Bitlayer 네트워크: 비트코인용 계산 계층

2.0

Bitlayer 연구팀
2025 년 7 월 9 일

개요

비트코인은 제한된 거래 처리량과 프로그래밍 가능성으로 인해 탈중앙화 금융(DeFi)에서 제한된 잠재력을 보이고 있다. 기존 레이어 2 솔루션은 새로운 신뢰 가정을 여럿 도입해왔으나, 이러한 솔루션의 보안성을 비트코인의 합의 메커니즘에 직접적으로 심는 데는 실패했다. 본 문서에서는 BitVM 패러다임을 활용해 이 문제를 해결하는 레이어 2 네트워크인 Bitlayer 를 소개한다. Bitlayer 의 핵심 기여는 새로운 재귀적 검증 프로토콜로서, Bitlayer 가 레이어 2 상태 전환의 연속적인 체인을 비트코인 상에서 검증 가능하게 정산할 수 있도록 최초로 구현했다는 점에 있다. 이는 단순한 데이터 기록을 넘어 비트코인의 작업 증명(PoW)에 기반한 보안을 실현한다. 또한 BitVM 브리지를 롤업 프로토콜에 깊게 통합하여 비트코인 자산의 안전한 전송을 가능하게 한다. 마지막으로, Bitlayer 는 빠른 합의 메커니즘으로 구동되는 모듈식 튜링 완전 실행 엔진을 설계되어 1 초 미만의 소프트 파이널리티를 제공한다. Bitlayer 는 새로운 세대의 탈중앙화 애플리케이션에 그동안 활용할 수 없었던 비트코인의 방대한 자본을 활용 가능하게 하여, 비트코인 디파이 생태계를 위한 기초 인프라를 구축한다.

1 소개

비트코인[1]은 탈중앙화 금융(DeFi)과 관련하여 막대한 잠재력을 지니고 있지만, 그 핵심 설계로 인해 거래 처리량과 프로그래밍 가능성 측면에서 제한을 받는다. 그렇기 때문에, 아직 활용되지 못한 비트코인 방대한 자본을 활성화하기 위해서는 안전하고 확장 가능한 레이어 2 솔루션이 필요하다.

그러나 기존 비트코인 확장 접근법에는 한계가 있다. 연합 다중 서명에 의존하는 사이드체인은 중앙화된 신뢰를 도입하여 비트코인의 보안 모델을 근본적으로 훼손한다. 한편, 비트코인을 위한 초기 롤업 설계는 거래 데이터를 L1 에 게시할 수 있지만, 여기에는 온체인 상태 전환의 유효성을 강제하는 메커니즘이 부재하다. 이러한 설계의 보안은 비트코인의 컨센서스에 의해 완전히 보장되지 않기 때문에 취약한 보안을 야기한다.

이는 중대한 의문을 제기한다. 바로 '새로운 신뢰 가정 없이 비트코인 메인넷 자체의 상태 유효성을 보장하면서, 확장 가능한 계산을 수행하는 비트코인 L2 를 구축할 수 있는가' 하는 의문이다.

본 백서는 롤업 아키텍처[7]와 BitVM 패러다임[2]을 통해 긍정적인 해답을 제시하는 레이어 2 네트워크인 Bitlayer 를 소개한다. Bitlayer 는 비트코인과 기존 레이어 2 솔루션의 한계를 극복하여 확장 가능한 계산을 가능하게 하는 동시에 그 기반이 되는 비트코인 블록체인에 그 보안을 앵커링한다. Bitlayer 의 주요 기여 사항은 다음과 같다.

- **모듈식 튜링 완전 실행 레이어:** Bitlayer 는 튜링 완전 스마트 계약을 구현하는 모듈식 실행 레이러를 설계하고 구현하여, 철저하게 설계된 블록체인 프로토콜을 활용하여 1 초 미만의 소프트 파이널리티를 실현하고, 디파이 및 게임 등 까다로운 애플리케이션에 이상적인 반응형 경험을 제공한다.

- **롤업을 위한 재귀적 비트코인 정산 프로토콜:** Bitlayer 는 재귀적 BitVM 기반 프레임워크를 활용하여 비트코인 상에서 레이어 2 상태 전이의 연속적인 클레임 체인을 정산하는 최초의 롤업 프로토콜을 설계하고 정식화한다. 이러한 프로토콜은 L2 의 유효성을 L1 에 직접 정착시켜 보안을 제공한다.
- **시너지를 창출하는 브리지와 롤업의 통합:** Bitlayer 는 BitVM 브리지 아키텍처에서 영감을 받아 안전한 자산 브릿지를 설계하고 구현한다. Bitlayer 의 핵심 혁신은 롤업 프로토콜과의 심층적 통합으로, 자산 보안과 롤업 유효성이 통합된 신뢰 모델에 의해 관리되는 원활하고 안전한 자산 전송 제공한다.

2 네트워크 아키텍처

Bitlayer 는 빠른 블록 생성을 위한 지분 증명(PoS) 합의와 비트코인 네트워크에 보안을 더하는 롤업 프레임워크를 결합한 이중 계층 아키텍처로 작동한다. PoS 계층은 검증자가 거래를 순차화하고 블록을 신속하게 생성할 수 있게 하여 높은 처리량과 EVM 호환 환경을 제공한다. 롤업 계층은 L2 체인의 상태를 주기적으로 비트코인 블록체인에 커밋하고 정산한다. 이 설계는 보안과 데이터 가용성을 위한 최상위 계층으로 비트코인을 활용하는 동시에, Bitlayer 네트워크가 확장 가능하고 효율적인 계산 레이어 역할을 수행하도록 한다.

2.1 네트워크 참여자 및 역할

네트워크는 검증자와 풀 노드, 이 두 핵심 참여자에 의해 관리된다.

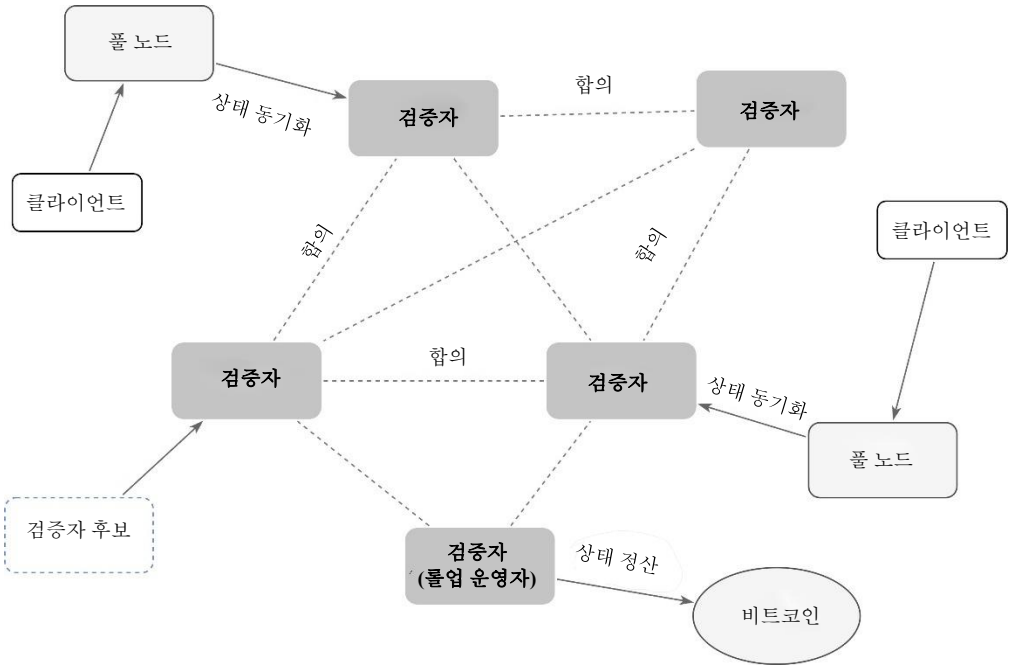

그림 1: 네트워크 아키텍처

- **검증자**: 검증자는 PoS 컨센서스의 중추적인 역할을 한다. 이들은 L2 블록을 생성하고 검증하여 네트워크의 안전성과 활성 상태를 보장한다. 검증자 집합에 참여하려면 후보자는 BTR 토큰을 스테이킹해야 하며, 합의에 대한 영향력은 총 스테이킹량에 비례한다. 여기에는 다른 BTR 보유자가 위임한 토큰도 포함될 수 있다.
  - **롤업 운영자**: 롤업 운영자는 검증자 집합에서 단일 검증자에게 할당되는 전문적인 역할로, 교대로 역할이 수행된다. 운영자는 L2 상태 전이를 배치 단위로 묶고, 암호학적 증명을 생성하며, 이를 비트코인 L1 에서 정산 처리하도록 제출한다. 책임을 보장하고 사기 행위를 억제하기 위해 운영자는 L1 에 상당량의 BTC 를 담보로 예치해야 한다. 검열과 중앙화를 방지하기 위해 운영자 역할은 주기적으로 순환된다.
- **풀 노드:** 풀 노드는 Bitlayer 네트워크 블록체인의 완전한 사본을 유지하며, 검증자를 신뢰하지 않고 모든 거래와 상태 전이를 독립적으로 검증한다. 이들은 프로토콜 규칙을 시행하고 네트워크 투명성을 보장하는 데 핵심적인 역할을 수행한다.

2.2 이중 거래 완결성(Dual-Level Transaction Finality)

Bitlayer 는 이중 완결성 모델을 제공하여 사용자와 애플리케이션이 속도와 비트코인 수 준의 중 원하는 것을 선택할 수 있도록 한다.

- **소프트 파이널리티(Soft Finality):** 거래 단계에서 블록은 Bitlayer 의 PoS 합의에 의해 확인되고, 확인이 이루어지면 1 초 이내로 소프트 파이널리티를 달성한다. 이로써 Bitlayer 는 검증자 집단의 경제적 지분 기반의 보안에 더불어 높은 반응성을 갖춘 사용자 경험을 신속하게 제공한다.
- **하드 파이널리티(Hard Finality):** 하드 파이널리티는 최고 보안 요소로, 비트코인 블록체인에서 거래를 포함하는 L2 상태가 정산되고 최종 확정될 때 이루어진다. 옵티미스틱 롤업의 이의 제기 기간으로 인해 이 과정에는 약 7 일이 소요된다. 하드 파이널리티의 보안은 사기 행위에 이의를 제기할 수 있는 단 한 명의 정직한 참여자만을 필요로 하므로, 비트코인의 자체 보안과 거의 동등한 수준의 보안을 제공한다.

드물게 L1 이의 제기가 성공하여 L2 상태와 정산된 L1 상태 간 불일치가 발생할 경우, 프로토콜은 설계에 따라 중단된다. 이후 네트워크 복구는 사용자 자산의 무결성을 보장하기 위해 이해관계자들 간의 사회적 합의에 따라 진행된다.

3 비트코인에서의 L2 상태 정산

레이어 2 롤업인 Bitlayer 의 보안은 비트코인으로부터 파생되었다. 이 장에서는 이러한 관계를 뒷받침하는 핵심 메커니즘인 정산에 대해 상세히 설명한다. 정산은 Bitlayer 의 고처리량 환경에서 실행되는 L2 상태 전이가 비트코인 L1 에 커밋되고 최종 확정되는 과정이다. 이 과정을 통해 Bitlayer 는 비트코인의 보안성을 상속한다. 그러나 문제는 이러한 보안성을 비트코인의 제한적이고 튜링 불완전 스크립트 환경에서 달성할 수 있는가다.

Bitlayer 는 BitVM 패러다임에서 영감을 받은 새로운 정산 프로토콜에서 그 해결책을 찾았다. 이 장에서는 Bitlayer 의 프로토콜을 체계적으로 해체하여 살펴보겠다. 우선, 상태 청구 개념을 정의하고, Bitlayer 의 하이브리드 검증 접근법에 대해 설명해 보도록 하겠다. 필요한 암호화 프리미티브(cryptographic primitive)에 대해 소개한 다음, 단일 상태 클레임을 해결하는 프로토콜에 대한 설명으로 넘어가도록 하겠다. 그리고 마지막으로는 이 프로토콜이 L2 클레임의 연속적인 체인을 해결하는 재귀적 프로토콜로 확장되어 전체 롤업의 핵심 구조를 형성하는 지에 대해 기술하도록 하겠다.

3.1 L2 상태 클레임 정의

블록체인은 근본적으로 **상태 전이 함수(STF)**로 정의할 수 있으며, 이는 Y 로 표현된다. 이 결정적 함수는 네트워크의 **상태**(s)가 어떻게 진화하는지를 규정한다. 상태는 모든 계좌 잔액과 계약 데이터를 포함하며, 32 바이트 머클 루트로 표현된다. STF 는 현재 상태인 s_t 와 L2 **트랜잭션 배치**(T)를 받아 다음 상태인 s_{t+1}:

$$s_{t+1} = \mathrm{Y}(s_t,\ T)\text{를 생성한다.}$$

여기서 t 는 트랜잭션 배치의 인덱스다. 블록체인의 역사는 초기 **제네시스 상태**(s_0)에서부터 시작된다.

상태 클레임(Φ)은 롤업 운영자가 비트코인 L1 의 스마트 계약에 제출하는 공식적 주장이다. 주장의 목적은 특정 트랜잭션 배치 처리를 하여 생성된 새로운 L2 상태를 확정하는 것이다. 이 클레임은 L2 활동을 L1 에 연결하는 앵커 역할을 하여 Bitlayer 네트워크가 비트코인의 보안을 계승할 수 있도록 한다.

$$\Phi = \{s_{t-1},\ s_t,\ T\}$$

3.2 암호화 프리미티브

정산 프로토콜은 두 개의 고급 암호화 프리미티브, 간결한 비대화형 인수(SNARG)과 해시 기반 일회성 서명 체계에 큰 기반을 두고 있다.

3.2.1 Groth16 SNARG

Groth16 문서[4]에 따르면, 관계 R 에 대한 SNARG 는 세 가지 확률적 다항시간 알고리즘(설정, 증명, 검증)으로 구성된다.

- $\delta \leftarrow$ SNARG.Setup(R): 주어진 관계에 대해 공통 참조 문자열 δ 를 생성하는 설정 알고리즘.
- $\pi \leftarrow$ SNARG.Prove($R,\ \delta,\ \Phi,\ \omega$): 공통 참조 문자열 δ, 주장 Φ, 증인 ω 를 입력으로 받아 증명 논증 π 를 생성하는 증명 알고리즘.
- 0/1 $\leftarrow$ SNARG.Vfy($R,\ \delta,\ \Phi,\ \pi$): 증명을 수락하거나 거부하는 검증 알고리즘.

SNARG 는 완벽한 완전성, 계산적 타당성, 그리고 Bitlayer 정의 상의 완전한 간결성을 충족한다.

정의 1(완전 간결성). *프로토콜(Setup, Prove, Vfy)은 검증 절차 Vfy 가 보안 매개변수 λ 에 대해 다항 시간 안에 실행되는 동시에, 증명 π 의 크기가 λ 에 대해 다항식일 때, 완전히 간결한 것으로 정의된다.*

3.2.2 해시 기반 일회성 서명(HOTS)

비트코인 스크립트 언어는 OP CHECKSIG 명령 코드[6]를 통해 임의의 오프체인 메시지 검증이 아닌 거래 서명을 검증하도록 설계되었다. 이러한 기능을 확장하기 위해 BIP348 과 같은 개선 제안이 소개되었지만, 이를 채택하기 위해서는 네트워크 합의 변경이 필수적이다. Bitlayer 는 이러한 한계를 극복하기 위해 해시 기반 일회성 서명 방식(HOTS)[5, 8]을 활용한다. 이 방식의 장점은 해시 함수가 비트코인 스크립트에서 네이티브하며 저렴한 계산 비용을 제공한다는 점이다.

