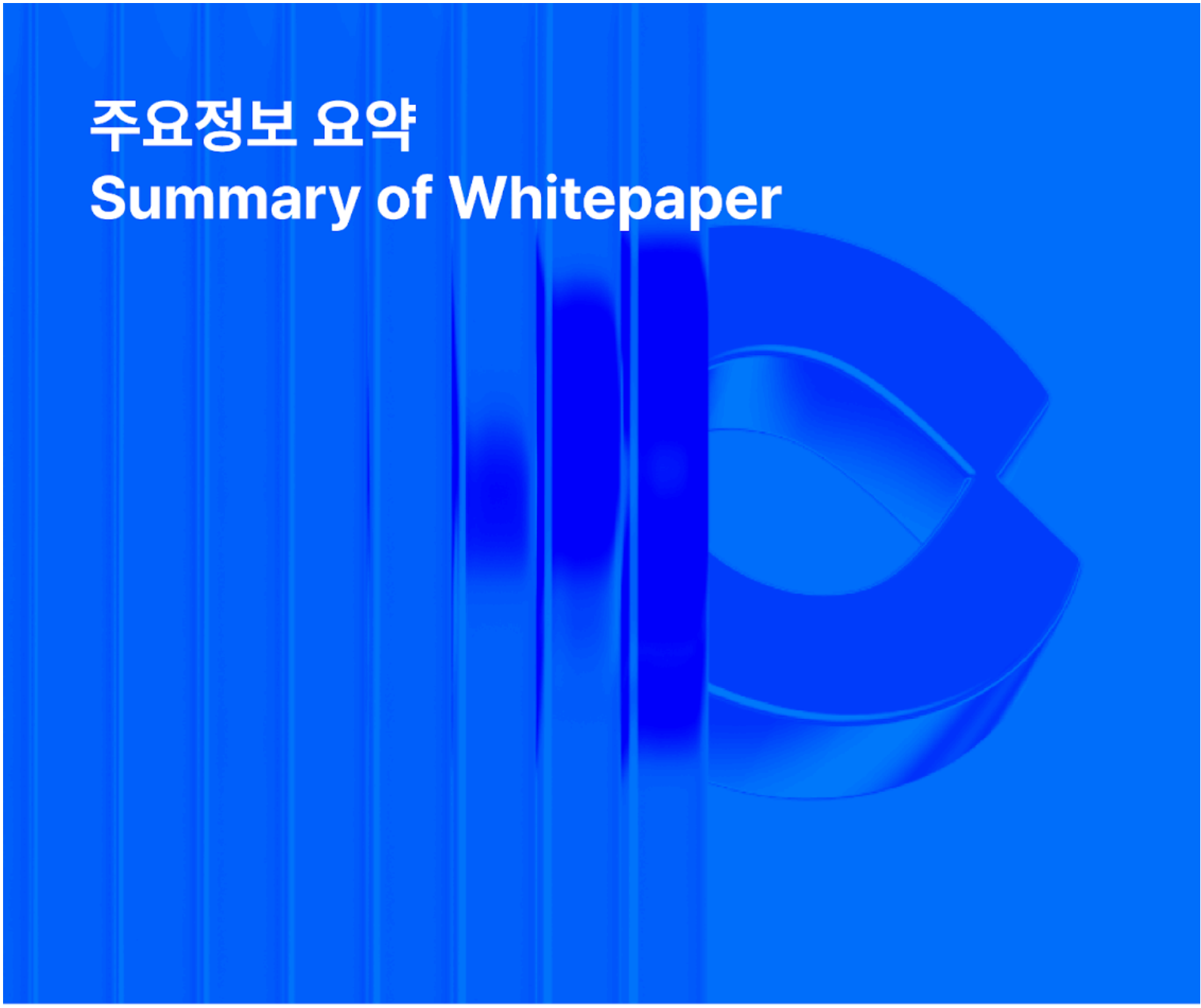
주요정보 요약
Summary of Whitepaper



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

모나드(MON)

2025.11.18

기본 정보

가상자산 카테고리	유틸리티
거래지원 네트워크	Monad Mainnet
홈페이지	https://www.monad.xyz/
참고문헌 (백서, Docs 등)	https://www.xinfin.org/docs/whitepaper-tech.pdf https://www.monad.xyz/announcements/mon-tokenomics-overview

1. 프로젝트 정보

소개

Monad는 고성능, 완전한 탈중앙화, 그리고 EVM 호환성을 제공하는 레이어-1 블록체인입니다. Monad의 목표는 탈중앙화의 효용을 극대화하고, 탈중앙화와 성능 사이의 트레이드오프를 제거하는 데 있습니다.

Monad는 전 세계적으로 분산된 대규모 네트워크를 지원하며, 누구나 노드를 운영할 수 있도록 최소한의 하드웨어 요구사항을 유지합니다. 성능은 고사양 장비나 노드의 물리적 집중이 아니라 소프트웨어 아키텍처 개선을 통해 달성합니다.

Monad의 코드베이스는 전면 오픈소스로 제공되며(컨센서스: monad-bft, 실행: monad), C++와 Rust 기반으로 고성능을 목표로 개발되었습니다.

Monad는 다음 다섯 가지 주요 영역에서 새로운 아키텍처를 도입합니다.

- MonadBFT: tail-forking 문제를 해결하는 차세대 BFT 합의 메커니즘
- RaptorCast: 효율적인 블록 전송 구조
- 비동기 실행(Asynchronous Execution): 합의와 실행의 파이프라이닝을 통해 실행 시간 예산을 확대
- 병렬 실행(Parallel Execution) 및 JIT 컴파일: 효율적인 트랜잭션 실행
- MonadDb: 이더리움 상태 저장을 위한 효율적인 스토리지 구조

Monad의 개선점은 기존 병목 구간을 해결하면서도 개발자(EVM 바이트코드 완전 호환)와 사용자(Ethereum RPC API 호환) 경험을 유지합니다.

Why Blockchain?

블록체인은 다양한 참여자들이 두 가지 사항에 대해 탈중앙화 방식으로 합의하는 시스템입니다. 첫째는 트랜잭션의 공식적 순서(ledger)이며, 둘째는 계정 잔액과 여러 프로그램의 상태를 포함한 세계의 공식적 상태입니다.

이더리움과 같은 현대적 블록체인에서 트랜잭션은 잔액 전송, 신규 프로그램 생성, 기존 프로그램에 대한 함수 호출로 구성됩니다. 지금까지의 모든 트랜잭션의 누적 결과가 현재

상태를 형성하므로, 트랜잭션 순서에 대한 합의는 곧 상태에 대한 합의를 의미합니다.

블록체인 시스템은 프로토콜 규칙(컨센서스 메커니즘)을 가지고 있으며, 이는 현재 동기화된 분산 노드들이 새로운 트랜잭션을 어떻게 상호 통신하고 원장에 추가할지에 대한 규칙입니다. MonadBFT는 이러한 컨센서스 메커니즘의 한 예입니다.

블록체인은 귀납적 방식으로 노드 간 일관성을 유지합니다. 모든 노드는 동일한 상태에서 시작하며 동일한 트랜잭션을 동일한 방식으로 적용합니다. 따라서 새로운 트랜잭션 목록을 적용한 이후에도 상태가 일관되게 유지됩니다.

