

알트레이어(AltLayer)

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.

개요(Overview)

AltLayer란 무엇인가? (What is AltLayer?)

AltLayer는 롤업(Rollup)을 위한 개방적이고 탈중앙화된 프로토콜이다. AltLayer는 리스테이킹 롤업 (Restaked Rollup)이라는 새로운 개념을 도입하여, OP Stack, Arbitrum Orbit, ZKStack, Polygon CDK 등 어떤 롤업 스택으로 구축된 기존 롤업에도 향상된 보안성, 탈중앙화, 상호운용성, 그리고 암호경제적 신속 최종성(fast finality)을 제공한다.



리스테이킹 롤업(Restaked rollup)은 OP Stack, Arbitrum Orbit, ZKSync의 ZKStack, Polygon CDK 등과 같은 롤업 스택을 사용해 손쉽게 롤업을 구축할 수 있는 편의성과, EigenLayer의 리스테이킹 메커니즘(restaking mechanism)을 결합하여 네트워크 보안을 강화하고 탈중앙화된 네트워크를 구성한다.

리스테이킹 롤업은 특정 롤업을 위해 온디맨드 방식으로 생성되는 세 가지 수직 통합형 능동 검증 서비스 (Actively Validated Services, AVS)의 집합으로, 어떤 롤업 스택과도 연동되어 작동할 수 있다. 이 AVS들은 특히 애플리케이션 특화 롤업(app-specific rollup)에 대해 다음 세 가지 핵심 서비스를 제공한다.

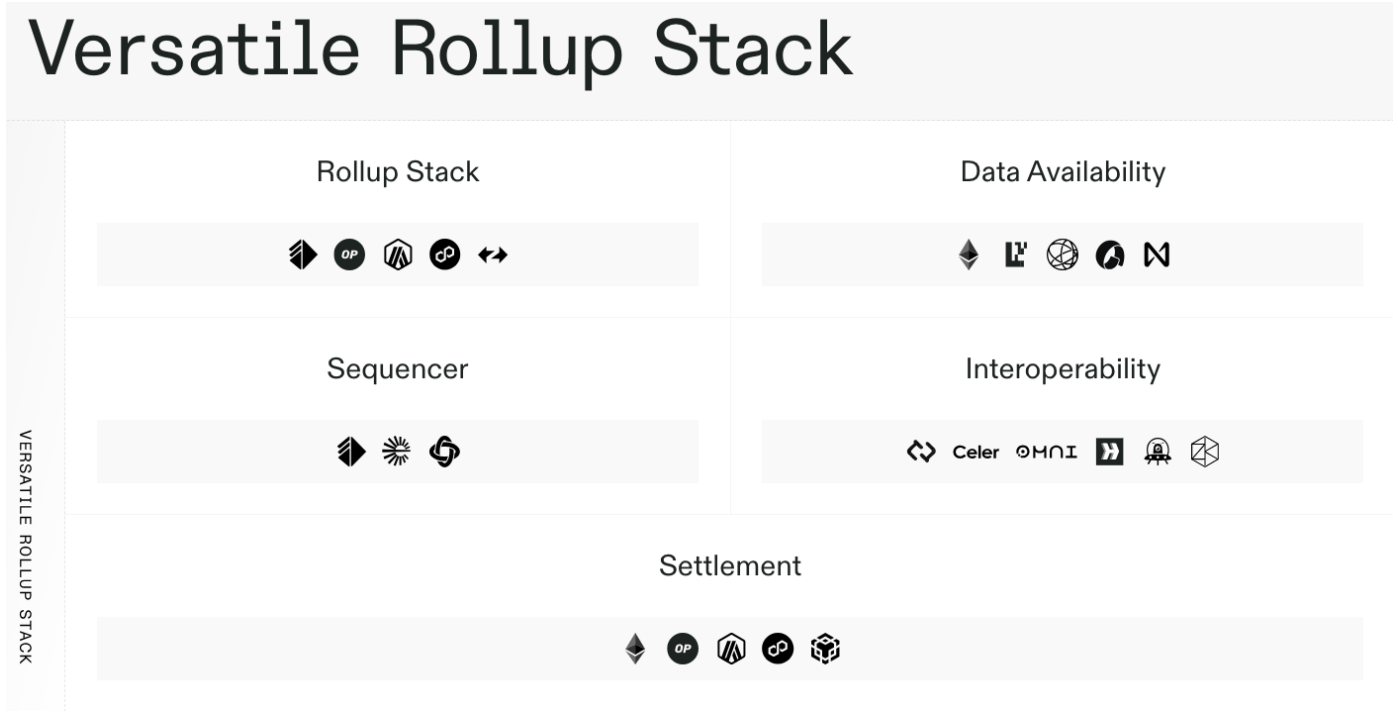
1. 롤업 상태의 정확성 검증
2. 신속한 최종성(Fast Finality)
3. 탈중앙화 시퀀싱(Decentralised Sequencing)

이러한 서비스는 세 가지 모듈형 구성요소로 제공된다.

- VITAL (롤업 상태의 탈중앙화 검증을 위한 AVS)
- MACH (신속한 최종성을 위한 AVS)
- SQUAD (탈중앙화 시퀀싱을 위한 AVS)

이 프로토콜 위에 구축된 AltLayer는 개발자뿐만 아니라 코딩 경험이 거의 없는 사용자도 몇 번의 클릭만으로 단 2분 안에 맞춤형 롤업을 실행할 수 있는 다기능 노코드(no-code) 롤업-서비스형(Rollups-as-a-Service, RaaS) 런치패드를 제공한다. 이 RaaS 제품은 멀티체인 및 멀티 VM 환경을 염두에 두고 설계되어, EVM과 WASM을 모두 지원한다. 또한 OP Stack, Arbitrum Orbit, Polygon zkEVM, ZKSync의 ZKStack, Starkware 등 다양한 롤업 SDK뿐 아니라 Espresso와 Radius 같은 공유 시퀀

싱 서비스, Celestia, EigenLayer, Avail 등 다양한 데이터 가용성(DA) 레이어를 비롯해 롤업 스택의 여러 모듈형 서비스를 지원한다.



또한 AltLayer가 롤업 설계 분야에 도입한 주요 혁신 중 하나는 **에페메럴 롤업(ephemeral rollup)** 개념이다. 에페메럴 롤업을 통해, 특정 애플리케이션의 트래픽 증가를 예상하는 dApp 개발자는 다음과 같은 방식으로 동작할 수 있다: 1) 레이어 1에 의해 보안이 보장되는 빠르고 확장성 높은 애플리케이션 맞춤형 롤업을 신속하게 구축하고, 2) 필요한 기간 동안 해당 롤업을 사용한 뒤, 3) “수명 종료(end-of-life)” 정산(settlement)을 통해 레이어 1에 롤업을 종료할 수 있다. 에페메럴 롤업은 애플리케이션 특화 롤업과 범용 레이어 1의 장점을 모두 갖춘 고도로 자원 효율적인 구조이며, RaaS 프레임워크 내에서 AltLayer는 에페메럴 롤업과 영구형(persistent) 롤업을 모두 제공한다.

AltLayer 프로토콜은 팀들이 상당한 자본과 수년의 개발 기간을 절약할 수 있도록 하며, 완전히 개방적이고 퍼미션리스한 환경에서 혁신과 빠른 실험을 촉진한다.

왜 AltLayer인가? (Why AltLayer?)

우리는 수천 개의 롤업이 존재하는 세상을 상상한다. 일부는 Arbitrum One, Optimism Mainnet, ZKSync 등과 같이 범용 목적(general-purpose)을 위해 설계된 롤업이며, 다른 일부는 이종적이고 모듈형 기술 스택으로 구축된 애플리케이션 특화(app-specific) 롤업이다. 예를 들어, 한 롤업은 Arbitrum Orbit으로 구축되어 Arbitrum One을 데이터 가용성(DA) 및 정산(settlement) 레이어로 사용할 수 있고, 또 다른 롤업은 ZK Stack으로 구축되어 Celestia를 DA 레이어로, Ethereum을 정산 레이어로 사용할 수도 있다.

이러한 이질적인 환경에서는 다음과 같은 필수 기능을 비롯한 다양한 핵심 요구를 충족하기 위해 롤업들이 접근할 수 있는 보편적이고 중립적인 프로토콜이 필요하다.

1. 탈중앙화 검증(Decentralised Verification): 즉시 인출(Instant Withdrawal) 및 보안 보장
 2. 신속한 최종성(Faster Finality) 및 롤업 간 상호운용성(Cross-Rollup Interoperability)
 3. 유연하고 탈중앙화된 시퀀싱(Flexible and Decentralised Sequencing)
-

문제 1 (왜 탈중앙화 검증인가?): 옵티미스틱 롤업(Optimistic Rollup)은 롤업 운영자가 Ethereum에 커밋한 상태(state)가 실제로 올바른 상태 전이 함수(state transition function)에 따른 결과인지 검증하는 ‘정직한 검증자(honest validator)’가 최소 한 명 존재한다는 전제하에 보안을 유지한다.

그러나 많은 애플리케이션 특화 롤업은 범용 롤업만큼 성숙한 생태계를 보유하지 못하고 있어, 네트워크를 검증할 충분한 참여자나 인센티브가 부족할 수 있다.

또한, 최근에는 L1 Ethereum 스마트컨트랙트 내에서 옵티미스틱 패턴을 사용하는 새로운 형태의 시스템이 등장하고 있다. 예를 들어, 어떤 사용자가 연산 결과를 커밋(commit)하고, 이후 부정행위가 감지되면 사기 증명(fraud proof)을 통해 검증하는 방식이다. AltLayer가 제안한 에페메럴 롤업도 이와 유사한 사례로, 롤업 시퀀서가 정산 시점에 Ethereum에 롤업 상태를 커밋하며, 해당 상태는 다크 포레스트(Dark Forest)와 같은 특정 애플리케이션을 위한 오프체인 연산 결과를 나타낸다. 이 온디맨드형 시스템에서는 롤업 상태를 감시하고, 이상 상태를 도전(challenge)할 수 있는 탈중앙화된 검증자 네트워크의 존재가 필수적이다.

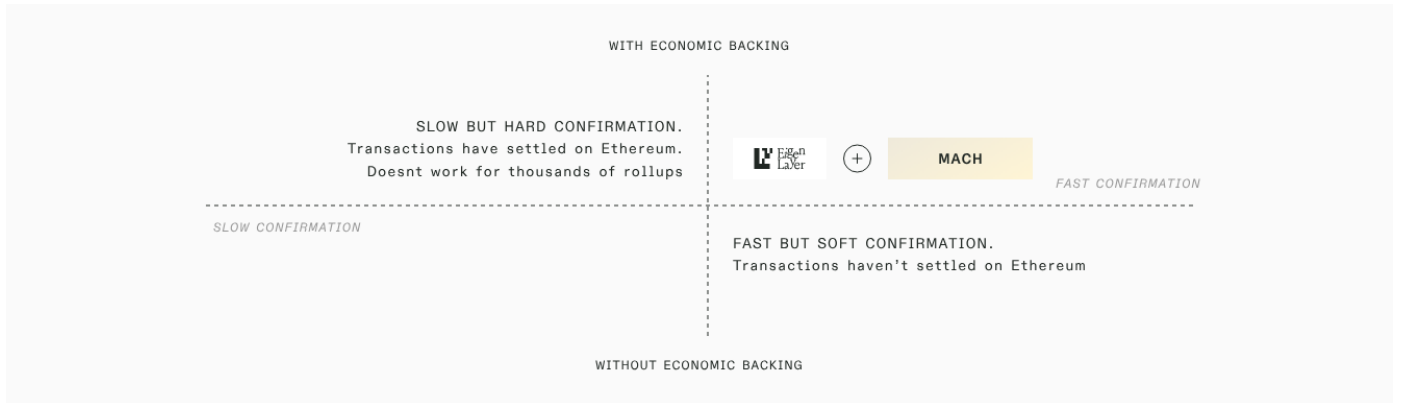
검증은 단순히 보안을 위한 요소일 뿐 아니라, 상호운용성을 보장하기 위해서도 필수적이다. 사실상 모든 크로스도메인 메시징(cross-domain messaging)은 시스템 상태를 검증해 독립적인 다른 시스템에서 후속 동작을 수행할 수 있어야 한다는 전제 위에 성립한다.

문제 2 (왜 신속한 최종성인가?):

Ethereum은 거의 100만 개의 검증자(validator)를 보유한 가장 탈중앙화된 네트워크이며, 또한 가장 높은 유동성(liquidity)을 가진 네트워크이기도 하다. 따라서 대부분의 롤업이 Ethereum을 정산 레이어로 선택하는 것은 당연한 일이다. 그러나 Ethereum은 트랜잭션이 최종 확정(finalize)되기까지 약 13분이 걸릴 정도로 속도가 빠른 네트워크는 아니다.

이처럼 트랜잭션 최종화까지 시간이 오래 걸리기 때문에, 게임이나 소셜 앱처럼 낮은 지연(latency)을 요구하는 애플리케이션을 위한 롤업은 종종 “시퀀서가 추후 온체인에 트랜잭션을 게시하겠다”는 약속에 의존한다.

이러한 방식은 신뢰할 수 있는 주체가 시퀀서를 운영할 경우에는 수용 가능할 수 있지만, 시퀀서 운영 주체에 대한 신뢰가 없을 경우에는 아무런 보장을 제공하지 못한다. 더욱이 중앙화된 시퀀서는 경제적 이해관계가 없으므로, 그들이 제공하는 ‘소프트 최종성(soft finality)’은 경제적 담보가 없는 단순한 약속에 불과하다.



또한 시퀀서가 제공하는 최종성 보장이 약하기 때문에, 시퀀서가 발행한 영수증(receipt)은 상호운용성을 위해 충분하지 않다. 예를 들어, 롤업 A에서 토큰을 소각(burn)하고 롤업 B에서 발행(mint)하는 형태의 크로스롤업 메시지의 경우, 롤업 B는 롤업 A의 시퀀서가 실제로 해당 수량의 토큰을 소각했는지에 대한 강력한 보장을 필요로 한다.

문제 3 (왜 탈중앙화 시퀀싱인가?):

현재 대부분의 롤업은 프로토콜을 개발한 주체가 운영하는 중앙화된 시퀀서를 사용한다. 단일 시퀀서가 롤업을 운영하는 것은 이상적이지 않지만, 이미 잘 알려진 범용 롤업의 경우 운영 주체가 명성이 있는 만큼, 그들이 평판을 해칠 행동을 하지 않을 것이라는 신뢰로 인해 일시적으로는 허용될 수 있다. 그러나 이러한 ‘비프로토콜 신뢰(off-protocol trust)’는 신원이 알려지지 않은 개발자가 운영하는 수많은 애플리케이션 특화 롤업에는 적용될 수 없다.

중앙화 시퀀싱의 또 다른 문제는 RaaS 제공업체와 관련이 있다. RaaS 제공업체는 롤업을 구축하려는 팀에게 재정적·인적 자원을 절약할 수 있는 유용한 서비스를 제공하지만, 그들의 비즈니스 모델은 중앙화된 시퀀싱과 실행 구조에 크게 의존한다. 대부분의 RaaS 제공업체는 시퀀싱 수익의 일부를 수수료로 취하며, 따라서 사업 지속성을 위해 고객을 가능한 한 오랫동안 자사 플랫폼에 묶어두려는 유인을 갖는다. 이러한 벤더 종속(vendor lock-in)은 특정 RaaS 제공업체가 일방적으로 수수료 구조를 변경하거나, 임의의 MEV를 추출해 수익을 극대화하려는 위험을 초래할 수 있다.

게다가 대부분의 RaaS 제공업체가 채택한 중앙화된 시퀀서/실행자 모델은 원칙적으로 개방성과 투명성, 탈중앙화를 지향하는 웹3 철학에 반하는 폐쇄적 환경을 만든다.

