

아스타(Astar)

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.

아스타 네트워크(Astar Network)에 대하여

아스타 네트워크는 일본에서 가장 널리 사용되는 스마트 컨트랙트 플랫폼으로, EVM(이더리움 가상머신)과 웹어셈블리(WebAssembly, Wasm) 환경을 모두 지원하며, 크로스 가상머신(Cross-Virtual Machine)을 통해 두 환경 간 상호운용성(interoperability)을 제공한다. 아스타 네트워크는 다양한 개발자와 그들이 익숙하게 사용하는 도구 및 언어에 친화적으로 설계되었다. 폴카닷(Polkadot)의 공유 보안(shared security)을 기반으로, 아스타는 역동적이고 건강한 생태계 속에서 독자적인 존재감을 드러내며, 블록체인 산업 전반에서 글로벌 기업의 채택과 소비자들의 웹3(Web3) 기술 관심을 이끌고 있다.

개발자 인센티브의 부재

아스타의 빌드투언(Build2Earn) 시스템은 네트워크를 혁신적인 방식으로 성장시키는 동시에, 참여자와 빌더 모두에게 보상을 제공하도록 설계되었다. 개발자는 자신이 구축하고 유지하는 탈중앙화 애플리케이션(dApp)에 대해 인센티브를 받을 수 있고, 사용자는 자신이 선호하는 프로젝트를 지원함으로써 보상을 얻을 수 있다. 이러한 구조는 생태계 전반의 자발적 성장과 참여를 촉진한다.

확장성(Scalability)

블록체인은 설계상 확장성이 제한되어 있다. 이는 탈중앙화된 합의 메커니즘(consensus mechanism)이 제공하는 보안적 특성 때문이다. 블록체인이 더 빠를수록 보안성은 낮아진다. 개별 블록에 저장할 수 있는 데이터의 양에는 한계가 있으며, 네트워크가 포화 상태에 이르면 트랜잭션이 장시간 미확정(pending) 상태로 머무르는 일이 발생해 사용자 경험이 저하된다.

지금까지의 해결책은 브리지(bridge)를 통해 느슨하게 연결된 여러 독립 블록체인을 사용하는 방식이었다. 그러나 이러한 구조는 수탁형(custodial) 요소에 의존하기 때문에 공격에 취약하며, 공통된 보안 표준이나 상호운용성 메커니즘이 존재하지 않는다. 결과적으로 ‘크로스체인(cross-chain)’은 한 블록체인에서 다른 블록체인으로 스마트 컨트랙트를 단순 복제하는 수준에 머물렀다. 가치 있는 프로젝트들은 쉽게 복제되어 다른 체인에 배포되었고, 그 과정에서 진정한 혁신은 거의 이루어지지 않았다.

아스타 네트워크는 이러한 한계를 극복하고, 차세대 멀티체인 스마트 컨트랙트의 새로운 방향을 제시한다. 아스타는 EVM과 웹어셈블리 가상머신의 장점을 모두 활용할 수 있으며, 폴카닷 생태계 전반의 다양한 애플리케이션 특화 블록체인과 연동이 가능하다. 이를 통해 아스타는 미래 기술 변화에도 대응할 수 있는 확장성과 유연성을 갖춘 플랫폼으로 자리잡고 있다.

상호운용성(Interoperability)

폴카닷의 공유 보안으로 강화된 아스타의 상호운용성 프레임워크는 개발자들이 기존과는 다른 방식으로 스마트 컨트랙트를 개발하고 활용할 수 있도록 한다. 여러 블록체인과 애플리케이션 간의 협력과 통합을 촉진함으로써 복합적인 시너지를 만들어내고, 이는 다른 어떤 환경에서도 구현할 수 없는 진정한 혁신적 솔루션으로 이어진다.

현재 다양한 블록체인이 존재하지만, 상호운용성을 핵심 가치로 삼는 네트워크는 드물다.
아스타 네트워크가 바로 그 중심에 있다.

계정(Accounts)

개요(Overview)

아스타 네트워크(Astar Network)의 계정은 개인 키(private key)와 공개 키(public key)로 구성된다. 공개 키는 외부에서 누구나 접근할 수 있기 때문에, 계정 주소(address)라고도 불린다. 예를 들어 이메일 주소처럼 공개적으로 알려진 개념이다. 반면 개인 키는 해당 주소에 접근하고 관리하기 위한 열쇠이며, 오직 본인만이 이를 통해 자산을 제어할 수 있다. 즉, 누구나 당신의 주소로 토큰을 보낼 수는 있지만, 그 토큰을 사용할 수 있는 사람은 개인 키를 가진 당신뿐이다. 따라서 개인 키를 안전하게 보관하는 것은 무엇보다 중요하다.

아스타 네트워크는 웹어셈블리(WebAssembly, Wasm)와 EVM(이더리움 가상머신) 두 환경을 모두 지원하므로, 각각에 대응하는 두 가지 계정 형식을 사용한다.

서브스트레이트 계정(Substrate Accounts)

아스타는 블록체인을 구축하기 위한 프레임워크인 서브스트레이트(Substrate) 기반으로 개발되었으며, 서브스트레이트 계정 구조를 사용한다. 서브스트레이트 기반 체인은 공개 키를 바탕으로 하나 이상의 공개 주소를 파생(derive)시킨다. 공개 키를 직접 사용하는 대신, 하나의 공개 키-개인 키 쌍으로부터 여러 주소 형식과 주소를 생성할 수 있다. 즉, 한 번만 키 쌍을 생성하면, 다양한 서브스트레이트 체인에서 각각의 주소를 도출해 사용할 수 있다.

개인 키는 암호학적으로 안전한 무작위 숫자 시퀀스로 구성된다. 사람이 읽기 쉽도록, 이 개인 키는 ‘비밀 시드 구문(secret seed phrase)’ 또는 ‘니모닉(mnemonic)’이라 불리는 무작위 단어 시퀀스로 표현된다.

