

소닉SVM(Sonic SVM)

해당 번역본은 이용자가 디지털자산 관련 정보를 편리하게 확인할 수 있도록
단순 참고용으로 번역된 것이며, 투자 권유를 목적으로 하지 않습니다.
그 내용이 정확하지 않을 수 있으므로 정확한 정보 습득을 위해서는
원문을 참고하시거나 원문 작성 주체 측에 문의하시기 바랍니다.
두나무는 해당 번역본 내용의 진실성 및 정확성을 보장하지 않고,
제공된 정보에 의한 투자 결과에 대하여 어떠한 책임도 지지 않으며,
투자 손실은 투자자에게 귀속됩니다.

소닉에 오신 것을 환영합니다

소닉(Sonic)은 솔라나(Solana) 상에서 자주적인 게임 경제를 실현하기 위해 설계된 최초의 아토믹 SVM(Solana Virtual Machine) 체인이다.

이 프로젝트는 블록체인 게임이 직면한 특정 과제, 즉 **고성능을 유지하면서도 진정한 탈중앙화를 보장하는 복잡한 게임을 어떻게 운영할 것인가**라는 문제를 해결한다. 아토믹 SVM 체인으로서 소닉은 게임이 자체 경제를 유지하면서도 솔라나의 보안성과 연결된 상태로 운영될 수 있도록 한다.

소닉의 기술 스택

하이퍼그리드 프레임워크 (HyperGrid Framework)

소닉의 하이퍼그리드 프레임워크는 각 게임이 자체 전용 그리드에서 작동할 수 있도록 함으로써 이 생태계를 혁신적으로 변화시킨다. 이는 대규모 멀티플레이어 게임이 수천 명의 동시 플레이어 행동을 처리할 수 있고, 트레이딩 카드 게임이 복잡한 카드 상호작용과 마켓플레이스를 관리할 수 있음을 의미한다. 이 모든 것이 동일한 블록체인 자원을 두고 경쟁하지 않고 동시에 실행된다.

소라다 (Sorada)

소라다는 SVM 데이터 아키텍처 접근 방식에 근본적인 변화를 나타낸다. 읽기 연산(read operation)을 핵심 트랜잭션 처리 파이프라인에서 분리함으로써, 데이터 조회 성능에서 전례 없는 향상을 제공하는 시스템을 구축했다.

테스트 결과, 표준 RPC 노드와 비교했을 때 읽기 성능이 **30~40배 향상**되었다. 모든 애플리케이션에 대해 이는 읽기 연산 응답 시간이 **최소 5ms 수준**까지 줄어듦을 의미하며, 이는 진정으로 반응성 높은 블록체인 게임 경험을 가능하게 하는 혁신적 개선이다.

러시 (Rush)

러시는 블록체인 게임 개발을 보다 자연스럽게 접근하기 쉽게 만들기 위해 설계된 소닉의 ECS(Entity-Component-System) 프레임워크이다. 러시는 선언적 접근 방식을 취하여, 게임 개발자가 복잡한 블록체인 코드를 작성하지 않고도 단순한 설정 파일로 게임의 세계, 엔티티, 컴포넌트를 정의할 수 있게 한다. 복잡한 블록체인 개념을 학습할 필요 없이, 러시는 이러한 설정으로부터 모든 필수 스마트 컨트랙트와 SDK를 자동으로 생성한다.

이제 함께 소닉 생태계를 탐험해 봅시다. 준비하세요, 멋진 여정이 될 것이다!!

시작하기

소닉은 솔라나 상에서 자주적인 경제를 실현하기 위해 구축된 최초의 아토믹 SVM 체인이다.

솔라나의 최초 동시 확장 프레임워크인 하이퍼그리드(HyperGrid) 위에 구축된 소닉의 기술 스택은 애플리케이션이 솔라나와 아토믹 결제를 유지하면서도 맞춤형 블록 공간을 가질 수 있도록 한다.

지갑 설정하기

지갑은 소닉에서 사용자의 디지털 신원을 나타내며, 블록체인과 상호작용하고 자산을 관리하며 트랜잭션을 실행할 수 있도록 한다. 이 가이드는 소닉에서 지갑을 설정하고 구성하는 과정을 단계별로 안내한다.

지갑 (Wallets)

다음 중 하나의 지갑을 사용할 수 있다!

- 사용 중인 지갑의 공식 웹사이트를 방문하세요.
- 브라우저 확장 프로그램을 다운로드 및 설치하세요.
- 지갑의 설정 과정을 따라 새 지갑을 생성하거나 기존 지갑을 가져오세요.

지갑	이미지	링크
Backpack	backpack.png	https://backpack.app/
OKX Web3 Wallet	okx.png	https://www.okx.com/zh-hans/web3/wallet/sonic-svm
Nightly Wallet	nightly.png	https://nightly.app/
Bybit	bybit.png	https://www.bybit.com/en/web3/home
NuFi	Screenshot 2025-04-02 at 2.23.28 PM.png	https://nu.fi/

소닉으로 자금 브릿징하기

솔라나에서 소닉으로 자금을 브릿징하려면 다음의 브릿지를 사용할 수 있다.

브릿지	링크	이미지
Sonic bridge	https://bridge.sonic.game/	Screenshot 2025-02-15 at 7.15.34 PM.png
Hyperlane bridge	https://www.usenexus.org/	Primary Logo.svg
Superbridge	https://sonic-svm.superbridge.app/	Screenshot 2025-03-25 at 9.28.06 AM.png

소닉 커뮤니티에 참여하세요!

최신 뉴스와 공지를 확인하려면 소닉 커뮤니티에 참여하세요!

X(트위터) 팔로우: [@SonicSVM](#)

[디스코드 서버](#)에 참여하세요! 🗨️

아키텍처 (ARCHITECTURE)

개요 (Overview)

소닉(Sonic)은 솔라나(Solana) 상에서 자주적인 경제를 실현하기 위해 구축된 최초의 아토믹 SVM(Solana Virtual Machine) 체인이다.

솔라나의 최초 동시 확장 프레임워크인 하이퍼그리드(HyperGrid) 위에 구축된 소닉의 기술 스택은 애플리케이션이 솔라나와 아토믹 결제를 유지하면서도 맞춤형 블록 공간을 가질 수 있도록 한다.

소닉 SVM 백서 (Sonic SVM Whitepaper)

백서 링크: [Github](#)

\
기술 세부 사항 및 거버넌스

1. [Sonic Hypergrid 기술 로드맵](#)
2. [Sonic 거버넌스 프레임워크](#)

비전 및 아키텍처 (Vision & Architecture)

소닉은 다음과 같은 방향성을 가진 블록체인 게임의 미래를 상상한다.

- 게임 경제에 대한 진정한 자주권
- 전례 없는 성능과 확장성
- 기존 게임 개발 워크플로우와의 완전한 통합
- 솔라나 생태계와의 완전한 구성 가능성

아키텍처 철학 (Architectural Philosophy)

소닉의 아키텍처는 세 가지 근본적인 축 위에 구축되었다.

1. 상태 주권 (State Sovereignty)

각 게임 생태계는 자체 특화된 그리드(Grid) 내에서 작동하며, 이를 통해 다음이 가능하다.

- 독립적인 상태 관리
- 맞춤형 실행 환경
- 최적화된 성능 특성
- 유연한 자원 할당

2. 수평적 확장성 (Horizontal Scalability)

진정한 확장은 다음을 통해 달성된다.

- 다중 그리드 동시 실행
- 동적 자원 할당
- 병렬 트랜잭션 처리
- 그리드 특화

클러스터 (Clusters)

소닉은 여러 개의 검증자 그룹(Validator Groups)을 가지고 있으며, 이를 **클러스터(Clusters)** 라고 한다. 각 클러스터는 전체 생태계 내에서 서로 다른 목적을 수행하며, 해당 클러스터의 **JSON-RPC** 요청을 처리하기 위한 전용 API 노드를 포함하고 있다.

클러스터 내 개별 노드는 제3자에 의해 소유 및 운영되며, 각 노드에는 공개 엔드포인트가 제공된다.

링크 (Links)

네트워크 이름 (Network Name)	Sonic Testnet
Mainnet Alpha	https://api.mainnet-alpha.sonic.game
Helius RPC URL	https://sonic.helius-rpc.com/
Testnet RPC URL	https://api.testnet.sonic.game
Faucet URL	https://faucet.sonic.game
Block Explorer URL	https://explorer.sonic.game

하이퍼그리드 프레임워크 (HyperGrid Framework)

하이퍼그리드(HyperGrid) 프로토콜은 전용 솔라나 가상 머신(Solana Virtual Machine, SVM) 생태계 롤업 운영자를 위한 롤업 확장 및 오케스트레이션 프레임워크이다. 하이퍼그리드는 상태 압축(state compression)과 비잔틴 장애 허용(Byzantine Fault Tolerance, BFT)을 활용하여 다중 그리드 간의 수평 확장을 가능하게 함으로써 잠재적으로 무한한 트랜잭션 처리량을 달성한다. 그 대표적인 예가 솔라나에 정산(settlement)되는 게임 특화 그리드인 소닉(Sonic)이다.

그 외에도 하이퍼그리드의 핵심 기능 중 하나는 아토믹 상호운용성(Atomic Interoperability)이다. 하이퍼그리드는 솔라나와 상호운용되도록 설계되었다. 이 시스템은 솔라나에서 완전히 캡슐화된 프로그램이 하이퍼그리드가 오케스트레이션하는 그리드에 실행을 위임할 수 있는 상호운용성 인터페이스를 구현한다. 이러한 접근 방식은 다음과 같은 이점을 제공한다.

- 솔라나에서 실행 중인 모든 프로그램에 대한 단일 진실 원장(single source of truth)을 생성한다.
- 프로그램 계산을 하이퍼그리드로 위임할 수 있다.
- 롤업이 베이스 레이어의 서비스와 유동성 혜택을 누릴 수 있다.

하이퍼그리드 인프라 (HyperGrid Infrastructure)

이 섹션은 하이퍼그리드 공유 상태 네트워크(HyperGrid Shared State Network, HSSN)를 중심으로 한 기술적 해결 방안을 심층 분석한다. 이 포괄적인 문서는 다음과 같은 주요 측면들을 다룬다.

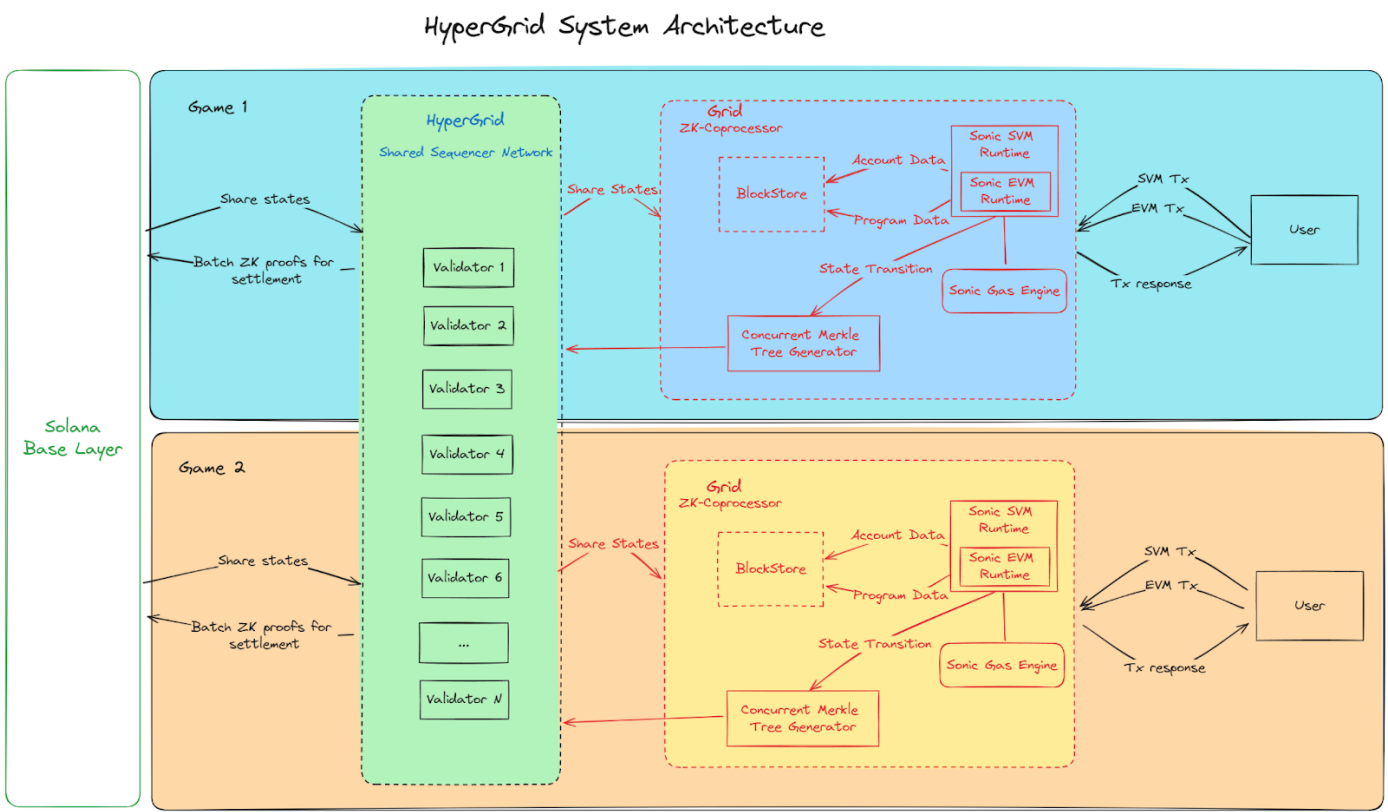
1. 하이퍼그리드의 데이터 상호작용 경로를 재구성하며, 시스템 내에서 정보가 어떻게 흐르는지를 상세히 설명한다.
2. 공유 상태 네트워크가 솔루션의 다양한 구성 요소를 오케스트레이션하고 관리하는 데 있어 수행하는 핵심 역할을 탐구한다.
3. 필요한 제품 및 개발 작업에 대한 설계 수준의 상세 분해를 제공한다. 이를 통해 구축되어야 할 구성 요소와 구현 방법을 명확히 정의하여 모든 필수 기능이 완비되도록 한다.

