

T A O Green Paper (록서)

목차

T A O Green Paper (록서).....	- 1 -
About	- 3 -
Document History	- 3 -
Executive Summary.....	- 4 -
1. 이더리움 과 EVM (이더리움 가상머신)의 이해	- 5 -
1.1 이더리움 클라이언트 노드의 구성	- 5 -
1.2 EVM (Ethereum Virtual Machine) 의 유용성과 장점.....	- 5 -
1.3 EVM (Ethereum Virtual Machine) 의 취약점과 단점.....	- 6 -
1.4 이더리움 개발진의 EVM 단점을 극복하기 위한 개발방향.....	- 7 -
1.5 샤딩(Sharding) ^{[27],[28]} 과 SideChain ^[22] 비슷한 개념을 적용한 사례	- 7 -
1.5 비트코인 ^[1] 체인의 특성	- 7 -
1.6 EVM 의 단점을 극복하기 위한 TAM 제작.....	- 8 -
1.7 TAM , EVM , WASM 동작 및 리소스 소모 비교도	- 9 -
1.8 저장장치 Scalability 방안	- 10 -
2. TAO 체인의 이해	- 11 -
2.1 블록체인에 대한 기본적인 이해	- 11 -
2.2 TAO 체인의 목적.....	- 12 -
2.3 TAO 체인의 장점.....	- 13 -
2.3.1 기존 컨트랙 DDOS 사례탐구를 통한 개선방안 도출안	- 13 -
2.3.2. TAM (TAO 어플리케이션 머신) 을 개발 적용.....	- 14 -
2.3.3 TAM 의 주요 구성규격.....	- 14 -

2.4 TAO 체인에서 사용되는 블록체인 노드구조.....	- 15 -
2.4.1 기존 PoW / POS / POA 체인 클라이언트 노드 구조변경.....	- 15 -
2.4.2 iTAO LiteChain 에서 적용될 iTAO 클라이언트 노드.....	- 15 -
2.4.3 TAO Full Chain 에서 적용될 클라이언트 노드.....	- 16 -
2.5 TAO 체인에서 콘텐츠 플랫폼으로서의 방안.....	- 17 -
2.5.1 콘텐츠 플랫폼 Backend 로드맵.....	- 17 -
2.6 DPOS/POS 계열 검증인 노드 보안성 강화 방안.....	- 17 -
2.6.1 기존 컨트랙 DDOS 사례탐구를 통한 개선방안 도출안.....	- 18 -
2.7 체인별 스테이트 공유 및 인터렉션 방안.....	- 19 -
3. TAO 블록체인 구조의 동작 프로세스.....	- 20 -
3.1 TAO 아키텍처 세부구조.....	- 20 -
3.1.1 TAO 구조 응용 레이어.....	- 21 -
3.1.2 TAO 블록체인 라우터 인터페이스.....	- 21 -
3.1.3 TAO 블록체인 스페이스.....	- 23 -
References.....	- 24 -

About

Author : KT Ahn (cpplover@trustfarm.net) , @trustfarm
Founder of trustfarm.net , Founder of tao.foundation

Advisor : Mr. Woo [Investing/Trade/Moderating]
Mr. NR [Marketing/Investing]

Document History

Date	Ver	Subject	Author
2016 / 07/ 13	Base	Draft Patents concepts paper on Automobile blockchain architecture	trustfarm
2018 / 07/ 23	V0.10	Documentize on word format	trustfarm
2018 / 07/ 27	V0.90	Complete review and wording on scattered papers.	trustfarm
2018 / 08/ 01	V0.91	Review on Documents and Corrects.	trustfarm
2018 / 08/ 02	V0.92	Revision and Altering the templates and marks	trustfarm
2018 / 08/ 03	V0.93	Adds TAM Client Architecture / moderate references	trustfarm
2018 / 08/ 08	V0.932	Add reference mark , add summary page	trustfarm,woo
2018 / 08/ 18	V0.94	Add Blockchain Universe Router Architecture	trustfarm

Executive Summary

TAO는 여러개의 독자적인 블록체인이 각각 제 기능을 수행하면서 각 체인을 동시에 연계시키는 블록체인 아키텍처입니다. 또한, 각각의 사물과 객체들이 1) 분권적 철학, 2) 존재의 의미, 3) 행위의 기본적인 방식을 가정합니다. 또한 4) 라이프 사이클이 존재하는 것을 기본개념으로 정의합니다.

이와 같은 사상을 블록체인의 구조에 접목시, 각각의 블록체인별로 특이한 강점들을 연계할 경우, 트래픽의 분산과 분권적화를 효율적으로 구성할 수 있습니다.

현재 블록체인의 문제점인, Scalability (확장성) 과 기존 스마트컨트랙 기반의 DAPP의 사용확대에 따른 트래픽의 분산, IoT 용에 적합한 체인개발과 아울러, IoT 기기에서 스마트컨트랙을 수행할수 있는 TAM (TAO Trust Application Machine)을 개발하는 프로젝트입니다.

TAO블록체인의 목적은 위의 구조를 적용하는 방법을 기본으로, 현실의 블록체인의 스칼라빌리티(확장성) 문제를 해결하고자 하는 것 입니다. 그리고, 하나의 블록체인으로 모든 기능을 효율적으로 담당할수도 없고, 체인안의 모든 데이터가 영구적으로 존재할 필요는 없습니다. 이에, TAO는 블록체인의 라이프사이클과 라이프타임으로 제시하고, 이를 통해서 확장성문제 해결과 네트워크 트래픽을 감소할 수 있는 효율적 구조를 지향합니다. 그리고, 라이프사이클이 종료는 체인은 분산된 저장공간에 아키브되고, 기존의 부모체인에 사이드체인의 해시로 남게 되고, 경우에 따라서, 스테이트가 부모체인으로 이전되게 됩니다.

또한, 아주 경량의 IoT 및 Sensor 디바이스에도 블록체인을 활용한 M2M 이 가능한 체인이 존재하여야 타오에서 보는 IoE(만물 인터넷)에 적합한 구조가 되므로, IoE 용 라이트체인이 필요하고, 이러한 체인에서 동작하는 TAM 이 필요하게 됩니다.

이러한 TAO 구조를 기반으로 콘텐츠 플랫폼이 주요 서비스로 생성되고, TEO 및 TEOS 등의 TAO구조안의 패런트체인은 이러한 서비스의 Backend 블록체인 역할을 수행하게 됩니다. DAPP 및 서비스의 형태에 따라서, 특징이 있는 패런트 체인을 기반으로 서비스가 탑재되게 되는 구조입니다.

1. 이더리움 과 EVM (이더리움 가상머신)의 이해

1.1 이더리움 클라이언트 노드의 구성.

현재 이더리움^[6] 구동을 위해서는 geth/party 의 클라이언트 노드 프로그램이 필요하며, 각각의 클라이언트 프로그램에는 이더리움 스펙 (Yellow Paper)^[7] 에 따른 EVM^{[24],[25]} (이더리움 가상머신)을 지원하게 되어 있습니다.

이더리움에서 스마트컨트랙을 구현하기 위해서는 그 하부구조에 EVM(Ethereum Virtual Machine)이 필요하고, EVM과 연동되어서 스마트컨트랙트^{[5],[7],[23]}(계약) 가 실행되게 됩니다.

또한, DAPP (DeCentralized Application) – 이더리움 블록체인 상에서 분권화 확장가능한 Application 또한, EVM 과 연동되어서 실행되게 됩니다.

다음은 이더리움 클라이언트 노드의 구성도 입니다.



Figure 1 Ethereum Client Node

1.2 EVM (Ethereum Virtual Machine) 의 유용성과 장점.

- ① 기존 비트코인 블록체인과의 차이는 블록체인 안에서 스마트 컨트랙트를 구현하고 실용화 했다는 것에 있습니다.
- ② 또한, 분권화 된 블록체인 기반의 응용프로그램을 돌릴수 있게되어서 중앙의 중재자나 서비스 제공자가 없어도 P2P 로 거래가 가능하게 되었습니다.