아스타와 같은 서브스트레이트 기반 체인에서는 **ss58** 주소 형식을 사용한다. ss58은 비트코인(Bitcoin)의 Base-58-check 인코딩을 일부 수정한 형식으로, 주소가 특정 네트워크에 속함을 구분할 수 있도록 주소 유형 접두어(prefix)를 포함한다.

아스타는 **프록시(Proxy) 계정** 과 **키리스(Keyless) 계정** 같은 특수 계정도 지원한다.

EVM 계정(EVM Accounts)

아스타의 EVM 환경은 서브스트레이트 기반 체인 위에서 **이더리움 스타일 주소(H160 형식)** 를 지원한다. 이 주소는 40자리의 16진수 문자와 ‘0x’ 접두어로 구성되어 총 42자 길이며, 하나의 개인 키와 연결된다. 이 개인 키를 사용하면 아스타 체인의 이더리움 측에서 트랜잭션 서명이 가능하다. 또한, 이 주소는 서브스트레이트의 **밸런스 팔렛(Balance pallet)** 내 저장 슬롯에 매핑되어 **서브스트레이트 스타일 주소(H256 형식)** 로 변환된다.

아스타 토큰(Astar Token)

개요(Overview)

아스타 네트워크(Astar Network)의 고유 토큰은 **아스타 토큰(ASTR)** 이다. ASTR의 초기 발행량은 **70억 개(7,000,000,000)** 였다. 현재 아스타 네트워크의 토큰노믹스(tokenomics)는 **인플레이션 모델** 을 기반으로 하며, 연간 약 **6억 6,500만 개** 의 ASTR이 새로 발행되는 것으로 추정된다. ASTR 토큰은 **소수점 18자리(18 decimals)** 까지 표현된다.

제네시스(Genesis) 시점에서 ASTR의 초기 배분은 다음과 같다.

배분 대상(Allocation)	비율(%)
사용자 및 초기 지지자 (Users and Early Supporters)	30
2021년 패러체인 경매 (Parachain Auction in 2021)	20
패러체인 경매 예비금 (Parachain Auctions Reserve)	5
프로토콜 개발 (Protocol Development)	10
온체인 DAO (On-Chain DAO)	5
마케팅 (Marketing)	5
초기 재무 후원자 (Early Financial Backers)	10
팀(직원 인센티브 포함) (Team / Employee Incentives)	5
재단 (Foundation)	10

활용도(Utility)

아스타 토큰(ASTR)은 아스타 네트워크의 핵심 유틸리티 토큰으로, 다음의 세 가지 주요 기능을 수행한다.

1. dApp 스테이킹(dApp Staking)
2. 트랜잭션(Transactions)
3. 온체인 거버넌스(On-chain Governance)

dApp 스테이킹(dApp Staking)

아스타 네트워크의 핵심 팀은 **dApp 스테이킹** 이라는 독창적인 개념을 개발했으며, 이는 폴카닷(Polkadot) 생태계 내에서 아스타에서만 제공되는 기능이다. 아스타 네트워크의 목표는 폴카닷의 **탈중앙화 애플리케이션(dApp) 허브** 로 자리 잡는 것이다. 이를 위해 아스타 네트워크에서 스마트 컨트랙트를 구축하는 dApp 개발자에게 인센티브가 제공된다.

dApp 스테이킹은 간단하지만 매우 효과적인 메커니즘이다. 아스타의 혁신적인 구조를 통해 ASTR 토큰 보유자는 자신이 지지하는 dApp에 ASTR을 스테이킹함으로써 생태계의 성장에 기여하고, 동시에 블록 보상을 통해 ASTR 토큰을 얻을 수 있다. 또한 dApp 개발자는 자신들의 애플리케이션에 스테이킹된 금액에 따라 보상을 받을 수 있다. 이와 같은 시스템은 **지속 가능성과 탈중앙화** 를 추구하는 아스타 네트워크의 철학을 반영하며, 개발자가 아스타 네트워크 커뮤니티를 위한 스마트 컨트랙트나 인프라를 구축함으로써 보상을 받을 수 있도록 한다. 결과적으로, dApp 스테이킹은 강력하면서도 분산된 형태의 보조금(grant) 메커니즘 역할을 한다.

트랜잭션(Transactions)

아스타 네트워크에서 이루어지는 모든 온체인 트랜잭션은 송신자가 수수료(fee)를 지불해야 한다. 이 수수료의 일부는 소각(burn)되고, 일부는 콜러이터(collator)에게 보상으로 지급된다.

온체인 거버넌스(On-chain Governance)

아스타 네트워크는 **탈중앙화된 의사결정의 힘** 을 중요하게 생각한다. 이러한 철학을 바탕으로, 아스타는 모든 커뮤니티 구성원이 네트워크의 미래 방향성에 직접 참여할 수 있는 **온체인 거버넌스** 를 구현하고 있다. 아스타 토큰(ASTR)은 이 거버넌스 과정의 핵심적인 수단으로, 투표(voting) 및 국민투표(referenda)와 같은 의사결정 활동을 수행하는 데 사용된다.

시덴(Shiden)과 시부야(Shibuya)

시덴과 시부야는 각각 자체 토큰을 보유하고 있다. 시덴의 토큰은 **SDN** 이며, 초기 배분 내역은 다음과 같다.

