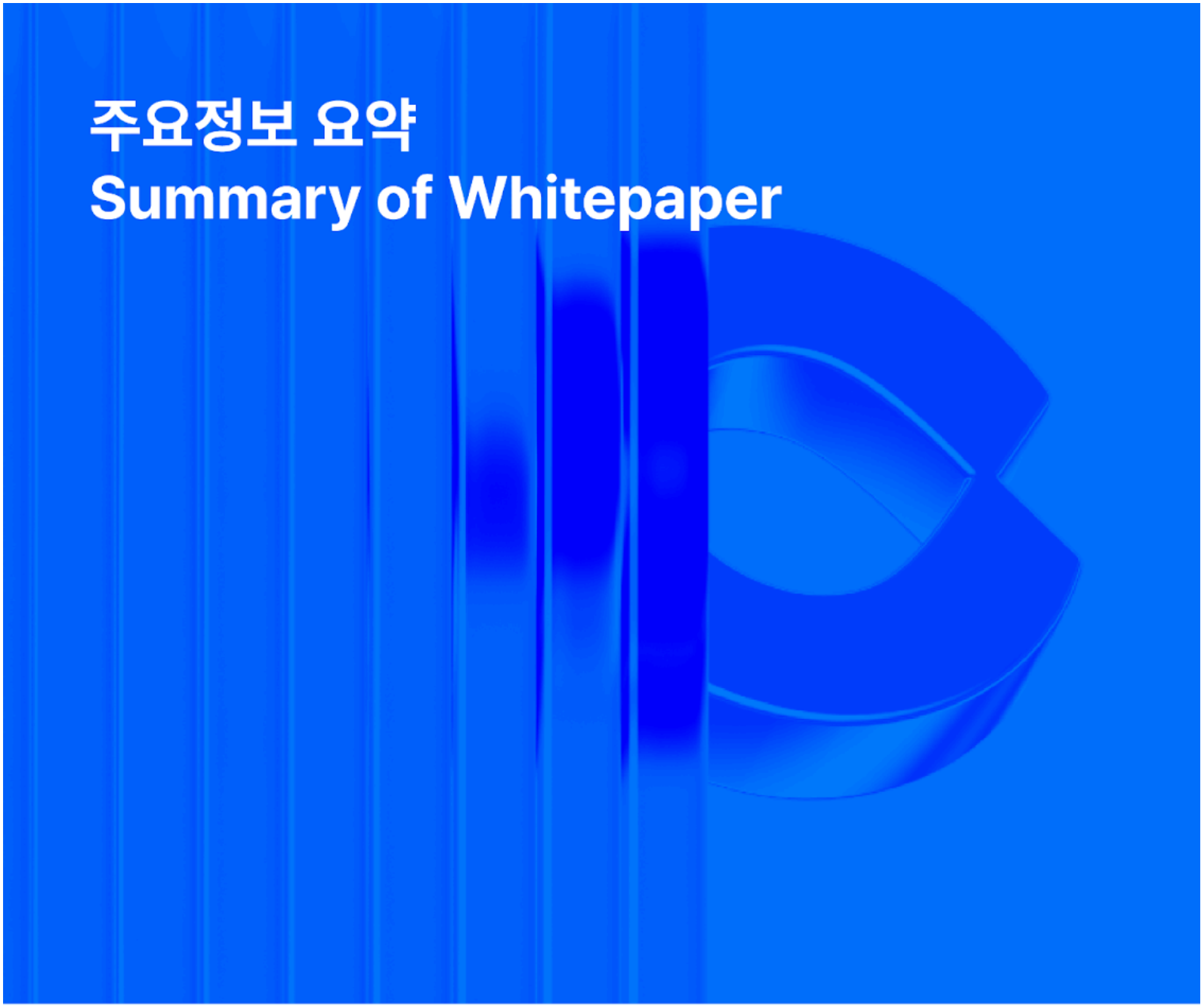
주요정보 요약
Summary of Whitepaper



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

coinone

기본 정보

가상자산 카테고리	유틸리티
거래지원 네트워크	Zircuit
홈페이지	https://www.zircuit.com/
참고문헌 (백서, Docs 등)	https://docs.zircuit.com/

1. 프로젝트 정보

개요

저킷(Zircuit)은 EVM 호환 제로지식 롤업으로 웹3의 무한한 가능성을 지원하는 플랫폼입니다. 혁신적인 L2 연구를 바탕으로 구축된 네트워크는 검증된 인프라와 제로지식 증명을 결합한 독특한 하이브리드 아키텍처를 통해 개발자들에게 최상의 환경을 제공합니다. 고급 성능과 보안 기능을 결합한 시퀀서 레벨의 기술을 통해 사용자는 빠른 트랜잭션, 낮은 수수료, 그리고 완벽한 보안을 경험할 수 있습니다.

저킷은 현재 메인넷에서 운영 중이며, 사용자는 유동성 허브에 참여하거나 dApp 생태계를 탐험하며, 개발자를 위한 "Build to Earn" 프로그램을 통해 다양한 기회를 발견할 수 있습니다.

주요 특징

선구적인 연구
지난 1년 반 동안 저킷은 롤업 보안 도구, 롤업 압축 기술, 확장형 암호화 등 다양한 주제에 대한 연구를 선도해 왔습니다. 이러한 연구 성과는 Ethereum Foundation으로부터 다수의 L2 연구 지원금을 받는 데 기여했습니다.

시퀀서 레벨의 보안 (Sequencer Level Security)
저킷은 트랜잭션 풀(mempool)을 모니터링하여 악의적인 트랜잭션을 탐지하고 블록에 포함되지 않도록 방지함으로써 사용자 보호를 강화합니다. 이는 기존의 애플리케이션 또는 스마트 컨트랙트 수준에서의 보안 접근법을 넘어, 근본적인 시퀀서 레벨에서 보안을 구현하는 혁신적인 접근 방식입니다.