- ③ 또한, 콘텐츠 거래 P2P간의 스마트 계약 기반한 거래를 도와주는 분권화 앱을 통해서 거래의 편의성과 중앙중재자의 신뢰없이도, 블록체인내에서 신뢰가능한 거래를 활성화 할수있도록 되었습니다.
- ④ 이는, 콘텐츠 판매시, youtube 같은 곳에 사용자나 광고자가 돈을 주는 것이 아닌, 개별 콘텐츠 제작자가 직접 수익을 소비자로부터 중재자 없이 받을수 있고, 광고자가 직접 광고를 클릭한 소비자에게 베네핏을 코인으로 줄수있는 기술적 기반을 만든것임에 상당히 혁신적인 개념의 구현이 가능하게 되었습니다.

이 자체만으로도 큰 의미가 있습니다. 그러나, 2015년 초기 Launch 이후 부터, 지금까지 여러 번의 EVM 의 취약점을 공격하여, 이더리움 노드를 마비시킨 경우도 있었습니다. 현재 스마트 컨트랙트는 완벽하지 않습니다.

1.3 EVM (Ethereum Virtual Machine) 의 취약점과 단점.

- ① 현재 EVM을 구동하기 위해서는 노드의 런타임 메모리 용량이 많이 필요한 단점이 있습니다.

2016년 10월 11월 하반기 , EVM Gas Spam DDOS공격시에 16GB 이상의 메모리가 필요하였고, 노드가 메모리가 없을경우, OOM (Out Of Memory) Exception 으로 블록체인에 참여하는 대다수 노드가 다운되기도 하였습니다.

- ② 또한, 컨트랙 코드가 포함된 트랜잭션이 블록에 많이 포함될경우, 16GB 이상의 고사양 컴퓨터에서도 블록체인 노드의 동기화가 어렵습니다.
- ③ 또한, Ethereum 의 가격이 높아지고, 트랜잭션 SmartContract 이 포함된 트랜잭션이 기하급수적으로 많아지면서, 정상적인 이더리움 거래 및 특히, 컨트랙을 수행하는데 Gas 비가 많이 높아져서 향후 DAPP 의 활성화에 걸림돌이 되고 있는 상황입니다.
- ④ 특히, 고정적, 안정적으로 장시간 클라이언트를 고용량 메모리와 고성능 PC가 필요하며 효율이 좋지 않습니다.
- ⑤ 보안관점에는 여전히 디도스와 , 스팸 컨트랙트로 인한 네트워크 트래픽증가가 우려될수 있고, 그로인해 블록사이즈가 증가하여 GAS비증가, 네트워크 유지비용및 처리속도가 오래 걸려 트랜잭션 처리가능한 Full 노드의 감소로 이어지고 있습니다.

1.4 이더리움 개발진의 EVM 단점을 극복하기 위한 개발방향.

- ① OffLine Transaction 을 활용한 SideChain 기법을 연구하고 있으나, 이는 또다른 중앙화된 트랜잭션 처리 전문서비스의 등장이 될수있는 우려는 남아있습니다.
- ② 현재 EVM에서 WASM(웹 어셈블리) 쪽으로 접근하려는 개발도 시도되고 있습니다. EVM 보다는 좀더 Instruction (명령어체계) 가 실제 CPU 에 유사하고, WebEngine 기반의 LOW LEVEL 인 머신랭귀지기반으로 새로운 Contract 언어와 시스템을 만들고자 하는것입니다.
- ③ Scalability(확장성)^[12] 강화를 위한 Sharding(샤딩)^{[11], [26]}과 Lite Client 를 준비하고 있습니다.

1.5 샤딩(Sharding)^{[27], [28]} 과 SideChain^[22] 비슷한 개념을 적용한 사례

- ① 코스모스(Cosmos)^[20]의 Zone 개념의 분산처리와 이기종 체인간의 인터커넥션을 제공하여 상호연동 운영이 될수있는 구조를 제안하고 있습니다.
- ② 패러티 (Parity.io) 에서는 폴카닷(Polka Dot)^[18] 을 개발하고 있으며
- ③ 스테이트 공유방안으로, 이더리움 클래식, 칼리스토 등은 컨트랙트 기반 메시지 기능,
- ④ 비트코인 계열에서는 아토믹스왑^[21], DEX(거래소) 등 다양한 방법으로 문제를 해결에 접근하고 있습니다.
- ⑤ 기존 POW 체인 에서 POS 체인 으로의 이전, 또는 POW / POS 하이브리드 형태로 체인구조를 변경하여 사용하고 있기도 합니다.

1.5 비트코인^[1] 체인의 특성

현재 P2P 기반의 블록체인은 용량에 비례하지만 동작시 시스템로드가 낮습니다. 또한, 트래픽이 증가된다 하더라도 예측이 가능합니다.

트래픽 증가된다는 개념은 네트워크 트래픽의 증가를 의미하며 노드의 시스템 부하는 상대적으로 적게 증가합니다.

(고성능의 PC가 아니어도 비트코인 체인을 구동하는데 별 어려움이 없는 이유입니다)

1.6 EVM 의 단점을 극복하기 위한 TAM 제작

EVM 은 Lite 한 시스템에서는 구동하기가 어렵습니다. 글자 그대로 작동만 됩니다. 작동된다 하여도, 컨트랙의 복잡도나 컴퓨팅 시스템 자원에 제약을 받아서, 동작이 중단되기도 합니다.

따라서, Lite 한 시스템에서는 컨트랙을 몇 개 구동시키기에도 어려움이 있기에 스마트폰이나 IoT 기기들, 공유기/NAS 서버등 에는 적용되기 어렵습니다.

기기별 컴퓨팅 리소스 대비 적용 가능한 시스템을 정리하자면 아래와 같습니다.

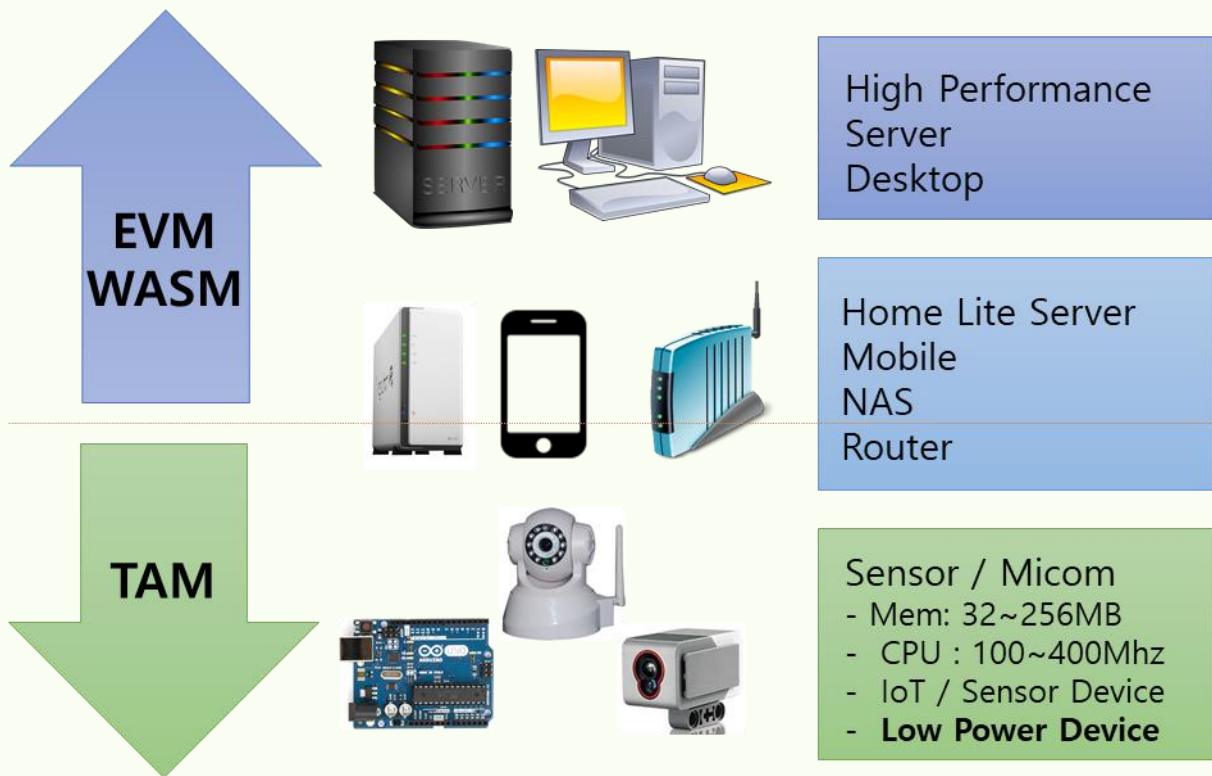


Figure 2 Contract Applicable System by System Power

현재 Micom의 CPU 코어는 ARM 이나 MIPS 두 곳이 대부분을 차지하지만 ARM 계열이나 MIPS 코어를 기반한 CPU 를 사용하기에는 위 두가지 코어 모두 법적인 라이선스 문제가 존재합니다.

