 일반적인 암호화 알고리즘은 암호 텍스트 상태의 데이터를 계산할 수는 없습니다. 동형암호를 사용하는 경우 계좌 정보, 정보 전송 및 수집을 암호화해 거래 정보의 개인 정보를 보호할 수 있습니다.

사용자는 데이터가 암호화된 경우에만 블록체인에 데이터를 넣는 것에 대해 완전히 확신을 가질 수 있으며 데이터 서비스 제공자들은 데이터를 암호화하는 동형암호 기술을 사용해야만 사용자가 원하는 서비스를 제공할 수 있습니다. 따라서 동형암호는 현재 데이터 프라이버시와 서비스 제공 간 의견 불일치를 해결하는 최상의 솔루션입니다.
AOS 는 다양한 비즈니스 시나리오에 맞춰 사용자들이 선택할 수 있도록 다양한 동형암호 체계를 제공할 것입니다.
예비 단계에서는 기존 기술을 기반으로 곱셈동형암호 기반 체계를 사용할 것입니다. 실행 방식은 다음과 같습니다.

우선, C1, C2 가 있다고 가정해봅시다.

$$CT = (C_1, C_2) = (g^r, h^r \cdot m) \qquad (1)$$

이 공식에서 r 은 암호화 과정에서 임의로 선택한 숫자, g 는 생성기, h 는 공개 키입니다. 암호문이 두 개인 경우:

$$CT_1 = (g^{r_1}, h^{r_1} \cdot m_1), CT_2 = (g^{r_2}, h^{r_2} \cdot m_2) \quad (2)$$

이 두 암호문의 첫 번째 부분과 두 번째 부분을 따로 곱한다면 다음과 같은 공식이 나옵니다.

$$CT = (g^{r_1} \cdot g^{r_2}, h^{r_1} \cdot m_1 \cdot h^{r_2} \cdot m_2) = (g^{r_1+r_2}, h^{r_1+r_2} \cdot m_1 m_2) \quad (3)$$

즉, 곱셈 후 암호 텍스트는 m1m2 에 정확히 해당하는 부분의 암호입니다. 이 경우 사용자는 m1, m2 의 결과를 해독합니다.

이를 처리하는 동안에도 AOS 는 리소스에 지속적으로 투자해 동형 암호 라이브러리인 HElib 을 최적화하고 키 생성에 필요한 시간을 단축시키며, 더 많은 시나리오에 보다 실용적인 동형암호 알고리즘을 제공하기 위해 노력할 것입니다

c) 프록시 재암호화(Proxy re-encryption, PRE)

현재 낯선 사람들 사이 신뢰 거래를 구축하는데 더 나은 방법은 프록시 재 암호화입니다. 중개자 또는 에이전트의 역할을 하는 프록시는 두 사용자 간에 메시지를 전달하거나 메시지를 수정하고 실제 메시지를 다른 사람에게 전달하는 역할을 합니다. 메시지의 진위 여부를 확인할 수는 있지만 내용을 알 수는 없습니다.

프록시 재 암호화는 다음과 같이 사용됩니다. 예를 들어 Bob 이 공개 키로 암호화된 메시지의 내용을 개인 키가 아닌 제 3 자 Chris 에게 공개하려는 경우, Bob 은 프록시 재 암호화를 사용할 수 있습니다. Bob 은 에이전트가 메시지의 내용을 읽을 수 없도록 하고 Chris 에게 보낼 메시지를 다시 암호화하기 위한 프록시를 지정할 수 있습니다. Chris 가 메시지를 해독하는데 사용할 수 있는 새로운 키가 생성이 됩니다. 이제 Bob 이 Bob 의 키로 암호화된 메시지를 Chris 에게 보내면 에이전트가 Chris 가 메시지를 해독할 수 있도록 메시지를 변경합니다. 이 방법을 사용하면 이메일 전달, 법 집행 모니터링 및 컨텐츠 배포 등과 같은 다양한 응용이 가능합니다.

약한 재암호화 체계는 에이전트가 양쪽의 키를 동시에 보유하는 방식으로 하나의 키는 하나의 일반 텍스트를 해독하고 다른 하나는 이를 암호화하는 것입니다. 대부분의 경우 프록시 재암호화 체계의 목표는 키 또는 기본 텍스트를 프록시에게 공개하지 않는 것이므로 이 방법은 적합하지 않습니다.

c) 영지식증명(Zero knowledge proof)

단순한 기술 이상인 영지식증명은 무수한 사람들의 직관적인 이해를 변화시킵니다.