최첨단 성능
저킷은 서킷을 세분화하고 증명을 집계하여 효율성을 극대화하고 운영 비용을 줄였습니다. 대규모 트랜잭션 배치와 가속화된 증명 처리 방식을 결합하여 사용자에게 빠르고 저렴한 트랜잭션 환경을 제공합니다.

이더리움 애플리케이션 호환성
저킷은 MetaMask와 같은 주요 지갑 및 Hardhat과 같은 도구를 완벽히 지원하며, 새로운 프로그래밍 언어나 프레임워크를 학습할 필요 없이 기존의 Ethereum dApp을 쉽게 배포할 수 있습니다.

저킷은 개발자와 사용자 모두에게 최적의 성능과 보안을 제공하며, 웹3 생태계의 확장을 이끄는 강력한 플랫폼입니다.

아키텍쳐

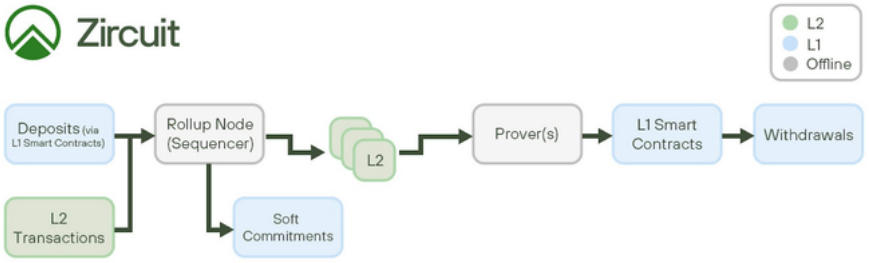
저킷은 전투에서 검증된 롤업 인프라와 제로 지식 증명을 결합한 새로운 하이브리드 아키텍처를 기반으로 합니다. 아키텍처는 다음으로 구성됩니다.

- L2 블록을 구성하기 위해 거래를 처리하는 시퀀서
- 이러한 블록에 대한 유효성 증명을 생성하는 증명기
- L1에서 시스템과 상호 작용하는 스마트 계약

결과적으로 인출 거래에 대한 도전 기간이 필요 없는 빠른 확정성을 갖춘 zkEVM 롤업이 생성됩니다.

거래 흐름

거래 흐름 개요는 다음 이미지에 설명되어 있습니다.



출처: Zircuit Docs

Zircuit의 거래 흐름

롤업의 거래는 L1에서 시작되거나 L2에서 직접 시작될 수 있습니다.

L1에서 시작되는 거래는 ETH 또는 기타 자산이 L2에 브리지되는 입금 거래이거나 크로스 도메인 함수 호출일 수 있습니다. L2 거래는 L2 계정 간의 자산 이체이거나 L2에 자금이 있는 계정에서 L2에 배포된 계약에 대한 호출일 수 있습니다. 어느 경우든 트랜잭션은 시퀀서, 실행 엔진, 배처로 구성된 롤업 노드에서 처리됩니다.

시퀀서는 L1에서 스마트 계약 이벤트에서 생성된 입금 트랜잭션을 추가하여 실행 엔진에 블록에 어떤 트랜잭션이 포함되어야 하는지 지시합니다.

배처는 L2 트랜잭션 배치를 L1에 제출하여 사용자가 전체 데이터를 사용할 수 있도록 합니다. 사용자는 이 데이터를 사용할 수 있는데, 이는 트랜잭션이 완료되었다는 소위 소프트 커밋먼트이기 때문입니다. 소프트 커밋먼트는 트랜잭션이 L2 체인에 포함될 것이라는 커밋먼트이지만 실행이 아직 증명되지 않았기 때문에 아직 최종으로 간주되지 않습니다.

실행 엔진은 배치의 트랜잭션을 처리하고 새로운 L2 상태를 생성합니다. 실행 엔진은 이러한 트랜잭션을 L2 블록에 넣어 처리합니다.

이러한 L2 블록은 각각 특정 역할을 가진 Zircuit 프로버에서 처리합니다. 예를 들어, 하나는 트랜잭션이 올바르게 실행되었음을 증명하는 반면, 다른 하나는 관련 Keccak 작업이 올바르게 수행되었음을 증명할 수 있습니다. 세 번째는 이전 두 증명에서 수행한 작업의 집계를 증명할 수 있습니다.

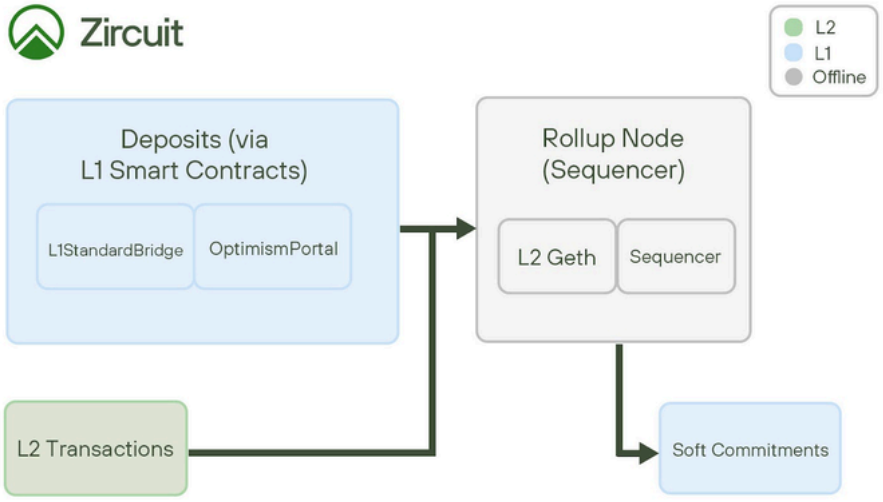
결과적으로 Zircuit은 신속한 증명 생성을 위해 병렬 증명 생성을 활용하지만, 증명 집계를 통해 체인에서 검증할 수 있는 단일 증명을 생성합니다. Zircuit은 회로를 특수 부분으로 분해하고 증명을 집계함으로써 더 낮은 운영 비용으로 더 큰 효율성을 달성합니다. 생성된 최종 유효성 증명은 L2 블록 배치에 대한 증명의 집계입니다.