공유 글로벌 상태는 탈중앙화 애플리케이션(Dapp)의 개발을 가능하게 합니다. 탈중앙화 앱은 블록체인 시스템의 모든 노드 위에서 실행되는 코드 조각과 그에 따른 지속적 상태로 구성됩니다. 사용자는 해당 앱의 특정 함수를 호출하는 트랜잭션을 제출하여 앱을 실행합니다. 블록체인의 모든 노드는 호출된 바이트코드를 정확히 실행할 책임이 있으며, 동일한 실행의 중복 구조가 각 노드의 정직성을 보장합니다.

왜 모나드인가 : 탈중앙화 + 성능
탈중앙화의 중요성

블록체인은 크게 두 가지 주요 구성요소로 이루어집니다. 첫째는 원장에 추가할 트랜잭션에 대한 합의를 형성하는 컨센서스 메커니즘이며, 둘째는 활성 상태를 유지하는 실행·스토리지 시스템입니다. 성능을 높이기 위해 모든 노드를 물리적으로 가까운 위치에 두어 합의 비용을 줄이거나, 대량의 RAM을 요구해 상태 전체를 메모리에 보관하는 방식으로 단순화할 수 있으나, 이는 탈중앙화를 심각하게 훼손합니다.

탈중앙화는 블록체인의 핵심 가치입니다. Why Blockchain 섹션에서 설명한 바와 같이, 탈중앙화된 공유 글로벌 상태는 여러 참여자가 단일하고 객관적인 진실을 기반으로 상호 조정할 수 있도록 합니다. 소규모 노드 운영자 집단(혹은 극단적인 경우 단일 운영자)이 유지하는 블록체인은 무신뢰성(trustlessness), 공정성, 검열 저항과 같은 이점을 제공하지 못합니다.

따라서 모든 블록체인 네트워크는 탈중앙화를 최우선 요소로 삼아야 하며, 성능 향상이 탈중앙화를 희생하는 방향으로 이루어져서는 안 됩니다.

현재 성능 병목 요인

이더리움의 현행 실행 제한(초당 1.25M 가스)은 매우 보수적으로 설정되어 있으며, 다음과 같은 여러 이유가 존재합니다.

- 비효율적인 스토리지 접근 패턴
- 단일 스레드 기반 실행 구조
- 실행 없이는 컨센서스가 진행될 수 없어 실행 예산이 매우 제한적임
- 상태(state) 증가에 대한 우려 및 향후 상태 접근 비용 증가 문제

Monad는 이러한 제한을 알고리즘 개선과 아키텍처 변화로 해결하며, 향후 이더리움에도 적용될 가능성이 있는 새로운 구조를 제안합니다. 높은 수준의 탈중앙화를 유지하면서 실질적인 성능 향상을 달성하는 것이 핵심입니다.

최적화를 통한 병목 해결

Monad는 네 가지 주요 영역에서 파이프라이닝과 다양한 최적화를 적용하여 EVM 성능을 크게 향상시키고, 탈중앙화와 확장성 사이의 트레이드오프를 개선합니다. 이후 페이지에서 각 영역을 자세히 설명합니다.

- MonadBFT: tail-forking 문제를 해결하는 차세대 BFT 합의 구조
- RaptorCast: 효율적인 블록 전송 구조
- 비동기 실행(Asynchronous Execution): 합의와 실행의 파이프라이닝을 통한 실행 시간 예산 확대
- 병렬 실행(Parallel Execution)과 JIT 컴파일
- MonadDb: 이더리움 상태 저장을 위한 효율적 스토리지 구조

사용자 관점에서 보는 Monad

Monad는 고성능과 이더리움 호환성을 동시에 제공하는 레이어-1 블록체인으로, 사용자에게 **이식성**과 **성능**을 모두 제공합니다.

이식성 측면에서 Monad는 이더리움 가상머신(EVM)에 대해 **완전한 바이트코드 호환성**을 제공합니다. 즉, 이더리움용으로 개발된 애플리케이션은 코드 변경 없이 그대로 Monad에서 실행할 수 있습니다. 또한 **이더리움 RPC 완전 호환성**을 제공하므로 Etherscan, The Graph와 같은 인프라도 그대로 사용할 수 있습니다.

성능 측면에서 Monad는 **초당 10,000 tps** 처리량(하루 10억 건 트랜잭션), **400ms 블록 주기**, **800ms 파이널리티**를 제공합니다. 이를 통해 기존 블록체인보다 훨씬 많은 사용자와 더 높은 상호작용성을 제공하며, 트랜잭션 비용도 매우 낮게 유지합니다.

Monad의 익숙한 점

사용자 관점에서 Monad는 이더리움과 매우 유사하게 동작합니다. 동일한 지갑(Phantom, MetaMask 등)과 블록 탐색기(Etherscan 등)를 사용하여 트랜잭션을 확인하거나 서명할 수 있습니다. 이더리움용 애플리케이션은 코드 수정 없이 Monad로 이식할 수 있으므로, 이더리움에서 사용하던 주요 애플리케이션을 Monad에서도 동일하게 사용할 수 있습니다. 주소 체계도 이더리움과 동일하므로 기존 키를 그대로 사용할 수 있습니다.

이더리움과 마찬가지로 Monad는 선형 블록 구조를 가지며, 블록 내부의 트랜잭션도 선형 순서를 유지합니다.

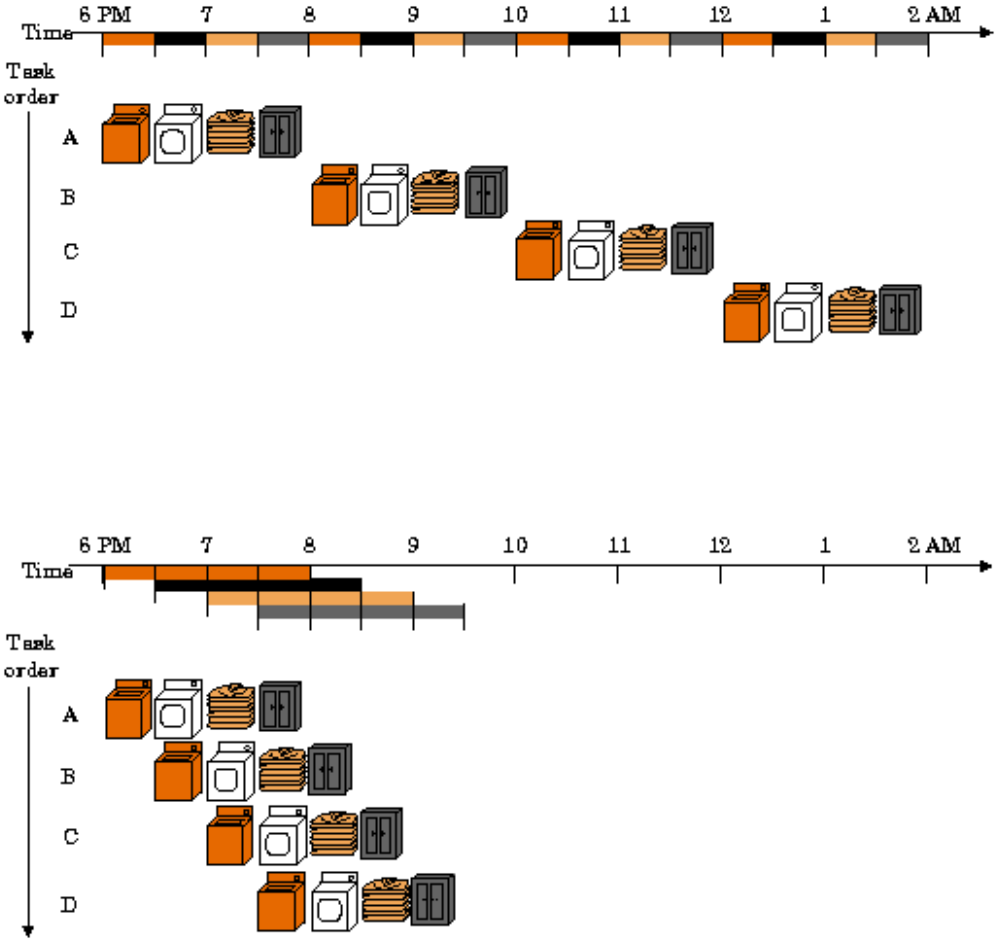
또한 이더리움과 같이 Monad는 탈중앙화된 검증자 세트가 운영하는 지분증명(PoS) 네트워크입니다. 누구나 노드를 운영하여 트랜잭션 실행을 독립적으로 검증할 수 있으며, 하드웨어 요구사항을 낮게 유지하기 위해 많은 주의를 기울였습니다.

