

웜홀(Wormhole)

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.

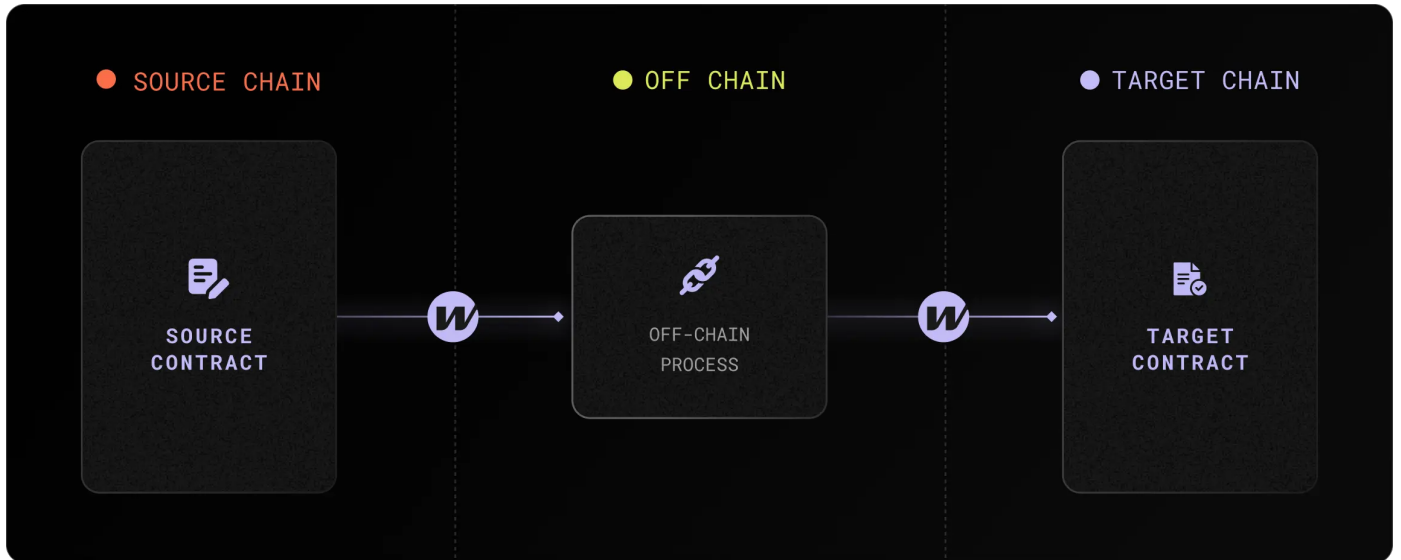
프로토콜

웜홀 소개

빠르게 진화하는 블록체인 기술 환경에서, 서로 다른 블록체인 간 상호운용성은 여전히 중대한 과제로 남아 있습니다. 개발자들은 여러 블록체인 전반에서 원활하게 동작하는 애플리케이션을 만드는 데 종종 어려움을 겪으며, 이는 혁신과 탈중앙화 생태계의 잠재력을 제한합니다.

웜홀은 블록체인 간 안전하고 효율적인 통신을 가능하게 하는 범용 메시지 전송 프로토콜을 제공함으로써 이 문제를 해결합니다. 다양한 블록체인 네트워크 간 데이터와 자산 전송을 허용함으로써, 웜홀은 전통적으로 이러한 생태계를 분리해 왔던 장벽을 허물어 줍니다.

웜홀은 강력한 보안성, 확장성, 투명성에 중점을 둔 점이 특징입니다. 프로토콜은 모든 크로스체인 트랜잭션의 무결성을 보장하는 탈중앙화된 검증자 네트워크의 지원을 받습니다. 이는 실제 응용 분야에서 입증된 웜홀의 성능과 결합되어, 개발자들에게 멀티체인 애플리케이션을 자신 있게 구축하고 확장할 수 있는 신뢰할 수 있는 플랫폼을 제공합니다.



웜홀을 통해 개발자들은 하나의 체인에 얽매이지 않고 여러 블록체인 생태계의 강점을 활용할 수 있습니다. 이는 애플리케이션이 통합되고 효율적인 사용자 경험을 유지하면서, 솔라나의 높은 처리량, 이더리움의 보안성, Cosmos의 상호운용성과 같은 다양한 네트워크의 고유 기능을 누릴 수 있음을 의미합니다.

이 페이지는 웜홀이 빠르고 안전하며 확장 가능한 크로스체인 통신을 가능하게 하는 데 필요한 핵심 개념과 구성요소를 소개합니다.

웜홀이 해결하는 문제는 무엇인가?

상호운용성은 빠르게 진화하는 블록체인 환경에서 중요하고도 어려운 과제입니다. 개별 블록체인들은 종종 고립되어 있어, 여러 생태계 전반에 걸쳐 작동하는 통합형 애플리케이션의 잠재력이 제한됩니다. 웜홀은 블록체인 간 원활한 통신을 가능하게 함으로써, 각 네트워크의 고유 기능을 활용하는 멀티체인 애플리케이션을 개발할 수 있도록 이 문제를 해결합니다.

웜홀이 해결하는 핵심 문제는 다음과 같습니다:

- **블록체인 고립:** 웜홀은 이질적인 블록체인을 연결하여, 네트워크 간 자산, 데이터, 거버넌스 행동의 전송을 가능하게 합니다.
- **크로스체인 복잡성:** 웜홀은 크로스체인 통신의 복잡성을 추상화하여, 개발자가 크로스체인 애플리케이션을 더 쉽게 구축하고 배포할 수 있게 합니다.
- **보안성과 탈중앙화:** 웜홀은 메시지를 검증하고 서명하는 탈중앙화된 가디언 네트워크를 통해 보안을 우선시하여, 크로스체인 상호작용의 무결성을 보장합니다.

웜홀이 제공하는 것

웜홀은 폭넓은 사용 사례를 지원하는 도구와 프로토콜의 모음을 제공합니다:

- **크로스체인 메시징:** 블록체인 간 임의의 데이터를 안전하게 전송하여, 크로스체인 탈중앙 애플리케이션 개발을 가능하게 합니다.
- **자산 전송:** [포탈\(Portal\)](#)과 같은 웜홀 기반 프로토콜에 의해 지원되는 체인 간 토큰 이동을 손쉽게 할 수 있습니다.
- **D- eveloper tools:** 웜홀의 [TypeScript SDK](#), [Wormholescan](#), [Wormholescan API](#) 및 문서를 활용하여, 크로스체인 애플리케이션을 빠르고 효율적으로 구축하고 배포할 수 있습니다.

웜홀이 아닌 것

- **웜홀은 블록체인이 아닙니다:** 웜홀은 서로 다른 블록체인을 연결하여 상호작용하게 하는 통신 레이어로서 작동하며, 그 자체가 블록체인은 아닙니다.
- **웜홀은 토큰 브리지가 아닙니다:** 웜홀은 토큰 전송을 지원하지만, 훨씬 더 다양한 크로스체인 애플리케이션을 지원하여 일반적인 브리지보다 훨씬 다재다능합니다.

웜홀의 사용 사례

다음은 웜홀로 구현 가능한 잠재적 애플리케이션의 예시입니다:

- **크로스체인 거래소:** [Wormhole 커넥트](#)를 사용하면, 웜홀에 연결된 어떤 체인에서든 예치를 허용하는 거래소를 구축할 수 있어 유동성 접근성이 크게 향상됩니다.
- **크로스체인 거버넌스:** 여러 블록체인에 걸친 커뮤니티를 가진 프로젝트는 웜홀을 사용해 각 체인의 투표를 지정된 거버넌스 체인으로 전달하여, 결합된 제안을 통해 통합된 의사결정을 가능하게 할 수 있습니다.
- **크로스체인 게임:** 솔라나와 같은 고성능 네트워크에서 게임을 개발하고, 보상은 이더리움과 같은 다른 네트워크에서 발행할 수 있습니다.

더 살펴보기

웜홀 생태계, 구성요소, 프로토콜에 대해 더 알아보세요:

- **[아키텍처\(Architecture\)](#):** 프로토콜 구성요소를 탐색합니다.
- **[프로토콜 사양\(Protocol Specifications\)](#):** 웜홀 위에 구축된 프로토콜에 대해 학습합니다.

데모

데모는 튜토리얼보다 실제 구현에 더 가깝습니다:

- **웜홀 Scaffolding**: Scaffolding 리포지토리로 프로젝트를 빠르게 설정하세요.
- **데모 튜토리얼(Demo Tutorials)**: 여러 블록체인 전반에서 웜홀의 역량을 보여주는 다양한 데모를 살펴보세요.

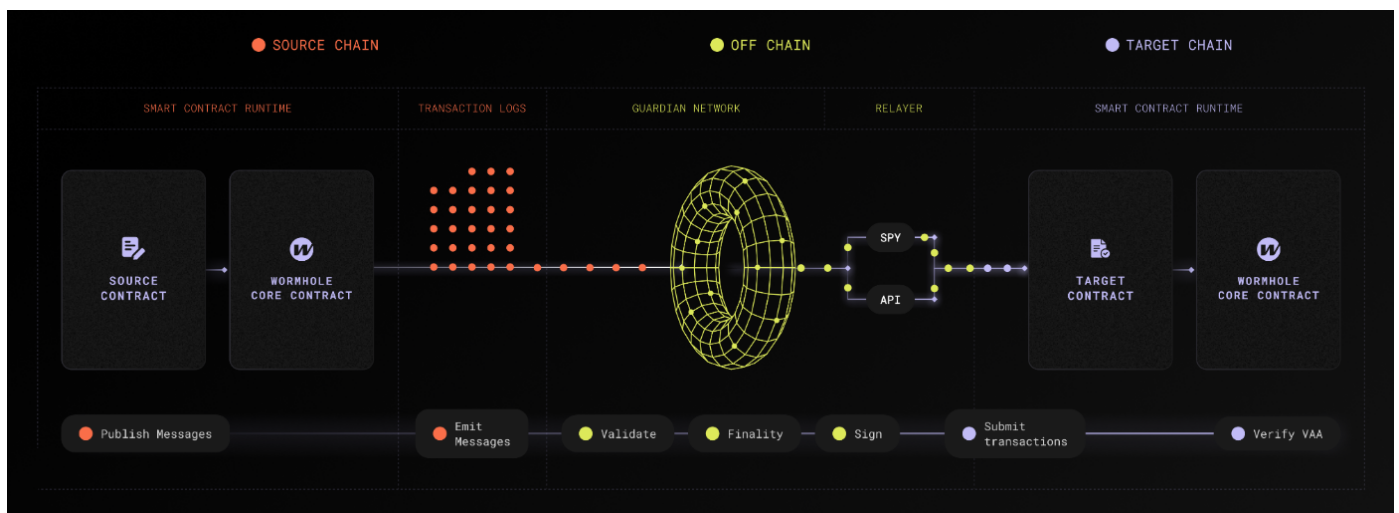
제품별 지원 네트워크

웜홀은 점점 더 많은 블록체인을 지원합니다. 각 웜홀 제품이 어떤 네트워크를 지원하는지는 [Supported Networks by Product](#) 페이지에서 확인하세요.

아키텍처 개요

아키텍처

웜홀에는 주목할 만한 여러 구성요소가 있습니다. 각 구성요소를 자세히 논의하기에 앞서, 이 페이지에서는 주요 요소들이 어떻게 결합되어 동작하는지 개요를 제공합니다.



앞선 다이어그램은 웜홀 아키텍처를 통한 멀티체인 통신의 엔드 투 엔드 흐름을 다음과 같이 설명합니다:

1. **소스 체인**: 소스 계약이 소스 체인의 웜홀 코어 컨트랙트와 상호작용하여 메시지를 발생시키며, 코어 컨트랙트는 블록체인의 트랜잭션 로그에 메시지를 게시합니다.
2. **가디언 네트워크**: 가디언은 이러한 메시지를 검증하고 서명하여 VAA(Verifiable Action Approval)를 생성합니다.
3. **릴레이어**: 오프체인 릴레이어나 애플리케이션이 VAA를 가져와 대상 체인으로 릴레이합니다.

4. **타깃 체인:** 타깃 체인에서 메시지는 적절한 계약에 의해 소비됩니다. 이 계약은 웜홀 코어 컨트랙트와 상호작용하여 VAA를 검증하고, 의도된 멀티체인 작업을 실행합니다.

릴레이어에서 타깃 체인으로의 흐름에는 용도에 따라 달라질 수 있는 엔트리 포인트 계약이 포함됩니다:

- 일부 애플리케이션에서는 타깃 계약이 엔트리 포인트 역할을 하며 코어 컨트랙트를 통해 검증을 수행합니다.
- Wrapped Token Transfers(WTT)와 같은 제품에서는, WTT 계약 자체가 코어 컨트랙트와 상호작용합니다.

온체인 구성요소

- **Emitter:** 코어 컨트랙트의 메시지 게시 메소드를 호출하는 계약입니다. 메시지를 식별하기 위해, 코어 컨트랙트는 Emitter와 시퀀스 번호에 대한 세부 정보를 트랜잭션 로그에 이벤트로 기록합니다. 이는 귀하의 크로스체인 디앱이거나 기존 생태계 프로토콜일 수 있습니다.
- **웜홀 코어 컨트랙트:** 기본 계약으로, 가디언이 관찰하며 멀티체인 통신을 근본적으로 가능하게 하는 계약입니다.
- **트랜잭션 로그:** 가디언이 코어 컨트랙트가 발생시킨 메시지를 관찰할 수 있도록 하는 블록체인별 로그입니다.

오프체인 구성요소

- **가디언 네트워크:** 자체 P2P 네트워크에 존재하는 검증자들입니다. 가디언은 지원되는 각 체인의 코어 컨트랙트가 발생시킨 메시지를 관찰·검증하여 VAA(서명된 메시지)를 생성합니다.
- **가디언:** 가디언 네트워크의 19개 검증자 중 하나로, VAA 멀티시그에 기여합니다.
- **스파이:** 가디언 네트워크 내에서 게시되는 메시지를 구독하는 데몬입니다. 스파이는 네트워크 트래픽을 관찰하고 전달할 수 있어 VAA 배포 확장에 도움이 됩니다.
- **API:** VAA 또는 가디언 네트워크에 대한 세부 정보를 조회하는 REST 서버입니다.
- **VAA:** Verifiable Action Approval(VAA)은 웜홀 코어 컨트랙트에서 관찰한 메시지에 대한 서명된 인증서입니다.
- **릴레이어:** VAA를 타깃 체인으로 릴레이하는 모든 오프체인 프로세스입니다.
 - **웜홀 릴레이어:** 웜홀 릴레이어 계약을 통해 온체인으로 요청된 메시지를 전달하는 탈중앙화 릴레이어 네트워크입니다.
 - **커스텀 릴레이어:** 특정 프로토콜이나 멀티체인 애플리케이션의 VAA만을 처리하는 릴레이어입니다. 오프체인에서 커스텀 로직을 실행할 수 있어 가스 비용을 줄이고 멀티체인 호환성을 높일 수 있습니다. 현재는 멀티체인 애플리케이션 개발자가 커스텀 릴레이어를 개발·호스팅할 책임이 있습니다.

웜홀 생태계 (The Wormhole Ecosystem)

웜홀은 여러 블록체인에 걸쳐 탈중앙화 애플리케이션을 연결하는 크로스체인 메시징 프로토콜입니다. 웜홀은 다양한 멀티체인 과제를 해결하기 위한 상호운용성 도구 모음을 제공하며, 개발자가 필요에 따라 각 제품을 조합해 사용할 수 있도록 합니다.

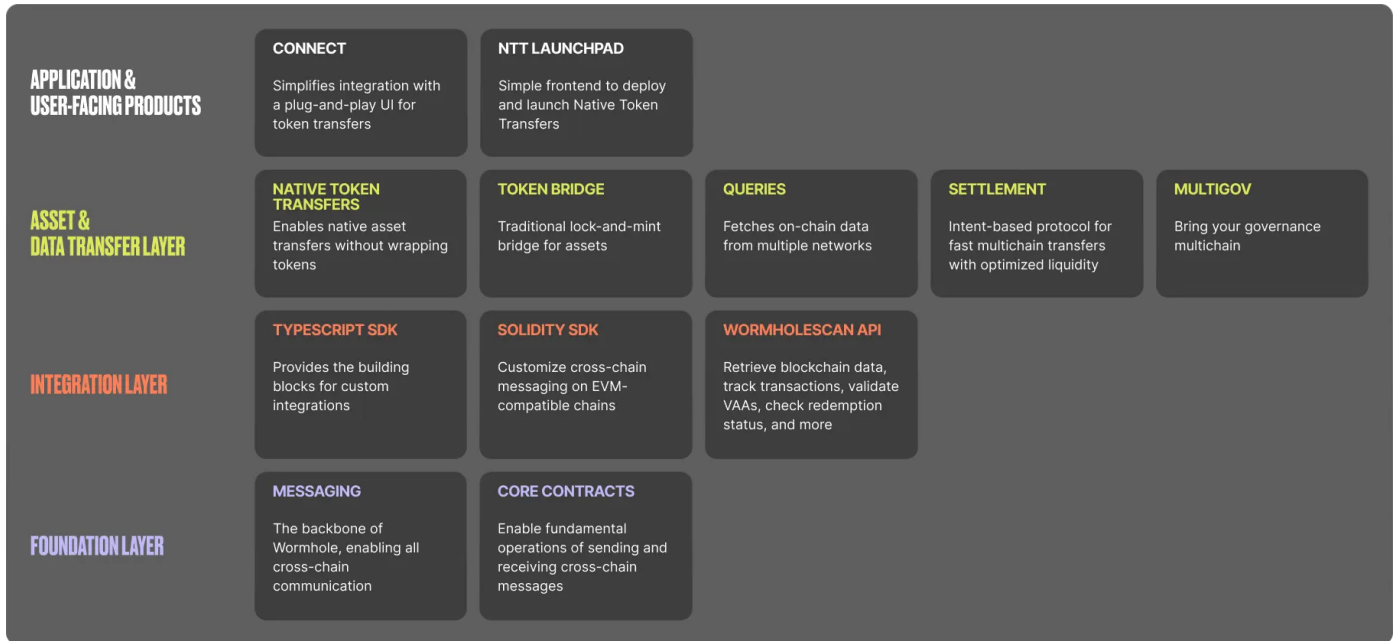
UI 기반의 단순한 브리징 경험부터, 래핑 자산 없이 네이티브 토큰 전송 플로우, 실시간 크로스체인 데이터 조회, 복잡한 자산 이동을 위한 고급 정산 계층까지 — 웜홀은 각 목적에 맞는 제품을 제공합니다. 모든 솔루션은 웜홀의 핵심 메시징 네트워크와 통합되어, 각 모듈이 독립적으로 작동하거나 함께 사용할 수 있도록 설계되었습니다.

이 페이지에서는 웜홀의 톨 구조적 구성, 계층별 관계, 그리고 멀티체인 애플리케이션 구축 시 각 기능을 어떻게 결합할 수 있는지를 설명합니다.

생태계 개요 (Ecosystem Overview)

아래 다이어그램은 웜홀의 모듈형 스택을 4개의 계층으로 나누어 보여줍니다:

- **애플리케이션 및 사용자 중심 제품층:**
상단에는 사용자를 위한 인터페이스형 솔루션(예: 커넥트, NTT Launchpad)이 포함됩니다. 커넥트는 간단한 브리징 UI를, NTT Launchpad는 네이티브 자산 배포를 간소화합니다.
- **자산 및 데이터 전송 계층:**
핵심 브리징 및 데이터 솔루션이 위치하며, NTT, WTT, Queries, Settlement, MultiGov 등이 여기에 포함됩니다.
이는 토큰 이동, 실시간 데이터 요청, 고급 크로스체인 정산, 크로스체인 거버넌스를 담당합니다.
- **통합 계층 (Integration Layer):**
TypeScript SDK와 WormholeScan API를 통해 개발자 친화적 통합 기능을 제공합니다.
이 계층은 애플리케이션에 크로스체인 기능을 손쉽게 추가할 수 있게 합니다.
- **기반 계층 (Foundation Layer):**
웜홀 메시징 시스템과 코어 컨트랙트가 위치하며, 네트워크 전체의 검증 및 메시지 전달을 보안적으로 보장합니다.



상호운용성의 실제 구현 (Bringing It All Together)

웜홀의 모듈화된 구조는 필요한 구성 요소만 선택하여 사용할 수 있도록 설계되었습니다. 예를 들어, 디앱에 빠르게 브리징 기능을 추가하고 싶다면 커넥트(상위 계층)를 사용하고, 내부적으로는 파운데이션 계층이 메시징을 처리합니다. 또는 체인 간 원시 메시지를 주고받고 싶다면 통합 계층(Integration Layer)(TypeScript/ Solidity SDK)을 통해 메시징 레이어를 직접 통합할 수 있습니다. 추가로, 쿼리를 사용해 실시간 데이터 호출 기능을 추가하거나 Native Token Transfers를 통해 더 유연한 브리징 흐름을 구현할 수도 있습니다. 이러한 구성요소들은 독립적이지 않고 상호 결합될 수 있도록 설계되었습니다. 예를 들어, 쿼리를 사용해 한 체인에서 잔액을 조회한 뒤, 정산을 통해 다른 체인에서 온체인 스왑을 수행할 수 있습니다. 모든 웜홀 제품은 가디언 네트워크에 의해 보호되는 메시징 백본을 기반으로 하여, 모든 크로스체인 상호작용의 신뢰성과 보안을 보장합니다.