이 문서의 주요 목적은 하이퍼그리드 기술 솔루션의 성공적인 배포 및 통합을 위한 명확하고 실행 가능한 청사진을 제시하는 것이다. 이는 개발자, 아키텍트, 이해관계자들이 이 혁신적 시스템을 실현하는 데 참고할 수 있는 핵심 지침 자료로 사용될 것이다.

[architectural-overview](#)
[grids-and-network-relationships](#)

아키텍처 개요 (Architectural Overview)

하이퍼그리드의 아키텍처는 다중 그리드(multi-grid) 접근 방식을 기반으로 한다. 각 그리드는 반자율적으로 작동하면서도 합의(consensus) 및 최종성(finality)을 위해 솔라나 메인넷에 앵커(anchor)되어 있다. 이 섹션에서는 구조적 구성 요소와 그 상호 관계를 세부적으로 분석한다.



하이퍼그리드 시스템 아키텍처 (HyperGrid System Architecture)

주요 구성 요소 (Key Components)

- 솔라나 베이스 레이어(Solana Base Layer):** 하이퍼그리드 시스템이 구축된 기반으로, 최종 합의(consensus)와 최종성(finality)을 제공한다.
- 하이퍼그리드 공유 상태 네트워크(HyperGrid Shared State Network, HSSN):**
 - 아키텍처의 핵심으로, 모든 그리드를 아우른다.
 - 다수의 검증자(Validator 1 ~ Validator N)를 포함한다.
 - 그리드 간 및 솔라나 베이스 레이어 간의 상태(state) 공유를 지원한다.
 - 정산(settlement)을 위한 배치 ZK 증명(batch ZK proofs)을 관리한다.
- 그리드 구조(Grid Structure)** (예: Grid 1 및 Grid 2):

- 각 그리드는 반자율적 생태계를 나타내며, 특정 애플리케이션(예: 서로 다른 게임)에 전용될 수 있다.
- 각 그리드 내 구성 요소:
 - **ZK-Coprocessor**: 그리드별 연산 및 머클 증명(Merkle proof) 처리를 담당한다.
 - **SVM Runtime**: 솔라나 가상 머신(SVM) 상에서 그리드를 위한 실행 환경을 제공한다.
 - **Sonic Gas Engine**: 연산 자원을 관리한다.
 - **Concurrent Merkle Tree Generator**: 상태 전환(state transition)을 효율적으로 처리한다.

4. 사용자 상호작용(User Interaction):

- 사용자는 각 그리드와 독립적으로 상호작용할 수 있다.
- 트랜잭션(SVM Tx 및 EVM Tx)은 사용자와 해당 그리드의 런타임 간에 흐른다.
- 트랜잭션 응답은 다시 사용자에게 반환된다.

데이터 흐름 (Data Flow)

1. 상호운용성 – 상태 공유(Interoperability – State Sharing):

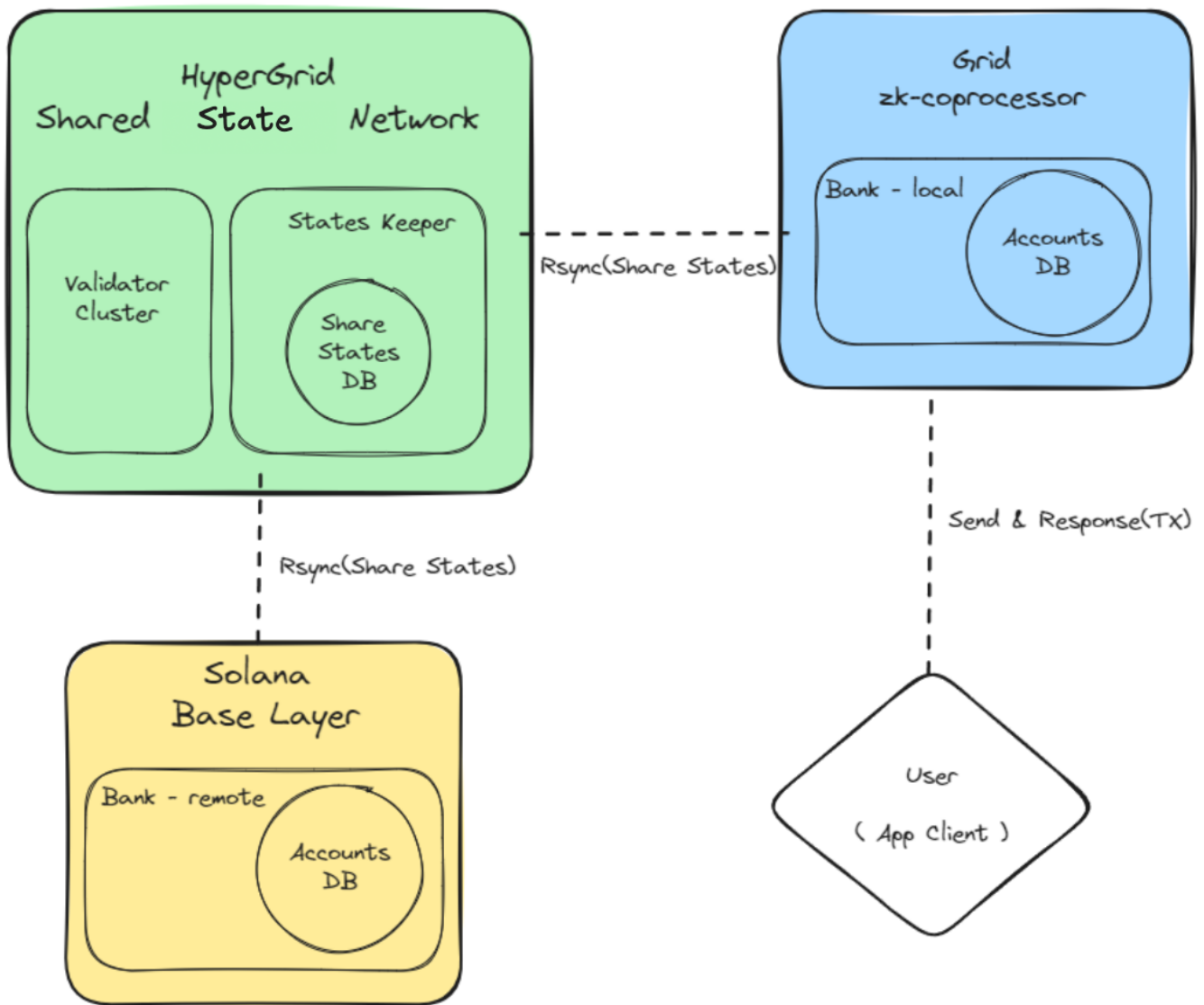
- 솔라나 베이스 레이어와 HSSN 간에 양방향 상태 공유가 발생한다.
- HSSN은 개별 그리드와 상태를 공유한다.
- 이후 설명하겠지만, 그리드 간에도 상태 공유가 가능할 수 있다.

2. ZK 증명(ZK Proofs):

- 트랜잭션은 압축 및 집계되어 머클 트리(Merkle tree)로 구성된다.
- 각 블록마다 해당 루트 상태 해시(root state hash)를 커밋한다.
- 이 블록의 유효성 증명(Validity Proof)은 그리드 상에서 계산된다.
- 정산을 위한 ZK 증명은 HSSN에서 솔라나 베이스 레이어로 게시된다.

이 아키텍처는 하이퍼그리드가 높은 확장성과 유연성을 달성하는 동시에 솔라나 블록체인이 제공하는 보안성과 최종성과의 강력한 연결을 유지하도록 한다.

그리드 및 네트워크 관계 (Grids and Network Relationships)



하이퍼그리드 공유 상태 네트워크 데이터 흐름 (HyperGrid Shared State Network Data Flow)

하이퍼그리드 프레임워크는 광범위한 애플리케이션 특화 네트워크 또는 탈중앙화 애플리케이션(dApps)을 지원하도록 설계되었다. 특히 게임, DeFi, AI 에이전트 등과 같은 고수요 애플리케이션에 중점을 둔 생태계를 제공한다. 이 아키텍처의 목표는 다음과 같다.

1. 솔라나 베이스 레이어의 성능 부하를 감소시킨다.
2. 베이스 레이어와 다양한 도메인 특화 dApp 간의 블록 공간 경쟁 및 성능 충돌을 최소화한다.

하이퍼그리드는 dApp 또는 운영자가 특정 애플리케이션을 위해 자신만의 그리드 인스턴스를 생성하거나, 자체 도메인 특화 SVM 네트워크를 구현할 수 있도록 지원한다.

[그리드에 대해 더 알아보기](#)

주요 기능 (Key Features)

- **그리드 네트워크 생성자를 위한 유연성(Flexibility for Grid Network Creators):** 개발자는 다음 중에서 선택할 수 있다.
 - 하이퍼그리드 공개 네트워크(HyperGrid public network)를 사용
 - 특정 요구사항에 맞춘 전용 네트워크를 수평적으로 확장하여 생성
- **성능 및 비용 최적화(Performance and Cost Optimization):** 공개 네트워크와 전용 네트워크 중 어떤 것을 사용할지는 개발자가 성능 요구사항과 관련 비용을 평가하여 결정한다.
- **네트워크 독립성(Network Independence):** 개발자는 자신이 관리하는 네트워크를 비활성화하더라도 생태계 내 다른 네트워크에는 영향을 미치지 않는다.

운영 프레임워크 (Operational Framework)

- **검증(Validation):** 각 네트워크는 트랜잭션 및 상태 변경의 검증을 자율적으로 처리한다.
- **로깅(Logging):** 트랜잭션 및 상태 변경 로그는 각 네트워크별로 독립적으로 유지된다.
- **조회(Retrieval):** 데이터 조회 프로세스는 각 네트워크에서 자율적으로 실행된다.

솔라나와의 통합 (Integration with Solana)

하이퍼그리드와 솔라나 베이스 레이어 간의 연결은 각 그리드 인스턴스 내에서 연산이 독립적으로 처리되더라도, 모든 연산이 궁극적으로 솔라나에 앵커되어 최종 합의(consensus)와 보안성을 확보하도록 보장한다.

이 아키텍처는 dApp 배포를 위한 확장 가능하고 유연한 환경을 제공하여, 개발자가 솔라나 블록체인의 보안성과 안정성을 활용하면서도 애플리케이션의 성능을 최적화할 수 있도록 한다.

그리드 (Grids)

정의 (Definition)

하이퍼그리드 아키텍처에서 그리드(Grid)는 대규모 하이퍼그리드 프레임워크 내에 존재하는 반자율적 네트워크로, 고수요 애플리케이션 특화 사용 사례를 호스팅하고 운영하도록 설계되었다.

예시로는 게임(Gaming), 디파이(DeFi), 온체인 AI 에이전트(On-chain AI Agents), 또는 특정 애플리케이션 전용 네트워크가 포함된다.

소닉(Sonic)은 고성능 게임을 위해 최적화된 그리드의 대표적인 예이다.

그리드는 단일 노드 형태로 존재할 수도 있고, 전체 클러스터로 구성될 수도 있다.

그리드의 특징 (Characteristics of a Grid)

다음은 그리드의 주요 특징이다.