Monad의 차별점

Monad는 Ethereum Virtual Machine에 ****병렬 실행(Parallel Execution)****과 ****슈퍼스칼라 파이프라이닝(Superscalar Pipelining)****을 도입하여 기존을 뛰어넘는 성능을 제공합니다.

병렬 실행은 멀티코어와 멀티스레드를 활용하여 내부적으로 여러 작업을 병렬 처리하되, 최종 결과는 원래 트랜잭션 순서대로 커밋되도록 하는 구조입니다. 사용자와 개발자 관점에서는 직렬 실행처럼 보이며, 실행 결과도 직렬 실행과 동일합니다.

슈퍼스칼라 파이프라이닝은 여러 작업 단계를 생성하고 이를 병렬적으로 실행하는 방식입니다.



예시 그림: Pipelining laundry day. Top: Naive; Bottom: Pipelined. Credit: Prof. Lois Hawkes, FSU
출처 : https://docs.monad.xyz/introduction/monad-for-users

예를 들어 네 번의 세탁을 할 때, 순차적으로 세탁-건조-정리-보관을 마친 뒤 다음 세탁을 시작하는 방식이 기존 방식이고, 파이프라이닝 방식은 1번 세탁이 건조기로 넘어가는 순간 2번 세탁을 시작하는 방식입니다. 여러 자원을 동시에 사용해 전체 작업 효율을 높이는 개념입니다.

Monad는 파이프라이닝을 도입하여 상태 저장, 트랜잭션 처리, 분산 합의 구조에서 발생하는 기존 병목을 해결합니다. 구체적으로 Monad는 다음 다섯 영역에서 파이프라이닝 및 최적화를 적용합니다.

- MonadBFT: 고성능·포크 저항성을 갖춘 BFT 합의
- RaptorCast: 효율적인 블록 전송
- 비동기 실행(Asynchronous Execution): 합의와 실행을 파이프라이닝하여 실행 시간 예산 확대

- 병렬 실행(Parallel Execution) 및 JIT 컴파일
- MonadDb: 효율적인 상태 접근 구조

C++와 Rust로 처음부터 새롭게 구현된 Monad 클라이언트는 이러한 아키텍처 개선을 반영하며, 전 세계적 규모의 채택을 목표로 하는 탈중앙화 애플리케이션 플랫폼을 제공합니다.

왜 중요한가?

탈중앙화 애플리케이션은 기존 중앙화 서비스의 대안으로 다음과 같은 장점을 제공합니다.

- **오픈 API / 조합성**: 탈중앙화 앱끼리 원자적으로 호출할 수 있어 기능 조합이 용이함
- **투명성**: 앱 로직이 코드를 통해 완전히 공개되어 있으며, 상태는 투명하고 검증 가능함
- **검열 저항성과 공정성**: 누구나 트랜잭션을 제출하거나 애플리케이션을 업로드할 수 있음
- **글로벌 접근성**: 인터넷만 있으면 누구나 금융 서비스에 접근 가능

그러나 탈중앙화 앱이 본래 목표한 영향을 실현하기 위해서는 저렴하고 고성능인 인프라가 필요합니다. 예를 들어 하루 100만 명의 활성 사용자(DAU)가 1인당 트랜잭션 10건을 사용하는 앱은 하루 1,000만 건 트랜잭션, 즉 초당 100 tps가 필요합니다. L2Beat 자료를 확인해보면 현재 어떤 EVM 레이어-1 또는 레이어-2도 이러한 처리량을 만족하지 못합니다.

Monad는 EVM 호환 블록체인의 성능을 구조적으로 개선하여, 향후 이더리움 발전 과정에서도 표준이 될 수 있는 혁신을 제시합니다.

Monad를 통해 개발자, 사용자, 연구자는 기존 EVM 생태계에서 축적된 애플리케이션, 라이브러리, 암호 연구 성과를 그대로 활용할 수 있습니다.