보안 (Security)

핵심 보안 가정 (Core Security Assumptions)

웜홀은 가디언 노드 네트워크에 의해 보안이 유지됩니다.

가디언들은 메시지를 검증하고 서명하며, 전체의 초다수(예: 19명 중 13명)가 동일한 메시지에 서명하면 유효하다고 간주됩니다.

타깃 체인의 스마트 계약은 서명과 메시지 형식을 검증한 후 트랜잭션을 승인합니다.

- 웜홀의 핵심 보안 메커니즘은 **서명된 메시지(VAA)** 입니다.
- 가디언 네트워크는 전 세계 최고 수준의 19개 검증자 회사로 구성되어 있습니다.
- 가디언은 코어 컨트랙트 통합자 요청 시 서명된 상태 증명(VAA)을 생성합니다.
- 각 가디언은 웜홀이 지원하는 모든 블록체인의 풀 노드를 실행하며, 만약 특정 블록체인에 합의 공격이나 하드포크가 발생할 경우, 해당 블록체인은 네트워크에서 자동으로 분리됩니다.
- 어떤 체인의 코어 컨트랙트라도 다른 체인에서 발행된 서명된 VAA의 진위를 검증할 수 있습니다.
- 릴레이어는 웜홀 생태계 내에서 **신뢰되지 않은 주체(Untrusted)** 로 간주됩니다.

요약하자면:

- 코어 통합자는 자신이 통합하지 않은 체인과 계약으로부터 위험에 노출되지 않습니다.
- 기본적으로 신뢰하는 것은 웜홀의 서명 프로세스와 각 체인의 코어 컨트랙트뿐입니다.
- 필요에 따라 통합 계약과 체인 의존성을 확장할 수 있습니다.

가디언 네트워크 (Guardian Network)

웜홀은 발전 중인 플랫폼입니다.

현재 가디언 세트는 19개의 검증자로 구성되어 있지만, 이는 현 블록체인 기술의 한계에서 비롯된 수치입니다.

거버넌스 (Governance)

거버넌스는 계약 업그레이드가 이루어지는 과정입니다.

가디언은 가디언 네트워크 내부에서 발생한 거버넌스 제안에 수동으로 투표하며, 이 제안은 이후 생태계 계약에 제출됩니다.

이는 거버넌스 행위가 시스템의 나머지 부분과 동일한 보안 표준을 적용받음을 의미합니다.

모든 거버넌스 행위는 가디언의 3분의 2 초과 동의가 필요합니다.

거버넌스 메시지는 코어 컨트랙트나 Wrapped Token Transfers(WTT) 등 다양한 웜홀 모듈을 대상으로 할 수 있습니다.

가디언이 메시지에 서명하면 해당 제안에 대한 투표를 의미하며, 가디언의 2/3 이상이 서명하면 제안은 유효한 것으로 간주됩니다.

가디언은 거버넌스를 통해 다음을 수행할 수 있습니다:

- 가디언 세트 변경
- 가디언 세트 확장
- 생태계 계약 업그레이드

거버넌스 시스템은 오픈소스로 관리됩니다.

모니터링 (Monitoring)

웜홀의 다층 보안 전략의 핵심 요소는 가디언이 모두 숙련된 검증자 기업으로, 자체적인 운영, 모니터링, 보안 프로세스를 보유하고 있다는 점입니다.

이러한 이질적(homogeneous하지 않은) 구조는 잠재적 사기 행위를 조기에 감지할 가능성을 높이며, 단일 실패 지점을 줄입니다.

가디언은 웜홀 검증자뿐만 아니라 웜홀 내 모든 블록체인의 검증자 역할도 수행하므로, 네트워크 전반에서 포괄적 모니터링이 가능합니다.

가디언은 다음을 모니터링합니다:

- 각 블록체인의 블록 생성 및 합의 상태: 합의 위반 시 해당 블록체인은 일시적으로 네트워크에서 분리됩니다.
 - 스마트 계약 수준 데이터: 가디언은 거버너(Governor) 프로세스를 통해 모든 체인의 토큰 이동과 순환 공급을 지속적으로 모니터링합니다.
 - 가디언 활동: 가디언 네트워크는 자율적인 탈중앙 컴퓨팅 네트워크로 작동하며, 각 검증자 간 독립적 보안 조치를 유지합니다.
-

자산 계층 보호 (Asset Layer Protections)

웜홀의 핵심 강점 중 하나는 가디언이 여러 블록체인 전반의 자산 무결성을 검증하고 보호할 수 있다는 점입니다.

이를 위해 **글로벌 회계사(Global Accountant)**가 모든 체인의 웜홀 자산 총 유통량을 추적하여, 공급 불변성(supply invariant)을 위반할 수 있는 자산 전송을 방지합니다.

또한 가디언은 **거버너**가 추적하는 인플로우·아웃플로우 규칙을 위반하지 않는 전송만 서명할 수 있습니다. Governor는 모든 블록체인의 자산 이동을 모니터링하며, 잠재적 익스플로잇을 시사하는 의심스러운 전송은 지연시킵니다.

오픈소스 (Open Source)

웜홀은 완전한 오픈소스 프로젝트로, 모든 코드와 인프라를 공개적으로 운영합니다.

- [Wormhole core repository](#)
- [Wormhole Foundation GitHub organization](#)
- [Wormhole contract deployments](#)

감사 (Audits)

웜홀은 대대적인 감사를 거쳤으며, 제3자 감사를 29건 완료했고 더 많은 감사가 진행 중입니다. 감사를 수행한 업체는 다음과 같습니다:

- Trail of Bits
- Neodyme
- Kudelski
- OtterSec
- Certik
- Hacken
- Zellic
- Coinspect
- Halborn
- Cantina

모든 감사와 최종 보고서는 [GitHub 저장소의 보안 페이지](#)에서 확인할 수 있습니다.

버그 바운티 (Bug Bounties)

웜홀은 소프트웨어 개발 분야에서 가장 큰 규모의 버그 바운티 프로그램 중 하나를 운영하며, 화이트햇 커뮤니티와의 협업에 지속적으로 헌신해 왔습니다.

웜홀은 Immunefi 프로그램을 통해 버그 바운티를 운영하며, 최고 지급액은 500만 달러입니다.

웜홀 보안에 기여하고자 한다면, [화이트햇 시작하기](#) 섹션을 확인하고, [웜홀 기여자 가이드라인](#)을 따르시기 바랍니다.

버그 바운티 프로그램에 대한 제출 관련 자세한 정보는 [Wormhole Immunefi 페이지](#)를 참고하세요.

웜홀 코어 컨트랙트 (Wormhole Core Contracts)

웜홀 코어 컨트랙트는 지원되는 각 블록체인 네트워크에 배포되어 있습니다. 이 컨트랙트는 웜홀 상호운용성 프로토콜의 기본 구성요소로서, 안전하고 효율적인 멀티체인 메시징을 가능하게 하는 기초 계층 역할을 합니다. 모든 멀티체인 애플리케이션은 코어 컨트랙트와 직접 상호작용하거나, 코어 컨트랙트와 상호작용하는 다른 컨트랙트를 통해 간접적으로 동작합니다.

이 페이지는 코어 컨트랙트의 핵심 기능을 요약하고, 코어 컨트랙트가 어떻게 동작하는지를 설명합니다.

핵심 기능

- **멀티체인 메시징:** 메시지의 형식을 표준화하고 보안성을 부여하여, 웜홀로 연결된 블록체인 네트워크 간 메시지 전송 시 일관된 통신을 가능하게 하며, 개발자가 각 네트워크의 고유 기능을 활용할 수 있도록 합니다.
- **검증 및 유효성 확인:** 대상 체인에서 수신된 모든 VAA를 가디언 서명 확인을 통해 검증하여, 메시지가 변조되지 않았고 정당함을 보장합니다.
- **가디언 네트워크 조정:** 웜홀의 가디언 네트워크와 조정하여 체인 간 안전하고 신뢰 최소화 (trustless)된 통신을 촉진하고, 검증된 상호작용만 처리되도록 보장해 프로토콜 전반의 보안성과 신뢰성을 강화합니다.
- **이벤트 발생:** 처리되는 모든 멀티체인 메시지에 대해 이벤트를 발생시켜, 메시지 상태 추적, 디버깅, 실시간 멀티체인 이벤트에 반응하는 애플리케이션 등의 네트워크 활동 모니터링을 가능하게 합니다.

코어 컨트랙트 동작 방식

웜홀 코어 컨트랙트는 안전하고 효율적인 멀티체인 트랜잭션을 촉진하는 핵심입니다. 서로 다른 블록체인 네트워크 간 통신을 위해 트랜잭션 데이터를 표준화된 메시지로 패키징하고, 그 진위를 검증한 뒤, 대상 체인에서 올바르게 실행되도록 보장합니다.

다음은 메시지 전송에서 웜홀 코어 컨트랙트의 역할을 설명합니다:

1. **메시지 제출:** 사용자가 멀티체인 트랜잭션을 시작하면, 소스 체인의 웜홀 코어 컨트랙트가 트랜잭션 데이터를 표준화된 메시지 페이로드로 패키징하여 가디언 네트워크에 검증을 위해 제출합니다.

2. **가디언 검증:** 가디언들은 메시지를 독립적으로 관찰하고 서명합니다. 충분한 수의 가디언이 메시지에 서명하면, 이 서명 모음은 메시지와 메타데이터와 함께 결합되어 VAA를 생성합니다.
3. **메시지 수신 및 실행:** 대상 체인에서 웜홀 코어 컨트랙트는 검증된 메시지를 수신하고, 가디언 서명을 확인한 후, 토큰 발행, 상태 업데이트, 특정 스마트 컨트랙트 함수 호출 등 해당 조치를 실행합니다.

체인 간 메시지가 어떻게 흐르고 어떤 구성요소가 관여하는지 자세히 보려면 [아키텍처 개요 페이지](#)를 참고하세요.

메시지 제출

소스 체인의 코어 컨트랙트에 함수를 호출하여 멀티체인 메시지를 전송할 수 있으며, 그러면 코어 컨트랙트가 메시지를 게시합니다. 메시지 게시 전략은 체인별로 다를 수 있으나, 일반적으로 코어 컨트랙트는 블록체인 로그에 다음 항목들을 기록합니다:

- **emitterAddress:** 해당 메시지를 게재하도록 요청하는 컨트랙트입니다.
- **sequenceNumber:** 특정 emitter(및 암묵적으로 체인)에 대해 메시지마다 증가하는 고유 번호입니다.
- **consistencyLevel:** 가디언이 게시된 이벤트를 관찰하고 서명하기 전에 도달해야 하는 파이널리티 수준입니다. 이는 재조직(reorg)과 롤백에 대한 방어 수단이며, 일단 트랜잭션이 “최종”으로 간주되면 그로 인해 발생한 상태 변화가 되돌려지지 않음을 보장합니다. 체인마다 합의 메커니즘이 다르므로, 각 체인은 서로 다른 파이널리티 가정을 가지며, 이 값은 체인별로 다르게 처리됩니다. 각 체인의 파이널리티 옵션은 [Wormhole Finality](#) 참조 페이지에서 확인하세요.

메시지를 게시하는 데에는 솔라나에서 게시할 때를 제외하면 별도의 수수료가 없지만, 향후 변경될 수 있습니다.

메시지 수신

대상 체인의 코어 컨트랙트에서 멀티체인 메시지를 수신할 때는 일반적으로 VAA의 구성요소를 파싱하고 검증해야 합니다. VAA를 수신·검증함으로써 가디언 네트워크가 해당 메시지에 적절히 서명했음을 확인하고, 체인 간 전송된 데이터의 무결성과 진위를 유지합니다.

멀티캐스트 모델

멀티캐스트란, 단일 메시지 또는 트랜잭션을 송신·수신 함수에 대상 주소나 체인을 지정하지 않고 여러 블록 체인에 동시에 브로드캐스트하는 것을 의미합니다. VAA는 “이 체인의 이 컨트랙트가 이런 내용을 말했다”는 사실을 증명합니다. 따라서 VAA는 기본적으로 멀티캐스트이며, 사용되는 모든 체인의 코어 컨트랙트에서 진위를 검증받을 수 있습니다.

이 기본 멀티캐스트 모델은 생태계 전반의 상태를 쉽게 동기화하도록 합니다. 하나의 블록체인은 단일 액션으로 낮은 지연(latency)으로 모든 체인에 데이터를 제공할 수 있으며, 이는 여러 블록체인으로 데이터를 라우팅할 때 발생하는 n^2 복잡도를 줄여줍니다.

이는 애플리케이션이 목적지 주소나 체인을 지정할 수 없다는 의미는 아닙니다. 예를 들어, Wrapped Token Transfers(WTT)와 웜홀 릴레이어 컨트랙트는 대상 체인에서 일부 목적지 정보를 전달·검증하도록 요구합니다.

VAA 생성이 릴레이 과정과 분리되어 있기 때문에, 단일 체인만을 대상으로 할 때 멀티캐스트 모델은 추가 비용을 발생시키지 않습니다. 특정 블록체인에서 데이터가 필요하지 않다면 그곳으로 릴레이하지 않으면 되며, 이 경우 비용이 들지 않습니다.

VAA(Verified Action Approvals)

Verified Action Approvals(VAA)는 웜홀의 핵심 메시징 프리미티브입니다. 이는 크로스체인 애플리케이션 컨트랙트가 코어 컨트랙트와 상호작용할 때마다 발생하는 크로스체인 데이터 패킷입니다.

가디언은 컨트랙트에서 발생한 메시지를 대상 체인으로 보내기 전에 검증합니다. 가디언 다수가 메시지가 유효하다고 동의하면, 메시지 본문의 keccak256 해시에 서명합니다.

메시지는 VAA라는 구조로 래핑되며, 이는 메시지와 가디언 서명을 결합해 하나의 증명(proof)을 이룹니다. VAA는 (emitter_chain, emitter_address, sequence) 튜플로 고유하게 인덱싱됩니다. VAA를 얻으려면 이 정보를 사용해 Wormholescan API를 조회할 수 있습니다.

sequence 필드는 송신 체인의 블록 최종 순서에 의존합니다. 낮은 일관성(consistency) 수준(즉, 파이널리티 대기 없음)을 선택하면, 체인 재조직으로 인해 사용자 관점에서 “같아 보이는” 메시지에 대해 서로 다른 VAA가 나타날 가능성이 있습니다.

해당 체인이 재조직을 겪지 않고 메시지를 포함한 블록이 사실상 파이널리티에 도달한 경우에만 (emitter_chain, emitter_address, sequence) 튜플을 고유하다고 간주할 수 있습니다. 다만, 이론적으로 긴 재조직이 발생하여 이전에 발생한 시퀀스 번호를 무효화하거나 변경할 가능성은 항상 존재합니다.

VAA 형식

기본 VAA는 다음과 같이 헤더와 바디로 구성됩니다:

헤더(Header) : 현재 VAA에 대한 메타데이터, 활성 Guardian 세트, 현재까지 수집된 서명 목록을 보관합니다.

- `version byte` : VAA 버전.
- `guardian_set_index u32` : 어떤 가디언 세트가 서명 중인지 나타냅니다.
- `len_signatures u8` : 저장된 서명 개수.
- `signatures []signature` : 가디언 서명 모음.

각 서명은 다음과 같습니다:

- `index u8` : 가디언 세트에서 해당 가디언의 인덱스.
- `signature [65]byte` : ECDSA 서명.

바디(Body) : 온체인 메시지에서 결정론적으로 도출됩니다. 동일한 메시지를 처리하는 가디언들은 동일한 바디를 도출해야 하며, 이를 통해 VAA와 메시지 간 1:1 관계를 유지하고 중복 처리(double-processing)를 피합니다.

- `timestamp u32` : 메시지가 게시된 블록의 타임스탬프.
- `nonce u32`
- `emitter_chain u16` : 메시지를 발생시킨 체인의 ID.
- `emitter_address [32]byte` : 코어 컨트랙트를 호출한(게시한) 컨트랙트 주소 (Wormhole 포맷).
- `sequence u64` : 해당 emitter가 게시한 메시지 수를 나타내는 자동 증가 정수.
- `consistency_level u8` : 이 emitter가 요구하는 일관성(파이널리티) 수준.
- `payload []byte` : 조치에 필요한 임의의 데이터 바이트.

바디의 결정론적 성질은 체인 상태가 파이널리티에 도달했을 때에만 엄밀히 성립합니다. 재조작이 발생해, 이전에 블록 X에 포함되었던 트랜잭션이 블록 Y로 대체되는 경우, 서로 다른 포크를 관찰하는 가디언이 동일한 메시지라고 믿는 것에 대해 서로 다른 VAA를 생성할 수 있습니다. 블록이 충분히 깊어지면 가능성은 낮아지지만, 더 빠른(덜 파이널한) 일관성 수준을 선택하면 이러한 위험이 증가합니다.

바디는 VAA를 처리·활용하는 컨트랙트나 기타 시스템에 유의미한 정보를 포함합니다.

`parseAndVerifyVAA` 같은 함수를 호출하면 바디가 반환되며, 이를 통해 `emitterAddress`를 검증해 VAA가 신뢰 가능한 컨트랙트에서 기원했는지 확인할 수 있습니다.

VAA는 목적지가 없으므로 사실상 멀티캐스트입니다. 네트워크 내 어떤 체인의 코어 컨트랙트라도 VAA의 진위를 검증합니다. 특정 목적지를 가지는 VAA인 경우, 릴레이어가 해당 전달을 적절히 완료할 책임을 집니다.

일관성과 파이널리티

일관성 수준은 가디언이 체인의 최종 커밋 상태를 기다릴지, 아니면 덜 파이널한 조건에서 더 빨리 VAA를 발행할지를 결정합니다. 즉각적 파이널리티가 없는 블록체인의 경우, 블록이 충분히 확정될 때까지 재조직 위험이 남아 있으므로 특히 중요합니다.

가디언 워처(watcher)는 각 블록체인을 실시간으로 모니터링하는 특화 프로세스입니다. 이들은 선택된 일관성 수준을 준수하여, 충분한 커밋 상태가 충족되었는지를 판단한 뒤 서명하고 VAA를 내보냅니다. 어떤 체인은 하나의 커밋 수준(사실상 파이널)만 허용하는 반면, 다른 체인은 거의-파이널 혹은 완전 파이널 중에서 선택할 수 있습니다. 더 빠른 옵션은 VAA 생성 속도를 높이지만 재조직 위험을 키우고, 더 보수적 옵션은 더 오래 걸리지만 롤백 가능성을 낮춥니다.

서명

VAA의 바디는 keccak256으로 두 번 해시되어 서명 대상 다이제스트가 생성됩니다.

페이로드 유형

웜홀 위에 구축된 다양한 애플리케이션은 VAA에 첨부되는 페이로드 형식을 별도로 정의할 수 있습니다. 이 페이로드는 대상 체인과 컨트랙트에 관한 정보를 담아, 예를 들어 특정 주소로 토큰을 민팅하는 등의 조치를 취할 수 있게 합니다.

토큰 전송

많은 브리지는 체인 간 토큰을 전송하기 위해 **락업/민트(lock/mint)** 및 **소각/언락(burn/unlock)** 메커니즘을 사용합니다. 웜홀의 범용 메시지 전달 프로토콜은 이러한 락업 및 소각 이벤트의 라우팅을 처리하여, 웜홀의 **Wrapped Token Transfer (WTT)**가 체인 불가지론적(chain-agnostic)으로 작동하며 어떤 네트워크에도 빠르게 통합될 수 있도록 합니다.

토큰을 송신 체인에서 목적지 체인으로 전송하기 위해서는 다음 단계가 필요합니다:

1. 송신 체인에서 토큰을 락업합니다.
2. 송신 체인은 토큰 락업이 완료되었음을 증명하는 메시지를 발행합니다.
3. 목적지 체인은 송신 체인에서의 락업 이벤트를 확인하는 메시지를 수신합니다.
4. 목적지 체인에서 해당 토큰이 민트됩니다.

송신 체인이 락업을 검증하기 위해 발행하는 메시지를 **전송 메시지(transfer message)** 라고 하며, 다음과 같은 구조를 가집니다:

- `payload_id u8` : 페이로드 ID. 토큰 전송의 경우 **1**로 설정해야 합니다.
- `amount u256` : 전송되는 토큰 수량.
- `token_address u8[32]` : 소스 체인상의 토큰 주소.
- `token_chain u16` : 소스 체인의 체인 ID.
- `to u8[32]` : 목적지 체인의 주소.
- `to_chain u16` : 목적지 체인의 체인 ID.
- `fee u256` : 릴레이어에게 지급되는 수수료.

이 구조에는 목적지 체인이 락업 이벤트를 이해하는 데 필요한 모든 정보가 포함되어 있습니다. 목적지 체인은 이 페이로드를 수신하면 해당 자산을 민트할 수 있습니다.

목적지 체인은 송신 측의 토큰이 어떻게 락업되었는지에 대해서는 알 필요가 없습니다. 해당 토큰이 민트 과정에서 소각되었거나, 커스터디 계정에 락업되었을 수도 있습니다. 가디언들이 해당 이벤트의 존재를 충분히 입증하면 프로토콜은 이를 릴레이합니다.

증명(Attestation)

목적지 체인은 송신 체인으로부터 락업 이벤트를 알리는 메시지를 신뢰할 수 있지만, 올바른 토큰이 실제로 락업되었는지 자체적으로 검증할 방법은 없습니다. 이를 해결하기 위해 WTT는 **토큰 증명(token attestation)** 기능을 지원합니다.