1. **반자율적 운영(Semi-autonomous operation)**: 각 그리드는 상당 부분 독립적으로 작동하지만, 전체 하이퍼그리드 시스템과 연결된 상태를 유지한다.
2. **전용 자원(Dedicated resources)**: 각 그리드는 다음과 같은 자체 구성 요소를 가진다.
 - 그리드별 연산을 관리하는 ZK-coprocessor
 - 계정 및 프로그램 데이터를 처리하는 로컬 블록스토어(Local BlockStore, "Bank - local"이라고도 함)
 - Sonic SVM 런타임 및 Sonic EVM 런타임과 같은 실행 환경(Runtime environments)
 - 상태 전환을 효율적으로 처리하는 Concurrent Merkle Tree Generator
3. **상태 공유(State sharing)**: 그리드는 하이퍼그리드 공유 상태 네트워크(HSSN)와 상호작용하여 시스템 전체의 일관성을 유지하기 위해 상태를 동기화한다.
4. **사용자 상호작용(User interaction)**: 사용자는 그리드와 직접 상호작용하여 트랜잭션을 전송하고 응답을 받을 수 있다.
5. **확장성(Scalability)**: 그리드는 수평적 확장을 허용하여 개발자가 특정 요구사항에 맞춘 전용 네트워크를 생성할 수 있도록 한다.
6. **성능 격리(Performance isolation)**: 애플리케이션을 서로 다른 그리드로 분리함으로써, 시스템은 애플리케이션 간의 성능 충돌을 완화하고 솔라나 베이스 레이어의 부하를 줄일 수 있다.
7. **유연성(Flexibility)**: 개발자는 성능 요구사항과 비용 고려에 따라 공개 하이퍼그리드 네트워크를 사용하거나 자체 전용 그리드를 생성할 수 있다.
8. **독립적 생명주기(Independent lifecycle)**: 각 그리드는 시스템 내 다른 그리드에 영향을 주지 않고 개별적으로 활성화 또는 비활성화될 수 있다.
9. **SVM 클라이언트 다양성(SVM Client Diversity)**: 하이퍼그리드는 클라이언트 비종속적(client-agnostic)이므로, 어떤 SVM 클라이언트 구현체와도 함께 사용할 수 있다. 이를 통해 솔라나 네트워크 운영자는 하이퍼그리드의 공유 상태 네트워크(HSSN)를 합의(consensus) 및 상호운용성(interoperability)을 위한 기반으로 삼으면서 고성능 SVM 클라이언트를 구현할 수 있다.

본질적으로, 그리드는 하이퍼그리드 시스템 내에서 확장 가능하고 유연하며 반자율적인 노드 클러스터 또는 환경으로 기능한다. 이는 특정 dApp을 호스팅하고 관리하도록 설계되었으며, 동시에 전체 하이퍼그리드 아키텍처와 솔라나 베이스 레이어에 연결된 상태를 유지한다.

솔라나와의 상호운용성 (Interoperability with Solana)

베이스 레이어와의 상호운용성 (Interoperability with Base Layer)

하이퍼그리드와 솔라나 간의 상호운용성은 두 시스템이 정보를 원활하게 교환하고 자산을 이동할 수 있는 능력을 의미한다. 이 통합을 통해 사용자와 개발자는 마찰 없이 서로 다른 레이어 간에서 작업할 수 있으며, 이는 솔라나 블록체인 생태계의 전반적인 기능을 크게 향상시킨다.

상호운용성의 핵심 요소 (Key Aspects of Interoperability)

1. **크로스 레이어 연산(Cross-Layer Operations):**
 - 한 번 배포하면 영원히 실행 — 솔라나에 프로그램을 배포하면서, 트랜잭션은 그리드에서 실행할 수 있다.
 - 하이퍼그리드 네트워크에서 고성능 트랜잭션(예: 트레이딩)을 수행하면서, 솔라나 베이스 레이어에서 안전하게 정산할 수 있다.
2. **그리드 간 자산 전송(Inter-Grid Asset Transfer):**
 - 하이퍼그리드 내 서로 다른 그리드 간에 자산 및 데이터를 전송할 수 있다.
3. **원활한 사용자 경험(Seamless User Experience):**
 - 사용자는 두 시스템을 단일 통합 플랫폼처럼 매끄럽게 이용할 수 있다.

상호운용성의 이점 (Benefits of Interoperability)

1. **확장성 강화(Enhanced Scalability):**
 - 하이퍼그리드는 솔라나의 트랜잭션 처리 능력을 확장한다.
2. **보안성 향상(Improved Security):**
 - 솔라나의 강력한 보안성을 최종 정산에 활용한다.
3. **유동성 증가(Increased Liquidity):**
 - 자산이 레이어 간에 유연하게 이동할 수 있도록 한다.
4. **가용성 향상(Higher Availability):**
 - 여러 시스템 간에 부하를 분산하여 다운타임 위험을 줄인다.
5. **생태계 성장(Ecosystem Growth):**
 - 하이퍼그리드에서 실행되는 dApp 간의 협력과 개발을 촉진한다.

솔라나 생태계에 미치는 영향 (Impact on Solana Ecosystem)

상호운용성은 솔라나 블록체인의 전반적인 역량을 확장하는 데 있어 핵심적인 요소이다. 이를 통해 생태계는 다음을 실현할 수 있다.

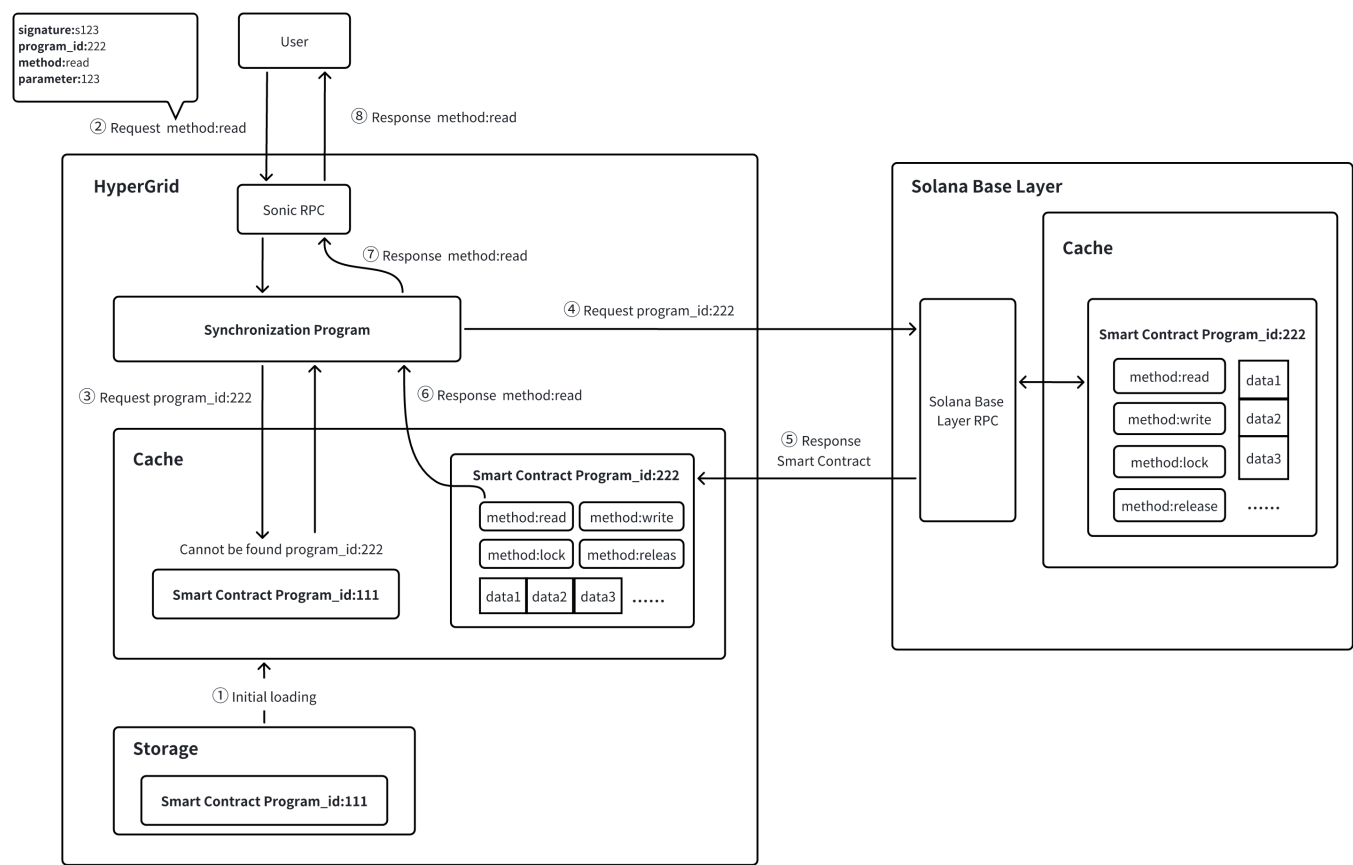
- 더 많은 트랜잭션 처리
- 복잡한 애플리케이션 지원
- 다양한 서비스 제공
- 더 넓은 사용자 및 개발자 기반 확보

하이퍼그리드와 솔라나 간의 원활한 상호작용을 가능하게 함으로써, 이 상호운용성 기능은 현대의 다양한 탈중앙화 애플리케이션 요구를 충족할 수 있는 보다 강력하고 유연하며 확장된 블록체인 환경을 구축한다.

하이퍼그리드와 솔라나 베이스 레이어 간의 데이터 동기화 (Data Synchronization Between HyperGrid and Solana Base Layer)

솔라나에서 하이퍼그리드로 데이터 읽기 (Reading Data from Solana to HyperGrid)

Reading Data from the Solana Base Layer



솔라나에서 하이퍼그리드로 데이터 읽기 (Reading Data from Solana to HyperGrid)

위 다이어그램은 솔라나에서 하이퍼그리드의 특정 그리드(예: 소닉)로 상태(state)를 동기화할 때의 흐름을 보여준다.

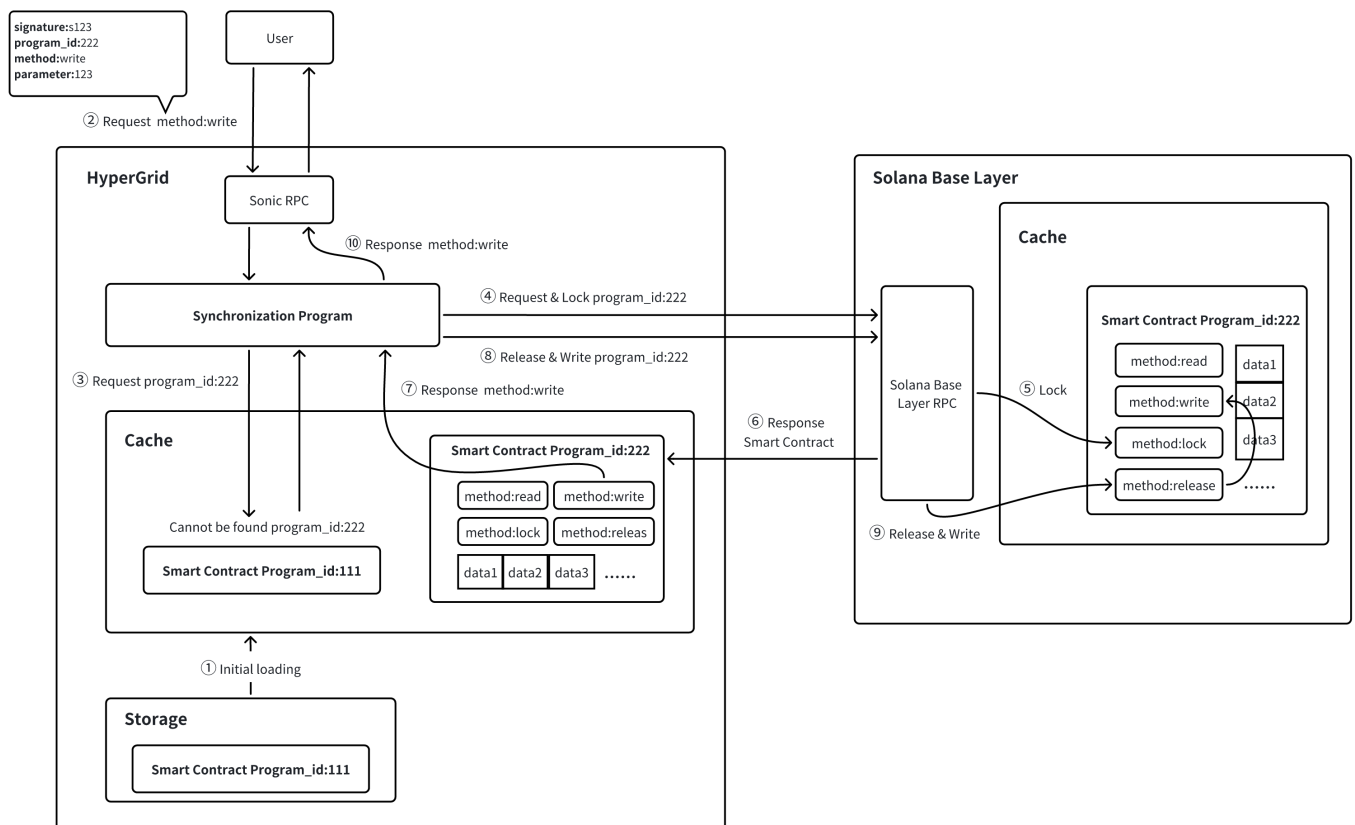
1. 초기 로딩(Initial loading): 기존의 솔라나 프로그램이 스토리지에서 하이퍼그리드의 캐시

(Cache)로 로드된다.