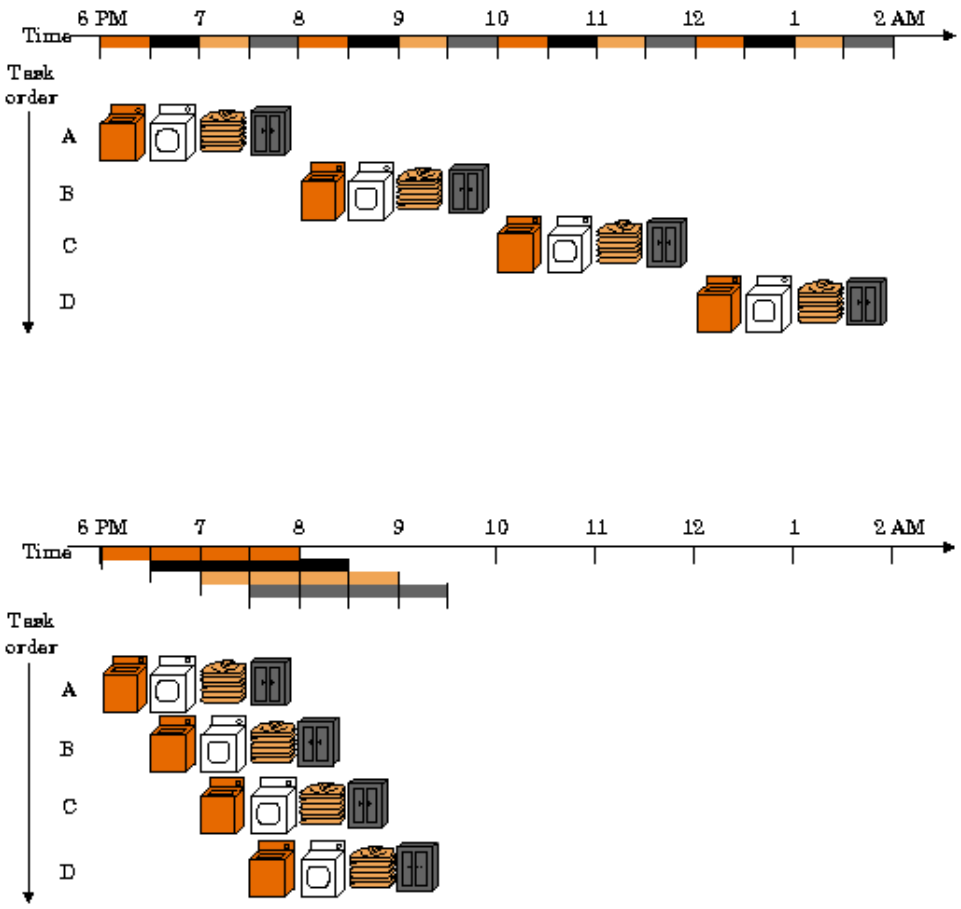
아키텍쳐

파이프라이닝(Pipelining)

파이프라이닝은 하나의 작업을 더 작은 여러 단계로 분할하여 병렬로 처리할 수 있도록 하는 기법입니다.

파이프라이닝은 컴퓨터 프로세서에서 동일한 클록 주기 내에서 연속된 명령 실행의 처리량을 높이기 위해 사용됩니다. 프로세서에서는 처리량 향상을 위해 파이프라이닝 외에도 다양한 기법이 활용됩니다. 명령어 수준 병렬성(Instruction-Level Parallelism, ILP)에 대한 더 자세한 내용은 관련 문서를 참고할 수 있습니다.

파이프라이닝의 단순한 예시는 아래와 같습니다.



출처 : https://docs.monad.xyz/monad-arch/concepts/pipelining

네 번의 세탁을 할 때, 순차적 방식은 1번 세탁을 세탁–건조–정리–보관까지 모두 마친 뒤 2번 세탁을 시작합니다. 반면, 파이프라이닝 방식에서는 1번 세탁이 건조기로 이동하는 순간 2번 세탁을 시작합니다. 이 방식은 여러 자원을 동시에 활용하여 작업 효율을 높이는 구조입니다.

MonadBFT

MonadBFT는 비잔틴 장애 허용(Byzantine Fault Tolerant, BFT) 합의 분야에서 큰 도약을 이루는 합의 프로토콜입니다. MonadBFT는 1만 건 이상 초당 트랜잭션 처리(10,000+ tx/s), 1초 미만 파이널리티, 그리고 대규모 합의 노드 세트(validator set)를 지원하면서, Monad 네트워크가 유효한 블록 제안에 대해 효율적이고 안전하게 합의하도록 책임을 집니다.

MonadBFT는 이러한 특성을 모두 제공하는 동시에, 파이프라인 구조의 리더 기반 BFT 프로토콜에서 리더가 직전 리더의 블록을 포크(fork)해 버릴 수 있는 **tail-forking** 취약점에 대해 내재적인 내성을 가집니다.

MonadBFT에 대한 전체 설명과 기술적 세부 내용은 다음 자료를 참고합니다.

- MonadBFT 연구 논문: https://arxiv.org/abs/2502.20692
- Category Labs 블로그: https://www.category.xyz/blogs/monadbft-fast-responsive-fork-resistant-streamlined-consensus

MonadBFT는 다음을 달성합니다.

- 한 번의 라운드에서 **추정적 파이널리티(speculative finality)** 를 제공하고, 두 번의 라운드에서 **완전한 파이널리티**를 제공합니다.
- 정상(happy path) 상황에서 **메시지 및 서명 검증 복잡도가 선형(linear)** 이므로, 합의 노드 수를 수백 개까지 확장할 수 있습니다.
- **낙관적 응답성(optimistic responsiveness)** 을 지원하여, 일반적인 경우뿐만 아니라 실패 라운드에서 회복할 때에도 최악의 네트워크 지연을 기다리지 않고 라운드를 진행합니다.
- 파이프라인 리더 기반 BFT 합의에서 리더가 직전 블록을 포크해 제거할 수 있는 **tail-forking 공격에 대한 내재적 방어** 를 제공합니다. 이는 기존 파이프라인 리더 기반 BFT 합의 메커니즘에서 존재했던 중대한 취약점을 해결합니다.

핵심 특성

MonadBFT가 제공하는 주요 특성은 다음과 같습니다.

1. **추정적 파이널리티(1라운드)**
    - 한 번의 라운드에서 블록을 `Voted` 상태로 올리는 시점부터, 해당 블록이 사실상 파이널리티에 매우 근접한 상태로 간주할 수 있습니다.
    - 이 추정적 파이널리티는 이중 서명(equivocation), 리더 실패, 상당한 비잔틴 비율 등이 동시에 발생하는 매우 드문 경우에만 되돌아갈 수 있습니다.
2. **완전 파이널리티(2라운드)**
    - 두 번의 연속된 라운드를 거치면 블록은 `Finalized` 상태가 됩니다.
    - 이는 이론적으로도, 구현 상으로도 800ms 수준의 파이널리티를 목표로 합니다.
3. **선형(Linear) 메시지 복잡도**
    - 각 라운드는 리더 → 검증자(브로드캐스트), 검증자 → 다음 리더(투표) 구조로 통신합니다.
    - 모든 노드가 서로 통신하는 all-to-all(제곱형) 통신이 아니라, 선형 통신 패턴을 사용해 검증자 수를 수백 개 수준까지 확장할 수 있습니다.
4. **낙관적 응답성(Optimistic Responsiveness)**
    - 네트워크가 평균적으로 양호한 상황에서는, 이론상 최악의 네트워크 지연을 기다리지 않고 라운드를 진행합니다.
    - 실패한 라운드에서 복구할 때에도, 동일한 철학으로 빠른 라운드 전환과 재제안을 지원합니다.

5. **Tail-forking 방어**

    - 기존 파이프라인 HotStuff 계열 프로토콜에서 발생하던 “뒤쪽 블록이 꼬리처럼 잘려 나가는” tail-forking 문제를 구조적으로 차단합니다.

    - 리더가 의도적으로 QC 생성을 방해해 직전 블록을 버리고 MEV를 재추출하는 공격 벡터를 제거하는 방향으로 설계되어 있습니다.

기본 합의 구조 및 개념
BFT 기본 모델

- 노드 수는 `n = 3f + 1` 로 가정하며, 이 중 최대 `f`개 노드가 비잔틴(장애) 노드일 수 있습니다.

- 초과다수(supermajority)는 지분 비중 기준 2/3 이상(= `2f+1`)을 의미합니다.

- 합의는 라운드(또는 뷰, view) 단위로 진행되며, 각 라운드마다 리더가 하나씩 존재합니다.

주요 객체

- **블록(Block)**: 트랜잭션 목록(payload), QC, 블록 번호로 구성되며, 블록 번호는 부모 블록 번호 + 1입니다.

- **QC(Quorum Certificate)**: 특정 제안에 대해 초과다수의 YES 투표가 있었다는 증거입니다. MonadBFT는 BLS 서명 집계를 활용해 QC 검증 비용을 줄입니다.