많은 사람들이 증거를 "신뢰할 수 있는 것을 보여주는 것"과 "데이터 문제 증명"이라고 이해합니다. 이런 증명 과정에서 정보 유출이 일어난다는 것은 의심의 여지가 없습니다. 당신의 수학 선생님이 수학 문제에 대한 증거를 보여준다면 당신은 그 지식을 얻게 됩니다. 그런 시나리오가 영지식증명의 예시입니다. 즉, 제시한 정보에는 어떤 제안의 정확성을 증명하는데 필요한 정보 외에는 다른 정보가 포함되지 않습니다.

종종 컨트랙트에는 "컴퓨터 비밀번호를 해독 완료", "지정한 작업 완료" 등과 같은 표현이 나타납니다. 실제로 어떻게 컴퓨터 비밀번호를 해독했는지, 어떤 특정 작업이 완료되었는지에 대해서는 다루지 않아도 된다는 점이 영지식증명의 매력입니다. 지식에는 개인 정보도 필요한데 이 방식은 양당사자의 개인 정보를 모두 보호합니다.

영지식증명의 자세한 이론적 설계와 관련, AOS 가 잠정적으로 제안하는 AOS 영지식증명을 이용한 거래 방식은 다음과 같습니다.

1) Alice 가 Bob 에게 AOS 영지식증명을 통해 거래 증명을 줄 때, Alice 는 두 개의 파라미터를 제공해야 합니다: Pub_Key(공개 키) 와 Prv_Key(개인 키), 그리고 AOS 로부터 주어진 컨트랙트 상의 논리적 프로세싱을 통해 영지식증거 AOSP_π 를 생성하게 됩니다. 즉, Alice 는 토큰 ID 라고 불리는 해시 토큰 α 의 값, Alice 의 공개 키, 그리고 32 비트의 무작위 값(수용 토큰의 유일성을 제공하기 위해 사용)을 제공하여 AOS 영지식증명 중개 토큰 자산 Z_A 를 생성하게 됩니다.

$$Z_A = H(\alpha|pk_A|\sigma) \quad (4)$$

상기 방식을 통해 Alice 가 두 개의 AOS 영지식증명 중개 토큰 자산을 생성한다고 가정했을 때, 두 개의 AOS 영지식을 획득하게 되어 연결과 해싱 및 Alice 의 공개 키인 pk_A 를 통해 Z′A 와 Z′′A 값이 각각 10α and 5α 라는 것을 증명할 수 있습니다. 뿐만 아니라, 무작위 숫자인 σ_1 과 σ_2 는 acceptance type tong assets 와도 관련되어 있다는 것도 증명할 수 있습니다.

Z′_A 와 Z′′_A 는 각각 10α 와 5α 이며, Alice 는 Bob 에게 이를 전송하고자 합니다.

$$Z'_A = H(10\alpha|pk_A|\sigma_1) \quad (5)$$

Alice 는 본인의 AOS 컨트랙트 라인(contract line)에 하나의 영지식증거 AOSP_π 를 생성하고 이를 AOS 주소로부터 AOS 영지식증거 허브(Hub) 컨트랙트에 제출합니다.

2) AOS 영지식증거에서, 중개 토큰 자산은 수용 유형 머클 트리에 저장되는데, 이는 AOS 영지식증거의 머클트리 내에서 이와 같은 AOS 영지식증거 토큰 자산이 존재하는지 보여주는 데 사용되며, 어떠한 AOS 영지식증거 토큰 자산이 사용되고 있는 지 나타냅니다. 오리지널 프로세스와 반대로 진행해보면, Alice 는 리프 노드(leaf node), 즉 증명된 Z′A 와 Z′′A 에서 출발하여 반복 해싱(트리의 깊이 - 1 회)을 통해 AOS 영지식 증거 자산의 머클트리 루트 R 을 생성할 수 있다. 이는 각 레벨에서 형제 노드(siblings)과의 반복적인 연결 및 해싱을 통해 최종적으로 이뤄집니다.

$$
\begin{aligned}
H'_1 &= H(Z'_A \mid sibling\ of\ Z'_A) \\
H'_2 &= H(H'_1 \mid sibling\ of\ H'_1) \\
&\vdots \\
R &= H(H'_{31} \mid sibling\ of\ H'_{31}) \\
H''_1 &= H(Z''_A \mid sibling\ of\ Z''_A) \\
H''_2 &= H(H''_1 \mid sibling\ of\ H''_1) \\
&\vdots
\end{aligned}
\qquad (6)
$$

3) Alice 는 본인의 개인키 sk_A 와 그에 해당하는 무작위 수인 σ_1 과 σ_2 를 사용해 Z'A 에 해당하는 디스트럭터(destructors) N10α 와 N5α 를 생성합니다.