배분 대상(Allocation)	비율(%)
사용자 및 초기 지지자 (Users and Early Supporters)	30
패러체인 경매 (Parachain Auctions)	30
패러체인 경매 예비금 (Parachain Auctions Reserve)	5
프로토콜 개발 (Protocol Development)	10
온체인 DAO (On-Chain DAO)	5
마케팅 (Marketing)	5
생태계 성장 (Ecosystem Growth)	10
팀 (직원 인센티브 포함) (Team / Employee Incentives)	5

시부야의 토큰은 **SBY** 이다. 테스트 목적의 SBY는 **Astar Portal의 Faucet** 기능을 통해 획득할 수 있다.

dApp 스테이킹 개요(dApp Staking Overview)

아스타는 모든 사용자가 자신의 데이터와 자산을 완전히 통제할 수 있는, 사람 중심의 자유로운 인터넷을 실현하고자 한다. 이러한 사용자 주도형 인터넷에 대한 비전은 우리가 추구하는 웹3(Web3)의 핵심이자 아스타의 존재 이유이다.

dApp 스테이킹은 이러한 웹3 비전을 실현하기 위한 핵심 동력이다. 이는 사용자에게 뛰어난 dApp을 제공하기 위한 개발 기반을 마련하고, 누구나 웹3 환경에 쉽게 접근할 수 있도록 하는 촉매 역할을 한다.

dApp 스테이킹이란(What is dApp Staking)

dApp 스테이킹 은 탈중앙화 애플리케이션(dApp)을 개발하는 개발자에게 경제적 인센티브를 제공하기 위한 아스타만의 독창적인 메커니즘이다. 개발자는 dApp 스테이킹에 참여함으로써 기본적인 수익을 얻을 수 있으며, 이를 통해 지속적으로 dApp을 개발하고 개선할 수 있다. 또한 dApp 스테이킹은 개발자, 스테이커(staker), 그리고 dApp 사용자 간의 유대 관계를 강화하고, 생태계 참여자 모두가 보상을 받을 수 있는 구조를 만든다.

아스타 및 시덴 네트워크의 dApp 스테이킹은 검증자 validator 스테이킹과 유사한 방식으로 작동한다. 하지만 새로운 패러다임에서는 **dApp 스테이커** 로 불리는 개인이 자신이 지지하는 특정 dApp에 아스타(ASTR) 또는 시덴(SDN) 토큰을 직접 스테이킹할 수 있다.

네트워크의 매 블록마다 일정 비율의 인플레이션 물량이 dApp 스테이킹에 배정되며, 해당 보상은 dApp 운영자(개발자)와 스테이커에게 분배된다.

이 메커니즘은 단순하다. 개발자는 뛰어난 프로젝트를 구축하고 지지자를 유치함으로써 보상을 받고, 스테이커는 자신이 선호하는 dApp을 지원함으로써 수익을 얻는다.

아스타 및 시덴 네트워크에서의 dApp 스테이킹 혜택(Benefits of dApp Staking on Astar and Shiden Network)

웹3 생태계가 성장하기 위해서는 dApp 스테이커, 개발자, 사용자 간의 **상호 의존적 관계** 가 필수적이다. 아스타와 시덴에서의 dApp 스테이킹은 이러한 **상호 이익 구조** 를 가능하게 하는 핵심 메커니즘이다.

• 개발자 보상(Developer Compensation):

기존에는 dApp 개발자가 수익을 얻기 위해 보조금(grant) 프로그램이나 토큰 발행, 자금 조달에 의존해야 했다. 그러나 아스타시덴 네트워크의 dApp 스테이킹은 개발자에게 **기본 수익원(basic income)** 을 제공한다. dApp이 커뮤니티의 지지를 받아 노미네이트(nominate)되면, 개발자는 지속적인 보상을 받을 수 있다.

- 인기 상승에 따른 보상 증가(Increased Popularity, Increased Reward): dApp이 인기를 얻고 커뮤니티로부터 더 많은 노미네이터를 확보할수록, 해당 dApp의 개발자는 더 큰 비율의 블록 보상을 받을 수 있다. 이는 개발자가 사용자와 커뮤니티에 실질적인 가치를 제공하는 **고품질 dApp**을 개발하도록 유도한다.
- 가스비 의존도 감소(Reduced Dependency on Gas Fees): 다른 블록체인에서는 트랜잭션이나 상호작용 시 높은 가스비를 지불해야 하지만, 아스타시덴 네트워크에서는 dApp 스테이킹을 통해 개발자가 보상을 받을 수 있어 가스비에 대한 의존도를 줄이고, 보다 경제적인 dApp 개발 환경을 조성할 수 있다.

스테이커의 혜택: 토큰 가치 상승 및 보상 획득(Benefits for Stakers: Increase in Token Value & Earn Tokens)

아스타(Astar) 또는 시덴(Shiden) 네트워크의 dApp 스테이커는 자신이 보유한 토큰의 가치가 상승하길 원한다. 그러나 토큰의 시장 가치가 높아지면, 그 토큰이 기반하고 있는 **기초 자산의 내재 가치**가 함께 상승해야 한다.

이러한 가치 상승은 네트워크가 성장하거나 유통 중인 토큰의 수가 감소할 때 발생한다. 따라서 스테이커는 자신이 보유한 토큰을 dApp에 스테이킹함으로써 장기적으로 토큰의 가치를 높이고, 동시에 높은 연이율(APR) 형태의 보상을 얻을 수 있다.

이러한 시장 메커니즘은 스테이커가 장기적으로 더 큰 자산 가치를 확보함과 동시에, 스테이킹을 통해 실질적인 수익을 얻을 수 있도록 한다.

빌더의 혜택: 보상 획득(Benefits for Builders: Earn Rewards)

훌륭한 dApp이 탄생하기 위해서는 우수한 개발자가 필요하며, 개발자가 뛰어난 dApp을 만들기 위해서는 **경제적 인센티브**가 필요하다.

웹3(Web3) 생태계에서 가장 중요한 인적 자원은 바로 개발자이다. dApp 스테이킹이 도입되기 전까지, 개발자가 퍼블릭 블록체인에서 dApp이나 인프라를 구축하더라도 이에 대한 **지속 가능한 보상 체계**는 거의 존재하지 않았다.

dApp 스테이킹을 통해 개발자는 아스타 또는 시덴 네트워크에서 dApp을 개발하면서 **기본 수익(basic income)**을 얻을 수 있다. 이러한 보상 구조는 더 많은 개발자가 생태계에 참여하고, 자신의 dApp을 지속적으로 개선하도록 유도한다. 개발자가 많아질수록 생태계 전체의 성장 잠재력은 커진다.