토큰 증명을 생성하기 위해 송신 체인은 토큰의 메타데이터를 포함한 메시지를 발행하며, 목적지 체인은 이를 사용하여 해당 토큰의 이름, 심볼(symbol), 소수점 자리수(decimals) 등을 보존할 수 있습니다.

토큰 증명 메시지 형식은 다음과 같습니다:

- `payload_id u8` : 페이로드 ID. 증명의 경우 **2**로 설정해야 합니다.
- `token_address [32]byte` : 원본 토큰 컨트랙트 주소.
- `token_chain u16` : 원본 토큰의 체인 ID.
- `decimals u8` : 토큰의 소수점 자리수.
- `symbol [32]byte` : 자산의 짧은 이름.
- `name [32]byte` : 자산의 전체 이름.

증명 관련 팁

토큰 증명 작업 시 다음 사항을 유의해야 합니다:

- 증명은 UTF8 토큰 이름과 심볼 데이터를 인코딩하기 위해 **고정 길이 바이트 배열**을 사용합니다. 이로 인해 멀티바이트 유니코드 문자가 잘릴 수 있습니다.
 - 증명 VAA를 전송할 때는, 문자가 잘리지 않는 UTF8 문자열의 가장 긴 접두사(prefix)를 보내고, 남은 공간은 0 바이트로 패딩하는 것이 좋습니다.
 - 증명 VAA를 파싱할 때는 모든 후행 0 바이트를 제거하고, 나머지 데이터를 UTF-8로 변환하는 것이 권장됩니다.
 - 체인마다 VAA 파서가 다를 수 있으므로, 긴 문자열 또는 잘못된 UTF8 문자가 포함된 경우 체인마다 이름/심볼이 다르게 표시될 수 있습니다.
 - 토큰의 소수점 자리수를 모르면 목적지 체인은 전송 시 올바른 수량의 토큰을 민트할 수 없습니다.
→ 따라서 WTT는 각 토큰 전송마다 **증명(Attestation)** 을 요구합니다.
-

메시지를 포함한 토큰 전송

메시지가 포함된 토큰 전송 구조는 토큰 전용 구조와 동일하지만, 다음과 같은 차이점이 있습니다:

- **fee 필드**: from_address 필드로 대체됩니다.
- **payload 필드**: 새로 추가되며, 임의의 바이트 데이터를 포함할 수 있습니다.
디앱은 이 필드에 애플리케이션 특화 데이터(application-specific data)를 포함하여 특정 동작을 유도할 수 있습니다.

이 VAA 유형은 이전에 **Contract Controlled Transfer** 로 알려졌으며, 종종 **payload3 메시지**라고도 불립니다.

해당 데이터 구조는 다음과 같습니다:

- payload_id u8 : 페이로드 ID. 메시지가 포함된 토큰 전송의 경우 **3**으로 설정해야 합니다.
 - amount u256 : 전송되는 토큰 수량.
 - token_address u8[32] : 소스 체인상의 주소.
 - token_chain u16 : 소스 체인의 체인 ID.
 - to u8[32] : 목적지 체인의 주소.
 - to_chain u16 : 목적지 체인의 체인 ID.
 - from_address u8[32] : 소스 체인에서 WTT를 호출한 주소.
 - payload []byte: 애플리케이션 특화 메시지 데이터.
-

거버넌스

거버넌스 VAA는 앞선 형식처럼 `payload_id` 필드를 가지지 않습니다. 대신 배포된 컨트랙트에서 특정 행동(예: 업그레이드)을 트리거합니다.

액션 구조

거버넌스 메시지는 사전에 정의된 액션을 포함하며, 온체인에 배포된 다양한 뮌 모듈을 대상으로 할 수 있습니다. 구조는 다음 필드를 포함합니다:

- `module u8[32]`: 모듈 식별자(오른쪽 정렬).
- `action u8`: 실행할 사전 정의된 거버넌스 액션.
- `chain u16`: 액션이 실행될 체인. (모든 체인의 경우 0으로 설정)
- `args any`: 액션에 전달되는 인자들.

거버넌스 액션 목록

각 숫자형 액션의 의미는 뮌 설계 문서에 정의되어 있습니다:

- [Core 거버넌스 액션](#)
- [WTT 거버넌스 액션](#)

메시지의 수명 주기

누구나 VAA를 대상 체인에 제출할 수 있습니다.

가디언들은 트랜잭션 수수료를 피하기 위해 보통 이 단계를 수행하지 않으며, 대신 뮌을 기반으로 구축된 애플리케이션이 가디언 RPC를 통해 VAA를 받아 별도의 흐름으로 제출합니다.

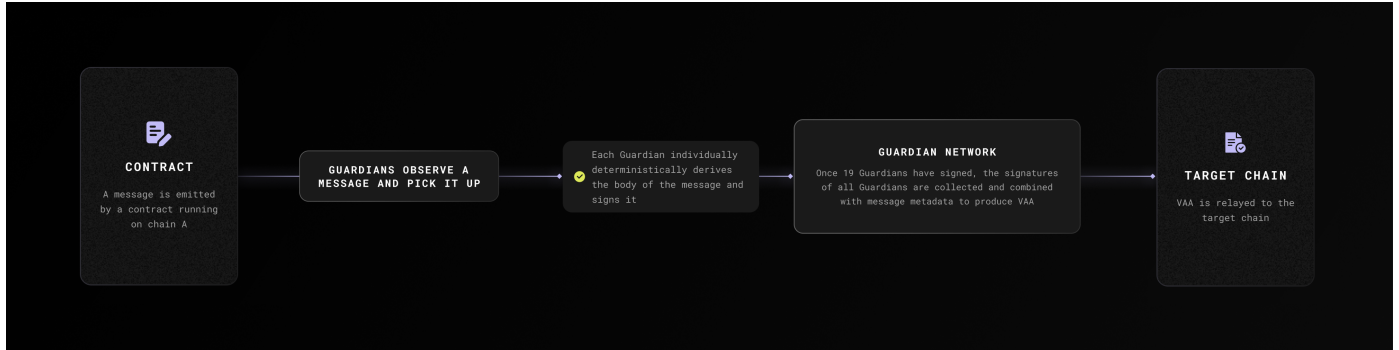
이제 개념이 정의되었으므로, 뮌 네트워크가 두 체인 간 메시지를 라우팅하기 위해 수행하는 전체 흐름을 단계별로 설명할 수 있습니다:

- 체인 A의 컨트랙트가 메시지를 발행함:**
가디언들은 모든 체인의 이벤트를 관찰하도록 프로그래밍되어 있습니다. 여기서 가디언은 단일 엔티티로 단순화되어 표시되지만, 실제로는 19명의 가디언 각각이 개별적으로 메시지를 관찰합니다.
- 서명 집계(Signature Aggregation):**
가디언들은 메시지를 독립적으로 관찰하고 서명합니다. 충분한 가디언이 메시지에 서명하면, 서명 모음은 메시지와 메타데이터와 결합되어 VAA를 생성합니다.

3. VAA 대상 체인 제출:

VAA는 가디언들이 메시지 페이로드의 존재를 집단적으로 증명했다는 것을 나타내는 증거로 작동합니다.

이 VAA는 릴레이를 통해 대상 체인에 제출되며, 수신 컨트랙트가 이를 처리하여 최종 단계를 완료합니다.



가디언(Guardians)

웜홀은 여러 블록체인의 상태를 모니터링하는 19개의 분산 노드 세트에 의해 운영됩니다.

이 노드들은 웜홀에서 **가디언(Guardians)** 이라고 불립니다.

현재 가디언 세트는 [대시보드](#)에서 확인할 수 있습니다.

가디언은 메시징 프로토콜에서 다음과 같은 역할을 수행합니다:

1. 각 가디언은 메시지를 관찰하고, 다른 가디언과 독립적으로 해당 페이로드에 서명합니다.
2. 가디언들은 독립적으로 생성한 서명을 결합하여 멀티시그(multisig)를 형성합니다.
3. 이 멀티시그는 웜홀 네트워크의 대다수가 동일한 상태를 관찰하고 동의했음을 증명합니다.

웜홀은 이러한 멀티시그를 [검증 가능한 액션 승인\(Verifiable Action Approvals, VAAs\)](#)이라고 부릅니다.

가디언 네트워크

가디언 네트워크는 웜홀의 탈중앙화 오라클 역할을 수행하며, 안전한 크로스체인 상호운용성을 보장합니다. 이 중요한 생태계 구성요소를 이해하면 웜홀 프로토콜의 작동 원리를 보다 잘 이해할 수 있습니다.

가디언 네트워크는 다음 **5가지 핵심 원칙**을 기반으로 설계되었습니다:

- **탈중앙화(Decentralization):** 네트워크의 제어권이 여러 주체에 분산되어 있습니다.

- **모듈성(Modularity)**: 독립적인 구성요소(예: 오라클, 릴레이어, 애플리케이션)를 통해 유연성과 업그레이드 가능성을 확보합니다.
 - **체인 불가지론(Chain Agnosticism)**: EVM, 솔라나, 그 외 다양한 블록체인을 지원하며, 단일 네트워크에 의존하지 않습니다.
 - **확장성(Scalability)**: 대규모 거래량과 고가치 전송을 처리할 수 있습니다.
 - **업그레이드 가능성(Upgradeable)**: 기존 모듈의 구현을 통합자에게 영향을 주지 않고 교체할 수 있습니다.
-

탈중앙화

탈중앙화는 상호운용성 프로토콜의 핵심 과제입니다.

과거 솔루션은 완전히 중앙화되어 있었으며, 새로운 모델조차 단일 주체 또는 소수 노드에 의존하는 경우가 많았습니다.

두 가지 일반적인 접근법에는 각각의 한계가 존재합니다:

- **지분증명(PoS)**: 탈중앙화의 대표 모델로 여겨지지만, 자체 스마트 계약 없이 여러 블록체인을 검증하는 네트워크에는 적합하지 않습니다.
보안성이 입증되지 않았으며, 설계 목표 달성을 어렵게 만드는 복잡성을 유발합니다.
- **영지식증명(ZKP)**: 신뢰 없는 탈중앙화 접근법을 제공하지만, 기술적으로 초기 단계에 있으며, 온체인 검증이 고비용이라 실질적 배포를 위해서는 여전히 멀티시그 기반의 보조 메커니즘이 필요합니다.

현재 대부분의 주요 블록체인은 소수의 전문 밸리데이터 기업에 의해 보호되고 있습니다.

이러한 기업들을 상호운용성을 위한 합의 메커니즘으로 통합할 수 있다면, 토큰 기반 네트워크보다 더 강력한 보안성과 성능을 확보할 수 있습니다.

웜홀의 설계적 선택

- **임계 서명(threshold signature)**은 이상적이지만, 대부분의 블록체인에서 아직 표준화되지 않았습니다.
검증 비용이 높고 복잡합니다.
- **t-Schnorr 멀티시그**가 현실적 대안입니다.
폭넓게 지원되며 검증 비용이 낮습니다. 그러나 서명자 수가 많을수록 가스비가 선형적으로 증가하므로, 적정 규모의 검증자 세트가 필요합니다.
- **19개의 검증자 세트**가 이상적인 절충점입니다.
2/3 합의 기준을 적용하면 13개의 서명만 검증하면 되므로, 가스비를 낮추면서도 강력한 보안을 유지할 수 있습니다.

- **토큰이 아닌 평판 기반 보안 모델** 을 채택했습니다.

익명 노드 대신 실체가 명확한, 업계에서 가장 신뢰받는 밸리데이터 기업들이 가디언 역할을 수행합니다.

이 구조는 모든 가디언이 동등한 지위를 갖는 **권위증명(PoA, Proof-of-Authority)** 합의 모델을 기반으로 합니다.

향후 임계 서명이 더 널리 지원되면 참여자 수를 늘릴 수 있으며, ZKP가 실용화되면 완전한 신뢰 불필요 (trustless) 모델로 진화할 수 있습니다.

모듈성

웜홀은 단일 목적에 최적화된 단순한 구성요소로 설계되었습니다.

보안 및 합의(가디언)와 메시지 전달([릴레이어](#))를 분리함으로써, 한 구성요소를 업그레이드하더라도 다른 구성요소에 영향을 주지 않습니다.

체인 불가지론

웜홀은 단순한 기술 구조(t-Schnorr 서명), 유연한 릴레이어 모델, 견고한 밸리데이터 네트워크 덕분에 현존하는 어떤 상호운용성 프로토콜보다 더 넓은 생태계를 지원합니다.

스마트 계약 런타임에 대한 [Core Contract](#) 가 개발되는 즉시, 새로운 생태계로 확장할 수 있습니다.

확장성 (Scalability)

웜홀은 높은 거래량과 TVL(총 예치금)을 처리할 수 있는 확장성을 갖추고 있습니다.

모든 가디언은 웜홀이 지원하는 각 블록체인의 **풀 노드(full node)** 를 직접 실행해야 합니다.

초기 설정은 무겁지만, 일단 동작하면 실제 연산 부하는 가볍습니다.

성능 병목은 가디언 네트워크 자체가 아니라, **기반 블록체인의 속도**에 의해 결정됩니다.

업그레이드 가능성

웜홀은 다음과 같은 방식으로 진화할 수 있도록 설계되었습니다:

- **가디언 세트 확장:** 향후 임계 서명 도입을 통해 가디언 수를 늘릴 수 있습니다.
- **ZKP 통합:** ZKP 기술이 널리 지원되면 완전한 신뢰 불필요 네트워크로 전환할 수 있습니다.

이러한 원칙들은 탈중앙화 컴퓨팅 전반을 포괄하는 완전한 신뢰 불필요 상호운용성 계층으로 발전할 수 있는 기반을 제공합니다.

스파이(Spy)

웜홀 생태계에서 **스파이**는 가디언 네트워크의 메시지를 모니터링하는 데몬(daemon, 지속 실행되는 백그라운드 프로세스)입니다.

스파이는 가디언과 달리 검증을 수행하지 않으며, 웜홀 네트워크 상의 메시지 트래픽을 관찰하여 애플리케이션과 사용자가 실시간 데이터에 접근할 수 있도록 합니다.

스파이의 주요 목적은 가디언 네트워크 전반에서 브로드캐스트되는 메시지를 구독(subscribe)하고, 이를 통해 통합자 및 애플리케이션이 합의 작업에 직접 참여하지 않고도 **실시간 네트워크 활동을 추적**할 수 있게 하는 것입니다.

주요 기능

- **실시간 메시지 모니터링:** 지원되는 체인 전반에서 게시되는 웜홀 메시지를 거의 실시간으로 관찰 가능.
- **필터 가능한 메시지 스트림:** 체인, emitter, 기타 기준별 필터링이 가능하여 특정 계약이나 이벤트만 추적 가능.
- **개발 친화적 통합:** gRPC 및 WebSocket 인터페이스를 제공해, 커스텀 툴·대시보드·인덱싱 서비스에 쉽게 통합 가능.
- **다중 메시지 프로토콜 지원:** WTT, CCTP, NTT 등 다양한 웜홀 메시징 프로토콜에서 발생하는 메시지를 관찰 가능.
- **경량 설계:** Validator 수준 인프라 없이 인덱싱 서비스나 백엔드 환경에 통합 가능.

통합자 사용 사례

스파이는 검증이나 합의에 직접 참여하지 않고도 가디언 네트워크의 실시간 활동을 관찰할 수 있는 메커니즘을 제공합니다.

이를 통해 통합자는 **VAAs, Observations, Guardian Heartbeats** 등의 멀티체인 이벤트를 추적할 수 있습니다.

이 모니터링 기능은 실시간 데이터 인사이트가 필요한 애플리케이션에 특히 유용합니다.

스파이를 운영함으로써 애플리케이션은 메시지 승인, 이벤트 발생, 가디언의 생존 신호를 즉시 인지할 수 있으며,

네트워크 리소스에 추가 부담을 주지 않고도 신속하고 반응적인 동작을 유지할 수 있습니다.

관찰 가능한 메시지 유형

스파이는 가십 프로토콜(gossip protocol)을 통해 공유되는 다음과 같은 유형의 메시지에 접근할 수 있습니다:

- **Verifiable Action Approvals (VAAs)**
멀티체인 데이터 패킷입니다.
 - 스파이는 가디언 네트워크에서 VAA가 승인되었는지를 감지할 수 있어, 실시간 멀티체인 검증이 필요한 애플리케이션에 매우 유용합니다.
 - **Observations**
웜홀의 코어 컨트랙트에서 발생하며, 가디언이 이를 수집해 네트워크 전반에 전달합니다.
 - 스파이는 이러한 메시지를 모니터링하여 블록체인 이벤트의 투명성과 가시성을 제공합니다.
 - **Guardian heartbeats**
가디언 노드의 상태를 나타내는 하트비트 메시지입니다.
 - 스파이는 하트비트를 모니터링하여 가디언의 생존(liveness) 및 네트워크 연결 상태를 감지할 수 있습니다.
-

릴레이어(Relayers)

이 섹션은 웜홀 네트워크 내 릴레이어의 역할, 유형, 그리고 멀티체인 프로세스를 촉진하는 이점에 대한 포괄적 가이드를 제공합니다.

릴레이(relaying)는 크로스체인 메시지, 특히 Verified Action Approval (VAA)을 출발 체인에서 대상 체인으로 전달하는 과정을 의미합니다.

멀티체인 애플리케이션에서 메시지가 출발 체인에서 발생하고 가디언의 서명을 받은 후, 해당 메시지를 대상 체인의 컨트랙트로 전달하는 역할이 바로 **릴레이어의 책임**입니다.

릴레이어는 신뢰가 필요하지 않습니다.

웜홀 메시지의 보안은 가디언 네트워크의 VAA 서명에서 비롯되며, 릴레이어는 이를 변경할 수 없습니다.

즉, 릴레이어는 메시지의 **내용(content)** 이나 **결과(outcome)** 에 영향을 줄 수 없으며, 오직 **전달 시점(availability)** 만 영향을 줄 수 있습니다.

따라서 개발자나 사용자는 릴레이어 서비스의 무결성을 신뢰할 필요가 없으며, 단지 메시지를 전달할 만큼 온 라인 상태로 유지되길 기대하면 됩니다.

기본 원칙

릴레이어는 뮌헨 네트워크 내에서 기본적으로 **신뢰 불필요(trustless)** 한 주체입니다. 즉, 운영에 있어 신뢰가 필요 없으며, 동시에 무조건 신뢰할 이유도 없습니다. 그들은 단순히 VAA를 출발지에서 목적지로 전달하는 역할을 수행하는 전달 메커니즘입니다.

- **누구나 메시지를 릴레이할 수 있음**

가디언은 서명된 VAA를 공개적으로 브로드캐스트하므로, 누구나 VAA를 가져와 대상 체인에 제출할 수 있습니다.

가디언 서명은 보편적 검증성을 제공하므로, 어떤 뮌헨 컨트랙트나 클라이언트도 이를 검증할 수 있습니다.

이 속성 덕분에 릴레이는 **퍼미션리스(permissionless)** 하며 **신뢰 불필요(trustless)** 한 프로세스로 유지됩니다.

한 릴레이어가 중단되더라도 다른 주체(사용자 포함)가 VAA를 대신 제출할 수 있습니다.

- **보안은 VAA에 내재됨**

가디언의 서명은 메시지의 진위를 보장합니다.

릴레이어는 추가적인 정보나 오프체인 데이터를 제공할 수 있지만, 컨트랙트는 반드시 **검증된 VAA** 또는 **온체인 데이터**만 신뢰해야 합니다.

따라서 릴레이어가 악의적으로 동작하더라도 보안을 침해할 수 없으며, 최악의 경우 전달 지연이 발생할 뿐 메시지가 조작되지는 않습니다.

- **사용자 경험 vs 인프라 고려**

릴레이어는 크로스체인 단계를 자동화하여 사용자 경험을 개선하지만, 수수료 및 인프라에 대한 고려가 필요합니다.

개발자는 외부 릴레이어 서비스에 의존하거나 자체 릴레이어를 운영할 수 있습니다.

뮌헨은 완전한 클라이언트 사이드(릴레이어 없음) 접근, 뮌헨 제공 릴레이 네트워크, 또는 커스텀 릴레이어 구축 중 하나를 선택할 수 있는 유연성을 제공합니다.

수동 릴레이 vs 자동 릴레이

뮌헨 메시징을 통합할 때 개발자는 **수동(클라이언트 사이드)** 릴레이와 **자동 릴레이** 중 하나를 선택해야 합니다.

두 방식의 차이는 VAA를 대상 체인에 전달하는 주체가 누구인지에 있습니다.

- **수동 릴레이(Manual Relaying)**

사용자가 직접 크로스체인 단계를 수행합니다.

예를 들어, 체인 A에서 VAA가 생성되면 사용자가 뮌헨 API나 Explorer를 통해 이를 가져와 체인 B에서 트랜잭션으로 제출해야 합니다.

추가 서버나 백엔드가 필요하지 않으며, 단순하지만 여러 번의 서명과 체인 간 자금 유지가 필요해 UX가 제한적입니다.

테스트, 데모, MVP에는 적합하지만, 프로덕션 환경에는 비효율적입니다.

- **자동 릴레이(Automated Relaying)**

릴레이어 서비스나 네트워크가 자동으로 VAA를 전달합니다.

사용자는 단일 액션으로 메시지를 전송할 수 있으며, 자산은 자동으로 대상 체인으로 전달됩니다.

자동 릴레이는 사용자 개입을 최소화해 UX를 극대화하지만, 서비스 수수료가 발생합니다.