$$
\begin{aligned}
N_{10\alpha} &= H(sk_A|\sigma_1) \\
N_{5\alpha} &= H(sk_A|\sigma_2)
\end{aligned}
\qquad (7)
$$

4) AOS 영지식증거에서, 토큰 자산 Z′ A 와 Z′′ A 가 사용하는 공개키 pk_A 는 각각의 디스트로이어(destroyers)에서 사용되는 동일한 비밀 키인 sk_A 에서 파생된 것입니다.

$$pk_A = H(sk_A) \quad (8)$$

5) Alice 는 공개키 pk_A 를 사용하여 각각 Alice 와 Bob 을 위한 두 개의 신규 AOS 영지식 증거 중개 토큰 Z′′′ A 와 Z_B 를 생성합니다.

$$Z_A''' = H(3\alpha|pk_A|\sigma_3)$$

$$Z_B = H(12\alpha|pk_B|\sigma_4) \quad (9)$$

6) 디스트로이어(destroyer) 내의 값은 생성된 AOS 영지식증거 값과 동등하므로(10α+5α=3α+12α) 증거는 유효합니다. 상기 6 개 단계가 모두 완료되면, AOS 영지식은 검증 컨트랙트 상의 검증 기능과 전송된 트랜잭션을 받을 때의 공개 인풋을 통해 허브 컨트랙트가 유효함을 증명하게 됩니다. 또한 이는 검증이 성공적으로 이루어졌을 시에만 AOS 영지식증거 자산 전송이 성공적으로 이루어졌는지 여부를 검증해줍니다..

d) 개인 상태 트리(Private state tree)

AOS 는 암호화 기술을 사용해 거래자가 아닌 다른 사람들이 민감한 데이터를 열람하는 것을 방지해줍니다.
널리 사용되는 몇 가지 상태 트리(머클 트리)외에도 개인 상태 트리가 추가되었습니다.
AOS 는 스마트 컨트랙트 프레임워크를 사용해 개인정보보호 데이터를 분리합니다. 블록체인 운영은 이른바 블록 제안과 검증 프로세스의 두 파트로 나뉩니다. 블록 검증 프로세스는 트랜잭션 컨트랙트 코드를

처리함으로써 진행됩니다. 예를 들어, 트랜잭션 당사자와 관련된 공개 및 개인 트랜잭션은 모든 노드에 의해 검증되지만, 그 노드들은 다른 개인 트랜잭션에 대한 컨트랙트 코드의 실행 프로세스는 무시합니다.
이와 같은 오퍼레이션을 수행하기 위해, 상태 데이터베이스는 개인 상태 데이터베이스와 공개 상태 데이터베이스로 나뉩니다. 네트워크에 있는 모든 노드의 공개 상태는 완벽한 합의 상태에 있지만, 개인 상태 데이터베이스의 상황은 다릅니다. 클라이언트 노드의 상태 데이터베이스가 더 이상 전체 글로벌 상태 데이터베이스의 상태를 저장하지 못한다고 해도 실질적으로 분산된 블록체인과 그 안에 포함된 모든 트랜잭션은 여전히 모든 노드와 동기화되어 위 변조 방지(anti-tampering), 암호화 및 보안 프로세싱이 가능합니다. 개인 트랜잭션 데이터는 해당 노드의 개인 키를 통해서만 해독할 수 있기 때문에 설계의 안전성과 유연성을 동시에 향상시킵니다.