따라서 아스타 네트워크는 개발자가 **보상과 개발을 동시에 수행할 수 있는 환경**을 제공하는 것을 핵심 가치로 삼고 있으며, 불필요한 부담 없이 개발에 집중할 수 있도록 지원한다.

 **참고:**
dApp 빌더로 참여하는 방법은 [이 섹션](#)을 참고하세요.

dApp 사용자 혜택: dApp 및 네트워크 활용성 강화(Benefits for dApp Users: Increase in dApp & Network Utility)

dApp 스테이킹은 생태계의 품질을 향상시키는 핵심 요소로, dApp의 성장과 발전을 촉진한다. 이를 통해 개발자는 고품질 dApp을 구축하면서 동시에 수익을 창출할 수 있는 기회를 얻게 된다.

고품질 dApp이 늘어날수록 사용자 유입과 유지율이 높아지고, 이는 **네트워크 효과(network effect)**를 일으킨다. 이러한 선순환 구조는 생태계 확장을 가속화하고, 장기적으로 네트워크의 내재 가치를 높인다.

요약하자면, dApp 스테이킹의 중요성은 **지속적인 품질 향상과 생태계 활용성 강화**에 있다. 고품질 dApp의 증가와 사용자 기반의 확장은 상호작용을 통해 네트워크 효과를 증폭시키며, 결과적으로 생태계의 전반적인 가치를 장기적으로 끌어올린다.

dApp 스테이킹의 용어와 파라미터 이해하기(Understand the Terminology and Parameters of dApp Staking)

dApp 스테이킹 v3의 구조를 이해하려면, 다음의 기본 개념을 숙지하는 것이 중요하다.

에라(Era)

에라(Era)는 dApp 스테이킹의 기본 시간 단위이다. 하나의 에라는 **7,200 블록**으로 구성되며, 이는 약 **1일**에 해당한다.

기간(Period) 및 하위 기간(Subperiods)

기간(Period)은 에라보다 더 긴 시간 단위이다.

쉽게 말해, dApp 스테이킹에서의 기간(Period)은 여러 날(즉, 여러 에라)로 이루어진 ‘한 달’ 개념과 유사하다.

각 기간(Period)은 두 개의 하위 기간(Subperiod)으로 구성된다.

- 투표(Voting)
- 빌드 앤 연(Build & Earn)

앞선 비유를 이어가면, 한 달을 두 부분(길이가 다를 수 있음)으로 나눈 것과 같다. 첫 번째 부분은 투표 단계, 두 번째 부분은 빌드 앤 연(Build & Earn) 단계에 해당한다.

투표(Voting, Staking) 하위 기간(Subperiod)

이 하위 기간 동안 스테이커(staker)는 자신이 보유한 토큰을 어떤 dApp에 스테이킹할지를 결정할 수 있다. 이 시기에는 dApp 개발자와 팀이 자신의 제품을 홍보하고, 캠페인을 진행하며, 스테이커를 유치하는 데 집중한다.

투표 하위 기간 동안에는 **스테이킹 보상** 이 발생하지 않는다. 하지만 이 시기에 dApp에 투표 및 스테이킹한 사용자는 이후 **보너스 리워드(Bonus Reward)** 를 받을 자격이 주어진다.

⚠️ 중요:

dApp 개발자 및 팀은 **투표 하위 기간** 전후로 마케팅과 캠페인을 철저히 준비해야 하며, 가능한 한 많은 스테이커와 토큰을 확보하는 것이 매우 중요하다.

빌드 앤 언(Build & Earn) 하위 기간(Subperiod)

이 하위 기간에는 스테이커와 dApp이 실제로 보상을 받기 시작한다. 사용자는 빌드 앤 언 기간 중에도 토큰을 스테이킹하여 추가 보상을 얻을 수 있지만, 이 시기에 스테이킹된 금액은 **보너스 리워드 계산에는 포함되지 않는다**.

빌드 앤 언(Build & Earn) 하위 기간이 종료되면, 해당 기간(Period)은 끝나고 새로운 기간이 시작된다. 이때 모든 토큰은 dApp에서 **언스테이킹(unstaked)** 되지만, 여전히 **잠금(locked)** 상태를 유지한다. 이후 새로운 **투표 하위 기간(Voting Subperiod)** 이 시작된다.

참여 전 고려해야 할 주요 파라미터(Parameters)

사용자나 dApp 소유자는 dApp 스테이킹을 이용하기 전에 다음의 파라미터를 이해해야 한다.

파라미터(Parameters)	아스타 네트워크 (Astar Network)	시덴 네트워크 (Shiden Network)	시부야 (Shibuya)
기간당 에라 수 (Eras Per Period)	122 (~122일)	61 (~61일)	28 (~7일)
투표 하위 기간의 에라 수 (Eras Per Voting Subperiod)	11 (~11일)	6 (~6일)	8 (~48시간)
빌드 앤 언 하위 기간의 에라 수 (Eras Per Build&Earn Subperiod)	111 (~111일)	55 (~55일)	20 (~120시간)
에라당 블록 수 (Blocks Per Era)	7,200 (~24시간)	7,200 (~24시간)	1,800 (~6시간)
언락 기간 (Unlocking Period)	9 에라 (~9일)	4 에라 (~4일)	4 에라 (~1일)
최소 스테이킹 수량 (Minimum Amount to Stake)	500 ASTR	50 SDN	5 SBY

전체 파라미터 목록은 dApp 스테이킹 파라미터 페이지와 Tokenomics 2.0 페이지를 참고하면 된다.

티어 시스템(Tier System)

티어 시스템은 각 dApp에 스테이킹된 총 가치(Total Value Staked, TVS)를 기준으로 dApp의 순위를 결정하는 방식이다. 이 티어 시스템은 각 **빌드 앤 언(Build&Earn) 하위 기간(Subperiod)** 이 종료될 때마다 갱신된다.