그러므로 컴퓨터 구조적으로 분류되는 연산명령어 체계 [CISC/RISC] 중 Embedded 기기에서 많이 사용되

는 구조인 RISC 구조(라이선스가 없는 RISC-V^[15] 또는 OpenRISC^[16] 등이 적용됨) CPU 명령어세트를 채택합니다. RISC 코어 명령어 기반의 CPU를 가상화 또는 기존 Core에서 Emulation 하고, 이 명령어 기반에서 OS와 컨트랙트 응용프로그램이 동작될 수 있는 머신을 만듭니다. 이는 Sensor / Micom 레벨의 기기에서도 블록체인 컨트랙이 수행가능하도록 리소스를 최소화 하고, 라이선스 프리한 Application Machine 을 구현하는데 있습니다.

이 가벼운 Application Machine 이 **TAM (TAO Trust Application Machine)** 으로서 경량 머신에서 스마트 컨트랙과 M2M 등 IoT 커넥트비티에 충실한 기능을 수행합니다.

참고로, 컴퓨터 CPU의 기계명령어 체계는 다음과 같이 분류됩니다.

- CISC (Complex Instruction Set Computer)^[14] : 인텔의 X86 CPU가 대표적이고 (윈도우 기반의 PC Desktop / Notebook , 서버등에서 많이 사용) , 연산명령어가 복잡하며 파워를 많이 소모합니다. 단, 단일명령어로 복잡한 배치기능을 수행하는데 뛰어납니다.
- RISC (Reduced Instruction Set Computer)^[13] : ARM , ALPHA , MIPS (기존 휴대폰 및 임베디드 계열에 많이 쓰이고, 기존 유닉스 시스템에서 서버에서 많이 사용) 단순하고 명령어 동작 시간이 일정합니다. CISC와 달리, 여러 개의 단일명령어 조합으로 복잡한 배치기능을 수행하고, 연산 수행 길이 가 짧고, 일정해서 병렬연산 , 저전력구동에 뛰어납니다.

1.7 TAM , EVM , WASM 동작 및 리소스 소모 비교도

기존 체인은 현재와 같이 일반 트랜잭션 및 컨트랙트 트랜잭션이 모두 포함된 올인원 체인 개념입니다.

트랜잭션과 스마트 컨트랙트 트랜잭션, 토큰 컨트랙션, DAPP (분권화응용프로그램) 등등이 전부 포함이 됩니다.

따라서, 기존의 노드에서는 블록을 전파 받았을경우, 노드에서 일반 트랜잭션 검증, 컨트랙트 수행 및 검증, DAPP 프로그램 실행 및 결과 블록에 저장등등 모든 것을 한번에 수행하고, 이를 한 개로 만들어진 체인에 기록해야 합니다.

따라서, 다음과 같이 컨트랙이 동작할 때, 노드 시스템에 몇 개의 인스턴스가 존재할지는 전적으로 노드의 컴퓨팅 리소스에 영향을 받습니다.

이를 개략적으로 도식화하면 다음과 같습니다.

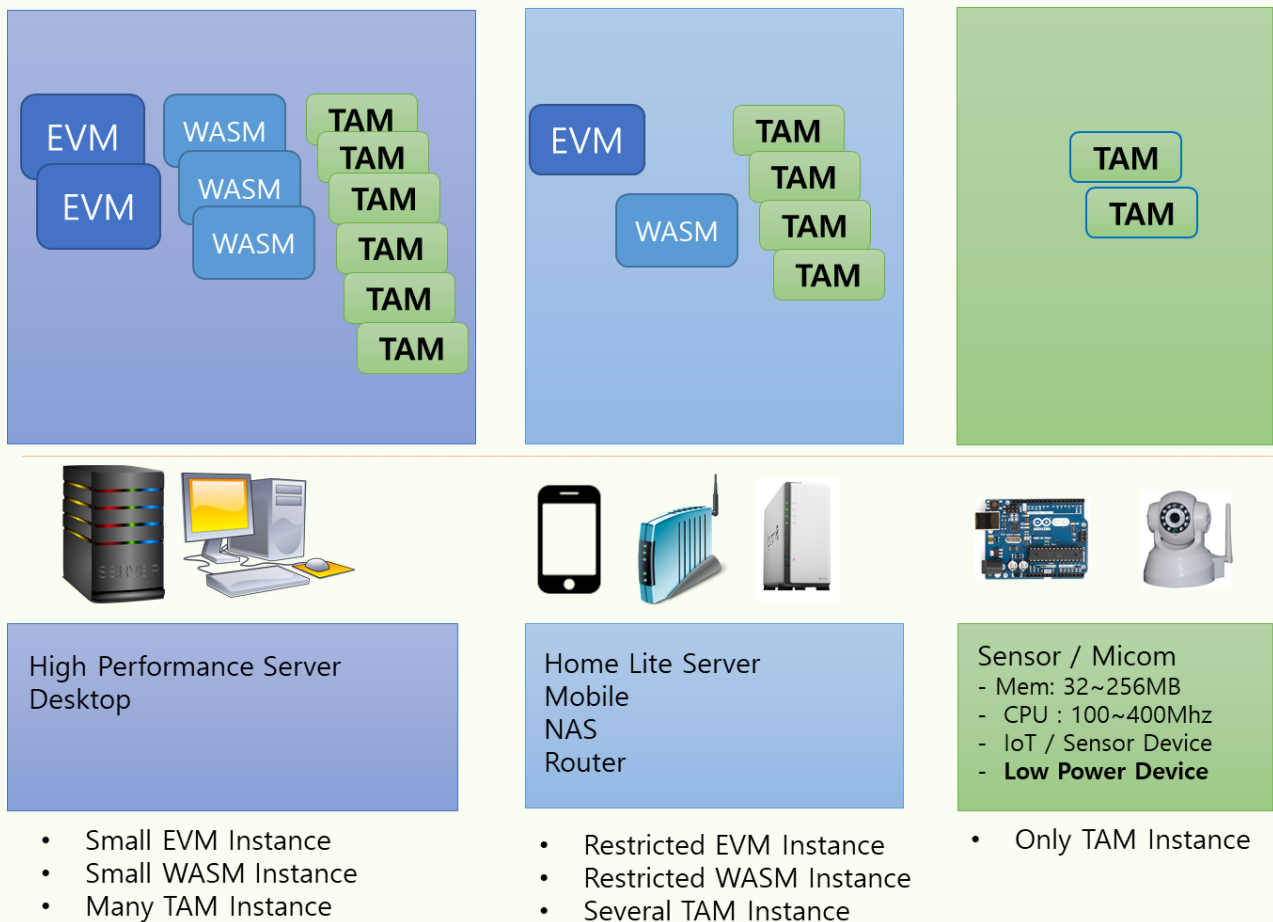


Figure 3 Concurrent Instances by Machine

TAO에서는 IoT / Sensor 기기와 같이 메모리 및 컴퓨팅 자원, 저전력이 요구되는 micom 과 같은 CPU 에서 구동가능한 TAM 을 만들고, 여기에 적합한 Lite Client Chain 을 만들어서, 기존 대용량 서버급 PC 에서 구동될 컨트랙이 수행될수 있도록 하여, 스마트 컨트랙이 가능한 M2M (Machin to Machine) 블록체인의 확산에 기여하고자 합니다.

1.8 저장장치 Scalability 방안

기존의 이더리움 블록체인과 이더리움 체계를 채용하는 DPOS 계열의 체인의 경우 늘어나는 TPS (Transaction Per Seconds) 를 늘리기 위한 방법을 위해서, 샤딩 (Sharding) 등의 기법을 위해서 Offline 블록체인, 사이드 체인 기법등 여러 방안이 연구되고 실험되고 있습니다.