Bitlayer 의 HOTS 변형은 네 가지 알고리즘으로 구성된다.

- (*sk, pk*) ← HOTS.setup(λ): 보안 매개변수로부터 비밀 키와 공개 키 쌍을 생성.
- s ← HOTS.publish(*pk, b*): 비트코인 스크립트에 커밋을 게시하여 *b* 길이의 메시지에 대한 서명을 검증하도록 준비.
- w ← HOTS.sign(*sk, m*): 비밀 키로 메시지 m 에 서명하여 증인 w 를 생성.
- (0/1, *m*) ← HOTS.verify(*pk,* w): 증인 w 검증. 유효할 경우, '1'이 반환되고 스택에 원본 메시지인 *m* 을 게시하여 추가 온체인 처리를 진행한다.

이 최종 속성(서명된 메시지의 온체인 게시)은 연속적인 상태 클레임 연결에 있어 핵심적인 구성 요소이다. 이 부분에 대해서는 3.5 절에서 상세히 다루고 있다.

3.3 프로토콜 개요

전체 정산 프로토콜은 BitVM 스타일의 스마트 계약에 구현되어 있으며, 이때 계약은 단일한 모놀리식 계약이 아닌 사전 서명된 여러 비트코인 트랜잭션을 포함하는 복잡한 그래프 구조를 취한다. 참여자들은 공동으로 이 트랜잭션 그래프에 사전 서명해야 하며, 사전 정의된 경로를 엄격하게 따라 소통해야 한다. 기존 BitVM 프로토콜이 브리징을 위해 외부 체인과 비트코인 양측의 이벤트에 대한 청구 정산에 중점을 두었던 데에 반해[3], Bitlayer 의 프로토콜은 더 복잡한 구조를 취한다. Bitlayer 프로토콜에서는 L2 상태의 개별적인 변화를 나타내는 일련의 *연속적인* 청구을 정산해야 하며, 이 일련의 청구가 연속적이고 끊기지 않도록 보장해야 한다.

이 프로토콜은 재귀적 구조라는 개념으로 정의할 수 있다. 3.4 절에서는 단일 상태 청구를 정산하는 하위 프로토콜을 상세히 설명한다. 이후 3.5 절에서는 이 단일 클레임 검증 메커니즘이 연속적인 클레임 체인을 정산하는 더 광범위한 프로토콜에서 어떻게 재귀적으로 구현되었는지를 살펴 본다. 이 두 구성 요소를 결합함으로써, Bitlayer 는 비트코인 상에서 Bitlayer 네트워크 상태 정산을 위한 완전한 롤업 프로토콜을 구축하였다.

3.4 단일 클레임 해결

3.4.1 BitVM2 패러다임

클레임의 온체인 검증은 옵티미스틱 방식으로 수행된다. Bitlayer 의 검증자 프로그램은 비트코인 스크립트로 표현된다. 그러나 BitVM Alliance 의 Groth16 검증기에 대한 획기적인 연구에서 입증된 바와 같이, 이러한 검증기를 단일하게 구현하게 될 경우 단일 비트코인 트랜잭션에서 이를 직접 실행하기에 그 크기가 너무 방대해진다는 문제가 존재한다. 따라서 BitVM2 패러다임[2]은 대규모 검증 프로그램을 더 작은 하위 프로그램 체인("청크")으로 분할한다. 그런 다음 사기 증명 게임(fraud-proof game)을 진행한다. 이때 운영자의 클레임은 이의 제기자가 특정한 두 청크 사이의 잘못된 계산을 정확하게 지적하지 않는 한 올바른 것으로 가정한다.

3.4.2 프로토콜 역할

BitVM 클레임 처리에 사용되는 스마트 계약은 다음과 같이 정의된 참여자 집합을 포함한다.

1. **증명 위원회:** 위원회는 새로운 주체가 아닌 기존 Bitlayer 네트워크의 검증자 집합으로 구성된다. 위원회는 프로토콜을 정의하는 트랜잭션 그래프에 사전 서명하는 책임을 공동으로 갖는다.

2. **프로토콜 참여자:** 정산 게임(settlement game)의 활성 참여자는 클레임 제출을 담당하는 단일 지정 **운영자**와 임의의 수의 **감시자**를 포함한다. 감시자 역할은 다른 검증자를 포함한 누구나 맡을 수 있으며, 운영자를 감시하고 부정한 클레임에 이의를 제기하는 일을 수행한다.

3.4.3 단일 클레임 검증 프로토콜

단일 클레임을 검증하는 프로토콜은 비트코인 타임락에 의해 통제되는 시간 제한형 이의제기-응답 게임으로 진행된다. 운영자와 감시자는 모두 지정된 시간 내에서 작업을 마쳐야 하며, 그렇지 못할 경우 페널티를 받는다. 프로토콜은 세 가지 주요 단계로 나뉘며, 이는 증명 위원회가 사전 서명한 일련의 상호 연결된 비트코인 거래에 대응한다.

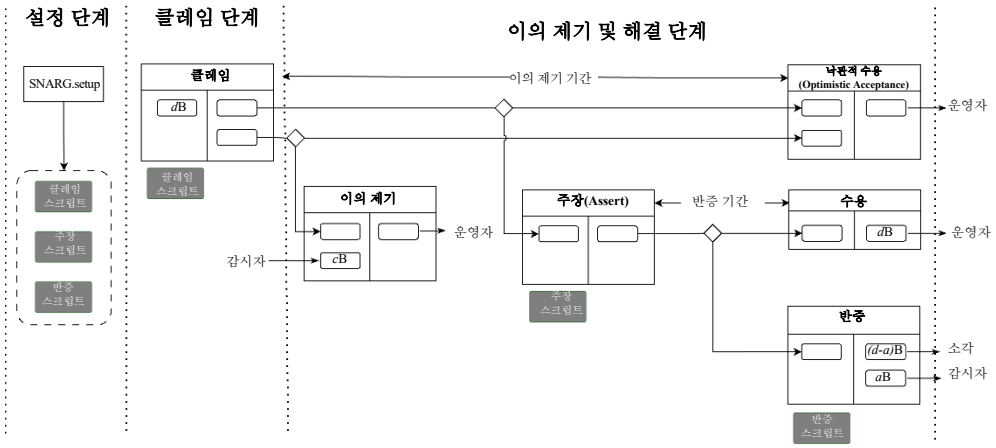

그림 2: 단일 클레임 검증 프로토콜

1. 설정 단계 게임 시작 전, 신뢰할 수 있는 공개 설정 환경에서 필요한 스크립트를 생성한다. STF(Y)를 사용하여 SNARG.Setup 을 통해 공통 참조 문자열 δ 를 생성한다. $f(\pi) = \text{SNARG.Vfy}(R, \delta, \Phi, \pi)$ 의 검증기 프로그램을 하위 프로그램 집합 $\{f_i(I_i)\}$으로 분할한다. 세 가지 잠금 스크립트(프로세스를 시작하는 클레임 스크립트, 운영자가 모든 f_i 결과를 게시하는 주장 스크립트, 그리고 감시자가 사기 행위를 증명하는 데 사용하는 하위 프로그램별 반증 스크립트)를 생성한다. 클레임 스크립트와 주장 스크립트는 운영자가 서명된 클레임과 모든 하위 프로그램의 결과를 게시하도록 강제하는 다중 HOTS 로 구성된다. 반증 스크립트는 모든 하위 프로그램을 포함하며 Taproot 주소에 구성된다. 생성된 스크립트는 사전 서명된 트랜잭션 그래프의 아웃풋에 내장된다.

2. 클레임 단계 운영자가 *클레임* 트랜잭션을 발행하여 프로세스를 시작한다. 이때 트랜잭션은 이전 UTXO 에서 지출되며 담보(예: d BTC)를 동결된다. 동결을 해제하기 위해서는 운영자가 HOTS 키를 사용하여 상태 클레임 Φ_t 에 서명하고 이를 게시해야 한다. 이 작업은 타이머 $\Delta_{\text{challenge}}$ 를 시작한다. 감시자는 이 클레임을 관찰하여 로컬에서 STF 를 실행할 수 있다. 클레임이 유효하지 않다고 판단할 경우, 타임아웃 전에 *이의 제기* 트랜잭션을 발행할 수 있다. 이의가 제기되지 않을 경우, 운영자는 낙관적 수용 트랜잭션을 통해 담보를 회수하며, 클레임이 해결된 것으로 간주된다.

3. 이의 제기 및 해결 단계 감시자가 *이의 제기* 트랜잭션을 발행할 경우(사소한 건에 대한 이의 제기 방지를 위해 더 적은 담보금(c BTC)을 예치해야 할 수 있음), 비관적 프로세스를 수행한다. 이 경우, 운영자는 *주장* 트랜잭션을 실행해야 한다. 운영자는 분할된 검증 프로그램을 로컬에서 실행하고 HOTS 키를 사용하여 *모든* 하위 프로그램 $\{f_i\}$의 결과에 서명해야 하며, 이를 게시해야 한다. 이 작업은 또 다른 타이머인 Δ_{disprove}를 시작한다.

이후, 감시자는 운영자가 게시한 결과 시퀀스를 검토한다. 감시자가 운영자가 게시한 입력값에서 단 한 건이라도 운영자가 공개한 출력값이 하위 프로그램 f_i의 실행 결과와 불일치하는 f_i를 발견할 경우, 사기 행위가 발견된 것으로 판단한다. 감시자는 이후 *반증* 트랜잭션을 구성할 수 있다. 이 트랜잭션은 부정한 하위 프로그램의 출력값을 가져와서, 해당 논리 청크를 온체인에서 실행하여 불일치를 증명한다. *반증* 트랜잭션이 성공할 경우, 운영자의 담보가 몰수되며, 일정 부분(a BTC, where $a > c$)이 감시자에게 지급된다. 운영자가 모든 f_i의 결과를 올바르게 게시하고, 감시자가 유효한 *반증* 트랜잭션을 제때 제출하지 못할 경우, 운영자는 *수용* 트랜잭션을 통해 클레임을 확정하고 담보를 회수한다.

3.4.4 보안 속성

본 프로토콜은 최소한 한 명의 정직한 감시자가 존재한다는 가정 하에 보안성을 갖추도록 설계되었다. 이 보안성은 세 가지 핵심 속성을 기반으로 하며, 이는 6.2 장에서 상세히 기술되어 있다.

- **완전성:** 프로토콜을 정확하게 따르고 유효한 상태 클레임을 제출하는 정직한 운영자는 페널티를 받지 않는다.
- **건전성:** 정직한 감시자는 언제나 유효한 *반증* 트랜잭션을 구성할 수 있으므로, 부정한 클레임을 제출한 부정직한 운영자는 페널티를 반드시 받게 된다.
- **효율성:** 전체 클레임 검증 프로세스는 승인 또는 거절 여부와 관계없이 프로토콜의 타임 락에 의해 정의된 제한된 시간 내에 종료된다.

3.5 클레임 체인 처리

독립적인 단일 클레임 처리는 위에서 설명한 프로토콜만을 참조하여 실행할 수 있다. 그러나 롤업은 L2 상태의 지속적인 진화를 나타내는 일련의 클레임을 계속해서 처리할 것을 요구한다. 이러한 처리는 재귀적으로 클레임을 연결하도록 프로토콜을 확장하여 이루어진다.

3.5.1 HOTS 를 통한 청구 연결

클레임 연결의 핵심은 트랜잭션 그래프의 구조에 있다. 각 *클레임* 트랜잭션은 다른 출력값 외에도 **클레임 커넥터(claim connector)**라는 특수한 UTXO 를 생성한다. 다음 클레임(클레임 N+1)을 제출하기 위해서, 운영자는 클레임 N 트랜잭션에서 생성된 클레임 커넥터 UTXO 를 사용해야 한다. 이 커넥터의 잠금 스크립트는 운영자가 HOTS 키를 사용하여 클레임 N+1 의 데이터 패키지를 서명하고 공개하도록 요구한다. 각 클레임 트랜잭션은 선행 트랜잭션의 출력값을 취해야만 생성되므로, 이러한 설계는 인접한 클레임을 위조 불가능한 체인으로 자연스럽게 시간순으로 연결한다. 비트코인 타임락은 규칙적인 주기를 강제하여 운영자가 클레임을 너무 빠르거나 느리게 제출하는 것을 방지한다.

3.5.2 트렁크 트랜잭션 그래프와 병렬 검증

이 프로토콜의 재귀적 구조는 일련의 클레임을 연결하는 기본 **트렁크**를 가진 트랜잭션 그래프를 생성한다. 트렁크의 각 클레임은 단일 클레임 검증(3.4 절 참조)을 위한 완전한 하위 그래프으로 나누어진다.

이 설계의 핵심 특징은 다음 클레임 제출하기 위해 선행 클레임 검증의 하위 프로토콜의 최종 처리를 기다릴 필요가 없다는 점이다. 운영자는 클레임 N에 대한 이의 제기 창이 열려 있는 한 클레임 N+1 을 제출할 수 있다. 이러한 병렬 처리는 효율적이지만, 동시에 연쇄적 장애를 처리할 메커니즘을 필요로 한다. 클레임 N에 대한 이의 제기가 성공적으로 이루어지면, 프로토콜은 해당 상태를 무효화하여 이후 모든 클레임(N+1, N+2, ...)의 전제를 자동으로 무효화한다. 클레임에 대한 이의 제기가 성공적으로 이루어졌을 경우, 각 클레임에 대한 담보가 몰수되므로 합리적인 운영자는 경제적인 이유로 추가 클레임의 제출을 중단하게 된다. 이후 트렁크는 ClaimTimeout 트랜잭션을 통해 종료된다.

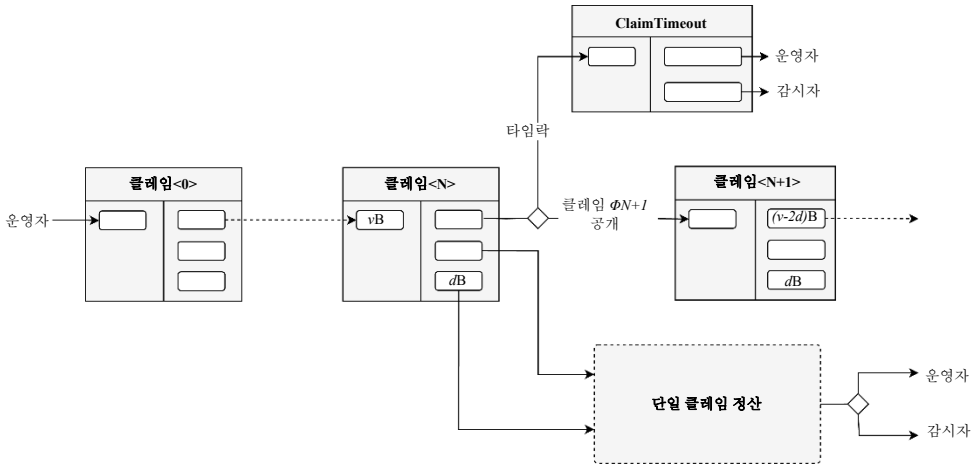

그림 3: 트렁크 트랜잭션 그래프

3.5.3 트랜잭션 그래프 재구성 및 에포크

롤업의 전체 수명 주기(예. 100 년) 동안 사용될 트랜잭션 그래프를 검증자가 구축, 사전 서명 및 저장하는 것은 계산 및 물리적인 측면에서 불가능하다. 그렇게 하기 위해서는 선제적으로 불가능한 양의 비트코인을 담보로 동결해야 하며, 향후 프로토콜의 업그레이드를 불가능하게 한다.