L2 블록 배치에 대한 증명은 스마트 계약을 통해 Ethereum에서 검증됩니다. 검증이 성공하면 L2 상태 루트가 관련 계약에서 업데이트되어 증명이 방금 검증된 배치에 포함된 L2 블록에 포함된 변경 사항을 기록합니다. 이때 해당 L2 블록은 최종으로 간주됩니다. 이제 추가 지연 없이 L2에서 인출할 수 있습니다.

유효성 증명 검증을 통해 거래 흐름이 한눈에 보입니다. 다음 두 섹션에서는 입금, L2 및 출금 거래 처리에 대해 자세히 설명합니다.

입금 및 L2 거래
이 섹션에서는 입금 및 L2 거래와 관련된 아키텍처를 더 자세히 다룹니다. 입금 거래는 Ethereum에서 Zircuit으로 Ether 또는 ERC-20 토큰과 같은 자산을 연결합니다. L2 거래는 계정 간 Ether 전송이나 Zircuit에서 스마트 계약을 호출하는 것과 같이 Zircuit 자체에서 발생하는 거래입니다.

다음 이미지는 입금 또는 L2 거래에 관련된 Zircuit의 특정 구성 요소를 강조 표시합니다.



출처: Zircuit Docs

입금 거래는 L1에서 스마트 계약을 호출하여 시작됩니다. 두 가지 계약을 사용하여 Zircuit에 ETH를 입금할 수 있습니다. L1StandardBridge 계약과 OptimismPortal 스마트 계약입니다. L1StandardBridge 계약은 적절한 함수를 호출하여 Ether를 입금하는 데 사용할 수 있습니다. OptimismPortal 계약은 receive 함수를 구현하여 Ether를 연결합니다. 즉, 호출할 함수를 지정하지 않고 이 계약으로 전송된 모든 Ether는 자동으로 연결됩니다.

L1StandardBridge만 ERC-20 토큰을 연결하는 데 사용할 수 있으며, ERC-721 토큰은 L1ERC721Bridge 계약을 통해 연결할 수 있습니다.

L2에서 발생하는 거래는 자연스럽게 이러한 스마트 계약을 호출할 필요가 없습니다. 대신 RPC 호출이나 지갑을 통해 체인으로 전송되고 롤업 노드에서 직접 처리됩니다.

롤업 노드가 입금을 관찰하거나 L2 거래를 수신하면 시퀀서 기능을 통해 해당 거래가 포함된 블록을 구성합니다.

롤업 노드는 L1에서 관찰된 각 입금 이벤트에 대해 입금 거래를 생성하고 이러한 거래를 모든 L2 거래와 함께 실행 엔진에 전달합니다. 실행 엔진은 Geth의 수정된 버전으로, 입금

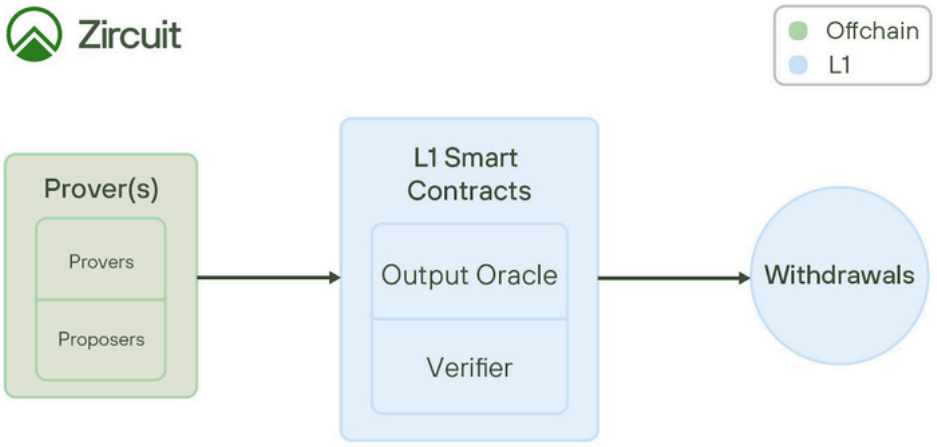
거래 유형과 롤업 작업에 필요한 기타 사소한 변경 사항을 지원합니다.

롤업 노드는 또한 블록에 포함된 거래를 배처 서비스에 전달하고, 배처 서비스는 거래와 해당 인수를 L1에 calldata로 게시하여 전체 데이터 가용성을 제공합니다.

이 시점에서 거래는 증명자가 거래를 포함하는 관련 블록을 수신한 후 L2에 포함됩니다. 거래 발신자와의 추가 상호 작용은 필요하지 않습니다.

인출
인출 거래는 ETH와 같은 자산을 Zircuit에서 꺼내 L1로 반환합니다. 인출 거래는 먼저 적절한 함수를 호출하는 L2StandardBridge 계약에 L2 거래를 보내서 시작됩니다.

해당 거래가 L2에 포함되면 다음 다이어그램의 구성 요소가 인출 거래를 완료하는 데 관련이 있습니다.



출처: Zircuit Docs

인출은 L2 인출 거래가 포함된 상태 루트가 해당 증명과 함께 L1에 포함되면 완료될 수 있습니다. 증명은 인출 기능을 완료하는 데 필요한데, 이는 자금을 인출하는 계정에 처음부터 자금이 있었음을 보장하기 때문입니다.

배치가 증명되면 결과 상태 루트는 마지막 상태 루트에서 상태 전환에 대한 유효성 증명과 함께 L2OutputOracle 스마트 계약을 통해 L1로 푸시됩니다. 검증자가 증명을 검증하면 상태 루트는 최종으로 기록됩니다. 이에 의존하는 모든 인출은 즉시 처리할 수 있습니다.