또한, 그런 기법을 일부 적용하고 있습니다.

하지만, 현실의 문제는 이더리움 Full 노드를 동기화 할 경우, 현재, 6.07M Block (2018.08기준) 약 1.35TB 용량의 저장장치를 사용하고, 늘어나는 컨트랙트 트랜잭션, DAPP (특히, 클립토게임등의 분권화앱) 등의 컨트랙트 트랜잭션이 늘어나면서, 기하 급수적으로 체인의 저장소 크기가 늘어나고 있습니다.

또한, 많은 트랜잭션을 검증하기 위해서, 컴퓨팅 리소스뿐만 아니라, 기존 HDD 가 아닌, 초고속의 SSD 를 이용해야만, Full 노드 동기화가 가능한 상황입니다.

이는 , 현재의 블록체인 저장구조가 올인원 방식으로 저장하기 때문입니다.

TAO 에서는 이러한 저장장치 집중화와 점점 줄어드는 Full 노드의 해결책으로 , 독립적으로 기능별 충실한 블록체인을 별도로 생성하고, 기존 체인과 신규 체인과의 State 연동을 통해서, 1차적으로 저장소와 기능별 분산방안 구현합니다. 각각의 체인에서 모두 컨트랙이 가능하지만, 효율적인 체인에서 이를 수행하게 별도의 분산 체인을 제공합니다.^[4]

즉, 기존 PoW 체인 (TEO) 에서는 일반 트랜잭션과 기본적인 컨트랙정도를 수행하고,

향후, 트랜잭션 능력이 기존 PoW 보다 뛰어난, PoS/PoA 체인에서는 토큰 전송, DAPP 과 같이 빠른 트랜잭션과 컨트랙을 위한 용도의 체인을 제공합니다.

그리고, Lite한 IoT Device 및 M2M 컨트랙이 필요한 곳에는 iTAO 와 같은 체인을 별도로 만들어서 담당합니다.

2. TAO 체인의 이해

2.1 블록체인에 대한 기본적인 이해

체인은 필요함에 따라 만들어지고 포크되고 사라지기도 하며, 다른 메인체인이 있으면 흡수될 수도 있고, 메인체인이 없어졌다면 기능이 특화된 다른 체인이 나와서 그 역할을 대체하여 작동할 수도 있습니다.

굳이 POW만을 추구해서 DAPP을 돌리기 위해 POW만 이용할 이유는 없습니다.

만약 DAPP을 구동하기 위함이라면 빠른 트랜잭션과 컨트랙이 가능한 PoS^[8] 또는 PoA^[17] 체인을 하나 만들면 되는 것 뿐입니다.

한 체인에서 POW(채굴) + POS(포스) 를 동시에 하려는 것이 기존 하이브리드 체인입니다. 그러나, 스케일

러빌리티(확장성)등 문제가 있으므로 독립된 체인으로 포크되고, 스테이크를 공유하는 방식이 더 효율적이라 판단합니다. 즉, 독립된 체인으로 가되 각각 체인의 State 및 Balance 스왑만 제대로 해주면서 (PoW로 채굴한 것 과 PoS 로 채굴 및 Stake이자 수익이 1:1로 스왑(교환)되도록 하여 같은 가치를 가진다는 것이 증명되면 사용상 문제 없습니다.

즉, 기능적으로 체인을 사용하는것이 효율적이고, 전체적 트래픽 분산 관점에서 바람직합니다.

이에, TAO 체인 구조에서는 각각의 기능별로 체인을 별도로 만들고, 전체적으로 이들 체인들이 상호 어떻게 혼용되고, 효율적으로 통신이 될수 있는 지에 대해서 구조를 만드는데 집중합니다. 또한, 이를 위한 단위 요소기술의 조사, 개발, 인티그레이션 하게 됩니다. 현재 TAO 에서 주요 체인들의 명칭과 특성은 다음과 같습니다.

- **TAO** : 궁극적으로 콘텐츠,IOT 플랫폼을 추구함. 블록체인의 특징별 조합 , State 연계, Scalability를 위한 전체적인 타오 블록체인 구조와 기본 플랫폼을 통칭함.
- **TEO** : 타오 체인의 POW 체인을 지칭, 초기에 블록체인 Contents Platform의 Backend 체인으로 사용되고, 기존 이더리움의 채굴 알고리즘을 변경하여, ASIC 채굴저항성을 가지도록 함.
- **TEOS** : 타오체인의 POS 체인을 지칭. – DAPP 특화.
- **TAOA** : 타오체인의 POA 체인을 지칭. – DAPP 특화.
- **iTAO** : 타오체인의 IoT, M2M 용 LiteChain 을 지칭.
- **TAM** : 타오전체 구조에서 IoT M2M 컨트랙 및 응용을 실행하기 위한 (TAO Trust Application Machine) 을 지칭.

2.2 TAO 체인의 목적

TAO는 여러개의 블록체인이 각각 제 기능을 수행하면서 각 체인을 동시에 연계시키는 블록체인 아키텍처입니다.

TAO에는 각각의 사물과 객체들이 1) 각각의 분권적 철학과, 2) 각각의 객체의 존재의 의미와 , 3) 각각의 행위의 기본적인 방식이 존재한다고 가정합니다.

따라서, 이를 블록체인의 구조에 접목했을시, 각 블록체인의 담당하는 것들에 특이한 강점들이 있고, 또한, 4) 블록체인 자체도 라이프사이클이 존재하는 것을 기본 개념으로 정의합니다.

따라서, 타오 블록체인 아키텍처에서는 위의 1)~4) 의 개념을 기본으로 블록체인 아키텍처를 구현함에 첫 번째 목적으로 정의합니다.

TAO블록체인의 목적은 위의 구조를 적용하는 방법을 기본으로, 현실의 블록체인의 스칼라빌리티(확장성) 문제를 해결하고자 하는것입니다.

하나의 블록체인으로 모든것을 만족 할수는 없습니다. 또는 모든 데이터가 영구적으로 진행될 필요는 없습니다.

이에, TAO는 블록체인의 라이프사이클과 라이프타임으로 제시하고, 이를 통해서 확장성문제 해결과 네트워크 트래픽을 감소할 수 있는 효율적 구조를 지향합니다.

또한, 아주 경량의 IoT 및 Sensor 디바이스에도 블록체인을 활용한 M2M 이 가능한 체인이 존재하여야 타오에서 보는 IoE(만물 인터넷)에 적합한 구조가 되므로, 그러한 체인이 필요할것이고, 이러한 체인에서 동작하는 TAM 이 필요하게 됩니다.

2.3 TAO 체인의 장점

2.3.1 기존 컨트랙 DDOS 사례탐구를 통한 개선방안 도출안

사례분석 1.
※ 2016.10월에 EVM 디도스 문제가 있었고 그것에 대한 해결방안중 한가지를 제시하고 있습니다. ^[2] https://gist.github.com/trustfarm-dev/cbbc9aa1ab76983cb7dbaa329097c1c3
문제 내용인용:
ETHEREUM 노드의 런타임 메모리가 16GB 가 넘어가고 있기에 Lite 한 IoT 시스템에 올리기에는 런타임 메모리 사용때문에 현실적으로 불가능한 상황이다. 굳이 한다고 해도, 원격지 서버 노드에 접속해서 클라이언트 기능만 넣을수있지, Full node 는 힘든 상황이다. 메모리 제한을 할수있는 새로운 방법이 있어야 하고, 이를 적용한 포크 노드가 나와야 한다. Gas 가격을 올리는 하드 포크를 하더라도, DDOS 를 막기는 힘들것이다. 단, DDOS 하려고 하는 자는 엄청난 가스 비용을 감내 해야 하는 상황인 것이지만, 다른 공격 루트는 또 찾을 것이기 때문이다.

하지만, Gas 비를 조정함으로써 DDOS 를 막는데는 상당한 도움을 줄것이다.
개선안 요약
TAO에서는 EVM 디도스에 대한 방안으로 가스비를 조정하여 (거래소처럼 실시간으로 가스비를 올렸다, 내렸다하여) 디도스를 막는 하나의 방법이 될수 있습니다. 이는 런타임 통계기반에 의한 가스비의 조정 방안을 제시한다.