이 문제를 해결하기 위해 Bitlayer 는 **재구성**을 도입하였다. 프로토콜의 타임라인은 개별적인 **에포크**로 나뉘며, 각 에포크는 고정된 수의 클레임으로 구성된다(예. 2 주간 지속). 재구성 이벤트는 에포크 전환 시 발생한다. 검증자 집합은 각 증명식마다 다음 에포크의 트렁크 트랜잭션 그래프에 대해서만 사전 서명을 진행하면 된다. 그렇게 때문에 이러한 방식은 검증자의 부담을 덜어준다.

종료 창(Exit Window) 또한 재구성 시점은 프로토콜 업그레이드나 검증자 집합 변경이 이루어질 수 있는 시점이기도 하다. 이러한 변경 사항은 시스템의 보안 가정이나 신뢰 매개변수를 변경할 수 있다. 사용자 주권을 보호하기 위해 Bitlayer 는 의무적으로 **종료 창**을 제공한다. 에포크 $N + 2$ 의 구성은 에포크 N 번동안 제안되고 확정된다. 이를 통해 사용자는 에포크 $N + 2$ 의 새로운 검증자 집합과 트랜잭션 그래프를 검토하는 데 에포크 $N + 1$ 의 시간을 전부 사용할 수 있게 된다. 사용자가 예정된 변경 사항을 승인하지 않을 경우, 전체 에포크 시간 내에 새로운 구성이 적용되기 전에 자산을 인출(예. BitVM 브리지를 통한 BTC 페깅 아웃)하여 시스템을 완전히 이탈할 수 있다.

검증자 인센티브 모든 검증자는 참여를 위해 BTR 토큰을 스테이킹해야 한다. 각 에포크의 트랜잭션 그래프 사전 서명식은 L2 시스템 계약을 통해 조정된다. 서명식 제때 참여하지 못할 경우, 검증자가 스테이킹한 BTR 의 일부가 몰수되게 하여 프로토콜을 지연시키려는 공격을 강하게 억제한다.

3.5.4 재구성 프로세스

재구성 프로세스는 L2 시스템 계약에 의해 조정된다. 지정된 운영자는 다음 에포크의 트랜잭션 그래프에 필요한 모든 정보를 준비하며, 각 검증자는 독립적으로 이를 생성 및 서명하고, 서명을 L2 계약에 제출한다. 유효 서명이 압도적 다수($N-f$)만큼 수집되면, 집계되어 증명이 완료된다.

이 프로세스는 비트코인에서의 **재구성 트랜잭션**에서 마무리된다. 이 트랜잭션에서는 새 에포크의 모든 클레임에 필요한 총 담보가 동결되고, 검증자 프로그램 커밋먼트 δ, 운영자 신원, 타임락 값 등 업데이트된 구성 매개변수가 기록된다. **재구성 트랜잭션**은 신속한 구성 공지를 위해 사전 서명이 완료된 즉시 발행되어야 한다. 최초의 재구성 트랜잭션인 **에포크 0 재구성 트랜잭션**은 전체 롤업 프로토콜을 부트스트랩하고 비트레이어 네트워크의 제네시스 상태 s_0을 기록한다.

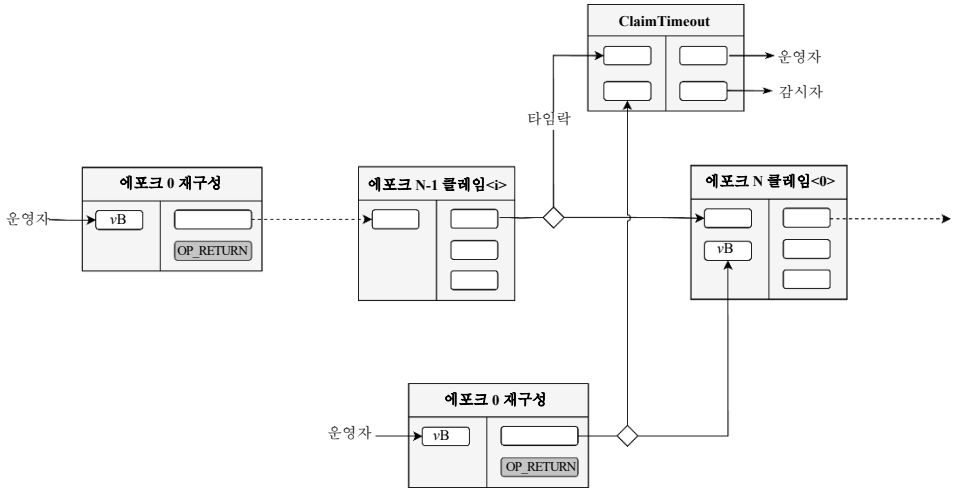

그림 4: 트랜잭션 그래프 재구성

3.6 요약

요컨데, Bitlayer 정산 프로토콜은 비트코인 상에 끊임 없지만 비교적 부담이 적은 BitVMstyle 트랜잭션으로 구현된다. 이 그래프는 순환적이며, 재구성 트랜잭션을 통해 서로 연결된 에포크별 하위 그래프로 구성된다. 각 에포크의 하위 그래프는 시간순으로 연결된 상태 클레임의 트렁크를 포함하며, 각 클레임에는 자체 검증 하위 그래프가 동반된다. 이때 이 하위 그래프는 한 명의 정직한 참여자만으로도 L2 상태의 정확성을 보장할 수 있게끔 하는 정교한 이의 제기-응답 게임으로 이루어져 있다. 이러한 아키텍처는 Bitlayer 가 비트코인의 독보적인 PoW 합의에 기반할 수 있도록 하는 동시에 높은 수준의 확장성과 프로그래밍 가능성을 달성할 수 있도록 한다.

4 상태 전이 함수 및 배치 실행

3 장에서는 비트코인 상태 클레임 정산 프로토콜에 대해 알아봤다면, 본 장에서는 클레임의 유효함을 주장하는 계산 프로세스인 비트레이어 상태 전이 함수(STF)를 정의한다. L2 블록 배치의 올바른 상태 전이는 롤업을 위한 진척의 기본 단위이다. 본 장에서는 Bitlayer 고유의 EVM 기반 STF 구성 요소에 대해 알아보고, 이를 실행하기 위해 증명을 생성하는 다단계 재귀 증명 파이프라인을 살펴본다. 이 계산 프로세스는 그 자체가 운영자의 클레임의 주장을 구성하며, 감시자는 누구든 정산 게임을 통해 이에 이의 제기할 수 있다.

4.1 Bitlayer 네트워크 STF

Bitlayer 네트워크 STF 는 이더리움 EVM [9]의 검증된 원칙에 부합하여 개발자에게 친숙하고 강력한 환경을 제공한다. 그러나 비트코인 롤업으로서, 브리지된 비트코인 자산 처리 및 L1 메시지 처리와 같은 고유한 요구사항을 해결하기 위해 추가적인 시스템 레벨 논리와 특수 계약을 통해 EVM 을 확장한다.

4.1.1 가스 및 수수료

Bitlayer 네트워크의 거래 수수료는 **BTC** 로만 결제가 가능하다. 이러한 설계는 네트워크 사용하기 위해 새로운 네이티브 토큰을 매수할 필요 없이 이미 보유하고 있는 자산을 사용할 수 있도록 하기 때문에 비트코인 사용자에게 원활하고 일관된 경험을 제공한다. 트랜잭션은 BTC 로 결제되지만, 레이어 2 아키텍처의 효율성을 반영하여 네트워크는 매우 낮은 수준의 수수료율을 제공한다.

Bitlayer 는 트랜잭션 비용을 세 가지 구성 요소로 나누는 다차원 가스 모델을 채택한다.

- **실행 수수료:** 표준 이더리움 모델에서와 유사하게, EVM 에서 트랜잭션을 실행하는 데 드는 계산 비용을 충당하는 데 사용되는 수수료이다.
- **저장 수수료:** 새 계좌 생성 또는 계약 저장소 업데이트와 같은 L2 상태 수정에 대한 비용이다.

실행 수수료와 저장 수수료는 네트워크의 보안을 책임지는 검증자에게 분배된다. 이러한 수수료 분배 메커니즘은 정밀하고 지속 가능한 경제 모델을 창출한다.

4.1.2 프로토콜 계약

프로토콜 계약은 제네시스 시점에 존재하는 특수 목적의 스마트 계약 집합으로, Bitlayer 프로토콜의 핵심 구성 요소이다. 이 계약의 논리가 Bitlayer 네트워크 운영의 핵심임에도, 이를 네이티브 코드가 아닌 스마트 계약으로 구현한 이유는 명확한 인터페이스 제공과 확립된 거버넌스 절차를 통한 향후 업그레이드를 가능하게 하기 위해서이다.

시스템 구성 계약 시스템 구성 계약은 네트워크의 중앙 제어판 역할을 하며 핵심 프로토콜 매개변수를 키값 쌍으로 관리된다. 이 예시로는 블록 가스 한도(<블록 가스 한도, 10,000,000>)와 검증자 집합 크기가 있다. -

- **재구성 가능성:** 대부분의 매개변수는 거버넌스 제안을 통해 업데이트할 수 있다.
- **업데이트 주기:** 업데이트 주기는 그 영향에 따라 결정된다. 특정 수수료 배수와 같은 일부 매개변수의 경우, 블록 경계 시점마다 수정될 수 있다. 반면 합의 엔진과 같이 시스템에 더 큰 영향을 미치는 매개변수의 경우, 안전하고 질서에 맞게 업데이트가 이루어질 수 있도록 에포크 경계 시점에서만 수정이 가능하다.
- **보안:** 다수의 매개변수가 네이티브 프로토콜 코드에 의해 직접 읽히므로, 수정 사항은 네트워크 안정성이나 보안을 훼손하지 않도록 수정 사항을 신중하게 평가되어야 한다.

검증자 관리 계약 이 계약은 L2 블록 생성 및 L1 증명식에서 중요한 Bitlayer 네트워크 검증자 집합의 수명 주기를 규정한다.

- 검증자 승인 및 제거:
  - 검증자 후보가 되기 위해서는 계약에 BTR 토큰을 최소 수량만큼 스테이킹해야 한다. 후보자는 다음 에포크 시작 시점에 활성 검증자 집합으로 승격된다. 네트워크 안정성 유지를 위해 후보 대기열에서 활성군으로 승격되는 신규 검증자 수는 각 에포크마다 제한된다(예. 전체 집합 크기의 10%).
  - 활성 검증자의 자발적 이탈 또한 에포크별로 제한된다.
  - 슬래싱으로 인해 스테이킹 금액이 필수 최소값 미만으로 떨어진 검증자는 다음 에포크 경계 시점에서 활성군에서 강제 제거된다.
- 운영자 선출:
  - 각 에포크마다 프로토콜은 검증자 활성군에서 한 명의 롤업 운영자를 선출한다. 선출된 운영자는 지정된 시간 내에 비트코인 L1 에 담보 예치 트랜잭션을 제출해야 한다. 그러지 못할 경우, 해당 검증자에게 페널티가 부여되며, 정산 프로세스의 운영을 위해 새로운 선출이 진행된다.
- 보상과 페널티:
  - 운영자는 네트워크 보안 유지에 참여한 대가로 BTR 토큰을 보상으로 받는다. 보상은 각 검증자의 총 스테이크 비율에 따라 분배된다.
  - L2 블록 제안자는 트랜잭션 수수료와 더 큰 블록 보상을 받는다.
  - 지정된 롤업 운영자는 비트코인 L1 에서 상태 클레임 정산에 사용된 운영 비용의 보상으로 추가 BTR 보상을 받는다.
  - 각 에포크의 트랜잭션 그래프 사전 서명식에 참여한 검증자는 증명 보상을 받는다.
  - 프로토콜 규칙을 준수하지 못할 경우(예. 블록 투표 누락, 사전 서명식 참여 실패) 페널티가 부과되어 검증자가 스테이킹한 BTR 의 일부가 삭감된다.

비트코인 라이트 클라이언트(BLC) 계약 BLC 계약은 네트워크가 비트코인 L1 의 무신뢰 게이트웨이 역할을 한다. 이 계약은 캐노니컬 비트코인 체인 추적과 L1-L2 메시지 처리, 두 가지 주요 역할을 가진다.

- **캐노니컬 체인 추적:** 프로토콜은 오라클을 통해 비트코인 블록 헤더를 BLC 에 제출한다. 이 역할은 기본적으로 롤업 운영자가 수행한다. 그러나 운영자가 이를 수행하지 못할 경우, 누구나 블록 헤더를 제출할 수 있으므로 프로세스는 계속 활성 상태를 유지할 수 있다. BLC 계약은 일시적 포크에서 생성된 헤더를 포함해 제출된 모든 헤더를 추적하며, 가장 무거운 체인을 선택하는 최중량 체인 규칙(heaviest-chain rule)을 따라 캐노니컬 체인을 유지한다. 이후, 제출된 블록은 **시스템 구성 계약**에서 정의한 임계값(예. 6)에서 정의된 확인 횟수를 달성하면 최종 확정된 것으로 간주된다.

- **L1-to-L2 메시지 처리:** BLC 계약은 최종 확정된 비트코인 블록에서 특정 L1-to-L2 메시지를 스캔하고 이를 실행하는 L2 트랜잭션인 **내재적 트랜잭션(intrinsic transaction)**으로 변환한다. 이러한 메시지는 다음 내용을 포함한다.
  - **브리지 예금 이벤트:** 사용자가 L1 에서 BitVM 브리지 계약에 BTC 를 예금할 경우, 브리지 예금 이벤트가 기록된다. BLC 계약은 이 이벤트를 찾아서 L2 에서 해당 브리지-민트의 내재적 트랜잭션을 생성하여 사용자에게 동등한 래핑된 자산을 지급한다. 이로써 L2 에서 별도의 사용자 작업 없이 페그인 프로세스가 자동화된다.
  - **강제 트랜잭션:** 사용자는 비트코인 블록체인에 직접 데이터를 기록함으로써 L2 트랜잭션을 강제로 포함할 수 있다. 이는 강력한 검열 저항 메커니즘으로, L2 검증자 집합 전체가 사용자를 검열하고자 할 시에도 사용자가 언제든 롤업과 상호작용할 수 있도록 한다.

프로토콜은 컨센서스 및 롤업 메커니즘을 통해 이러한 메시지가 제때 처리될 수 있도록 보장한다.

- **컨센서스 집행:** 새로운 블록을 제안하기 전에 검증자는 BLC 계약을 쿼리하여 보류 중인 내재적 트랜잭션을 생성해야 한다. 이러한 트랜잭션은 제안된 블록의 일반 사용자 트랜잭션보다 앞 부분에 위치할 수 있도록 맨 처음에 포함되어야 한다. 내재적 트랜잭션 수가 단일 L2 블록의 용량을 초과할 경우, 결정적 순서로 여러 블록에 걸쳐 처리된다.