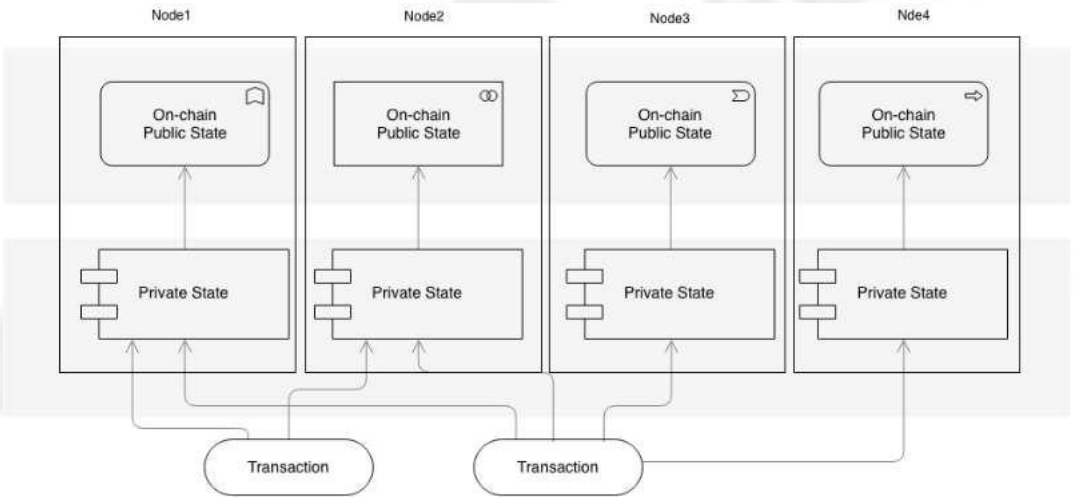


그림 4. AOS 개인 상태 트리 및 상태 데이터베이스

e) 스마트 컨트랙트

AOS 는 WebAssembly(WASM)을 가상 머신으로 사용해 스마트 컨트랙트 서비스를 사용자들에게 제공할 것입니다.

사용자들은 WASM 에서 필요한 모든 요구사항을 실현할 수 있습니다. 다수의 기본 암호화 작업이 많은 리소스를 소비함에 따라 호스트 컴퓨터의 컴퓨팅 성능을 사용해 복잡한 암호 작업을 계산하고 실행을 위해 결과를 가상 컴퓨터로 피드백 하도록 WASM 에 약간의 변경을 했습니다.

이런 디자인은 컨트랙트에서 개발자의 리소스 사용 문제를 극복하는 동시에 비밀번호 툴과 개인 정보 및 익명 요구사항을 마이닝을 하도록 독려합니다.

AOS 합의 메커니즘

트랜잭션 시나리오에서 낮은 대기 시간과 높은 처리량 요구를 고려해 도출된 합의 메커니즘은 DPos-pBFT 를 사용해 설계되었습니다.

a) DPoS

AOS 생태계 내에서 디앱을 사용하는 수백만명의 사용자들과의 호환성을 보장하기 위해서는 효율적으로 실행되는 합의 메커니즘이 필요합니다. PoW, PoS 및 DPoS 간의 성능을 비교한 표 2 를 보면 DPos 의 성능이 상대적으로 더 높다는 것을 알 수 있습니다.

표 2. PoW, PoS and DPoS 성능 비교

합의 알고리즘	**리소스 소비량**	**네트워크 스케일**	**트랜잭션 승인 소요 시간**	**안전성**	**처리량**
PoW	H	H	H	H	L
PoS	M	H	M	M	L
DPoS	L	H	L	M	M

(H-높음 M-중간 L-낮음)

동시에 EOS 는 DPoS 를 최대한 활용했으며 DPoS 의 모든 네트워크 노드는 블록 생성을 담당하는 수퍼 노드를 투표해 투표와 같은 온라인 거버넌스를 형성했습니다. 오직 한정된 숫자의 수퍼 노드만이 유권자들의 합의를 관리할 수 있는 권한이 주어졌으며 실제 우리가 아는 민주주의적 투표 시스템을 실현할 수 있습니다.

b) pBFT

pBTF 는 대부분의 경우 request, pre-prepare, prepare, commit, replay 의 다섯가지 과정 및 구간으로 나눠져 있습니다. 전반적으로 다음과 같은 전형적인 시나리오를 가정하겠습니다.

전체 시나리오에는 4 개의 노드를 각각 나타내는 고유한 클라이언트 C, 0, 1, 2, 3 이 있습니다. 일반적으로 0 을 기본 노드로 사용하고 1, 2, 3 을 보조 노드로 사용합니다. 다른 노드도 기본 노드 역할을 할 수 있습니다. 0 에서 오류가 발생한다면 오직 서버에서만 모니터링 할 수 있습니다. 서버가 일정 기간 내에 클라이언트 요청을 완료하지 못한다면 다른 노드를 기본 노드로 바꾸도록 뷰 교체 프로토콜이 트리거됩니다. 다음과 같이 'x'로 표시된 3 개의 노드는 결함 노드입니다.