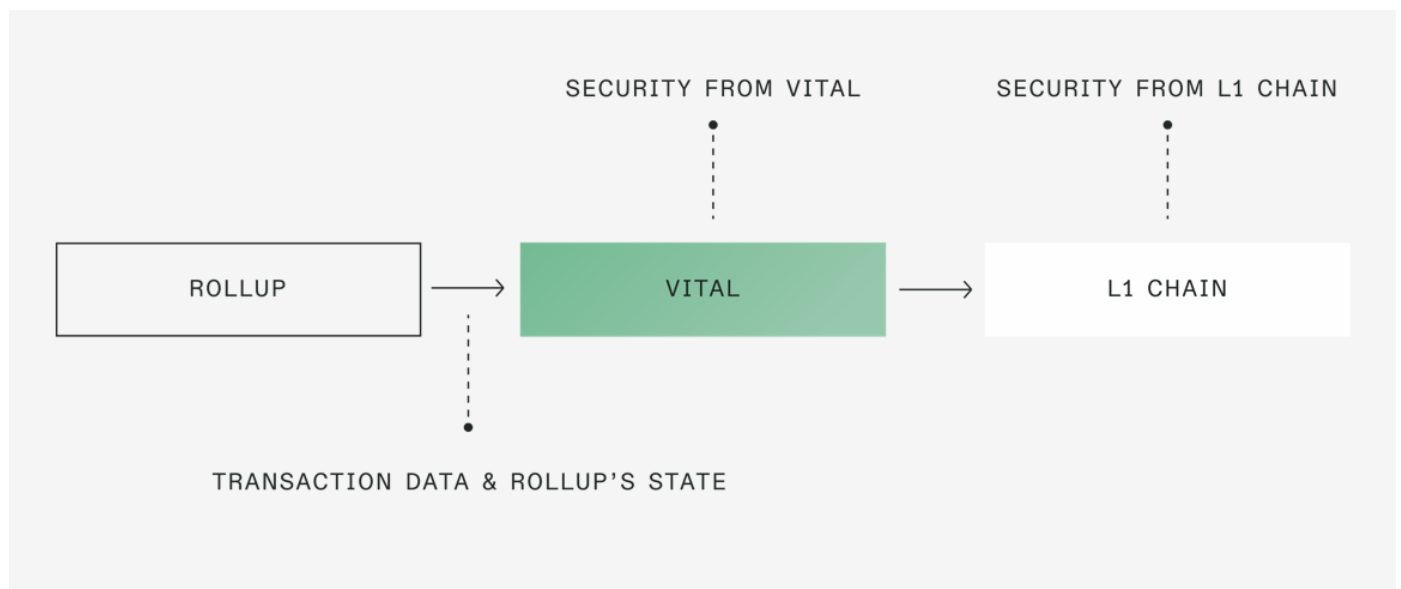
AltLayer는 이러한 문제들을 해결하기 위해 각 롤업을 지원하는 다수의 AVS 세트를 구축한다. 또한 dApp 개발자가 AVS 운영자를 선택하는 방식, 보상 구조, 운영 규칙, 그리고 규칙 위반 시의 슬래싱(slashed) 조건을 직접 설정할 수 있도록 높은 유연성을 제공한다.

리스테이킹 롤업(Restaked Rollups)

탈중앙화 검증을 위한 VITAL (VITAL for Decentralised Verification)

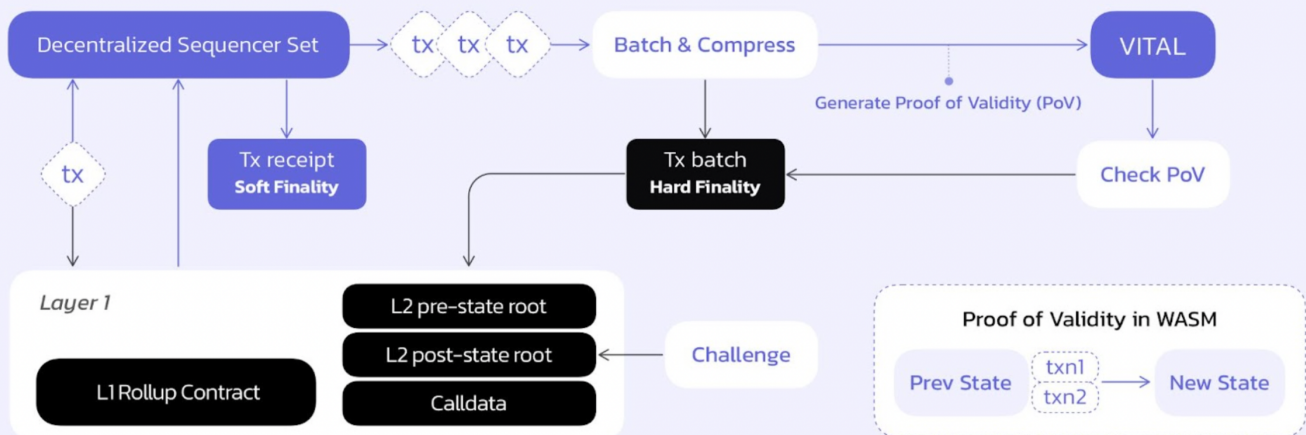
VITAL은 롤업을 위한 내장형 검증 레이어로 작동한다. 이는 AVS에 등록된 운영자(operator) 네트워크로 구성되며, SQUAD 운영자가 제안하는 모든 신규 상태를 검증한다. VITAL 운영자는 잘못된 상태 루트(state root)를 감지하고, 분할(bisection) 프로토콜에서 SQUAD 운영자에게 도전(challenge)할 수 있다.

VITAL은 또한 옵티미스틱 ZK 증명(optimistic ZK proofs) 방식으로 동작할 수 있는데, 이 경우 VITAL 운영자는 분할 프로토콜에 참여하는 대신, 문제가 된 상태 루트에 대해 SQUAD 운영자에게 ZK 증명의 생성을 요구한다. 또 다른 동작 모드는 L1로 제출할 필요가 없는 중간 증명(intermediate proofs)을 검증하는 것이다. VITAL은 MACH AVS가 신속 최종성(fast finality) 레이어를 제공하기 위해 이를 활용한다는 점에서 핵심적이다.



VITAL은 신규 상태를 검증하는 운영자 네트워크로 구성된다. 롤업 시퀀서가 제공하는 사전 확정(pre-confirmation)과 달리, VITAL이 제공하는 확정은 그 기저의 경제적 담보(economic backing)로 인해 훨씬 더 강력하다. 이는 자산을 즉시 인출할 수 있게 해준다.

Verify Rollup's State Transition Function



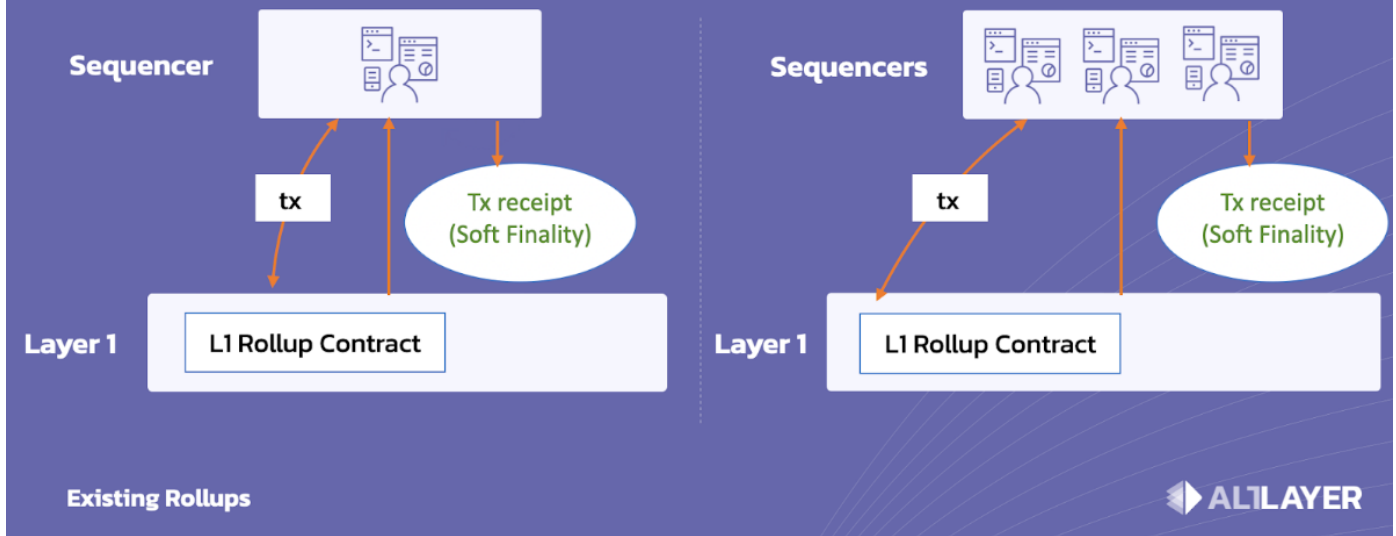
Txn batch is submitted after PoV verification. This lowers the chances of a challenge via fraud proofs.

티어-1 최종성(Tier-1 Finality)

이를 이해하기 위해, 오늘날 Arbitrum 또는 Optimism에서의 롤업 트랜잭션 생애주기를 생각해 보자. 이러한 네트워크는 중앙화된 시퀀서로 운영되므로, 시퀀서는 사용자의 트랜잭션을 수신·처리하고 소프트 최종성(soft finality)에 대한 증거로서 트랜잭션 영수증(receipt)을 발행한다. 이 트랜잭션은 시퀀서에게 직접 전달될 수도 있고(예: 일반 트랜잭션), L1에서 전달될 수도 있다(Layer 2 → Layer 1의 인출 요청인 경우). 그러나 소프트 최종성이 단일 중앙화 시퀀서에 의해 발행되므로, 여기서 제공되는 보장은 대부분 “신뢰 기반(trust me)” 보장에 불과하다. 또한 시퀀서가 가용하지 않게 되는 상황에서는 이 보장 자체를 애초에 얻을 수 없다.

반면 AltLayer의 롤업은 SQUAD를 통해 탈중앙화 시퀀서를 가지며, 이 시퀀서들은 상호 간 합의 프로토콜을 실행할 수도 있다. 따라서 이 수준에서 제공되는 소프트 최종성 보장은 중앙화 시퀀서를 사용하는 롤업이 제공하는 동일 수준의 보장보다 실제로는 다소 더 “강한(harder)” 보장이다. 우리는 이를 티어 1 최종성(Tier 1 finality)이라 부른다.

Better Security Guarantees from Decentralized Sequencing



티어-2 최종성(Tier-2 Finality)

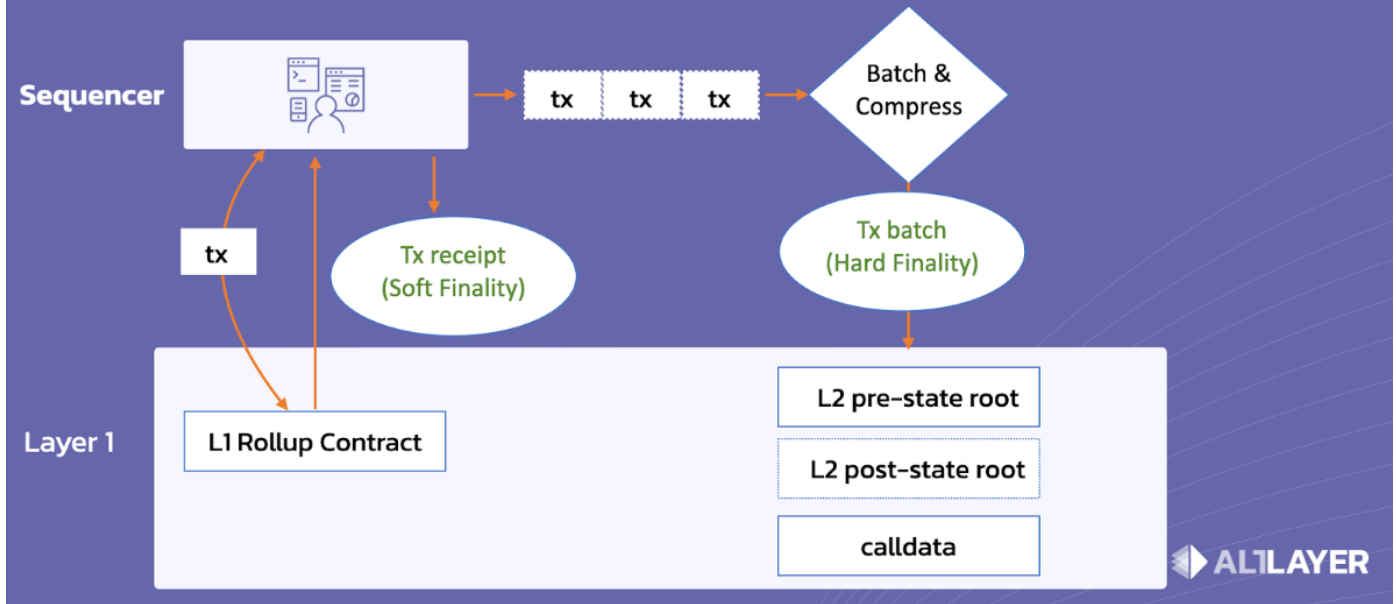
대부분의 롤업에서, 1~3분 후 시퀀서는 그동안 처리된 트랜잭션들을 배치(batch)로 묶어 calldata를 사용해 기저 레이어 1로 푸시한다. 이 데이터는 온체인 스토리지로 남지만 L1의 전역 계정 상태(global account state)의 일부가 아니므로 다른 컨트랙트에서 접근할 수 없다. 이제 L2 트랜잭션은 그것이 포함된 L1 블록과 동일한 최종성(hard finality)을 갖게 된다.

트랜잭션 데이터에 더해, 시퀀서는 구(old, 배치된 트랜잭션 이전의 상태)와 신(new, 배치된 트랜잭션이 실행된 이후의 상태) 상태 루트(state root)도 제출하는데, 이는 상태 변경의 정당성을 입증하는 데 필요하다.

시퀀서가 배치를 제출하면, 컨트랙트는 사전 상태 루트가 기존 상태 루트와 일치하는지 검증한다. 두 값이 일치하면, 컨트랙트는 오래된 상태 루트를 폐기하고 시퀀서가 제안한 새로운 상태 루트를 저장한다.

이 단계에서 L1은 시퀀서가 새로운 상태 루트를 생성했다고 주장하는 모든 트랜잭션을 보유하고 있음을 유의해야 한다. 그러나 시퀀서의 주장과 그 상태 루트에 대한 커밋이 올바르다는 보장은 없다. 바로 이러한 이유로 옵티미스틱 롤업에는 누구나 시퀀서의 주장을 다룰 수 있도록 출금 대기 기간(withdrawal period)이 존재한다.

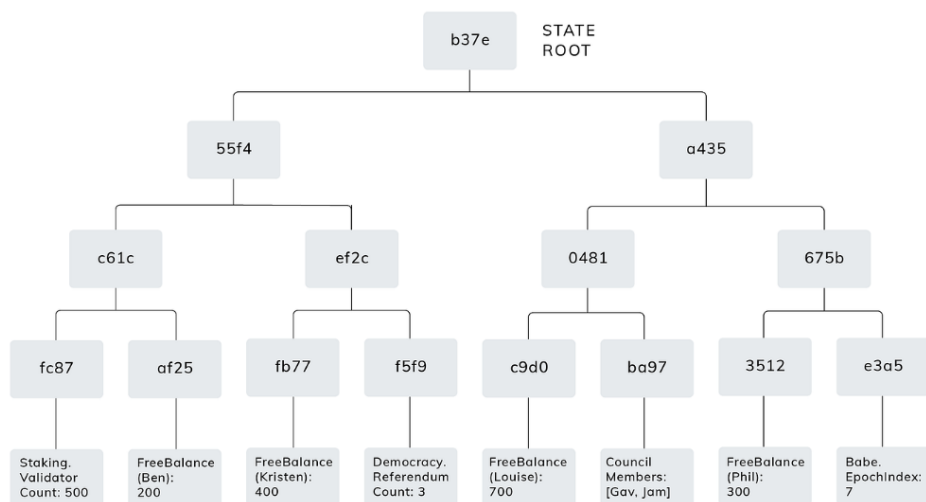
Hard Finality with Other Rollups



AltLayer는 이를 두 가지 방식으로 개선한다. 첫째, SQUAD를 통한 시퀀싱 네트워크가 탈중앙화되어 있고 시퀀서들이 합의 프로토콜을 실행할 수 있으므로, 시퀀싱 네트워크가 잘못되거나 부정확한 상태 루트를 생성할 가능성은 낮다.

둘째, 더 중요한 점은 시퀀서들이 VITAL 운영자에게 ‘유효성 증명(proof of validity)’이라 불리는 메커니즘을 통해 상태 전이의 타당성 검증을 의존한다는 것이다. VITAL은 유효성 증명을 통해 시퀀서의 책임성을 담보하는 중간 네트워크로 기능한다.

