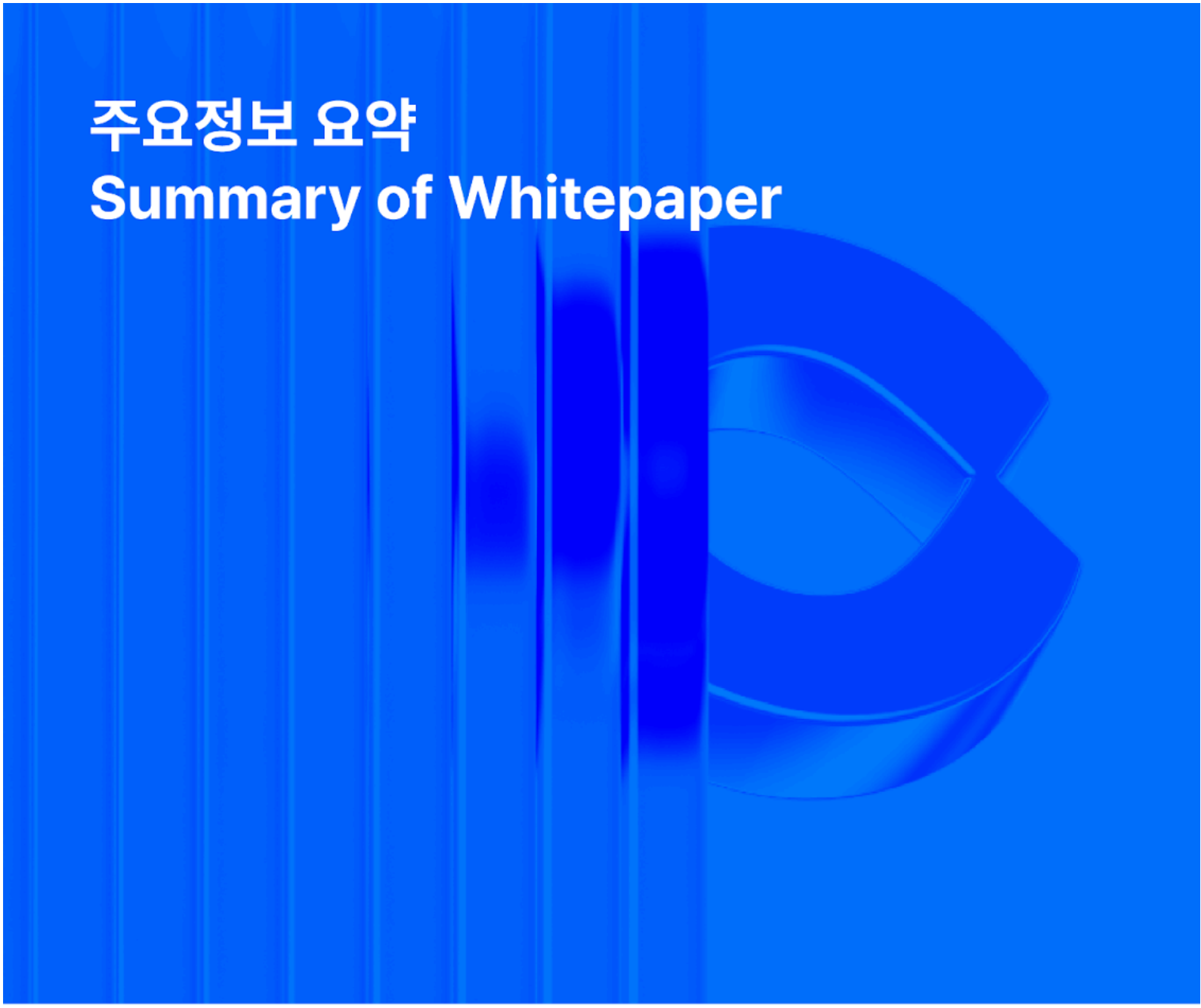
주요정보 요약
Summary of Whitepaper



본 문서는 거래지원 가상자산 백서의 주요 내용을 한글로 설명한 주요정보 요약입니다.
코인원은 거래지원 가상자산의 주요정보 요약을 주기적으로 점검하여 가능한 한 최신 정보를 제공할 예정입니다.