- **롤업 집행:**STF 정의에 따르면, 일괄 처리된 L2 블록에 대한 상태 클레임은 해당 최종 확정된 비트코인 블록으로부터의 모든 L1-to-L2 메시지를 정확히 처리해야 합니다. L1 메시지를 누락하거나 잘못 처리한 상태를 기반으로 클레임을 제출한 운영자는 사기성 클레임을 제출한 것으로 간주되어 성공적으로 이의 제기를 받고 처벌을 받게 됩니다.STF 정의에 따르면, 일괄 처리된 L2 블록에 대한 상태 클레임은 해당 최종 확정된 비트코인 블록으로부터의 모든 L1-to-L2 메시지를 정확히 처리해야 합니다. L1 메시지를 누락하거나 잘못 처리한 상태를 기반으로 클레임을 제출한 운영자는 사기성 클레임을 제출한 것으로 간주되어 성공적으로 이의 제기를 받고 처벌을 받게 됩니다. STF 정의에 따라, L2 블록 배치의 상태 클레임은 해당 최종 확정된 비트코인 블록의 *모든* L1-to-L2 메시지를 정확히 처리해야 한다. L1 메시지를 누락하거나 잘못 처리한 상태를 기반으로 클레임을 제출한 운영자는 부정한 클레임을 제출한 것으로 간주되어, 이의 제기가 받아들여지며 페널티를 받게 된다.

브리지 계약 L2 의 브리지 계약은 L1 의 BitVM 브리지 계약과 함께 자산의 안전한 양방향 흐름을 지원한다.

- **페그인(Peg-In):** 계약은 BLC 가 생성한 브리지-민트 내재적 트랜잭션을 처리하여 사용자의 계정에 해당하는 L2 래핑 자산을 발행한다.
- **페그아웃(Peg-Out):** 자산을 인출하려면 사용자는 L2 에서 브리지 계약을 호출하는 트랜잭션을 시작해야 한다. 계약은 사용자의 L2 자산을 소각하고 L2 이벤트를 발생시킨다. 이 이벤트는 추후 L1 브리지 메커니즘이 인출 처리를 위해 수신하는 메시지 역할을 한다.
- **준비금 증명(PoR):** 계약은 Bitlayer 상의 모든 브리지된 자산의 완전하고 투명한 원장을 유지하여 누구나 언제든지 준비금 증명(PoR)을 생성할 수 있도록 한다.

브리지와 페그아웃 메커니즘의 상세한 아키텍처는 5 장에서 더 자세히 살펴보도록 하겠다.

4.2 증명 파이프라인

이 롤업 프로토콜의 상태 전이를 검증하기 위해, 프로토콜은 영지식 가상 머신(zkVM) 기반의 다단계, 비동기, 재귀적 증명 시스템을 채택한다. 이 시스템은 비트코인 네트워크에서 검증하기 쉽고 간결한 래핑된 증명을 생성하도록 설계되었으며, 동시에 거버넌스 프로세스를 통해 전체 증명 시스템이 안전성과 업그레이드 가능성을 보장한다.

4.2.1 코드 제어 그룹(CodeControlGroup)을 통한 무결성 및 업그레이드 가능성

증명 파이프라인 전체의 무결성은 핵심 계산 엔진의 유효성과 무결성에 기반한다. 즉, 블록 실행 로직, 배치 집계 로직, 재귀 로직을 포함한 zkVM 에서 실행되는 모든 프로그램이 유효성과 무결성이 보장되어야 한다는 것이다. 코드 제어 그룹은 이러한 목적을 보장하기 위해 설계된 핵심 보안 메커니즘이다.

코드 커밋먼트(CodeCommitment): **zkVM 프로그램의 고유한 지문** zkVM 은 완료된 프로그램 모음에 대해 각각 고유한 코드 커밋먼트를 생성한다. 이 커밋먼트는 특정 버전의 프로그램에 대한 유일하고 불변한 지문 역할을 한다. 코드의 변경은 사소한 변경이라 할지라도 코드커밋먼트를 완전히 다르게 한다. 커밋먼트에 대한 이러한 전체적인 접근은 매우 중요하다. 이를 통해 일부 구성 요소를 조작하여 부정한 증명을 생성하는 공격 벡터를 효과적으로 방지할 수 있기 때문이다.

코드 제어 그룹: **인덱스 기반 인가 등록부** 코드 제어 그룹은 모든 유효한 코드 커밋먼트를 기록하는 인가 목록으로, 암호학적으로 강제 적용된다. 등록부의 핵심 설계는 평면 집합이 아닌 **인덱스 기반 구조**로 되어 있다.

증명 파이프라인 전반에 걸쳐 모든 프로그램은 고유한 **인덱스**(예. 블록 높이)로 표시되는 특정 컨텍스트에서 실행한다. 코드 제어 그룹은 각 인덱스를 유효한 코드 커밋먼트의 화이트리스트에 매핑한다. 이 데이터 구조는 **머클 마운틴 레인지(MMR) 구조의 계층적 머클 트리**로 구현된다.

이 인덱스 기반 메커니즘은 매우 중요하다. 이를 통해 시스템이 과거 증명 검증 시 사용된 프로그램이 그에 해당하는 특정 인덱스의 화이트리스트에 포함되었는지를 정확하게 판단할 수 있기 때문이다. 이러한 확인은 재귀적 체인 전체 내 모든 프로그램의 무결성을 보장하여, 승인되지 않거나 구식 프로그램 버전을 사용하여 과거 증명을 생성할 가능성을 배제한다. 루트인 코드 제어 그룹은 전체 인가 내역에 대한 커밋먼트 역할을 하므로, 등록부가 변조될 경우 이를 모두 등록부에 표시한다.

안전한 업그레이드 경로 시스템의 업그레이드 경로는 개별 **에포크**(3.5.3 절에서 설명)를 중심으로 구성된다. 변경 사항은 각 전이마다 **에포크 재구성** 이벤트를 통해 도입된다. 이 이벤트는 새로운 시스템 매개변수 집합을 정의한다. 코드 제어 그룹은 이 정의의 핵심 구성 요소이며, 데이터는 비트코인과 L2 모두에 제출된다.

파이프라인 구현 및 신뢰 관리를 간소화하기 위해, 증명 파이프라인은 블록에서 이 정보를 스캔하지 않는다. 대신, 주어진 컨텍스트에 대한 적절한 코드 제어 그룹을 직접 구성 입력값으로 수신한다.

이 전체 프로세스의 무결성은 에포크 전이와 **사전 서명** 메커니즘의 동기화를 통해 보장된다. 새로운 코드 제어 그룹은 거버넌스를 통해 최종 확정되며, 그 코드 제어 루트는 새로운 에포크가 활성화되기 전에 사전 서명 메커니즘을 통해 온체인에 동결된다. 이를 통해 증명자가 잘못된 코드 제어 그룹을 사용할 경우, 코드 제어 루트가 해당 에포크의 사전 서명된 값과 일치하지 않아 온체인 검증 과정에서 증명이 거부된다.

이로 인해 zkVM 핵심 논리의 진화에 필요한 투명하고 안전한 프레임워크를 제공한다. 루트인 코드 제어 루트는 시스템의 완전한 인가된 내역 대한 최종 커밋먼트 역할을 하며, 전체 재귀적 증명 체인의 유효성을 암호학적 확실성 및 거버넌스 컨센서서스 기반에 앵커링한다.

4.2.2 4 단계 재귀적 증명 파이프라인

개요 프로토콜의 증명 워크플로는 별도로 이루어진 네 가지 단계로 구성된 순차적 파이프라인이다. 단계별 설명을 하기 전, 관련된 기본 데이터 구조를 이해하는 것이 중요하다. 각 **블록**은 고유한 **블록 번호**를 가지며, 이는 코드 제어 그룹 내에서의 맥락적 검증을 위한 **인덱스** 역할을 한다. 블록들은 집계를 위해 **배치**로 그룹화된다. 배치는 핵심 통제 가능한 시스템 매개변수를 기준으로 구성되며, 이는 블록의 고정 수 또는 특정 시간 간격일 수 있다. 마지막으로, 프로토콜의 타임라인은 시스템 전반의 재구성을 관리하기 위해 **에포크**로 구성된다.

파이프라인의 네 단계는 다음과 같다. **단일 블록 증명, 배치 집계, 배치 재귀, 증명 래핑.** 이 구조는 중첩된 파이프라인으로 작동한다. 각 배치 내에서 증명은 단계별로 순차적으로 진행되며, 상위 수준의 파이프라인은 재귀 단계를 통해 연속적인 배치들을 연결한다. 각 단계는 특정 입력값을 수신하고 zkVM 내에서 계산을 수행하며, 다음 단계로 전달되거나 래핑된 증명에 기여하는 출력을 생성한다. 이러한 재귀적 아키텍처는 증명이 효율적으로 집계되고 압축될 수 있도록 하여 시스템 확장성을 향상시킨다. 다음 도표는 이 중첩된 파이프라인 구조를 나타낸다.

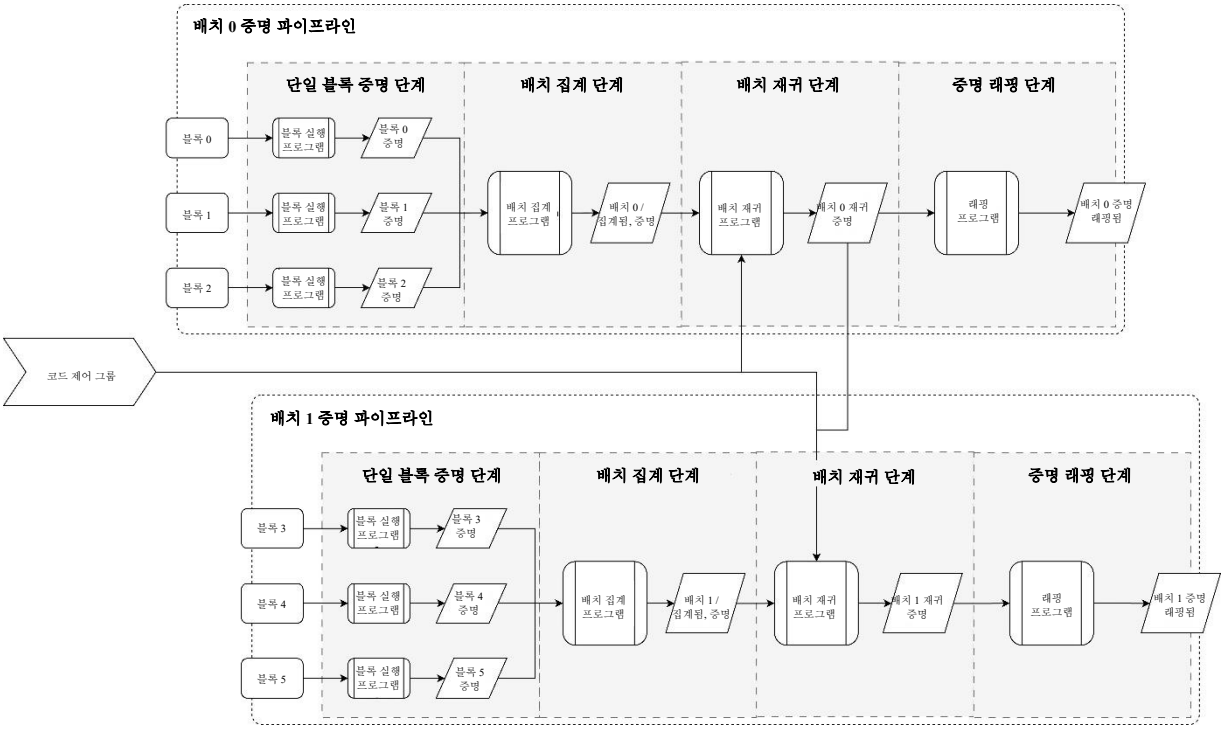

그림 5: 3 개 블록의 크기로 표현된 증명 파이프라인의 아키텍처.

파이프라인 설계 근거 다단계 파이프라인 설계는 증명 효율성, 시스템 복잡성, 보안성 간의 균형을 최적화하기 위해 의도적으로 채택된 아키텍처다.

- **효율성을 위한 1 단계 병렬화**: 단일 블록의 상태 전이 증명은 전체 프로세스에서 가장 계산 집약적인 작업이다. 각 블록의 증명 과정을 분리함으로써, 프로토콜은 배치 형성이 완료되는 것을 기다리지 않고도 증명 작업을 시작할 수 있게 된다. 이를 통해 블록이 생성되자마자 병렬 처리될 수 있어 증명자 자원 활용도를 극대화하고 전체 증명 효율성을 크게 향상시킨다.

- **프로세스 복잡성의 간소화를 위한 2 단계 진행 시점:** 첫 번째 단계에 비해 여러 블록 증명을 단일 배치 증명으로 통합하는 작업은 계산 집약도가 훨씬 낮으며 더 빠른 속도로 이루어진다. 그렇기 때문에 프로토콜은 더 복잡한 증분적 통합 방식이 아닌, 배치 내 모든 블록이 증명된 후에만 통합 단계를 시작하는 방식을 채택한다. 이러한 방식은 명확히 정의되고 작업이 완료된 배치를 집계 작업의 컨텍스트로 사용하게 하여 코드 제어 그룹의 관리 복잡성을 효과적으로 간소화한다.

- **무결성을 위한 3 단계의 업무 분리:** 배치 재귀는 무결성을 보장하고 재귀적 압축을 달성하는 데 핵심적인 역할을 한다. 집계 프로세스와 분리하는 이유는 책임 분담을 명확하게 하기 위해서다. 집계 단계는 “배치 내” 상태 연속성에 초점을 두는 반면, 재귀 단계는 “배치 간” 연결을 처리하여 현재 배치의 유효성을 체인의 전체 내역과 연결한다. 또한, 배치 재귀의 증명 작업은 비교적 더 발전되어 있어 집계와 동시 실행될 필요가 없다.

즉, 전체 시스템은 상위 수준에서 파이프라인 형태로 증명 과정을 연결하여 단일 배치 내에서뿐만 아니라 여러 배치에 걸쳐 실행한다. 이 중첩된 파이프라인 설계는 코드 제어 그룹 내 검증 로직을 관리 가능한 범위 내로 유지하는 동시에 높은 효율성을 제공하여, 시스템의 보안성과 업그레이드 편의성을 보장한다.

4.2.3 1 단계: 단일 블록 증명

이 단계의 목표는 단일 블록 실행의 유효성 증명을 생성하는 것이다. 이 과정은 두 단계로 이루어진다.

1. **오프체인 시뮬레이션:** 증명 생성이 시작되기 전, 시스템은 먼저 zkVM 외부에서 블록 실행을 시뮬레이션한다. 이 단계의 목표는 **읽기-쓰기 집합**과 해당 **머클 증명** 등 증명에 필요한 모든 입력 데이터의 취득하는 것이다.

2. **무상태 증명 생성:** 이전 단계에서 취득한 입력 데이터는 무상태 zkVM 인스턴스에 제공된다. zkVM 은 폐쇄된 환경에서 상태 전이를 재실행하고 제로 지식 증명을 실행한다.

이 단계의 출력값은 **단일 블록 증명(Single Block Proof)**으로, 이는 블록 실행의 정확성을 입증한다. 단일 블록 증명은 핵심적으로 블록 실행 전후의 상태 루트(FromState 및 ToState), 고유 블록 번호(BlockNumber), 그리고 증명을 생성하는 데 사용된 프로그램의 커밋먼트(CodeCommitment)를 요약한다.