2. 사용자가 특정 프로그램에 대한 읽기 요청(read request)을 하이퍼그리드의 소닉 RPC로 전송한다.
3. 동기화 프로그램(Synchronization Program)은 요청된 프로그램을 캐시에서 확인하지만, 존재하지 않는다.
4. 동기화 프로그램이 솔라나 베이스 레이어 RPC에 프로그램 데이터를 요청한다.
5. 솔라나 베이스 레이어가 프로그램 데이터를 응답으로 반환한다.
6. 동기화 프로그램이 응답을 수신하고, 새로운 프로그램 데이터를 하이퍼그리드의 캐시에 업데이트한다.
7. 동기화 프로그램이 읽기 응답(read response)을 소닉 RPC로 전송한다.
8. 소닉 RPC가 읽기 응답을 사용자에게 전달한다.

솔라나 베이스 레이어로 업데이트 동기화하기 (Synchronizing Updates Back to the Solana Base Layer)

Synchronizing Updates Back to the Solana Base Layer



그리드에서 베이스 레이어로 업데이트 동기화 (Synchronizing Updates from Grid to Base Layer)

위 다이어그램은 하이퍼그리드의 특정 그리드(예: 소닉)에서 솔라나로 상태(state)를 동기화할 때의 흐름을 보여준다.

1. 초기 로딩(Initial loading): 기존 프로그램이 스토리지에서 하이퍼그리드의 캐시로 로드된다.
2. 사용자가 특정 프로그램에 대한 쓰기 요청(write request)을 하이퍼그리드의 소닉 RPC로 전송한다.
3. 동기화 프로그램이 요청된 프로그램을 캐시에서 확인하지만, 존재하지 않는다.
4. 동기화 프로그램이 솔라나 베이스 레이어에 프로그램 잠금(lock) 요청을 전송한다.
5. 솔라나 베이스 레이어 RPC가 요청된 프로그램을 잠근다.
6. 솔라나 베이스 레이어가 프로그램 데이터를 응답으로 반환한다.
7. 동기화 프로그램이 응답을 수신하고, 새로운 프로그램 데이터를 하이퍼그리드의 캐시에 업데이트한다.
8. 동기화 프로그램이 잠금 해제(release lock) 요청과 함께 업데이트된 프로그램 데이터를 솔라나 베이스 레이어에 기록(write)하도록 요청한다.
9. 솔라나 베이스 레이어 RPC가 잠금을 해제하고 업데이트된 데이터를 기록한다.
10. 동기화 프로그램이 쓰기 응답(write response)을 소닉 RPC로 전송한다.
11. 소닉 RPC가 쓰기 응답을 사용자에게 전달한다.

하이퍼그리드 공유 상태 네트워크 (HyperGrid Shared State Network, HSSN)

소개 (Introduction)

하이퍼그리드 공유 상태 네트워크(HSSN)는 그리드 생태계에서 핵심적인 구성 요소로, 합의 레이어 (consensus layer), 통신 허브(communication hub), 상태 관리 클러스터(state management cluster) 역할을 수행한다.

이 네트워크는 그리드와 솔라나 베이스 레이어 간의 상호작용을 조정하며, 그리드 롤업에서 솔라나 베이스 레이어로의 블록 데이터 주기적 동기화를 포함한 모든 통신 상태를 관리한다.

주요 구성 요소 및 특징 (Key Components and Features)

1. **HSSN 아키텍처(HSSN Architecture):** 코스모스(Cosmos) 프레임워크를 기반으로 구축되어, 체인 간 통신(cross-chain communication)에서 안정성과 보안을 보장한다.
2. **데이터 구조(Data Structures):** 그리드와 솔라나 베이스 레이어 간의 상태를 관리하며, 다음을 포함한다.

- 그리드 등록(Grid registration)
 - 통신 데이터 소스(Communication data sources)
 - 버저닝(Versioning)
 - 읽기/쓰기 상태(Read/write states)
3. **확장된 계정 데이터 필드(Extended Account Data Fields):** HSSN이 관리하는 새로운 필드를 수용하기 위해 솔라나 베이스 레이어의 기본 계정 데이터 필드를 확장하여, 그리드 계정 상태와의 동기화를 보장한다.
 4. **리팩터링된 그리드 RPC(Refactored Grid RPC):** 그리드와 HSSN 간의 직접 통신을 가능하게 하여, 생태계 내 상호운용성을 촉진한다.
 5. **가스 청구 및 할당 메커니즘(Gas Charging and Allocation Mechanism):**
 - 사용자는 특정 그리드 요청에 대해 가스 요금을 지불한다.
 - 전용 그리드(Sonic Grid)가 가스 계산 프로그램을 실행한다.
 - 전체 그리드 생태계에서 가스를 중앙집중적으로 관리한다.

상호작용 및 데이터 흐름 (Interactions and Data Flow)

1. **그리드 → HSSN 통신(Grid-to-HSSN Communication):**
 - 그리드는 상태 업데이트(state updates)를 HSSN으로 전송한다.
 - HSSN은 여러 그리드 간 상태를 동기화한다.
2. **HSSN → 솔라나 베이스 레이어(HSSN-to-Solana Base Layer):**
 - 블록 데이터를 롤업(rollup) 형태로 주기적으로 동기화한다.
 - 확장된 계정 데이터 필드를 관리한다.
3. **그리드 간 통신(Inter-Grid Communication):**
 - HSSN은 서로 다른 그리드 간의 상태 공유를 지원한다.
4. **가스 관리(Gas Management):**
 - 소닉 그리드(Sonic Grid)가 가스 요금을 계산하고 할당한다.
 - HSSN과 정기적으로 가스 청구 기록을 동기화한다.
 - 청구 기록에 따라 그리드 계정, HSSN 계정, 공식 계정에 자금을 할당한다.

다른 구성 요소와의 관계 (Relationship to Other Components)

HSSN은 그리드 생태계의 중심 역할을 수행하며, 다음과 밀접하게 상호작용한다.

- 개별 그리드(Individual Grids)
- 솔라나 베이스 레이어(Solana Base Layer)
- 가스 계산 프로그램(Gas Calculation Program, Sonic Grid)

HSSN은 상태 관리 및 통신을 위한 합의 레이어로 기능함으로써, 전체 그리드 생태계의 확장성, 상호운용성, 효율성을 향상시킨다.

HSSN 탐색기 개요 (HSSN Explorer Overview)

소개 (Introduction)

하이퍼그리드 공유 상태 네트워크(HSSN)는 코스모스(Cosmos) 기반 네트워크로, 그리드와 솔라나 베이스 레이어 사이에서 작동한다. 이 네트워크는 모든 통신의 상태를 관리하며, 그리드 간 상호작용(inter-grid interactions)과 블록 데이터를 롤업 형태로 솔라나 베이스 레이어에 주기적으로 동기화하는 역할을 수행한다.

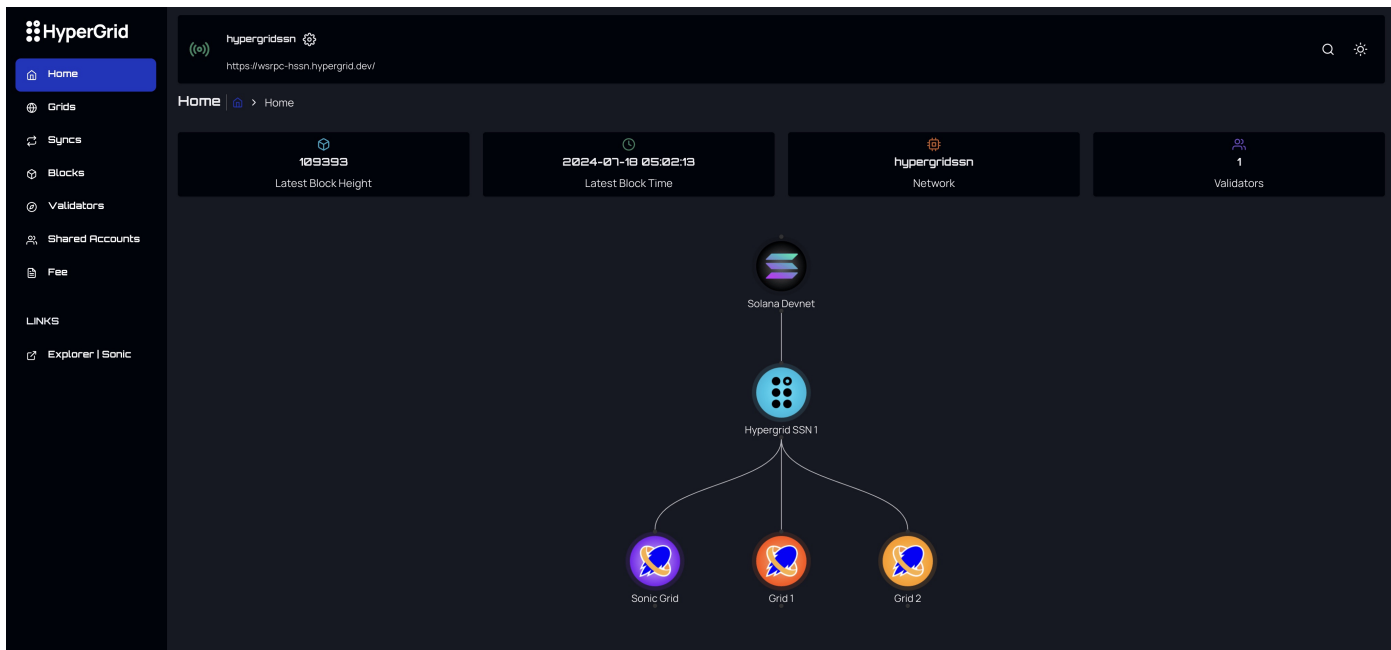
HSSN의 운영 상태를 모니터링하기 위해 전용 대시보드가 제공된다.

HSSN 탐색기 대시보드는 다음 링크에서 접근할 수 있다: <https://explorer-hssn.hypergrid.dev/>

https://drive.google.com/file/d/13y-B5pgrdlKZ5HvMqUvviO-uPJCg_ike/view?usp=sharing

대시보드 페이지 (Dashboard Pages)

1. 홈 페이지 (Home Page)



홈 페이지는 전체 HSSN의 개요를 제공한다.

- 최신 블록 높이(Latest Block Height)
- 최신 블록 시간(Latest Block Time)
- 네트워크 이름(Network Name) (현재 "hypergridssn")
- 검증자 수(Number of Validators)
- HSSN을 통해 그리드와 솔라나 간의 통신 흐름을 보여주는 연결 다이어그램(Connection Diagram)

2. 그리드 페이지 (Grids Page)

HSSN 내의 모든 노드 정보를 표시한다.

- 이름(Name)
- 역할(Role) — Solana, HSSN, Sonic Grid 또는 Grid
- RPC 인터페이스(RPC interface)
- 공개 키(Public key)
- 데이터 계정(Data account)
- 시작 시간(Start time)

3. 동기화 페이지 (Syncs Page)

NAME	ROLE	RPC	PUBKEY	DATA ACCOUNT	START TIME
Solana Devnet	Solana	https://api.devnet.solana.com/	39cwwUEpgka9bU7Sn4my82VViMDWaCxi4YoPevfZxLf3		-
Hypergrid SSN 1	HSSN	https://rpc-hssn.hypergrid.dev/	48aPaDzM4BdnmptSi4WWgdS8fKReqPUq8rJpQHC91MH	BYTVSj22e37H5jy4tGZpWQ16APr7rzayGnoQZxt3f65A	2024-07-11 18:19:56
Sonic Grid	Sonic Grid	https://grid-sonic.hypergrid.dev/	Erh16dEssgsk3pDTX2Dh1Uhj9LbJk4uUJfokLKH8B6	2BzQJEFNKzzicvfrRPz8Sek4AyUfMTEMqAby2jKk7yE	2024-07-11 18:19:56
Grid 2	Grid	https://grid-2.hypergrid.dev/	FRF59Lse9u8oHyTG7DmseTvyh3un9f09Lpyu5jQ5TrA	AFzhVsd78Yc7ireNLX4eKri8CHEWBenX7mDfNV41nPtf	2024-07-11 18:19:59
Grid 1	Grid	https://grid-1.hypergrid.dev/	HxOyiOxmVipgohFSdEx3TSy/fKfp6yttf1T3Rdp7fzfP	4cxFx4N2Q4cSOcADwgUpOkUW2rj7DMKLtMrCBOLZhH	2024-07-11 18:19:59

그리드 노드와 솔라나 간의 동기화 데이터를 표시한다.

- ID
- 그리드 이름(Grid name)
- 계정 정보(Account information)
- 슬롯 번호(Slot number)
- 트랜잭션 해시(Transaction hash)
- 생성자 계정(Creator account)

4. 블록 페이지 (Blocks Page)

HEIGHT	APP HASH	TXS	TIME
124072	794d1151b8819db09e59368926a5c5afd3866ac7d2679522e6ac9c4f3e9c74d5	0	a few seconds ago
124071	63e74d71a4443b909b230ba73431f8db385cf548549875f67330a4c74258161	1	a few seconds ago
124070	0c4777b011ad72ce3aaab8b933231167d913c094c3b83d9b70f31bd27854b4e1	0	a few seconds ago
124069	9a40fec0ab5128a499f20a35c17399d92ebcda7e063a1e13feeee38371354d20	0	a few seconds ago

HSSN 내 블록체인 정보를 제공한다.