2.3.2. TAM (TAO 어플리케이션 머신) 을 개발 적용.

EVM 은 현재 light 한 디바이스에 넣을수가 없습니다. EVM은 상위레벨의 추상화된 개념의 가상머신으로서 실제 CPU기반의 머신과 차이가 많습니다. 또한, 이러한 개념은 가상머신으로 많이 알려진 Java Virtual Machine 보다 더 추상화 되어있기에 그렇기도 합니다.

따라서, TAO 에서 제시하는 현실적인 접근방법은 실제 CPU 명령어에 기반한 Native 기계에 가까운 가상머신 또는 에뮬레이터를 만들고, 여기에 블록체인 API 및 Instruction 이 수행될수있도록 하자는 아이디어입니다. 기존 체인에서 EVM이외에 TAM 을 구동할수있고, 별도의 체인에서 이를 운용할 수도 있습니다.

또한, 더 나아가 라이트하고, 네이티브한 가상머신과 연계되는 라이트 체인을 만들어서 IoT 디바이스간 M2M도 블록체인에서 구현하고자 하는것입니다.

2.3.3 TAM 의 주요 구성규격.

TAM 의 구성은 작은, 저전력, 초소형 Micom CPU에 적용될수 있도록 기본 Instruction Set 을 선택하여 구성할 예정이며 , 실제로 구동되는 CPU의 코어 명령어세트를 기반으로 만들어서 좀 더 Machine 에 직접적인 동작이 수행되게 하여, 빠른 구동과 불필요한 컴퓨팅 리소스를 최소화하여 구현하게 됩니다.

이것에 적합한 CPU대한 몇가지 예시를 들자면, 기존PC 및 서버에 많이 사용되는 CPU인 X86 계열칩과 모바일 및 Embedded 기기에 많이 사용되는 ARM CPU의 경우, 라이선스가 상업적 회사들 소유로서 사용하기 어렵습니다.

따라서, 그러한 지적재산권 문제가 없는 OpenSource Free License 의 CPU 구조중 저전력 마이콤에 사용이 적합한 RISC 기반 InstructionSet CPU 명령어 체계를 사용합니다.

- RISC-V (<https://riscv.org/>)^[15] : CPU 코어 자체 오픈소스에 대한 연구하는 곳이며 라이선스가 없기에 이런 CPU Architecture 및 명령어 세트를 TAM에 맞게 적용할 예정입니다.
버클리대학 등 예전부터 연구되던 것으로 이 기반으로 실제 CPU 로 만들어서, 리눅스 및 RTOS 등이 잘 작동되고, 실제 이를 기반한 CPU 및 개발보드, GNU 기반의 Toolchain , Linux, RTOS 가 이미 검증되었고,

여러 반도체 회사들이 참여하고 있습니다.

- OPENRISC (<https://openrisc.io/>)^[16] : 이 또한, 위의 RISC-V 처럼 OpenSource 및 FreeLicense 로서 연구 및 개발되고 있는 CPU Architecture 입니다.

2.4 TAO 체인에서 사용되는 블록체인 노드구조.

2.4.1 기존 PoW / POS / POA 체인 클라이언트 노드 구조변경.



Figure 4 Comparison with TAO client node

TAO 구조에서는 기존 EVM 이외에 TAO Lite 체인에 사용될 TAM 이 추가됩니다..

2.4.2 iTAO LiteChain 에서 적용될 iTAO 클라이언트 노드

Figure 5 에 iTAO Lite 클라이언트의 구조도를 개략적으로 도면화 하였고, 이는 컨트랙과 Lite 한 M2M iDAPP 을 구동하기 위한 TAM Emulation Layer 가 존재합니다. 이는 유사한 RISC 기반의 CPU 에 최적화 될수 있도록, 가상머신이 아닌, 좀 더 가벼운 Emulation Layer 가 존재하고, 동작 instance 도 1개의 instance 로 단일화 하게 됩니다.

경량 IoT 기기에서 TAM 을 구동하기 위해서는 같은 속성의 TAM API가 동작될수 있도록 Emaultion Layer 를 사용합니다.

iDAPP (경량 IoT DAPP) 은 TAM API 스택위에서 앱이 수행되게 되고, Legacy App 은 본래의 IoT 기기에서 수행되는 본연의 앱을 의미합니다.

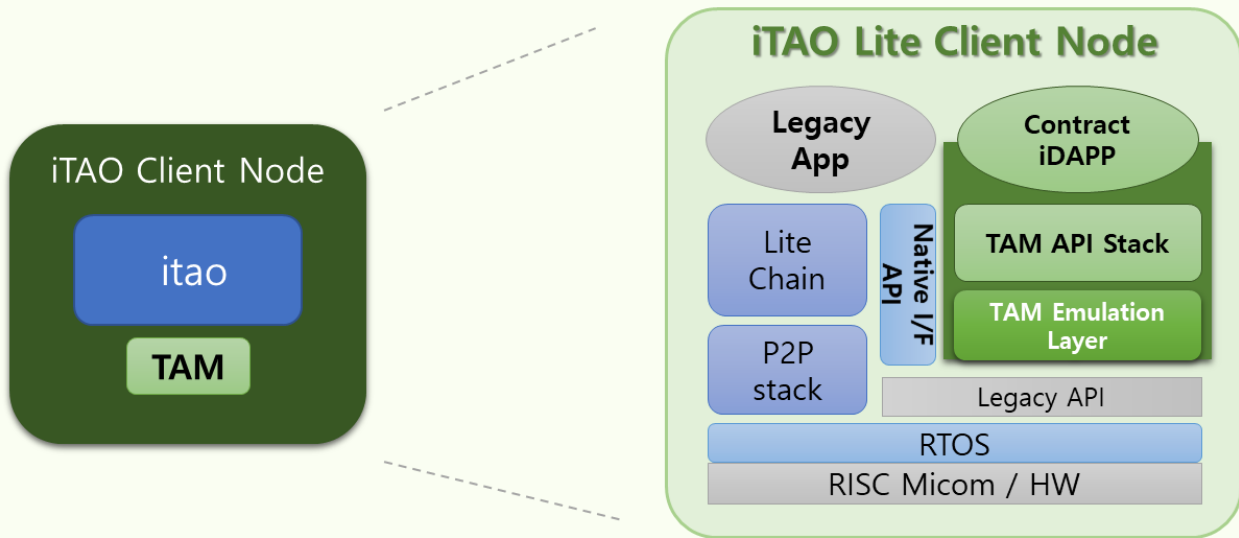


Figure 5 iTAO Lite client and TAM diagram

2.4.3 TAO Full Chain 에서 적용될 클라이언트 노드

Figure 6 에 TAO Full 클라이언트의 구조도를 개략적으로 도면화 하였습니다. 이는 컨트랙과 Lite 한 M2M iDAPP 을 멀티로 구동할수 있는 컴퓨팅 리소스가 충분한 시스템에서 동작됩니다.

이때 TAM 의 경우는 Emulation Layer 가 아닌, TAM RISC CPU를 그대로 가상화한 가상머신 레이어가 존재합니다. 따라서, TAM 용 Lite OS 인 RTOS (Realtime OS) 가 탑재되고, 이 기반위에서 TAM API Stack 이 위치합니다.

또한, 독립적인 기계에 OS 가 구동되는 환경이기 때문에, 여러 개의 iDAPP 이 동작될수 있도록 설계됩니다. 독립적인 프로세스 방식이 아닌, 멀티쓰레드 방식으로 iDAPP 을 수행할수 있는 체계를 제공합니다.