coinone

기본 정보

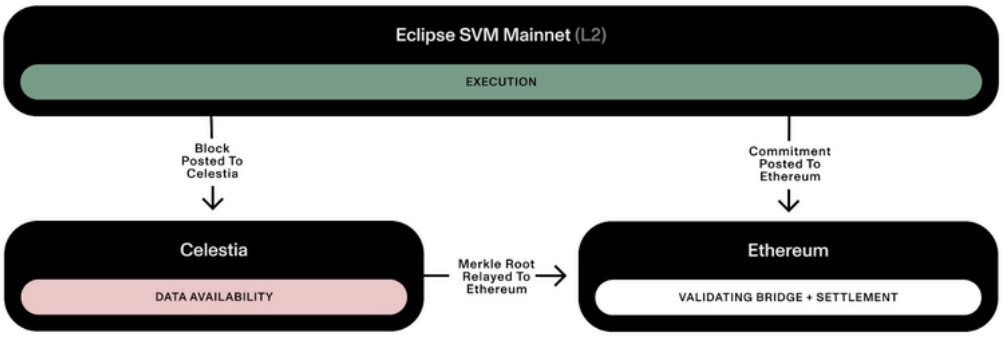
가상자산 카테고리	유틸리티
거래지원 네트워크	Ethereum
홈페이지	https://www.eclipse.xyz/
참고문헌 (백서, Docs 등)	https://docs.eclipse.xyz/

1. 프로젝트 정보

이클립스 메인넷이란?

Eclipse 메인넷은 Ethereum 기반 최초의 솔라나 가상 머신(Solana Virtual Machine, SVM) L2입니다. Eclipse 메인넷은 모듈형 블록체인 스택의 장점들을 결합한 구조를 가지고 있습니다.

- 결제(Settlement): Ethereum
  Eclipse는 Ethereum을 결제 계층으로 사용합니다. 즉, 공식 검증 브릿지는 Ethereum에 구축되며, 트랜잭션 수수료로 ETH를 사용합니다.
- 실행(Execution): Solana Virtual Machine (SVM)
  Eclipse는 고성능의 SVM을 실행 환경으로 채택합니다.
- 데이터 가용성(Data Availability): Celestia
  Eclipse는 확장 가능한 데이터 가용성을 위해 데이터를 Celestia에 게시합니다.
- 증명(Proving): RISC Zero
  Eclipse는 중간 상태 직렬화 없이 RISC Zero를 활용하여 사기 증명(Fraud-proof)용 영지식(ZK) 증명을 수행합니다.



결제 계층: 이더리움 (Settlement: Ethereum)

현재 주요 롤업들과 마찬가지로, Eclipse 메인넷은 Ethereum을 결제 계층으로 사용합니다.

Eclipse 메인넷의 주요 기능과 보안 이점

Validating Bridge 통합

Eclipse 메인넷은 네트워크 인프라 내에 직접 구축된 validating bridge를 포함하고 있습니다. 이 브리지는 모든 Eclipse 노드가 신뢰하는 "정본 체인(canonical chain)"을

구성하는 데 핵심적인 역할을 합니다.

병렬 Ethereum 풀 노드 운영

정확한 트랜잭션 순서를 보장하기 위해, Eclipse 노드는 Eclipse 노드와 함께 Ethereum 풀 노드도 병행 실행해야 합니다. 이 통합은 Ethereum의 보안성을 Eclipse 네트워크로 가져옵니다.

보안 기능

- Validating bridge는 모든 트랜잭션을 인증하여, 잘못된 상태가 Eclipse 네트워크에 제출되는 것을 방지합니다.
- 시퀀서가 실패하거나 검열을 시도하는 경우에도 네트워크가 계속 작동하도록 하는 eventual liveness를 보장합니다.
- L2 수준에서 검열이 발생하더라도 사용자가 브리지를 통해 강제로 트랜잭션을 제출할 수 있어 검열 저항성을 제공합니다.

Ethereum L2 분류

Eclipse 메인넷은 Validium 및 Optimum과 유사한 형태로 Ethereum Layer 2 네트워크로 분류됩니다.

Ethereum을 기본 화폐로 사용

DeFi 및 NFT 시장에서 Ethereum 기반 자산의 중요성을 고려할 때, Eclipse 메인넷은 탈중앙화된 통화로 ETH를 선택하였습니다. ETH는 네트워크에서 발생하는 트랜잭션 수수료로도 사용됩니다.

향후 수수료 유연성

향후에는 수수료 추상화(Fee Abstraction)를 도입하여, USDC와 같은 토큰으로도 가스 수수료를 지불할 수 있도록 할 계획입니다.

실행: 솔라나 가상 머신 (Execution: Solana Virtual Machine (SVM))

가상 머신(Virtual Machine, VM) 기초

블록체인에서의 가상 머신(VM)은 트랜잭션을 처리하기 위해 기계 명령어를 실행하는 소프트웨어입니다. 트랜잭션이 블록체인에 도달하면, 해당 트랜잭션은 가상 머신에 의해 처리됩니다. Solidity, Rust, Move 등 어떤 언어로 작성되었든, 최종적으로는 바이트코드(bytecode)로 컴파일되어 블록체인의 가상 머신에서 실행됩니다.