- 블록 탭(Block tab): 블록 높이(Height), 앱 해시(App Hash), 트랜잭션 수(Number of transactions), 생성 시간(Creation time)
- 트랜잭션 탭(Transactions tab): 트랜잭션 해시(TX Hash), 결과(Result), 메시지(Messages), 높이(Height), 시간(Time)

TX HASH	RESULT	MESSAGES	HEIGHT	TIME
---------	--------	----------	--------	------

트랜잭션 탭 (Transactions Tab)

5. 검증자 페이지 (Validators Page)

VALIDATOR	STATUS	VOTING POWER
hssn1	Active	100

Page 1 of 1 Go to page: 1 Show 10

HSSN 검증자(Validators)에 대한 정보를 제공한다.

- 검증자 이름(Validator name)
- 상태(Status)

- 투표 권한(Voting power)

6. 공유 계정 페이지 (Shared Accounts Page)

HyperGrid

hypergridssn
https://wsrpc-hsnn.hypergrid.dev/

Shared Accounts

ADDRESS	SOURCE	SLOT	ACCOUNT INFO	CREATOR	VERSION
metaqbxxUerdq28cj1RbAWkYQm3ybzjb6a8bt518x1s	Solana Devnet	312494674	[{"lamports":1141440,"owner":"BPFLoaderUpgradeab1e1111111111..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
PwDfXfQsGra4sFTT8r1QWRMD4vfmWC1jFWNdrYT	Solana Devnet	312494848	[{"lamports":5526775920,"owner":"BPFLoaderUpgradeab1e1111111111..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
HB2GzqnKDUQs8u7dELjKWqV71RmhsahFOkVktTe4S5	Solana Devnet	312493910	[{"lamports":2422080,"owner":"5XU1wZkTwAw9mc5FbM3eBgAT83TKg..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
FaGW5o1NH5qHfLyFeU6kYACTwoHVDqLpe1JC5QJkn3hY	Solana Devnet	312275200	[{"lamports":1461600,"owner":"TokenkegQfeZyNwAJbNbGKPFXCWu..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
Csz6Y33L28jpD58NodtPrshUgXQmfx9HyLfaMc8B9	Solana Devnet	310918752	[{"lamports":3308373360,"owner":"BPFLoaderUpgradeab1e1111111111..."}]	cosmos1rhn3gp308i90qyhauq6wv8bracm0ahcu54cw	0
CmePeEQUFxaUDPbzD71Lwph6KU4hPMGvuCGkTX2BrC	Solana Devnet	312478412	[{"lamports":1461600,"owner":"TokenkegQfeZyNwAJbNbGKPFXCWu..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
BxWUTqEspj3bdtJNBPTVMkcnAPEWRuvRG1cGwsvGZP	Solana Devnet	312475965	[{"lamports":2422080,"owner":"5XU1wZkTwAw9mc5FbM3eBgAT83TKg..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
BQunTBT8gmvsQXtDKSsmJoUC38CT2NDMxMbM6Hxfkt	Solana Devnet	312494218	[{"lamports":1461600,"owner":"TokenkegQfeZyNwAJbNbGKPFXCWu..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
B8fedfnZ6gLnCBnn1G1YdkjAwSJuck4zQte7Wp259c	Solana Devnet	312479952	[{"lamports":1461600,"owner":"TokenkegQfeZyNwAJbNbGKPFXCWu..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
A3b25P8N8Qnx1PMQrkt5bsSyWtNBG9TWdYRwCj1BX	Solana Devnet	312493603	[{"lamports":2422080,"owner":"5XU1wZkTwAw9mc5FbM3eBgAT83TKg..."}]	cosmos1cscvs3ae5xe4t5v6yww7yh7rppj5ks9jnnv7jet	0
8RMSJryKHaQghTWSSJ3NsCd2yuuBPxCfn9MaYpARt1P	Grid 1	1402585	[{"lamports":4084640,"owner":"5pHITXvzhE6j9LzjXeAn3ooRNS2zd..."}]	cosmos1cscvs3ae5xe4t5v6yww7yh7rppj5ks9jnnv7jet	0
7MNa572qz6eS7hGtYW8usj6YMaR4jmfWqphEngkWSEN7	Solana Devnet	312489892	[{"lamports":1461600,"owner":"TokenkegQfeZyNwAJbNbGKPFXCWu..."}]	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sypwec	0
5pHITXvzhE6j9LzjXeAn3ooRNS2zdVGV6pPucBhGxH	Grid 1	1399001	[{"lamports":1141440,"owner":"BPFLoaderUpgradeab1e1111111111..."}]	cosmos1cscvs3ae5xe4t5v6yww7yh7rppj5ks9jnnv7jet	0
4WTUyxNcf6QCEj76b3aRDLPeWkPGkYFzkyf3A3vua1z	Solana Devnet	310918406	[{"lamports":1141440,"owner":"BPFLoaderUpgradeab1e1111111111..."}]	cosmos1rhn3gp308i90qyhauq6wv8bracm0ahcu54cw	0

HSSN 내 통신 상태 정보를 나열한다.

- 주소(Address)
- 소스 노드(Source node)
- 슬롯(Slot)
- 계정 정보(Account info)
- 생성자(Creator)
- 버전(Version)

7. 수수료 페이지 (Fee Page)

수수료 청구 및 정산 기록을 표시한다.

- **Grid Block Fee 탭:** 원본 트랜잭션 수수료 기록
- **Fee Settlement 탭:** 수수료 청구서 기반의 주기적 정산 기록

HyperGrid Home Grids Syncs Blocks Validators Shared Accounts Fee LINKS Explorer Sonic	hypergridssn https://wsrpc-hssn.hypergrid.dev/ Fee Grid Block Fee Fee Settlement						
	ID	GRID	BLOCKHASH	SLOT	BLOCKTIME	FEE	CREATOR
	653	Grid 1	9cA2v4W8hziVSq4NcYtrNsTq7jorgDUXpSAEQo826i8z	1486901	1721325281	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	652	Grid 1	Bg4vDsP8t5c4isCnQ52yeBx8w1oa1Hwf2M7CXISrXgU	1486749	1721325220	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	651	Grid 1	GLaH5EDf9rCUOnNxZT73vj2ZMqTKmnzajEbPytMdgAM	1486748	1721325219	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	650	Grid 1	8nXpvgYH6ApsrYfWbJZ8PTWgFug6jB05attNva5Bnp	1486724	1721325209	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	649	Grid 1	2ZY7mXqhoXoDhLP6dkQEycfBaW6yRXmZhwnc2Py34LmZ	1486664	1721325185	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	648	Grid 1	B693J8d7689xnedG2RE5thhknf3qarxVrFb2wPkpMTgm	1486549	1721325138	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	647	Grid 1	6hx18DroamBroCLzRYtiqXSPTW1hSifMWQxuhDEid	1486546	1721325137	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	646	Grid 1	Gcyrj463JtYajd95chHwCqcExtSY3cUjpe7h6zJ2uz6j	1486537	1721325133	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	645	Grid 1	6ubttEvA6CSEFkt53y8ga47K432RqkQYga647CXgBQFW	1486482	1721325111	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	644	Grid 1	vSwfkk63xaC1FBYoGTqBVLbUjsu18zs1SpDZdzJoNM	1486379	1721325069	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	643	Grid 1	3nbnGWUvUitDUaLV2NVjwYg7XVsRmtZnMSMVZYH9X6	1486356	1721325060	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	642	Grid 1	78d7QZev3ifk27iWOEgWENTBnWoGbK3TKQhTFB86fWQB	1486333	1721325051	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	641	Grid 1	HXZfrbXyN4XgF2kkZtENxpx9VU7jKdpKvLRydaVaY6mj	1486261	1721325021	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	640	Grid 1	Hcczx8cSVSojn7SqAVRALgh888sekYGB3ZGVk77vCuvi	1486172	1721324985	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec
	639	Grid 1	2ASBK4ennm5tEQ8197scqtV7rkWOhTdDxYWP1W5FzU8Ge	1486157	1721324979	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690syypwec

수수료 페이지 (Fee Page)

HyperGrid

Home

Grids

Syncs

Blocks

Validators

Shared Accounts

Fee

LINKS

Explorer | Sonic

hypergridssn

https://wsrpc-hssn.hypergrid.dev/

Fee

Grid Block Fee

Fee Settlement

ID	FROM ID	END ID	BILL	STATUS	CREATOR
9	29	41	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":15000,"FRF59..."}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
8	6	29	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":120000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
7	3	5	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":20000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
6	0	2	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":10000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
5	0	2	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":10000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
4	0	2	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":10000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
3	0	2	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":10000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
2	0	2	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":10000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
1	0	2	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":10000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw
0	0	2	[{"Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUtFokLKHB86":10000}]	0	cosmos1rin3gp308l90lqyhauq6www8pracm0ahcu54cw

수수료 정산 페이지 (Fee Settlement Page)

다른 구성 요소와의 관계 (Relationship to Other Components)

HSSN 대시보드는 그리드 생태계의 중심적 역할을 수행하는 HSSN의 운영에 대한 핵심 인사이트를 제공한다. 이를 통해 다음에 대한 가시성을 확보할 수 있다.

- 그리드 간 통신(Communication between Grids)
- 솔라나 베이스 레이어와의 동기화(Synchronization with the Solana Base Layer)
- 가스 요금 계산 및 정산(Gas fee calculations and settlements)

이 포괄적인 뷰를 제공함으로써 탐색기 대시보드는 투명성을 강화하고 전체 그리드 생태계의 효율적인 운영 관리를 지원한다.

HSSN 가스 요금 메커니즘 (HSSN Gas Fee Mechanism)

개요 (Overview)

HSSN 아키텍처에서는 사용자가 시작한 트랜잭션의 가스 요금이 네 부분으로 분배된다. 트랜잭션마다 가스 요금 분배 비율은 달라질 수 있다. 아래는 수수료 분배 방식을 설명하기 위한 샘플 트랜잭션이다.

샘플 트랜잭션 (Sample Transaction)

수신자(Recipient)	금액(Amount, SOL)	비율(Percentage)
소각(Burned)	0.00000125	50%
HSSN 수수료(HSSN Fees)	0.00000313	12.52%
SONIC 수수료(SONIC Fees)	0.00000313	12.52%
GRID 수수료(GRID Fees)	0.00000625	25%
총 수수료(Total Fee)	0.00000250	100%

이 트랜잭션은 [Sonic Explorer](#)에서 확인할 수 있다.

분배 설명 (Distribution Explanation)

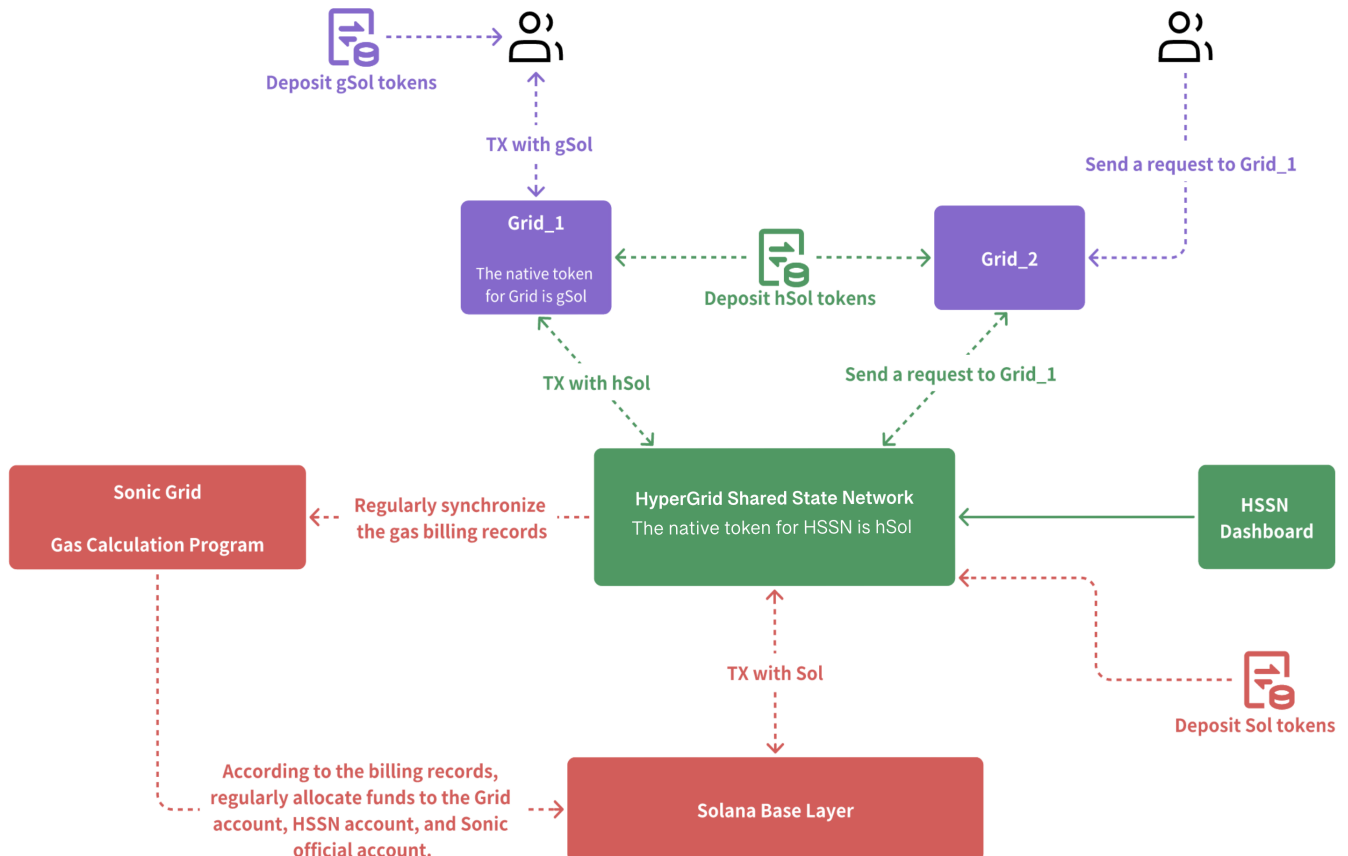
- 소각 수수료(Burned Fees):** 가스 요금의 일부가 소각되어 유통량에서 완전히 제거된다. 이는 토큰 공급을 조절하고 남은 토큰의 가치를 상승시킬 가능성을 높인다.
- HSSN 수수료(HSSN Fees):** 이 부분은 트랜잭션 정보를 관리하고 전송하는 역할을 하는 하이퍼그리드 공유 상태 네트워크(HSSN) 검증자에게 분배된다.
- SONIC 수수료(SONIC Fees):** 이 금액은 청구서 패키징과 실행을 담당하는 소닉 그리드(Sonic

Grid)에 할당된다.