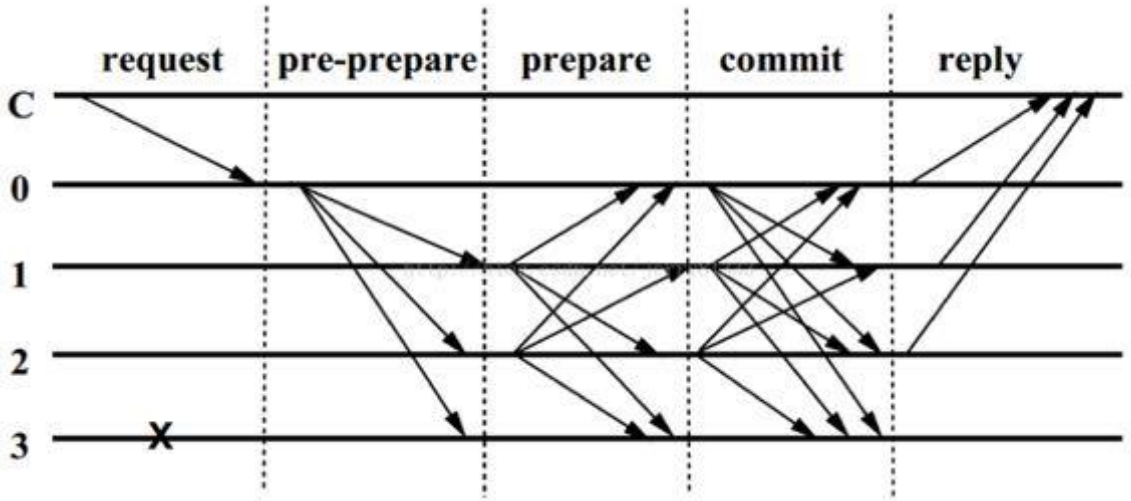


그림 5. pBFT 프로세스 차트

위에 있는 전형적인 시나리오에 따르면 프로세스는 다음과 같습니다.

1) Request

클라이언트 C 는 1 차 노드 (여기에서는 노드 0 이라고 칭함)에 request 를 전송하고;

2) Pre-Prepare

C 로부터 request 를 수신한 후 노드 0 은 1, 2, 3 에게 발신합니다.

3) Prepare

노드 1, 2, 3 은 다운타임으로 인해 노드 3 은 발신할 수 없기 때문에 수신 후 노드 1 -> 노드 0, 2, 3, 노드 2 -> 노드 0, 1, 3 으로 발신합니다. (이 단계는 기본 노드가 다른 request 를 각자 다른 보조 노드에 보낼 수 없게 하기 위함)

4) Commit

Prepare 단계에서 노드 0, 1, 2, 3 이 동일한 요청을 특성 횟수 (2F, 실제 사용 시 F 가 허용가능한 비잔틴 노드의 수) 이상 수신하게 된다면 Commit 단계로 들어가 Commit 요청을 발신하게 됩니다.

5) Reply

Commit 단계에서 노드 0, 1, 2, 3 중 하나가 동일한 요청에 대한 특정 수 (2F + 1) 이상을 수신하면 C 로 피드백이 됩니다.

위의 과정을 거친 후 $N \geq 3F + 1$ 인 경우 일관성 문제를 해결할 수 있다. 여기서 N 은 총 컴퓨터의 수이며 F 는 해당 컴퓨터 수입니다.

c) 잠정 알고리즘 DPoS-pBFT 구현 레퍼런스

1. DPoS 투표 결과를 노드가 차례대로 작업했을 때 블록이 끝나면 해당 작업 시간의 검사자(forger)가 해당 검증의 메인 노드의 역할을 합니다.

2. 주 작업 노드는 pre-prepare 메시지를 전파합니다. <pre-prepare,m,n,i,d,s>에서 m 은 새로운 블록, n 은 이번 회에서 빠지는 블록의 일련 번호, i 는 이번 회에서 빠지는 블록의 노드 일련 번호, d 는 m 의 써머리 (d=d(m), hashing byte 과정), s 는 써머리의 서명입니다.
3. 다른 보조 노드들이 처음으로 메시지를 수신해 합법적인지 확인 후 <prepare,m,n,i,d,s>메시지를 전파합니다. 메시지 내용은 위와 동일합니다.
4. 주 작업 노드 및 보조 노드를 포함한 모든 노드가 prepare 메시지를 수신하면 노드는 메모리에 받은 메시지의 수를 합산하기 시작합니다. 만약 같은 prepare 메시지를 2F + 1 개의 다른 노드로부터 수신하면 노드는 블록의 각 트랜잭션을 검증해 각각의 검증 결과를 메시지 M 에 첨부한 후 <commit,m,n,i,d,s>메시지를 전파합니다.
5. 각 노드가 2F +1 이상의 다른 노드로부터 commit 메시지를 받으면 합의에 도달했다고 생각하고 pass 상태로 접어들어 트랜잭션을 블록으로 패키지화 시켜 불가역적 고정 블록으로 변환합니다.
6. 주 작업 노드 또는 보조 노드가 메시지를 받으면 타이머가 시작됩니다. 타이머가 만료되었는데 합의에 도달하지 않는다면 합의가 파기됩니다.

실행 시나리오

스마트 컨트랙트와 익명성의 조합으로 수많은 시나리오가 나올 수 있습니다.