4.2.4 2 단계: 배치 집계

이 단계의 목표는 배치 내 연속된 여러 블록의 **단일 블록 증명**을 단일 증명으로 통합하는 것이다. 이 단계는 두 가지 핵심 기능을 수행한다.

1. **상태 연속성 검증:** 인접 블록 증명 간 상태 전이가 연속적인지 확인 및 보장한다. 즉ㅡ 블록 N의 ToState 가 블록 N+1 의 FromState 와 동일한 지를 확인하는 것이다.

2. **프로그램 커밋먼트 기록:** 이 단계에서는 각 입력 증명에서 CodeCommitment 를 추출하여 **기록한다**. 이 단계의 핵심은 커밋먼트만 기록한다는 점이다. 코드 제어 그룹에 대한 검증은 다음 단계로 연기된다.

이 단계는 전체 배치 실행의 정확성을 나타내는 **배치 집계 증명(Batch Aggregated Proof)**를 출력한다. 이 증명은 배치의 전반적인 상태 전이를 요약하며, 이는 배치의 블록 높이 범위(첫 BlockNumber 와 끝 BlockNumber), 배치 내 첫 번째 블록의 초기 상태 루트(FromState), 마지막 블록의 최종 상태 루트(ToState), 그리고 배치 내 모든 증명에 기록된 프로그램 커밋먼트(CodeCommitment)의 완전한 목록을 포함한다.

4.2.5 3 단계: 배치 재귀

지속적으로 증가하는 증명 체인을 일정 크기의 증명 하나로 압축하도록 설계된 파이프라인의 핵심 재귀 단계다.

이 단계의 입력은 두 부분으로 구성된다.

1. 현재 배치의 **배치 집계 증명**(2 단계에서 생성된 배치).

2. 이전 배치의 **배치 재귀 증명**(이전 3 단계 실행에서 생성된 배치). 시스템의 첫 번째 배치에는 이 입력이 존재하지 않는다. 대신, 해당 집계 증명의 초기 상태 루트(FromState)를 프로토콜의 제네시스 스테이트(GenesisState)와 직접 비교하여 전체 증명 체인의 시작점을 앵커링한다.

이 단계에서 zkVM 이 수행하는 핵심 작업은 실행된 모든 프로그램 버전의 인가 내역을 코드 제어 그룹에 비교하여 포괄적으로 검증하는 것이다. 이 검증은 각 프로그램이 실행된 특정 컨텍스트(BlockNumber)를 기반으로 하며, 다음 점검에 반영된다.

- **블록 실행 프로그램 검증:** 배치 내 각 **단일 블록 증명**마다 해당 블록 번호를 인덱스로 사용하여 해당 코드 커밋먼트를 코드 제어 그룹과 비교하여 검증한다.
- **집계 프로그램 검증:** 현재 집계 로직(2 단계)의 코드 커밋먼트를 실행 컨텍스트의 코드 제어 그룹과 대조하여 검증한다.
- **재귀 프로그램 검증:** 이전 재귀 증명에 사용된 프로그램의 코드 커밋먼트 또한 해당 과거 컨텍스트를 사용하여 코드컨트롤그룹과 대조 검증된다.

이를 통해 신뢰 체인은 제네시스 상태부터 각 재귀 단계를 거쳐 올바르게 전파된다. 이 단계의 출력값은 새롭고 업데이트된 **재귀적 증명**으로, 현재까지 처리된 모든 배치의 내역을 압축하여 포함한다.

4.2.6 4 단계: 증명 래핑

이 단계는 배치 증명 파이프라인의 종점으로, 이전 단계의 재귀적 증명을 비트코인 네트워크에서 최종 검증에 적합한 고도로 최적화된 래핑된 증명으로 변환하는 것을 목표로 한다.
이 단계에서는 가장 최근의 재귀적 증명을 **Groth16** 래핑 증명 형태로 "래핑"한다. Groth16 증명 시스템은 극히 작은 크기의 증명을 생성할 수 있으며, 신속한 검증을 지원하기 때문에 이에 채택되었다. 이러한 기능은 비트코인 스크립트의 계산 및 비용 제약 하에서 효율적인 검증을 달성하는 데 핵심적인 역할을 한다.

래핑된 증명 생성 과정에서는 재귀 증명 입력값의 프로그램 버전을 코드 제어 그룹에 비하여 검증한다. 이를 완료한 코드 제어 그룹은 그 머클 루트인 코드 제어 그룹으로 압축되며, 이는 래핑된 증명의 공개 입력으로 포함된다. 래핑 프로그램은 신뢰 계층 구조의 최상위 프로그램이기 때문에, 그 자체는 코드 제어 그룹에 포함되지 않는다. 대신, 그 무결성은 코드 커밋먼트를 온체인 검증 스크립트에 직접 하드코딩함됨으로써 보장된다.

4.3 비트코인 스크립트를 통한 온체인 검증

온체인 검증 스크립트 템플릿은 신뢰의 최종 심판자 역할을 한다. 에포크 라이프사이클과 동기화된 거버넌스 및 사전 서명 프로세스(4.4.1.1 절 참고)의 결과로, 각 에포크에 대한 일련의 불변 신뢰 앵커가 스크립트에 하드코딩된다. 이러한 앵커는 다음을 포함한다.

- 상태의 시작점을 앵커링하는 제네시스 상태(GenesisState).
- 해당 에포크의 모든 업그레이드 가능 프로그램에 대한 전체 인가 내역을 커밋하는 코드 컨트롤 루트(CodeControlRoot).

- 최종 검증자 역할을 하는 업그레이드가 불가능한 래핑 프로그램(4 단계)의 코드커밋먼트 (CodeCommitment).

검증 과정에서는 래핑된 증명 자체와 공개 입력값(FromState, ToState, CodeControlRoot 포함)가 동적 데이터로 제공된다. 온체인 스크립트의 핵심 논리는 증명에서 공개 입력으로 제공된 CodeControlRoot 가 해당 에포크 스크립트에 하드코딩된 CodeControlRoot 와 정확히 일치하는지 **확인**하는 것이다.

이를 바탕으로 스크립트는 이러한 동적 입력과 정적 앵커를 결합하여 포괄적인 ClaimHash 를 형성한다. 이후 BitVM 프로토콜은 낙관적으로 ZK 증명 검증 로직을 호출한다. 검증에 성공하면 다음 사실을 확인한다.

1. FromState 에서 ToState 로의 상태 전이가 zkVM 프로그램이 강제하는 규칙에 따라 계산적으로 유효한지.
2. 재귀적 증명 내 모든 프로그램이 CodeControlRoot 에 의해 승인되었는지.
3. 래핑된 증명이 승인된 래핑 프로그램에 의해 생성되었는지.
4. 전체 계산 내역이 GenesisState 까지 추적 가능한지.

5 비트코인과 Bitlayer 네트워크 간의 브리징

안전한 롤업은 그에 부합하는 안전한 L1 과 L2 간의 자산 이체 메커니즘을 필요로 한다. 이 장에서는 비트코인과 Bitlayer 네트워크 간의 자산 이체 메커니즘인 Bitlayer 자산 브리지에 대해 상세히 알아본다. 이 브리지는 Bitlayer 의 정산 프로토콜과 동일한 BitVM 패러다임을 기반으로 구축되어 상태 유효성과 자산 보관에 대한 통합된 보안 모델을 보장한다.

5.1 역할

브리지 프로토콜은 다음과 같은 주요 역할을 포함한다.

1. **사용자:** 비트코인과 Bitlayer 네트워크 간의 전송을 시작하는 자산 보유자이다.
2. **중개인:** 사용자의 입출금 준비를 지원한다. 이러한 준비에는 초기 트랜잭션 그래프 구성과 검증자로부터 서명을 받는 작업이 포함된다. 중개인은 원활한 상호작용을 위해 사용자와 직접 교류하면서 BitVM 프로토콜의 복잡성을 추상화한다.
3. **증명 위원회:** 롤업 프로토콜에서의 검증자 집합과 동일한 집합이다. 특정 에포크 N 에 선출된 증명 위원회는 해당 에포크에서 시작된 모든 브리지 요청에 대한 트랜잭션 그래프에 사전 서명할 책임을 갖는다.
4. **감시자:** 프로토콜을 감시하고 악의적인 행위에 이의를 제기하는 비허가형 감시자이다.

5.2 자산 크로스체인 플로우

아래에서는 BTC 를 예시로 자산 입출금의 전체 프로세스를 설명한다.

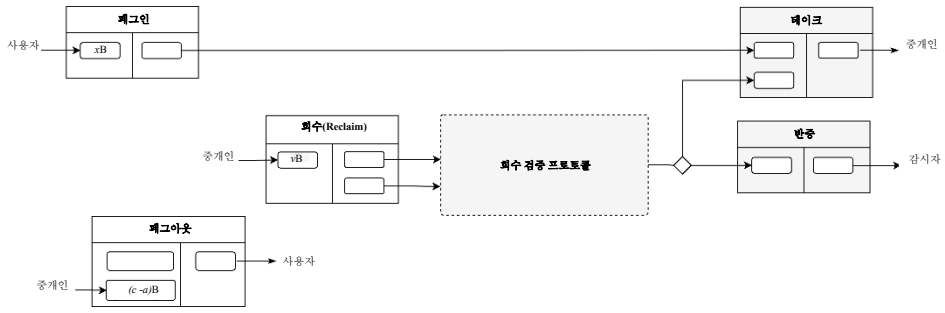

그림 6: 자산 크로스체인 플로우

5.2.1 자산 예금(페그인)

페그인 프로세스는 자산을 비트코인에서 Bitlayer 로 이전하며, 이는 사용자에 의해 여러 단계에 걸쳐 개시된다.

1. **요청 시작:** 사용자는 모든 브로커에게 **페그인 요청(PeginRequest)**을 제출하며, 예치할 UTXO, 대상 Bitlayer 네트워크 주소, 트랜잭션 잔액을 받을 비트코인 주소를 지정한다.

2. **준비:** 각 중개인은 완료된 **페그인 트랜잭션**과 관련 트랜잭션 그래프로 응답한다. 사용자는 수신된 모든 거래 그래프의 정확성을 신중히 검증하여 매개변수가 모든 기대값을 충족하는지 확인해야 한다.

3. **전파(Broadcast) 및 발행(Mint):** 검증 후 사용자는 비트코인 네트워크에서 **페그인 트랜잭션**을 발행한다. 페그인 거래가 L1 에서 확인되면, Bitlayer 네트워크의 **비트코인 라이트 클라이언트**(4 장 참고)가 이를 감지하여 다시 Bitlayer 프로토콜이 자동으로 내재적 트랜잭션을 생성하도록 트리거한다. 이 트랜잭션은 브리지 계약을 호출하여 사용자가 지정한 주소로 동등한 양의 BTC 를 발행한다.

5.2.2 자산 인출(페그아웃)

표준 페그아웃 프로세스는 효율성을 위해 설계되었으며, 신속한 사용자 경험을 위해 중개인이 선제적으로 유동성을 제공하도록 한다.

1. **소각 시작:** 사용자가 Bitlayer 네트워크의 브리지 계약에 **소각 트랜잭션**을 시작한다. 이 트랜잭션은 L2 에서 특정량의 BTC 를 소각한다. 결정적으로, 여기에는 의도된 L1 인출을 정의하는 **부분 서명 비트코인 트랜잭션**(PSBT)이 포함된다. 중개인은 L2 에서 소각된 금액과 PSBT 출력에 명시된 금액의 차액을 수수료로 받는다.

2. **중개인 자금 선지급:** 중개인은 브리지 계약에서 이러한 **소각 트랜잭션**을 모니터링한다. 암묵적 수수료를 수락한 중개인은 사용자가 제공한 PSBT 를 받아 자체 입력값과 서명을 추가하여 완료한 후, 최종 트랜잭션을 비트코인 네트워크에 발행함으로써 요청을 이행할 수 있다. 이 메커니즘은 단 하나의 중개인 선지급 트랜잭션만이 온체인에서 확인될 수 있도록 보장한다.

자금을 선지급한 후, 중개인은 아래에 설명된 보안 메커니즘을 통해 프로토콜로부터 자본을 회수해야 한다.

5.3 중개인 자금 회수

선지급한 자금을 회수하려면, 중개인은 프로토콜에 **킥오프(KickOff) 트랜잭션**을 제출하여 검증 프로세스를 시작해야 한다. 이 제출은 중개인이 유효한 **소각 트랜잭션**을 정당하게 이행했음을 주장하는 역할을 한다. 검증은 제 3 장에서 상태 정산에 사용된 것과 동일한 낙관적 이의 제기-응답 방식을 따르며, 이의가 제기되지 않는 한 중개인의 주장이 올바르다고 가정한다.

- **이의 제기 절차:** 감시자는 이 회수 청구의 정당성을 검증한다. 무효성이 발견될 경우(예. 해당 소각 트랜잭션이 존재하지 않거나 무효인 경우), 감시자는 이의를 제기한다.
- **주장 및 페널티:** 이의 제기를 받을 시, 중개인은 지정된 시간 내에 **Groth16 ZKP** 를 포함한 주장 트랜잭션으로 응답해야 한다. 감시자가 이 증명이 무효임을 검증할 수 있을 경우, 중개인을 처벌하고 담보의 일부를 보상으로 받기 위해 **반증 트랜잭션**을 게시할 수 있다. 이 게임-이론적 프로세스는 3 장에서 설명한 단일 클레임 검증 프로토콜과 구조적으로 동일하다.

회수 검증 메커니즘은 **Bitlayer 라이트 클라이언트**에 기반하며, Bitlayer 네트워크 거래의 최종성을 갖추기 위해 비트코인 메인넷에 준거한다. 그러므로 **소각 트랜잭션**은 배치에 포함되고 비트코인에서 하드 파이널리티를 달성한 후에만 유효한 것으로 간주된다. 이의가 제기될 경우, 중개인이 제공한 Groth16 증명은 **소각 트랜잭션**의 유효성과 진위를 입증하기 위해 Bitlayer 네트워크의 제네시스 상태부터 현재 상태까지의 완전한 검증 체인을 포함해야 한다.

5.4 탈출구

이 브리지는 L2 프로토콜이 중단되는 경우에도 사용자에게 자산에 대한 주권을 보장할 수 있도록 탈출구 포함한다. 운영자가 새 클레임을 제출하는 것을 반복적으로 실패하거나 제출된 클레임에 대한 이의가 받아들여질 경우, 프로토콜이 중단될 수 있다. 이렇게 될 경우, 운영자의 담보는 슬래싱되고 L2 상태는 추가 무효 업데이트로부터 보호된다. 하지만 동시에 사용자 자금이 잠길 수도 있다. 탈출구는 새로운 인출 경로를 제공하며, 이 역시 중개인의 유동성을 활용한다.

프로세스는 다음과 같이 진행된다.

1. **사용자 주도 강제 인출:** 사용자는 강제 포함 인출 거래(force-inclusion withdrawal)를 비트코인 L1 에 직접 전파하여 긴급 인출을 시작한다. 이 트랜잭션은 L2 계정 소유권을 증명하는 서명을 포함하며, 자금을 수령할 L1 주소를 명시한다. 이 L1 트랜잭션은 중단된 롤업에서 완전히 처리될 수는 없지만, 변경 불가능한 온체인 출금 요청으로서의 역할을 한다.
2. **중개인 자금 선지급:** 중개인은 비트코인 L1 에서 이러한 강제 출금 요청을 모니터링한다. 사전 정의된 임계값만큼 요청을 집계하면, 중개인은 유동성을 선지급하여 자금을 사용자가 지정한 L1 주소로 직접 송금할 수 있다.
3. **중개인 자금 회수:** 선지급한 자본을 회수하기 위해 중개인은 단일 Groth16 증명을 포함한 회수 클레임을 브리지 프로토콜에 제출한다. 이 증명은 다음 세 가지 조건을 검증해야 한다.