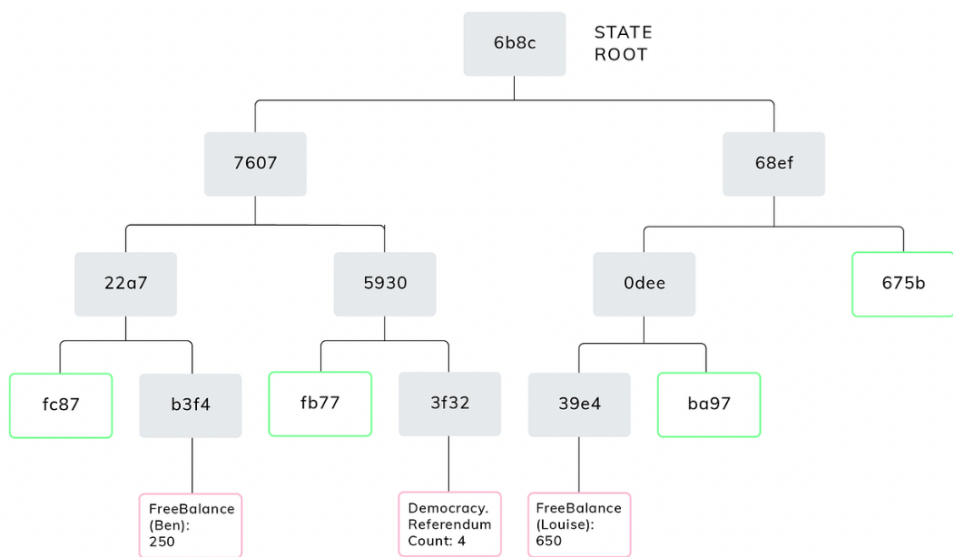
유효성 증명(Proof of Validity): 롤업의 상태는 머클 트리(Merkle tree)에 저장된다. 예를 들어, 키는 계정 ID가 될 수 있고 값은 해당 계정이 보유한 토큰 수가 될 수 있다.



AltLayer에서는 롤업 네트워크가 ‘유효한 상태 전이의 증명(유효성 증명, proof of validity)’을 생성하여 VITAL 운영자가 전체 상태에 접근하지 않고도 롤업의 상태 전이를 검증할 수 있게 한다. 실제로 필요한 것은 다음과 같다.

- 블록(상태 전이 목록),
- 해당 블록이 변경하는 상태 데이터베이스의 값들,
- 머클 트리에서 변경되지 않은 지점들의 해시.

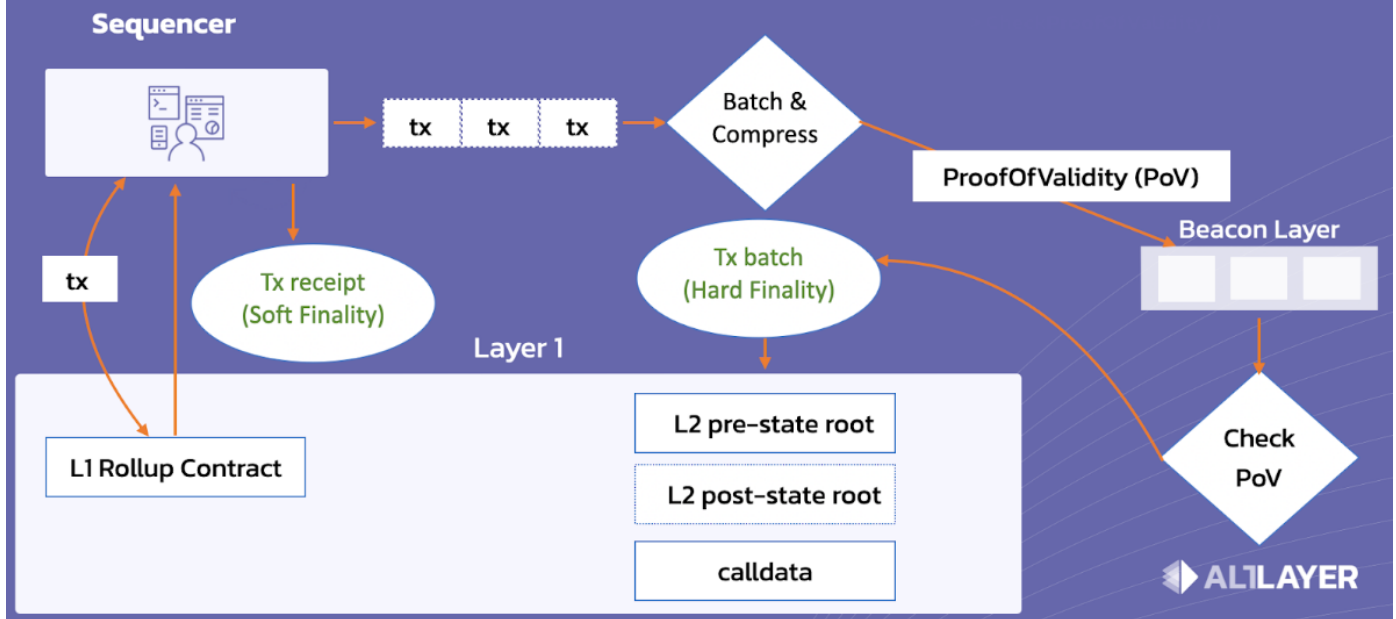
이 정보들을 통해 VITAL 운영자는 새로운 상태 루트를 생성하고, 시퀀서가 주장한 상태 루트와 자신이 생성한 상태 루트가 동일한지 비교할 수 있다. 그 결과, VITAL은 롤업을 위한 탈중앙화된 ‘상태 비보관 (stateless) 라이트 클라이언트(light client)’ 네트워크로 기능한다.



롤업 네트워크는 유효성 증명을 생성하여 주기적으로(롤업 블록 6개마다) PoV 블록이라 불리는 특수 블록에 담아 VITAL에 제출한다. 롤업 네트워크가 생성한 유효성 증명을 검증할 수 있도록, VITAL 운영자는 검증자의 런타임의 일부인 웹어셈블리(WebAssembly) 코드 형태의 상태 전이 검증 함수(STVF, state transition validation function)를 구현한다. STVF는 입력으로 PoV 블록을 받아 해당 블록의 유효성을 출력한다. 증명이 유효하면, VITAL은 트랜잭션 배치와 사전/사후 상태 루트를 L1에 제출한다. 기존 롤업에서는 이 과정을 시퀀서가 직접 수행하지만, AltLayer의 경우 이는 VITAL 운영자의 책임이다.

VITAL은 시퀀서 수준의 사기(fraud)를 훨씬 이른 단계에서 감지하므로, 기저 레이어 1에서 분쟁이 발생할 가능성을 감소시킨다. 간단히 말해, VITAL은 사기 판단을 위한 하급심(lower court)으로, 기저 레이어 1은 최종심(supreme court)으로 작동한다. 그 결과, VITAL은 시퀀서만 있는 롤업 대비 더 나은 최종성 보장을 제공한다. 우리는 이를 티어 2 최종성(Tier 2 Finality)이라 부른다.

Proof of Validity Verification on Beacon Layer



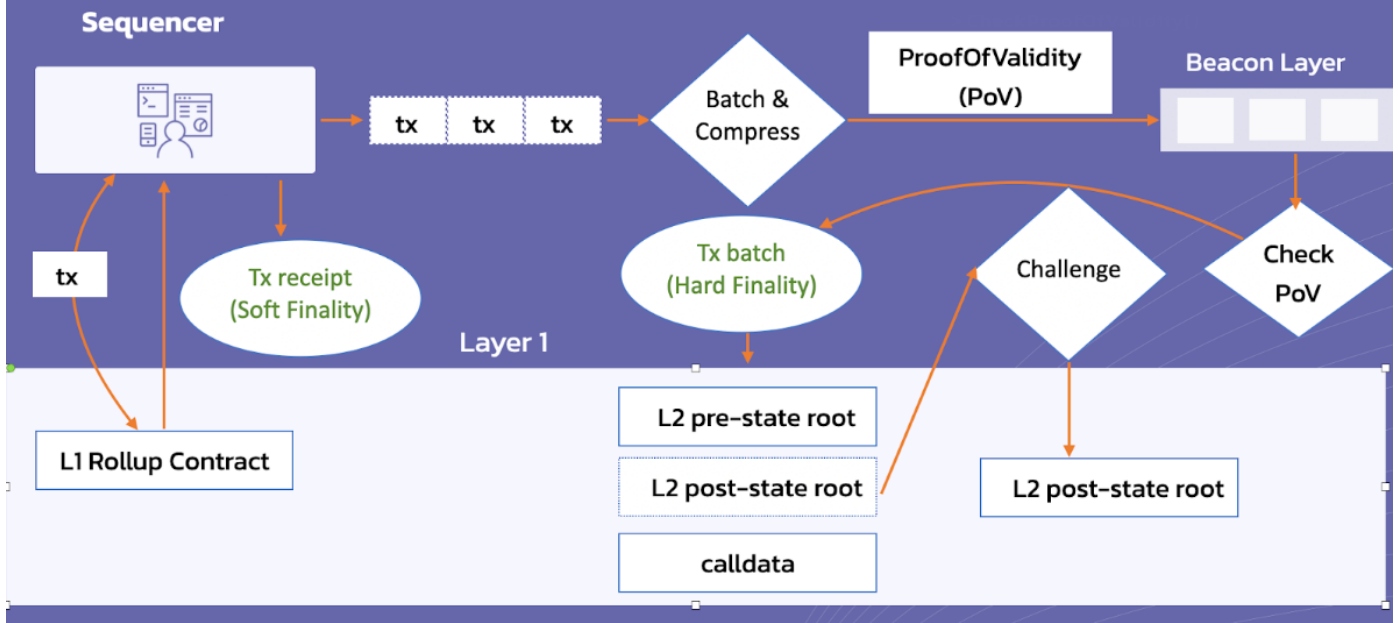
티어-3 최종성(Tier-3 Finality)

이 단계에서 트랜잭션 데이터와 새로운 상태는 레이어 1에 커밋된다. 이제 롤업 시스템은 챌린지 기간 (challenge period)에 진입한다.

제안된 상태 커밋(state commitment)이 챌린지 기간 동안 아무도 이의를 제기하지 않으면, 해당 커밋은 최종적(final)으로 간주되며, 이후 레이어 1의 스마트컨트랙트는 그 커밋을 기반으로 롤업 상태에 대한 인출 증명(withdrawal proof)을 안전하게 수락할 수 있다. 반면, 상태 커밋이 성공적으로 이의 제기될 경우, 해당 잘못된 배치(batch)와 그 이후에 게시된 모든 배치는 되돌려지며, 롤업은 이전 상태 루트로 복원된다. 이후 롤업 프로토콜은 트랜잭션을 다시 실행하고 롤업의 상태를 재갱신해야 한다.

챌린지 기간이 끝나면, 레이어 2에서 레이어 1로의 인출이 최종적으로 허용된다. 인출은 레이어 2에서 시작되어 레이어 1에서 실행되는 트랜잭션으로 마무리되는 크로스도메인 트랜잭션(cross-domain transaction)이다. 예를 들어, 레이어 2 계정에서 레이어 1 계정으로 토큰을 전송하는 경우가 이에 해당한다. 사용자는 calldata를 사용하여 머클 증명(Merkle proof)을 롤업 컨트랙트에 제출하면, 자신의 트랜잭션이 롤업의 상태 루트에 포함되었음을 증명할 수 있다.

Proof of Validity Verification on Beacon Layer



신속한 최종성을 위한 MACH (MACH for Faster Finality)

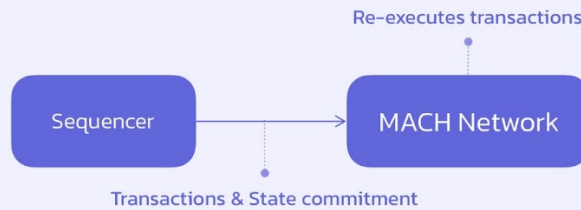
롤업의 느린 최종성 문제를 해결하기 위해, 우리는 다음의 핵심 목표(desiderata)를 가진 Ethereum 롤업용 신속 최종성 레이어인 **MACH**를 제안한다.

1. 롤업 트랜잭션에 대한 빠른 확정(Fast confirmation)
2. 악의적 네트워크 참가자를 처리하기 위한 암호경제적 보안성(Crypto-economic security)
3. ZK 롤업과 옵티미스틱 롤업 모두 지원
4. 다양한 증명 시스템과 런타임에 대응할 수 있는 범용성(Generic compatibility)

최종성을 보장하기 위해, MACH 네트워크는 롤업 상태의 유효성을 검증해야 하며, 이를 통해 롤업 운영자가 올바른 상태 전이 함수(state transition function)를 따랐음을 보장해야 한다. 이를 위해 MACH는 세 가지 상태 유효성 모드(state validity mode)를 지원한다.

비관적 모드(Pessimistic Mode)

Pessimistic Mode



Simplest but not the most efficient as MACH nodes act as full nodes.

Future: Support stateless clients.

비관적 모드에서는 모든 트랜잭션이 기본적으로 유효하지 않은 것으로 간주되며, 따라서 재실행(replay)이 필요하다. 그 결과, 롤업 운영자는 트랜잭션 데이터를 직접 MACH 네트워크로 전송하고, MACH 네트워크는 해당 트랜잭션을 재실행하여 롤업 운영자가 제안한 상태의 유효성에 대한 합의를 형성한다.

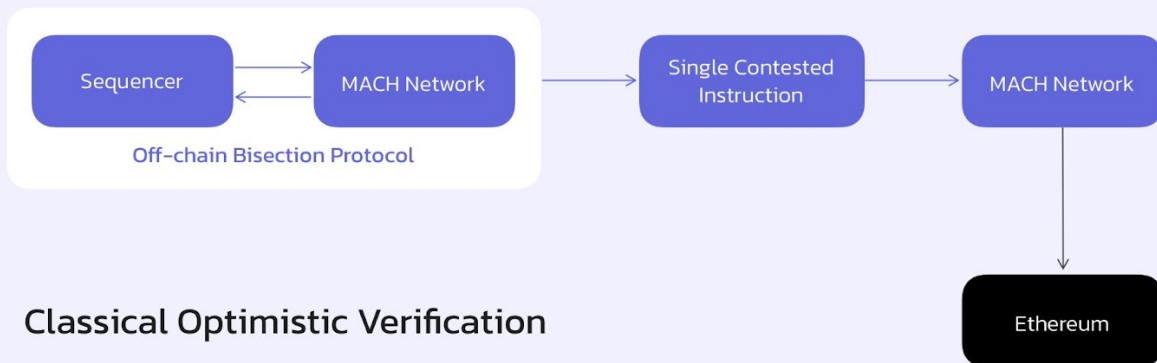
이 운영 모드는 구조적으로 가장 단순하지만, 큰 단점 중 하나는 효율성이 떨어진다는 점이다. MACH가 사실상 롤업의 풀노드(full node) 네트워크처럼 작동하기 때문이다. 이로 인해 높은 노드 사양과 자원 요구가 발생한다.

향후 개선 방향으로, 롤업 노드 운영 시 필요한 상태 데이터를 최소화하여 더 작은 상태 풋프린트(state footprint)로 동작할 수 있는 **비상태(stateless) 클라이언트**를 구축하는 것이 있다.