- **TC(Timeout Certificate)**: 일정 시간 내 유효 제안이 오지 않아 라운드가 실패했다는 것을 초과다수 서명을 통해 증명하는 객체입니다. 타임아웃 메시지들로부터 만들어집니다.

- **NEC(No-Endorsement Certificate)**: 특정 블록을 “보지 못했다”는 승인 없음 메시지가 초과다수 모였을 때 생성되며, 그 블록이 네트워크에 충분히 전파되지 않았음을 증명합니다.

블록 상태

MonadBFT 관점에서 블록은 세 가지 상태를 가집니다.

1. `Proposed` – 리더가 제안하고, 검증자가 수신·검증한 단계

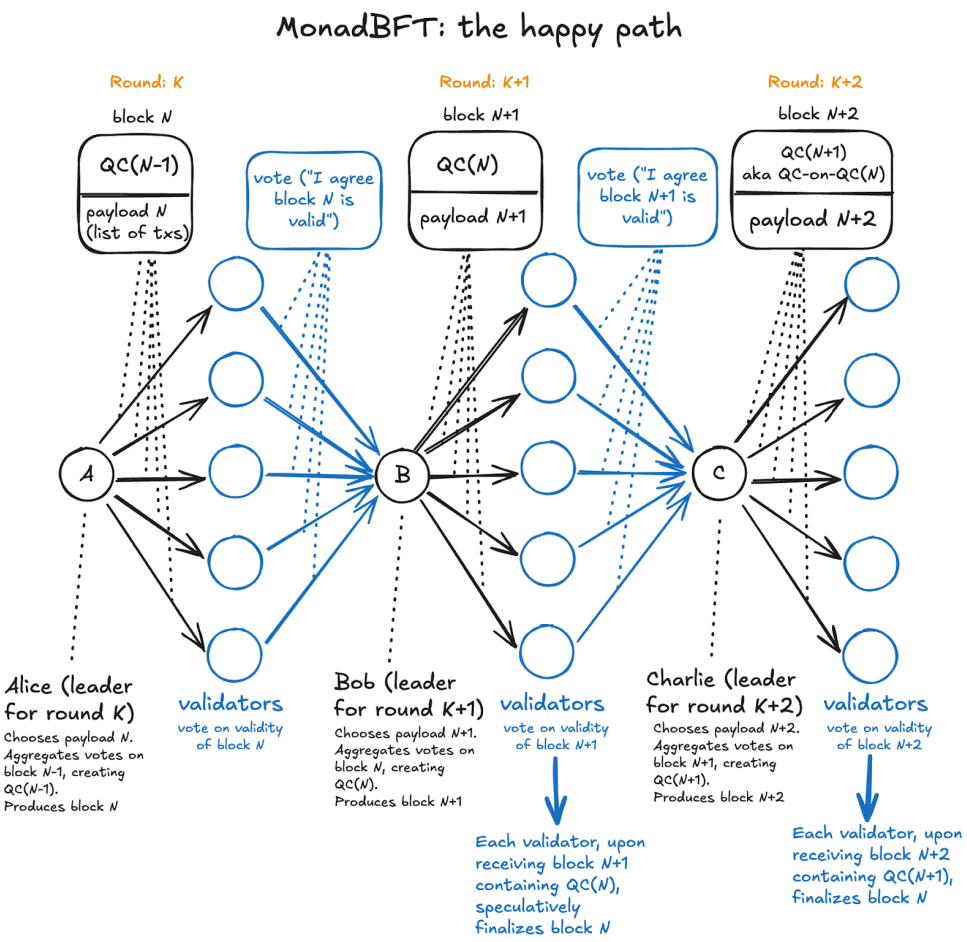
2. `Voted` – 해당 블록에 대한 QC가 형성된 단계(추정적 파이널리티 부여 가능)

3. `Finalized` – 손자 블록까지 이어지는 구조에서 완전 파이널리티에 도달한 단계

`Verified` 상태는 실행 레이어(비동기 실행)에서 별도로 다루는 개념이므로, 합의 메커니즘 설명에서는 제외됩니다.

정상 경로(Happy Path): 기본 동작 흐름

정상 경로에서는 라운드가 연속적으로 성공하며, 블록이 다음과 같이 진행됩니다.

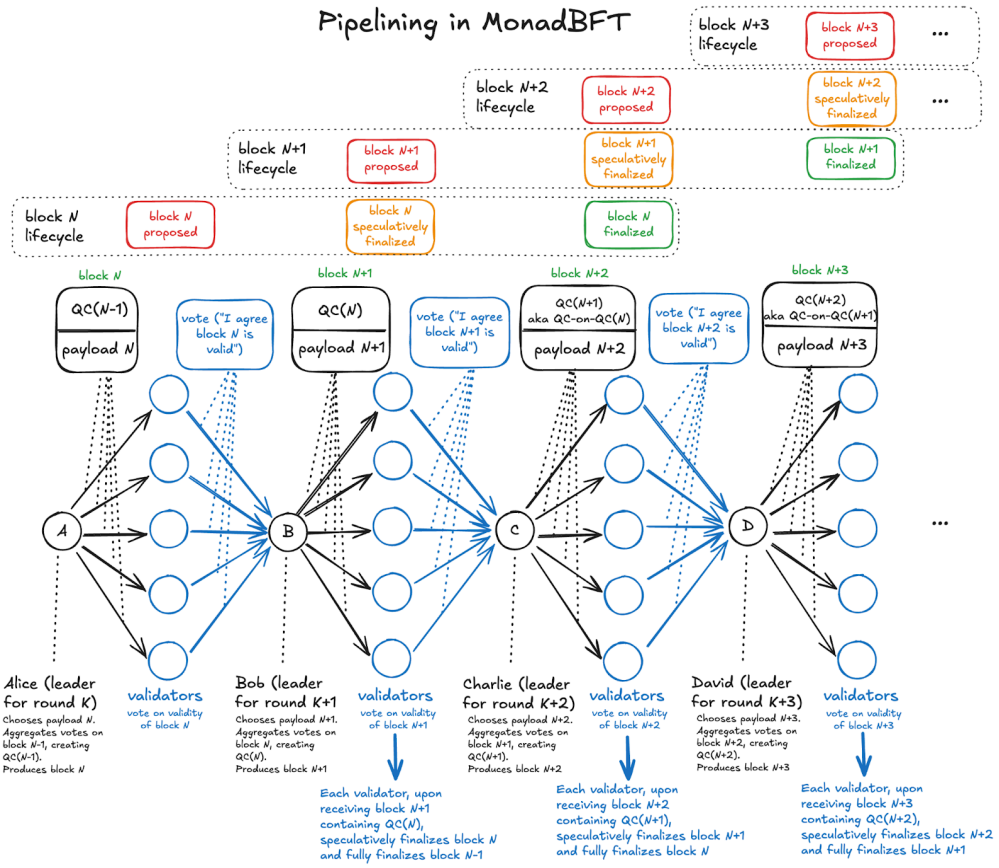


출처 : https://docs.monad.xyz/monad-arch/consensus/monad-bft

1. **라운드 K – Alice의 제안**
    - 리더 Alice가 멤풀에서 트랜잭션을 선택해 페이로드를 구성하고, 이전 QC를 포함한 블록 `N`을 제안합니다.
    - 검증자들은 Alice 제안이 유효하면 투표를 생성해 다음 리더 Bob에게 전송하고, 로컬에서 `Proposed`로 표시합니다.
    - Bob이 초과다수의 YES 투표를 수신하면 이를 집계해 Alice 제안에 대한 QC를 생성합니다.
2. **라운드 K+1 – Bob의 제안**
    - Bob은 새로운 페이로드와 Alice 제안에 대한 QC를 포함해 블록 `N+1`을 제안합니다.
    - 검증자들은 제안을 검증하고 YES인 경우 Charlie에게 투표를 전송하며, 블록 `N`을 `Voted`, `N+1`을 `Proposed`로 표시합니다.