언어마다 컴파일 가능한 바이트코드는 다른데, 예를 들어, Solidity는 일반적으로 Ethereum Virtual Machine (EVM) 바이트코드로 컴파일되지만, Solang 컴파일러를 사용하면 WebAssembly (WASM) 또는 Berkeley Packet Filter(BPF)와 같은 다른 바이트코드로도 컴파일할 수 있습니다.

가상 머신 비교

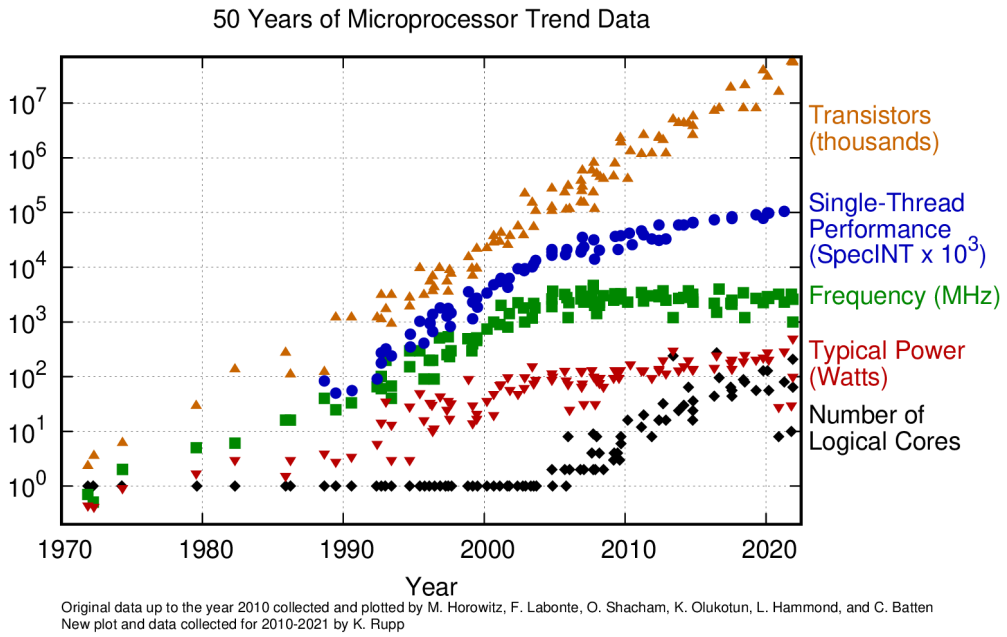
가상 머신은 다양하게 존재합니다. 어떤 가상 머신이 가장 적합한지는 다음 기준을 통해 판단할 수 있습니다.

- 성능(Performance)
  일부 가상 머신은 트랜잭션 간 병렬 처리를 가능하게 설계되어 있습니다. 예를 들어, Eclipse가 사용하는 SVM은 병렬성을 극대화하도록 설계되었습니다. 반면 EVM은 일반적으로 단일 스레드 기반입니다.
- 보안(Security)
  Rust와 같은 언어는 Solidity가 방지하지 못하는 다양한 버그로부터 보호하기 쉬운 특성이 있습니다. 예를 들어, Ethereum의 스마트 컨트랙트는 재진입 공격(reentrancy attack)에 취약합니다.
- 커뮤니티(Community)
  Ethereum과 Solana는 각각 EVM과 SVM을 중심으로 활발한 개발자 커뮤니티가 형성되어 있어 툴링과 개발자 지원 측면에서 유리합니다. 반면, Move VM이나 Fuel VM과 같은 신규 VM은 아직 생태계가 성숙하지 않았습니다.
- 사용 편의성(Ease-of-use)
  Solidity는 비교적 쉽게 코딩할 수 있으며, 모든 바이트코드가 Solidity에서의 컴파일을 지원하는 것은 아닙니다.

Eclipse 메인넷은 SVM을 실행하며, Solana CLI나 Seahorse Lang과 같은 기존 SVM 개발 도구도 그대로 사용할 수 있습니다.

병렬 실행 최적화

SVM과 그 런타임인 Sealevel은 병렬 트랜잭션 실행을 가능하게 하는 것으로 잘 알려져 있습니다. 상태(state)가 겹치지 않는 트랜잭션은 순차적으로가 아니라 동시에 실행될 수 있습니다. 이 구조는 하드웨어가 더 많은 코어를 더 낮은 비용으로 제공할수록 SVM이 직접적으로 확장될 수 있게 합니다. 반면, 현재의 EVM과 같은 단일 스레드 런타임은 코어당 비용이 낮아져도 이점을 제대로 활용할 수 없습니다. 지난 10년간 단일 스레드 성능 개선은 점차 둔화되고 있으며, 대부분의 성능 향상은 코어 수의 증가에서 기인하고 있습니다. 따라서 병렬 처리를 통해 이 추세를 최대한 활용하는 것이 중요합니다. 현재 EVM 병렬화를 시도하는 초기적이면서 실험적인 접근이 일부 존재하지만, 기존과의 호환성 유지와 관련된 트레이드오프가 존재하며, 성능이 최적화되지 못하거나 상태 증가(state growth) 등의 다른 병목을 해결하지 못합니다. SVM처럼 컨트랙트가 상태 의존성을 사전에 선언하는 방식은 병렬 처리에 최적화된 구조입니다.