4. GRID 수수료(GRID Fees): 사용자 트랜잭션 요청을 최초로 수신하고 처리하는 그리드 노드에 분배된다.

사용자는 위의 내용이 단순한 예시임을 이해해야 한다. 실제 가스 요금 및 분배 비율은 실시간 트랜잭션 환경에서 달라질 수 있다. 네트워크 혼잡도, 트랜잭션 복잡성, 현재 가스 가격 등의 요인이 총 수수료와 분배 비율 모두에 영향을 줄 수 있다.

정확한 가스 요금 추정치와 분배 비율은 반드시 네트워크에서 제공하는 최신 정보를 참조해야 한다.



가스 요금 분배 프로세스 (Gas Fee Distribution Process)

1. 트랜잭션 요청 및 청구서 생성 (Transaction Request and Bill Generation)

그리드 노드가 사용자의 트랜잭션 요청을 수신하면 다음을 수행한다.

- 필요한 가스 요금을 계산한다.
- 사전 정의된 규칙에 따라 참여 노드 간에 해당 요금을 분배한다.
- 세부적인 청구서(Bill) 기록을 생성한다.

2. 청구서 전달 및 조회 (Bill Forwarding and Viewing)

- 그리드 노드는 hypergrid-aide 프로그램을 실행한다.

- 이 프로그램은 정기적으로 가스 요금 청구서를 HSSN 검증자에게 전달한다.
- 사용자는 [HSSN 대시보드](#)의 **“Grid Block Fee”** 탭에서 트랜잭션 청구서 세부 정보를 확인할 수 있다.

HyperGrid Home Grids Syncs Blocks Validators Shared Accounts Fee LINKS Explorer Sonic						
hypergridssn https://wsrpc-hssn.hypergrid.dev/						
Fee Grid Block Fee Fee Settlement						
ID	GRID	BLOCKHASH	SLOT	BLOCKTIME	FEE	CREATOR
653	Grid 1	9cA2v4W8hzVSq4NcytrNsTq7jorgDJXpSAEOo826I8z	1486901	1721325281	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
652	Grid 1	Bg4vDsP8t5c4isCnQ52yeBx8w1oa1Hw2M7CXfSrxgU	1486749	1721325220	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
651	Grid 1	GLaH5EDf9FICUQnNkZT73vj2ZMqTkmnzajEtbPytMdgAM	1486748	1721325219	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
650	Grid 1	8nXpvgvH6ApsnYYWbJZ8PTWgFuq6jB05attNva5Bnp	1486724	1721325209	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
649	Grid 1	2ZY7mXqhoXoDhLP6dkQEycyBaW6yRkmZhrwec2Py34LmZ	1486664	1721325185	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
648	Grid 1	B693J8d7689xnedG2RE5ttHknf3qarXvFb2wPkpMTgm	1486549	1721325138	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
647	Grid 1	6hx18DroamBroCLzRYtiqXSPTnW1hSifmFWQxuDId	1486546	1721325137	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
646	Grid 1	Gcyrtj463JtYajd95chHwCqcExYSy3cUjpe7h6zJ2uz6j	1486537	1721325133	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
645	Grid 1	6ubtEvA5CSEFKt53y8ga47K432RqQlYg647CXgBQrW	1486482	1721325111	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
644	Grid 1	vSwfk63xaC1FBYoGTQbVLbUjsu18zs1SpDZZdzJoNM	1486379	1721325069	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
643	Grid 1	3nbnGWUvUitDuaLV2NVjvw7g7XVsRnvtZnM5MVZYH9X6	1486356	1721325060	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
642	Grid 1	78d7OZeV3ifk27WOEgWENTBnWoGbK3TKQhTFB86fWQB	1486333	1721325051	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
641	Grid 1	HXZfrbXN4XgFzkkZtENkpx9VU7jKdpKvLrydaVaY6mj	1486261	1721325021	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
640	Grid 1	Hcczx8cSVSojn7SqAVRALgh888sekYGB3ZGvk77vCuvi	1486172	1721324985	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec
639	Grid 1	2ASBk4enm5tEO8197scqtV7rkWDnTDdXyWP1W5FzU8Ge	1486157	1721324979	0.000005	cosmos182xps9mkr6w8fpp3xd9p6yhu7k98x690sywec

3. 소닉 그리드에 의한 청구서 실행 및 조회 (Execution and Viewing of Bills by Sonic Grid)

- 특수 서비스 노드인 소닉 그리드(Sonic Grid)는 전달된 청구서를 패키징한다.
- 이 청구서를 일정 주기로 배치(batch) 형태로 실행한다.
- 사용자는 [HSSN 대시보드](#)의 **“Fee Settlement”** 탭에서 패키징 및 실행된 청구서 세부 정보를 확인할 수 있다.

HyperGrid

Home

Grids

Syncs

Blocks

Validators

Shared Accounts

Fee

LINKS

Explorer | Sonic

hypergridssn

https://wsrpc-hssn.hypergrid.dev/

Fee

Grid Block Fee

Fee Settlement

ID	FROM ID	END ID	BILL	STATUS	CREATOR
9	29	41	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:15000, FRF59...	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
8	6	29	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:120000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
7	3	5	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:20000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
6	0	2	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:10000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
5	0	2	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:10000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
4	0	2	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:10000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
3	0	2	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:10000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
2	0	2	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:10000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
1	0	2	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:10000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw
0	0	2	[*Erh16dEssgsk3pDTX2Dh1Uhj9LGbjK4wJUrFokLKH86*:10000]	0	cosmos1rin3gp308I90lqyhauq6www8pracm0ahcu54cw

4. 최종 가스 요금 분배 (Final Gas Fee Distribution)

소닉 그리드가 청구서 분배를 실행할 때 다음 작업을 수행한다.

- 계산된 분배 계획에 따라 해당 가스 요금을 참여 노드의 지갑으로 전송한다.
- 여기에는 그리드 노드, HSSN 검증자 노드, 소닉 그리드 공식 노드, 그리고 솔라나 베이스 레이어 노드가 포함된다.
- 모든 분배는 사전에 계산된 규칙 및 계획에 따라 이루어진다.

투명성과 효율성 (Transparency and Efficiency)

이 프로세스는 다음을 보장한다.

- 효율적인 가스 요금 분배(Efficient gas fee distribution)
- 트랜잭션의 투명성(Transparency in transactions)
- HSSN 대시보드를 통한 사용자 청구서 정보 접근 용이성(Easy access to bill information for users via the HSSN Dashboard)

다른 구성 요소와의 관계 (Relationship to Other Components)

HSSN 가스 요금 시스템은 전체 그리드 생태계에서 핵심적인 역할을 수행한다.

- 초기 요금 계산을 위해 개별 그리드와 밀접히 상호작용한다.
- 청구서 전달 및 검증을 위해 HSSN을 활용한다.
- 최종 분배 단계에서 솔라나 베이스 레이어(Solana Base Layer)가 관여한다.
- 투명성과 사용자 접근성을 위해 HSSN 대시보드를 사용한다.

이 시스템은 명확하고 공정한 가스 요금 분배 메커니즘을 제공함으로써, 전체 그리드 생태계의 효율성과 지속 가능성에 기여한다.

하이퍼그리드의 검증 가능한 연산 및 영지식 증명 (Verifiable Compute & Zero-Knowledge Proofing on HyperGrid)

소개 (Introduction)

이 문서의 목적은 하이퍼그리드의 zk 증명 전략(zero-knowledge proving strategy)에 대한 통찰을 제공하는 것이다.

이 간결한 문서는 실사(due diligence) 질문에 대응하기 위한 것으로, 검토 결과가 만족스러울 경우 전체 아키텍처 문서로 확장될 예정이다.

관련 리소스는 본문 내 또는 문서 말미에 하이퍼링크 형태로 제공된다.

그리드 zk-코프로세서 구성 요소 (About the Grid zk-coprocessor Component)

소닉(Sonic)의 솔라나 L1 커밋(commit)을 위한 배치 트랜잭션 처리 계층(batch transaction processing layer)은 현재 **소닉 zk-코프로세서(Sonic zk-coprocessor)**라 불리는 구성 요소로 통합될 예정이다.

이 유닛은 소닉의 그리드 인스턴스뿐 아니라 하이퍼그리드에 의해 오케스트레이션되는 다른 그리드에서도 작동하게 된다.

현재 명칭은 "코프로세서(co-processor)"이지만, 이 구성 요소의 구현 패러다임은 여러 단계에서 다른 체인의 유사 네트워크들과 다르다.

\
따라서, 향후 명칭이 변경될 가능성이 있지만, 현재의 이름은 장기 전략과 잘 부합한다.

zk-코프로세서의 핵심 기능 (Core Functions of the zk-coprocessor)

- 모든 트랜잭션의 처리 및 압축(Processing and compression of all transactions)
- 각 블록의 상태 전환(state transitions)에 대한 증명(proofs) 생성
- 생성된 증명을 솔라나 메인넷에 커밋(Commitment of proofs to Solana mainnet)

이 문서는 다음 주제를 다룬다.

- 소닉 zk-코프로세서의 구현 단계(Implementation phases of the Sonic zk-coprocessor)
- 사용되는 런타임 프리미티브 및 프로토콜(Runtime Primitives & Protocols used)
- 관련 다이어그램과 함께 구현 단계의 고수준 개요(High-level breakdown of the implementation phases accompanied by relevant diagrams)

구현 단계 (Implementation Phases)

솔라나 생태계는 아직 영지식 런타임 프리미티브(zero-knowledge runtime primitives)의 초기 개발 단계에 있으며, **Light Protocol**과 같은 신흥 솔루션이 완전한 조합형 zkSVM을 현실적인 가능성으로 만들고 있다.

소닉의 zk-코프로세서 구현 전략은 두 단계로 구분된다.

1단계: 하이퍼그리드 옵티미스틱 롤업 (Phase 1: HyperGrid Optimistic Rollup)

다가오는 메인넷 릴리스에서 **하이퍼그리드 옵티미스틱 롤업(HyperGrid Optimistic Rollup)** 구현이 우선적으로 진행된다. 이 접근 방식은 구현 기간이 짧기 때문에, 현재 소닉(Sonic)과 통합 중인 게임을 온보딩할 수 있으며 최소 기능 제품(MVP, Minimum Viable Product) 요건을 충족시킨다. 이후에는 소닉 코프로세서(Sonic Co-Processor)의 완전한 영지식 롤업(Zero-Knowledge Rollup) 구현에 집중하여, 소닉의 트랜잭션 처리를 즉각적인 최종성(Instantaneous Finality)에 도달하도록 확장할 것이다.

구현 단계 (Implementation Steps)

1. 트랜잭션이 머클 트리(Merkle Tree)로 압축 및 집계된다.
2. 각 블록에 대해 해당 루트 상태 해시(root state hash)가 커밋된다.
3. 영지식 증명(Zero-Knowledge Proofing)이 수행되어 메인넷에 커밋된다.

2단계: 완전한 영지식 롤업 통합 (Phase 2: Full Zero-Knowledge Rollup Integration)

출시 이후에는 유효성 증명(Validity Proof)과 일관성 증명(Consistency Proof)을 단일 영지식 증명(Zero-Knowledge Proof)으로 통합하는 완전한 영지식 롤업을 구현할 계획이다. 이를 위해 SVM용 zk-회로(zk-circuits)를 구축하며, **Light Protocol** 또는 솔라나 네이티브 zk 프리미티브를 활용하게 된다.

소닉 zk-코프로세서 프리미티브 (Sonic zk-Coprocessor Primitives)

압축 계정 (Compressed Accounts)

소닉 롤업 내의 각 트랜잭션은 **압축 계정(Compressed Account)** 형태로 표현된다. 이러한 계정은 프로그램 소유가 가능하며, 선택적으로 **영구 고유 주소(PDA, Permanent Unique Address)**를 가질 수 있다. 소닉은 **Light Compressed Account** 구현체를 사용할 예정이다.