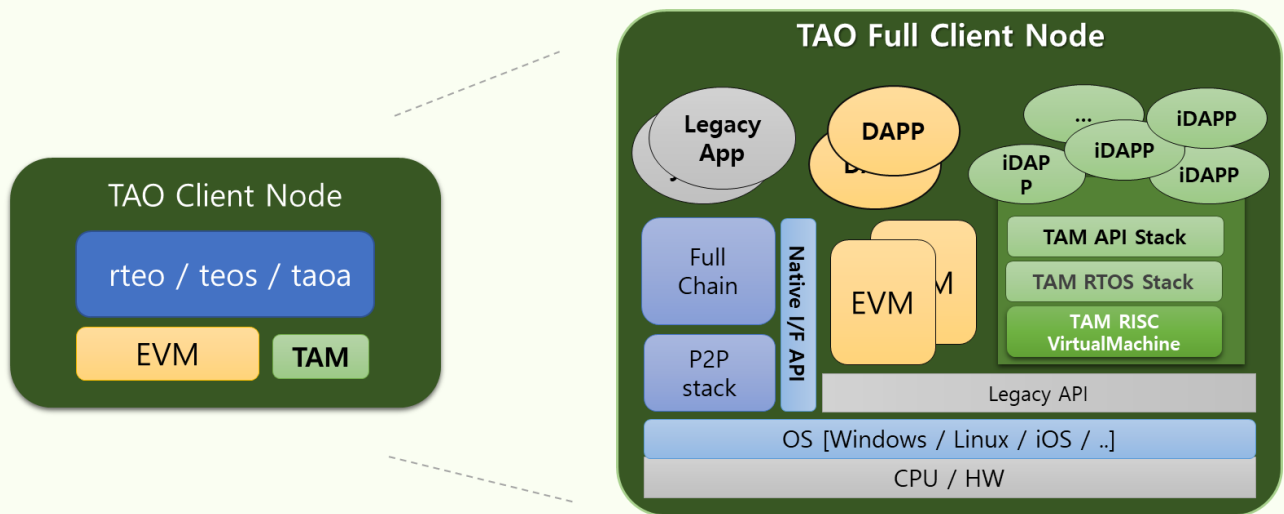


Figure 6 TAO Full client and TAM diagram

2.5 TAO 체인에서 콘텐츠 플랫폼으로서의 방안.

2.5.1 콘텐츠 플랫폼 Backend 로드맵.

TAO에는 블로깅 이나 음악, UCC, 포럼 등에 보상을 줄수 있는 플랫폼을 구성할것이며, 이를 위해서, 1차적으로 PoW 기반의 체인을 도입하여 블록체인 코인 Backend 로서 TEO (TrustEthReOrigin) PoW 체인을 시작합니다.

2차 시기에는 POA , POS 체인을 도입하여, 트랜잭션 속도를 높이고, 컨펌시간을 줄이며, 코인 Backend를 이동하는 로드맵을 추진하고 있습니다.

이렇게 적용하면 블럭타임도 줄고, 트랜잭션 스피드도 늘어나며 에너지도 절약되고, 콘텐츠 애플리케이션의 동작이 빨라지는 장점이 존재합니다.

2.6 DPOS/POS 계열 검증인 노드 보안성 강화 방안.

POS/POA 체인에서는 여전히 블록체인 1)보안관점에서는 POW에 비해서 고려할 점이 많이 있고, POS의 가장 기본적인 문제점인 2)부자는 더 부자가 되고 가난한 사람은 더 가난해 지는 것에 대한 단점이 있습니다.

위 1) 2) 두가지의 문제를 해결하기 위해 코인의 스테이킹에 있어서, 이자 지급방식과, 최소,최대 Stake 발

란상에 제한을 두는 합의안을 제안합니다.

2.6.1 기존 컨트랙 DDOS 사례탐구를 통한 개선방안 도출안

제안 내용 사례분석2
<p>※ DPOS 네트워크 보안리스크를 완화하고 중앙화를 막는 consensus 개선방안^[7] https://github.com/trustfarm/dev-documents/blob/master/HowToMake-DPOS-consensus-network-more-secure.md)</p>
내용 요약 인용:
<ul style="list-style-type: none"> ➤ 검증인 노드에 Hardcap/Softcap 의 fund 상한선을 정하는 컨센서스 정책으로, 한도 이상의 Fund 노드에는 Cap 이상의 Fund에 대한 이자는 지급하지 않는다. ➤ 최소 하한의 검증인 노드의 fund 를 지정한다. ➤ 최소 지정한 검증인 노드를 전체 목표노드의 150% 이내에서 여분으로 확대가능하고, 이를 논리적으로 분산시킬수있도록 한다. ➤ 검증인 노드가 위치한 물리적 네트워크를 다양하게 분산하여야 한다. 이는 IP를 통한 서버의 geolocation 정보를 기반으로 같은 네트워크 IDC 세그먼트에는 전체 검증인 노드중 10% 이상의 노드가 참여하지 못하게 한다. 10%라는 비중은 상대적이므로, 이는 현실적인 적용상황에서 다소 조정이 가능하다. 위의 경우는 50개 이상 노드가 참여한다 할경우를 기준으로 경험적으로 가이드를 잡은 것이다. ➤ 반드시 네트워크가 안좋은 지역의 geolocation 및 IDC 네트워크노드도 10% 정도 포함하게 한다. 이는 기회의 균등의 의미도 있지만, 네트워크 상황이 물리적으로 약한 경우, DDOS 자체가 10% 노드에 공격을 쉽게 못한다는 의미도 된다.
개선안 요약
<ul style="list-style-type: none"> ➤ 밸리데이터를 최대한 많이 두되 몇몇 집단이 독점하지 않고 넓게 퍼져야 한다. 즉, 노드를 지역적으로 분산시켜야 합니다. 예를 들어 한국에 밸리데이터가 10명있다고 보다는 몇 명씩 한국 , 미국, 중국 등 지역적으로 최대한 다양하게 분산되어야 합니다. 서버를 지역별로 구분할 필요가 있습니다. ➤ 필요이상의 코인이 Stake 로 Balancing 되지 않게 하여, 거래소 및 실제 적용되는 일반 거래에 코인의 유동성이 공급되도록 하여, 코인의 생태계를 활성화 시키는 것이 필요하다. ➤ Fund 의 분산을 통해서 P2P 들의 참여폭을 확대하고, 이를 기반으로 블록체인의 활성화, 가치를 상승시킨다.) ➤ 검증인 노드의 분산화, 분권화를 통하여, 블록체인 네트워크 자체의 보안성을 강화한다. ➤ 요약 : POS나 DPOS가 노드들에 대한 디도스에는 대단히 취약^[9] 하고, 이때, 여러 가지 공격 Vector 가 존재하므로, 근원적으로 분산 하는 것이 필요하다.

➤ DPOS 검증인 노드의 분산화, 분권화를 통하여, 블록체인 네트워크 자체의 보안성을 강화한다.
추가 배경 이론
<ul style="list-style-type: none"> ➤ 속도 보다는 DPOS 자체의 보안성을 높이겠다는 취지입니다. POS나 DPOS는 POW 보다는 트랜잭션 속도가 월등히 빠릅니다. ➤ 또한 TPS 만 강조하여, 블록타임을 0.5초 정도로 빠르게 하는것은, 검증인 노드를 최소화 하고, 초고속 하이스피드 밸리데이터 노드로 서버를 구성하고, 빠른 인터넷망에 노드를 두게 하면 이론적으로 기존 비자 속도도 가능합니다. 그런데 결국 이것은 중앙화 분산서버 개념과 큰 차이가 없으므로, 여기에 블록체인으로서의 의미를 부여하기는 어렵습니다. ➤ 실제로 분산을 시켜서 보안성과 사용성이 높아지는지가 중요합니다. 예를 들어서 수개의 아주 적은 노드로 블록을 채굴한다고 가정할경우, 이더리움 체인의 블록 타임이 3~20초 사이로 들쭉 날쭉 하게됩니다. PoW 의 특성에 따라서 그렇게 됩니다. 난이도가 낮고 노드가 적으면 이렇게 되지만, 분산이 되고, 채굴 노드의 수가 적정수 이상되면 원래 이더리움 타겟 블록타임인 14.4초 평균에 수렴하게 됩니다. ➤ 물리적으로 노드를 적게 해버리면 POW 체인이라도 난이도를 낮게해서 블록타임을 3초로도 운영이 가능합니다.

TAO 에서는 우선적으로 POA / POS 체인을 도입함에 있어서, TPS 보다는 보안성 우선방안으로 컨센서스를 수정하고 운영합니다. 또한, 현재 까지 제기된 취약점과 우려점에 대해서 이를 극복할 방안과 보완할 방법을 TAO 블록체인 구조에서 제안하고 있습니다.