두 가지 접근법이 있습니다:

- **커스텀 릴레이어 구축:** 직접 백엔드를 운영하여 VAAs를 청취 및 전달. 완전한 제어 가능하지만 인프라 유지 필요.
- **웜홀 릴레이 네트워크 활용:** 웜홀의 분산 릴레이 서비스를 이용해 간단한 온체인 통합으로 구현 가능.

항목	수동 릴레이(클라이언트 사이드)	자동 릴레이
VAA 전달 주체	사용자 또는 클라이언트 앱	릴레이어 서비스(커스텀/웜홀)
인프라 필요성	없음	백엔드 또는 웜홀 네트워크
사용자 경험	여러 서명, 수동 단계 필요	원클릭 자동 전달
비용 모델	대상 체인 트랜잭션 수수료만	서비스 수수료 + 가스비
신뢰성	사용자 수행 여부에 의존	릴레이어가 재시도 및 실행 보장
적합한 용도	테스트, MVP, 데모	프로덕션급 애플리케이션

웜홀 릴레이어 유형

자동 릴레이 도입을 간소화하기 위해 웜홀은 개발자를 위한 **릴레이어 인프라 및 API**를 제공합니다.

현재 웜홀은 세 가지 릴레이어 방식을 지원합니다:

- **실행자:** 퍼미션리스(permissionless) 네트워크 기반의 차세대 릴레이 프레임워크. 멀티체인 및 커스텀 요금 지원.
- **스탠다드 릴레이어:** 웜홀이 운영하는 EVM 체인용 릴레이 네트워크. 간단한 온체인 통합으로 즉시 사용 가능.
- **커스텀 릴레이어:** 애플리케이션이 직접 운영하는 릴레이 서비스. 최대 유연성 제공하지만 운영 부담이 높음.

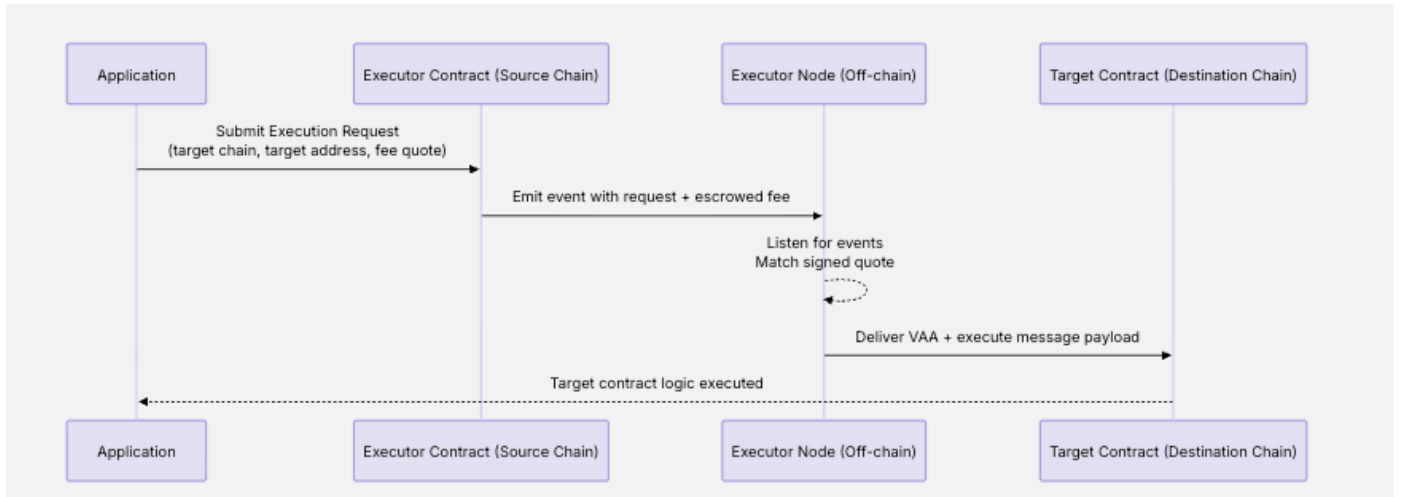
항목	실행자	스탠다드 릴레이어	커스텀 릴레이어
운영 주체	퍼미션리스 제공자 네트워크	웜홀 기여자	애플리케이션 팀
지원 체인	멀티체인(비EVM 포함)	EVM 한정	모든 웜홀 지원 체인
통합 방식	요청-견적(request-quote) 기반 실행자 계약	온체인 함수 호출	백엔드 서비스 직접 구현
인프라 필요성	없음 (온체인만)	없음 (온체인만)	풀 백엔드 필요 (24/7 가용성)
사용자 경험	넓은 체인 지원, 원활한 UX	EVM 전용, 간단한 UX	앱별 최적화 가능
트레이드 오프	초기 도입 중, 제한적 가용성	EVM 한정, 오프체인 로직 불가	높은 DevOps 부담, 보안 유지 필요

실행자

실행자는 웜홀의 차세대 크로스체인 실행 프레임워크로, EVM 이외의 체인까지 릴레이 기능을 확장하고 전달 과정에 유연성을 추가합니다.

실행자 시스템은 누구나 퍼미션리스 네트워크에서 **실행 제공자(Execution Provider)**로 참여할 수 있도록 하며, **요청-견적(request-quote)** 모델을 도입했습니다.

- 각 체인에 배포된 **실행자 계약**은 상태가 없고 퍼미션리스하며, 모든 사용자가 호출할 수 있습니다.
- 애플리케이션이 교차체인 메시지 전달을 요청하면, 실행자 계약은 요청을 기록하고 결제금을 에스스로 합니다.
- 오프체인 실행자 노드가 이 이벤트를 청취해 대상 체인으로 메시지를 전달합니다.
- 여러 제공자가 견적을 제시할 수 있어, 사용자는 경쟁력 있는 요금을 선택할 수 있습니다.



실행자는 완전한 탈중앙 메시지 전달 마켓플레이스를 구현하며, 개발자는 웜홀 코어 메시징 계약과 동일한 보안 모델(VAA 서명, 가디언 검증)을 유지하면서도 더 많은 체인과 커스텀 가격 정책을 활용할 수 있습니다.

기술 세부 사항은 [실행자 예시 구현 리포지토리](#)에서 확인할 수 있습니다.

스탠다드 릴레이어(Standard Relayer)

스탠다드 릴레이어는 웜홀이 운영하는 EVM 체인 전용 릴레이 네트워크입니다.

이 분산 네트워크는 소스 체인에서 메시지를 감지하고 자동으로 대상 체인으로 VAA를 전달합니다.

- 개발자는 백엔드를 운영할 필요가 없습니다.
- 소스 체인에서 `sendPayloadToEvm` 같은 함수를 호출하여 대상 체인과 수수료를 지정하면 됩니다.
- 대상 체인의 컨트랙트는 `IWormholeReceiver` 인터페이스를 구현해 메시지를 처리합니다.

장점

- 인프라 유지 필요 없음
- 빠른 온체인 통합
- 가디언 서명 기반 보안 유지

제한점

- 모든 연산은 온체인에서 수행해야 함 (오프체인 로직 불가)
- 솔라나, Sui 등 비EVM 체인 미지원
- 대상 체인 가스비 및 서비스 수수료 필요

커스텀 릴레이어(Custom Relayer)

특수한 요구사항이나 완전한 제어가 필요한 프로젝트는 **커스텀 릴레이어**를 구축할 수 있습니다. 이는 백엔드 서비스를 직접 운영하며 웜홀 네트워크의 VAA를 수신해 전달하는 방식입니다.

주요 장점

- 조건부 로직, 메시지 배치, 외부 데이터 기반 트리거 가능
- 가스 최적화 및 커스텀 인센티브 구조 설정 가능
- 웜홀에서 미지원 체인에 유연하게 대응 가능

트레이드오프

- 24/7 가용성을 위한 서버 유지 필요
- 지속적인 모니터링 및 보안 관리 필요
- 교차체인 수수료 관리 복잡

웜홀은 개발 편의를 위해 릴레이어 엔진(Relayer Engine)을 제공합니다.

이 도구는 가디언 리스닝, 메시지 파싱, 재시도 로직을 자동화하여, 개발자가 애플리케이션 로직(예: 필터링, 멀티체인 전달, 오프체인 검증)에만 집중할 수 있도록 합니다.

제품

웜홀은 자산 전송, 데이터 조회, 거버넌스 등을 여러 블록체인 생태계 전반에서 원활하게 수행할 수 있도록 하는 **모듈형 크로스체인 솔루션**을 제공합니다.

각 구성 요소는 가디언이 보안하는 동일한 메시징 레이어 위에 구축되어 있으며, 필요에 따라 단독 혹은 조합하여 사용할 수 있습니다.

전송 제품군 (Transfer Products)

웜홀은 다양한 사용 사례와 통합 요구사항에 맞춰 여러 가지 자산 전송 방식을 제공합니다:

- **네이티브 토큰 전송(Native Token Transfers, NTT)**
래핑 자산 없이 프로젝트의 네이티브 토큰을 체인 간 전송합니다.
여러 네트워크에서 동일한 자산의 기능과 대체성(fungibility)을 유지해야 하는 경우에 적합합니다.

- **래핑 토큰 전송(Wrapped Token Transfers, WTT)**

소스 체인에서 잠그고(lock), 목적지 체인에서 래핑 자산을 민팅(mint)하는 브리징 솔루션입니다. 크로스체인 유동성 확보와 메시지 전송이 필요한 프로젝트에 적합합니다.

- **결제 프로토콜(Settlement)**

인텐트 기반의 빠른 멀티체인 전송 프로토콜로, 전통적인 브리징 방식에 의존하지 않고 유동성과 상호 운용성을 극대화합니다.

항목	NTT	WTT	Settlement
지원 전송 유형			
토큰 전송	✓	✓	✓
메시지 포함 전송	✓	✓	✓
지원 자산 유형			
래핑 자산	✗	✓	✓
네이티브 자산	✓	✗	✓
특징			
기본 제공 UI	✗	✓	✓
이벤트 기반 동작	✓	✓	✗
인텐트 기반 실행	✗	✗	✓
빠른 결제	✗	✗	✓
요구사항			
스마트 계약 배포 필요	✓	✓	✓

자세한 비교는 [토큰 전송 개요\(Token Transfers Overview\)](#) 페이지를 참고하세요.

브리징 UI (Bridging UI)

커넥트는 미리 구축된 브리징 UI로, 크로스체인 토큰 전송을 빠르고 간단히 통합할 수 있습니다. 스마트 계약 수정 없이 손쉽게 브리징 기능을 제공하려는 프로젝트에 적합합니다.

실시간 데이터 (Real-Time Data)

쿼리는 여러 네트워크의 온체인 데이터를 조회할 수 있는 서비스입니다. 멀티체인 데이터 분석, 리포트, 데이터 집계 기능이 필요한 애플리케이션에 유용합니다.

멀티체인 거버넌스 (Multichain Governance)

MultiGov는 단일 메커니즘으로 여러 체인에 걸친 프로토콜 거버넌스를 관리할 수 있는 통합 프레임워크입니다. 여러 체인에 분산된 거버넌스와 업그레이드를 효율적으로 관리하려는 프로젝트에 적합합니다.

토큰 전송 (Token Transfers)

웜홀 토큰 전송 개요 (Token Transfers Overview)

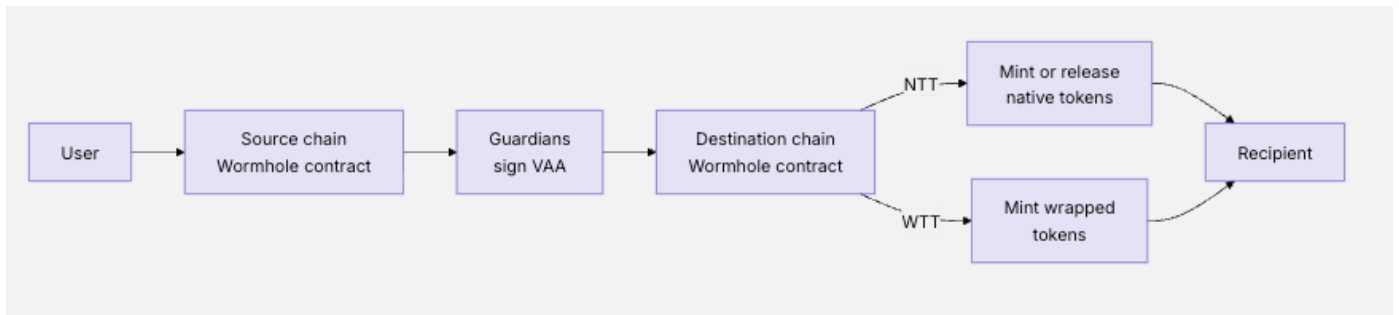
웜홀의 토큰 전송 기능은 체인 간 자산 이동을 간단하고 안전하게 수행할 수 있게 합니다. 개발자는 NTT(네이티브 토큰 전송) 또는 WTT(래핑 토큰 전송) 중 하나를 선택할 수 있으며, 두 방식 모두 가디언의 서명된 메시지(VAA)를 기반으로 작동하며 동일한 크로스체인 메시징 레이어를 사용합니다.

토큰 전송의 작동 방식 (How Token Transfers Work)

NTT와 WTT는 모두 가디언이 서명한 메시지(VAA)를 통해 체인 간 안전한 자산 전송을 보장합니다. 두 방식의 차이는 목적지 체인에서 자산이 어떻게 표현되는가에 있습니다.

1. 사용자가 소스 체인의 웜홀 계약에 토큰을 전송합니다.
2. 계약은 이벤트를 발생시키고, 가디언들이 이를 서명하여 VAA를 생성합니다.
3. 이 VAA가 대상 체인으로 제출됩니다.
4. 전송 방식에 따라 다음과 같이 처리됩니다:

- **NTT**: 에스크로에서 토큰을 해제하거나 새로운 네이티브 토큰을 민팅합니다.
- **WTT**: 래핑된 토큰을 수신자의 지갑에 민팅합니다.



NTT와 WTT 비교 (Choosing Between NTT and WTT)

웜홀은 두 가지 크로스체인 자산 전송 모델을 제공합니다.

NTT는 유연성과 업그레이드 가능성을 중시하며,

WTT는 빠르고 간편한 무허가형(permissionless) 통합을 제공합니다.

항목	NTT (네이티브 토큰 전송)	WTT (래핑 토큰 전송)
적합한 사용 사례	DeFi 거버넌스, 네이티브 자산 멀티체인 유동성, 스테이블코인, 기관용 서비스, 자체 거버넌스 기반 프로젝트	소비자 앱, 게임, 래핑 토큰 기반 유동성, 빠른 브리징 솔루션
전송 메커니 즘	소각 및 재발행(burn-mint) 또는 허브-스포크 구조	잠금 및 민팅(lock-mint)
보안 설정	속도 제한, 일시정지, 접근 제어, 임계치 서명 등 구성 가능. Global Accountant 통합	사전 구성된 속도 제한 및 Global Accountant 통합
계약 소유권	사용자가 각 체인의 계약 소유권 및 업그레이드 권한을 유지	웜홀 거버넌스에 의해 관리, 래핑 토큰 계약의 소유권은 이전 불가
토큰 계약 구조	프로토콜이 직접 소유한 네이티브 토큰 계약. 동일한 토큰을 모든 체인에서 유지	목적지 체인에 새로운 래핑 버전을 생성하는 WTT 계약
통합 난이도	고급 배포를 위한 유연한 프레임워크 제공 (선택 사항: EVM용 NTT Launchpad 사용 가능)	단순하고 무허가형 통합
사용자 경험	모든 체인에서 동일한 토큰 사용 가능	래핑 자산은 Explorer 메타데이터 업데이트 가 필요할 수 있음
예시	NTT 커넥트 , NTT TypeScript SDK	Portal Bridge UI

네이티브 토큰 전송 (Native Token Transfers)

개요 (Overview)

네이티브 토큰 전송 개요 (Native Token Transfers Overview)

네이티브 토큰 전송(NTT)은 네이티브 토큰을 다양한 블록체인 간에 전송할 수 있는 유연한 프레임워크를 제공합니다.

기존의 래핑 자산과 달리, NTT는 각 체인에서 토큰의 네이티브 속성을 유지합니다.

이를 통해 메타데이터, 소유권, 업그레이드 가능성, 사용자 정의 기능과 같은 중요한 측면을 완전히 제어할 수 있습니다.

주요 기능 (Key Features)

- **제어 및 사용자 정의:**
소유권과 구성 가능한 접근 제어, 권한, 임계값을 보장하여 무단 호출을 방지합니다.
- **고급 전송 속도 제한:**
체인 및 기간별로 전송 속도 제한을 설정하여 오용을 방지하고, 네트워크 혼잡을 관리하며, 배포를 제어합니다.
- **글로벌 회계 관리자(Global Accountant):**
체인에서 소각되고 전송된 토큰의 양이 발행된 토큰의 양을 초과하지 않도록 보장합니다.
- **래핑 토큰 없음:**
토큰이 네이티브 생태계 내에서 직접 사용되어, 중간 전송 단계를 제거합니다.

배포 모델 (Deployment Models)

NTT는 기존 토큰에 대해 두 가지 운영 모드를 제공합니다:

- **허브 앤 스포크(Hub-and-spoke):**
중앙 “허브” 체인에 토큰을 잠그고 “스�포크” 체인에서 동일한 양을 민팅하여, 허브에 총 공급량을 유지합니다.
이는 기존 토큰을 원래 계약을 변경하지 않고 새로운 블록체인에 통합하기에 이상적입니다.
- **소각 및 발행(Burn-and-mint):**
소스 체인에서 토큰을 소각하고 목적지 체인에서 새로 발행하여, 총 공급량을 여러 체인에 분산시킵니다.
이는 완전한 네이티브 멀티체인 토큰을 위해 기존 계약을 업그레이드할 의향이 있는 새로운 토큰 배포에 가장 적합합니다.

지원되는 토큰 표준 (Supported Token Standards)

네이티브 토큰 전송은 주로 ERC-20 토큰, 즉 이더리움 및 기타 EVM 호환 체인에서 가장 널리 사용되는 대체 가능 자산 표준을 지원하며, 크로스체인 전송 시 소스 체인에서 소각이 필요한 경우 ERC-20 Burnable 토큰도 지원합니다. 또한, SVM(Solana Virtual Machine) 지원 체인에서 대체 가능한 SPL 토큰을 지원하여 안전한 크로스체인 전송이 가능합니다.

NttManager는 지원되는 블록체인 간에 네이티브 토큰의 안전하고 신뢰할 수 있는 전송을 관리하는 계약입니다.

이 계약은 표준 **IERC20 인터페이스**와 OpenZeppelin의 **SafeERC20** 라이브러리를 활용하여, 체인 간 안전한 토큰 상호작용을 보장합니다.

NTT는 현재 비대체성 토큰(NFT) 또는 ERC-1155와 같은 멀티 토큰 표준을 지원하지 않습니다. 지원은 ERC-20 토큰에 한정됩니다.

배포 절차 (Deployment Process)

NTT를 배포할 때 포함되는 주요 단계는 다음과 같습니다:

- **토큰 준비:** ERC-20 또는 SPL 토큰이 준비되어 있는지 확인합니다.
- **배포 모델 선택:** 소각 및 발행 또는 허브 앤 스포크 모델 중 하나를 선택합니다.
- **배포 도구 선택:** [NTT Launchpad](#) (EVM 체인용) 또는 [NTT CLI](#)를 사용합니다.
- **초기화:** 대상 체인과 토큰 세부 정보를 지정하고, CLI를 사용하는 경우 환경을 설정합니다.
- **계약 배포:** NTT 매니저 계약을 모든 선택된 체인에 배포하고, 트랜잭션을 확인하며 가스비를 지불합니다.
- **설정 마무리:** 민팅 권한을 부여하고, 전송 속도 제한을 구성하며, 피어 매니저 연결(`setPeer` / `set_peer`를 통해; 로컬 구성, 크로스체인 메시지 없음)을 설정하고, 관리 역할을 지정합니다.
- **모니터링 및 유지관리:** 배포를 검증하고, [Global Accountant](#)를 통해 총 공급량을 모니터링하며, 필요에 따라 구성을 조정합니다.

사용 사례 (Use Cases)

- **크로스체인 스왑 및 유동성 통합**
 - **Native Token Transfers:** 네이티브 자산을 체인 간 전송합니다.

- **커넥트**: 사용자 친화적인 자산 전송을 관리합니다.
 - **쿼리**: 최적의 거래 실행을 위한 실시간 가격을 가져옵니다.
 - **체인 간 대출 및 차입**
 - **Native Token Transfers**: 담보를 네이티브 자산으로 이동합니다.
 - **메세징**: 대출 요청 및 청산을 체인 간 전파합니다.
 - **쿼리**: 실시간 금리 및 자산 가격을 가져옵니다.
 - **가스 추상화(Gas Abstraction)**
 - **Native Token Transfers**: 가스비 지불을 위한 네이티브 토큰 변환을 촉진합니다.
 - **메세징**: 가스비 결제 메시지를 체인 간 전송합니다.
 - **크로스체인 결제 위젯**
 - **Native Token Transfers**: 직접적인 네이티브 자산 전송을 보장합니다.
 - **커넥트**: 다양한 토큰으로 원활한 결제를 지원합니다.
 - **크로스체인 스테이킹**
 - **Native Token Transfers**: 스테이킹된 자산을 네트워크 간 네이티브하게 전송합니다.
 - **메세징**: 스테이킹 보상 및 거버넌스 신호를 체인 간 이동시킵니다.
-

NTT 아키텍처 (NTT Architecture)

원활 생태계 내의 **네이티브 토큰 전송(NTT)** 아키텍처는 여러 블록체인 간에 안전하고 효율적인 토큰 전송을 위한 견고한 프레임워크를 제공합니다.