모듈형 증명 디자인

이 페이지에서는 Zircuit Prover 설계에 대한 개요를 제공하고, 구성 요소와 적용된 설계 원칙을 포착합니다.

아이디어
Zircuit은 기본 zkEVM 기술 스택을 추상화하는 검증자와 이를 중심으로 아키텍처를 구축했습니다. 이를 통해 증명 시스템, 회로, 커밋먼트 체계, 곡선 및 재귀 체계를 쉽게 교체할 수 있는 모듈식 스택이 생성됩니다.

설계 원칙
스냅인/스냅아웃
이러한 모든 모듈은 최소한의 노력으로 교체할 수 있습니다. 회로 토폴로지에 대한 의사

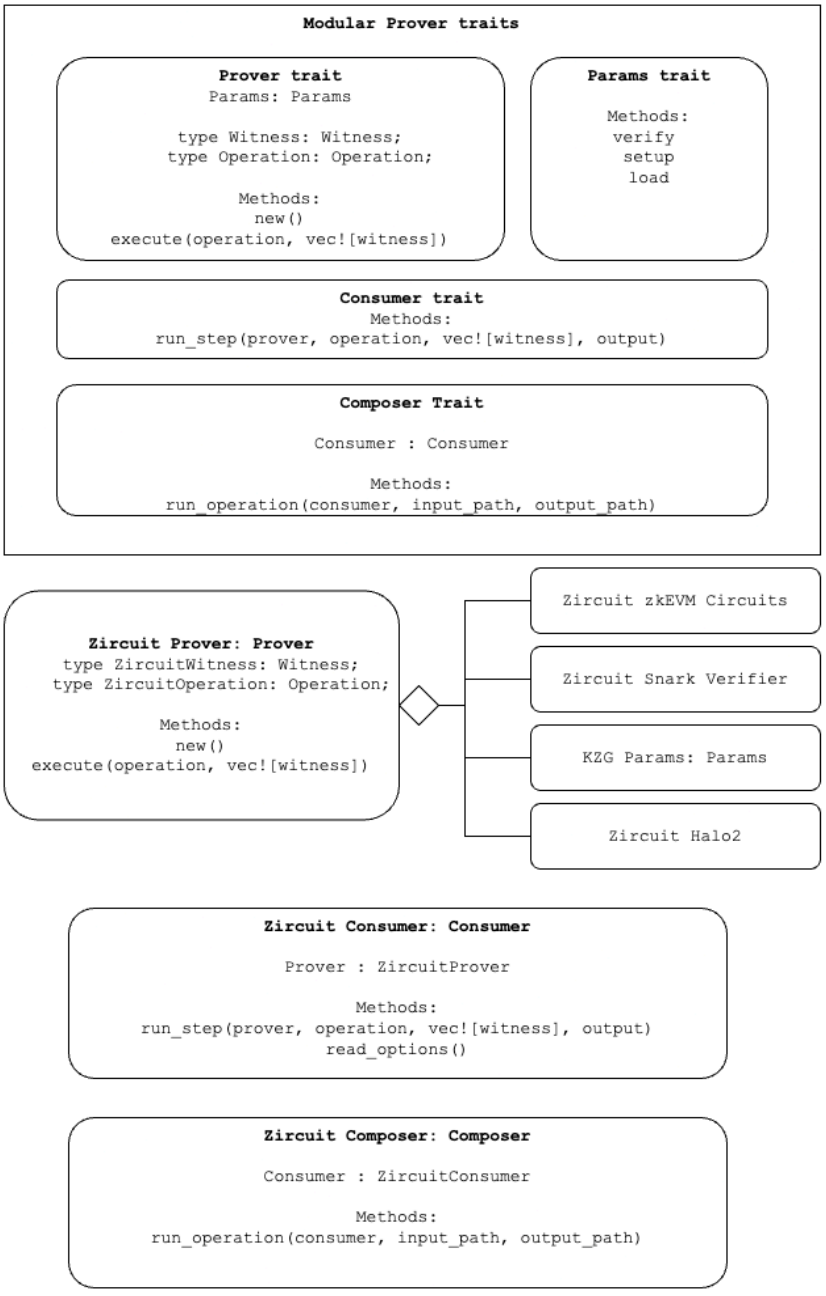
결정은 구성 모듈이라는 단일 모듈에 집중되어 있습니다.

낮은 결합도
각 모듈은 특정 기능에만 집중합니다. 다른 모듈을 호출해야 하는 경우 독립성을 유지하기 위해 직접 호출하는 대신 인터페이스나 특성을 사용합니다.

높은 응집도
각 모듈은 자체 포함되고 수행하는 작업에 집중하여 시스템 내에서 명확한 경계와 책임을 보장합니다.

확장성/확장성
이 시스템은 상당한 재설계 없이도 추가 모듈이나 복잡성 증가를 쉽게 확장하고 수용하도록 설계되었습니다. 이 원칙은 진화하는 요구 사항과 기술 발전에 대비하여 시스템을 미래에 대비하는 것을 목표로 합니다.

설계



출처: Zircuit Docs

모듈식 증명기 특성
설계 원칙을 충족하고 개발 전반에 걸쳐 일관성을 유지하기 위해 시스템 구성 요소가 준수해야 하는 특성 집합을 정의했습니다.

매개변수(Params)
이 특성은 기본 증명 시스템이 증명을 계산하는 데 필요한 매개변수를 정의합니다. 이 특성은 체계를 추상화하고 상위 수준 API만 노출하여 외부 구성 요소에서 하위 논리를 캡슐화합니다. 노출된 API는 setup, load 및 verify입니다. setup은 개발 및 테스트를 제외하고는 거의 사용되지 않습니다. load는 증명 시스템에서 사용할 매개변수를 준비하는 방법이고 verify는 매개변수가 손상되지 않았는지 확인하기 위해 매개변수에 대한 보안 검사를 수행하는 방법입니다.