(a) **L2 중단 증명:** 롤업 프로토콜이 중단되었음을 입증하는 증거. CommitBatchTimeout 트랜잭션(운영자가 새 배치를 제출하지 못했다는 것을 표시) 또는 성공적인 슬래시 트랜잭션(마지막으로 제출된 배치가 사기였음을 표시)의 제시를 통해 확인 가능하다.

(b) **유효한 사용자 요청 증명:** 사용자의 출금 요청이 정당함을 입증하는 증거. 이를 입증하기 위해서는 L1 상의 강제 포함 트랜잭션 존재를 증명(비트코인 라이트 클라이언트 사용)하고, 사용자가 마지막으로 올바르게 최종 확정된 L2 상태에서 충분한 잔액을 보유했음을 증명해야 한다.

(c) **이행 증명:** 중개인이 이미 해당 자금을 L1 상의 사용자에게 송금했음을 증명하는 증거. 비트코인 라이트 클라이언트를 통해 확인 가능하다.

이 탈출구 메커니즘은 사용자가 언제나 자산을 통제할 수 있도록 보장하며, 오직 비트코인 L1 의 보안과 중개인 네트워크의 경제적 인센티브에만을 기반으로 한다. Bitlayer 는 계정 추상화를 더욱 탐색하여, 보다 지능적인 인출 논리를 정의하고, 이 메커니즘을 확장하여 스마트 계약 내 보유 자산의 긴급 인출을 지원하는 방안을 모색할 예정이다.

6 보안 분석

본 장에서는 Bitlayer 롤업의 기반이 되는 보안에 대한 종합적인 분석에 대해 알아본다. BitVM 스타일 스마트 계약에 사용되는 일반적인 보안 모델에 대해 먼저 소개하고, 안전성과 생존성 속성에 대한 정의와 증명에 대해 알아본다. 그 다음, 3 장에서 논의된 비트코인 정산 보안 속성에 대해 상세히 분석한다. 마지막으로는 탈중앙화 네트워크의 본질적인 검열 저항성에 대해 간략히 설명한다.

6.1 BitVM 스타일 스마트 계약 보안

BitVM 스타일 스마트 계약은 보편적인 트랜잭션 그래프 구조를 따른다. 이 절에서는 Bitlayer 정산 프로토콜을 포함한 모든 BitVM 스타일 계약에 적용 가능한 일반적인 보안에 대해 분석한다.

6.1.1 시스템 모델 및 가정

BitVM 스타일 스마트 계약에서는 최소한 세 가지 역할의 협력이 이루어진다.

- **트랜잭션 그래프 제안자:** 제안자는 계약 인스턴스를 시작하는 역할을 담당한다. 이를 위해 제안자는 커밋먼트 겸 부정행위에 대한 담보를 목적으로 사전 정의된 금액의 BTC 를 스테이킹해야 한다.

- **증명자:** Bitlayer 는 n 명의 증명자가 존재하며, 그 중 m 명이 정직한 증명자라고 가정한다. 그 외 $n - m$ 명은 반(半)정직 증명자로 가정한다. 이는 그들이 프로토콜을 따르고 사전 서명된 서명을 구축하기 위해 협력하지만, 프로토콜 외적인 측면으로는 사전 서명 후에도 키를 보유하여 사전 서명 완료 후 추가적인 무인가 트랜잭션에 서명하려는 시도하는 등 예측 불가능한 행동을 할 수 있다는 것을 의미한다. 각 사전 서명에는 최소 $n - m + 1$ 명의 증명자가 참여해야 한다.

- **감시자:** 감시자는 제안자가 제출한 온체인 상태를 모니터링하여 정확성을 보장한다. 부적절한 행동이 감지되면 트랜잭션에 이의를 제기할 수 있으며, 이를 통해 스테이킹된 BTC 에 대한 페널티를 부과함으로써 제안자에게 책임을 지울 수 있다. 이 모델은 합리적이고 정직한 활성 감시자가 최소 한 명 이상 있다는 것을 가정한다.

또한, 참여자와 비트코인 네트워크 간의 모든 통신이 알려진 유한 시간인 Δ 내에 이루어지는 **동기화된 네트워크**를 사용한다고 가정한다. 모든 참여자는 합리적이고 다항식 시간에 제한을 받는다고 가정한다. 이는 BitVM 스타일 스마트 계약에서 사용되는 모든 암호화 도구가 안전하다는 것을 의미한다.

6.1.2 트랜잭션 그래프 모델

트랜잭션 그래프는 BitVM 스타일 스마트 계약의 핵심으로, 방향성 비순환 그래프(DAG) 구조로 되어 있다. 이 모델은 계약 실행에 명확성과 강제력을 부여한다.

- **선행 트랜잭션:** 트랜잭션은 제안자가 스테이킹한 담보금과 감시자의 담보금 등 계약 실행에 필요한 초기 출력을 제공한다. 증명자는 사전 서명 전에 해당 거래의 존재와 정확성을 검증해야 한다.
- **사전 서명 트랜잭션:** 증명자가 사전 서명해야 하는 트랜잭션으로, BitVM 스타일 계약의 논리를 규정한다.
- **싱크 트랜잭션:** DAG 상에서 후속 연결이 없는 트랜잭션으로, 자금의 인출을 의미한다.

6.1.3 설계 원칙

- **스테이크:** 계약을 시작하려면 제안자는 BTC 를 지정된 금액만큼 스테이크해야 한다. (그래프 내 d BTC).
- **슬래시 가능성:** 제안자가 제출한 잘못된 STF 는 스테이킹된 BTC 의 슬래싱으로 이어질 수 있다.
- **종료:** 사전 서명된 트랜잭션 내 금액을 포함하는 모든 출력은 트랜잭션의 종료를 보장하기 위해 싱크 트랜잭션으로 이어지는 타임락 경로(여러 트랜잭션 포함 가능)를 반드시 가져야 한다.

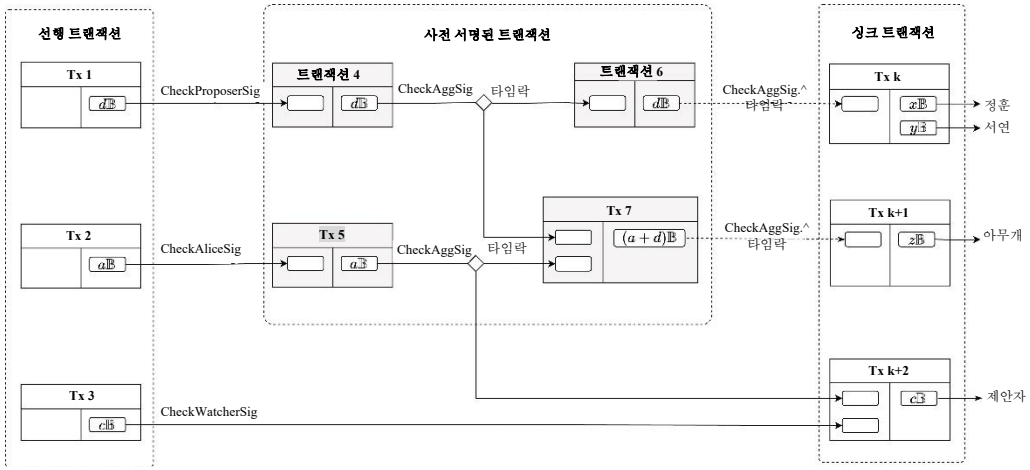

그림 7: 트랜잭션 그래프 DAG 모델

6.1.4 안전

안전 목표

- **유효성:** 트랜잭션 그래프의 모든 트랜잭션은 사전 서명 후에도 유효하게 유지되어야 한다.
- **무결성:** 사전 서명 후에는 트랜잭션 그래프에 새로운 트랜잭션을 추가할 수 없다.
- **유연성:** BitVM 스타일 스마트 계약은 애플리케이션 시나리오에 따라 다양한 보안 가정을 수용할 수 있다.

부명제 1. *$\{tx_1, ..., tx_n\}$을 $utxo_a$를 소비하는 사전 서명된 트랜잭션으로 가정할 경우. 트랜잭션 $tx' \notin \{tx_1, ..., tx_n\}$는 $utxo_a$를 소비할 수 없다.*

증명. 이 중요한 부명제를 모순을 통해 증명해보겠다. 위원회 $\{attester_0, ..., attester_{n-m}\}$ 이 사전 서명을 수행했다고 가정해 보자. tx' 의 존재는 증명자들이 사전 서명 프로세스 외부에 추가 서명을 수행했다는 것을 명시하여, 이 $n - m + 1$ 명의 증명자들이 반정직하다는 것을 나타낸다. 즉, 가정에 모순되는 것이다. □

부명제 2. *각 사전 서명 위원회는 최소한 한 명의 정직한 증명자를 포함해야 한다.*

정리 1(유효성). *트랜잭션 tx 에 대한 유효한 사전 서명 δ 가 생성을 경우 tx 는 유효하게 된다.*

증명. 부명제 2 에 따라, 최소한 한 명의 정직한 증명자 s_i 가 사전 서명에 참여하여 tx 에 대한 부분 서명 σ_i 를 기여하게 된다. 따라서 s_i 가 수신한 tx 는 유효하게 된다. δ 의 유효성은 모든 증명자가 tx 에 부분 서명을 기여하는 데 의존하므로, tx 는 반드시 유효해야 한다. □

정리 2(무결성).

증명. 싱크 트랜잭션을 제외한 모든 출력은 사전 서명 위원회의 다중 서명을 필요로 한다. 부명제 1 에 따라, 모든 참여자는 사전 정의된 경로를 따르는 트랜잭션 그래프의 UTXO 만을 지출할 수 있다는 결론을 내릴 수 있으며, 이를 통해 BitVM 스타일 스마트 계약의 무결성이 보장됨을 확인할 수 있다. □

정리 3(유연성).
증명. 사전 서명 위원회가 최소 한 명의 정직한 노드를 포함하기만 하면, 증명자의 보안 가정은 애플리케이션 시나리오의 요구사항에 따라 동적으로 조정이 가능하다. 이를 통해 부명제 1 과 2 을 기반으로 유효성과 무결성을 추론할 수 있게 된다. □

6.1.5 생존성(Liveness)

생존성 목표

- **자금 유동성:** 거래의 선행 트랜잭션에 포함된 자금은 무기한정 동결되어 있을 수 없다.

정리 4(자금 유동성)**.**

증명. 타임락 기간이 명시되어 있고 유한하므로, 트랜잭션 그래프의 종료 원칙은 결국 모든 자금이 잠금 해제되어 유한 시간 내에 싱크 트랜잭션으로 흘러갈 수 있도록 한다. □

6.2 비트코인 정산 보안

본 절에서는 3.4.4 장에서 소개된 비트코인 정산 보안 속성의 증명을 중점적으로 다룬다.

정리 5(완정성). *프로토콜을 정확하게 따르고 유효한 상태 클레임을 제출하는 정직한 운영자는 페널티를 받지 않는다.*

증명. 정직한 운영자는 유효한 클레임과 하위 프로그램 결과를 요구된 시간 내에 게시하여 불일치가 발생하지 않도록 한다. 그로 인해 어떤 감시자도 *반증 스크립트*를 해제할 수 없으며, 운영자는 처벌받지 않는다. 공간 절약을 위해 세부 사항은 생략한다. □

정리 6(건정성). *정직한 감시자는 언제나 유효한 반증 트랜잭션을 구성할 수 있으므로, 부정한 클레임을 제출한 부정직한 운영자는 페널티를 반드시 받게 된다.*

증명. 부정직한 운영자가 Δ_{claim} 내에 Φ 를 공개하지 않을 경우, 해당 운영자는 벌칙을 받게 된다. 클레임 단계에서 부정직한 운영자가 Φ.를 게시하면 감시자들의 로컬에서 SNARG.Vrfy 가 실패하게 된다. 그러면 감시자들은 $\Delta_{challenge}$ 내에 이의를 제기하여 프로토콜을 이의 제기 단계로 진행시킨다. 운영자가 Δ_{assert} 시간 내에 모든 하위 프로그램의 결과를 게시하지 않을 경우, 해당 운영자는 벌칙을 받게 된다.

감시자가 운영자가 게시한 하위 프로그램 실행과 모순되는 입력과 출력으로 *반증 스크립트*를 해제할 수 있는 반증 알고리즘이 존재할 경우, 부정직한 운영자는 반드시 페널티를 받게 된다.

반증 알고리즘의 존재에 대한 증명은 다음과 같다. 우선, 운영자가 게시한 입력과 출력이 로컬 하위 프로그램 실행과 모순되므로, 하위 프로그램에 의해 생성된 불일치 출력이 최소 한 개 이상 존재할 것이다. 이를 f'라 하겠다. 이 f'의 입력이 일관되는 지를 확인해 보겠다. 모든 입력이 일관되었다면 f'을 이의 제기 대상 하위 프로그램으로 선택한다. 일관되지 않은 경우, 불일치 입력 중 하나에 대해 첫 번째 단계를 재귀적으로 실행한다. 이로써 반증 알고리즘은 이의를 제기할 하위 프로그램을 성공적으로 선택할 수 있는 것이다. □

정리 7(효율성)**.** *전체 클레임 검증 프로세스는 승인 또는 거절 여부와 관계없이 프로토콜의 타임 락에 의해 정의된 제한된 시간 내에 종료된다.*

증명. 프로토콜의 각 단계에는 제한된 시간이 존재한다. 운영자와 감시자 모두가 프로토콜을 정직하게 따를 경우, 주어지는 낙관적 시간 제한은 $\Delta_{claim} + \Delta_{challenge}$이다. 운영자나 감시자 중 어느 한 쪽이라도 프로토콜을 망치려 할 경우, 주어지는 시간 제한은 $\Delta_{claim} + \Delta_{assert} + \Delta_{disprove}$이다. 따라서 확인에 주어지는 최대 시간 제한은 $\Delta_{claim} + \max\{\Delta_{challenge}, \Delta_{assert} + \Delta_{disprove}\}$이다. 그렇기 때문에 프로토콜은 클레임의 수용 여부와는 상관없이 언제나 종료된다. 이처럼 본 프로토콜은 설계상 효율성을 보장한다. □

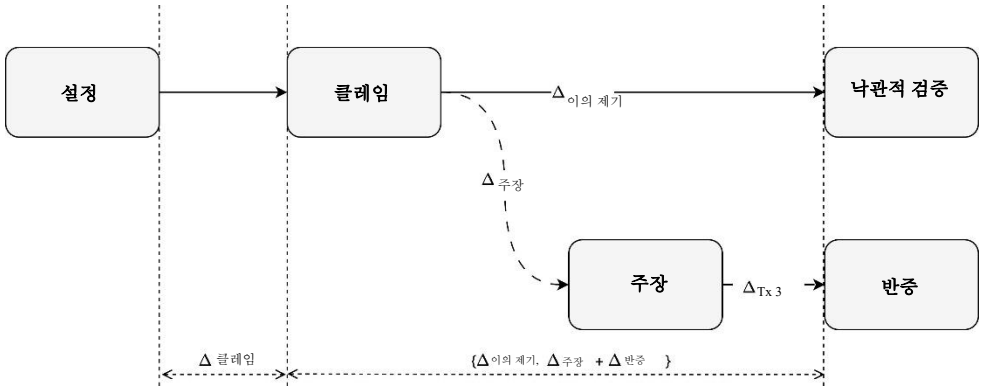

그림 8: 비트코인 정산 보안

6.3 검열 저항

기존 L2 아키텍처가 단일 시퀀서에 의존하는 것과 달리, Bitlayer 의 설계는 검증자 간 순환 방식을 통해 블록을 생성한다. 이러한 탈중앙화 시퀀싱 메커니즘은 어느 한 주체도 트랜잭션을 일방적으로 검열할 수 없도록 보장한다. 블록 생산이 검증자 간 비허가 및 스테이크 가중치 방식으로 순환되므로, 유효한 거래를 배제하려는 시도는 후속 블록에서 우회된다. 이러한 방식은 강력한 검열 저항성을 내재화하고 네트워크 중립성을 강화한다.