옵티미스틱 모드(Optimistic Mode)

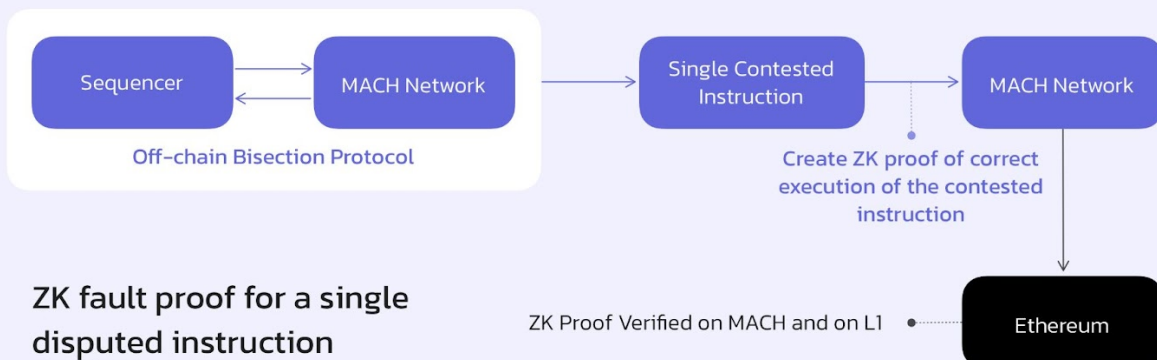
이 모드에서 롤업 운영자는 특정 트랜잭션 블록의 실행이 특정 상태 커밋(state commitment)에 도달했다고 MACH에 주장(assert)한다. MACH 네트워크 내의 모든 노드는 이 주장을 검증할 수 있으며, 만약 새로운 상태가 유효하지 않다고 판단되면 롤업 운영자와 **분할(bisection) 프로토콜**을 통해 도전(challenge)할 수 있다. 이는 아래 그림에서 보듯이 전형적인 옵티미스틱 모드의 구조이다.

Optimistic Mode (I)



분할 프로토콜은 도전자(challenger)가 상태 커밋이 잘못되었다고 판단할 때만 실행된다. 또는, 온디맨드 ZK 증명(on-demand ZK proofs)을 통해 분할 프로토콜을 대체할 수도 있다. 이 경우, 도전이 발생했을 때만 ZK 증명이 생성된다. 아래 도식은 이 과정을 설명한다.

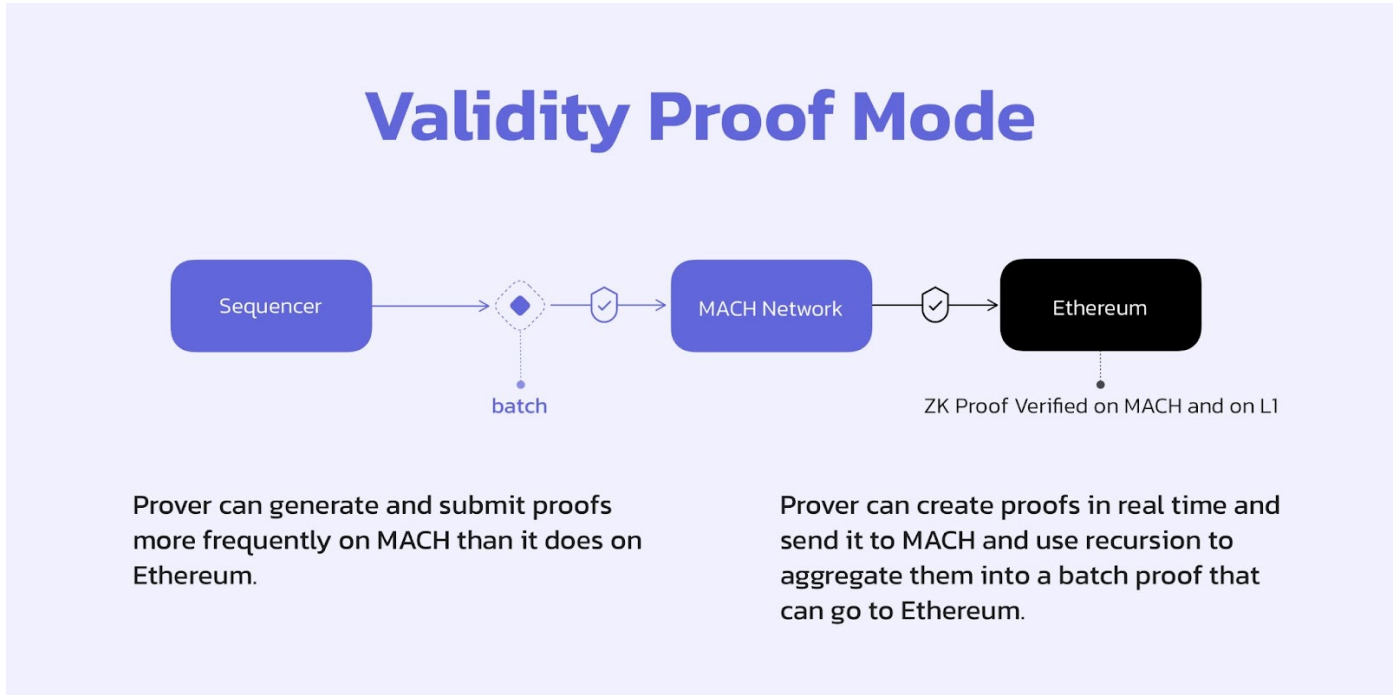
Optimistic Mode (II) With On-Demand ZK Fault Proofs



이 구성은 MACH 네트워크 내에 최소 한 개 이상의 정직한 노드(honest node)가 존재함을 전제로 하며, 대부분의 네트워크 노드는 관찰(observation) 모드로 동작한다.

유효성 증명 모드(Validity Proof Mode)

이 모드에서 MACH 네트워크는 유효성 증명(Validity proof)을 검증하는 탈중앙화된 검증자 네트워크로 작동한다. 시퀀서(sequencer)와 같은 롤업 운영자는 새로운 트랜잭션 집합, 그로부터 생성된 상태, 그리고 해당 유효성 증명을 MACH에 커밋한다. MACH 네트워크는 이 증명의 유효성을 검증하고, 그 결과에 대해 합의를 형성한다.



명시적으로 유효성 증명을 사용하는 이 모드는 옵티미스틱 롤업에서도 효과적으로 작동할 수 있다. 옵티미스틱 롤업의 경우, MACH 외부에서 적절한 인센티브를 갖춘 지정된 증명자(Prover)가 유효성 증명을 생성해 MACH 네트워크에 제출할 수 있으며, MACH는 이를 검증하고 합의를 형성한다.

ZK 롤업의 경우, 증명자는 Ethereum보다 더 자주 MACH에서 증명을 생성하고 제출할 수 있으며, 이는 신속한 최종성을 확보하기 위해 중요하다. 또한 이는 추가적인 증명 작업을 요구하지 않는다. 하나의 대규모 배치 증명을 생성하기 위해 기다리는 대신, 증명자는 실시간으로 증명을 생성해 MACH에 전송하고, 이후 재귀(recursion)를 통해 이를 하나의 배치 증명으로 집계해 Ethereum에 제출할 수 있다. 이러한 점진적 증명(incremental proof)이 즉시 MACH로 배포되기만 하면, 트랜잭션은 신속한 최종성을 경험할 수 있다.

MACH를 통한 상호운용성 (Interoperability via MACH)

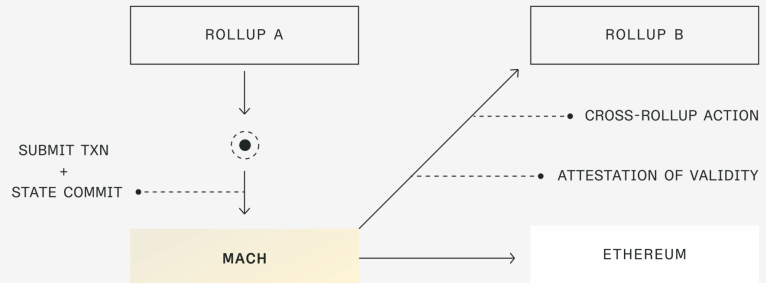
만약 한 RaaS 제공업체가 여러 개의 롤업을 실행(instantiated)하고 있다면, 이들 롤업 간 상호운용성을 확보하기 위해 일반적으로 데이터 가용성(DA) 레이어에 각 롤업별로 전용 브릿지(enshrined bridge)를 구축하게 된다. 또 다른 방법으로는 동일한 DA 레이어를 사용하는 두 롤업 간에 각각 브릿지를 구성하는 방식이 있다.

AltLayer는 이러한 두 가지 접근 대신, **MACH를 활용한 세 번째 설계 방식**을 채택하여 모든 롤업을 조율하는 **브릿징 허브(bridging hub)**로 MACH를 사용한다.

MACH

On top of fast finality for rollups, MACH enables bridges to perform fast interoperability without waiting for the slow finality of L1.

Faster Interop



AltLayer

탈중앙화 시퀀싱을 위한 SQUAD (SQUAD for Decentralised Sequencing)

왜 탈중앙화 시퀀싱인가? (Why Decentralized Sequencing?)

모든 롤업의 핵심 구성요소는 바로 **시퀀서(sequencer)**이다. 시퀀서는 롤업 네트워크 내에서 다음과 같은 역할을 수행하는 노드들이다.

1. **트랜잭션 집계(Aggregate transactions):** 사용자로부터 메모리풀(mempool)을 통해 트랜잭션을 수신한다.
2. **트랜잭션 정렬(Order transactions):** 메모리풀에서 트랜잭션을 선택하고, 특정 정책에 따라 순서를 결정한다.
3. **트랜잭션 실행(Execute transactions):** 대부분의 롤업에서는 시퀀서가 기반 가상머신(VM)에 따라 트랜잭션을 실행하는 역할도 맡는다.
4. **사전 확정 생성(Produce pre-confirmation):** 롤업 트랜잭션의 최종성은 궁극적으로 기저 레이어 1에서 확정되기 때문에, 시퀀서는 사용자 트랜잭션에 대해 **사전 확정(pre-confirmation)**을 생성하여 클라이언트나 애플리케이션이 레이어 1의 최종성을 기다리는 동안 진행할 수 있도록 한다.
5. **정렬된 트랜잭션 데이터를 DA 레이어로 전송(Send ordered transaction data to DA Layer):** 마지막 단계에서 시퀀서는 정렬된 트랜잭션 데이터를 DA 레이어(예: 기저 레이어 1)에 커밋한다. 이 데이터가 레이어 1에 게시되면, 트랜잭션은 레이어 1 수준의 최종성을 갖는다.

따라서 시퀀서는 롤업의 심장과도 같다. 그런데 만약 롤업이 단일 시퀀서만을 가지고 있고, 해당 시퀀서가 비가용 상태가 된다면 어떻게 될까? 이 경우 네트워크는 사실상 트랜잭션 처리를 중단하게 된다. 이는 단순한 가정이 아니라 실제로 시퀀서의 다운타임으로 인해 트랜잭션 처리가 중단된 사례가 존재한다.

물론 “시퀀서가 다운되더라도 사용자는 레이어 2 트랜잭션을 직접 레이어 1로 보낼 수 있다”는 반론이 있을 수 있다. 그러나 이러한 트랜잭션은 최종 확정까지 최대 24시간이 걸릴 수 있으며, 레이어 1과 동일한 수수료를 지불해야 하므로 사용자 경험이 매우 저하된다. 또한 24시간 지연은 부채 포지션 청산과 같은 시간 민감형 트랜잭션에는 치명적이다.

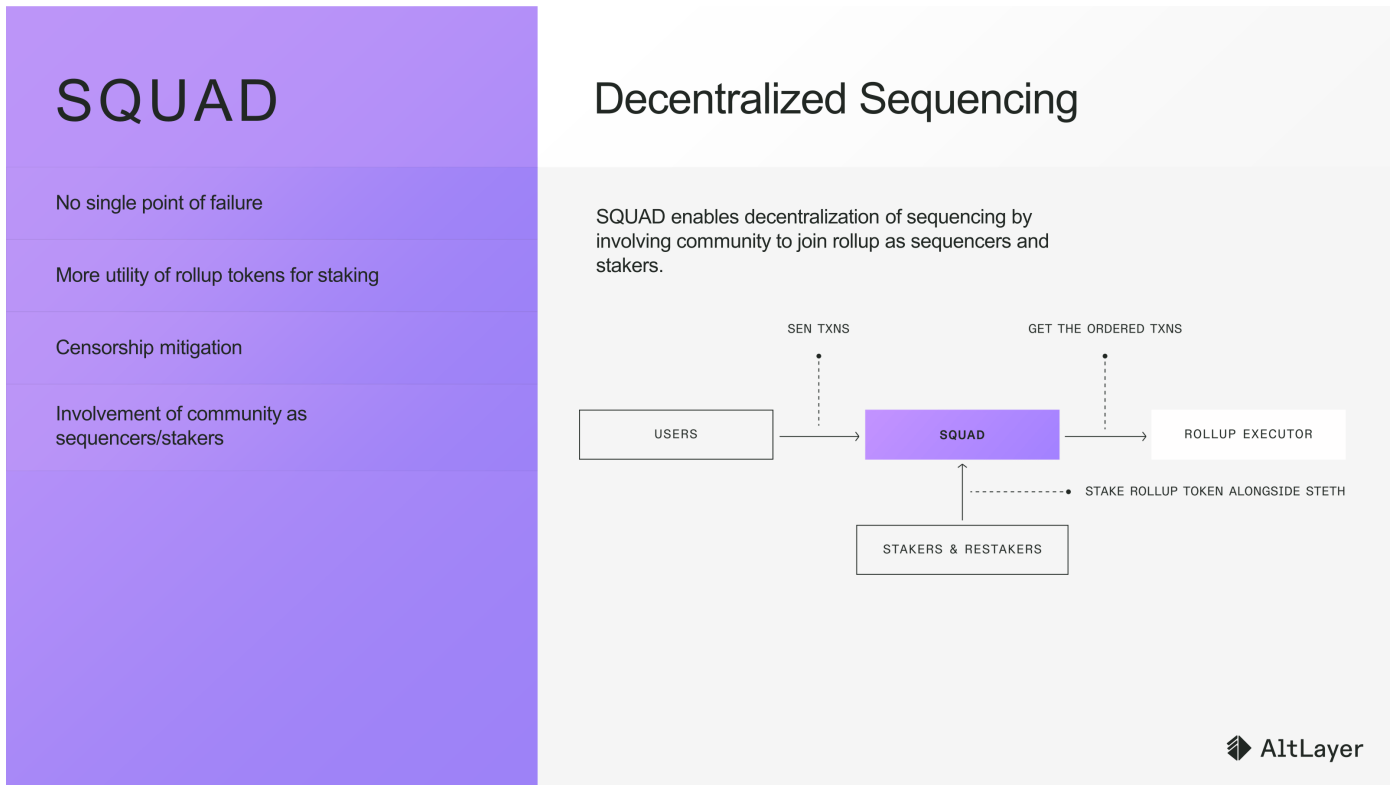
일반적으로 중앙화된 시퀀서는 다음과 같은 문제를 초래할 수 있다.

- 사용자 트랜잭션을 **검열(censorship)**할 수 있음
- **병목 지점(choke point)**이 되거나, 최악의 경우 완전히 비가용 상태가 될 수 있음
- **과도한 수익 추출(excessive rent)** 가능성
- 사용자 트랜잭션을 **프론트러닝(frontrun)**하거나, 부정적 MEV를 만들어 사용자에게 불리하게 작용할 수 있음

따라서 **시퀀서의 탈중앙화는 필수적이다**. 그러나 오늘날 운영 중인 어떤 롤업도 아직 완전한 탈중앙화 시퀀싱을 구현하지 못했다.

SQUAD를 통한 탈중앙화 시퀀싱 (Decentralized Sequencing via SQUAD)

AltLayer는 애플리케이션 개발자가 롤업 런치패드(launchpad)를 통해 원하는 롤업을 요청할 수 있도록 설계되어 있다. 이때 개발자는 자신의 롤업에 필요한 시퀀서의 개수를 지정할 수 있으며, 직접 시퀀서를 찾거나 부트스트래핑할 필요 없이, 또한 롤업의 어떤 구성요소도 직접 호스팅할 필요 없이 **간단하고 매끄럽게 탈중앙화된 시퀀서를 가진 롤업을 구축할 수 있다**.



이것은 **SQUAD**를 활용함으로써 실현된다. SQUAD는 EigenLayer AVS 운영자가 실행하는 노드가, 우리의 RaaS 노코드 대시보드에 의해 인스턴스화된 롤업의 시퀀서로 참여하고자 하는 의사를 등록할 수 있도록 한다. 최종 사용자가 우리의 RaaS 플랫폼을 통해 롤업을 요청할 때, 사용자는 롤업을 운영하기 위해 필요한 최소 및 최대 시퀀서 수, 각 시퀀서가 제공해야 하는 최소 경제적 담보 금액, 그리고 담보로 사용할 수 있는 토큰 목록을 지정한다. 담보는 EigenLayer를 통해 리스테이킹되어야 하며, 이후 부정행위를 한 시퀀서를 슬래싱하는 데 사용된다.

필요한 수의 시퀀서가 최소 담보를 예치하면, 이 시퀀서들은 롤업의 트랜잭션 시퀀싱을 시작할 수 있다.