증명기(Prover)
증명기는 두 개의 유형 연결과 매개변수 연결을 포함한 세 개의 연결로 구성됩니다. 증명기는 분명히 증명을 계산하기 위해 매개변수가 필요합니다. 연관된 유형은 위트니스와 오퍼레이션입니다. 위트니스는 프로버가 구현해야 하는 유형으로, 입력 블록이 위트니스되는 방식을 정의합니다. 오퍼레이션은 프로버가 지원하는 작업을 정의합니다. 이러한 작업은 고수준 비즈니스 로직을 달성하기 위해 조율할 수 있는 원자적 프로빙 오퍼레이션입니다. L2 블록 프로빙은 모든 zkEVM 회로가 병렬로 증명되어야 합니다. 로직의 분석은 오퍼레이션에서 캡처됩니다. 각 오퍼레이션은 프로버가 execute 메서드에서 구현합니다.

Consumer
외부 애플리케이션에 프로버가 작동하는 방식은 블랙박스입니다. 그들이 알아야 할 것은 프로버가 지원하는 오퍼레이션과 전달해야 하는 위트니스 유형뿐입니다. 이것은 프로버의 변경 사항이 프로빙 파이프라인(프로버를 사용하는 애플리케이션)을 절대 손상시키지 않도록 보장하는 추상화 계층입니다. 프로버 로직을 인식하지 못하는 소비자 바이너리는 각 프로버에서 지원되는 오퍼레이션/단계를 실행하고 결과를 출력할 수 있습니다.

Composer
L2 블록 프로빙과 같은 비즈니스 로직을 충족하려면 여러 단계를 병렬로 순차적으로 조율해야 합니다. 증명이 어떻게 수행되는지에 대한 세부 사항이나 특정 구현에 대한 너무 구체적인 내용은 언급하지 않고도 L2 블록을 증명하는 데는 다양한 회로 증명, 재귀, 데이터 가용성 커미트먼트 게시, 배칭 등이 포함된다고 생각할 수 있습니다. 작업을 수행하는 방법에 대한 결정은 종종 파이프라인에 하드코딩됩니다. 이는 문제가 되는데, 새로운 기술로 인해 변경 사항이 생기면 증명 파이프라인을 변경해야 하며 이러한 작업의 취약성으로 인해 비용이 많이 들고 오류가 발생하기 쉽기 때문입니다. 작성자는 어떤 작업을 어떤 순서로 수행해야 하는지에 대한 오케스트레이션 논리를 추상화합니다. 여기에는 run_operation 이라는 하나의 메서드가 있는데, 입력(마지막으로 실행된 작업의 출력)을 받아서 하나의 결과를 출력합니다. 기본적으로 입력으로 받은 현재 단계 결과에 따라 다음에 실행할 올바른 단계로 소비자를 호출합니다. 작성자는 예를 들어 블록 번호를 사용하여 여러 소비자를 지원하여 증명자 버전 관리를 활성화할 수 있습니다.

Zircuit Prover
이 Prover는 Prover 특성을 구현하고 Zircuit에서 사용되며, zkEVM 회로, Snark Verifier, Params, halo2 Proof System 등 여러 분리되거나 상호 교환 가능한 종속성을 캡슐화합니다.

Zircuit Consumer
Zircuit consumer는 Zircuit prover에서 지원하는 모든 단계를 노출합니다.

Zircuit Composer
Zircuit composer는 블록의 증명을 완료하기 위해 수행해야 하는 오케스트레이션 단계를 정의합니다.

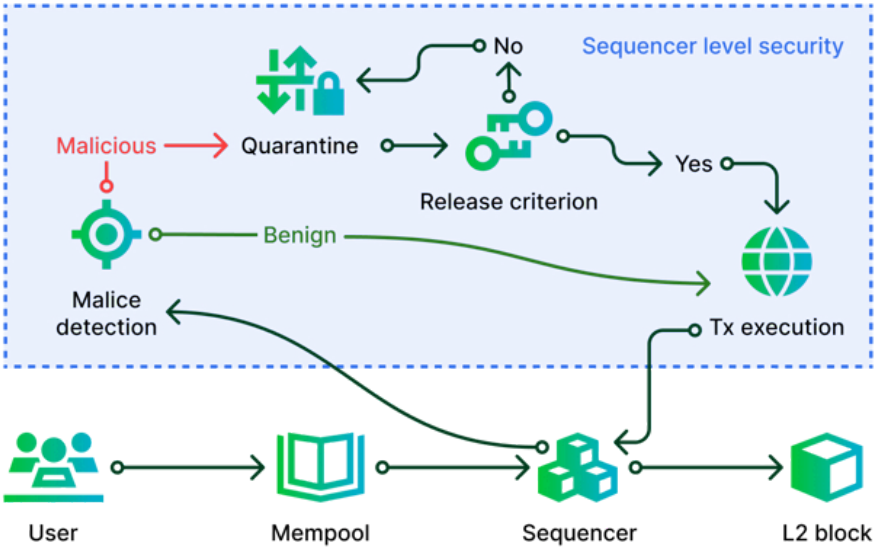
소결
이 설계의 유익한 측면은 시스템의 다른 부분을 건드릴 필요 없이 각 구성 요소를 업데이트, 업그레이드 또는 교체할 수 있다는 것입니다. 예를 들어, 커뮤니티에서 개발한 새로운 zkEVM 회로가 증명하는 데 훨씬 빠르다면, 모듈식 증명자는 매우 겸손한 노력으로 업데이트할 수 있으며, zkEVM 종속성과 증명자 자체 내의 접착 코드만 교체하면 됩니다. 또 다른 예는 종속성을 패치하고 코드의 다른 곳은 패치하지 않음으로써 GPU 기반 증명 시스템(예: halo2)을 지원하는 것입니다.