중앙 집권형 인터넷 애플리케이션에서는 사용자들은 개인 정보나 공정성을 갖지 못하며 이는 개인 정보 보호 기능이 없기 때문입니다. 중앙 집권형 인터넷 앱은 유저의 트럼프 카드를 직접 보고 사용자 정책을 그에 따라 조정할 수 있습니다. 그런 앱이 어떻게 공정할 수 있겠습니까?

AOS 는 사용자의 프라이버시를 우선시하며 이런 상황을 반대의 상황으로 바꾸려고 합니다. 전세계 사용자들이 함께 동참해 자율의, 분산된, 개방된, 투명한, 공정한, 익명의 공유 블록체인 커뮤니티를 구축하기 위해서는 공정한 블록체인이 필수적입니다.

하단에는 몇 가지 일반적인 AOS 응용 시나리오를 리스트로 작성하였으며 개발자 및 사용자들이 이를 통해 가까운 시일 내에 점점 더 풍부한 다양한 응용 프로그램을 만들 수 있기를 희망합니다.

a) 익명 자산

익명 자산은 AOS 의 첫번째 일반적인 응용 시나리오이자 AOS 익명성 기술을 최대한 활용하는 애플리케이션 시나리오입니다.

모네로, ZCash, Grin 과 같은 익명 통화는 사용자가 자신의 자산을 발행할 수 있도록 지원하지 않습니다.

실제로 사용자에게 부여된 기본 권한 중 하나는 자신의 익명 자산을 발행할 수 있는 권한입니다. 자산 자체를 발행하는 것은 블록체인이 사용자에게 제공하는 권력입니다. 중앙 집권형 기관없이 일반적인 사용자들이 본인의 자산을 발행할 수 있도록 하는 경우는 이번이 처음이며 익명성이 자산의 강력한 속성 중 하나입니다. 따라서 익명 자산의 발행 및 거래는 AOS 의 첫번째 일반적인 응용 프로그램으로 간주됩니다.

AOS 가 공식 익명 자산 분배 컨트랙트를 제공 및 배포를 하게 된다면 ETH 와 비교했을 때 AOS 에서 ERC20 자산을 발행하는 것이 더 편리합니다. 사용자들은 컨트랙트만 호출하면 그들의 익명 자산을 취득할 수 있습니다.

컨트랙트는 ERC20 패러다임 자산 모델을 지원하며 익명 및 비 익명 거래 모드를 지원하며 두 모드 간 변환 또한 지원합니다.

또한 AOS 의 특징 중 하나인 탈중앙화 교환은 다양한 자산의 상호 변환을 가능하게 해줄 것입니다.

AOS 는 강력한 기술 서포트를 받고 있고 AOS 가 제공하는 자유로운 자산 발행, 탈중앙화 트랜잭션 모드, 사용자 개인 정보 보호는 자본 시장에 필연적으로 큰 영향을 미칠 것입니다.

b) 회사 스톡 옵션

DAO 또는 DAC 관리 모드는 블록체인 및 탈중앙화로 인해 생기는 새로운 회사 및 팀 관리 방법입니다.

하지만 기존 회사 구조를 완전히 무시할 수 없으므로 다음과 같은 두가지 옵션이 제공됩니다. (i) 중앙 집권형 기업의 관리 모드를 건너뛰고 DAO 조직 형태로 바로 시작하는 것 (ii) 중앙 집권형 기업을 돕기 위해 다양한 매니지먼트 체계를 제공하고 점차적으로 DAO 로 발전할 수 있도록 돕는 방식이 있습니다.

두번째 방식이 더 실용적이고 실현 가능하기 때문에 AOS 는 일반 기업들에게 익명 기업 옵션 솔루션을 제공할 것입니다.

회사 초기 발전 단계에서 회사는 종종 직원들에게 스톡옵션으로 보상합니다. 직원의 급여와 마찬가지로 각 직원이 보유한 옵션은 기밀입니다. 이는 직원의 개인 정보를 존중하는 조치일 뿐만 아니라 기업 지배 구조의 일부이기도 합니다. 전통적인 회사는 중앙 집권형 방식으로 스톡옵션 발행을 관리합니다. 종종 스톡옵션은 거래를 할 수 없으며 다수의 경우에는 직원이 만약 중간에 회사를 떠나면 최종적으로 받을 수 있는 스톡 옵션을 못 받게 됩니다.

AOS 는 전통적 구조의 회사들이 AOS 의 익명 옵션 체계를 채택할 것을 제안합니다. 회사는 블록체인을 통해 직원들에게 회사 스톡옵션을 발행하고 AOS 익명 기술을 사용해 사용자의 개인 정보를 보호할 수 있습니다.