2.7 체인별 스테이트 공유 및 인터랙션 방안

- A. 아토믹 스왑^[21] : 가치를 같이 가져가기 때문에 체인별 1:1로 스왑하면 됩니다.
(하위 체인은 체인별로 비율을 별도 조정하여 적용가능합니다.)
- B. 이더 기반 체인끼리는 스마트컨트랙 기반으로 해서 메시지로 주고 받는 방법도 있습니다.
(칼리스토에서 메시지를 이용하여 체인별 가치 이동방안을 적용중입니다.)
- C. 패러티에서는 포카닷^[18] 의 브릿지^[19] 의 개념을 이용하였습니다.
- D. 리플의 경우 지역별로 게이트웨이를 두는 방식과 유사하며 이는 코스모스의 존 개념과 비슷하다고 할수 있습니다.
- E. 기존에 컨트랙이 불가능한곳은 게이트 웨이를 적용하고, 컨트랙이 가능한 곳은 컨트랙 기반으로

메세지 처리를 하면 됩니다. (실제 이렇게 하는 어카운트가 존재합니다.)

- F. 블록스트림에서 제시한 2웨이 사이드 체인닝^[22] 도 있습니다.
- G. 텍스(DeCentralized Exchange)통해서 교환을 하여도 됩니다. 이러한 DEX 서비스는 3rd party 업체에서 제공해도 됩니다.

※ 요약 : 기존에 인터렉션 방법 과 기술에 대한 검증은 충분히 되어있기 때문에 TAO에서 이를 어떤방식이 효율적인지 판단하고, 기술을 채용하여, 타오에서 이용하게 됩니다.

3. TAO 블록체인 구조의 동작 프로세스

3.1 TAO 아키텍처 세부구조

상기에서 기술한, TAO 체인의 이해를 바탕으로, 구체적인 블록도로 도식화 하면 다음과 같이 구성될수 있습니다. 이해를 돕기위해서 표시한 주요 블록체인인 TEO , TEOS , TAOA , iTAM 외에 필요에 의한 사이드 체인^[22]이 존재하게 되고, 이는 각각의 로컬노드에서 독자적으로 체인닝되고, 필요성이 존재 하지 않은 시점이 되면, 블록체인 라우트 인터페이스에 있는 Blockchain Lifecycle manager 가 판단해서, IPFS^[29] 형태의 P2P 분산 저장소에 보존되게 됩니다. 이 보존된 블록체인 데이터베이스의 해시값은 기존 부모체인, 블록도에서는 TEO 체인의 신규 생성되는 블록에 기록이 되게 되고, LifeCycle Manager에서 별도로 인덱스 및 URI를 기록하고, 현재의 블록체인 State [Alive,Pause] 등의 정보를 기록하게 됩니다.

또한, 블록체인 토큰 상태교환을 위해서, Token Swap 인터페이스가 존재하게 됩니다. Token Swap 이 됨에 따라서, 블록체인 상호간의 분산처리 및 상태의 공유가 가능하게 되기 때문입니다.

현재, 운영중인 부모체인에서 독립적인 Sidechain 이 발생할경우, 이의 상태도 교환할수있도록 Token Swap 영역이 담당하게 TAO 구조에서는 정의 합니다. 별도의 Sidechain 의 경우, 부모체인의 블록체인 기본속성을 그대로 수용하게 되므로, 트랜잭션을 사용하지 않을경우에는 불필요한 트랜잭션을 소비할수 있습니다. 이는 확장성(Scalability) 과 효율성을 위해 만든 구조의 목적에 부합하지 않은 경우가 존재할수 있습니다. 이를 효율적으로 방지하기위해서, 초기 사이드 체인이 생성시에는 기본적인 Lifetime [시간 및 블록증가번호] 을 기본값으로 정하도록 합니다.

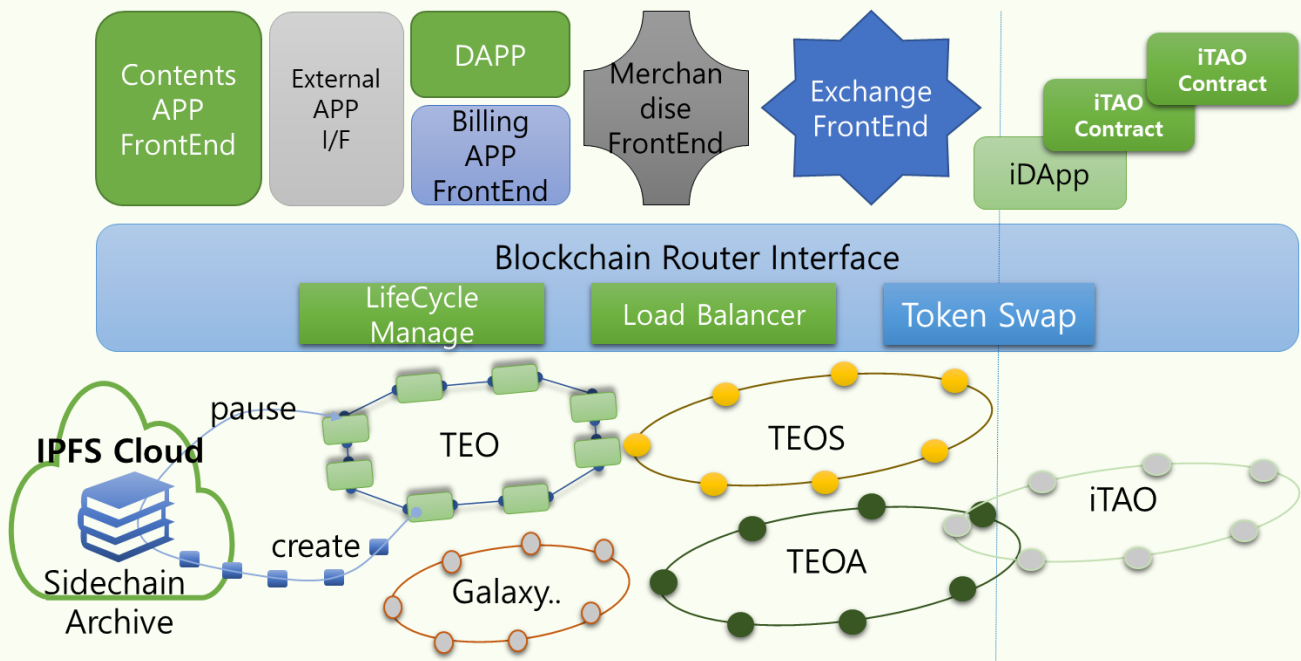


Figure 7 TAO Architecture block diagram

3.1.1 TAO 구조 응용 레이어

Figure 7 의 구조도에서 상위 레이어를 지정합니다. 이 응용단은 기존 블록체인 분권환앱(DAPP) 및 TAO 블록체인 구조에서 제공하는 여러 서비스들을 연속적이고, 투명하게 접근하고 사용할수 있습니다.

이는 기존의 특정된 블록체인 기반에서만 서비스가 가능한 것이 아닌, TAO 구조안에서 운용되는 어떤 블록체인이던지 각각의 특성화된 서비스를 공통적인 Interface를 통하여 이용할 수 있는 체계입니다.

따라서, 기존의 블록체인의 트래픽, 기능적 스케일러빌리티의 한계를 극복할 수 있는 구조입니다.

위의 TAO 구조안에서, 마이콤 수준의 작은 CPU 를 활용하는 응용프로그램이던, 대용량 서비스가 필요한 응용프로그램이던 연속적으로(Seamless) 하게 응용서비스를 구현할수 있습니다.

3.1.2 TAO 블록체인 라우터 인터페이스

TAO 구조의 가장중요한 운영체제와 관련된 내용들입니다. 이는 하부의 블록체인 Backend 와 응용단과의 유연하고, 연속적인(Seamless)한 정합성을 만들어 주기위한 일종의 Middleware 서비스 집합체 입니다. 가장 대표적으로 다음과 같은 기능을 담당합니다.