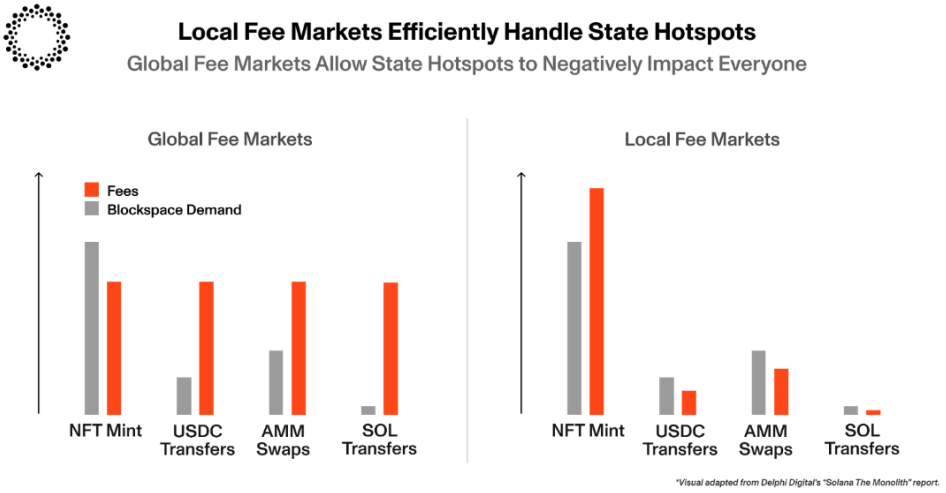


로컬 수수료 시장(Local Fee Markets)

대부분의 현재 블록체인 트랜잭션 수수료 시장은 글로벌하게 작동하여, 하나의 인기 애플리케이션이 모든 사용자에게 수수료 상승을 유발합니다. 예를 들어, 하나의 NFT 민팅이 전체 체인을 마비시켜서는 안 됩니다.

Solana의 로컬 수수료 시장 설계는 이러한 애플리케이션 간 상태 충돌 문제를 해결합니다. 현재 구현에서는 스케줄러가 충돌이 없는 트랜잭션을 우선 처리하여, 이들이 더 낮은 수수료로 실행되도록 합니다. 장기적으로는 로컬 수수료 시장이 프로토콜 수준에서 구현될 예정입니다. 이는 단일 앱의 수수료 급등이 체인의 다른 부분에 영향을 주지 않도록 보장합니다. 이러한 구조는 Solana의 병렬 처리 기반 런타임 덕분에 가능하며, EVM에서 상태 접근을 사전 선언하지 않고 휴리스틱 기반으로 로컬 수수료 시장을 구현할 경우, 비효율성 및 공격 가능성이 발생할 수 있습니다.

또한, 현재 각 애플리케이션이 자체적으로 생성하는 local value를 쉽게 내부화할 수 있도록 하는 초기 연구도 진행되고 있으며, 이는 현재까지는 앱 차원에서의 창의적인 설계가 요구되었습니다.



상태 증가(State Growth) 관리

EVM에서 병목 현상이 순차 실행으로 발생하기 전부터, 상태 증가가 훨씬 더 시급한 문제입니다. Solana는 전역 Merkle 트리를 사용하지 않기 때문에, 상태를 업데이트할 때마다 Merkle 트리를 갱신해야 하는 오버헤드가 없습니다. 대신 약 2.5일 간격의 epoch마다 전체 상태를 merklize합니다. 이는 EVM처럼 실시간 merklization을 요구하는 구조보다 훨씬 비용이 적습니다. 더 중요한 점은, EVM은 동적 계정 접근 방식을 사용하여 트랜잭션이 실행 중 언제든지 임의의 상태를 조회할 수 있습니다. 이로 인해 실행 전 상태를 메모리에 미리 적재할 수 없게 됩니다.

반면 SVM은 각 트랜잭션이 실행에 필요한 모든 상태를 사전에 명시하기 때문에, 상태 크기가 실행 성능에 영향을 주지 않습니다. 이러한 구조 덕분에, 검증 노드가 2년에 한 번씩 저장 장치를 업그레이드한다는 가정 하에, 스냅샷 크기를 2년마다 2배로 늘려도 안정적으로 운영이 가능합니다.