시퀀서 탐색을 용이하게 하기 위해, SQUAD는 모든 EigenLayer AVS 운영자가 참여할 수 있는 노드 네트워크로 설계되어 있으며, 사실상 WASM과 EVM 런타임을 모두 갖춘 완전한 블록체인 자체이다. SQUAD 내의 노드들은 일반적인 지분증명(PoS) 네트워크와 마찬가지로 **검증자(validators)**라 불린다. 검증자가 SQUAD 네트워크의 일부가 되면, 롤업의 시퀀싱 요구 사항을 등록하고, 이를 시퀀서와 매칭하기 시작할 수 있다.

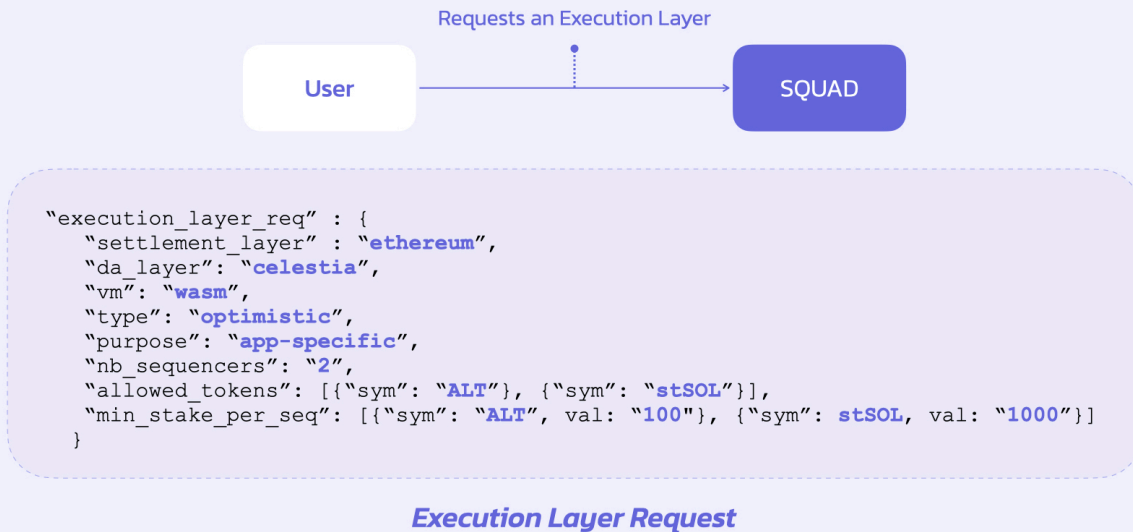
SQUAD 내 시퀀서 등록 (Registering as a Sequencer with the SQUAD)

시퀀서는 최소 요구량의 LST(Liquid Staking Token) 지분을 직접 예치(stake)하거나, LST 보유자가 시퀀서와 함께 스테이킹할 수 있도록 **위임 스테이킹(delegated staking)** 메커니즘을 구현할 수 있다. SQUAD에서의 스테이킹은 AVS(Actively Validated Service) 계약 내에서 네이티브 방식으로 구현되어 있다.

SQUAD에 롤업 요청 등록 (Registering a Rollup Request with the SQUAD)

개발자가 AltLayer의 롤업 런치패드를 통해 특정 유형(예: 플래시 레이어(flash layer) 또는 지속형 레이어(persistent layer))의 롤업을 배포하려는 경우, 먼저 ExecutionLayerRegistryContract라는 스마트컨트랙트에 호출 형태로 자신의 **실행 레이어(execution layer)** 요청을 SQUAD에 등록해야 한다.

Requesting an Execution Layer



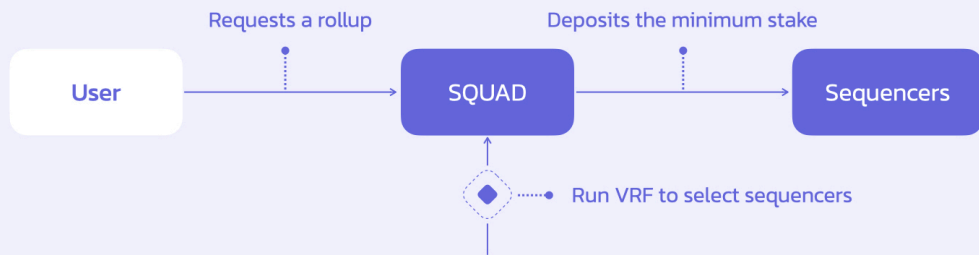
등록 요청은 다음과 같은 **JSON 형식**을 따른다. 개발자는 실행 레이어의 정산 레이어(settlement layer)와 데이터 가용성 레이어(DA layer), 런타임(runtime, 예: WASM, EVM 등), 실행 유형(옵티미스틱 실행 또는 ZK 실행), 실행 목적(범용 또는 앱 특화) 등을 명시한다. 또한 요청에는 해당 실행 레이어에 필요한 시퀀서 수 또한 포함된다.

```
"execution_layer_req" : {  
  "settlement_layer" : "ethereum",  
  "da_layer": "celestia",  
  "vm": "wasm",  
  "type": "optimistic",  
  "purpose": "app-specific",  
  "nb_sequencers": "3",  
  "allowed_tokens": [{"sym": "ALT"}, {"sym": "stSOL"}],  
  "min_stake_per_seq": [{"sym": "ALT", val: "100"}, {"sym": "stSOL", val:  
    "1000"}]  
}
```

롤업을 위한 시퀀서 선정 (Selecting Sequencers for a Rollup)

SQUAD는 롤업 요청을 수신하면, **스테이크(stake)** 규모와 **VRF(Verifiable Random Function)** 기반 **난수(randomness)**를 조합하여 최소 LST 수량을 예치한 검증자(validators) 중에서 시퀀서를 선택한다. 선정된 노드들은 해당 롤업의 시퀀서로서 역할을 수행할 수 있다.

Assigning Sequencers to Rollups



```
"execution_layers" : [  
  "id" : "1",  
  "sequencers": [  
    {"id" : "0x1b", "stake" : [{"sym": "ALT", val: "100"}, {"sym": "stETH", val: "1000"}]},  
    {"id" : "0x2b", "stake" : [{"sym": "ALT", val: "150"}, {"sym": "stETH", val: "1500"}]},  
    {"id" : "0x2b", "stake" : [{"sym": "ALT", val: "200"}, {"sym": "stETH", val: "2000"}]},  
  ]  
]
```

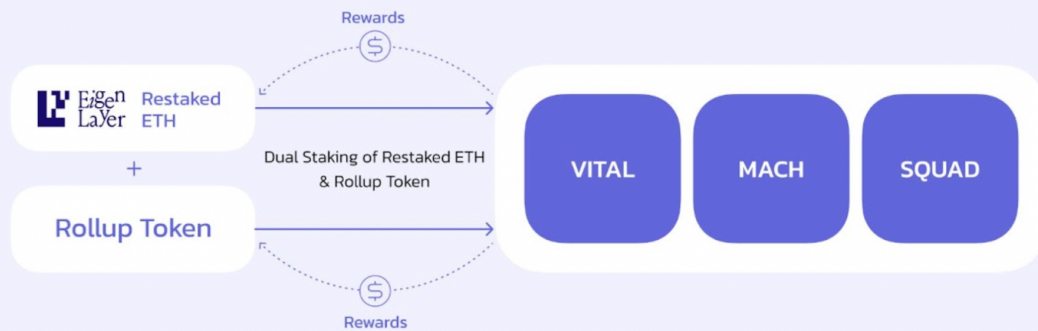
Execution Layer Registry

롤업이 인스턴스화(instantiated)되면, **체인 ID(chainID)**, **RPC 엔드포인트(endpoint)** 등 네트워크 관련 정보를 획득할 수 있으며, 이 시점에서 네트워크는 완전히 운영 가능한 상태가 된다. 이후 시퀀서들은 트랜잭션 처리 작업을 시작해야 한다.

dApp 토큰 스테이킹 (Staking of dApp Token)

각 AVS는 리스테이킹(restaking)을 활용해 부트스트래핑(bootstrap)되며, 동시에 **이중 스테이킹 (dual staking)** 메커니즘도 지원한다. 이중 스테이킹 구조에서는 AVS 운영자가 stETH와 같은 LST 토큰과 함께 **dApp 토큰**을 일정량 예치해야 한다. 이는 dApp 토큰에 롤업 서비스 차원의 실질적 유틸리티를 부여하며, dApp 팀이 AVS 보안 강화에 기여하는 운영자에게 dApp 토큰 스테이킹을 통해 인센티브를 제공할 수 있게 한다.

Restaked Rollups



또한 AVS 운영자가 프로토콜 규칙에서 이탈(deviate)할 경우, 해당 스테이킹 자산은 슬래싱(slash)될 수 있다.

소개 (Introduction)

Wizard는 EigenLayer 생태계 내 빌더(builder)를 위해 빠른 AVS(Actively Validated Service) 배포 및 관리를 제공하는 세계 최초의 **AVS-서비스형 플랫폼(AVS-as-a-Service)**이다. Wizard는 빌더가 복잡한 과정 없이, 저비용으로 AVS를 손쉽게 활용할 수 있도록 하며, 배포 및 관리에 필요한 시간을 수 개월에서 단 몇 분으로 단축시킨다.

- Multiple service manager templates
- Define quorums and strategies
- Seamless deployment of AVS base contracts and service manager contract
- Bundled with AltLayer hosted generic aggregator

Create AVS

- AVS contracts addresses
- AVS roles
- AVS metadata
- Rewards management
- Stake update
- Upgrade AVS service manager
- Quorums and strategies management

Manage AVS

- Operator status
- Operator whitelisting
- Operator ejection
- Activate AltLayer hosted operator
- AltLayer Hosted operator endpoint information

Operator management



Wizard

Wizard의 탄생 배경 (How Wizard was born)

AltLayer는 다수의 AVS를 실제 환경(production)에 배포하고, 50명 이상의 운영자(operator)를 온보딩하며, 그들의 운영을 관리한 풍부한 경험을 보유하고 있다. 이 과정에서 우리는 새로운 AVS 빌더가 **배포 시점뿐 아니라 이후의 운영 및 관리 단계에서 직면하는 다양한 어려움**을 명확히 인식하게 되었다.

AltLayer는 이러한 병목현상을 파악할 수 있는 전문성과 거시적 관점을 모두 보유하고 있었으며, AVS의 배포 및 관리 과정을 **10배 이상 간소화하고, 빌더의 시장 진입(go-to-market) 속도를 획기적으로 단축할 수 있는 플랫폼**을 구축할 수 있는 최적의 위치에 있었다.

Wizard는 **노코드(no-code) AVS 서비스 플랫폼**으로, 수동 배포 대비 **10배 빠른 배포 속도**를 제공하며, AVS와 리스테이킹(restaking)의 대중화를 초고속으로 촉진하는 것을 목표로 한다.

이러한 점을 염두에 두고, 이 제품은 다양한 놀라운 기능들을 제공한다.

- **AVS 배포(Deployment of AVSs):** AVS Wizard는 MACH, Coprocessor, Bridge 등 다양한 템플릿 AVS 계약을 제공하며, 완전한 기능을 갖춘 **커스텀 AVS 계약 편집기(custom AVS contract editor)**를 지원한다. 새로운 AVS 배포 파이프라인을 실험하고자 하는 개발자는 단순히 직관적인 대시보드에서 몇 번의 클릭만으로 손쉽게 AVS를 배포할 수 있다.
- **AVS 관리(Management of AVSs):** AVS 배포는 단발성 단계이지만, 배포된 AVS는 주기적인 관리와 유지보수가 필요하다. 예를 들어, **새로운 리스테이킹 전략(restaking strategy)**을 추가하여 새로운 토큰이 경제적 보안에 활용될 수 있도록 하는 것이다. Wizard는 이를 위해 **커스텀 구현 계약(custom implementation contract)** 배포 기능과, “**Upgrade Service Manager**” 기능을 통한 자체 구현 업그레이드 기능을 제공한다.
- **운영자 화이트리스트 및 온보딩(Operator Whitelisting & Onboarding):** AVS를 출시할 때 가장 큰 난관 중 하나는 **충분한 자산을 위임받은 다양한 운영자 집단을 온보딩**하는 것이다. 이 과정은 일반적으로 많은 시간과 노력을 요구한다. AltLayer AVS Wizard는 **체크박스 방식의 간단한 운영자 선택 UI**를 통해 이러한 복잡성을 제거한다. 개발자는 원하는 운영자를 선택하기만 하면 된다. 또한 AltLayer는 모든 AVS를 위해 **AltLayer 호스팅 운영자(hosted operator)**와 **통합 애그리게이터(agggregator)**를 제공하므로, 외부 운영자 온보딩 없이도 손쉽게 새로운 AVS 아이디어를 실험할 수 있다.

호스팅 운영자를 활성화하면, 해당 운영자는 자동으로 사용자의 AVS에 등록되어 운영자 집합에 포함된다. 또한 Wizard를 통해 네트워크에서 운영자를 제거(ejection)하는 등의 관리 작업도 수행할 수 있다. 이 운영자는 기본적으로 AltLayer의 범용 애그리게이터(generic aggregator)를 사용하도록 미리 설정되어 있으며, 개발자는 자신의 비즈니스 로직을 설계한 후, AltLayer의 **호스팅 운영자 API**를 호출하여 AVS와 상호작용할 수 있다. 운영자 API 문서는 곧 공개될 예정이다.
- **EigenLayer AVS 대시보드 연동(View your AVS on EigenLayer AVS dashboard):** 사용자가 EigenLayer의 AVS 대시보드 등록 요건에 따라 AVS 메타데이터를 업데이트하면, 해당 AVS는 자동으로 EigenLayer AVS 대시보드에 표시된다. 메타데이터 업데이트는 Wizard 내 AVS 프로필 페이지에서 간단히 수행할 수 있다.

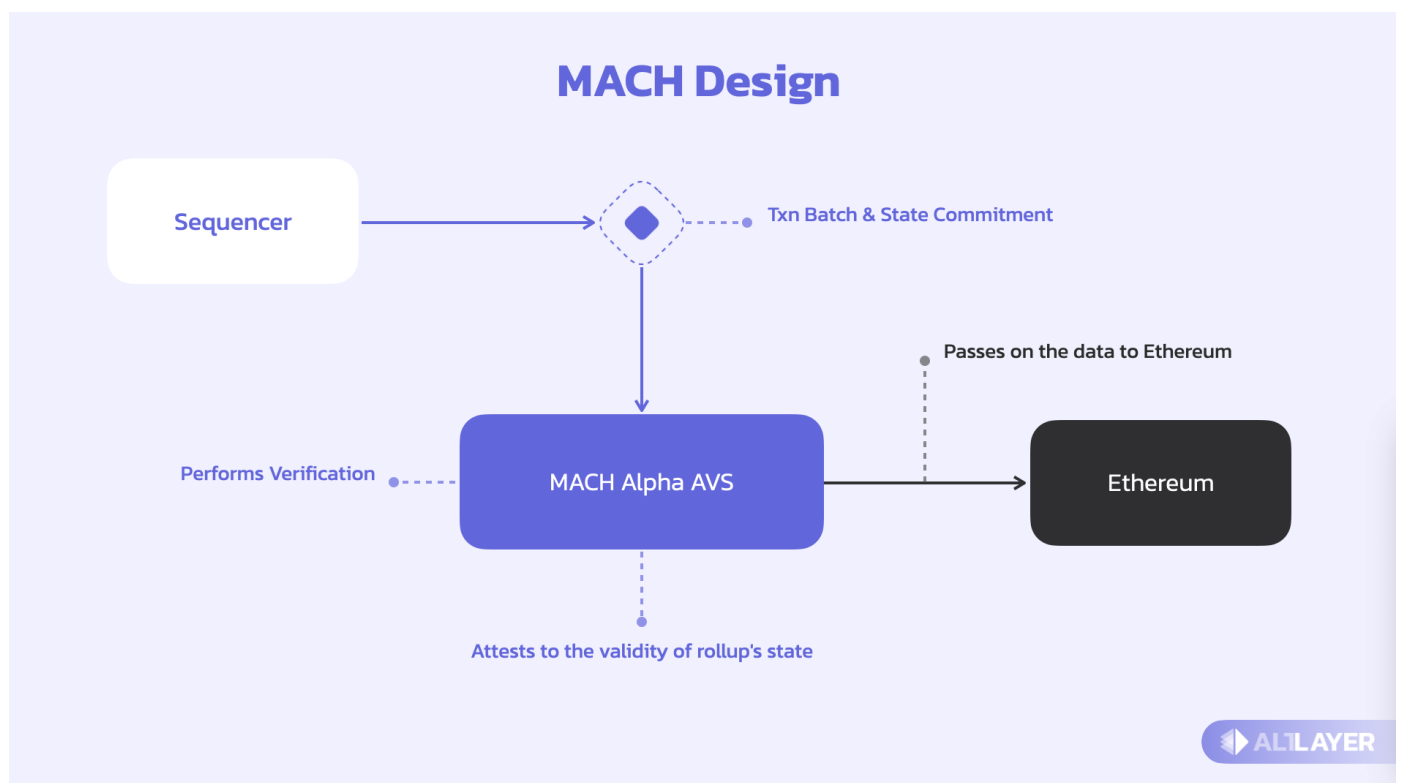
AltLayer가 지원하는 능동 검증 서비스 (Altlayer-facilitated Actively Validated Services)

개요 (Overview)

리스테이킹 롤업(Restaked Rollup) 프레임워크의 일환으로, AltLayer는 여러 롤업을 대상으로 **MACH**라 불리는 신속 최종성(fast finality) 기반의 **능동 검증 서비스(Actively Validated Service, AVS)** 출시에 기여하고 있다. MACH는 신속 최종성 AVS로서, 롤업 사용자에게 다음과 같은 핵심 서비스를 제공한다.