- 이 시점에서 블록 N은 추정적 파이널리티를 확보합니다.

3. **라운드 K+2 – Charlie의 제안**
   - Charlie는 Bob 제안에 대한 QC를 포함한 새 블록을 제안합니다.
   - 검증자들은 제안이 유효하면 David에게 투표를 전송하고, 블록 N을 `Finalized`, N+1을 `Voted`, N+2를 `Proposed`로 올립니다.



출처 : https://docs.monad.xyz/monad-arch/consensus/monad-bft

이 구조 덕분에, 매 라운드마다 “새 페이로드 + 이전 제안에 대한 QC”가 전파되며, 부모 블록은 추정적으로, 조부모 블록은 완전히 파이널라이즈되는 파이프라인 구조가 형성됩니다.

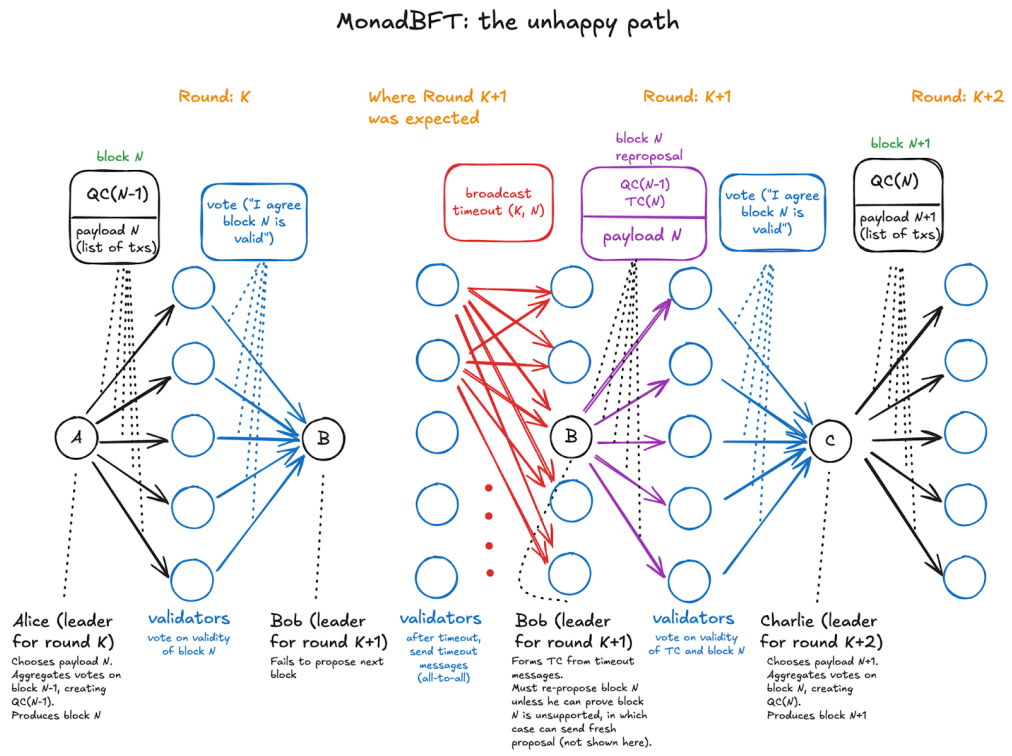
비정상 경로(Unhappy Path)와 복구 메커니즘

비정상 경로는 리더가 제안에 실패하거나, QC를 만들어야 하는 리더가 동작하지 않는 경우를 다룹니다.

대표적인 시나리오:

- Alice가 블록 N을 라운드 K에서 제안하고, 검증자들이 Bob에게 투표를 보냈으나,
- Bob이 라운드 K+1에서 블록을 제안하지 못한 경우입니다.

이때 복구 절차는 다음과 같습니다.



출처 : https://docs.monad.xyz/monad-arch/consensus/monad-bft

1. **타임아웃 발생 및 TC 생성**
    - 일정 시간 후 QC가 형성되지 않으면, 검증자들은 라운드 `K`에 대한 타임아웃 메시지를 all-to-all로 주고받습니다.
    - 초과다수의 타임아웃 메시지가 모이면, 라운드 `K` 실패를 증명하는 TC가 생성되며, 라운드는 `K+1`로 넘어갑니다.
    - TC에는 각 검증자의 Tip 정보가 포함되어 있으며, 이를 바탕으로 가장 높은 Tip인 `high_tip`이 계산됩니다. 이 예시에서는 Alice 블록이 `high_tip`가 됩니다.
2. **다음 리더의 의무: 재제안 또는 NEC**
    - MonadBFT 규칙상, 다음 리더(Bob)는 TC에 포함된 `high_tip`에 해당하는 블록(Alice 블록)을 재제안하거나, 초과다수로부터 NEC를 확보해 “해당 블록은 네트워크에 충분히 퍼지지 않았다”는 것을 증명해야 합니다.
    - Bob이 Alice 블록 전체를 가지고 있지 않은 경우, 다른 검증자에게 blocksync를 통해 블록 본문을 요청할 수 있습니다.
3. **세 가지 케이스**

    - **재제안(Reproposal)**: Bob이 Alice 블록을 확보했다면, TC와 함께 재제안하며, 이후 Charlie가 QC를 만들고 다시 정상 경로로 복귀합니다.
    - **새로운 제안(Fresh Proposal)**: Bob이 NEC를 확보한 경우에만, 같은 높이 `N`에서 새로운 블록을 제안할 수 있습니다. 이 조건은 임의로 기존 블록을 버리고 새 블록을 넣는 행위를 통제하기 위한 것입니다.
    - **미제안 반복(No Proposal)**: Bob이 다시 아무 제안도 하지 못하면, 라운드 `K+1`도 타임아웃되고 라운드 `K+2`로 넘어가며, 다음 리더가 동일한 의무(재제안 또는 NEC 확보)를 이어받습니다.

이 규칙 덕분에, "정상적으로 지지받아야 할 블록"은 결국 파이널라이즈되고, 실제로 네트워크에 충분히 전파되지 않은 블록만 NEC를 통해 제거될 수 있습니다.

Tail-Forking 방지 메커니즘

기존 파이프라인 HotStuff 계열에서 tail-forking이 발생하는 패턴은 대략 다음과 같습니다.