시퀀서 수준 보안(SLS)

개요
Zircuit은 악의적인 거래에 대한 mempool을 모니터링하고 블록에 포함되지 않도록 방지하여 시퀀서 수준에서 사용자를 보호합니다. 애플리케이션 및 스마트 계약 수준에 초점을 맞춘 일반적인 보안 노력과 비교할 때 Zircuit의 혁신적인 접근 방식은 기본 시퀀서 수준으로 직접 이동합니다.

이 접근 방식은 **"시퀀서 수준 보안"(SLS)**이라고 하며 Zircuit 네트워크에 보안 계층을 추가합니다. 이 새로운 접근 방식은 시퀀서가 레이어 2에서 완료되기 전에 잠재적인 악의적 의도가 있는지 거래를 면밀히 조사한다는 것을 의미합니다. SLS 프로토콜은 의심스러운 거래의 조기 감지 및 격리를 가능하게 함으로써 하드 포크 또는 블록 반전의 논란이 되는 조치를 필요로 하지 않고 스마트 계약과 레이어 2의 보안을 강화합니다.

자세한 설명
Zircuit은 시퀀서 수준에서 또 다른 보안 계층을 추가합니다. 시퀀서는 Ethereum과 유사하게 사용자의 거래를 수집하고 사전 정의된 규칙에 따라 정렬하는 권한이 있는 노드입니다.



출처: Zircuit Docs

SLS 프로토콜 개요
위의 그림은 프로토콜 개요를 보여줍니다. 여기에는 세 가지 주요 구성 요소가 포함됩니다. (1) 악의 탐지, (2) 격리-해제 기준, (3) 거래 실행.
SLS 시퀀서에 도착하면 mempool의 거래는 처음에 악의 탐지 모듈로 라우팅됩니다. 우리의 토크에서 이 모듈을 "오라클"이라고 합니다. 이는 거래가 양성인지 잠재적으로 악성인지 식별합니다. 양성 거래는 표준 시퀀싱 프로토콜을 준수하여 블록 포함을 위해 즉시 대기합니다. 반대로 악성으로 플래그가 지정된 거래는 중간 보류 영역 역할을 하는 격리-해제 기준 모듈로 전환됩니다. 여기에서 특정 해제 기준에 대한 엄격한 검증 프로세스를 거칩니다. 이러한 기준을 충족하는 거래는 거래 실행 모듈로 전달됩니다. 거래 실행 모듈은 다음 L2 블록에서 블록체인 상태에 대해 거래를 실행합니다. 성공적으로 실행된 거래는 다음 L2 블록에 포함되도록 SLS 시퀀서로 다시 순환됩니다.

악의 탐지(Malice Detection)
악의 탐지는 다음 단계를 통해 수행됩니다.

시퀀서에 의한 거래 선택: 시퀀서는 다음 블록에 잠재적으로 포함할 거래 목록을 선택합니다. 이 단계는 다른 표준 시퀀싱 프로토콜과 동일합니다. 여기에는 mempool의 거래와 L1에서 시작된 입금 거래가 모두 포함됩니다.

체인 끝의 병렬 시뮬레이션: 각 거래는 블록체인 끝의 현재 상태를 사용하여 독립적으로 시뮬레이션됩니다. 이 단계에서는 거래의 병렬 처리가 가능합니다. 이러한 시뮬레이션의 결과는 향후 종속성 분석 및 악의 탐지에 필수적인 데이터를 제공합니다. (1) 각 거래의 시뮬레이션 결과 (2) 각 거래에서 읽고 쓴 블록체인 상태.

거래 종속성 분석: 각 거래에서 읽고 쓴 상태에 대한 분석을 수행하고 모든 거래 간의 종속성을 식별합니다. 비공식적으로, 한 거래를 실행하면 다른 거래를 실행한 결과가 변경될 수 있는 경우 한 거래는 다른 거래에 종속됩니다.

독립 트랜잭션에 대한 병렬 감지 및 종속 트랜잭션에 대한 순차 감지: 다른 이전 트랜잭션에 종속되지 않은(즉, 독립된) 모든 트랜잭션의 경우 시퀀서는 시뮬레이션 결과에 대한 병렬 감지를 수행할 수 있습니다. 다른 종속 트랜잭션은 블록 컨텍스트 내에서 순차 시뮬레이션 및 감지를 위해 대기합니다.

트랜잭션 포함: 시퀀서는 양성으로 식별된 모든 트랜잭션을 포함하여 블록을 마무리합니다. 시간 제약이나 복잡성으로 인해 완전히 분석할 수 없는 종속 트랜잭션은 다음 주기로 연기됩니다. 이러한 트랜잭션이 다시 포함될지 고려될 때 다음 라운드에서 동일한 감지 프로세스가 적용됩니다.

시퀀서가 악성을 식별하는 데 사용하는 알고리즘에는 프로그램 분석, 머신 러닝 및 규칙 기반 방법이 포함됩니다.

격리-해제 기준(Quarantine-Release Criterion)
격리 중에는 트랜잭션이 실행되지 않으며 블록에 포함될 수 없습니다. 시퀀서는 트랜잭션이 격리에 배치된 시점에 대한 정보를 유지 관리합니다. 트랜잭션은 은퇴 기준 중 하나를 충족하면 mempool에서 삭제되거나 해제 기준 중 하나를 충족하면 격리에서 해제됩니다.

정확한 은퇴 기준과 해제 기준은 시퀀서가 정의할 수 있습니다.

Mempool 은퇴 기준:
Nonce 기준. 이 기준은 nonce가 더 이상 유효하지 않아 트랜잭션을 더 이상 블록에 포함할 수 없는 경우 충족됩니다.

시간 기준 및 메모리 제약. 이 기준은 트랜잭션이 노드가 유지하는 mempool을 어지럽히는 경우 충족됩니다. 이러한 트랜잭션은 네트워크에 다시 제출하여 mempool(격리된 상태)에 다시 들어갈 수 있습니다.