이 아키텍처는 매니저(Manager)와 트랜시버(Transceiver)라는 핵심 구성 요소에 의존하며, 이들이 함께 작동하여 크로스체인 통신 및 토큰 작업의 복잡성을 관리합니다.

시스템 구성 요소 (System Components)

NTT 프레임워크는 매니저(Manager)와 트랜시버(Transceiver)로 구성되며, 매니저는 전송 프로세스를 관리하고, 트랜시버는 크로스체인 메시징을 처리하여 원활하고 안정적인 토큰 전송을 보장합니다.

매니저 (Managers)

매니저는 서로 다른 블록체인 간의 토큰 전송 흐름을 관리하며, 소스 체인에서 토큰이 목적지 체인에서 발행되거나 해제되기 전에 잠기거나 소각되도록 보장합니다.

매니저의 주요 업무는 전송 속도 제한, 메시지 인증(메시지 증명), 그리고 여러 트랜시버 간의 상호작용 관리입니다.

각 매니저는 특정 토큰에 할당되지만 여러 체인에서 작동할 수 있습니다.

매니저의 핵심 책임은 소스 체인에서 토큰이 안전하게 잠기거나 소각된 후에 목적지 체인에서 발행 또는 해제되도록 하는 것입니다. 이 과정은 전송의 무결성을 보장하고 이중 지출을 방지합니다.

매니저의 책임은 다음과 같습니다:

- **토큰 전송 흐름 처리:**

전송 요청 시 `NttManager`는 구성에 따라 토큰을 잠그거나 소각하고, `TransferSent` 이벤트를 발생시킵니다. 소스 체인에서 목적지 체인으로 릴리스되기 전에 토큰에 접근할 수 없도록 하여 이중 지출을 방지하고 안전한 전송을 유지합니다.

- **전송 속도 제한 (Rate-limiting):**

`NttManager` 계약에는 네트워크 과부하나 대상 체인 트래픽 폭주를 방지하기 위한 전송 속도 제한 기능이 포함되어 있습니다. `NttManager`는 전송 흐름을 관리하고 네트워크 혼잡을 방지하기 위해 제한을 적용합니다. 제한은 송신 및 수신 전송 모두에 적용됩니다.

- **아웃바운드:** 제한을 초과하는 전송은 (만약 `shouldQueue`가 `true`라면) 대기열에 추가되거나 되돌려집니다.
- **인바운드:** 유사한 제한이 대상 체인에 적용되어, 용량이 초과되면 전송이 지연됩니다.

전송 속도 제한 기간과 대기열은 체인별로 사용자 지정이 가능하며, 한도 초과 시 이벤트를 통해 사용자에게 알림을 제공합니다.

- **메시지 인증 검증 (Message authenticity verification):**

`NttManager`는 여러 트랜시버로부터 다중 인증(Attestation)을 수집하여 전송 메시지의 진위를 확인함으로써 전송 보안을 보장합니다. 각 전송마다 트랜시버로부터 임계 수의 서명(Attestation)을 수집해야 하며, 검증이 완료되면 `NttManager`는 목적지 체인에서 토큰을 해제하여 인증된 전송만 처리되도록 보장합니다.

- **트랜시버와의 상호작용 (Interaction with transceivers):**

`NttManager`는 트랜시버와 협력하여 체인 간 전송 메시지를 전달하고 검증을 처리합니다. 트랜시버는 전송 세부 정보가 포함된 메시지를 목적지 체인으로 전달하며, `NttManager`와 협력하여 토큰이 잠기거나 소각된 후에만 릴리스되도록 합니다. 트랜시버는 다양한 보안 프로토콜과 연동될 수 있어 유연성을 제공합니다.

트랜시버 (Transceivers)

트랜시버는 서로 다른 블록체인 간의 메시지 전송을 담당하며, 매니저와 함께 작동하여 체인 간 토큰 전송이 정확하게 이루어지도록 보장합니다. 그들의 주요 역할은 전송 프로세스와 관련된 메시지가 올바르게 전달되고 토큰이 안전하게 체인 간 전송되도록 하는 것입니다.

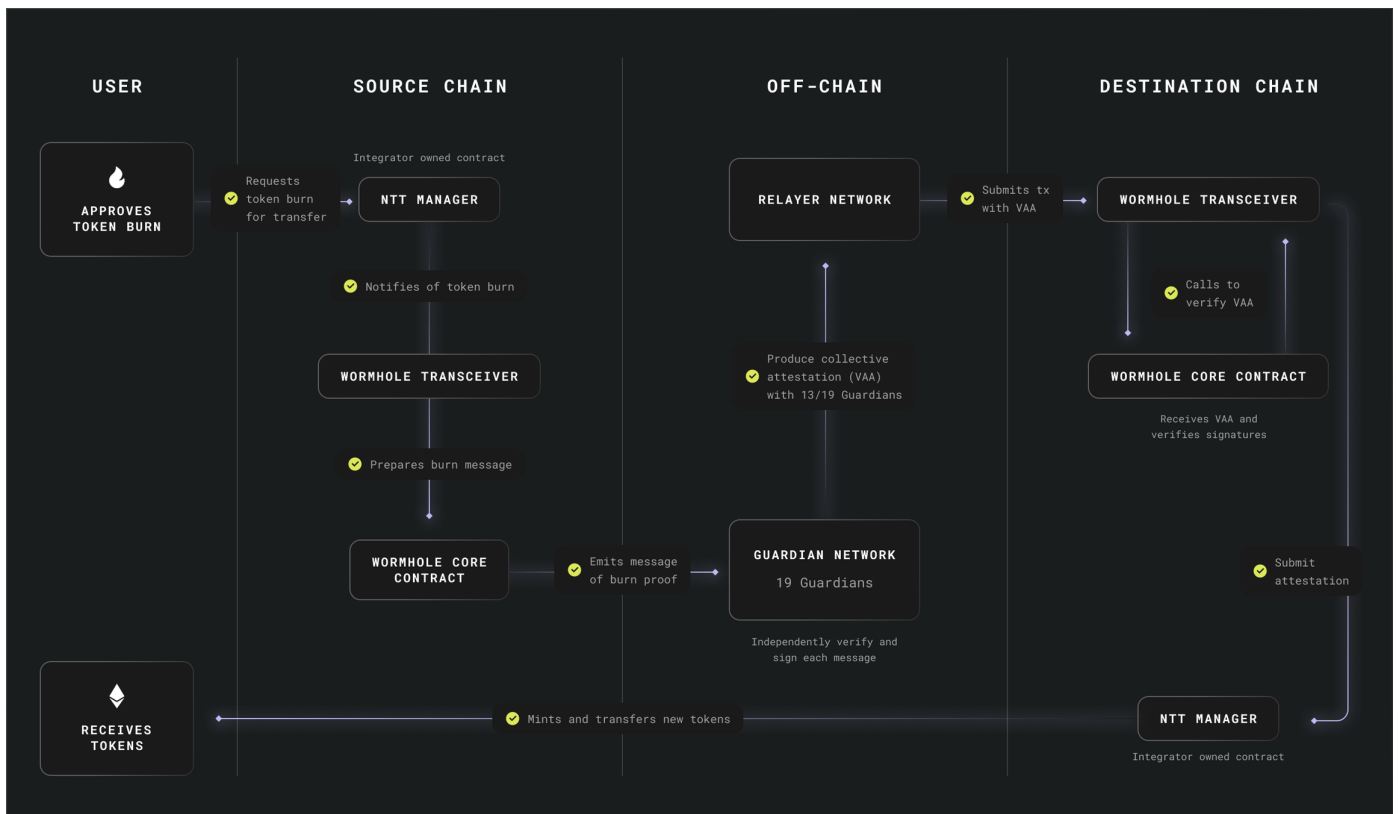
트랜시버는 원홀 생태계와 밀접하게 작동하지만, 원홀 코어 시스템과 독립적으로 구성될 수도 있어 높은 유연성을 제공합니다. 이러한 적응성 덕분에 트랜시버는 다양한 검증 백엔드와 통합될 수 있으며, 각 체인의 보안 요건에 맞게 구성할 수 있습니다.

트랜시버의 주요 역할은 다음과 같습니다:

- **메시지 전송:**
트랜시버는 체인 간 전송 메시지를 라우팅합니다. 전송이 시작되면, 트랜시버는 수신자 및 금액 등의 전송 세부 정보를 포함하는 메시지를 목적지 체인의 매니저에게 전송하여 검증 및 처리를 수행합니다.
- **매니저 조정:**
트랜시버는 매니저와 협력하여, 소스 체인에서 토큰이 잠기거나 소각된 후에만 목적지 체인에서 발행 되도록 보장함으로써 전송 보안을 강화합니다.
- **맞춤형 검증(Custom verification) 지원:**
트랜시버는 커스텀 검증 백엔드와 통합할 수 있어, 다양한 보안 프로토콜이나 체인 요구사항에 맞게 조정할 수 있습니다. 이를 통해 프로토콜은 필요한 경우 다른 인증 표준을 사용할 수 있습니다.

작동 방식:

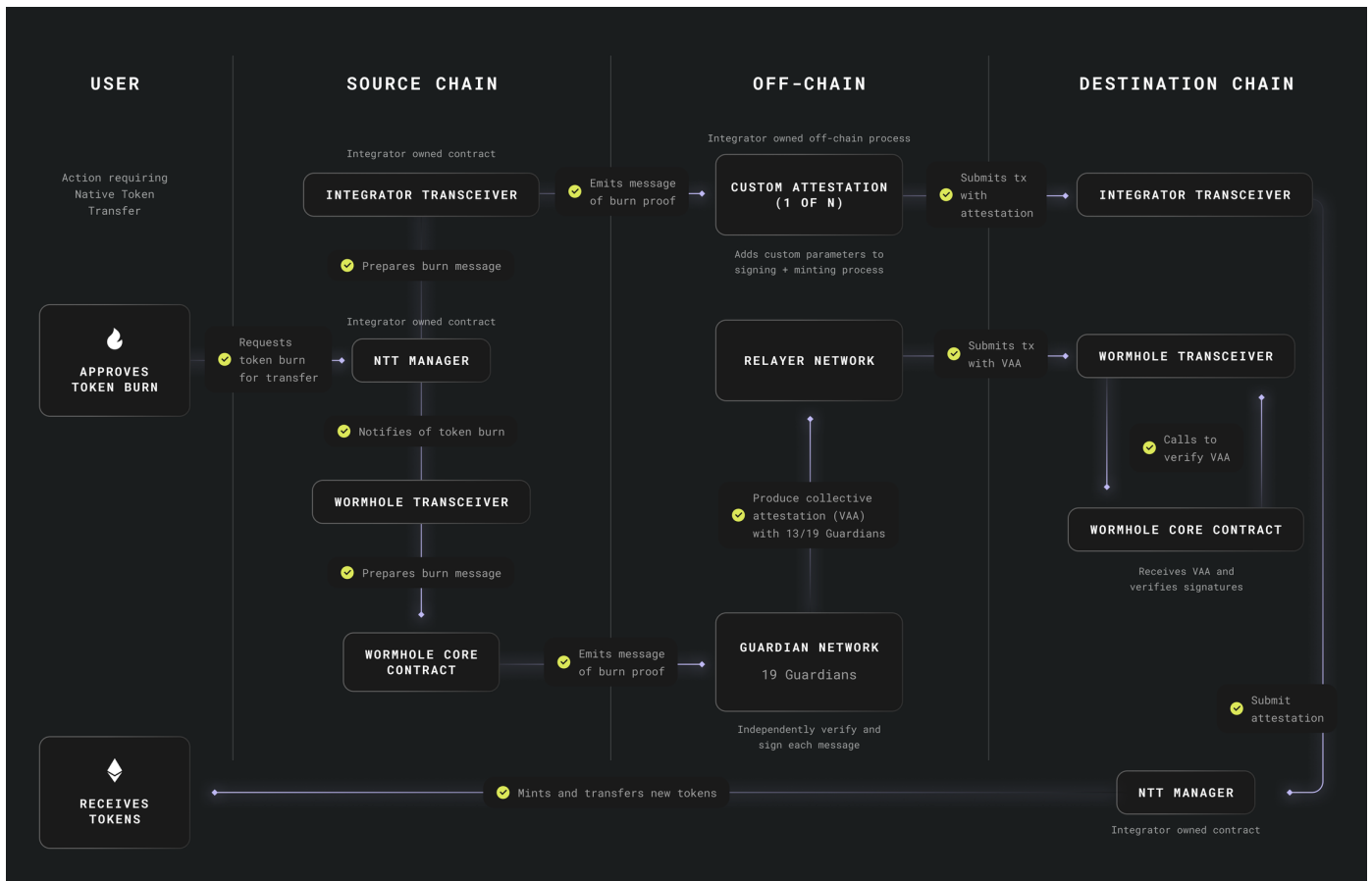
1. 트랜시버는 매니저로부터 체인 간 메시지 전송 지시를 받습니다.
2. 트랜시버는 전송 수수료를 산정하고, 메시지를 중계(relay)하며, 전송된 메시지가 안전하게 처리되었는지를 검증합니다.
3. 각 메시지에 대해 트랜시버는 매니저와 협력하여, 승인된 전송만이 목적지 체인에서 처리되도록 보장합니다.



커스텀 트랜시버 (Custom Transceivers)

NTT 프레임워크는 맞춤형 메시지 검증과 같은 고급 기능을 지원하여 보안성과 적응성을 강화합니다. 아키텍처에는 전송 시작, 속도 제한 관리, 토큰 작업 완료 등의 세부 프로세스가 포함되어 있으며, EVM 호환 체인 및 SVM 호환 체인에 대한 구체적인 지침과 이벤트가 제공됩니다.

NTT는 웜홀 가디언 메시지 검증 외에도 커스텀 메시지 검증을 지원할 수 있는 유연성을 가지고 있습니다. 커스텀 검증자는 트랜시버 계약으로 구현되며, 프로토콜별 또는 제3자 검증자에 의해 제공될 수 있습니다. 또한, 프로토콜은 토큰 전송을 유효로 간주하기 위한 인증 임계값(예: 2/2, 2/3, 3/5)을 구성할 수 있습니다.



검증자는 미리 정의된 기준에 따라 검사를 수행하고, 요건을 충족하는 트랜잭션에 대한 승인을 발급합니다. 이 승인은 웜홀 메시지에 통합되어, 웜홀 가디언 네트워크와 추가 검증자 모두에 의해 검증된 트랜잭션만 처리되도록 합니다. 이 모델은 브리징 과정에 추가 검증 단계를 포함하여 보안을 강화하고 트랜잭션 무결성에 대한 추가적인 보장을 제공합니다.

자세한 정보, 협업 또는 커스텀 트랜시버 예시에 대한 문의는 웜홀 기여자에게 연락하십시오.

온체인 상태 (On-Chain State)

NTT 계약은 최소한의 온체인 상태를 유지하여
전송을 안전하게 라우팅하고, 재실행을 방지하며,
여러 체인 간의 처리량을 관리합니다.

이 상태는 주로 NTT 매니저, 속도 제한기(Rate Limiter), 트랜시버 레지스트리에 의해 관리됩니다:

- **메시지 인증 기록(Message attestations):**
각 크로스체인 메시지에 대해 어떤 트랜시버가 인증했는지를 기록하고, M-of-N 인증 임계값을 적용하며, 이미 처리된 메시지의 재실행을 방지합니다.
- **피어 등록(Peer registrations):**
각 원격 체인을 관련된 NTT 매니저 및 토큰 소수점 구성과 매핑하여, 신뢰할 수 있는 피어만 토큰을 민팅하거나 해제할 수 있도록 보장합니다.
- **전송 속도 제한(Rate limiting):**
인바운드 및 아웃바운드 처리량 제한을 적용하고, 한도가 초과되면 전송을 대기열로 보내 유동성과 하위 네트워크를 보호합니다.
- **트랜시버 레지스트리(Transceiver registry):**
등록된 트랜시버 목록과 활성 상태를 유지하며, 거버넌스가 메시징 공급자를 추가하거나 제거할 수 있도록 합니다.

메시지의 수명 주기 (Lifecycle of a Message)

웜홀 생태계에서 네이티브 토큰 전송(NTT)의 메시지 수명 주기는 안전하고 정확한 크로스체인 토큰 전송을 보장하기 위한 여러 단계를 포함합니다. 이 수명 주기는 사용 중인 블록체인에 따라 다를 수 있으며, 다음 설명은 EVM 및 SVM 구현에 초점을 맞춥니다. 핵심 단계에는 전송 시작, 전송 속도 제한 처리, 메시지 송수신, 마지막으로 목적지 체인에서 토큰 민팅 또는 잠금 해제가 포함됩니다.

전송 (Transfer)

프로세스는 클라이언트가 전송을 시작할 때 시작됩니다.

EVM에서는 `transfer` 함수를 사용하고, SVM에서는 프로그램이 잠금(locking) 또는 소각(burning) 모드인지에 따라 `transfer_lock` 또는 `transfer_burn` 명령을 사용합니다.

클라이언트는 전송 금액, 수신자 체인 ID, 수신자 주소, 그리고 전송이 속도 제한에 걸렸을 때 대기열에 추가할지를 결정하는 플래그(`should_queue` — EVM 및 SVM 모두)를 지정합니다.

두 경우 모두 다음과 같습니다:

- 소스 체인이 잠금 모드인 경우, 토큰은 소스 체인에서 잠기며, 목적지 체인에서 잠금 해제됩니다.
- 소스 체인이 소각 모드인 경우, 토큰은 소스 체인에서 소각되고, 목적지 체인에서 새로 발행됩니다.

전송이 시작되면, EVM의 `TransferSent` 이벤트나 SVM의 해당 로그와 같은 이벤트가 발생하여 전송 프로세스의 시작을 알립니다.

전송 속도 제한 (Rate Limit)

EVM과 SVM 모두 남용 또는 네트워크 과부하를 방지하기 위해 전송 속도 제한(rate-limiting)을 구현합니다.

속도 제한은 소스 체인과 목적지 체인 모두에 적용됩니다.

현재 용량을 초과하는 경우, `shouldQueue` 플래그가 `true`로 설정되어 있으면 대기열에 추가될 수 있습니다.

- **EVM의 경우:**

전송이 속도 제한에 도달하면 구성된 속도 제한 지속 시간에 해당하는 지연과 함께 아웃바운드 큐에 추가됩니다.

`shouldQueue`가 `false`로 설정된 경우, 전송은 오류와 함께 되돌려집니다.

- **SVM의 경우:**

전송은 `insert_into_outbox` 메소드를 통해 Outbox에 추가되며,

속도 제한에 도달하면 `release_timestamp`와 함께 대기열에 등록됩니다.

`shouldQueue`가 `false`인 경우, `TransferExceedsRateLimit` 오류와 함께 전송이 되돌려집니다.

두 체인 모두 전송이 속도 제한에 걸리거나 대기열에 추가될 때 이벤트 또는 로그를 발생시킵니다.

송신 (Send)

트랜시버로 전달된 후 메시지는 체인 간 전송됩니다. 트랜시버는 토큰 전송 세부 정보를 포함한 메시지를 전달할 책임이 있습니다. 트랜시버의 구현 방식에 따라 메시지는 웜홀 릴레이어나 기타 사용자 지정 릴레이 솔루션을 통해 라우팅될 수 있습니다. 메시지가 전송되면, 성공적인 전송을 나타내는 이벤트가 발생합니다.

- **EVM의 경우:**

메시지는 `sendMessage` 함수를 사용하여 전송되며, 이는 트랜시버의 구현에 따라 송신을 처리합니다.

트랜시버는 웜홀 릴레이어나 커스텀 릴레이 솔루션을 사용하여 메시지를 전달할 수 있습니다.

- **SVM의 경우:**

전송 메시지는 Outbox에 저장된 후 `release_outbound` 명령을 통해 릴리스됩니다.

웜홀 트랜시버와 같은 SVM 트랜시버는 가디언이 검증을 위해 관찰하는 `post_message` 명령을 사용하여 메시지를 전송할 수 있습니다.

두 경우 모두, 메시지가 전송되었음을 나타내기 위해 EVM의 `SendTransceiverMessage` 이벤트나 SVM의 유사한 로그가 발생합니다.

수신 (Receive)

목적지 체인에서 메시지를 수신하면, 오프체인 릴레이어가 메시지를 목적지 트랜시버로 전달하여 검증을 수행합니다.

- **EVM의 경우:**

메시지는 목적지 체인의 `NttManager`에서 수신되며, 메시지의 진위가 검증됩니다.

인증 프로세스에 대해 설정된 `M-of-N` 임계값에 따라, 메시지는 여러 트랜시버의 인증이 필요할 수 있습니다.

- **SVM의 경우:**

메시지는 원홀 트랜시버 프로그램의 `receive_message` 명령을 통해 수신됩니다.

메시지는 검증되어 `VerifiedTransceiverMessage` 계정에 저장된 후 `Inbox`에 추가되어 추가 처리를 기다립니다.

두 체인 모두에서 재실행 방지 메커니즘이 작동하여 메시지가 두 번 이상 실행되지 않도록 합니다. 메시지가 성공적으로 수신되면 EVM의 `ReceivedMessage` 이벤트나 SVM의 `ReceiveMessage` 로그가 발생합니다.