- Alice 블록이 제안됐지만, Bob이 QC를 만들지 않거나 제안 자체를 하지 않고 슬롯을 놓칩니다.
- 이후 프로토콜이 Bob 슬롯 실패만을 근거로 Alice 블록을 건너뛰게 되면, Bob은 MEV를 재추출하기 위해 Alice 트랜잭션을 제거하거나 재정렬한 새 블록을 같은 높이에 제안할 수 있습니다.

MonadBFT에서는 TC가 Alice 블록의 존재를 명시적으로 전파하고, `high_tip` 규칙과 NEC 조건을 통해 다음 리더가 기존 블록을 임의로 버릴 수 없도록 합니다. 구체적으로:

- `2f+1` 투표가 Alice 블록에 대한 QC를 만들어낸 이상, 동일한 `2f+1` 규모의 서명을 필요로 하는 TC에는 적어도 `f+1`개의 공통 비비잔틴 참여자가 존재하며, 이들은 Alice 블록을 여전히 참조합니다.
- 따라서 `high_tip`가 Alice 블록을 가리킬 가능성이 매우 높으며, 리더는 NEC가 없는 이상 Alice 블록을 재제안해야 합니다.
- NEC는 초과다수가 "해당 블록을 보지 못했다"고 서명해야 만들어지므로, 이미 QC가 존재하는 블록에 대해 NEC를 만들기는 사실상 불가능합니다.

이 설계로 인해, "리더가 의도적으로 QC를 생략해 직전 블록을 잘라내는 공격"이 구조적으로 막히게 됩니다.

추정적 파이널리티와 안전성 직관

검증자 입장에서, 블록이 `Voted` 상태로 전환될 때(즉, 해당 블록에 대한 QC를 수신했을 때) 이미 다음과 같은 사실이 보장됩니다.

- 초과다수의 노드가 해당 블록을 수신하고 YES로 투표했습니다.
- 설령 다음 리더가 일부 노드에만 제안을 보내고 나머지에는 오프라인처럼 행동하더라도, 타임아웃 → TC → `high_tip` → 재제안/NEC 절차를 통해 해당 블록이 다시 떠오를 가능성이 높습니다.
- 이중 서명(equivocation)·리더 장애·상당한 비잔틴 비율 등 매우 이례적인 조건이 중첩되지 않는 한, 그 블록은 결국 파이널라이즈될 가능성이 높습니다.

따라서 MonadBFT에서 `Voted` 상태는 “거의 확정에 가까운 상태”로 간주할 수 있으며, 이는 사용자 경험 측면에서 사실상 1라운드 단위의 빠른 파이널리티를 제공하는 효과를 가져옵니다.

참고 문헌(References)

- Mohammad Mussadiq Jalalzai, Kushal Babel. *MonadBFT: Fast, Responsive, Fork-Resistant Streamlined Consensus*, 2025.
- Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. *HotStuff: BFT Consensus in the Lens of Blockchain*, 2018.
- Mohammad M. Jalalzai, Jianyu Niu, Chen Feng, Fangyu Gai. *Fast-HotStuff: A Fast and Resilient HotStuff Protocol*, 2020.
- Rati Gelashvili, Lefteris Kokoris-Kogias, Alberto Sonnino, Alexander Spiegelman, and Zhuolun Xiang. *Jolteon and Ditto: Network-adaptive efficient consensus with asynchronous fallback*, arXiv preprint arXiv:2106.10362, 2021.
- The Diem Team. *DiemBFT v4: State Machine Replication in the Diem Blockchain*, 2021.

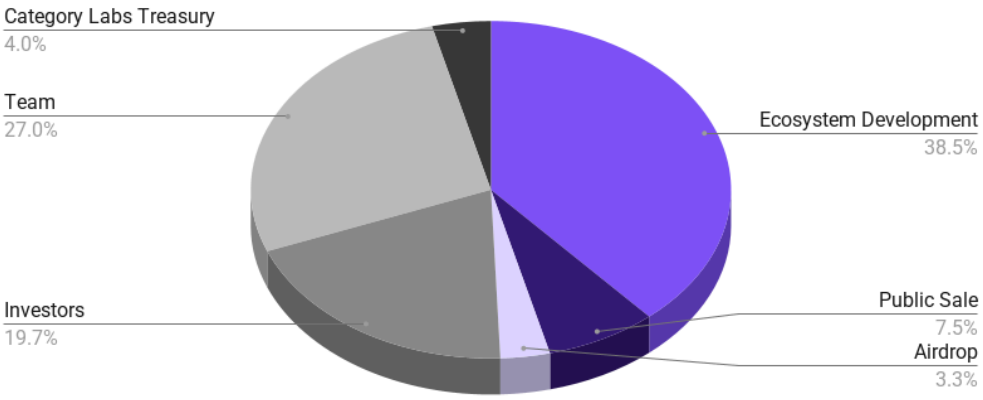
2. 토큰 이코노미

가상자산 소개

모나드(MON)은 Monad 프로토콜을 구동하는 네이티브 토큰입니다. 모나드(MON) 토큰은 주로 Monad 네트워크에서의 트랜잭션 수수료 지불 및 네트워크 보안을 위한 스테이킹에 사용됩니다.

발행량 및 유통량계획

Monad 퍼블릭 메인넷(Public Mainnet) 출시 시점의 모나드(MON) 초기 총 공급량은 1,000억 개입니다. 모나드(MON)의 그룹별 초기 배분은 아래와 같습니다.

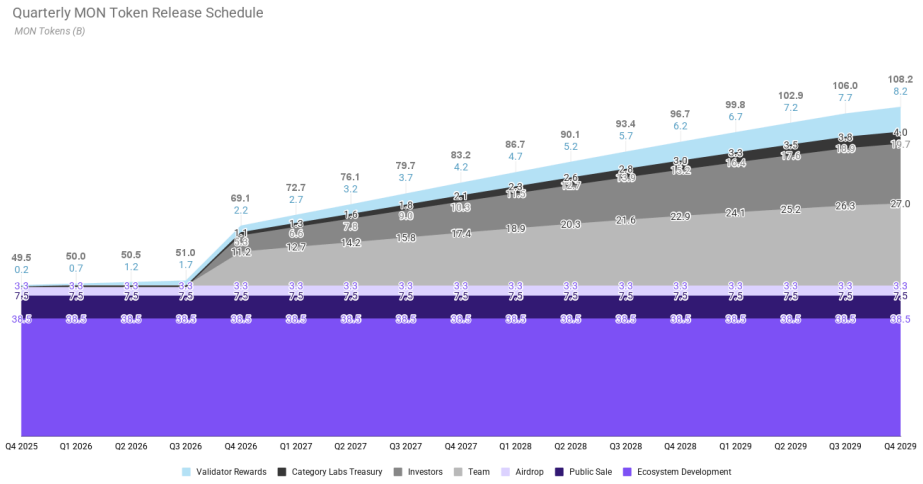


Allocation Group	Status at Public Mainnet	# Tokens	%
Ecosystem Development	Unlocked	38,544,142,854	38.5%
Team	Locked	26,989,187,887	27.0%
Investors	Locked	19,683,237,451	19.7%
Public Sale	Unlocked	7,500,000,000	7.5%
Category Labs Treasury	Locked	3,952,848,412	4.0%
Airdrop	Unlocked	3,330,583,396	3.3%
Total Initial Token Supply		**100,000,000,000**	**100.0%**

출처 :https://www.monad.xyz/announcements/mon-tokenomics-overview

아래 도표는 예정된 토큰 출시 일정을 나타냅니다. ‘Released’ 토큰은 언락(unlocked) 상태이며, 팀 물량의 경우 베스팅(vesting)을 완료한 물량을 의미합니다. 직원·투자자 물량의 순차적 언락 및 스테이킹 보상 지급에 따라 유통량은 시간이 지나면서 증가합니다.