Zircuit의 현재 구현은 격리된 트랜잭션이 시퀀서에 의해 은퇴 기준에 따라 주기적으로 확인되도록 합니다. 다음은 보안 관점에서 실행 가능한 격리 해제 기준 목록입니다.

해제 기준:
시간 기준. 시간 기준은 시퀀서가 사용자에게 악의적인 거래에 대응하도록 제공하는 반응 시간을 나타냅니다. 거래가 필요 이상으로 오랫동안 격리된 경우 해제되어 블록 포함을 고려할 수 있습니다. 거래가 격리에 머무르는 데 필요한 정확한 시간은 구성 매개변수입니다.

실패 기준. 체인 상태의 변경으로 인해 거래가 실패하면 되돌리기가 발생하므로 안전하게 블록에 포함할 수 있습니다. 되돌린 거래는 블록체인 상태를 변경하지 않습니다.

관리 기준. 악의적 행위가 탐지되면 가끔 거짓 양성 반응이 발생할 것으로 예상됩니다. 이러한 상황에서 보안 전문가로 구성된 시퀀서 운영 팀은 거래를 해제하기로 한 결정을 관리적으로 무시할 수 있습니다.

다른 가능한 해제 기준에 대해서는 시퀀서 수준 보안 문서를 참조하세요.

출시 시점에 Zircuit 시퀀서는 매우 긴 기간(년 순서)으로 설정된 시간 기준만 사용합니다. 이 기간 동안 시퀀서는 특권 당사자가 릴리스 기준을 트리거할 때까지 기다립니다. 격리된 보증금을 청구하기 위한 지원은 Discord에서 문의하세요.

거래 실행
격리에서 해제되면 시퀀서는 다음 블록을 형성할 때 거래를 포함하는 것을 고려할 수 있습니다. 이는 일반 시퀀싱 규칙의 적용을 받습니다. 현재 체인 상태에 대해 원래 가격이 낮았기 때문에 다시 제출된 거래의 경우 거래를 다시 격리해서는 안 됩니다. SLS 프로토콜은 계정 주소, 거래 데이터(함수 선택기 및 호출 데이터) 및 값을 사용하여 새 수신 거래가 이미 격리에서 해제된 거래의 중복인지 여부를 확인해야 합니다.

빌더 요구 사항
SLS 프로토콜은 Zircuit에서 기본적으로 구현됩니다. 따라서 Zircuit에 배포된 모든 스마트 계약은 기본적으로 SLS에 포함되고 보호됩니다. 그러나 SLS는 AI로 구동되는 최선의 노력 서비스이므로 실수가 발생할 수 있습니다. 따라서 개발자는 최상의 엔지니어링 관행을 따르고 코드의 보안을 면밀히 조사하는 것이 좋습니다. 시퀀서 수준 보안 프로토콜은 추가 보조 보안 조치로 간주해야 합니다.

SLS는 악의를 탐지하고 자산을 보호하기 위해 가격 정보가 필요합니다. Zircuit에 기본적으로 배포된 토큰은 CoinGecko에 상장되어야 합니다. 이 기술은 이러한 가격 피드와 토큰을 자동으로 인식합니다. 다른 네트워크에 배포되고 Zircuit에 브리지된 토큰은 Oracle의 가격 책정 시스템에 빠르게 포함되도록 Discord를 통해 Zircuit 팀에 문의하는 것이 좋습니다.

2. 토큰 이코노미

가상자산 소개

QTUM은 퀀텀 생태계의 유틸리티 토큰으로, 퀀텀체인의 트랜잭션 수수료, 블록 제안 및 검증에 참여하기 위한 스테이킹 참여와 이에 대한 보상으로 사용됩니다.

발행량 및 유통량계획

저킷 토큰(ZRC)의 총 발행량은 10,000,000,000개입니다.

네트워크 및 컨트랙트 주소

이더리움 (Ethereum)	0xfd418e42783382E86Ae91e445406600Ba144D162
저킷 (Zircuit)	0xfd418e42783382E86Ae91e445406600Ba144D162

ZRC는 저킷 아키텍처의 핵심 역할을 하며, 참여자들이 추가 보상을 받을 수 있게 하고, 네트워크 애플리케이션의 공정한 출시(fair launches)에 참여할 수 있도록 하며, 생태계의 성장을 촉진합니다. 생태계의 중심축으로서 ZRC는 개발자와 사용자 간의 인센티브를 정렬하여 적극적인 협업과 혁신을 유도합니다.

- Coingecko: https://www.coingecko.com/en/coins/zircuit
- CoinMarketCap: https://coinmarketcap.com/currencies/zircuit/

시즌 1 에어드롭

총 ZRC 토큰 공급량의 7%가 시즌 1 포인트로 할당되었습니다. 262,200개의 고유 지갑 주소가 에어드롭 클레임 자격을 얻었으며 시즌 1 스냅샷은 2024년 7월 7일 UTC 기준 16:00:00에 진행되었습니다.