7 시스템 아키텍처

탄탄하게 잘 설계된 아키텍처는 Bitlayer 의 두 목표인 고성능과 신뢰 최소화 보안을 달성하기 한 필수 조건이다. 우리 설계의 핵심은 L2 의 고처리량 실행 계층을 L1 정산 및 보안 계층으로부터 깔끔하게 분리하는 이중 하위 시스템 모델이다. 본 장에서는 그 핵심 원칙인 분리를 살펴보고, 이어서 데이터 흐름에 대한 시스템 수준 개요, 그리고 검증자(성능) 하위 시스템과 롤업(보안) 하위 시스템 내 구성 요소에 대한 상세 분석을 통해 Bitlayer 의 아키텍처에 대해 상세히 알아본다.

7.1 L2 실행과 L1 정산의 분리

Bitlayer 아키텍처의 핵심 설계 원칙은 L2 실행과 L1 정산의 분리다. 이 전략적 분할은 시스템 기능을 검증자(성능) 하위 시스템과 롤업(보안) 하위 시스템, 두 가지 영역으로 구분한다.

- **검증자 하위 시스템:** 이 하위 시스템은 트랜잭션 순서 지정, 스마트 계약 실행 및 상태 저장을 포함한 L2 트랜잭션(L1 강제 트랜잭션 포함) 처리에 전적으로 초점을 맞춘 고성능 블록체인으로 이루어져 있다. 이는 높은 처리량과 낮은 지연 시간을 위해 설계되어, 사용자에게 1 초 미만의 소프트 파이널리티를 제공한다. 이 하위 시스템의 성능은 L1 상호작용과 무관하게 자체 합의 및 계산 기술에만 의존한다. 그러므로, 고빈도 상태 계산을 위해 설계된 이 하위 시스템을 **성능 영역(Performance Domain)**이라 칭한다.

- **롤업 하위 시스템:** 이 하위 시스템은 검증자 하위 시스템의 상태와 보안을 비트코인 네트워크에 앵커링한다. 이는 상태 커밋먼트 및 사기 방지 증명의 이의 제기-응답 프로토콜을 포함한 모든 L1 상호작용을 담당한다. 이 하위 시스템의 보안성은 암호학적으로 검증 가능한 방식을 통해 비트코인의 PoW 합의으로 뒷받침된다. L1 정산 프로세스의 무결성을 보장하는 것이 목적이기 때문에, 이를 보안 영역(Security Domain)이라 칭한다.

마이크로서비스 아키텍처와 유사하게, 이러한 목적 분리는 각 하위 시스템이 독립적으로 개발되고 최적화될 수 있게끔 한다. 예를 들어, 합의 엔진 업그레이드와 같은 검증자 하위 시스템의 최적화의 경우 L1 상호작용 프로토콜에 대한 변경 없이도 가능해진다. 반대로 롤업 하위 시스템은 핵심 L2 실행 계층의 재설계 없이도 새로운 증명 시스템을 통합할 수 있게 된다. 이러한 모듈식 설계는 견고한 시스템을 구현하면서도 이에 미래 기술 변화를 적용시킬 수 있는 가능성을 열어 둔다.

7.2 검증자 하위 시스템

검증자 하위 시스템은 전적으로 성능을 위해 설계된 시스템이다. 이 아키텍처는 **탈중앙화 시퀀서, 병렬 실행 엔진, 고동시성 데이터 저장소**, 이 세 가지 핵심 구성 요소로 이루어져 있다. 이러한 구성 요소들은 **능동적 계산 파이프라인**을 통해 통합되며, Bitlayer 의 트랜잭션 처리량과 저지연 특성의 주요 동력이 되어 **초당 수만 건의 트랜잭션**(TPS)을 지원할 수 있는 역량을 제공한다.

7.2.1 탈중앙화 시퀀서

이 시퀀서는 Bitlayer 플랫폼의 순서 지정 및 합의 핵심으로 기능한다. 이는 PoS 프로토콜 하에서 운영되는 검증자들로 구성된 탈중앙화 네트워크를 통해 구현된다. 주요 설계 목표는 트랜잭션 순서 지정을 위한 **무신뢰, 신뢰 중립적** 메커니즘을 제공하는 것으로, 이를 통해 단일 장애점, 악의적인 트랜잭션 순서 재지정(예: MEV 추출), 검열과 같은 중앙화 시퀀서에 내재된 위험을 완화하는 것이다. 시퀀서 네트워크는 네트워크 전반에서 트랜잭션을 수신하고 고성능 BFT 합의 프로토콜을 활용하여 블록 내 트랜잭션에 대한 캐노니컬 및 전역적 순서 지정을 수행하여, 이는 경제적 지분 기반의 **소프트 파이널리티**를 달성한다.

7.2.2 병렬 실행 엔진

이 실행 엔진은 EVM 의 순차적 실행 모델의 성능적 한계를 극복하도록 설계되었다. EVM 실행을 병렬화의 가장 큰 난점은 상태 접근 패턴이 교차되어 있어 의존성을 예측하는 것이 어렵다는 것이다. Bitlayer 의 핵심 혁신은 이 환경에 특별히 최적화된 종속성 분석과 상태 충돌 해결 메커니즘에 있다. 이 혁신은 블록체인 환경에 적용된 **낙관적 동시성 제어 원칙**을 활용한다. 엔진은 데이터 종속성이 감지될 때만 조정 메커니즘을 가동하며, 충돌이 발생하지 않는다는 가정 하에 트랜잭션을 병렬로 추측 실행한다.

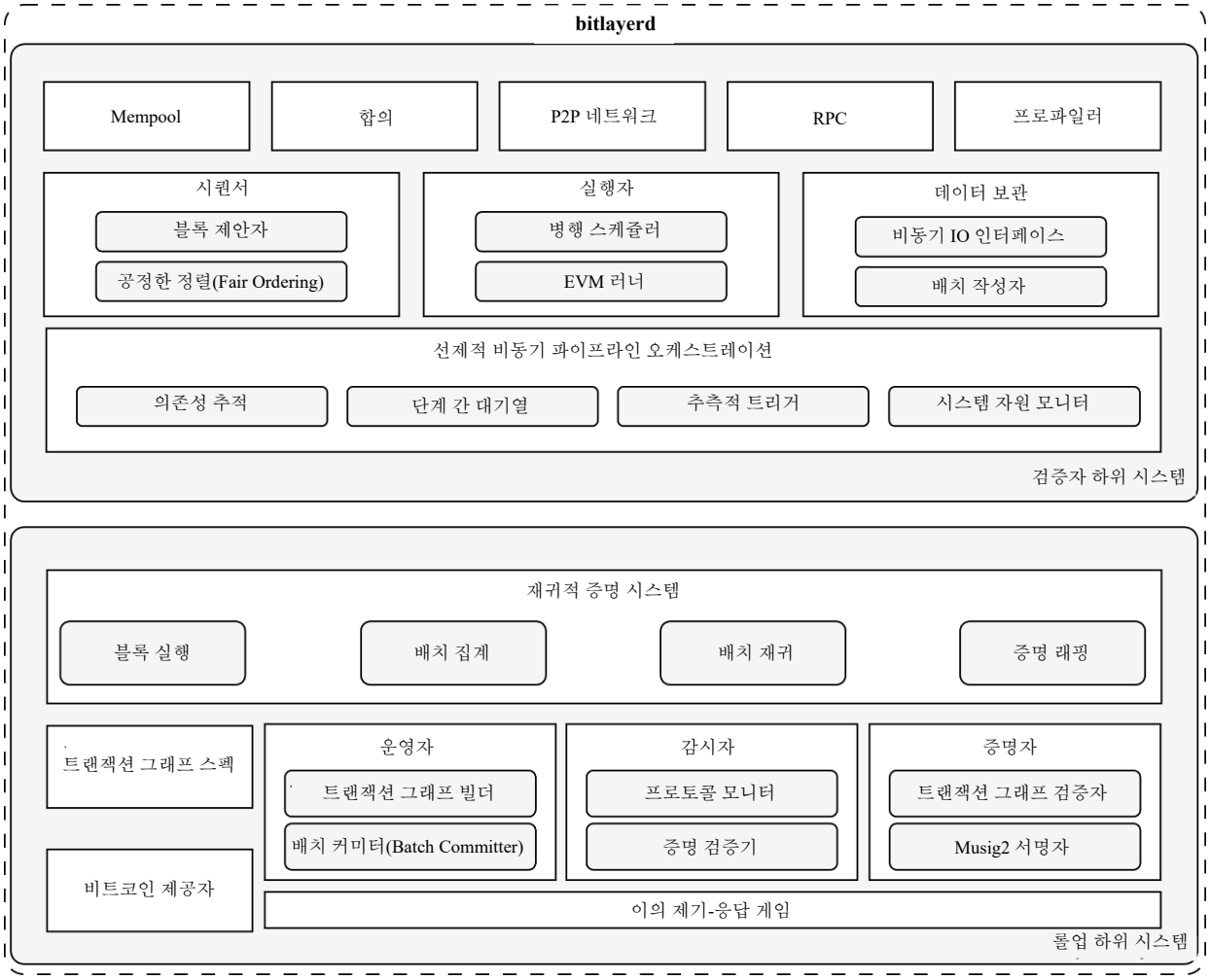

그림 9: 듀얼 하위 시스템 아키텍처의 고수준 개요.

여기에는 **작업 수준 충돌 해결** 및 **힌트 기반 사전 스케줄링**을 포함한 고급 기술이 적용되었다. 이를 통해 엔진은 전체 트랜잭션이 아닌 최소한의 충돌 작업 집합만 재실행하여, 다중 코어 프로세서 아키텍처의 활용도를 극대화하고 트랜잭션 처리 용량을 크게 향상한다.

7.2.3 블록체인 네이티브 스토리지 엔진

이 데이터 저장소는 병렬 실행 엔진의 접근 패턴에 맞춤화된 고성능 영속성 계층이다. 이는 일반 목적 데이터베이스(예. LevelDB/RocksDB)의 상태 증가 및 I/O 경합과 같은 병목 현상에 대한 해결을 제공한다. 또한, 이 저장소는 계정 상태, 스마트 계약 코드, 트랜잭션 영수증을 포함한 모든 캐노니컬 블록체인 데이터를 저장한다. **비동기 I/O 인터페이스**와 **배치 쓰기**를 통해 성능을 확보하며, 이를 통해 실행과 저장 지연 시간을 분리한다. 구조적으로는 **인덱스와 키-값(KV) 분리**와 **의미 인식 데이터 분할**을 채택한다. 이 설계는 락 경합을 최소화하고 쓰기 증폭을 감소하여, 병렬 실행 엔진의 높은 동시성 요구에 대한 강력한 저장소 지원을 제공한다.

7.2.4 능동적 계산 파이프라인

검증자 하위 시스템은 구성 요소를 비동기적 다단계 파이프라인에 통합하여 현대 CPU 의 비순차적 실행과 유사한 **능동적 계산 원칙**을 구현한다. 경직된 순차적 패러다임(합의 -> 실행 -> 지속 -> 체크포인트)을 채택하는 대신, Bitlayer 파이프라인은 이러한 작업을 분해하여 시간적 중첩을 허용한다. 이를 통해 시스템은 각자 다른 단계에서 여러 **블록**을 동시에 처리할 수 있게 된다. 예를 들어, **블록 N+1** 의 트랜잭션 순서를 지정하는 합의 프로세스는 **블록 N** 의 실행과 동시에 진행될 수 있는 것이다. 이러한 합의와 실행의 중첩은 자원 활용도를 극대화하고 종단 간 거래 지연 시간을 유의미하게 줄이는 데 핵심적인 역할을 한다.

7.3 롤업 하위 시스템

이 절에서는 검증자가 비트코인 네트워크 상에서 계산 결과를 검증이 가능하도록 정산하는 구성 요소를 상세히 기술한다. 이를 통해 비트코인 블록체인은 선제적 상태 검증 및 분쟁 해결 기능을 제공한다.

7.3.1 재귀적 증명 시스템

Bitlayer 보안은 **재귀적 증명 시스템**에 암호학적 기반을 둔다. 이 시스템은 비동기식 제로 지식 증명(ZKP) 생성기로 구현되며, 정산 기간 내 모든 상태 전이에 대해 단일하고 간결하며 반박 불가능한 유효성 증명을 생성한다. 이 설계의 핵심 혁신은 **L2 의 핵심 성능 경로와 이 시스템이 완전히 분리된다는 점**에 있다. 증명 생성은 백그라운드 프로세스로 작동하여 L2 블록 생성 속도와 트랜잭션 확인 지연 시간이 계산 집약적인 증명 프로세스와 독립적으로 유지될 수 있도록 한다. 이러한 아키텍처는 비허가 **증명 시장**의 향후 개발을 용이하게 하여, 시장 기반 경쟁을 통해 증명 효율성과 비용을 더욱 최적화할 수 있도록 한다.

7.3.2 L1 완결성 프로토콜의 엔지니어링 구현

복잡한 BitVM 기반 프로토콜을 견고한 자동화 시스템으로 전환시키는 것은 롤업 하위 시스템의 주요 엔지니어링 과제이다. 모듈식 소프트웨어 설계는 프로토콜 역할을 특정 구성 요소에 매핑하여 시스템 유지 관리성, 보안성 및 확장성을 보장한다.

이는 **운영자, 감시자, 증명자,** 이 세 가지 주요 소프트웨어 독립체를 중심으로 구현된다.

- **운영자:** 운영자는 프로토콜을 진행시키는 주요 주체이다. 운영자는 다음과 같은 여러 내부 모듈로 구성된 자동화 소프트웨어 스위트이다.
  - 트랜잭션 그래프 빌더: 이 모듈은 BitVM 패러다임의 핵심 구현체다. 이는 비트코인 트랜잭션 그래프 사양을 엄격히 준수하여 L2 상태 배치로부터 복잡한 비트코인 트랜잭션 그래프를 결정론적으로 구축한다. 트랜잭션 그래프 빌더에는 모든 이의 제기-응답 경로에 대한 비트코인 스크립트를 프로그래밍으로 생성 및 해시 락을 통해 L2 상태 루트를 임베딩하는 과정이 포함된다.
  - 배치 커미터: 이 모듈은 여러 L2 배치의 상태 커밋을 단일 **ClaimTransaction** 으로 집계하여 L1 상호작용 비용을 최적화하고 온체인 발자국을 분산한다.