도표 상단의 짙은 회색 레이블은 각 분기별 최대 언락 가능량을 의미하며, 검증자 보상에 따른 인플레이션을 포함합니다. 실제 언락량은 거래 수수료 소각에 따른 디플레이션 효과로 인해 도표보다 낮을 가능성이 있습니다.



출처 :https://www.monad.xyz/announcements/mon-tokenomics-overview

유통된 토큰

Monad 퍼블릭 메인넷 출시 시점에는 약 108억 모나드(MON)(10.8%)이 언락되어 유통됩니다. 이는 퍼블릭 세일과 모나드(MON) 에어드롭을 통해 배포된 물량입니다. 해당 토큰은 프로토콜 내 활동에 자유롭게 사용할 수 있습니다.

또한 약 385억 모나드(MON)(38.5%)이 생태계 개발(Ecosystem Development) 항목으로 배정되며, 이는 언락된 상태로 Monad Foundation이 관리합니다.

따라서 퍼블릭 메인넷 출시일 기준 전체 공급량의 약 494억 모나드(MON)(49.4%)이 언락 상태로 존재합니다.

락업된 토큰

투자자·팀·Category Labs Treasury 물량은 퍼블릭 메인넷 출시 시점에 모두 락업되며, 프로젝트의 장기적 성공과 정렬(alignment)을 위해 정의된 언락 및 베스팅 일정이 적용됩니다. 이들 모든 그룹은 최소 1년의 락업 기간을 가지며, 이후 그룹별 조건에 따라 언락 일정이 다르게 진행됩니다. 메인넷 출시 시점 기준 락업된 물량은 총 506억 모나드(MON)(50.6%)입니다.

락업된 토큰은 스테이킹이 불가능합니다. 이는 초기 스테이킹 보상이 공공 유통량 증가에 기여하도록 하고, 프로젝트 내부자(팀·투자자)에게 보상이 과도하게 집중되는 상황을 방지하기 위함입니다.

모든 초기 락업 물량은 2025년 11월 메인넷 출시를 기준으로, 4주년이 되는 2029년 4분기에 전량 언락될 예정입니다.

그룹별 배분 상세

최대 75억 모나드(MON)(초기 공급의 7.5%)이 Coinbase 토큰 세일 플랫폼에서 공개 판매됩니다. 판매 가격은 토큰당 0.025달러입니다. 퍼블릭 세일은 퍼블릭 메인넷 출시 직전 일반 참여자가 모나드(MON)을 구매할 수 있는 기회를 제공합니다. 판매 종료와 동시에 모나드(MON) 토큰은 구매자에게 분배됩니다. 퍼블릭 세일이 미달될 경우, 미판매 물량은 생태계 개발 항목으로 재배정됩니다.

- **에어드롭(Airdrop)**

  약 33억 모나드(MON)(3.3%)이 모나드(MON) 에어드롭을 통해 Monad 커뮤니티 구성원 및 넓은 범위의 암호화폐 커뮤니티에 분배되었습니다. 대상 기준 및 상세 내용은 'The MON Airdrop' 문서를 참고합니다.

- **생태계 개발(Ecosystem Development)**

  약 385억 모나드(MON)(38.5%)이 생태계 개발을 위해 배정됩니다. 이는 언락된 물량으로, 현재 및 미래의 생태계 확장 활동을 위해 활용됩니다. Monad Foundation은 해당 물량을 장기간 운영하며, 전략적인 보조금·인센티브 지급 및 Validator Delegation Program 운영을 담당합니다.

  이 항목에는 팀 물량이 포함되지 않으며, 팀 물량은 별도 항목으로 구성됩니다.

  현재 생태계 개발 물량 중 2% 미만만이 이미 여러 프로젝트와 인프라 제공자들에게 배정된 상태이며, 대부분의 물량은 메인넷 출시 이후 수년간 사용될

예정입니다. 또한 향후 일부는 Monad Foundation 운영비용으로 사용될 수 있습니다.

초기에는 생태계 개발 물량 중 상당 부분이 네트워크 보안 강화를 위해 Delegation Program을 통해 스테이킹됩니다. 초기 약 150억 모나드(MON)을 위임할 예정이며, 1년 차에는 150억~250억 모나드(MON) 범위에서 위임할 계획입니다. Delegation Program을 통해 발생하는 스테이킹 보상은 생태계 개발 항목에 귀속됩니다.

- **팀(Team)**

  약 270억 모나드(MON)(27%)이 Monad Foundation 및 Category Labs 소속 팀 구성원(직원·창립자·계약자 포함)에게 배정됩니다. 팀 물량은 락업과 베스팅 조건을 모두 충족해야 언락됩니다. 일반적으로 개인별 베스팅 일정은 3~4년이며, 프로젝트 기여 시작 시점에 따라 결정됩니다.

  팀 물량은 퍼블릭 메인넷 출시일로부터 1년 동안 전량 락업되며, 이후 1주년 시점부터 언락이 시작되고 다음 3년 동안 순차적으로 언락됩니다.

  언락과 베스팅이 모두 충족된 팀 물량은 1년 차 시점 기준 전체 공급의 약 10.7%입니다.

- **투자자(Investors)**

  약 197억 모나드(MON)(19.7%)이 Category Labs(과거 Monad Labs)의 초기 및 후속 투자 라운드 참여 투자자에게 배정되었습니다. 투자자 물량은 퍼블릭 메인넷 출시일 기준 4년 락업이 적용되며, 1년 클리프 후 36개월간 월 단위 균등 언락(1/48) 방식으로 해제됩니다.

- **Category Labs Treasury**

  Category Labs(구 Monad Labs)는 Monad 프로토콜의 시스템 엔지니어링·연구팀이며, 향후 인력 보상용으로 약 39.5억 모나드(MON)(3.95%)이 배정됩니다. 해당 물량은 메인넷 출시일부터 4년 락업이 적용되며, 1년 클리프 후 36개월간 월 단위 균등 언락됩니다.

토큰 공급 변화

총 공급량은 블록 보상에 따른 인플레이션과 거래 수수료 소각에 따른 디플레이션이 함께 적용됩니다.

인플레이션(Inflation)

네트워크 보안을 위해 검증자와 스테이커에게 블록당 신규 모나드(MON)이 발행됩니다. 각 블록은 25 모나드(MON)의 인플레이션 보상을 생성하며, 해당 블록을 생성한 검증자의 스테이커에게 분배됩니다. 연간 환산 시 이는 초기 총 공급량의 약 2%에 해당하는 약 20억 모나드(MON) 인플레이션 규모입니다.

디플레이션(Deflation)

트랜잭션 수수료는 기본(base) 수수료와 우선순위(priority) 수수료로 구성되며, 기본 수수료는 전량 소각됩니다.

3. 참고자료

gmonads.com

출처 : https://www.gmonads.com/

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