1. 롤업 트랜잭션의 빠른 확정(Fast confirmation)
2. 악의적인 네트워크 참여자 탐지를 위한 암호경제적 보안성(Crypto-economic security)
3. 롤업 상태의 탈중앙화 검증(Decentralized validation of rollup states)

MACH는 롤업 블록을 검증하여, 해당 블록이 롤업의 상태 전이 함수(state transition function)에 부합하는지 확인함으로써 작동한다. 만약 잘못된 블록이 감지되면, MACH 운영자는 **경고(alert)**를 발송하며, 충분한 수의 운영자가 이에 동의할 경우 이 경고는 상위 레이어의 dApp으로 전달된다. MACH 서비스를 구독하는 dApp은 이러한 경고를 수신하면 즉시 대응하여, 잘못된 상태(state)가 승인되지 않도록 차단할 수 있다.



MACH 서비스 이용 방법 (How to use MACH's Services)

MACH가 제공하는 서비스는 최종 확정 여부를 보고하는 **RPC 엔드포인트(RPC endpoint)** 형태로 사용자에게 제공됩니다. 이 RPC 엔드포인트는 어떤 롤업 블록이 최종적으로 확정되었는지를 알려주며, 해당 엔드포인트를 롤업 상의 dApp에 통합하면 보다 빠르고 신뢰성 높은 확정을 구현할 수 있다. dApp은 MACH가 제공하는 **경제적 보증(economic guarantees)**을 활용함으로써, UI나 프론트엔드 상에서 스마트컨트랙트 상태를 안전하게 갱신할 수 있으며, 이를 통해 보다 **안정적이고 우수한 사용자 경험(UX)**을 제공할 수 있다.

또한 MACH는 **거래소(exchanges)**, **브릿지(bridges)**, **경량 탐색기(lightweight explorers)** 및 롤업 상태와 블록에 대한 스트리밍 접근이 필요한 기타 보조 서비스들에게도 유용하다. 이들은 MACH를 통해 더 빠르고 효율적인 서비스를 사용자에게 제공할 수 있다. 예를 들어, 거래소는 OP Mainnet이나 Arbitrum One과 같은 롤업에서 사용자의 자산 인출을 승인하기 위해 더 이상 긴 대기 시간을 요구하지 않아도 된다.

롤업-서비스형(Rollup-as-a-Service)

롤업-서비스형(RaaS)이란 무엇인가? (What is Rollup-as-a-Service?)

롤업-서비스형(Rollup-as-a-Service, RaaS)은 특정 애플리케이션에 맞춘 맞춤형 롤업(application-tailored rollup)을 손쉽게 출시할 수 있도록 하는 제품이다. 이는 사용자가 소프트웨어나 하드웨어의 어느 부분도 직접 유지하거나 호스팅할 필요가 없는 **서비스형 소프트웨어(Software-as-a-Service, SaaS)** 모델과 유사하다. 따라서 복잡한 인프라 관리 부담에서 완전히 자유로워진다.

RaaS 제품을 통해 개발자는 SDK나 대시보드를 활용하여 애플리케이션 특화형 롤업을 구축할 수 있으며, 노드를 직접 찾거나, 코드를 빌드하거나 유지보수할 필요가 없다.

AltLayer의 RaaS 제공 모델 (AltLayer's RaaS Offering)

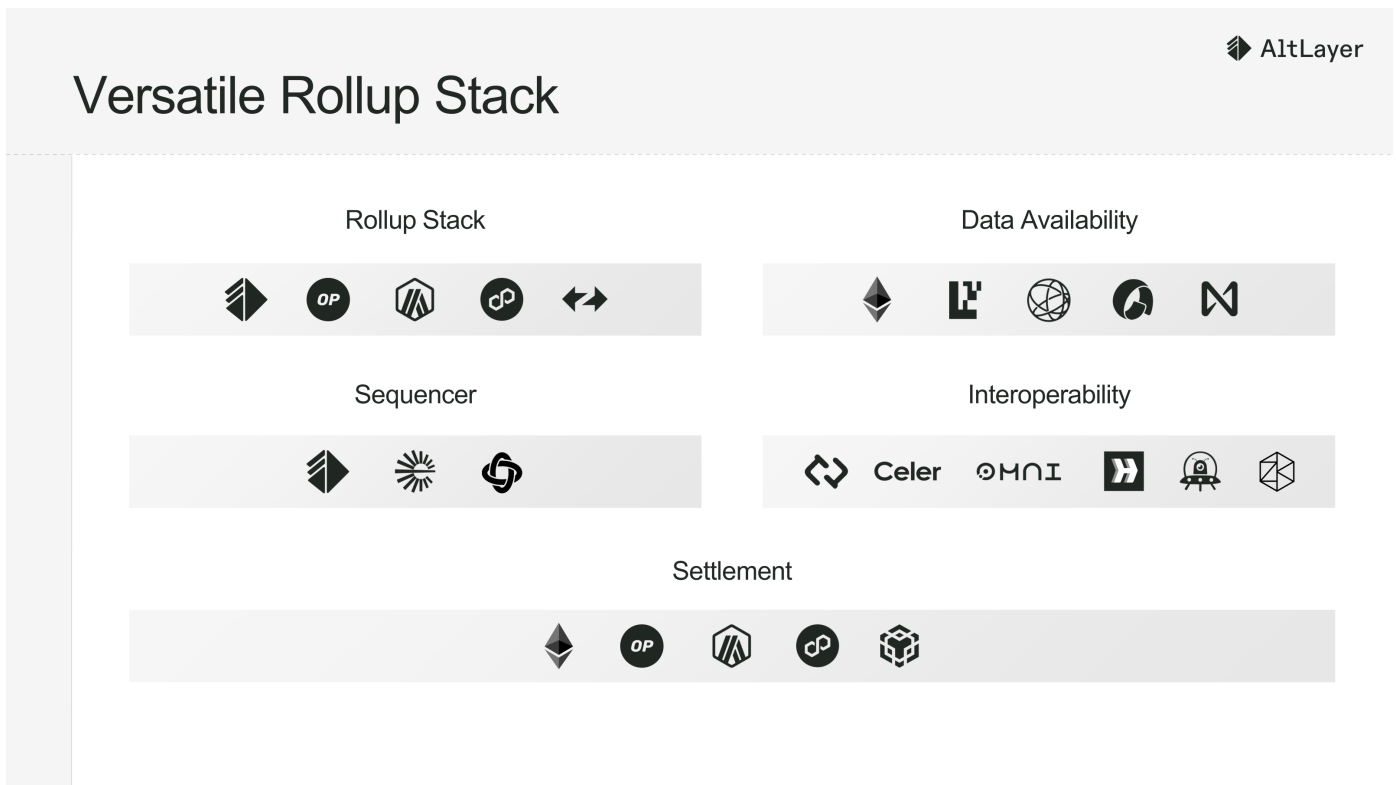
AltLayer의 RaaS 제품은 세 가지 핵심 구성요소로 이루어진다.

1. **리스테이킹 롤업(Restaked Rollups):** AltLayer는 특정 롤업을 위해 온디맨드 방식으로 생성되는 세 개의 수직 통합형 능동 검증 서비스(Actively Validated Services, AVS)를 제공한다. 이들 AVS는 함께 작동하여 애플리케이션 롤업(app rollup)을 위한 세 가지 핵심 서비스를 제공하는데, 즉, **탈중앙화 시퀀싱(decentralized sequencing)**, **롤업 상태의 정확성 검증(verification of rollup's state correctness)**, 그리고 **신속한 최종성(faster finality)**이다. 이러한 서비스들

은 롤업의 탈중앙화를 촉진하고, 보안을 강화하며, 리스테이킹을 통해 Ethereum의 신뢰 네트워크를 활용하면서 롤업 간 상호운용성을 향상시킨다. 리스테이킹 롤업은 다음의 세 가지 모듈형 구성요소로 설계되어 있다.

- **Vital:** 롤업 상태의 탈중앙화 검증을 위한 AVS
- **Mach:** 신속한 최종성을 위한 AVS
- **Squad:** 탈중앙화 시퀀싱을 위한 AVS

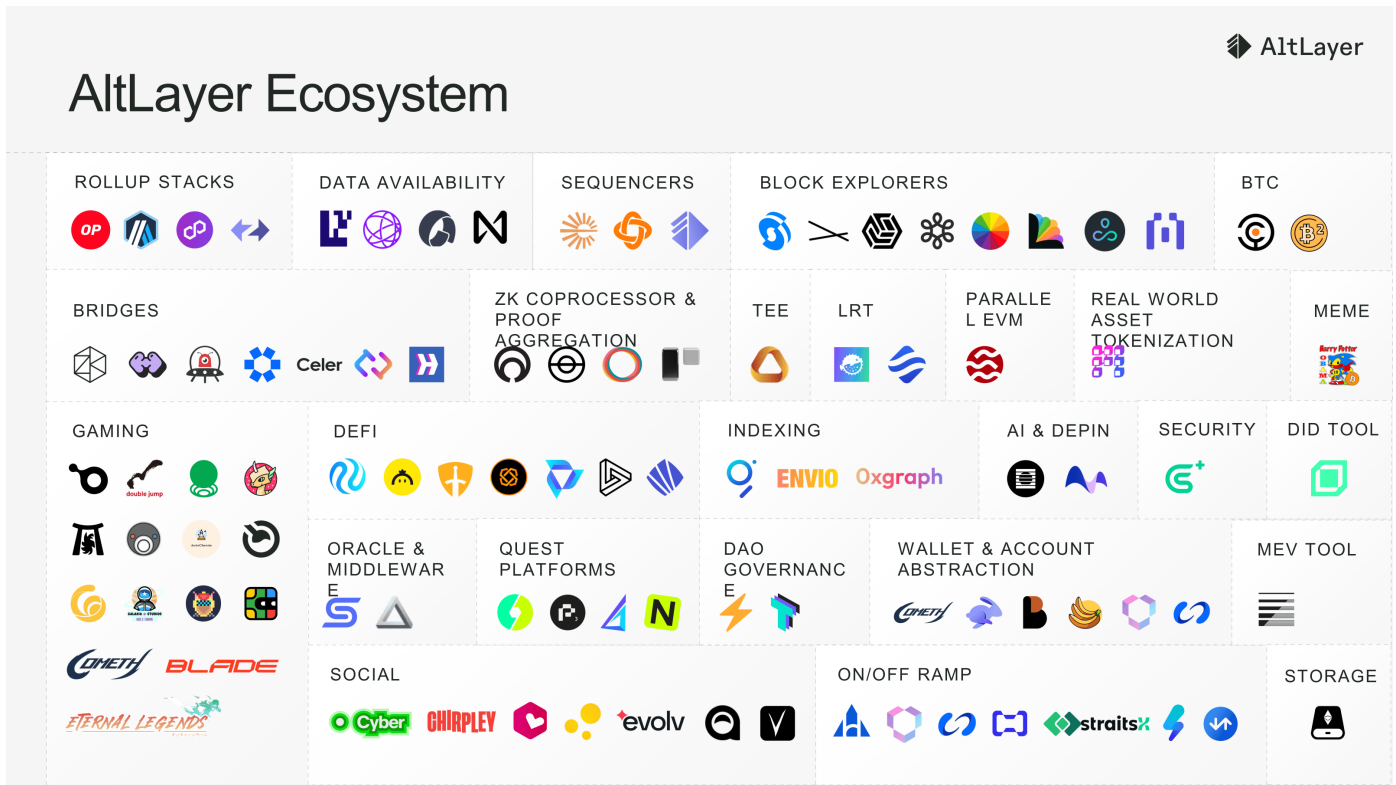
2. **다양한 롤업 스택(Versatile Rollup Stack)** AltLayer는 모든 주요 롤업 스택, 데이터 가용성 (DA) 레이어, 정산(settlement) 레이어, 탈중앙화 시퀀서 세트, 상호운용성 브릿지를 폭넓게 지원한다. 주요 RaaS 파트너로는 **Optimism, Arbitrum, Polygon, zkSync, EigenLayer, Celestia, Hyperlane** 등이 있으며, 이를 통해 개발자가 **모듈형 롤업 구성요소를 손쉽게 교체 (swap in/out)**할 수 있는 유연한 환경을 제공한다.



3. **노코드 대시보드(No-code Dashboard):** AltLayer는 개발자뿐만 아니라 코딩 경험이 거의 없거나 전혀 없는 사용자도 몇 번의 간단한 클릭만으로 5분 이내에 맞춤형 실행 레이어(customized execution layer)를 구축할 수 있는 노코드 대시보드를 제공한다. 이 대시보드가 제공하는 커스터마이징 옵션에는 다음과 같은 항목들이 포함된다. 네트워크 수준의 매개변수(예: 시퀀서 수), 체인 수준의 매개변수(예: 블록 가스 한도, 가스 토큰, 컨트랙트 크기), 그리고 브릿지나 RPC 제공자와 같은 미들웨어 솔루션 등이 있다.

AltLayer 생태계 (AltLayer Ecosystem)

AltLayer의 생태계는 다양한 통합 및 서비스 확장을 통해 빠르게 성장하고 있으며, 이를 통해 고객이 빠르게 개발을 시작할 수 있도록 지원한다.



가장 최신의 생태계 맵은 [AltLayer 공식 웹사이트](#)에서 확인할 수 있다.

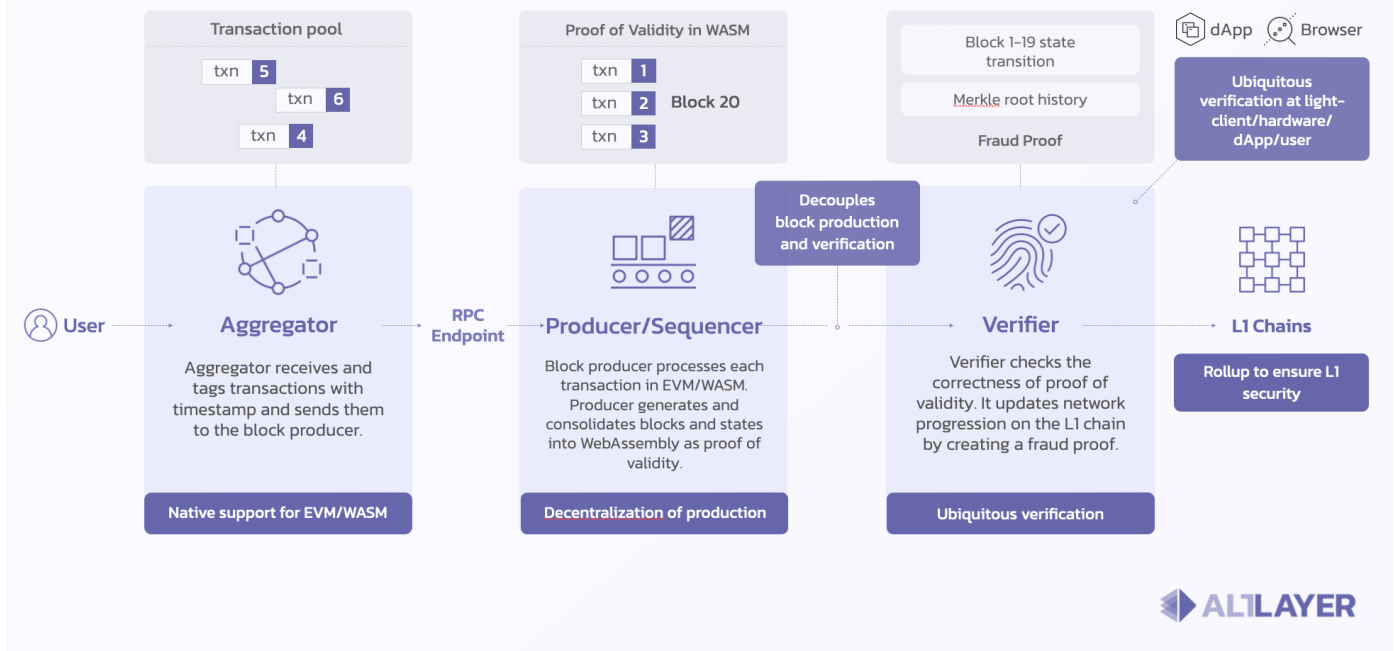
AltLayer의 인하우스 롤업 스택 심층 분석 (AltLayer's In-house Rollup Stack in Depth)

AltLayer의 인하우스 롤업 스택은 어떻게 작동하는가? (How does AltLayer's in-house rollup stack work?)