- 이의 제기 응답기: 이 방어 모듈은 L1 에서 그 모듈 커밋에 대한 이의 제기를 모니터링한다. 이의가 감지될 시, 증명 시스템에서 해당 영지식 증명을 검색하고 사양에 정의된 적절한 응답 트랜잭션을 전파한다.

- **감시자:** 감시자는 탈중앙화 보안 메커니즘을 대표하는 요소로, 모든 풀노드 참여자가 운영자를 감사하기 위해 실행할 수 있는 소프트웨어다.
  - 증명 검증기: 이 모듈은 L1 에서 상태 커밋먼트와 그에 해당하는 L2 블록 데이터를 독립적으로 가져오는 역할을 한다. 운영자가 제출한 상태 루트의 무결성을 검증하기 위해 상태 전이를 재실행하여, 사기에 대한 첫 번째 방어선이 된다.
  - 이의 제기자: 증명 검증자가 불일치를 감지하면, L1 에서 이의 제기 트랜잭션을 구성 및 전파하기 위해 이의 제기자 모듈이 활성화된다. 이를 통해 온체인 분쟁 해결 절차가 시작되며, 운영자가 스테이킹한 담보에 이의가 제기된다.

- **증명자:** 트랜잭션 그래프의 유효성을 보장하는 고액 스테이킹 검증자 집합이다.
  - 트랜잭션 그래프 검증자: 서명 전 각 증명자는 이 모듈을 통해 운영자가 구축한 트랜잭션 그래프를 공식 사양에 따라 독립적으로 검증하여 악용 가능하거나 무효한 경로가 포함되지 않도록 한다.
  - Musig 서명자: 검증 성공 시, 이 모듈은 고급 다중 서명 방식(예. MuSig2)을 사용하여 트랜잭션 그래프에 대한 단일 통합 서명을 생성한다. 이러한 엔지니어링은 기존 CHECKMULTISIG 작업 대비 뛰어난 효율성, 개인정보 보호 및 확장성을 제공한다.

이러한 구성 요소들은 비트코인 제공자 모듈(RPC 래퍼)을 통해 L1 과 인터페이스하여, 명확한 업무 분리를 갖춘 자동화 L1 완결성 시스템을 형성한다. 이 모듈식 구현은 테스트 가능성과 유지보수성을 향상하며, 향후 프로토콜 업그레이드를 위한 견고한 기반을 제공한다.

7.4 트랜잭션 라이프사이클

- **시퀀싱:** 사용자에 의해 L2 네트워크에서 트랜잭션가 시작된다(1a 단계). 트랜잭션이 검증자 사이에서 시퀀서 구성요소에 의해 수집되며, 예금 및 강제 포함 트랜잭션과 같은 온체인 상호작용(1b 단계)이 비트코인에 제출된다. **비트코인 리스너** 구성 요소가 비트코인 네트워크를 지속적으로 모니터링하며 L1 트랜잭션을 시퀀서에 동기화한다.

- **블록 합의:** 트랜잭션이 수집되면 시퀀서 중 하나가 트랜잭션 집합을 제안하도록 선택된다. 모든 검증자는 이후 합의 과정을 통해 제안된 트랜잭션의 순서와 내용에 대해 합의를 이룬다(2 단계). 합의가 달성되면 순서가 지정된 L2 블록이 생성된다(3 단계).

- **실행:** 블록은 생성되는 즉시 검증자에 의해 실행된다. 실행기는 블록 내 거래를 병렬로 실행하여 새로운 월드 상태(world state)를 산출한다.

- **영속성:** 새로운 월드 상태는 효율적으로 영속된다. 이 시점에서 L2 블록은 **소프트 파이널리티**를 달성한다(4 단계).

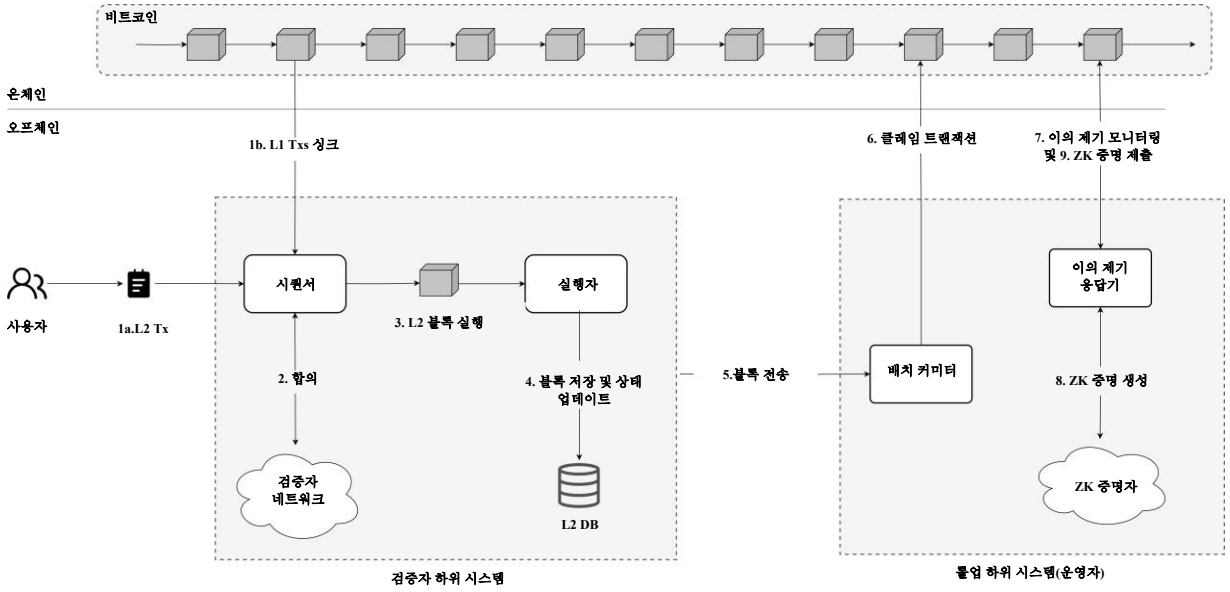

그림 10: 상세 트랜잭션 라이프사이클

- **하위 시스템 간 핸드오프:** 검증자 네트워크 내 **소프트 파이널리티**가 달성되면, L2 블록의 상태 데이터는 내부 비동기 API 를 통해 롤업 하위 시스템으로 전송된다. 이 비동기 통신은 성능을 격리하는 데 매우 중요하며, 검증자 하위 시스템의 높은 처리량이 롤업 하위 시스템의 증명 생성 지연(단계 5)으로 인해 방해받지 않도록 보장한다.

- **상태 클레임:** 운영자의 배치 커미터 구성 요소는 여러 블록 데이터를 **집계**하고, 롤업 배치 정보를 조립하며, 3 장에서 설명된 대로 비트코인 정산 프로세스를 시작한(6 단계).

- **이의 제기 응답기:** 감시자가 운영자가 비트코인에 게시한 상태 주장에 이의를 제기하면(7 단계), 운영자는 주장된 상태의 정확성을 방어하기 위해 영지식 증명을 생성하여 응답한다(8 단계). 증명자(Prover)는 실행 추적 기록과 상태 전환을 기반으로 영지식 증명을 생성하며, 이는 롤업 상태의 유효성을 입증하는 증거로 **운영자에 의해 비트코인에 제출된다**(단계 9).

7.5 결론

궁극적으로 Bitlayer V2 아키텍처는 비트코인을 궁극적인 탈중앙화 신뢰 및 정산 계층으로 활용하는 동시에, Bitlayer 네트워크를 통해 이를 기반으로 많은 처리량을 검증할 수 있는 계산 계층 기능을 구현하여, 비트코인 확장성을 위한 명확한 청사진을 제시한다. 이 설계는 비트코인 생태계 내 방대한 잠복 자본을 활용할 수 있는 실행 가능한 경로를 제공하며, 비트코인 기반의 안전하고 확장 가능하며 활기찬 탈중앙화 금융 생태계를 위한 기초 인프라를 구축한다.

8 한계점과 향후 방향

본 장에서는 Bitlayer 네트워크의 현재 설계를 검토하고, 그에 내재된 상충 관계와 이를 통해 제시되는 촉망되는 연구 방향에 대해 논의한다. 먼저 현재 Bitlayer 프로토콜의 주요 한계를 전반적으로 살펴본 다음, 이러한 과제를 해결하고 비트코인 L2 의 역량을 더욱 발전시키기 위한 향후 작업에 대해 살펴보도록 하겠다.

8.1 한계점

Bitlayer 네트워크는 비트코인 계산 계층에 필요한 견고한 프레임워크를 제공하지만, 현재로서는 다음과 같은 몇 가지 설계 상 상충 관계가 존재한다.

1. **검증자 집합의 정직성에 대한 의존성:** 현재 브리지 및 정산 프로토콜의 보안은 증명 위원회 역할을 수행하는 활성 검증자 집합 내 정직한 다수 가정에 의존한다. 보안성이 암호학적으로 강제되기는 하지만, 이는 비트코인의 PoW 를 넘어선 신뢰 가정을 요구한다. 향후 비트코인 프로토콜 업그레이드를 통해 이러한 외부의 정직한 다수에 대한 의존성을 제거하는 것은 더욱 완전한 무신뢰 시스템의 구축에 있어 여전히 핵심적인 목표로 남아있다.

2. **중앙화된 운영자 및 생존성:** 현재 모델은 시퀀싱 및 정산에 단일 순환 롤업 운영자 방식을 채택한다. 이는 효율적이지만, 운영자가 오프라인 상태가 될 경우, 생존성 관련 잠재적 단일 장애 지점을 초래할 수 있다. 그렇기 때문에 다중 운영자 메커니즘을 개발이 필요할 것으로 판단된다.

3. **중개인 유동성에 대한 의존:** 신속한 인출 및 비상 탈출 메커니즘은 활발한 제 3 자 중개인의 네트워크가 제공하는 선행 유동성에 의존한다. 이러한 의존도를 낮추는 프로토콜 네이티브 솔루션을 사용한다면 시스템의 사용자 경험과 자본 효율성을 더욱 개선할 수 있을 것이다.

8.2 향후 방향성

Bitlayer 는 이러한 한계를 해결하고 네트워크 역량을 확장하기 위해 여러 개선 방안을 적극적으로 연구를 계속하고 있다.

1. **미래 비트코인 업그레이드 활용(약정):** 새로운 약정 명령 코드(예. OP CTV[10], OP CAT 또는 유사 제안)를 도입하는 등 향후 이루어질 수 있는 바트코인 프로토콜 업그레이드는 비트코인에서 직접적으로 더 많은 무신뢰 스마트 계약 기능 구현을 가능하게 할 수 있을 것으로 보인다. Bitlayer 는 이러한 발전 상황을 면밀히 모니터링하고 있으며, 해당 기능들이 안정적으로 제공될 시 이를 통합할 수 있도록 준비하고 있다. 이는 다음과 같은 변화로 이어질 수 있다.

    - **증명 위원회 제거:** 특정 증명 프로세스에 대한 증명 위원회의 필요성을 잠재적으로 제거하여, 보다 완전한 무신뢰 모델로의 전환.

    - **비허가성 강화:** 분쟁 해결 프로토콜에서 사전 서명된 거래나 특정 역할에 대한 의존도를 낮추어, 보다 개방적이고 비허가적인 시스템의 구현.

- **온체인 운영자 선출:** 롤업 운영자의 선출 및 교체를 온체인에서 직접적으로 관리하는 구조로 전환하여, 이를 통한 생존성과 탈중앙화의 강화.
- **최적화된 담보 관리:** 보안성을 저해하지 않는 동시에 동일 에포크 내에서 보다 정교한 담보 재사용 메커니즘을 사용하여 운영자의 자본 비용을 절감.

2. **고급 증명 시스템:** 영지식 증명 시스템 및 기타 암호화 기술의 발전을 지속적으로 평가하고 통합하여 증명 생성 효율성을 개선하고, 온체인 검증 비용을 절감하여, 전반적인 시스템 성능을 향상.

9 결론

백서를 통해 정직한 다수 가정에 기반한 보안성, 확장성 및 EVM 호환성을 갖춘 비트코인 계산 계층인 Bitlayer 에 대해 알아보았다. Bitlayer 는 BitVM 패러다임을 기반으로 구축되어 복잡한 범용 계산을 가능하게 하는 동시에, 그 보안성을 비트코인 네트워크에 직접 앵커링한다. Bitlayer 의 핵심 기여는 비트코인 상에서 레이어 2 상태 전이의 지속적이고 검증 가능한 정산을 가능케 하는 최초의 재귀적 정산 프로토콜을 구현했다는 점이다. 이 프로토콜은 동일한 보안 모델을 공유하는 시너지 자산 브리지와 완전한 EVM 호환 실행 계층과 더불어 탈중앙화 애플리케이션을 위한 완벽하고 실용적인 플랫폼을 구축한다.

Bitlayer 는 BTCFi 생태계를 위한 최상위 인프라 구축을 향한 기초적인 단계로 생각된다. 비트코인이 최종 정산 계층으로, Bitlayer 가 효율적이고 검증 가능한 계산 계층으로 기능하는 명확한 아키텍처를 제시하는 Bitlayer 의 결과물은 비트코인의 방대한 잠재력을 실현하기 위한 실용적인 청사진을 제공한다. 낮은 거래 비용과 강력한 검열 저항성을 우선시하는 이 설계가 앞으로 비트코인 기반의 확장성 및 안전성을 갖춘 애플리케이션에 대한 추가 연구를 촉진하기를 바란다.

참고 문헌

[1] S. Nakamoto. 비트코인: P2P 전자 현금 시스템, 2009. http://bitcoin.org/ bitcoin.pdf.

[2] Linus, Robin, Lukas Aumayr, Alexei Zamyatin, Andrea Pelosi, Zeta Avarikioti, Matteo Maffei. BitVM2: 비트코인 레이어 2 브리징. https://bitvm.org/bitvm_bridge.pdf.

[3] Robin Linus. BitVM: 비트코인으로 무엇이든 계산하는 법, 2023 년 12 일. https://bitvm.org/ bitvm.pdf.

[4] J. Groth. 페어링 기반 비대화형 논증의 크기 분석, 2016. https://eprint. iacr.org/2016/260.pdf.

[5] Dan Boneh, Victor Shoup. 응용 암호학 대학원 과정. https://toc. cryptobook.us/book.pdf.

[6] 비트코인 위키. 스크립트, 2025. https://en.bitcoin.it/wiki/Script.

[7] Kalodner, H., Goldfeder, S., Chen, X., Weinberg, S. M., 및 Felten, E. W. (2018). Arbitrum: 확장 가능한 프라이빗 스마트 계약. 제 27 회 USENIX 보안 심포지엄(USENIX 보안 18). https://www.usenix.org/system/files/conference/usenixsecurity18/ sec18-kalodner.pdf

[8] J. Buchmann, E. Dahmen, S. Ereth, A. H¨ulsing, M. R¨uckert. 윈터니츠 일회성 서명 체계의 보안성에 관하여, 2011. https://eprint.iacr.org/2011/191.pdf

[9] Wood, G. (2014). “이더리움: 안전한 탈중앙화 및 일반화 거래 원장.” 이더리움 프로젝트 황서. https://ethereum.github.io/yellowpaper/paper.pdf.

[10] Rubin, J. (2020). “BIP-0119: CHECKTEMPLATEVERIFY.” 비트코인 개선안. https://github.com/bitcoin/bips/blob/master/bip-0119.mediawiki.