직원이 회사의 발전에 대해 낙관적이라면 회사가 상장될 때까지 옵션을 보유하거나 적절한 가격이라 생각되면 회사를 떠난 후에도 보유할 수 있습니다. 이 모든 방식은 직원의 개인 정보를 보호한다는 전제 하에 진행할 수 있습니다.

AOS 의 익명 옵션 체계가 직원의 동기 부여 및 결속력을 크게 강화하고 직원과 회사가 함께 성장할 수 있도록 지원하며 직원들에게 더 많은 선택권을 제공할 것이라고 생각합니다.

c) 스마트 포커

스마트 포커는 전형적인 AOS 애플리케이션의 시나리오입니다. 한편으로 플레이어는 라이센싱 프로세스가 공정하고 무작위 적이길 바라면서 그와 동시에 플레이어는 다른 사람들이 자신의 카드를 보지 못하길 바랍니다.

이 방식은 전통적으로 진행되는 게임도 EOS 의 인기있는 게임에서도 달성할 수 없습니다.

전통적 게임에서는 라이센싱 프로세스가 불공평하며 중앙 집권화로 인해 모든 것이 제어됩니다. 플레이어의 손에 있는 카드도 중앙에 표시되며 볼 수 있습니다. 마치 콜로세움의 검투사와 같이 플레이어는 공정한 결투를 하고 있다고 생각하지만 모든 공연이 노예를 소유한 주인들이 자비롭게 제공한 기회라는 것을 고려한다면 이들은 영원한 노예일 수밖에 없습니다.

이게 과연 우리에게 필요한 것일까요? 물론 아닙니다. 블록체인은 우리에게 새로운 희망을 줄 것입니다. EOS 로 대표되는 블록체인은 시작부터 더 공정한 게임을 할 수 있도록 노력하고 있습니다.

EOS 에서 더 공정한 랜덤 숫자를 사용한 새로운 게임 모드를 사용할 수 있지만 대부분 이러한 게임은 기술적 한계로 인해 딜러-플레이어 1 회 라이센싱 모드만 가능합니다. 따라서 이중 라이센싱을 위해서는 하나의 랜덤 숫자가 필요합니다. 예를 들어 주사위 던지기 게임인 EOSBet 이 있습니다.

하지만 고급 포커 게임에서는 EOS 에서 공정성을 실현할 수 없습니다.

반면 이런 공정성은 AOS 가 딜러 및 플레이어들에게 제공하는 영지식증명을 통해 달성할 수 있습니다. 딜러는 그의 라이센싱이 공평하게 되었다는 것을 증명할 수는 있지만 어떤 라이센싱 인지에 대해서는 플레이어들에게 공개되지 않습니다. 플레이어들 또한 본인이 속이지 않고 본인의 카드는 딜러에게 받은 것이라는 것을 증명할 수 있지만 플레이어 당사자를 제외한 다른 사람들은 남은 카드에 대한 정보를 알 수 없습니다.

자산 계획

총 100,000,000,000 개의 AOS 토큰이 AOS 출시를 위해 생성될 것입니다.

개인 정보 보호를 지원하는 스마트 컨트랙트 플랫폼인 AOS 는 스마트 컨트랙트 개발자가 자체 디앱, 토큰 및 애플리케이션 생태를 개발하도록 권장합니다. AOS 의 모든 토큰 사용자들은 전송 작업을 수행하려는 경우 AOS 를 메모리, 대역폭, CPU 및 기타 시스템 리소스로 교환해야 합니다. AOS 네트워크의 가치를 측정하는 기본 단위로 AOS 코인이 사용될 것이며 토큰 생태 간 가치 이전 기능을 수행합니다. AOS 에 더 많은 애플리케이션 생태가 구축됨에 따라 AOS 시스템의 가치가 높아질 것입니다.

AOS 의 미래 애플리케이션 생태계를 구현하기 위해서는 강력한 성능을 갖춘 블록체인 합의 알고리즘이 필요합니다. 하단에서는 PoW 와 PoS 알고리즘을 비교해 왜 AOS 가 DPoS 를 선택했는지 설명할 것입니다.

비트코인이 채택한 PoW 합의 알고리즘은 특정 컴퓨팅 워크로드를 통해 그에 상응하는 보상을 얻어야 한다. 마이너들은 부기(bookkeeping) 권한과 새로운 통화를 얻기 위한 수학 연산을 수행해야 하므로 부기의 진위성과 효과성을 보장하기 위해 많은 시간과 자원을 소비해야 합니다. PoW 는 허용되는 수학적 공식을 통해 간단한 알고리즘을 쉽게 구현할 수 있다는 장점을 가지고 있습니다. 만약 누군가가 PoW 시스템을 파괴하고 싶다면 많은 컴퓨팅 비용을 투자하고 특정 보안 방식이 필요합니다. 하지만 PoW 는 이미 잘 알려진 단점도 있습니다.