동시 머클 트리 (Concurrent Merkle Trees)

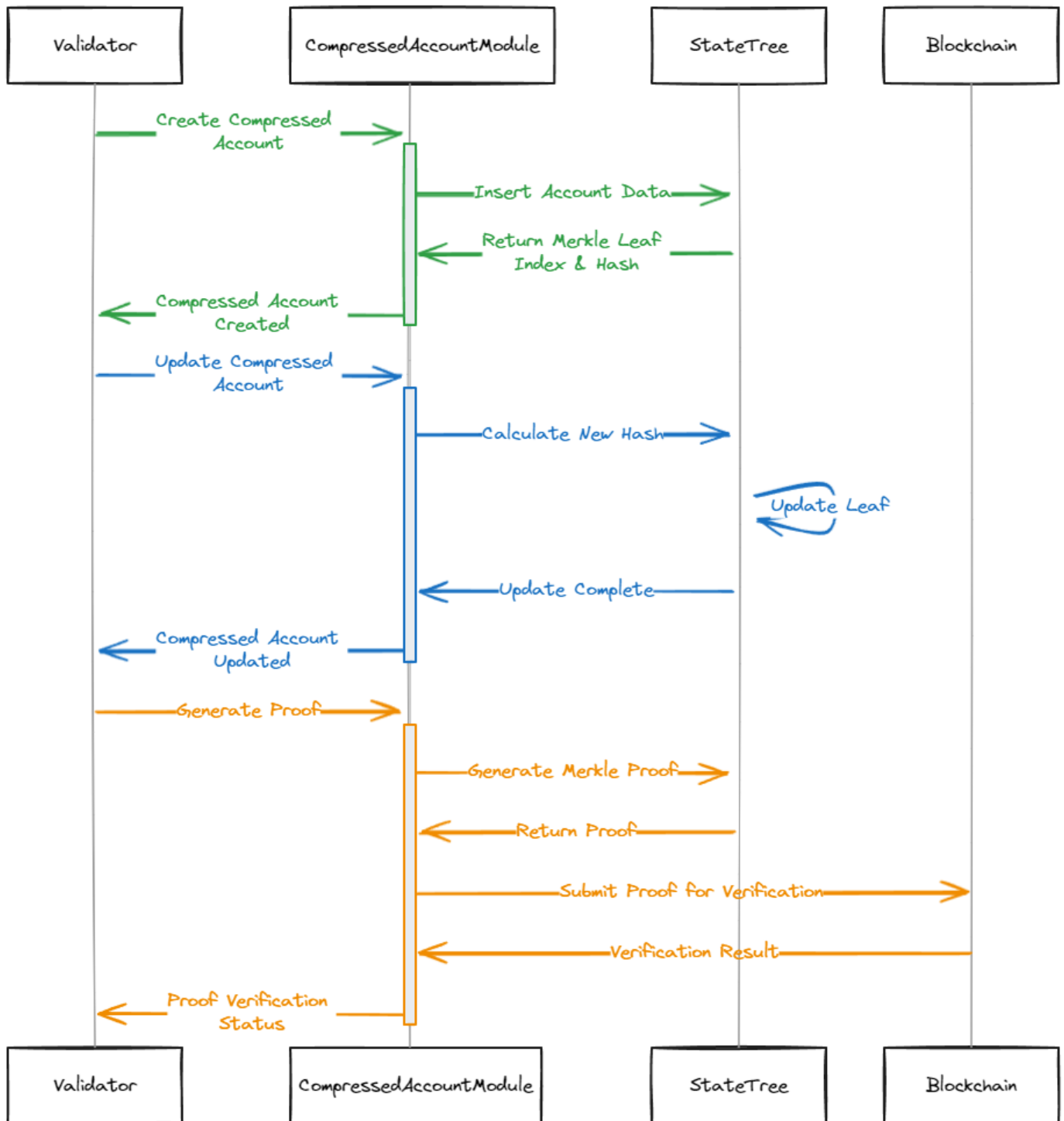
압축된 트랜잭션 상태는 **동시 머클 트리(Concurrent Merkle Tree)** 데이터 구조에 저장된다. 트랜잭션에서 생성되거나 사용된 각 데이터 조각은 상태 트리(State Tree)의 하나의 리프(Leaf)를 나타낸다. 모든 리프는 재귀적으로 해시되며, 온체인에는 최종 32바이트 루트 해시만 저장된다.

단일 솔라나 트랜잭션 내에서 다수의 상태(압축 계정, Compressed Accounts)의 유효성을 검증하기 위해, 소닉은 **Light의 영지식 암호 기술(Zero-Knowledge Cryptography)** 을 활용하여 모든 상태 증명을 약 128바이트 크기의 단일 유효성 증명(Validity Proof)으로 압축한다.

1단계 - 하이퍼그리드 옵티미스틱 롤업 (Phase 1 - HyperGrid Optimistic Rollup)

소닉의 **Go-To-Market(GTM)** 구현에서는 옵티미스틱 롤업 전략을 사용하여 유효성 증명(Validity Proof)을 L1의 검증 프로그램(Verifier Program)에 제출한다. 단계별 절차는 다음과 같다.

1. 트랜잭션이 머클 트리로 압축 및 집계된다.
2. 각 블록에 대해 해당 루트 상태 해시가 커밋된다.
3. 이 블록의 유효성 증명(Validity Proof)은 소닉 상에서 계산된다.
4. 이 새로운 상태와 증명은 솔라나 L1의 소닉 검증 프로그램(Sonic Verifier Program)을 통해 검증된다.



고려된 프로토콜 (Protocols Considered)

- **Light Protocol의 검증기(Verifier)**
- **자체 검증기(In-house Verifier)** — 자체 구축하거나 Light Protocol의 검증기를 사용할 수 있다.

일관성 증명(Consistency Proof)이 유효할 경우, 최신 기록이 베이스 레이어(Base Layer)에 커밋된다. 만약 증명이 유효하지 않을 경우, 증인 노드(Witness Nodes) 또는 관찰 노드(Observation Nodes)가 챌린지를 개시한다.

2단계: 완전한 영지식 롤업 통합 (Phase 2: Full Zero-Knowledge Rollup Integration)

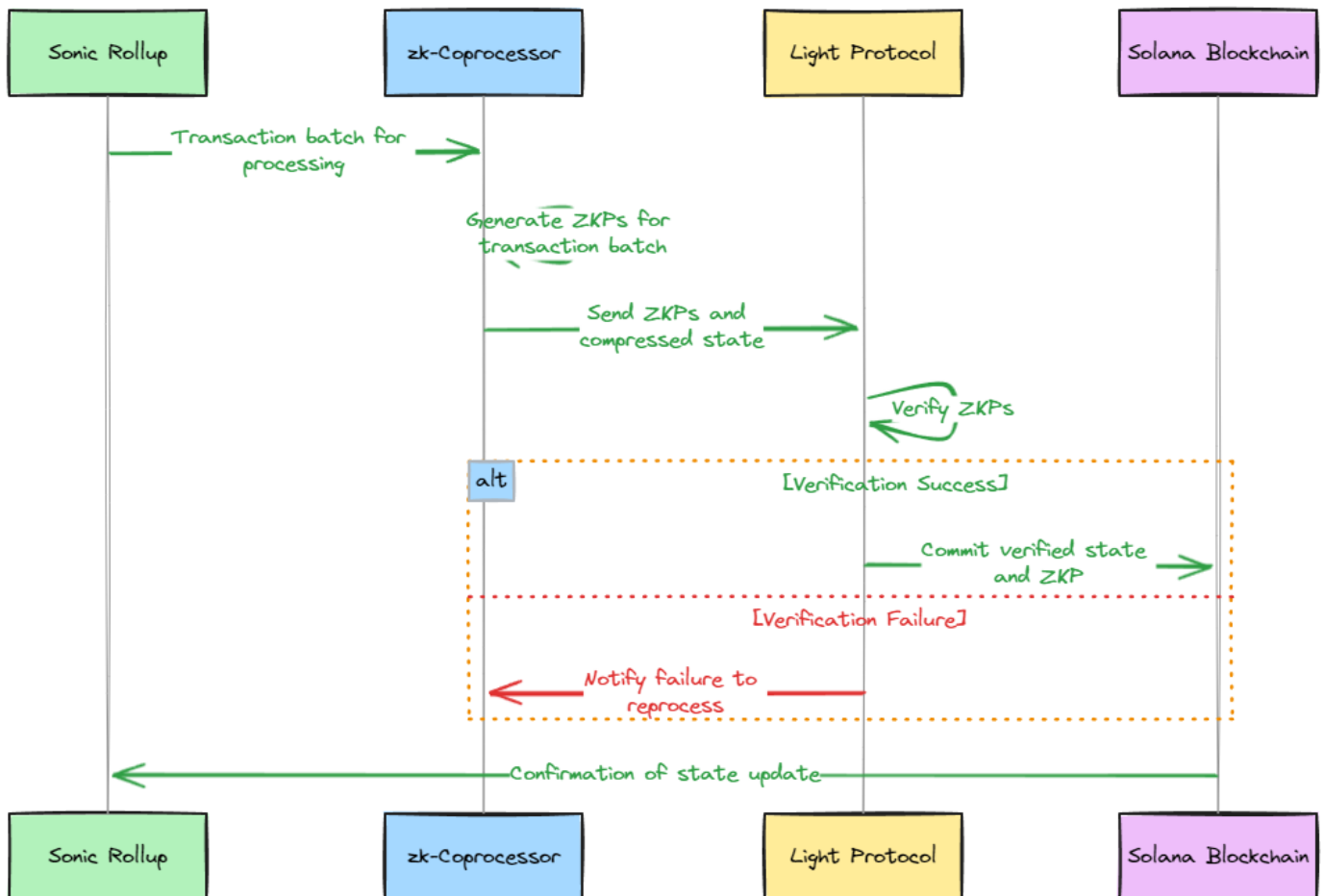
출시 이후 이 단계에서는 유효성 증명(Validity Proof)과 일관성 증명(Consistency Proof)을 단일 영지식 증명(Zero-Knowledge Proof)으로 통합한다. 구현 단계는 다음과 같다.

1. 트랜잭션이 머클 트리로 압축 및 집계된다.
2. 각 블록에 대해 해당 루트 상태 해시가 커밋된다.
3. 영지식 증명이 수행되어 메인넷에 커밋된다.

또한 이 계획은 세 가지 주요 마일스톤으로 나뉜다.

- 트랜잭션 처리를 지원하도록 런타임을 확장(Augment runtime to support transaction processing, building zk-circuits)
- zk-회로의 출력 분석을 통한 zk 런타임 트레이싱(zk runtime tracing by analyzing the output of zk-circuits)
- 런타임 실행 추적(runtime execution trace)으로부터 증명 생성(Proof generation from runtime execution trace)

이 단계의 목표는 소닉(Sonic)에서 L2 트랜잭션의 즉각적인 최종성(Instant Finality)을 실현하는 것이다.



왜 라이트 프로토콜인가? (Why Light Protocol?)

우리가 자체 zkSVM을 구축하는 단계에서, **Light Protocol**은 이미 zkSVM 구현에 필요한 프리미티브 (Primitives)의 상당 부분을 구축한 상태이다.
우리는 Light Protocol 팀과 협업 중이며, 그들의 새로운 개발 동향을 면밀히 모니터링하고 있다.

참고 자료 (References)

- **Light Protocol 공식 문서 (Official Documentation)**
 - **Light Protocol 시스템 개요 (System Overview)**
 - **압축 계정(Compressed Account)**
 - **머클 트리 프리미티브(Merkle Tree Primitive)**
-

Sorada

서론 (Introduction)

소라다(Sorada)는 ****Sonic SVM**의 아카이브 RPC 데이터 인덱싱 및 읽기 최적화 솔루션(Archival RPC Data Indexing and Read-Optimization Solution)

블록체인의 데이터 레이어 문제 (Blockchain's Data Layer Problem)

블록체인 네트워크는 데이터를 블록(blocks), 트랜잭션(transactions), 그리고 메타데이터(metadata)를 통해 집계한다.

이 데이터를 네트워크로부터 직접 가져오기 위한 쿼리 API는 공개되어 있으며, 문서화도 잘 되어 있다.

이 방식은 소규모의 탈중앙화 애플리케이션(DApp)처럼 히스토리컬 데이터 요구량이 적은 경우에는 적합하지만, 대규모의 히스토리컬 데이터가 필요한 애플리케이션에는 두 가지 주요 비즈니스 요건을 충족하지 못한다.

- 성능(Performance)
- 확장성(Scalability)

성능 (Performance)

기업은 네트워크가 제공하는 기본 쿼리 API 엔드포인트의 응답 시간(Response Time)이나 데이터 연산 최적화(Data Operation Optimization)에 거의 제어권이 없다.

이는 비즈니스 요구사항 측면에서 매우 중요하다. 왜냐하면 성능은 웹사이트 속도, 검색 엔진 최적화(SEO), 전환율(Conversion Rate)에 직접적인 영향을 미치기 때문이다.