시즌 2 에어드롭

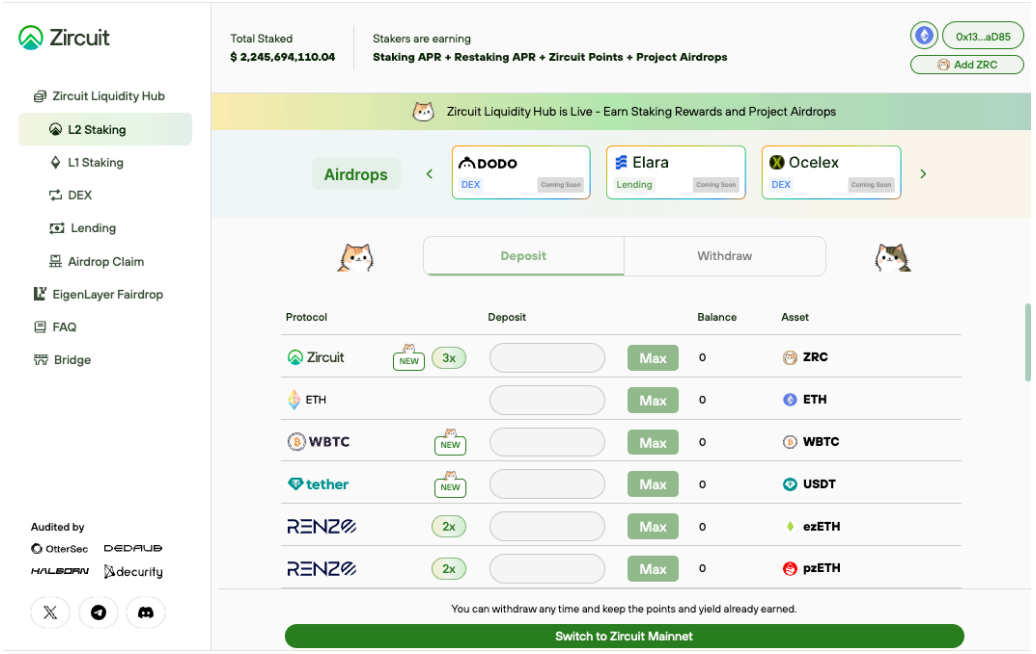
총 ZRC 토큰 공급량의 3%가 시즌 2 포인트로 할당되었습니다. 스냅샷은 2024년 11월 20일 UTC 기준 오전 5:00:00에 진행되었습니다.

할당 및 베스팅

- **21.00%** 에어드롭 및 커뮤니티 리워드:
  - **7.00%** 시즌 1 에어드롭
  - **3.00%** 시즌 2 에어드롭
  - **2.45%** 캠페인(Fairdrop, Catizen, Binance Web3 등)
  - **8.55%** 향후 에어드롭 및 보상: 6개월 및 12개월 락업 후 24개월 동안 선형적으로 베스팅
- **13.08%** 커뮤니티 프로비전: 1년 락업 후 24개월 동안 선형 베스팅
- **17.93%** 생태계 개발: 1년 락업 후 24개월 동안 선형 베스팅
- **18.70%** 재단: 1년 락업 후 24개월 동안 선형 베스팅
- **18.74%** 팀: 1년 락업 후 24개월 동안 선형 베스팅
- **10.55%** 투자자: 1년 락업 후 24개월 동안 선형 베스팅

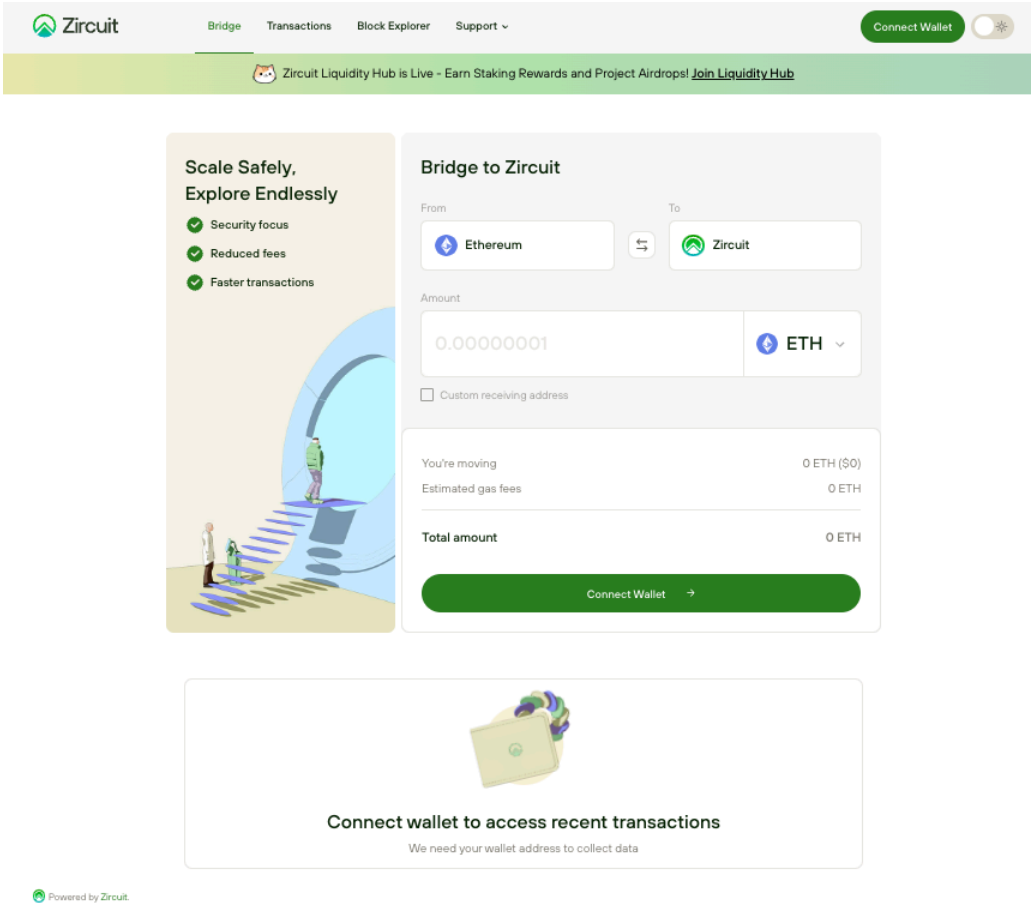
3. 참고자료

Liquidity Hub



출처: Ziricuit 공식 홈페이지_*Liquidity Hub*

Bridge to Zircuit



출처: Zircuit 공식 홈페이지_*Bridge*

위험고지 안내
Disclaimer

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

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

가상자산은 법정화폐가 아니므로 특정 주체가 가치를 보장하지 않습니다.

coinone