PoW 는 많은 전력을 필요로 하고 처리 효율성이 낮아 초당 7 개의 트랜잭션만을 처리할 수 있습니다.
생태학적으로 풍부한 AOS 네트워크를 지원하기에는 성능이 부족해 보입니다.

사람들이 점점 중앙 집중화 컴퓨팅 전력 분배와 충격적인 에너지 소비에 지치게 되면서 PoS 가 등장했습니다. PoS 는 화폐 보유에 따라 이자를 생성하고 아웃 바운드 노드에게 보상을 합니다. PoS 는 화폐 연령 = 통화 수량 x 화폐 보유 기간이라는 독특한 개념이 있습니다. 만약 PoS 노드가 블록을 찾는다면 화폐 연령이 지워지며 동시에 블록으로 보상을 받습니다. PoS 블록에 참여하는 노드가 오래 될수록 다음 블록의 블록 가중치를 얻을 수 있는 가능성이 높습니다. 이로써 합의에 도달하는 시간이 단축되고 효율성이 향상될 수 있습니다. PoS 의 단점은 이해관계가 있는 참여자들이 이자 보유량을 늘리기 위해 화폐를 보유할 수 있어 독점하기 쉽다는 것입니다. 또한 PoS 네트워크 성능은 PoW 네트워크 성능보다 떨어지며 트랜잭션 확인을 위해서는 여전히 수 분이 소요되며 디앱의 성능 요구사항을 거의 충족시키지 못합니다.

공유 승인 인증서라고도 하는 DPoS 는 각 홀더가 투표할 수 있도록 해 특정 수의 대표자 (아웃 바운드 노드)를 생성해 이 아웃 바운드 노드는 네트워크 유지 작업을 완료합니다. 이런 아웃 바운드 노드는 동일한 권리를 갖고 있습니다. DPoS 는 투표를 하는 이사회와 같으며 홀더는 아웃 바운드 노드의 숫자를 정하는 투표를 하게 됩니다. 이런 노드는 설정된 스케줄에 따라 블록을 생성합니다. 전체 네트워크를 잘 유지하지 못하는 대표들은 제거가 됩니다. DPoS 는 PoW 및 PoS 와 비교해 진위여부 검증 및 계산에 참여하는 노드 수를 크게 줄여 효율성을 크게

향상시킨 것이 장점입니다. 하나의 트랜잭션을 확인하는데 필요한 시간은 0.5 초 이내이며 1 초 내 승인할 수 있는 트랜잭션의 수가 크게 증가합니다.

DPoS 설계에 따르면 AOS 네트워크는 여러 발신 노드에 의해 유지됩니다. 모든 AOS 홀더들이 AOS 아웃 바운드 노드에 대해 투표를 실시합니다. 이 투표는 늘 업데이트 됩니다. 어느 시점에서나 최상위 노드 후보자에게 AOS 네트워크에 참여할 수 있는 블록 권한이 부여됩니다. 노드가 가진 투표권의 가중치는 투표자가 가지고 있는 AOS 토큰 양에 비례합니다. AOS 토큰을 더 많이 보유할수록 AOS 네트워크를 더 대표할 수 있게 됩니다. 따라서 AOS 토큰 보유는 AOS 온라인 투표 참여에 있어서 필요한 유일한 확인 조건입니다. 이런 AOS 아웃 바운드 노드를 보상하기 위해 네트워크 아웃 바운드에 참여하는 각 노드는 특정 AOS 토큰을 보상으로 받습니다. 이로 인해 매년 AOS 토큰 양이 조금씩 늘어날 것입니다. 하지만 AOS 의 인플레이션율은 연간 약 4%인 BTC 와 ETH 의 인플레이션율에 비해 훨씬 작을 것입니다.

결론적으로 일반 사용자들은 AOS 토큰을 보유해야만 AOS 네트워크를 사용할 수 있는 권한을 부여 받습니다. 또한 개발자는 컨트랙트를 배포하고 실행하는데 필요한 리소스를 지불하기 위해 AOS 토큰을 보유해야 합니다. 동시에 AOS 아웃 바운드 노드는 네트워크 아웃 바운드에 참여하기 위해 AOS 토큰 홀더의 투표권을 얻어야 합니다. 따라서 AOS 토큰은 AOS 네트워크의 가치를 보유하는 유일한 수단이 될 것입니다.