또한, Helius와 같은 팀들이 히스토리 데이터 접근성을 향상시키고, 압축을 통해 상태 크기를 줄이기 위한 노력을 활발히 진행 중입니다.

메타마스크 스냅 (MetaMask Snaps)

과거에는 EVM 사용자들이 비-EVM 체인에 온보딩되는 것이 큰 장벽이었습니다. 하지만 최근 발표된 MetaMask Snaps는 이러한 장벽을 허물고 있습니다. EVM 사용자는 기존의 MetaMask를 유지하면서도 새로운 체인과 상호작용할 수 있습니다.

Drift의 오픈소스 기여로 훌륭하게 구현된 MetaMask Snap 덕분에, 사용자는 EVM 체인과 유사한 사용자 경험으로 Eclipse 메인넷의 앱과 상호작용할 수 있습니다. 또한, 사용자는 MetaMask뿐만 아니라, Salmon과 같은 Solana 기반 지갑도 사용할 수 있습니다.

파이어댄서 (Firedancer)

Firedancer는 Jump가 개발 중인 고성능 Solana 클라이언트로, 네트워크의 처리량,

복원력, 효율성을 크게 향상시킬 것으로 기대되고 있습니다. Eclipse 메인넷은 초기에는 Solana의 기본 클라이언트와 최대한 동일하게 운영되지만, Firedancer 코드가 안정화되는 시점에 채택할 계획입니다.

보안(Safety)

Solana 런타임은 공격 표면이 현저히 줄어들어, 재진입 공격과 같은 악명 높은 취약점들을 방지합니다. Solana는 프로그램 간의 임의 재진입 호출을 허용하지 않으며, 자기 재귀(self-recursion)만 허용합니다. 또한, 상태와 코드를 분리하여 stateless code를 구현함으로써 테스트가 용이한 구조를 제공합니다.

더 간편한 증명 (Easier Proving)

SVM은 레지스터 기반이며 명령어 집합이 EVM보다 훨씬 작고 단순하여, ZK 기반 증명에 있어 SVM 실행이 더 용이합니다. 옵티미스틱 롤업의 경우에도, 레지스터 기반 구조는 체크포인팅(checkpointing)을 보다 쉽게 구현할 수 있도록 합니다.

데이터 가용성: 셀레스티아 (Data Availability: Celestia)

배경 지식으로서, 데이터 가용성(Data Availability)이란 무엇일까요? 데이터 가용성은 "이 데이터가 실제로 게시되었는가?"라는 질문에 답하는 개념입니다. 구체적으로는, 새로운 블록이 체인에 추가될 때 노드는 해당 블록의 데이터를 수신하여 데이터 가용성을 검증합니다. 노드는 새로운 블록에 포함된 모든 트랜잭션 데이터를 다운로드하려고 시도하며, 이를 모두 다운로드할 수 있다면 데이터가 실제 네트워크에 게시되었음을 증명한 것이며, 데이터 가용성을 성공적으로 검증한 것입니다.

Eclipse 메인넷은 Data Availability를 위해 Celestia를 사용합니다

Eclipse 메인넷이 목표로 하는 처리량과 수수료 수준은 아쉽게도 현재 Ethereum의 대역폭으로는 지원되지 않습니다. 이 문제는 EIP-4844(일명 프로토-댕크샤딩(Proto-danksharding))가 도입되더라도 여전히 해결되지 않습니다. EIP-4844는 평균적으로 블록당 약 0.375 MB의 blobspace를 제공하며, 최대 블록 크기는 약 0.75 MB로 제한됩니다.
이를 기준으로 하면, 다음과 같은 처리량 한계가 발생합니다.

- 기본 압축(약 154 바이트/트랜잭션)을 적용한 ERC-20 전송의 경우, 모든 롤업을 합쳐도 약 213 TPS 수준입니다.
- 압축(약 400 바이트/트랜잭션)을 적용한 스왑의 경우, 모든 롤업을 합쳐도 약 82 TPS 수준입니다.

이에 비해, Celestia는 올해 말 2 MB 블록 크기로 출시될 예정이며, 충분한 Data Availability Sampling (DAS) 라이트 노드가 온라인 상태가 되고 네트워크 안정성이 입증되면, blobspace는 8 MB까지 증가할 것으로 예상됩니다.
DAS 라이트 노드는 두 가지 핵심 기능을 수행합니다.

1. 사용자가 Eclipse 블록 데이터가 실제로 가용한지 직접 검증할 수 있도록 합니다.
2. DA 계층 전체의 안전한 확장에 기여합니다. DAS 라이트 노드가 많아질수록 데이터 처리량을 더 안전하게 증가시킬 수 있습니다.