티어는 한정된 개수만 존재하며, 각 티어는 정해진 **슬롯 수(slot)** 와 dApp이 해당 티어에 진입하기 위한 **최소 기준치(threshold)** 를 가진다. dApp은 충분한 지지를 받아야 티어에 진입할 수 있으며, 공정한 경쟁을 위해 이 기준치는 **동적으로 조정(dynamically adjusted)** 된다. 상위 티어로 올라갈수록 더 높은 보상이 주어진다.

사용자는 **Astar Portal**의 **Projects Leaderboard** 에서 각 dApp이 어느 티어에 속하는지 언제든지 확인할 수 있다.

다음 단계(What should I do next)

dApp 스테이킹의 개념은 단순하다.

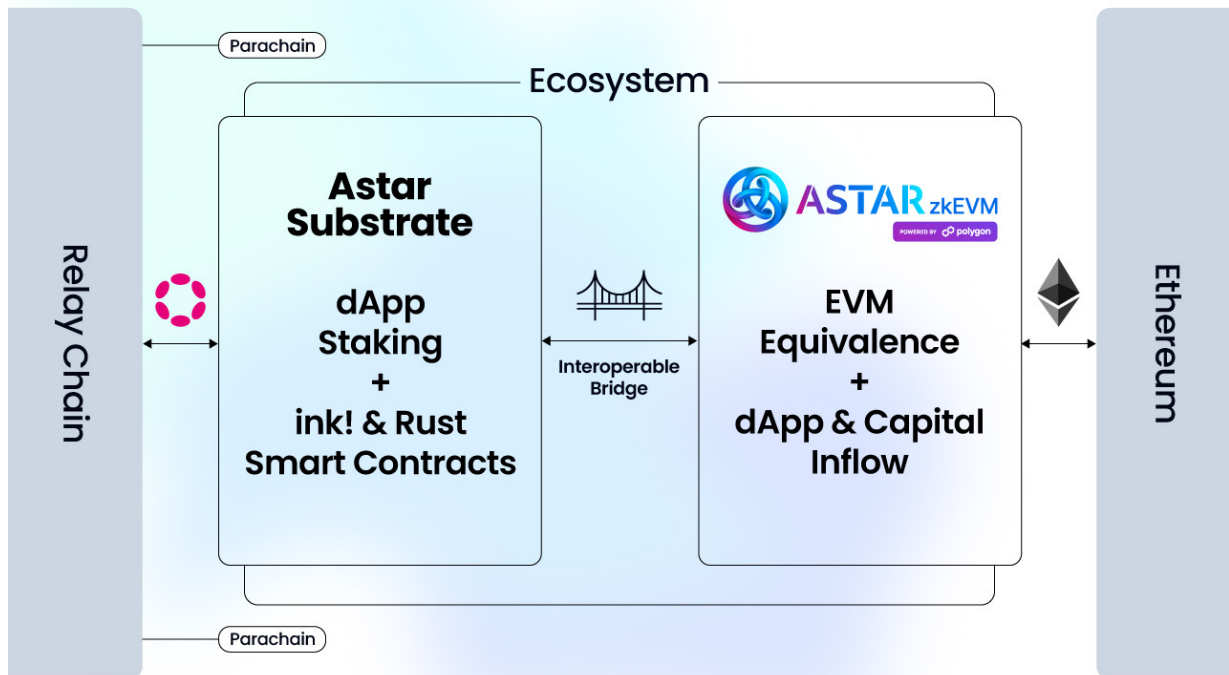
사용자는 생태계 내 dApp에 스테이킹함으로써 이를 지원하고, dApp이 충분한 지지를 얻으면 인플레이션(inflation)을 통해 생성된 네이티브 토큰으로 보상을 받는다. 그 결과, 스테이커는 자신의 기여와 스테이킹 활동에 대한 보상을 얻게 된다.

아키텍처(Architecture)

개요(Overview)

아스타 네트워크(Astar Network)를 이해하고 이를 활용하기 위해서는 먼저 폴카닷(Polkadot) 과 이더리움(Ethereum) 에 대한 기본적인 이해가 필요하다. 아스타 네트워크는 두 블록체인을 모두 연결하며, 각각은 고유한 장점과 한계를 가진다.

Astar Network Architecture



폴카닷 패러체인(parachain) 측의 주요 특징으로는 dApp 스테이킹(dApp Staking), ink! 스마트 컨트랙트, 크로스체인 메시징(XCM), 포크 없는 업그레이드(forkless upgradability) 등이 있다. 반면, 이더리움 레이어 2 측은 최고 수준의 EVM 호환성, 레이어 2 네트워크 간의 유동성 공유(shared liquidity), 그리고 블록체인 산업 내 최대 규모의 개발자 풀에 대한 접근성을 갖추고 있으며, 이를 통해 아스타는 다양한 수요에 맞춰 제품을 신속하게 개발할 수 있다.

지금부터는 독립적인 레이어 1 블록체인이든, 혹은 폴카닷 릴레이체인(Relay Chain)과 같은 다른 블록체인에 의존하여 보안과 상호운용성을 공유하는 형태이든, 모든 블록체인에 공통으로 적용되는 기본 원리를 살펴보자.

블록체인 기초(Blockchain Basics)

블록체인은 정보를 일련의 블록으로 기록하는 **탈중앙화 원장(decentralized ledger)** 이다. 각 블록에는 상태 변화를 일으킬 수도 있고 그렇지 않을 수도 있는 **순차적인 명령 집합** 이 포함되어 있다.

블록체인 네트워크에서는 **노드(node)** 라 불리는 개별 컴퓨터들이 서로 통신하며, **탈중앙화된 P2P(peer-to-peer) 네트워크** 를 형성한다. 이 네트워크에는 중앙 통제 기관이 존재하지 않으며, 일반적으로 블록 생성에 참여하는 각 노드는 **정식 체인(canonical chain)** 을 구성하는 블록의 사본을 저장한다.