민팅 또는 잠금 해제 (Mint or Unlock)

마지막으로, 메시지가 검증되고 인증되면 토큰은 민팅(소스 체인에서 소각된 경우)되거나 잠금 해제(소스 체인에서 잠긴 경우)될 수 있습니다.

그 후 토큰은 목적지 체인의 수신자에게 전송되어 크로스체인 전송이 완료됩니다.

- **EVM의 경우:**

소스 체인에서 소각된 경우 새 토큰이 발행되거나, 잠긴 경우 잠금이 해제됩니다.

`TransferRedeemed` 이벤트가 토큰이 성공적으로 전송되었음을 나타냅니다.

- **SVM의 경우:**

프로그램이 잠금 모드인지 소각 모드인지에 따라 토큰이 잠금 해제되거나 민팅됩니다.

`release_inbound_unlock` 또는 `release_inbound_mint` 명령이 전송을 완료하며, 이에 해당하는 로그가 생성됩니다.

두 경우 모두 토큰이 발행 또는 해제되면 전송 프로세스가 완료되며, 수신자는 토큰을 받습니다.

전송이 완전히 완료되었음을 알리기 위해 이벤트가 발생합니다.

전송 흐름 (Flow of a Transfer)

이 페이지는 네이티브 토큰 전송(NTT) 메시지의 전체 수명 주기를 다루며, 전송이 어떻게 시작되고, 전송되며, 검증되고, 체인 간 완료되는지를 설명합니다. 또한 NTT 매니저와 트랜시버의 역할을 강조합니다.

NTT 매니저는 전송을 감독하고, 속도 제한 및 인증을 처리하며, 각 토큰에 대해 여러 트랜시버를 관리합니다. 이들은 소스 체인에서 토큰이 잠기거나 소각되기 전까지 목적지 체인에서 발행 또는 해제되지 않도록 보장합니다.

트랜시버는 소스와 목적지 매니저 간 전송을 라우팅하여, 정확한 메시지 전달과 토큰 전송을 보장합니다. 이들은 원홀 코어와 독립적으로 작동하며, 다양한 검증 백엔드를 지원할 수 있습니다.

전송 흐름 단계 (Transfer Flow)

크로스체인 토큰 전송은 다음 단계를 따릅니다:

1. 소스 체인에서의 시작 (Initiation on the Source Chain)

전송은 사용자가 소스 체인의 NTT 매니저 계약을 호출할 때 시작됩니다.

- **소각 모드(Burning mode):**
사용자의 계정에서 토큰이 소각됩니다.
- **잠금 모드(Locking mode):**
토큰이 소스 체인 네이티브일 경우, 토큰은 NTT 매니저 계약에 잠깁니다.

2. 아웃바운드 속도 제한 확인 (Outbound Rate Limiting Check)

NTT 매니저는 전송 금액이 현재 아웃바운드 용량을 초과하는지 확인합니다.

- **용량 내:** 전송이 즉시 진행됩니다.
- **용량 초과 및 대기열 허용:** 속도 제한 기간 후 전송이 완료되도록 대기열에 추가됩니다.
- **용량 초과 및 대기열 비허용:** 전송이 실패합니다.

3. 메시지 생성 및 배포 (Message Creation and Distribution)

NTT 매니저는 전송 세부 정보가 포함된 NTT 메시지를 생성하고, 이를 활성화된 모든 트랜시버로 전달합니다. 각 트랜시버는 이를 자체 메시지 형식으로 패키징합니다.

4. 크로스체인 메시지 전송 (Cross-Chain Message Transmission)

각 트랜시버는 자체 검증 네트워크를 통해 메시지를 전송합니다.

- **원홀 트랜시버:**
원홀 가디언 네트워크를 사용하여 메시지 인증 및 자동 릴레이(선택적)를 수행합니다.

- **커스텀 트랜시버:**

검증 백엔드(예: 밸리데이터, 멀티시그 등)를 자유롭게 사용할 수 있습니다.

5. 메시지 수신 및 인증 (Message Reception and Attestation)

목적지 체인에서 트랜시버는 메시지를 수신하고 검증합니다.

- 각 트랜시버는 자체 검증 메소드에 따라 메시지를 검증합니다.
- 트랜시버는 검증된 메시지를 목적지 NTT 매니저로 전달합니다.
- NTT 매니저는 트랜시버로부터 인증(Attestation)을 수집합니다.

6. 임계값 검증 (Threshold Verification)

목적지 NTT 매니저는 전송에 대해 충분한 트랜시버가 인증할 때까지 기다립니다 (구성된 임계값에 따라 다름).

- **임계값 충족 시:** 전송이 실행됩니다.
- **임계값 미충족 시:** 추가 인증을 기다립니다.

7. 인바운드 속도 제한 확인 (Inbound Rate Limiting Check)

NTT 매니저는 수신된 전송이 인바운드 용량을 초과하는지 확인합니다.

- **용량 내:** 전송이 즉시 완료됩니다.
- **용량 초과:** 전송이 대기열로 지연됩니다.

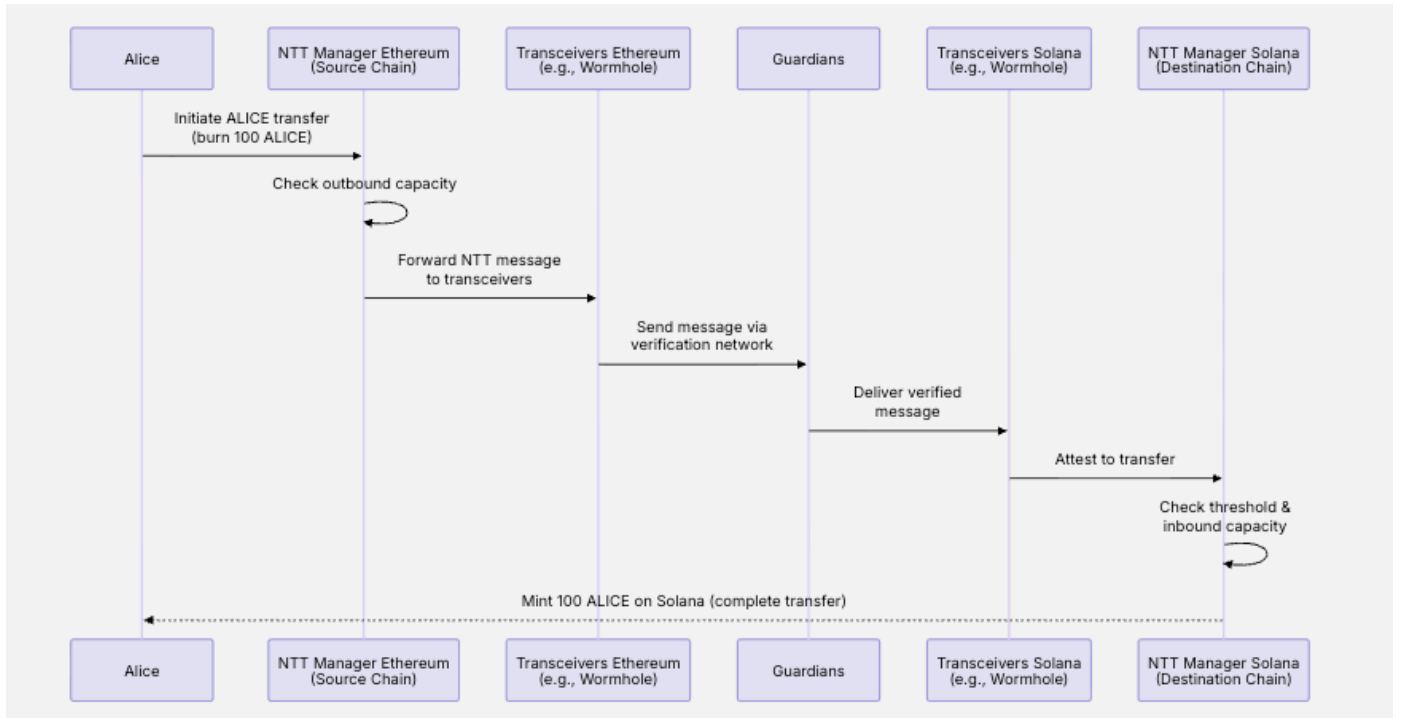
8. 목적지 체인에서의 전송 완료 (Transfer Completion on Destination Chain)

속도 제한 확인을 통과하면, NTT 매니저는 전송을 완료합니다.

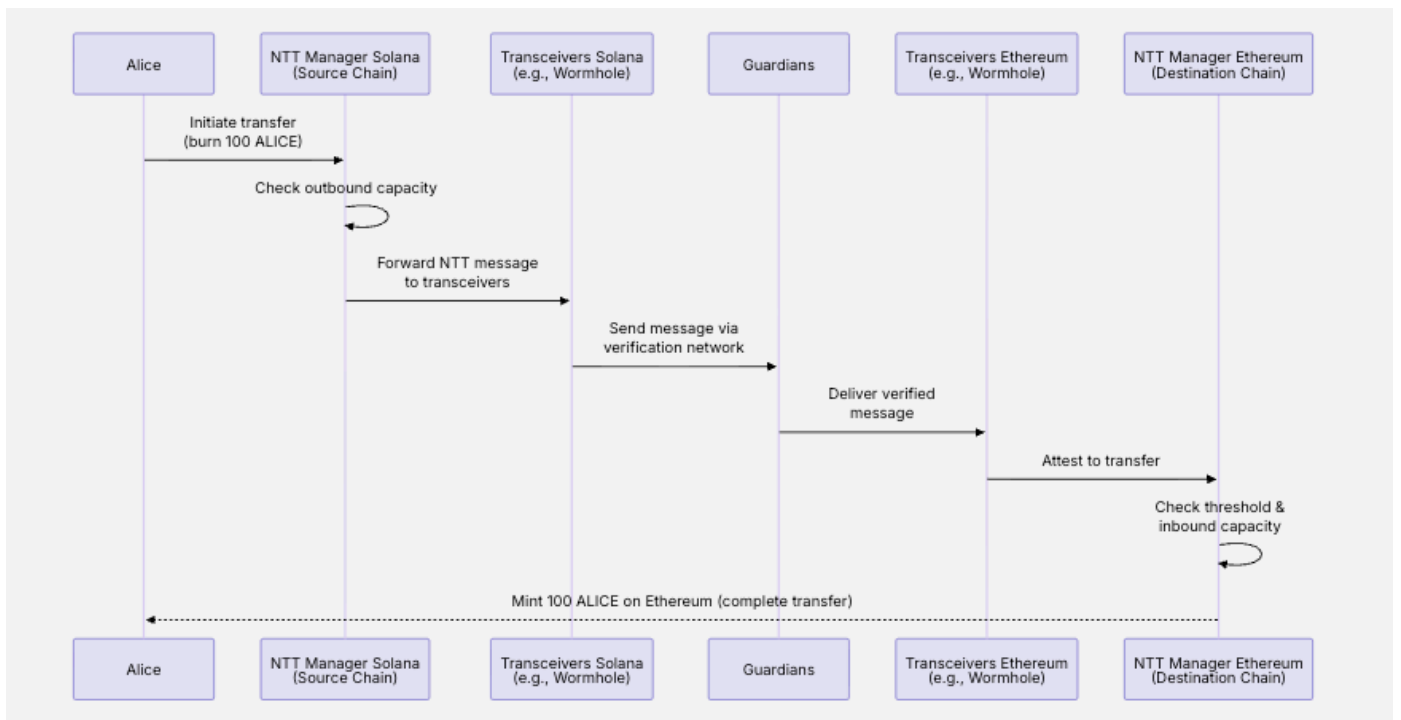
- **소각 모드:** 새 토큰이 수신자에게 발행됩니다.
- **잠금 모드:** 목적지 체인에서 네이티브 토큰인 경우, 계약에서 수신자에게 해제됩니다.

예시

Alice가 이더리움에서 솔라나로 NTT의 소각 모드를 사용하여 100 ALICE 토큰을 전송하려고 합니다. 이더리움의 NTT 매니저에서 ALICE가 소각되고, 트랜시버가 전송을 인증하며, 솔라나에서 동일한 양의 ALICE가 민팅됩니다. 아래 다이어그램은 이 전송 흐름을 보여줍니다.



이제 Alice가 솔라나에서 이더리움으로 ALICE를 다시 전송하려고 한다고 가정합니다. 솔라나의 NTT 매니저에서 ALICE가 소각되고, 동일한 양이 이더리움에서 민팅됩니다. 아래 다이어그램은 이 역방향 전송 흐름을 보여줍니다.



EVM 전송 흐름 세부 정보 (EVM Transfer Flow Details)

전송 (Transfer)

transfer 함수는 전송 세부 정보와 함께 호출되며, TransferSent 이벤트가 발생합니다.

속도 제한 (Rate Limiting)

소스 체인에서 전송이 속도 제한에 걸리고 `shouldQueue` 플래그가 활성화되어 있으면, 해당 전송은 아웃바운드 큐에 추가됩니다. 구성된 `_rateLimitDuration` 이 만료된 후 `completeOutboundQueuedTransfer` 메소드를 통해 전송을 해제할 수 있습니다. 이때 `OutboundTransferQueued` 및 `OutboundTransferRateLimited` 이벤트가 발생합니다.

클라이언트가 `rateLimitDuration` 이 만료되기 전에 큐에서 전송을 해제하려고 시도하면, 컨트랙트는 `OutboundQueuedTransferStillQueued` 오류와 함께 `revert` 됩니다.

유사하게, 목적지 체인에서 속도 제한에 걸린 전송은 인바운드 큐에 추가됩니다. 이러한 전송은 `completeInboundQueuedTransfer` 메소드를 통해 큐에서 해제할 수 있으며, `InboundTransferQueued` 이벤트가 발생합니다.

클라이언트가 `rateLimitDuration` 이 만료되기 전에 큐에서 전송을 해제하려고 시도하면, 컨트랙트는 `InboundQueuedTransferStillQueued` 오류와 함께 `revert` 됩니다.

속도 제한기를 비활성화하려면 `_rateLimitDuration` 을 0으로 설정하고 `NttManager` 생성자에서 `_skipRateLimiting` 필드를 활성화하세요. 이를 잘못 구성하면 오류가 발생합니다. 속도 제한기가 비활성화된 경우, 인바운드 및 아웃바운드 속도 제한을 0으로 설정할 수 있습니다.

메시지 전송 (Sending the Message)

`NttManager` 가 메시지를 트랜시버로 전달하면, 메시지는 `sendMessage` 메소드를 통해 전송됩니다. 트랜시버는 메소드 시그니처를 강제하지만, 실제 메시지 전송 구현(예: 웜홀 트랜시버를 통한 메시지가 웜홀 릴레이, 커스텀 릴레이어, 또는 코어 브리지를 통한 수동 게시로 전송되는지)은 트랜시버가 자유롭게 결정합니다.

메시지가 전송되면, 컨트랙트는 `SendTransceiverMessage` 이벤트를 발생시킵니다.

메시지 수신 (Receiving the Message)

소스 체인에서 트랜시버가 메시지를 발생시키면, 오프체인 프로세스(예: 릴레이어)가 해당 메시지를 수신자 체인의 대응 트랜시버로 전달합니다.

릴레이어는 메시지 수신을 위해 엔트리 포인트를 통해 트랜시버와 상호작용합니다. 예를 들어, 릴레이어는 `WormholeTransceiverMessage` 컨트랙트의 `receiveWormholeMessage` 메소드를 호출하여 메시지를 실행합니다. 이 과정에서 `ReceiveRelayedMessage` 이벤트가 발생합니다.

이 메소드는 또한 메시지를 목적지 체인의 `NttManager` 로 전달해야 합니다. 트랜시버 인터페이스는 이 메소드의 시그니처를 선언하지 않는데, 메시지 수신은 각 트랜시버에 특화되어 있으며 일률적인 방식은 과도하게 제한적일 수 있기 때문입니다.

NttManager 컨트랙트는 메시지를 안전하게 실행할 수 있는지 판단하기 위해 트랜시버 인증 (Attestation)에 대해 M of N 임계값을 허용합니다. 예를 들어, 임계값 요구 사항이 1이면 단일 트랜시버가 유효한 인증을 전달한 후 메시지가 실행됩니다. 임계값 요구 사항이 2이면 두 트랜시버가 유효한 인증을 전달한 후에만 메시지가 실행됩니다. 트랜시버가 메시지에 대해 인증하면 컨트랙트는 MessageAttestedTo 이벤트를 발생시킵니다.

NTT는 재실행 방지(replay protection)를 구현하므로, 특정 트랜시버가 동일한 메시지 인증을 두 번 전달하려고 하면, 컨트랙트는 TransceiverAlreadyAttestedToMessage 오류와 함께 revert 됩니다.

NTT는 또한 메시지의 재실행에 대한 재실행 방지를 구현합니다. 이 검사는 재진입(reentrancy) 방지로도 작동합니다.

메시지가 이미 실행된 경우, 컨트랙트는 오류로 revert 하는 대신 조기 종료하고 MessageAlreadyExecuted 이벤트를 발생시킵니다. 이는 `threshold < totalTransceivers` 인 상황에서 트랜시버들이 동일한 메시지를 전달하려고 시도하여 발생할 수 있는 경쟁 상태(race condition)를 완화하고, 클라이언트(오프체인 프로세스)에게 중복 전달을 시도하지 않도록 알립니다.

민팅 또는 잠금 해제 (Minting or Unlocking)

전송이 성공적으로 검증되면, 모드가 "burning" 인 경우 토큰을 수신자에게 민팅하고, "locking" 인 경우 토큰을 잠금 해제하여 목적지 체인 수신자에게 전달할 수 있습니다. 소스 토큰의 소수 자릿수는 와이어 포맷에서 강제되는 0 과 TRIMMED_DECIMALS 사이로 제한됩니다. 목적지 체인의 토큰 소수 자릿수가 TRIMMED_DECIMALS 를 초과하는 경우 전송 금액은 언트림(스케일-업)됩니다. 적절한 수량의 토큰이 수신자에게 민팅되거나 잠금 해제되면 TransferRedeemed 이벤트가 발생합니다.

보안 (Security)

글로벌 회계 관리자 (Global Accountant)

글로벌 회계 관리자는 모든 토큰 전송의 무결성을 점검하는 방어-심층(Defense-in-Depth) 보안 기능입니다. 이는 체인 간 잔액이 서로 격리된 상태를 유지하고, 어떤 체인에서도 발행된 양보다 더 많은 토큰이 소각되어 외부로 전송될 수 없도록 보장합니다.

이 기능은 네이티브 자산의 1:1 동질성을 보장합니다. 어느 시점에서든 스포크(spoke) 체인에서 나오는 자산이 해당 스포크 체인으로 보낸 네이티브 자산 수를 초과하지 않습니다. 회계 투명성을 집행하는 역할을 수행하는 가디언(Guardians)은 무결성 검사를 통과한 경우에만 네이티브 토큰 전송(NTT)에 대해 인증을 제공하여 추가적인 보안 계층을 제공합니다.

멀티체인 배포에서 글로벌 어카운턴트 구성을 원한다면 웜홀 컨트리뷰터에게 문의하세요.

거버넌스 및 업그레이더빌리티 (Governance and Upgradeability)

통합자(Integrators)는 프록시 패턴을 사용하여 컨트랙트를 업그레이드하고, 트랜시버의 추가/제거를 관리하는 거버넌스 메커니즘을 구현해야 합니다. 이는 NttManager 컨트랙트의 업그레이드 함수들에서 시연된 바와 같습니다. 이러한 프로세스는 인증 및 메시지 승인에 대한 임계값과 규칙을 설정할 수도 있습니다.

NTT 시스템의 레지스트리 구성 요소는 신뢰할 수 있는 트랜시버 목록을 유지하고 그 상태를 관리하는 데 핵심적입니다. 다음 작업에 대한 거버넌스 프로세스는 브리징 컨트랙트 하나 이상 또는 토큰 컨트랙트 중 해당되는 컨트랙트에 온체인으로 직접 제출할 수 있습니다:

- 레지스트리에서 트랜시버 주소 추가 또는 제거
- 브리징 컨트랙트에서 토큰 컨트랙트 주소 설정
- 브리징 컨트랙트에서 웜홀 코어 컨트랙트 주소 설정
- 토큰 컨트랙트에서 등록된 브리징 컨트랙트 주소 설정

이 거버넌스 모델은 배포 환경이 어떠하든 새로운 요구 사항에 적응 가능하면서도 시스템이 안전하게 유지되도록 보장합니다.

Wrapped Token Transfers

개요 (Overview)

랩드 토큰 전송 개요

랩드 토큰 전송(Wrapped Token Transfers, WTT)은 다양한 블록체인 네트워크 간에 래핑된 토큰을 브리징하기 위한 웜홀 모듈입니다. 한 네트워크에서 자산을 잠그고 다른 네트워크에서 해당 자산에 1:1로 대응하는 래핑된 토큰을 민팅함으로써, 안전하고 효율적이며 조합 가능한 멀티체인 토큰 이동을 가능하게 합니다.

이 개요는 WTT의 주요 기능, 일반적인 프로세스, 그리고 크로스체인 애플리케이션 구축을 시작하기 위한 가능한 다음 단계를 다룹니다.