Celestia는 생산 환경에서 DAS를 최초로 도입하는 DA 레이어가 될 것으로 기대됩니다. 이는 사용자가 직접 검증할 수 없는 전통적인 Data Availability Committee (DAC) 기반 접근법과 대조되며, DAC는 위원회의 정직성에 의존하는 기존 단일형 블록체인 구조와 유사한 신뢰 가정을 다시 도입하게 됩니다.

사용자가 Ethereum 메인넷에서 자금을 브리지하여 오프체인 DA를 사용하는 체인으로 이동할 경우, 다음과 같은 보안 가정이 내포됩니다. 기술적으로는 Celestia의 검증인들이 트랜잭션 데이터를 은닉하면서도 Ethereum 브리지에 데이터가 가용하다고 허위 주장할 수 있는 가능성이 존재합니다. 하지만 Celestia는 지분증명(PoS) 합의 알고리즘을 사용하기 때문에, 데이터 은닉 행위는 슬래시(징벌) 대상이 되며, 실질적으로 이러한 위험이 발생할 가능성은 매우 낮다고 판단하고 있습니다.
전반적으로, Celestia의 DAS(Datap Availability Sampling) 라이트 노드 지원이 출시 첫날부터 가능하다는 점, 암호경제적 보안 특성, 그리고 높은 확장성을 지닌 데이터 가용성(DA) 처리량 덕분에, 현재로서는 Eclipse 메인넷에 가장 적합한 선택지는 Celestia입니다.

일부에서는 앞서 설명한 이유들로 인해, 온체인 이더리움 DA를 사용하는 것이 진정한 'L2'의 요건이라고 보기도 합니다. 그러나 우리는 앞서 인용한 보다 일반적인 L2 용어 정의에 따르고 있으며, 보안에 관한 고려사항들을 명확히 하고자 합니다.

또한 우리는 EIP-4844 이후 이더리움의 DA 확장 진행 상황을 계속 주시할 계획입니다. 고급 분산 해시 테이블(DHT)을 사용하는 기존 접근 방식보다 더 빠르게 고처리량 DA를 제공할 수 있는 새로운 흥미로운 연구 결과들이 계속해서 발표되고 있습니다.
만약 이더리움이 Eclipse에 더 높은 확장성을 제공하고, 그것이 사용자에게 이득이 된다면, 이더리움 DA로의 마이그레이션 가능성을 검토할 것입니다.

증명: RISC Zero

SVM 사기 증명(Fraud Proofs)과 상태 직렬화(State Serialization)에 대한 개요

Eclipse의 증명 전략은 Anatoly가 제안한 SVM fraud proofs SIMD에서 영감을 받았으며, John Adler가 지적한 상태 직렬화 비용의 문제점에 대한 통찰을 기반으로 발전시킨 것입니다. Adler는 상태 직렬화에 드는 비용을 회피하는 것이 가능하다는 점을 밝혔으며, 이는 Eclipse의 접근 방식에서 핵심적인 개념입니다.

SVM에서 Merkle Tree를 회피하는 이유

초기 SVM 실험에서는 Sparse Merkle Tree를 통합하였습니다. 그러나 각 트랜잭션

이후 Merkle Tree를 갱신하는 작업은 성능에 심각한 저하를 초래하였습니다.
이에 따라 Eclipse는 SVM 내에서 Merkle Tree 재도입을 배제하기로 결정하였으며, 이는 전통적인 범용 롤업 프레임워크(예: OP Stack)에서 벗어나 보다 혁신적인 fault proof 아키텍처를 구축하는 계기가 되었습니다.

Fault Proof의 요건

Eclipse의 아키텍처에서 fault proof는 다음 세 가지 요건을 충족해야 합니다.

1. 트랜잭션 입력에 대한 커밋(Commitment to Transaction Inputs):
   트랜잭션 실행 이전에 모든 입력 값이 확정되고 커밋되어야 합니다.
2. 트랜잭션 실행(Transaction Execution):
   실제로 실행된 트랜잭션 자체를 포함합니다.
3. 출력 검증(Output Verification):
   트랜잭션을 다시 실행했을 때, 블록체인에 기록된 결과와 다른 결과가 나왔음을 증명하는 과정입니다.

Merkle 루트를 사용한 입력 커밋 방식 대신, Eclipse는 각 트랜잭션의 입력 및 출력 상세 목록을 게시합니다. 여기에는 계정 해시(account hashes), 관련 글로벌 상태 정보, 각 입력의 출처를 추적할 수 있는 인덱스 정보가 포함됩니다. 트랜잭션은 Celestia에 기록되며, 어떤 풀노드(full node)든 자신의 상태 데이터를 기준으로 입력 및 출력을 검증할 수 있습니다. 이를 통해 Ethereum에 기록된 커밋이 정확함을 보장합니다.