대부분의 경우, 사용자는 블록체인에 상태 변화를 요청함으로써 상호작용한다. 예를 들어 파일 소유자를 변경하거나, 한 계정에서 다른 계정으로 자금을 이체하는 요청이 이에 해당한다. 이러한 트랜잭션 요청은 네트워크의 다른 노드로 전파(gossip)되고, 블록 생성자(block author)에 의해 하나의 블록으로 조립된다. 데이터의 안전성과 체인의 지속적인 진전을 보장하기 위해, 노드들은 **합의(consensus)** 메커니즘을 통해 각 블록의 데이터 상태와 트랜잭션 실행 순서에 대해 일치된 의견을 형성한다.

[더 알아보기...](#)

블록체인에서 실행되는 애플리케이션(Applications Running on a Blockchain)

블록체인에서 실행되는 애플리케이션은 일반적으로 **탈중앙화 애플리케이션(dApp)** 이라 불리며, 프론트엔드 프레임워크로 작성된 웹 애플리케이션이지만, 백엔드에서는 **스마트 컨트랙트(smart contract)** 가 작동하여 블록체인의 상태를 변경한다.

스마트 컨트랙트(smart contract)는 블록체인 위에서 실행되는 프로그램으로, 특정 조건이 충족되면 사용자를 대신하여 트랜잭션을 수행한다. 개발자는 스마트 컨트랙트를 통해 트랜잭션의 결과가 프로그램적으로 기록되고 변조되지 않도록 보장할 수 있다. 다만, 스마트 컨트랙트는 **샌드박스 환경(sandboxed environment)** 내에서 실행되므로, 합의(consensus), 스토리지(storage), 트랜잭션 계층(transaction layer) 등 블록체인의 기본 기능에 직접 접근할 수 없다. 대신, 해당 체인의 고정된 규칙과 제약을 따라야 한다. 이러한 제약은 개발자가 핵심 설계 결정을 직접 내릴 필요를 줄여주며, 결과적으로 **개발 주기(development lifecycle)**를 단축시키는 트레이드오프로 받아들여진다.

이더리움 가상머신(Ethereum Virtual Machine, EVM)

이더리움 가상머신(EVM)은 네트워크 참가자(반드시 이더리움에 한정되지 않음)가 데이터를 저장하고, 그 상태(state)에 대해 합의할 수 있도록 하는 **가상 컴퓨터(Virtual Computer)**이다. 스마트 컨트랙트는 이 EVM 위에서 실행되며, 대부분의 레이어 1 블록체인은 **EVM 호환 가상머신 환경(EVM-compatible environment)**을 지원한다. 아스타 패라체인(Astar Parachain) 역시 EVM 호환 환경을 완벽히 지원한다.

프록시 계정(Proxy Accounts)

개요(Overview)

Pallet Proxy는 한 계정이 다른 계정에 특정 권한을 위임할 수 있도록 하여, 해당 계정이 본인 대신 트랜잭션을 실행할 수 있게 하는 기능이다. 이를 통해 사용자는 자신의 주요 계정의 보안을 유지하면서도 다양한 기능을 안전하게 대리 수행할 수 있다.

이 페이지에서는 프록시 팔렛(proxy pallet)에 포함된 **프록시 유형(Proxy Types)**, **엑스트린식(Extrinsics, 함수 호출)**, **일반 정의, 기능 설명** 등을 자세히 다룬다. 또한, 프록시 기능을 이해하고 실제로 활용할 수 있도록 돕는 튜토리얼도 제공된다.

프록시 유형(Proxy Types)

- Any**
이 프록시 유형은 프록시 팔렛에서 지원하는 **모든 호출(call)**을 대리 계정(delegate account)이 실행할 수 있도록 허용한다. 이는 가장 높은 수준의 권한을 가지므로 **신중하게 사용해야 한다**.
- NonTransfer**
이 프록시 유형은 자산 전송(asset transfer) 기능을 제외한 대부분의 호출을 대리 계정이 실행할 수 있도록 허용한다. 예를 들어, 사용자가 자산을 콜드 월렛(cold storage)에 안전하게 보관하면서 다른 모든 기능을 프록시 계정에 위임하고자 할 때 유용하다. 현재 이 유형은 **System, Identity, Multisig, Proxy, DappStaking, CollatorSelection, XcAssetConfig, Session 트랜잭션(set_keys)**, 그리고 **Vesting** 팔렛의 두 가지 트랜잭션(vest, vest_other)을 포함한다.
- Balances**
이 프록시 유형은 대리 계정이 **계정 및 잔액 관련 기능**을 제어할 수 있도록 한다. 전체 호출 목록은 [여기](#)에서 확인할 수 있다.
- Assets**
이 프록시 유형은 대리 계정이 **대체 가능한 자산(fungible assets)**과 관련된 기능(예: 발행, 전송, 민팅 등)을 제어할 수 있도록 한다. 단, 이 프록시 유형은 아스타(ASTR)나 시덴(SDN)과 같은 **네이티브 통화(native currency)**는 제어하지 않으며, 네이티브 자산은 **Balances** 프록시를 통해 관리된다.
- IdentityJudgement**
이 프록시 유형은 **이름 시스템(naming system)** 내에서 특정 아이덴티티에 대해 심사(judgement)를 제공하는 기능만 위임한다. 위임자는 반드시 **등록자(registrar)**여야 한다.
- CancelProxy**
이 프록시 유형은 대리 계정이 **미리 공지된(proxy-announced)** 프록시 호출을 거부하거나 제거할 수 있도록 한다.
- DappsStaking**
이 프록시 유형은 대리 계정이 **dApp 스테이킹 관련 트랜잭션**(예: 계약 등록/해지, 보상 청구 등)을 수행할 수 있도록 허용한다.
- Governance**
이 프록시 유형은 대리 계정이 **거버넌스 관련 트랜잭션**(예: 제안 투표, 프리이미지(preimage) 생성 및 예치, 제안 또는 재무 요청 등)을 수행할 수 있도록 한다.