핵심 기능 (Key Features)

WTT는 멀티체인 토큰 전송의 상호운용성 문제를 해결하도록 설계되었습니다. 핵심 기능은 다음과 같습니다:

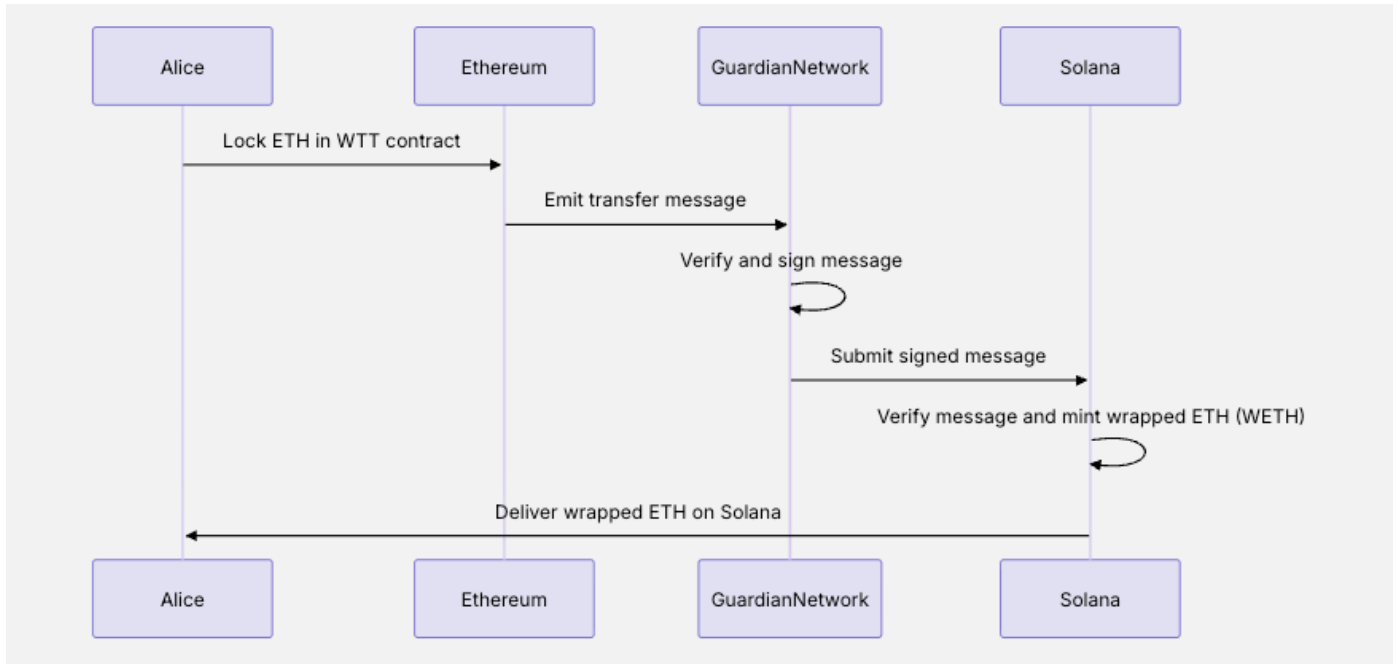
- **상호운용성**: 표준을 준수하는 토큰(예: ERC-20, SPL)을 30개가 넘는 지원 체인 간에 전송
- **락-앤-민트 메커니즘**: 소스 체인에서 잠긴 자산에 의해 1:1로 보증되는 래핑 토큰을 민팅
- **메타데이터 유지**: 이름, 심볼, 소수 자릿수 등 토큰 속성을 체인 간 일관되게 유지
- **페이로드 포함 전송**: 임의 데이터를 토큰 전송에 첨부하여 특정 동작을 트리거
- **탈중앙 보안**: 가디언 네트워크에 의해 검증되어 크로스체인 일관성과 메시지 진위성 보장

동작 방식 (How It Works)

WTT는 대규모 멀티체인 상호운용성을 위한 안정적인 기반을 제공합니다. 전송 프로세스는 다음 핵심 단계를 따릅니다:

1. **어테스테이션(Attestation)**: 토큰의 메타데이터(예: 심볼, 이름, 소수 자릿수)가 목적지 체인에 등록됩니다. 이 단계는 토큰마다 한 번만 필요합니다.
2. **락(Locking)**: 소스 체인에서 네이티브 토큰이 커스터디 계정에 잠깁니다.
3. **메시지 발생**: 가디언 네트워크가 검증 후 VAA를 발생시킵니다.
4. **검증(Verification)**: 목적지 체인에서 VAA를 제출·검증하여 진위성을 확인합니다.
5. **민팅(Minting)**: 목적지 체인에서 래핑된 토큰이 민팅되거나(또는 네이티브 토큰이 해제되어) 수신자에게 전달됩니다.

다음 다이어그램은 Alice가 이더리움의 ETH를 솔라나의 자신의 계정으로 브리징하는 단순화된 흐름을 보여줍니다.



사용 사례 (Use Cases)

다음은 WTT의 강력함과 범용성을 보여주는 핵심 사용 사례입니다.

- 탈중앙 플랫폼에서의 멀티체인 보상 및 토큰 유틸리티 (예: Chingari)
 - WTT: 체인 간 토큰 전송
 - 메세징: 보상 분배 및 클레임 프로세스 지원
- 토큰화된 게임 보상
 - WTT: 기반이 되는 안전한 락-앤-민트 로직 처리
 - 커넥트: 체인 간 게임 토큰 이동을 위한 사용자 친화적 UI 제공
- 멀티체인 디파이 차익거래
 - WTT: 디파이 자산의 빠르고 안전한 이동 가능
 - 커넥트: 디파이 애그리게이터 플랫폼 내 원활한 멀티체인 스왑 온보딩 위젯 제공

커넥트(Connect)

커넥트 개요

웜홀 커넥트 위젯을 사용하면 애플리케이션 내에서 사용자가 멀티체인 자산 전송을 직접 수행할 수 있습니다. 커넥트는 브리징의 복잡성을 단순화하여, 다양한 블록체인 전반에서 자산을 이동하기 위한 단일하고 직관적인 상호작용 지점을 제공합니다. 이를 통해 연결된 어떤 네트워크에서도 유동성과 기회를 원활하게 활용할 수 있습니다.

주요 기능

커넥트의 주목할 만한 기능은 다음과 같습니다:

- **인앱 멀티체인 전송:** 앱을 떠나지 않고 자산을 브리지합니다.
- **맞춤형 기능:** 체인과 커스텀 RPC 지정, 토큰 관리, WTT·CCTP·NTT와 같은 브리징 경로 선택.
- **맞춤형 UI:** 브리지 인터페이스를 브랜드에 맞게 스타일링합니다.
- **선택적 목적지 가스:** 대상 체인에서 초기 트랜잭션을 위한 가스를 제공합니다.
- **래핑 및 네이티브 자산 지원:** 래핑 및 네이티브 토큰을 모두 지원하며 세틀먼트와 통합됩니다.

체인별로 어떤 경로와 기능이 지원되는지 확인하려면 [Feature Support Matrix](#)를 참조하세요.

동작 방식

사용자가 멀티체인 전송을 시작하면, 커넥트는 다음 핵심 단계를 안내하고 백그라운드에서 전송 프로세스를 자동화합니다.

1. **전송 시작:** 선택한 지갑을 소스 체인에 연결하고, 전송할 자산과 소스 체인을 선택합니다.
2. **전송 설정 마무리:** 목적지 지갑을 연결하고 대상 체인을 선택한 뒤 브리징 경로(수동 또는 자동)를 선택합니다.
3. **소스 체인에서 트랜잭션 제출:** 초기 블록체인에서 자산 잠금 또는 예치를 트리거하기 위해 전송 세부사항을 확인합니다. 커넥트가 트랜잭션 과정을 안내합니다.
4. **VAA 또는 어테스테이션 생성:** 웜홀 가디언이 소스 트랜잭션을 관찰하고 Verifiable Action Approval(VAA)을 생성합니다.
5. **목적지로 릴레이:** VAA 또는 어테스테이션이 자동으로 목적지 체인으로 릴레이됩니다.
6. **목적지에서 검증:** 대상 체인의 컨트랙트가 수신된 VAA를 검증합니다.
7. **자산 해제/민팅:** 검증이 완료되면 해당 자산이 목적지 체인에서 해제되거나 민팅되어 지갑으로 전달됩니다.

사용 사례

다음은 커넥트의 강력함과 범용성을 보여주는 주요 사용 사례입니다.

- 크로스체인 스왑 및 유동성 집계
 - **커넥트**: 사용자 친화적인 자산 전송을 처리합니다.
 - **Native Token Transfers**: 네이티브 자산을 체인 간 이동합니다.
 - **쿼리**: 최적의 거래 실행을 위해 실시간 가격을 가져옵니다.
 - 크로스체인 결제 위젯
 - **커넥트**: 다양한 토큰으로 원활한 결제를 지원합니다.
 - **Native Token Transfers**: 직접적이고 네이티브한 자산 전송을 보장합니다.
 - Web3 게임 자산 전송
 - **커넥트**: 체인 간 게임 토큰 이동을 위한 사용자 친화적 방법을 제공합니다.
 - **Wrapped Token Transfers**: 기반이 되는 락-앤-민트 로직을 안전하게 처리합니다.
-

세틀먼트(Settlement)

세틀먼트 개요

웜홀 세틀먼트는 사용자가 직접 실행을 처리하지 않고도, 토큰 전송이나 스왑과 같이 원하는 결과를 지정할 수 있게 해주는 멀티체인 전송 시스템입니다. 대신 솔버(solver)라고 불리는 오프체인 에이전트들이 이러한 사용자 인텐트를 충족시키기 위해 경쟁합니다.

세틀먼트는 속도, 실행 품질, 신뢰성을 우선시합니다. 기본 경로인 Mayan Swift는 엄선된 솔버 집합 간의 빠른 오프체인 경매를 통해 낮은 지연의 브리징과 최소 슬리피지를 달성합니다. 모든 정산 단계는 웜홀 메시지를 통해 온체인에서 검증 가능합니다.

보다 광범위한 사용 사례와 프로토콜 수준의 실행을 위해 Mayan MCTP가 대안 경로를 제공합니다. 이는 Circle의 CCTP를 래핑하여 네이티브 USDC 브리징과 토큰 전달을 단일하고 검증 가능한 흐름으로 제공합니다. 체인 파이널리티 제약으로 인해 더 느리지만, MCTP는 신뢰할 수 있는 크로스체인 전송 메커니즘을 제공합니다.

주요 기능

- **인텐트 기반 아키텍처**: 사용자가 원하는 결과(예: 체인 Z에서 X를 Y로 스왑)를 표현하고, 솔버가 이를 실행합니다.
- **솔버 경매**: 솔버가 온체인 경매에서 인텐트 이행 권리를 두고 경쟁하여 실행 품질을 향상합니다.
- **빠르고 풀백 가능**: 고속 실행과 신뢰할 수 있는 풀백 경로를 결합합니다.

- **최소 슬리피지:** 세틀먼트는 복잡한 밸런싱 작업을 추상화하고 USDC 및 NTT로 배포된 토큰 같은 서틀 자산을 사용합니다.
- **온체인 검증 가능성:** 경매는 오프체인이지만, 모든 정산 단계는 웜홀 메시지를 통해 온체인에서 검증됩니다.
- **두 가지 통합 경로:** 속도를 위한 Mayan Swift, 호환성과 중복성을 위한 Mayan MCTP.

동작 방식

세틀먼트의 핵심 구성 요소는 다음 두 가지입니다.

- **인텐트(Intent):** 사용자가 원하는 결과(예: USDC를 다른 체인으로 보내고 ETH로 받기)를 정의하는 서명된 트랜잭션입니다. 사용자가 원하는 “무엇”을 추상화하며, “어떻게”는 포함하지 않습니다.
- **솔버(Solvers):** 인텐트를 이행하기 위해 경매에서 경쟁하는 서드파티 에이전트입니다. 자본을 선투입하고 스왑이나 전송을 수행하며, 그 대가로 수수료를 받습니다.

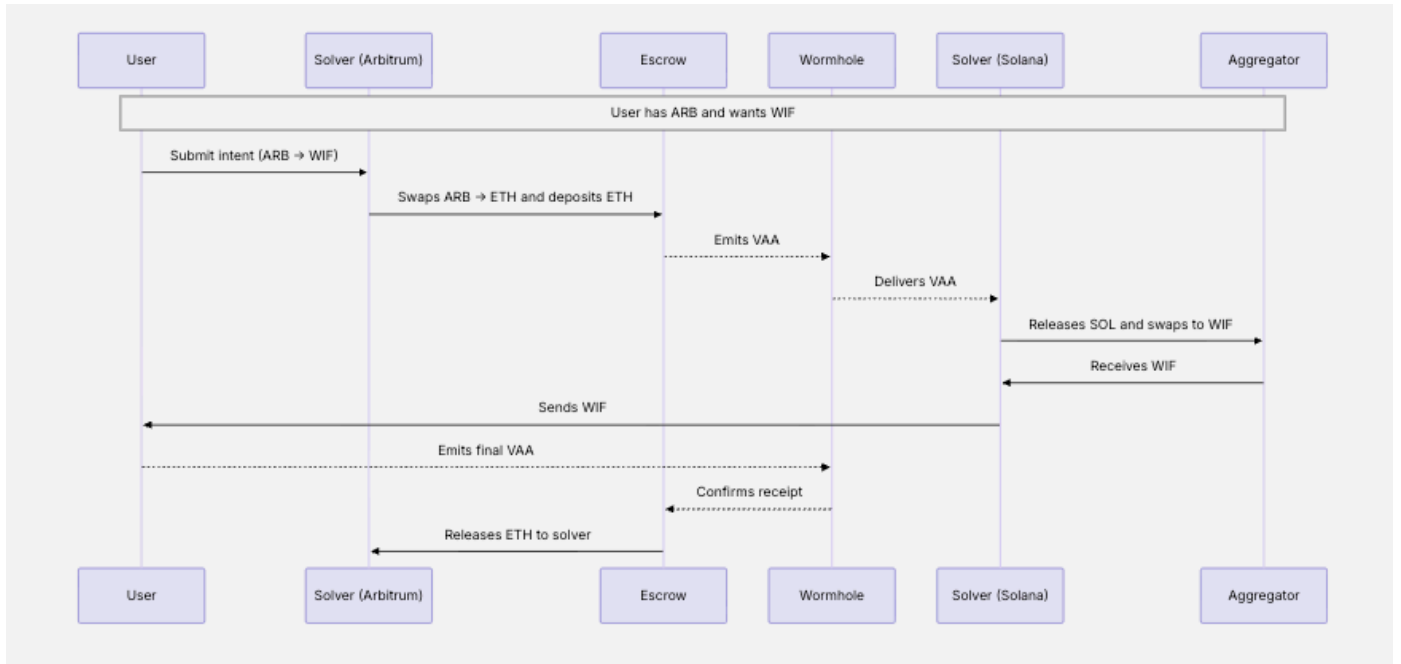
세틀먼트는 현재 다음의 통합 프로토콜을 지원합니다.

Mayan Swift

Mayan Swift는 전통적인 인텐트 기반 아키텍처를 구현하며, 솔버는 보유 자산을 활용해 사용자 인텐트를 이행하기 위해 경쟁합니다. 전형적으로 약 12초의 빠른 실행을 제공합니다. 참여하려면 솔버가 여러 체인에 자산을 보유해야 하며, 일부 체인이 고갈되고 다른 체인에 잉여가 쌓이는 불균형이 발생할 수 있습니다. 이는 주기적인 리밸런싱을 요구하고 운영 오버헤드를 증가시킵니다. 그럼에도 불구하고 Mayan Swift는 고속 전송에 이상적이며, 개방적이고 경쟁적인 경매는 실행 가격을 낮추는 데 도움이 됩니다.

아래 다이어그램은 사용자가 Arbitrum의 ARB를 솔라나의 WIF로 스왑하려 할 때 Mayan Swift가 크로스체인 인텐트를 처리하는 방식을 보여줍니다. 백그라운드에서는 솔버가 두 체인 전반의 유동성을 관리하는 더 복잡한 과정이 진행됩니다.

1. **솔버가 아비트럼에서 시작:** 솔버가 ARB → ETH로 스왑하고 아비트럼의 에스스로에 ETH를 예치합니다.
2. **솔라나로 VAA 발생:** Verifiable Action Approval(VAA)이 솔버가 솔라나에서 SOL을 해제하도록 트리거하고, 이는 애그리게이터를 통해 WIF로 스왑됩니다.
3. **사용자가 WIF 수령:** 사용자가 WIF를 수령하면, 두 번째 VAA가 발생하여 전송을 최종화하고 에스스로에 보관된 ETH가 솔버에게 해제됩니다.
4. **실패 처리:** 어느 단계에서든 실패 시, 에스스로의 ETH는 유지되거나 사용자에게 반환됩니다. 실행에 성공한 경우에만 솔버가 보상을 받습니다.



Mayan MCTP

Mayan MCTP는 Circle의 CCTP를 세틀먼트 프레임워크로 래핑한 풀백 프로토콜입니다. USDC 브리징과 스왑을 프로토콜 로직이 처리하는 단일 작업으로 묶습니다. 체인 파이널리티에 의존하므로 더 느리지만, 더 넓은 호환성과 중복성을 제공하여 더 빠른 경로가 사용 불가하거나 Swift가 지원되지 않는 체인을 대상으로 할 때 유용합니다. 일반적으로 프로토콜 수수료로 인해 더 비싸지만, 더 빠른 옵션이 없을 때 신뢰할 수 있는 정산을 보장합니다.

사용 사례

- **크로스체인 선물 거래**
 - **세틀먼트**: 체인 간 빠른 토큰 실행을 제공합니다.
 - **쿼리**: 체인 전반에서 실시간 가격을 가져오고 포지션 상태를 관리합니다.
- **브리징 인텐트 라이브러리**
 - **세틀먼트**: 사용자 정의 브리지 인텐트를 처리합니다.
 - **메세징**: 크로스체인 함수 호출을 트리거합니다.
- **멀티체인 예측 시장**
 - **세틀먼트**: 체인 간 토큰 흐름을 실행합니다.
 - **쿼리**: 시장 데이터를 가져오고 상태를 추적합니다.

CCTP Bridge

웜홀과 함께하는 CCTP 개요

Circle의 Cross-Chain Transfer Protocol(CCTP)과 웜홀 메시징 프로토콜의 통합은, 네이티브 USDC를 서로 다른 블록체인 네트워크 간에 안전하고 효율적으로 전송하는 동시에 더 복잡한 멀티체인 상호 작용을 가능하게 하는 견고한 시스템을 만듭니다. 이 조합은 스테이블코인 이동을 간소화하고, 위험을 줄이며, 탈중앙 애플리케이션을 위한 새로운 가능성을 엽니다.

주요 기능

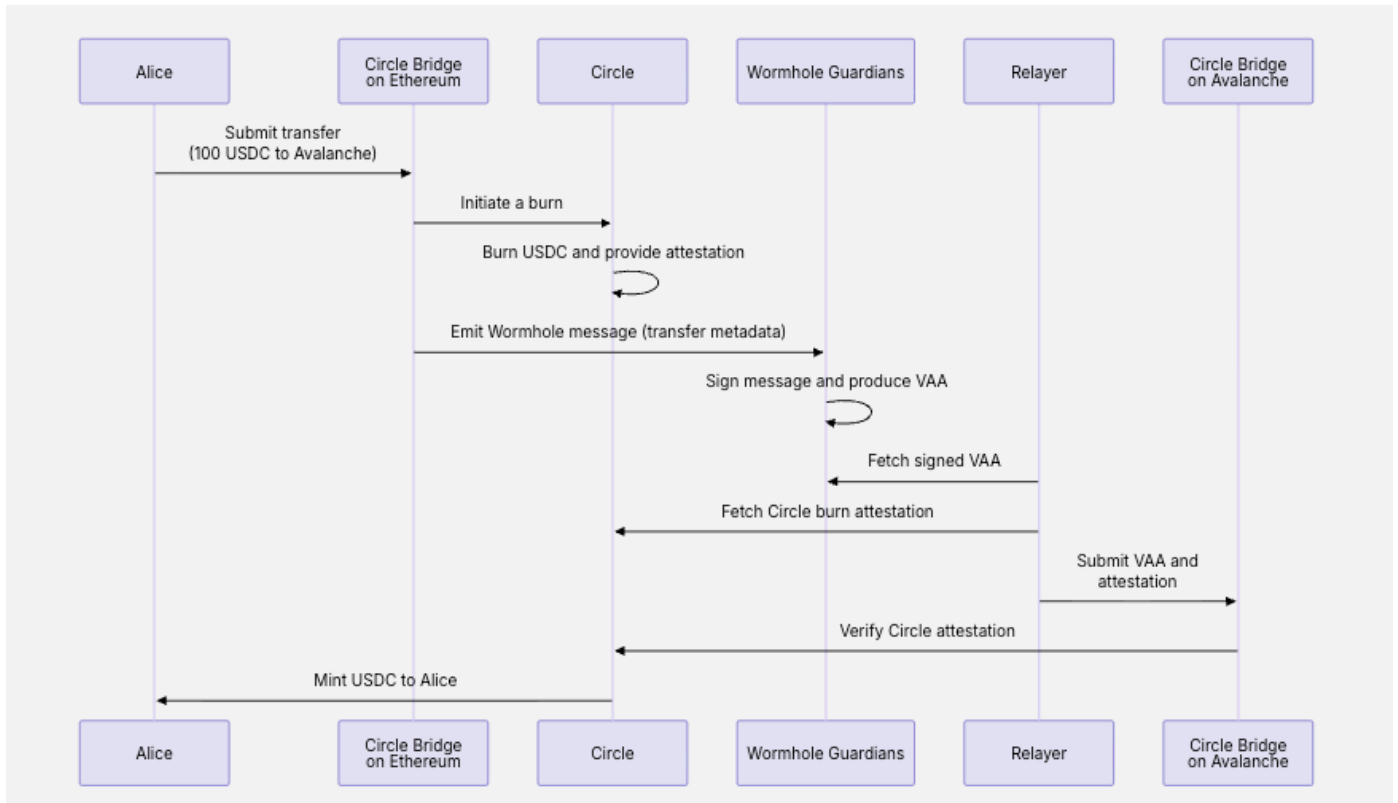
- **안전한 네이티브 USDC 전송:** CCTP의 핵심은 네이티브 USDC 전송을 위한 “소각 및 민팅” 메커니즘입니다. 이는 래핑 자산과 중간 브리지에 수반되는 위험을 제거합니다.
- **원자적 실행:** CCTP와 웜홀을 결합하면, 목적지 체인에서의 부가 명령 실행과 USDC 전송을 단일 원자 트랜잭션으로 처리할 수 있습니다.
- **자동 릴레이:** 사용자가 USDC 전송을 직접 상환(redeem)할 필요를 없앱니다.
- **향상된 조합성:** 전송과 함께 추가 데이터를 전달하여, 개발자가 더 정교한 크로스체인 애플리케이션을 구축할 수 있습니다.
- **가스 드롭오프:** 성공적인 전송 시 USDC 일부를 목적지 체인의 가스 토큰으로 변환할 수 있습니다.
- **가스 지불:** 자동 vs 수동 전송에서의 목적지 가스 처리:
 - **자동:** 보통 사용자는 목적지 가스 토큰을 미리 보유할 필요가 없고, 릴레이어가 가스 비용을 선지불하며 가스 드롭오프 또는 초기 수수료로 상환받습니다.
 - **수동:** 사용자가 목적지 가스를 직접 지불하며, 프로토콜이 클레임 이후 USDC→가스 변환을 제공할 수 있습니다.