- ① Token Swap : 2장에서 논의된 바와 같이, 가장 중요한 역할을 각각의 체인의 토큰 상태의 체인간 공유와 교환을 담당합니다.
 관련하여, 다양한 3rd Party 분권화거래소, Atomic Swap 기술, 메시지 기반의 컨트랙을 통한 State 교환 방법등의 다양한 기술들이 존재합니다.
 TAO 에서는 이러한 다양한 방법을 활용하여, Token Swap 서비스를 제공하게 됩니다.
- ② Load Balancer : 현재 블록체인의 기술적인계점으로 지적되는 TPS (트랜잭션처리능력) 및 한 개의 AllinOne 체인의 문제를 극복하기 위해서, 각각의 체인의 TPS 상태를 모니터링하여, 앱의 요청사항을 실시간으로 분산해주는 서비스와 기능을 담당합니다. 여기서는 요청하는 DAPP 에 현재 요청하는 기능을 제공하는 체인의 혼잡도를 앱에게 제공하여 주는 기본서비스를 비롯해, 씬리스크한 서비스를 제공하게 됩니다.
- ③ LifeCycle manage : 각각의 체인 및 SideChain 은 기본적인 LifeTime 을 정의 합니다.
 여기서는 특히, 필요에 의해서 생성된 사이드 체인의 LifeCycle 을 관리하게 됩니다.

즉, 필요에 의해서, SideChain 이 생성되고, 사이드 체인에서의 블록체인 스테이트 변경이 발생하여, 모체인에 적용을 Token Swap 서비스를 통해서 하게 되고, 향후, SideChain 이 필요하지 않은 경우는, 기존 Side Chain 을 IPFS 와 같은 분산 P2P 저장소에 Archive 하게 되고, 사이드 체인의 최종 스테이트를 메인체인에 적용하고, 최종 IPFS Archive 의 URI (저장위치) 정보를 메인체인의 현재 블록에 기록하게 됩니다.

이는, 사이드체인의 소멸이 아니고, 중지 상태를 의미합니다. 향후, 필요에 의해서 다시 사용할수 있기 때문입니다.

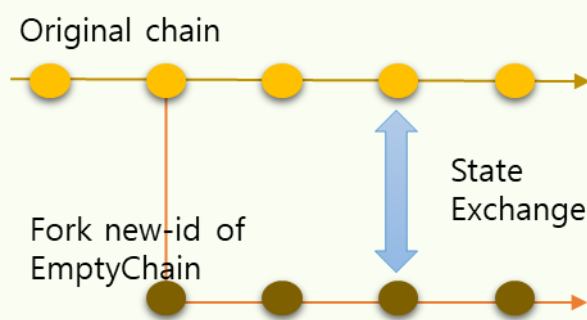


Figure 8 Scaling with Sidechain fork method

Figure 8 은 사이드체인 포크를 활용한 확장성 해소를 위한 방안을 블록도로 표현한것입니다. 사이드체인이 부모체인에서 포크되어서 생성될시, 부모체인의 속성을 그대로 유지하게 됩니다. 단,

Token 상태는 기존의 부모체인에서 포크시 그대로 상속하지 않고, 초기화된 체인형태로 포크됩니다. 스케일링을 원하는 트랜잭션에 해당하는 상태는, 상태교환을 통해서, 사이드체인으로 이동하게 되고, 사이드체인에서 관련된 트랜잭션을 수행하게 됩니다. 중간에 사이드체인에서 상태를 더 이상 사용할 필요가 없는 경우에는 다시, 부모체인으로 상태를 이전하게 됩니다.

3.1.3 TAO 블록체인 스페이스

Figure 7 에서 보듯이, TAO 블록체인 공간에는 여러 종류의 블록체인이 존재합니다. 또한, 각각의 블록체인은 사이드 체이닝 및 복제된 블록체인을 이용해서 여러 종류의 블록체인이 존재하게 됩니다.

그림에서 표현된 "Galaxy.." 는 기존 TAO 에서 제공되는 블록체인 이외에 외부의 존재하는 이기종의 블록체인을 의미합니다.

우리는 Token Swap 및 외부 3rd Party 의 분권화 거래소(DEX : DeCentralized Exchange) 및 기존 중앙화 거래소의 토큰 교환서비스를 이용해서, 각각의 블록체인과의 상태교환을 수행할 수 있습니다.

★ TAO 블록체인 유니버스에서는 어떤형태의 신규 블록체인과의 소통이 가능한 구조를 목표로 하고 있습니다.

꿈 ★ 은 이루어진다!

References

- 1) Bitcoin White Paper
<https://bitcoin.org/bitcoin.pdf>
- 2) trustfarm :: Ethereum Contract GasFee Autochange Idea. review on DDOS attacks.
<https://gist.github.com/trustfarm-dev/cbbc9aa1ab76983cb7dbaa329097c1c3>
- 3) trustfarm :: HowToMake-DPOS-consensus-network-more-secure
<https://github.com/trustfarm/dev-documents/blob/master/HowToMake-DPOS-consensus-network-more-secure.md>
- 4) trustfarm :: Storage Scalability - Proposal on Parity DBOptions change
<https://github.com/paritytech/parity-ethereum/issues/9033>
- 5) Smart Contract
https://en.wikipedia.org/wiki/Smart_contract
- 6) Ethereum White Paper
<https://github.com/ethereum/wiki/wiki/White-Paper>
- 7) Ethereum Yellow Paper (EVM Spec)
<https://ethereum.github.io/yellowpaper/paper.pdf>
- 8) PoS:: Several Proof of Stake FAQs
<https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs>
- 9) PoS:: Stake grinding attack discussion
<http://block-chain.jp/blockchain/proof-of-stake-stake-grinding-attack/>
- 10) RocksDB :: A fork of LevelDB for High performance embedded database for key-value data.
<https://en.wikipedia.org/wiki/RocksDB>
- 11) Sharding :: Facebook - RocksDB Bloom Filter
<https://github.com/facebook/rocksdb/wiki/RocksDB-Bloom-Filter>
- 12) Storage Scalability :: Facebook – RocksDB Tuning Guide
<https://github.com/facebook/rocksdb/wiki/RocksDB-Tuning-Guide>
- 13) RISC :: Reduced instruction set computer
https://en.wikipedia.org/wiki/Reduced_instruction_set_computer
- 14) CISC :: Complex instruction set computer
https://en.wikipedia.org/wiki/Complex_instruction_set_computer
- 15) RISC-V :: The Free and Open RISC Instruction Set Architecture
<https://en.wikipedia.org/wiki/RISC-V> , <https://riscv.org>
- 16) OpenRISC :: a free and open RISC instruction set architecture with DSP features

<https://en.wikipedia.org/wiki/OpenRISC> , <https://openrisc.io/>

- 17) POA Chain (Proof-of-Authority Chains)
<https://wiki.parity.io/Proof-of-Authority-Chains>
- 18) Polkadot :: the heterogeneous extensible multi-chain White Paper
<https://github.com/polkadot-io/polkadot-white-paper/blob/master/PolkaDotPaper.pdf>
- 19) Parity Bridge between any two ethereum-based networks
<https://github.com/paritytech/parity-bridge>
- 20) Cosmos ATOM White Paper
<https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md>
- 21) Atomic Swap
<https://github.com/decred/atomicswap>
- 22) Side-Chain :: Enabling Blockchain Innovations with Pegged Sidechain
<https://blockstream.com/sidechains.pdf>
- 23) 스마트 계약- 컨트랙트 이해
https://ko.wikipedia.org/wiki/%EC%8A%A4%EB%A7%88%ED%8A%B8_%EA%B3%84%EC%95%BD
- 24) EVM [모두의 연구소] Introducing Ethereum and Solidity 3장 요약본 관련 이해
<https://steemkr.com/kr/@tenihil/introducing-ethereum-and-solidity-3>
- 25) Ethereum Virtual Machine (EVM) 개요
<https://opentutorials.org/course/2869/18360>
- 26) 이더리움과 샤딩 (Ethereum And Sharding)
<https://steemit.com/kr/@kaipark/ethereum-and-sharding>
- 27) 샤딩은 어떻게 동작하는가 (How Sharding Works 번역) 관련이해
<http://eminentstar.tistory.com/54>
- 28) [DB] DB Sharding은 무엇이고, 적용 전략은? (적용시 고려사항)
<http://hanburn.tistory.com/106>
- 29) IPFS Docs, WhitePaper - IPFS - Content Addressed, Versioned, P2P File System
<https://docs.ipfs.io/>
<https://github.com/ipfs/ipfs/blob/master/papers/ipfs-cap2pfs/ipfs-p2p-file-system.pdf>
- 30)