엑스트린식(Extrinsics)

프록시 팔렛은 다음과 같은 주요 엑스트린식(함수)을 제공한다.

- addProxy(delegate, proxyType, delay)**
송신자가 본인을 대신해 호출을 수행할 수 있는 **프록시 계정(delegate)**을 등록한다. delay 값이 0보다 큰 경우, 해당 프록시 계정은 트랜잭션을 사전에 공지(announce)하고 지정된 블록 수만큼 대기한 후 실행해야 한다. 실행 시 **ProxyAdded 이벤트**가 발생한다.
- announce(real, callHash)**
딜레이가 설정된 프록시 트랜잭션이 사전에 공지되었음을 기록한다. 실행 시 **Announced 이벤트**가 발생한다.
- anonymous(proxyType, delay, index)**
개인 키에 접근할 수 없는 **익명 계정(anonymous account)**을 생성한다. 송신자는 지정된 프록시 유형과 지연 시간(delay)에 따라 이 계정의 프록시 역할을 수행하게 된다.
⚠ 주의: 프록시가 제거되면, 해당 기본 계정에 접근할 수 없게 된다. 실행 시 **AnonymousCreated 이벤트**가 발생한다.
- killAnonymous(spawner, proxyType, index, height, extIndex)**
이전에 생성된 익명 프록시를 제거한다.
- proxy(real, forceProxyType, call)**
프록시로서 트랜잭션을 수행한다. 실행 시 **ProxyExecuted 이벤트**가 발생한다.
- proxyAnnounced(delegate, real, forceProxyType, call)**

프록시로서 트랜잭션을 실행하고, 관련된 이전 공지(announcement)를 제거한다. 실행 시 **ProxyExecuted 이벤트**가 발생한다.

- **rejectAnnouncement(delegate, callHash)**
송신자가 주 계정(prime account)인 경우, 프록시 계정에서 특정 공지를 제거한다.
- **removeAnnouncement(real, callHash)**
송신자가 프록시 계정(proxy account)인 경우, 주 계정으로 전송된 특정 공지를 제거한다.
- **removeProxies()**
송신자와 연결된 모든 프록시 계정 등록을 해제 한다.
- **removeProxy(delegate, proxyType, delay)**
송신자와 연결된 특정 프록시 계정을 해제한다. 실행 시 **ProxyRemoved 이벤트**가 발생한다.

프록시 팔렛(Pallet Proxy)의 상수(Constants for Pallet Proxy)

상수(Constant)	설명(Description)
announcementDepositBase	공지를 생성할 때 예치해야 하는 기본 통화 금액.
announcementDepositFactor	공지 1건당 추가로 필요한 통화 금액.
maxPending	대기 상태로 허용되는 시간 지연 공지의 최대 개수.
maxProxies	단일 계정에 허용되는 최대 프록시 수.
proxyDepositBase	프록시를 생성할 때 예치해야 하는 기본 통화 금액.
proxyDepositFactor	프록시 1개 추가 시 필요한 통화 금액.

프록시 예치금이 필요한 이유(Why Proxy Deposits?)

네트워크 내에서 프록시를 생성하려면 **네이티브 통화(예: ASTR 또는 SDN)**로 예치금을 제공해야 한다. 이는 프록시를 추가할 때 **온체인 스토리지(on-chain storage)** 공간이 필요하기 때문이다. 모든 노드는 이 데이터를 네트워크 전체에 복제해야 하므로, 상당한 비용이 발생한다. 이러한 이유로 프록시 생성 과정은 **서비스 거부(DoS) 공격**의 잠재적 경로가 될 수 있으며, 이를 방지하기 위해 예치금이 요구된다.

프록시를 생성하면, 예치금이 사용자의 **가용 잔액(free balance)**에서 차감되어 **예치 잔액(reserved balance)**으로 이동한다. 이 메커니즘은 프록시 시스템의 **무결성과 안정성**을 유지하며, 프록시가 비활성화되거나 제거될 때 예치금이 반환될 수 있도록 보장한다.

프록시 개수가 n개일 때 필요한 총 예치금은 다음과 같다:

$\text{ProxyDepositBase} + \text{ProxyDepositFactor} * n$

순수/익명 프록시(Pure / Anonymous Proxy)

익명 프록시(Anonymous Proxy) 또는 **순수 프록시(Pure Proxy)**는 기존 계정으로부터 파생된, 완전히 새로운 계정이다. 기본 계정(primary account)이 프록시 역할을 맡아 익명 프록시를 대신하여 트랜잭션을 수행한다.

이러한 프록시는 **비결정적(non-deterministic)**이며, **개인 키(private key)**가 존재하지 않으므로 직접 제어할 수 없다. 하지만 **임의로 생성된 주소(randomly generated address)**를 가지며, 소유자가 없기 때문에 어느 누구도 해당 계정을 완전히 지배할 수 없다.

익명 프록시는 사용이 복잡하고 주의가 필요하지만, **보안 강화 및 멀티시그(multisig) 계정 관리** 등 다양한 장점을 제공한다.

💡 **참고:**

익명(순수) 프록시에 대한 더 자세한 정보와 제한 사항, 이점은 [Polkadot 공식 문서를](#) 참고하세요.

📺 **영상 자료:** [YouTube 링크](#)

시간 지연 프록시(Time-Delayed Proxy)

프록시의 보안을 강화하기 위해 **블록 단위로 측정되는 지연(delay) 메커니즘**을 적용할 수 있다. 예를 들어, 딜레이 값이 10이라면 **10블록(약 2분)** 동안 실행이 지연된다. 프록시는 proxy.announce 엑스트린식을 통해 수행할 작업을 미리 선언하고, 정의된 블록 수만큼 대기한 후 실행한다.

이 지연 기간 동안, 프록시가 수행하려는 작업은 다음과 같은 방식으로 취소할 수 있다.