AltLayer는 다양한 애플리케이션 요구사항에 맞춰 구성 가능한 **모듈형 확장 프레임워크(modular scaling framework)**로 설계되었다. 이 시스템은 **EVM**과 **WASM** 모두와 호환되며, Ethereum 또는 EVM/WASM 호환 체인에서 비롯된 모든 스마트컨트랙트를 지원한다.

AltLayer는 트랜잭션 실행의 생애주기를 여러 단계로 분리하여 관리한다. 즉, **트랜잭션 집계(transaction aggregation)** → **블록 생성(block production)** → **블록 검증(block verification)**의 단계로 나뉜다.

AltLayer Architecture



집계자(Aggregators)

집계자는 브라우저 확장 프로그램, 모바일 지갑, dApp 백엔드 등 다양한 채널로부터 트랜잭션을 수신한다. 이들은 단일 장애 지점(SPoF, Single Point of Failure)을 피하기 위해 **탈중앙화된 방식**으로 구성되며, 트랜잭션을 집계한 후 **하나의 블록 생성자(Block Producer)**가 과부하되지 않도록 여러 블록 생성자에게 **배치(batch)** 형태로 전송한다. 이는 일종의 **로드밸런싱(load balancing)** 역할을 한다.

dApp의 요구사항에 따라 집계자의 구성을 사용자 정의(customize)할 수도 있다. 예를 들어, 특정 dApp 개발자가 로드밸런싱을 원하지 않는다면, 블록 생성자가 사용자의 트랜잭션을 직접 수락하도록 시스템 구성을 단순화할 수 있다. 이 경우 집계자는 선택적(optional) 구성 요소가 된다.

또한 집계자는 수신한 트랜잭션에 **타임스탬프(timestamp)**를 부여하고, 이를 시퀀싱(sequencing)한 뒤 블록 생성자에게 전달한다. 여러 개의 집계자가 존재하더라도, 이들 간의 **글로벌 시퀀스 동기화(global sequence synchronization)**는 필요하지 않다. 각 집계자가 **신뢰할 수 있는 시간 소스(trusted time source)**를 가지고 있다고 가정하기 때문이다. 각 집계자가 생성한 시퀀스 번호(sequence number)는 누락된 트랜잭션을 식별하는 데 활용되며, 전역 합의(global consensus)를 형성할 필요 없이 **개별 노드 수준에서 동기화 및 추적**이 가능하다.

블록 생성자(Block Producers)

블록 생성자의 주요 역할은 집계자로부터 전달받은 **비부인성(non-repudiable)** 정보를 기반으로 트랜잭션의 순서를 결정하고, 트랜잭션을 실행하여 네트워크 상태를 갱신하는 것이다. 새롭게 생성된 블록은 옵티미스틱 롤업(Optimistic Rollup)에서의 **낙관적 실행(optimistic execution)**과 유사하게, 블록 진행의 낙관적 뷰를 나타낸다.

기본적으로 AltLayer는 **단일 블록 생성자(single-producer architecture)** 구조로 작동하여 트랜잭션 지연(latency)을 최소화한다. 하지만 롤업 소유자가 탈중앙화를 강화하기 위해 약간의 지연을 감수할 의향이 있다면, **다수의 블록 생성자**를 추가할 수도 있다. 이때 블록 생성자는 **GRANDPA 합의 프로토콜(consensus protocol)**을 사용하여 합의를 이룬다.

검증자(Verifiers)

검증자의 주요 역할은 블록 생성자가 만든 블록의 **유효성을 검증(verify)**하는 것이다. 또한 검증자는 **기저 레이어 1(Layer 1)**에 블록 진행 상황(block progression)을 주기적으로 업데이트하기 위해 **팩(pack)**이라 불리는 묶음을 생성한다. 팩을 생성하기 전, 검증자는 온체인 검증(on-chain verification) 최적화를 위해 추가 정보를 포함시켜야 한다. 예를 들어, 블록 단위가 아닌 트랜잭션 단위로 검증 부하를 줄이기 위해서는 각 트랜잭션별 **머클 루트 해시(Merkle root hash)**를 생성해야 한다.

블록 생성자는 검증자가 작업을 수행하는 데 필요한 **최소한의 정보**만 생성하여 블록의 생성 속도와 전체 처리량(throughput)을 개선한다. 반면, 온체인 검증을 위해 추가로 필요한 정보(예: 명령어 수준의 상태 해시)는 검증자가 직접 생성한다.

또한, 검증을 트랜잭션 단위보다 더 세분화하여 **단일 명령어 단위(single instruction level)**로 줄이기 위해서는, 각 명령어별로 루트 해시를 생성해야 한다. 이러한 **중간 상태 해시(intermediate state hashes)**를 포함하면, 팩을 세분화(bisect)하여 보다 세밀한 수준에서 검증할 수 있다. **명령어 상태 머클 루트 해시(instruction state Merkle root hash)**에는 해당 팩 내에서 실행되어야 하는 모든 명령어가 포함된다.

검증자의 책임성(Accountability of Verifiers)

악의적인 검증자(verifier)가 기저 체인에 잘못된 상태 루트 해시를 제출하는 경우를 탐지하고 처벌하기 위해, AltLayer는 **사기 증명(fraud proof)** 및 **분쟁 해결(dispute resolution)** 메커니즘을 내장하고 있다. 분쟁 해결은 **분할 접근법(bisection approach)**을 통해 수행되며, 논란이 되는 정확한 명령어를 찾아내므로 온체인 검증 비용을 최소화할 수 있다.

챌린지 기간(challenge period)이 끝날 때까지 성공적인 이의 제기(dispute)가 없으면, 해당 블록은 최종 확정(finalized)으로 간주된다. 단, 사용자는 챌린지 기간이 종료되기 전이라도 블록을 직접 검증하여 **해당 블록의 최종성에 대한 신뢰를 스스로 확보**할 수 있다.

블록 최종성 (Block Finalization)

블록체인마다 **최종화된 블록(finalized block)**의 정의는 다를 수 있다. 일부 블록체인에서는 최종화가 **결정론적(deterministic)**이며, 진정한 의미의 “최종”을 뜻한다. 즉, 하드포크(hard fork)와 같은 시스템 외부의 개입이 없는 한, 최종화된 블록은 절대로 되돌려지지 않는다. 반면, 다른 블록체인에서는 최종화가 **확률론적(probabilistic)** 개념으로, “최종화된 블록”이라도 더 길거나 무거운 체인이 등장하면 되돌려질 가능성이 아주 작지만 존재한다. 다만, 체인이 길어질수록 이러한 가능성은 점점 더 작아진다.

블록체인은 일반적으로 최종화된 블록과 최종화되지 않은 블록만 구분한다. 반면 AltLayer는 블록 최종화를 위해 계층적 방법론(tiered methodology)을 사용한다. 이 계층적 방법론은 최종 사용자가 자신의 **보안 예산(security budget)**에 따라 트랜잭션의 최종성 상태를 결정할 수 있도록 한다. 최종화는 세 가지 서로 다른 계층으로 구성된다: 실행 레벨(Execution-level, 낮은 보안 예산용), 검증 레벨(Verification-level, 중간 보안 예산용), 그리고 롤업 레벨(Rollup-level, 높은 보안 예산용)이다.

블록이 이러한 최종화 단계를 거치면서 최종성에 대한 신뢰 수준(confidence level)은 점차 높아진다. 실행 레벨에서의 신뢰 수준이 가장 낮고, 롤업 레벨에서 가장 높다.

실행 레벨 최종성 (Execution-level Finalization) 앞서 설명한 바와 같이, 트랜잭션은 트랜잭션 풀(transaction pool)에서 수집되어 블록 생성자(block producer)에 의해 블록으로 묶인다. 이후 블록 생성자들이 참여하는 합의 프로토콜(consensus protocol)에 의해 블록이 승인된다. 이 합의 프로토콜을 통해 승인된 블록과 그 내부의 트랜잭션들은 실행 레벨에서 최종화된(finalized at the execution-level) 것으로 간주된다.

검증 레벨 최종성 (Verification-level Finalization) 블록이 블록 생성자에 의해 생성되고 합의 프로토콜을 거친 후, 검증자(verifier)들에 의해 검증 과정을 수행한다. 이 검증 과정을 통과한 블록은 검증 레벨에서 최종화된(finalized at the verification-level) 것으로 간주된다.

보다 강력한 최종성을 확보하기 위해, 추가 검증자들이 기존 검증자가 커밋한 업데이트를 승인(affirm) 또는 거부(reject)할 수도 있다. 블록이 더 많은 검증자의 승인을 받을수록, 그 블록이 올바른 가능성은 높아진다. 또한, 검증자 집합의 정족수(quorum)를 기반으로 한 합의(consensus)는 AltLayer에서 선택적(optional) 기능으로 제공된다.

롤업 레벨 최종성 (Rollup-level Finalization) 검증자 정족수 합의를 통과한 블록은 일정 기간 동안 챌린지(Challenge)가 가능한 상태로 유지된다. 이 기간 동안 누구나 자신의 검증자를 운영하며 검증 과정에 참여할 수 있다. 만약 체인 상의 제안(proposal)과 검증자(challenger)의 결과 간에 불일치(discrepancy)가 발견되면, 검증자는 해당 블록에 대해 챌린지를 제기할 수 있다.

챌린지가 성공하면, 그 시점 이후의 모든 블록은 무효(invalid)로 간주되며, AltLayer는 마지막으로 유효한 블록까지 상태를 되돌림(state revert) 처리해야 한다. 반대로 챌린지가 실패하면, 챌린저가 제공한 스테이크(stake)는 슬래싱(slashed)된다.

탈중앙화 시퀀서 세트(Decentralized Sequencer Set)

AltLayer는 비컨 레이어(Beacon Layer)라 불리는 전용 네트워크를 전역 시퀀서 세트(global sequencer set)로 운영한다. AltLayer가 롤업을 프로비저닝(provision)할 때, 비컨 레이어의 노드들은 그들의 스테이크(stake)를 기준으로 선발된다.

사용자는 대시보드 또는 SDK를 통해 롤업 요청을 제출하며, VM 유형, 시퀀서 수, 최소 스테이크, 허용 토큰 목록과 같은 요구사항을 명시한다.

Requesting an Execution Layer



```
"execution_layer_req" : {  
  "settlement_layer" : "ethereum",  
  "da_layer": "celestia",  
  "vm": "wasm",  
  "type": "optimistic",  
  "purpose": "app-specific",  
  "nb_sequencers": "3",  
  "allowed_tokens": [{ "sym": "ALT", val: "100" }, { "sym": "stSOL", val: "1000" }],  
  "min_stake_per_seq": [{ "sym": "ALT", val: "100" }, { "sym": "stSOL", val: "1000" }]  
}
```

Execution Layer Request



이후 시퀀서 레이어(sequencer layer)가 해당 롤업에 배정할 노드들을 할당하며, 모든 시퀀서 할당이 완료되면 롤업은 운영 가능(operational) 상태가 된다.

Sequencers Assigned for Rollups



```
"execution_layers" : [  
  {  
    "id" : "1",  
    "sequencers": [{"id" : "0x1b", "stake" : [{"sym": "ALT", val: "100"}, {"sym": "stEth", val: "1000"}]}]  
}
```

Execution Layer Registry



공용의 탈중앙화 시퀀서 세트는 크로스체인 원자적 메시징(cross-chain atomic messaging)을 보장하는 데에도 활용될 수 있다.

사기 증명에 의한 보안(Security via Fraud Proof)

AltLayer가 프로비저닝한 롤업은 옵티미스틱(optimistic) 성격을 가지므로, 기저 체인(underlying chain)은 사기 증명(fraud proof)을 통해 보안을 제공한다. AltLayer는 논쟁 지점을 단일 VM 명령어(instruction)까지 좁히기 위해 분할(bisection) 프로토콜을 사용한다. 쟁점이 되는 구체적 명령어가 식별되면, 시스템은 해당 명령어를 해당 상태(state)와 함께 재실행(re-execute)한다.

본 절에서는 분할을 이용한 온체인 분쟁 해결(on-chain dispute resolution)을 제시한다. 분할 자체는, 특정 트랜잭션 집합과 그에 따른 상태 변화가 유효하다고 주장하는 제안자(proposer)와, 그렇지 않다고 주장하는 도전자(challenger) 간의 상호작용형 오프체인 프로토콜(interactive off-chain protocol)이다. 분할 프로토콜의 최종 결과는, 도전자가 잘못 실행되었다고 주장하는 특정 트랜잭션과 그 트랜잭션 내의 명령어를 가리키게 된다. 프로토콜은 다음과 같이 동작한다.

1. 도전자는 최종화 중인 팩(pack)의 하나 이상의 속성(예: 잘못된 상태 전이 해시, 잘못된 트랜잭션 루트 해시)에 대해 분쟁을 제기한다.
2. 분쟁 중인 각 속성에 대해, 배심(jury)은 제안자와 도전자 간의 쟁점 지점을 분할(bisection) 방식으로 통지받는다.
3. 도전자는 제안자가 주장한 임의의 머클 루트 해시(Merkle root hash)에 이의를 제기한다.
4. 제안자는 해당 머클 루트 해시를 생성하는 데 사용된 두 개의 해시(각각 상태 전이의 절반에 대응)를 제시한다.
5. 도전자는 두 절반 중 하나를 선택해 추가로 분쟁하며, 제안자는 선택된 해시를 생성한 두 개의 하위 해시를 제시한다. 이 과정은 분쟁 대상이 머클 트리의 리프 노드(leaf node) 해시로 좁혀질 때까지 반복된다.
6. 이 리프 노드가 바로 제안자와 도전자 사이의 실질적 쟁점(contestation)을 의미한다.

롤업 유형 (Rollup Types)

플래시 레이어 롤업 (Flash Layer Rollups)

블록체인이 직면한 가장 지속적인 문제 중 하나는 바로 확장성(scalability)이다. 우리는 종종 단일 dApp의 갑작스러운 인기로 인해 네트워크 전체가 거의 멈추는 현상을 목격한다. 이러한 확장성 문제의 근본적인 원인은 모든 dApp을 단일 모놀리식(monolithic) 블록체인 위에서 실행하는 구조적 한계에 있다.