확장성 (Scalability)

성능 문제와 유사하게, 기업은 네트워크가 제공하는 기본 쿼리 API 엔드포인트의 한계로 인해 사용자 수요에 맞춰 탄력적 자원 프로비저닝(Elastic Provisioning)을 최적화할 수 없다.

이 또한 중요한 비즈니스 요건이다. 왜냐하면 확장성은 단순한 성능 문제를 넘어 운영의 재정적 지속 가능성(Financial Feasibility)에 직접적인 영향을 미치기 때문이다.

블록체인 인덱싱이 이 문제를 해결하는 방법 (How Blockchain Indexing solves this problem)

블록체인 인덱싱(Blockchain Indexing)은 네트워크로부터 직접 히스토리컬 데이터를 쿼리(Query)하여 자체 프로비저닝된 인프라에 저장하는 과정이다.

이 과정을 통해 기업은 네트워크 외부에서 자체적으로 데이터 성능 및 확장성을 관리할 수 있다.

일반적이고 단순화된 블록체인 인덱싱 파이프라인은 다음과 같은 흐름을 가진다.

1. 네트워크 API를 사용하여 blocks 및 transactions 데이터를 가져온다.
2. 가져온 blocks와 transactions를 내부 스토리지 인프라에 저장한다.
3. 네트워크를 직접 쿼리하는 대신, 내부적으로 프로비저닝된 데이터 쿼리 엔드포인트를 통해 제품에서 데이터를 활용한다.

이 방식을 통해 기업은 히스토리컬 데이터 요구사항에 대해 **성능(Performance)** 과 **확장성(Scalability)** 을 내부적으로 통제할 수 있게 된다.

사례 연구 - 소닉 SVM의 85% 아카이브 읽기 비율 (Case Study - Sonic SVM's 85% Archival Reads)

소닉의 내부 하이퍼그리드(Hypergrid) 인프라 모니터링에 따르면, 소닉은 약 **85%의 아카이브 읽기 요청(Archival Read Requests)** 과 **15%의 쓰기 요청(Write Requests)** 을 기록했다.

이러한 읽기 요청은 `getTransaction`, `getBlock`, `getSignaturesForAddress` 엔드포인트를 중심으로 발생했다.

이는 소닉 네트워크의 높은 데이터 접근 수요를 반영하는 지표이지만, 엔지니어링 팀은 하이퍼그리드 인프라로 직접 전달되는 요청의 **대역폭(Bandwidth)** 및 **스토리지(Storage)** 비용에서 병목 현상을 겪기 시작했다.

이러한 병목은 일반적인 아카이브 읽기 요청을 소닉의 메인 하이퍼그리드 인프라에서 분리하고, **소라다(Sorada)** 로 이전함으로써 해결되었다.

소라다란 무엇인가? (What is Sorada?)

소라다는 소닉의 데이터 솔루션(Data Solution) 으로, 주요 목적은 아카이브 읽기 요청(Archival Read Requests) 을 네트워크 쓰기 요청(Network Write Requests) 으로부터 분리(Decoupling) 하는 것이다.

아카이브 읽기 요청을 메인 소닉 검증자(Validator)로부터 분리하여 별도의 데이터 최적화 인프라로 이전 함으로써, 대역폭과 스토리지 병목이 크게 완화되었다.

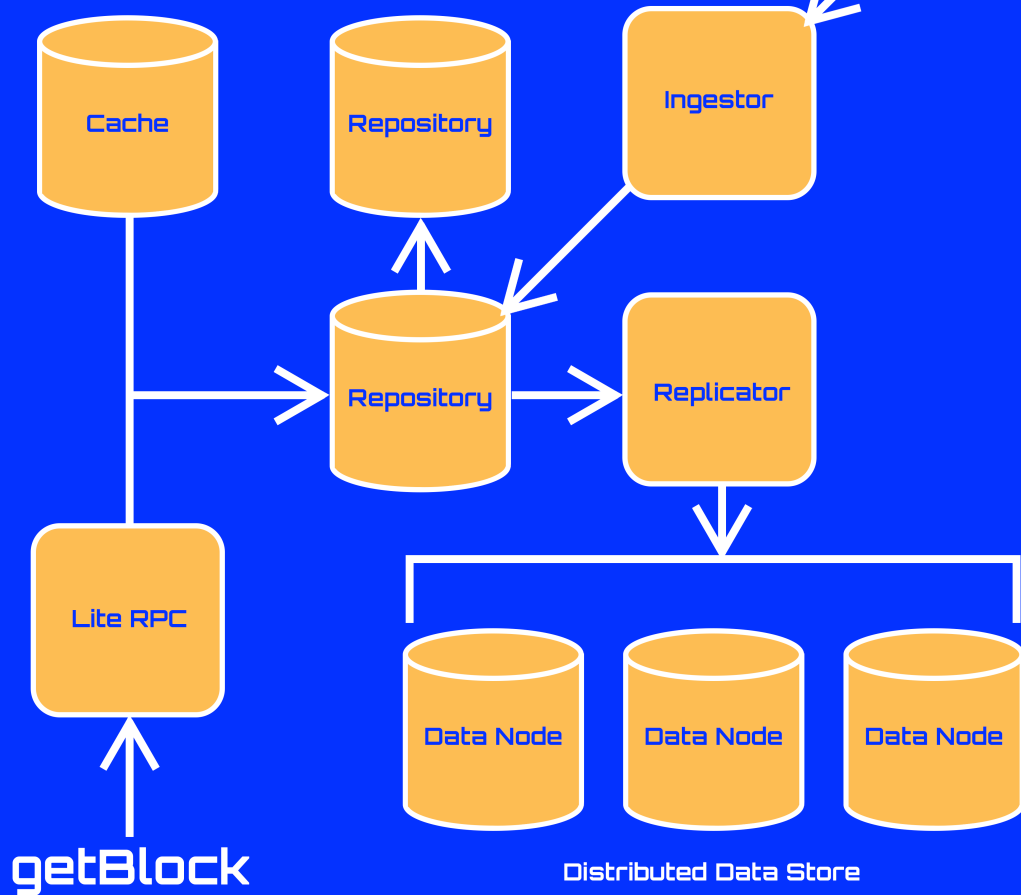
소라다는 SVM 검증자들이 트랜잭션 처리에 더 많은 컴퓨팅 자원을 할당할 수 있게 하며, 동시에 아카이브 요청의 읽기 성능을 **30~40배 향상**시킨다.

아키텍처 (Architecture)

Sorada Architecture

Sonic's Data Solution

Hypergrid



소라다 아키텍처 다이어그램 (Sorada Architecture Diagram)

서비스 (Services)

소라다는 소닉이 읽기 요청을 최적화할 수 있도록 지원하는 여러 서비스로 구성되어 있다.

Sorada Services

Sonic's Data Solution

Lite RPC

Ingestor

Replicator



소라다 서비스 구성도 (Sorada Services)

라이트 RPC (Lite RPC)

라이트 RPC 서비스는 솔라나 풀 노드(Solana Full Node)에서 데이터 검색용 엔드포인트(getBlock, getTransaction)만 유지한 **경량화된 최적화 RPC 서비스**이다.

이를 통해 블록체인 전체 데이터를 유지하지 않고도 효율적인 데이터 조회가 가능하다.

인제스터 (Ingestor)

인제스터(Ingestor) 서비스는 소닉 하이퍼그리드로부터 실시간 데이터를 가져와 **빅테이블 스토리지(Big Table Storage)**에 영구 저장한다.

이는 네트워크 상태 변화에 대한 실시간 인덱싱(Indexing) 및 아카이브 로그 축적을 가능하게 한다.

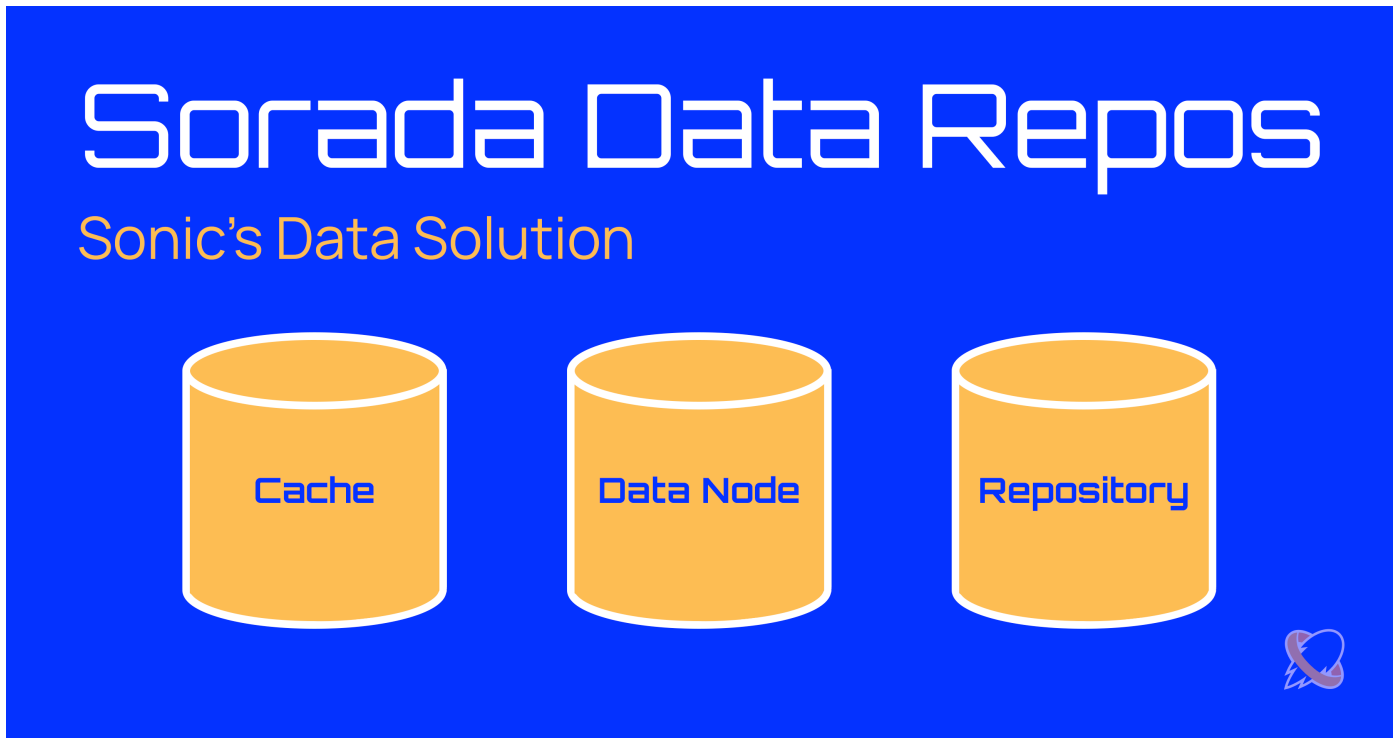
리플리케이터 (Replicator)

리플리케이터(Replicator) 서비스는 빅테이블 스토리지에 인덱싱된 데이터를 **분산 파일 시스템(Distributed Filesystem)**으로 복제(Replication)하여 보다 탄력적이고 안전한 데이터 보존을 보장한다.

이를 통해 단일 장애점(Single Point of Failure)을 제거하고, 고가용성(High Availability)을 유지한다.

데이터 저장소 (Data Repositories)

소라다(Sorada)는 강력한 데이터 보존 인프라를 구축하기 위해 세 가지 유형의 데이터 저장소(Data Repository)를 활용한다.



소라다 데이터 저장소 구성도 (Sorada Data Repositories)

캐시 (Cache)

소라다는 반복적인 요청에 대한 효율적인 데이터 검색을 위해 캐시(Cache)를 활용한다. 이를 통해 자주 조회되는 데이터에 대한 응답 속도를 향상시키고, RPC 요청 부하를 줄인다.

빅테이블 (Big Table)

소라다는 **빅테이블(Big Table)** 데이터베이스를 기본(primary) 데이터 저장소로 사용한다. 트랜잭션, 블록, 서명(Signatures) 등 대규모 구조화 데이터를 효율적으로 인덱싱하고, 실시간 읽기 및 기록 요청을 동시에 처리한다.

분산 파일 시스템 (Distributed Filesystem)

소라다는 **자가 복제(Self-Replicating)**형 분산 파일 시스템을 사용하여 데이터가 여러 데이터 노드에 안전하게 보존되도록 한다. 이 시스템은 노드 장애 시에도 데이터 무결성을 유지하며,고가용성(High Availability) 및 내결함성(Fault Tolerance)을 보장한다.

리소스 (RESOURCES)

감사 보고서 (Audit Reports)

1. [Sonic SVM Grid 감사 보고서 \(Sonic SVM Grid Audit Report\)](#)
2. [하이퍼그리드 공유 상태 네트워크\(HSSN\) 감사 보고서 \(HyperGrid Shared State Network Audit Report\)](#)
3. [Sonic SVM 토큰 컨트랙트 감사 보고서 \(Sonic SVM Token Contract Audit Report\)](#)
4. [veSonic 컨트랙트 감사 보고서 \(veSonic Contract Audit Report\)](#)
5. [Sonic 스테이킹 컨트랙트 감사 보고서 \(Sonic Staking Contract Audit Report\)](#)