주요 오류 유형과 대응 방식

Eclipse는 두 가지 주요 오류 유형을 예상하고 있으며, 이에 따른 대응 방식은 다음과 같습니다.

1. 잘못된 출력(Incorrect Outputs):
   출력이 잘못된 경우, 검증자(verifier)는 올바른 출력을 증명하는 Zero-Knowledge (ZK) proof를 블록체인 상에 직접 제출합니다. Eclipse는 SVM 실행을 기반으로 한 ZK proof 생성을 위해 RISC Zero를 사용하며, 이는 BPF 바이트코드 실행 증명 경험을 확장한 것입니다. 이를 통해 정산 컨트랙트(settlement contract)는 트랜잭션을 온체인에서 직접 실행하지 않고도 정확성을 검증할 수 있습니다.

2. 잘못된 입력(Incorrect Inputs):
   입력값이 조작되었거나 잘못된 경우, 검증자는 온체인에 존재하는 과거 데이터를 참조하여 입력값의 오류를 입증합니다. 이때 Celestia의 Quantum Gravity Bridge를 활용하여, 정산 컨트랙트가 해당 과거 데이터가 사기 주장을 뒷받침하는지를 검증할 수 있습니다.

방법론(Methodology)

Eclipse의 이 증명 방법론은 다음과 같은 이점을 제공합니다. 고비용의 상태 직렬화 과정을 제거함으로써 성능을 개선하고, 기존 롤업 프레임워크에 대한 의존도를 줄여 보다 독립적이고 최적화된 구조를 실현합니다.이러한 접근은 트랜잭션을 블록체인 상에서 직접 실행하지 않더라도, 정확성과 무결성을 충분히 검증할 수 있게 하며, 결과적으로 강력하고 확장성 높은 롤업 솔루션을 제공하게 됩니다.

왜 이클립스인가, 왜 이더리움인가, 왜 지금인가.

Rollup 기술과 Ethereum에 미치는 영향

Rollup은 가상자산 업계 내 연구 수준을 크게 향상시키며, Ethereum 사용자에게 비용 효율적인 Layer 1 대안을 제공해왔습니다. 그러나 이들 기술은 아직까지 대중적 채택을 위한 최신 기술 발전을 충분히 반영하지 못하고 있는 실정입니다.

초기 Rollup의 진화와 한계

초기의 Rollup은 주로 EVM 호환성 확보와 영지식 증명(ZK Proof) 최적화에 집중하였습니다. 당시에는 유효한 전략이었지만, 최근에는 다음과 같은 기술적 진보로 인해 점차 뒤처지고 있습니다.

- 고성능 병렬 처리 가상 머신: 예) SVM
- 데이터 가용성 확장 기술: Celestia와 같은 DAS 라이트 노드 기술
- 증명 인프라의 발전: RISC Zero를 통해 현실 적용 사례 확대
- 코드 및 사용자 이동성의 향상: Neon, Solang, MetaMask Snaps 등 툴 및 솔루션 등장

Eclipse의 전략적 통찰 (Strategic Insight from Eclipse)

Eclipse는 과거의 기술적 흐름에서 전략적 인사이트를 도출하고,
확장성과 지속 가능성을 모두 고려한 최적의 기술 스택을 선별하여 통합하였습니다. 많은 논의에서 "수백만 개의 앱 전용 Rollup" 가능성이 언급되고 있지만, 이러한 광범위한 맞춤화에 대한 실제 수요는 제한적입니다. 실제로 대부분의 신규 Rollup은 단순한 **EVM 포크(fork)**에 지나지 않으며, 이는 사용자 경험의 단편화를 해소하지 못합니다.

단일화된 체인의 비전 (Vision of Unified Chains)

경쟁력 있는 사용자 경험을 제공할 수 있는 다수의 애플리케이션 전용 체인을 뒷받침할 인프라는 아직 수년은 더 발전이 필요합니다. 현재 Optimism의 Superchain, zkSync의 Hyperchains, Arbitrum의 Orbit 체인 등은 자신들의 생태계 내에서 사용자 경험을 향상시키기 위한 노력을 하고 있으나, 통합된 공유 상태(unified shared state)를 제공하거나 생태계 간 상호운용성(cross-ecosystem interoperability) 문제를 해결하기에는 부족한 상황입니다.

Solana의 영향력과 호환성에 대한 오해

Eclipse는 Solana의 단순한 접근 방식을 높이 평가합니다. Solana는 대다수의 주요 사용 사례를 처리할 수 있도록 최적화된 단일 공유 상태 머신이라는 모델을 채택하고 있습니다. 이러한 구조는 Rollup 중심 로드맵과는 양립 불가능한 것으로 오해되기도 하지만, Eclipse는 양 전략의 장점을 종합하고자 하며, 이는 Solana의 철학과도 잘 부합합니다.