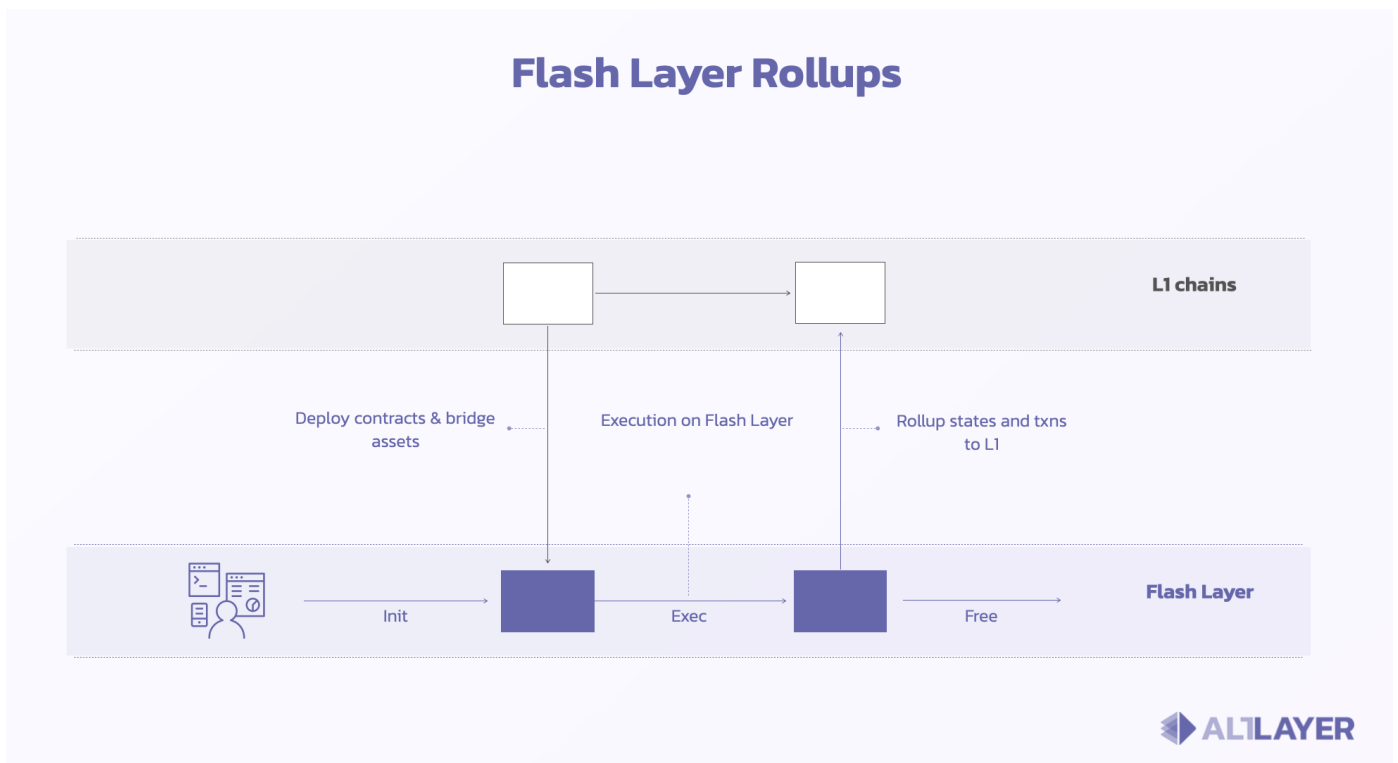
이용량 급증의 일반적인 원인으로는 토큰 세일(token sales), 플레이투언(play-to-earn) 게임 이벤트, 에어드롭(airdrop) 등이 있으며, 이들은 서로 상대적으로 독립적이고 일시적(transient) 성격을 지닌다는 공통점을 갖는다. 이러한 맥락에서 AltLayer는 **플래시 레이어(FlashLayer)**라는 독특한 형태의 롤업을 제공한다.

플래시 레이어(Flash Layer)는 선택적 사기 증명(fraud proof)을 지원하는 일회성(disposable) 애플리케이션 맞춤형 롤업이다.

플래시 레이어를 통해, dApp 개발자는 다음과 같은 방식으로 수요 급증에 대응할 수 있다.

1. 레이어 1에 의해 보안이 보장되는 빠르고 확장성 높은 롤업을 신속하게 가동(spin-up)
2. 필요한 기간 동안 해당 롤업을 운영하여 레이어 1의 과부하를 방지
3. 수요가 줄어들면 “수명 종료(end-of-life)” 정산(settlement)을 통해 롤업을 종료 및 폐기(dispose)

이 방식은 시스템 전체의 자원 효율(resource optimization)을 극대화한다. 실행 레이어와 그 자원은 dApp이 레이어 1에서 감당할 수 없는 수준의 수요를 예상할 때만 활성화되며, 수요가 줄어들면 dApp은 다시 레이어 1로 돌아갈 수 있다.

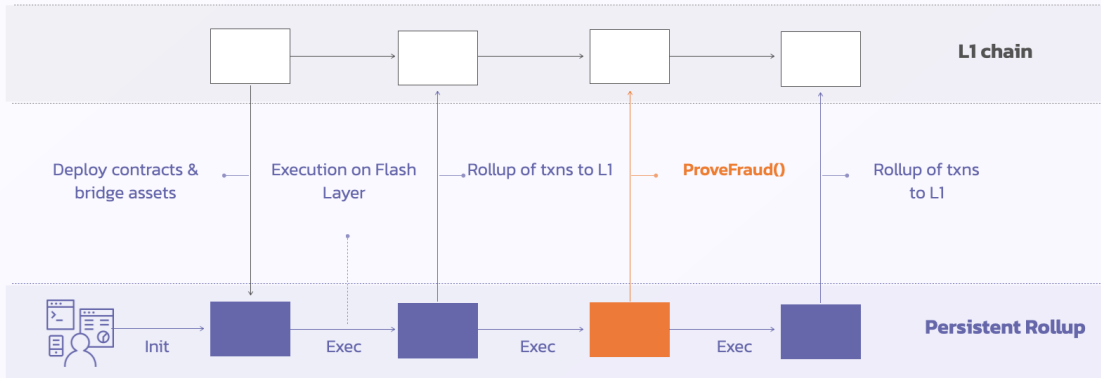


플래시 레이어는 기저 체인(underlying chain)의 보안에 의해 보호되는 일시적 실행 레이어(transient execution layer) 형태로 탄력적인 컴퓨팅 용량을 제공한다. 특정 이벤트를 위해 별도의 체인을 구축하거나 레이어 1에 직접 컨트랙트를 배포하는 대신, 이벤트 관리자는 일시적 컴퓨팅 수요(burst compute capacity)를 요청할 수 있다. 이 요청은 해당 dApp에 맞춤형 플래시 레이어를 생성하며, 적절한 상태(state) 및 자산 브리징(asset bridging) 기능을 포함한다. 지정된 기간이 종료되면 플래시 레이어의 상태와 자산은 기저 레이어 1 체인으로 롤업되어, 플래시 레이어는 폐기되고 관련 자원은 모두 해제된다.

영속형 롤업 (Persistent Rollups)

AltLayer가 제공하는 영속형 롤업(Persistent Rollup)은 표준 옵티미스틱 롤업(Optimistic Rollup)과 유사하지만, 애플리케이션 특화형(application-specific)이며 EVM과 WASM 모두 호환(EVM & WASM compatible)된다는 점에서 차별화된다.

Persistent Rollups



ALTLAYER

영속형 롤업은 다음과 같은 장기 운영형(long-lived) 애플리케이션에 이상적이다.

- GameFi (게임 파이낸스)
- SocialFi (소셜 파이낸스)
- MetaFi (메타버스 파이낸스)
- DeFi (탈중앙화 금융)

AltLayer 인하우스 롤업 스택의 핵심 기능 (Core Features of AltLayer's In-House Rollup Stack)

모듈형(Modular)

AltLayer 스택 전체는 **모듈성(modularity)**을 중심으로 설계되어, 사용자가 자신의 롤업에 필요한 구성 요소를 자유롭게 선택할 수 있도록 한다. 사용자는 다음 항목들을 직접 지정할 수 있다.

- **실행 환경(Run-time):** EVM, WASM 등
- **데이터 가용성 레이어(Data Availability Layer):** 기저 레이어 1, 데이터 가용성 위원회(DA committee), Celestia 등
- **정산 레이어(Settlement Layer):** Ethereum, Arbitrum One 등

이러한 구조를 통해 AltLayer는 다양한 기술 스택을 조합하여 사용자 맞춤형 롤업을 제공할 수 있다.

멀티-VM 지원 (Multi-VM Support)

AltLayer는 **WASM** 및 **WASM**으로 구현된 **EVM(EVM implemented in WASM)**을 기본적으로 지원한다. 기본 실행 환경이 WASM이므로, EVM이 아닌 런타임 또한 지원 가능하다. 따라서 사용자는 Ethereum, Avalanche, Arbitrum 등 다양한 정산 레이어 위에서 작동하는 롤업을 손쉽게 구축할 수 있다.

사기 증명(Fraud Proofs)

AltLayer의 롤업은 현재 사기 증명 기반(fraud-proof based) 구조를 채택하고 있다. 기존의 많은 롤업 구현체들이 사기 증명을 아직 구현하지 않았거나, 구현했더라도 화이트리스트된 노드만이 실행할 수 있도록 제한되어 있는 반면, AltLayer는 이미 완전한 사기 증명 기능을 구현하였다. 즉, 네트워크의 모든 참여자가 잘못된 상태 전이(state transition)가 실행되었다고 판단하면, 레이어 1에 블록 데이터를 커밋한 시퀀서에 대해 직접 이의를 제기(challenge)할 수 있다. 이를 통해 네트워크는 개방적이고 신뢰할 수 있는 검증 구조를 유지한다.

탈중앙화 시퀀서 (Decentralized Sequencers)

AltLayer의 RaaS(Rollup-as-a-Service)를 통해 출시된 롤업은 탈중앙화 시퀀서 네트워크(decentralized sequencer network)에 의해 운영된다. 이 시퀀서들은 비컨 레이어(Beacon Layer)라는 공유 네트워크에서 스테이크(stake) 및 난수(randomness)를 기반으로 선발된다.

비컨 레이어는 누구나 참여할 수 있는 오픈 네트워크이며, 자체적으로 WASM 및 EVM 런타임을 갖춘 완전한 블록체인(full-fledged blockchain)이다. 비컨 레이어의 노드는 검증자(validators)로 불리며, 일반적인 지분증명(PoS) 네트워크와 유사한 구조를 갖는다. 검증자가 비컨 레이어의 일원이 되면, AltLayer의 RaaS를 통해 인스턴스화된 롤업의 시퀀서로 선정될 수 있다.

AltLayer는 각 롤업당 최소 5명의 시퀀서를 구성할 것을 권장하며, 따라서 각 롤업은 비컨 레이어로부터 최소 5명의 서로 다른 검증자를 선택하여 운영된다.

비컨 레이어는 WASM과 WASM 기반 EVM을 모두 지원하지만, 일반 사용자를 위한 임의의 스마트컨트랙트 실행은 허용하지 않는다. 그 대신 네트워크의 주된 역할은 모든 시퀀서를 하나의 전용 네트워크로 집약하고, 이후 해당 검증자들을 롤업의 시퀀서로 선정 및 할당(assign)하는 것이다. 또한, 레이어 1에서 트랜잭션이 최종화되기 전, 소프트 확정(soft confirmation)을 제공한다.

계층형 최종성 (Tiered Finality)

블록체인은 일반적으로 최종화된 블록과 최종화되지 않은 블록만 구분한다. 반면 AltLayer는 블록 최종화를 위해 **계층형 방법론(tiered methodology)**을 사용한다. 이 계층형 방법론은 최종 사용자가 자신의 **보안 예산(security budget)**에 따라 트랜잭션의 최종성 상태를 결정할 수 있도록 한다. 최종화는 세 가지 서로 다른 계층으로 구성된다: 실행 레벨(Execution-level, 낮은 보안 예산용), 검증 레벨(Verification-level, 중간 보안 예산용), 그리고 롤업 레벨(Rollup-level, 높은 보안 예산용)이다.

블록이 서로 다른 최종화 계층을 거치면서, 블록의 최종성에 대한 신뢰 수준(confidence level)은 점점 높아진다. 신뢰 수준은 실행 레벨에서 가장 낮으며, 롤업 레벨에서 가장 높다.

실행 레벨 최종성 (Execution-level Finalization) 트랜잭션은 트랜잭션 풀에서 수집되어 블록 생성자에 의해 블록으로 묶인다. 이후 블록 생성자들이 참여하는 합의 프로토콜(consensus protocol)에 의해 블록이 승인된다. 이렇게 승인된 블록과 트랜잭션들은 실행 레벨에서 최종화된(finalized at the execution-level) 것으로 간주된다.

검증 레벨 최종성 (Verification-level Finalization) 블록이 합의 프로토콜을 거친 후, 검증자들이 해당 블록의 유효성을 검증한다. 검증을 통과한 블록은 검증 레벨에서 최종화(finalized at the verification-level)된 것으로 간주된다.

추가 검증자가 기존 검증자의 커밋을 **승인(affirm)**하거나 **거부(reject)**함으로써 더 강력한 최종성을 확보할 수 있다. 블록이 더 많은 승인(affirmation)을 받을수록 그 정확성이 높아진다. 검증자 정족수(quorum)를 통한 블록 합의는 AltLayer에서 **선택적(optional)** 기능으로 제공된다.

롤업 레벨 최종성 (Rollup-level Finalization)

검증자 정족수 합의를 통과한 블록은 일정 기간 동안 **챌린지(Challenge)**가 가능하다. 이 기간 동안 누구나 자신의 검증자를 운영하며 블록 검증 과정에 참여할 수 있다. 체인 상의 제안(proposal)과 챌린저(challenger)의 결과가 불일치할 경우, 챌린저는 블록에 대해 이의를 제기할 수 있다.

챌린지가 **성공**하면, 해당 시점 이후의 모든 블록은 무효 처리되며, AltLayer는 마지막으로 유효한 블록까지 **상태를 되돌림(state revert)** 한다. 반면 챌린지가 **실패**할 경우, 챌린저가 예치한 **스테이크(stake)**는 **슬래싱(slashed)**된다.

로드맵 (Roadmap)

2024년 1분기 (Q1 2024)

- Arbitrum Orbit와 Celestia, Espresso, EigenDA의 통합 테스트넷 출시
- OP Stack과 EigenDA, Avail 통합 테스트넷 출시
- ZK Stack 지원 추가

- **VITAL, MACH:** OP Stack 지원 버전 테스트넷 출시
 - **TGE (Token Generation Event)** 진행
-

17 2024년 2분기 (Q2 2024)

- Polygon CDK와 대체 데이터 가용성(alt DA) 통합 테스트넷 출시
 - **VITAL, MACH:** EigenLayer AVS 메인넷과 함께 OP Stack 지원 버전 메인넷 출시
 - 리스테이킹 롤업(restaked rollup)을 위한 **ALT 스테이킹(staking)** 지원
 - **VITAL, MACH:** Arbitrum Orbit 지원 버전 테스트넷 출시
-

17 2024년 3분기 (Q3 2024)

- **VITAL, MACH:** Arbitrum Orbit 지원 버전 메인넷 출시 및 EigenLayer AVS 메인넷 통합
 - 리스테이킹 롤업 메인넷(**Restaked Rollup Mainnet**) 출시
-

17 2024년 4분기 (Q4 2024)

- **WIZARD (AVS-as-a-Service 플랫폼)** 출시
 - **EigenLayer와 공동 AVS 해커톤(Hackathon)** 개최
 - Arbitrum Stylus 지원 추가
 - OP Stack과 Avail 및 기타 데이터 가용성 레이어 통합
 - Orbit Stack과 Avail 및 기타 데이터 가용성 레이어 통합
-

17 2025년 1분기 (Q1 2025)

- **Blitz 테스트넷 출시:** Babylon을 통한 BTC 리스테이킹 기반 신속 최종성(fast finality) 레이어
 - **RaaS 메트릭 대시보드(Metrics Dashboard for RaaS)** 출시
 - 신규 ZK 스택 및 DA 지원 추가
-

17 2025년 2분기 (Q2 2025)

- **AVS Wizard 메트릭 대시보드** 출시
 - **EigenLayer와 공동 AVS 부트캠프(Bootcamp)** 개최
 - **BolD 네트워크**를 통한 Orbit 체인용 탈중앙화 챌린저 네트워크 지원
-

17 2025년 3분기 (Q3 2025)

- **Monad** 및 **MegaETH** 지원
 - **Bera Chain**을 정산 레이어(Settlement Layer)로 지원
 - 병렬 EVM 롤업 스택(**Parallel EVM Rollup Stack**) 지원
 - **Blitz 메인넷(Mainnet)** 출시
-

2025년 4분기 (Q4 2025)

- **Superchain, AggLayer** 등 상호운용성(interoperability) 레이어 지원
- **인텐트 기반 상호운용성(Intent-based Interoperability)** 지원