동작 방식

이 섹션은 CCTP를 사용해 네이티브 USDC를 체인 간 전송하면서, 선택적으로 목적지 체인에서 동작을 트리거하는 엔드투엔드 흐름을 설명합니다. Circle과 웜홀은 안전하고 검증 가능한 전송 및 실행 과정을 보장하기 위해 각 단계를 조정합니다.

1. **Alice가 이더리움에서 전송을 시작:** Circle Bridge에 100 USDC를 아발란체로 전송하도록 요청합니다. 필요하면 선택적 페이로드 데이터를 포함할 수 있습니다.
2. **토큰 수탁 및 소각:** Circle Bridge가 Alice의 USDC를 수탁하고 Circle의 CCTP를 사용해 소각을 시작하며, 오프체인 어테스테이션 프로세스를 트리거합니다.

3. **웜홀 메시지 게시:** 전송 메타데이터가 웜홀 메시지로 게시됩니다. 가디언이 이를 검증하고 서명하여 **Verifiable Action Approval(VAA)** 을 생성합니다.
4. **릴레이어가 메시지를 자동 처리:** VAA와 Circle 어테스테이션이 준비되면, 릴레이어가 이를 Avalanche의 Circle Bridge에 제출합니다.
5. **토큰 민팅:** Circle Bridge가 두 증명을 검증하고 Circle의 CCTP를 통해 100 USDC를 Alice에게 민팅합니다. 페이로드가 포함된 경우, 이는 원자적으로 실행될 수 있습니다.



CCTP vs Wrapped Token Transfers(WTT)

기능	CCTP(네이티브 USDC)	WTT(래핑 토큰)
지원 자산	Circle 발행 USDC	표준 준수 토큰(예: ERC-20, SPL)
메커니즘	소스에서 소각, 목적지에서 민팅	소스에서 잠금, 목적지에서 래핑 민팅
목적지 결과	네이티브 USDC	웜홀-래핑 토큰
페이로드	선택적 페이로드 전송; USDC 민팅 시 목적지에서 실행	선택적 페이로드 전송; 상환 시 수신 컨트랙트가 실행
사용 시점	CCTP 지원 체인 간 네이티브 USDC	비-USDC 자산 또는 목적지에서 CCTP 미지원인 USDC

사용 가능한 경로는 [CCTP Supported Networks](#)에서 확인하세요.

CCTP 사용 시점

다음 상황에서 CCTP가 적합합니다:

- **CCTP 지원 체인 간 USDC 전송:** 두 체인이 Circle CCTP를 지원하고 자산이 네이티브 USDC인 경우에만 이 경로가 나타납니다.
- **페이로드를 포함한 USDC 전송:** 목적지 컨트랙트가 토큰이 민팅될 때 로직을 실행할 수 있습니다.

다른 전송의 경우, 다음 옵션을 고려하세요:

- **USDC를 이동하지 않는 데이터 전용 전송:** [메세징](#)을 사용하세요.
- **CCTP 미지원 목적지:** [WTT](#)를 통해 전송하며, 상환 시 실행되는 페이로드를 포함할 수 있습니다.

사용 사례

웜홀의 메시징과 CCTP 통합은 네이티브 USDC를 블록체인 간 안전하게 전송하여 다음과 같은 핵심 크로스체인 사용 사례를 가능하게 합니다.

체인 간 USDC 결제

- **CCTP:** Circle의 소각-민팅 프로토콜로 네이티브 USDC를 전송합니다.
- **웜홀 TypeScript SDK:** 어테스테이션 전달과 가스 처리를 자동화합니다.

- **커넥트:** 멀티체인 USDC 전송을 앱에 직접 임베드합니다.
-

MultiGov

MultiGov 개요

MultiGov는 여러 블록체인 네트워크 전반에 걸친 탈중앙 의사결정을 가능하게 하는 멀티체인 거버넌스 시스템입니다. 웜홀 메시징을 기반으로 하며, 단일 허브나 자산 브리징에 의존하지 않고도 연결된 어떤 체인에 서든 DAO가 제안, 투표, 실행을 관리할 수 있게 합니다. 이는 체인 간 투표력을 집계하고 안전한 제안 실행을 조정함으로써 진정한 멀티체인 거버넌스를 실현합니다.

주요 기능

MultiGov는 DAO 거버넌스를 블록체인 전반으로 확장하여 참여를 늘리고, 웜홀 메시징으로 보안을 개선하며, 대규모에서 통합 의사결정을 가능하게 합니다. 주요 기능은 다음과 같습니다:

- **멀티체인 거버넌스:** 토큰 보유자는 지원되는 어떤 체인에서든 제안에 투표하고 실행할 수 있습니다.
- **허브-스포크 모델:** 제안은 중앙 허브 체인에서 생성되며, 거버넌스 토큰이 존재하는 스포크 체인에서 투표가 진행됩니다.
- **안전한 투표 집계:** 투표 가중치는 체크포인트로 기록·검증되어 이중 투표를 방지합니다.
- **크로스체인 제안 실행:** 승인된 제안은 여러 체인에서 실행될 수 있습니다.
- **유연한 아키텍처:** 웜홀이 지원하는 어떤 블록체인과도 통합 가능합니다.
- **업그레이드 가능성과 확장성:** 구성 요소 전반의 업그레이드를 지원하며, 투표 기록과 시스템 연속성을 유지합니다.
- **Tally 연계:** 제안 생성, 투표, 실행은 Tally를 통해 조정됩니다.

동작 방식

1. **허브 체인에서 제안 생성:** 허브 체인은 투표 집계와 실행 스케줄링을 포함한 핵심 거버넌스 로직을 관리합니다.
2. **스포크 체인에서 투표:** 스포크 체인의 토큰 보유자는 SpokeVoteAggregators를 통해 로컬로 투표하며, 체크포인트가 투표력을 추적합니다.
3. **웜홀을 통해 투표 전송:** 투표는 VAA로 허브에 안전하게 전송되어 메시지 무결성과 크로스체인 검증을 보장합니다.

4. **허브에서 집계 및 최종화:** 허브 체인은 모든 스포크에서 투표를 수신·집계하고, 투표 기간이 끝나면 결과를 최종화합니다.
5. **체인 간 실행:** 승인되면, 제안은 다시 웜홀 메시지를 사용해 하나 이상의 체인에서 실행을 트리거할 수 있습니다.

사용 사례

- **크로스체인 금고(Treasury) 관리**
 - **MultiGov:** 지원되는 어떤 체인에서든 금고 액션에 투표합니다.
 - **메세징:** 제안 실행을 대상 체인으로 전송합니다.
 - **Wrapped Token Transfers (WTT):** 필요 시 자산을 이동합니다.
 - **체인 간 프로토콜 업그레이드 조정**
 - **MultiGov:** 여러 네트워크의 컨트랙트를 업그레이드하는 통합 제안을 생성합니다.
 - **메세징:** 업그레이드 지시를 VAA로 보내고 대상 체인에 실행 페이로드를 전달합니다.
 - **멀티체인 DAO의 점진적 탈중앙화**
 - **MultiGov:** 조정을 유지하면서 새로운 체인으로 거버넌스를 확장합니다.
 - **쿼리:** 원격 스포크에서 온체인 투표력을 가져옵니다.
 - **메세징:** 허브를 통해 결과를 집계하고 동작을 실행합니다.
-

쿼리(Queries)

쿼리 개요

쿼리는 가디언이 어테스트한 온체인 데이터에 대한 온디맨드 접근을 제공합니다. 이를 통해 스마트 컨트랙트가 멀티체인 생태계 전반(가격, 금리, 유동성 등)에서 실시간의 검증 가능한 데이터를 가져올 수 있습니다.

주요 기능

- **온디맨드 데이터 접근:** 가격 피드, 금리, 기타 데이터를 실시간으로 가져옵니다.
- **가디언 어테스트:** 모든 데이터는 가디언의 서명으로 신뢰 없이 검증 가능합니다.
- **크로스체인 준비:** 한 체인에서 요청하고 다른 체인에서 사용할 수 있습니다.

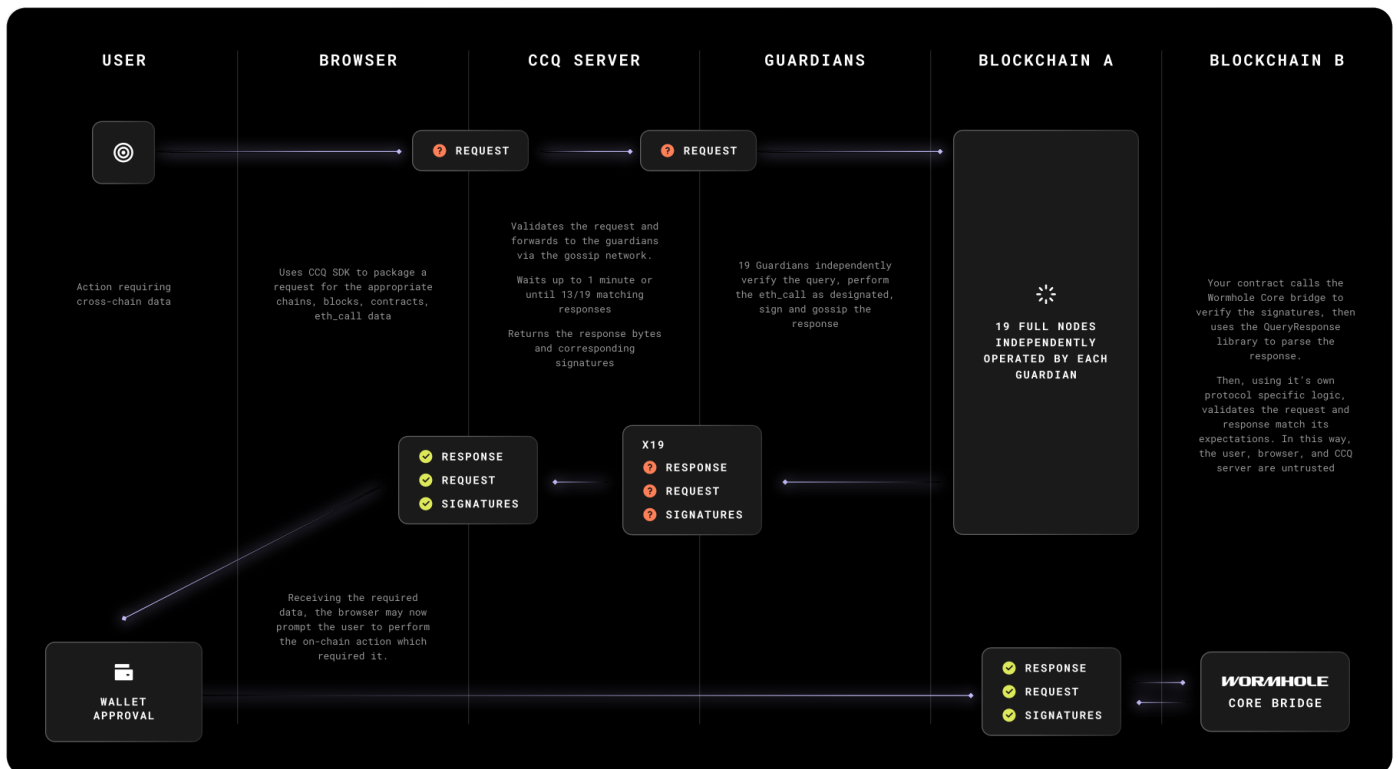
- **스마트 컨트랙트 통합:** 결과는 스마트 컨트랙트가 읽을 수 있는 Verifiable Action Approval(VAA)로 전달됩니다.
- **체인 불가지론적:** 지원되는 EVM 체인, 솔라나, Sui 등에서 동작합니다.

동작 방식

쿼리 요청은 단순하지만 견고한 라이프사이클을 따릅니다. 요청 처리를 담당하는 오프체인 서비스는 CCQ 서버(Cross-Chain Query Server)이며, 본 문서에서는 Query Server로도 지칭합니다.

1. 오프체인 앱이 HTTPS를 통해 CCQ 서버에 쿼리를 보냅니다.
2. CCQ 서버가 요청을 점검하고 이를 가디언과 공유합니다.
3. 가디언이 독립적으로 데이터를 가져와 검증하고 결과에 서명합니다.
4. 충분한 수(2/3 정족수)의 가디언이 일치하는 결과를 반환하면, CCQ 서버는 이를 집계해 최종 응답을 보냅니다.
5. 오프체인 앱은 이 결과를 스마트 컨트랙트에 제출하며, 컨트랙트는 가디언 서명을 검증하고 데이터를 사용합니다.

CCQ 서버는 허가형이지만 신뢰를 요구하지 않습니다. 대부분의 쿼리는 1초 미만에 처리되며, 가디언은 최대 1분 동안 실패한 요청을 재시도합니다. 최대 255개의 쿼리를 배치로 묶어 성능을 최적화하여 효율적인 멀티 체인 워크플로를 지원합니다.



사용 사례

쿼리는 다양한 크로스체인 애플리케이션을 가능하게 합니다. 아래는 일반적인 사용 사례와 구축 시 사용할 수 있는 웜홀 스택 구성 요소입니다.

- **체인 간 대출/차입(예: Folks Finance)**
 - **쿼리:** 실시간으로 금리와 가격을 가져옵니다.
 - **메세징:** 체인 간 액션을 동기화합니다.
 - **Native Token Transfers:** 담보를 네이티브 자산으로 전송합니다.
- **크로스체인 스왑 및 유동성 집계(예: StellaSwap)**
 - **쿼리:** 최적 실행을 위한 실시간 가격을 가져옵니다.
 - **커넥트:** 사용자 친화적인 자산 전송을 처리합니다.
 - **Native Token Transfers:** 네이티브 토큰을 이동합니다.
- **실시간 가격 피드 및 트레이딩 전략(예: Infinex)**
 - **쿼리:** 가격 피드를 가져옵니다.
 - **메세징:** 트레이드를 트리거합니다.
- **멀티체인 예측 시장**
 - **쿼리:** 시장 데이터와 배당률을 가져옵니다.
 - **세틀먼트:** 토큰 실행을 자동화합니다.
- **오라클 네트워크(예: Pyth)**
 - **쿼리:** 체인에서 데이터를 소싱합니다.
 - **메세징:** 네트워크 전반에 변조 방지 데이터 릴레이를 보장합니다.

메세징(Messaging)

메세징 개요

웜홀 메세징은 웜홀 생태계의 핵심 프로토콜로, 블록체인 간 안전하고 빠른 통신을 가능하게 하는 범용 멀티체인 메시지 전달 계층입니다. 이는 블록체인 고립 문제를 해결하여 데이터와 자산이 네트워크 간 자유롭게 이동할 수 있게 하며, 개발자가 진정한 멀티체인 애플리케이션을 구축하도록 돕습니다.

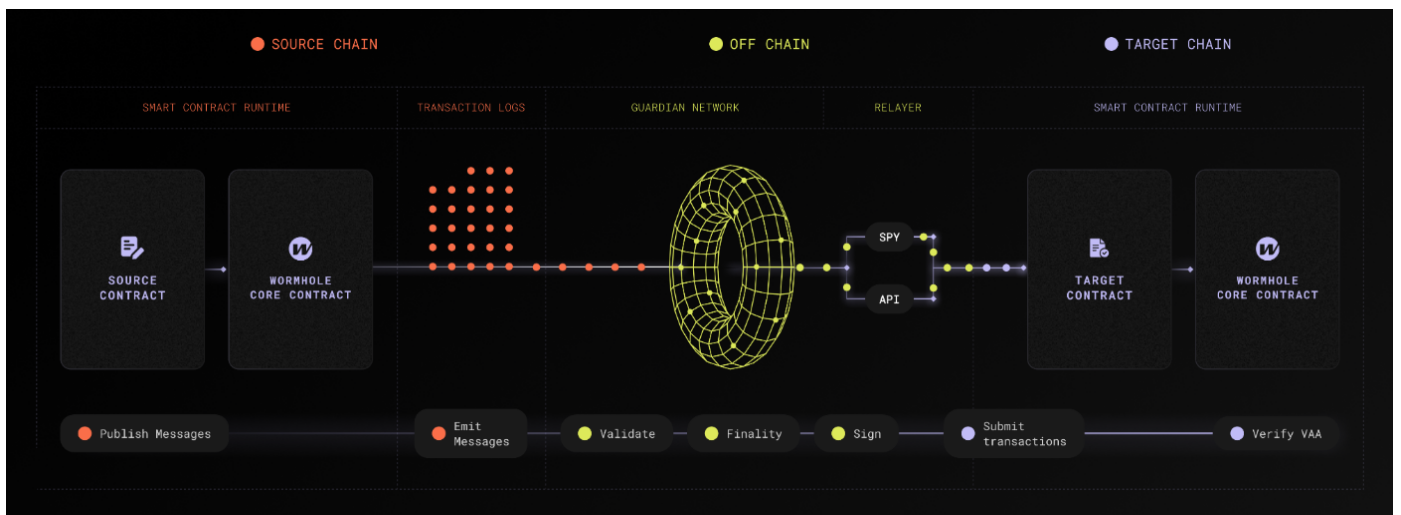
주요 기능

- **멀티체인 메시징**: 블록체인 간 임의 데이터를 전송하여 xDapp, 거버넌스 액션, 생태계 간 조정을 가능하게 합니다.
- **탈중앙 검증**: 독립적인 가디언 네트워크가 멀티체인 메시지를 관찰·서명하여 무결성을 보장하는 Verifiable Action Approvals(VAAs)을 생성합니다.
- **조합 가능한 아키텍처**: 스마트 컨트랙트, 토큰 브리지 또는 탈중앙 애플리케이션과 함께 동작하여 멀티체인 사용 사례를 위한 유연한 기반을 제공합니다.

동작 방식

메시징 흐름은 다음 핵심 구성 요소로 이루어집니다.

1. **소스 체인(Emitter 컨트랙트)**: 컨트랙트가 소스 체인의 웜홀 코어 컨트랙트를 호출하여 메시지를 발생시킵니다.
2. **가디언 네트워크**: 가디언이 메시지를 관찰·검증하고 서명된 VAA를 생성합니다.
3. **릴레이어**: 오프체인 또는 온체인 릴레이어가 VAA를 대상 체인으로 전송합니다.
4. **대상 체인(수신 컨트랙트)**: 대상 체인의 코어 컨트랙트가 VAA를 검증하고 지정된 애플리케이션 로직을 트리거합니다.



사용 사례

웜홀 메세징은 광범위한 멀티체인 애플리케이션을 가능하게 합니다. 아래는 일반적인 사용 사례와 구축 시 사용할 수 있는 웜홀 스택 구성 요소입니다.

- **체인 간 대출/차입(예: Folks Finance)**
 - **메세징**: 체인 간 동작을 조정합니다.

- **Native Token Transfers:** 담보를 네이티브 자산으로 전송합니다.
- **쿼리:** 실시간으로 금리와 가격을 가져옵니다.
- **오라클 네트워크(예: Pyth)**
 - **메세징:** 검증된 데이터를 릴레이합니다.
 - **쿼리:** 멀티체인 소스를 집계합니다.
- **가스 추상화**
 - **메세징:** 가스 로직을 조정합니다.
 - **Native Token Transfers:** 네이티브 토큰 스왑을 처리합니다.
- **브리징 인텐트 라이브러리**
 - **메세징:** 인텐트를 디스패치하고 실행합니다.
 - **세틀먼트:** 사용자 정의 브리징 인텐트를 실행합니다.
- **탈중앙 소셜 플랫폼(예: Chingari)**
 - **메세징:** 탈중앙 상호작용을 촉진합니다.
 - **Wrapped Token Transfers:** 토큰화된 보상을 가능하게 합니다.