현재 Rollup의 한계와 혁신의 필요성

현재 대부분의 Rollup은 단일 스레드 기반의 기본 EVM 구조에 의존하며, 초기 네트워크 효과를 활용하는 데 집중하고 있습니다. 하지만 이러한 구조는 다음과 같은 특정 고부하 애플리케이션에는 부적합합니다.

- 예: 고수요 NFT 민팅으로 인해 전체 체인의 수수료가 급등

각 애플리케이션마다 별도의 체인을 생성하는 대신, Eclipse는 병렬화된 VM과 로컬 수수료 시장(local fee markets)을 활용하는 접근을 지지합니다. 이러한 구조는 SVM이 대표적 사례입니다.

Eclipse 메인넷의 미래

Eclipse 메인넷은 Solana의 성능과 Rollup 기반의 보안성, 검증성, 네트워크 효과를 결합한 구조입니다. 이는 Ethereum의 혁신 중심 철학과도 부합하며, Layer 2 솔루션이 Ethereum의 네트워크 혜택을 그대로 누리면서 동시에 혁신적인 실행 환경을 탐색할 수 있는 기반을 제공합니다. Eclipse 메인넷은 현재에도 경쟁력 있는 Ethereum L2 플랫폼으로서 향후 기술 발전을 수용할 준비가 되어 있는 종합적 솔루션입니다.

2. 토큰 이코노미

가상자산 소개

ES는 Eclipse 메인넷의 네이티브 유틸리티 토큰으로서, 트랜잭션 수수료, 거버넌스, 스테이킹 용도로 사용됩니다.

Eclipse는 자체 수익(Native Yield) 기능을 활성화할 계획이며, 그 정확한 파라미터는 거버넌스 메커니즘을 통해 조정될 수 있습니다. 또한, 토큰 보유자를 위한 다양한 본딩(Bonding) 기회를 지원할 계획입니다.
요약하자면, Eclipse는 토큰 발행 후 다음과 같은 스테이킹 시나리오를 예상하고 있습니다.

- MEV 번들 경매(MEV bundle auctions) 참여
- 신뢰 최소화된 방식의 MEV 분배
- 사기 증명(Fraud Proof) 챌린지 보증금(bond) 제출

또한, PoS 기반의 퍼미션리스 시퀀싱(Permissionless Sequencing) 에서의 스테이킹 옵션도 고려 중입니다. 장기적으로는, ES 토큰이 지분 증명(Proof of Stake, PoS) 을 통해 시퀀서(sequencer) 참여를 가능케 하여 프로토콜의 탈중앙화를 더욱 촉진할 수 있을 것으로 보고 있습니다. 기타 고려 중인 사용 사례로는, ES를 사기 증명 챌린지에서 보증금(ES)을 게시하는 데 사용하는 것이 있습니다. 네트워크 참가자들은 이 과정에서 리소스를 제공하여 분쟁을 효율적으로 해결하고 ES 보상을 획득할 수 있습니다.

이러한 기능들 외에도, ES는 Eclipse 고유의 MEV 요구사항을 해결하는 데 사용될 수 있습니다. 예를 들어 MEV 번들 경매에 참여하거나 ES 스테이킹을 통한 무신뢰 기반의 MEV 재분배와 같은 메커니즘은 사용자로부터 악의적인 MEV를 추출하려는 시퀀서를 슬래시(slash)하는 등의 방식으로 사용자를 보호할 수 있게 합니다.

발행량 및 유통량계획

ES의 총 발행량은 1,000,000,000개입니다.
ES의 분배율 및 유통량계획은 다음과 같습니다.

- Ecosystem + Development: 35% (재단 준비금 포함, 거버넌스 관리 하 단계적 집행)
- Early Supporters / Investors: 31% (3년 락업)
- Contributors: 19% (4년 베스팅 + 3년 락업)
- Airdrop + Liquidity : 15% (TGE)

상기 유통량계획에 따라, TGE 시점 기준 유통 가능한 초기 물량은 150,000,000 개입니다.

3. 참고자료

https://docs.eclipse.xyz/

위험고지 안내
Disclaimer

본 문서에 기재된 정보는 당사(코인원)가 본 가상자산 심사 시점에 접근 가능한 정보 채널을 통하여 확인한 것으로, 정확하지 않거나 투자시점에는 변경 또는 유효하지 않을 수 있습니다.

가상자산 발행자가 공시한 내용 및 백서를 통해 정확한 정보를 확인하신 후 투자하시기 바랍니다.

가상자산은 법정화폐가 아니므로 특정 주체가 가치를 보장하지 않습니다.

coinone