- 프록시 자체가 proxy.removeAnnouncement를 호출하여 취소
- 프록시를 제어하는 주 계정이 proxy.rejectAnnouncement를 호출하여 취소

프록시가 공지한 트랜잭션은 **해시(call hash)**로 식별되므로, 특정 작업의 진위를 검증하고 관리할 수 있다.

이 지연 메커니즘을 통해 악의적인 행위가 발생하더라도 지정된 기간 내에 탐지 및 취소할 수 있다. 지연 시간이 지난 후에는, 프록시가 proxy.proxyAnnounced 엑스트린식을 사용해 공지된 트랜잭션을 실행할 수 있다.

⚠ 주의:

순수 프록시(pure proxy)를 사용하여 proxy 엑스트린식으로 서명하려는 시도는 실패한다. 자세한 내용은 [이 문서를](#) 참고하세요.

이 구현은 프록시의 신뢰성과 안전성을 한층 강화하며, 지정된 지연 기간 동안 모든 행동이 관찰 및 취소될 수 있다는 점에서 보다 **안심할 수 있는 사용 환경**을 제공한다.

💡 참고:

시간 지연 프록시 설정 방법은 [이 영상](#)을 참고하세요.

스마트 컨트랙트(Smart Contracts)

개요(Overview)

아스타(Astar)는 **스마트 컨트랙트 배포 플랫폼**으로, 개발자는 다양한 프로그래밍 언어를 사용해 아스타 네트워크(Astar Network)에 스마트 컨트랙트를 작성하고 배포할 수 있다. 이더리움(Ethereum) 스마트 컨트랙트와 호환되는 **솔리디티(Solidity)**, 또는 폴카닷(Polkadot) 생태계를 위한 러스트(Rust) 기반의 **ink!** 언어를 비롯해 여러 언어가 지원된다. 이러한 호환성 덕분에 다른 블록체인 생태계의 개발자들이 아스타로 원활히 이전할 수 있으며, 이는 **상호운용성(interoperability)**을 촉진하고 아스타 네트워크의 채택을 확산시키는 데 기여한다.

아스타 네트워크는 **고성능 합의 메커니즘(consensus mechanism)**을 채택하여 빠르고 효율적인 컨트랙트 실행을 보장한다. 특히 **지분증명(PoS)**과 **작업증명(PoW)**을 결합한 독창적인 하이브리드 구조를 통해 보안을 강화하면서도 확장성을 유지한다. 이러한 아키텍처 덕분에 아스타 네트워크는 **대규모 스마트 컨트랙트 트랜잭션 처리**가 가능하며, 높은 연산 성능을 요구하는 탈중앙화 애플리케이션(dApp)에 적합하다.

또한 아스타 네트워크는 스마트 컨트랙트의 **개발, 테스트, 배포 전 과정**을 지원하기 위한 포괄적인 도구 세트를 제공한다. 여기에는 강력한 **소프트웨어 개발 키트(SDK)**, 세밀한 **개발자 문서화(Documentation)**, 직관적인 **통합 개발 환경(IDE)**이 포함된다. 이러한 리소스들은 개발자가 보안성과 효율성을 갖춘 스마트 컨트랙트를 작성하도록 돕고, 잠재적 취약점을 줄이며 탈중앙화 애플리케이션의 안정성을 보장한다.

웹어셈블리(WebAssembly, Wasm) 스마트 컨트랙트

아스타 및 시덴(Shiden)의 런타임은 **서브스트레이트(Substrate)**를 기반으로 하며, 두 네트워크 모두 pallet-contracts 모듈을 포함하고 있다. 이 모듈은 **샌드박스 환경(sandboxed environment)**에서 웹어셈블리(WebAssembly, Wasm) 스마트 컨트랙트를 배포하고 실행하는 역할을 한다.

Wasm으로 컴파일 가능한 언어라면 모두 이 Wasm 가상머신(Wasm Virtual Machine)에서 실행될 수 있다. 단, 해당 코드가 pallet-contracts의 [API](#)에 호환되어야 한다.

불필요한 복잡성과 반복 코드를 줄이기 위해, pallet-contracts를 직접 대상으로 하는 **도메인 특화 언어(eDSL)**를 사용하는 것이 권장된다. 예를 들어, 러스트 기반의 [ink!], 어셈블리스크립트(AssemblyScript) 기반의 [ask!], 그리고 생태계 확장에 따라 추가될 수 있는 다른 언어들이 있다.

컨트랙트가 컴파일된 후에는 **Wasm 블롭(blob)**으로 변환되어 **온체인(On-chain)**에 저장 및 배포된다.

이더리움 가상머신(Ethereum Virtual Machine, EVM) 스마트 컨트랙트

아스타의 **EVM 구현(Astar EVM implementation)**은 서브스트레이트 팔렛-EVM(Substrate Pallet-EVM)을 기반으로 하며, 러스트(Rust)로 완전히 구현된 **EVM 환경**을 제공한다.

아스타 EVM 상의 스마트 컨트랙트는 **솔리디티(Solidity)**, **바이퍼(Vyper)** 등 EVM 호환 바이트코드로 컴파일 가능한 언어를 사용해 개발할 수 있다. **Pallet-EVM**의 목표는 기존 이더리움 개발 도구 체인(toolchain)과 호환되면서도 낮은 진입 장벽과 높은 보안성을 갖춘 스마트 컨트랙트 개발, 테스트, 실행 환경을 제공하는 것이다.

토크노믹스 2.0(Tokenomics 2.0)

아스타 네트워크(Astar Network)의 토크노믹스(Tokenomics)는 사용자, 개발자, 그리고 기업 파트너 등 다양한 참여자의 요구를 충족시키기 위해 새롭게 개편되었다. **토크노믹스 2.0**은 지속 가능한 토큰 분배 구조와 네트워크 수수료 체계를 도입하고, 인플레이션을 완화된 모델이다.

i 정보(Info)

토크노믹스 2.0은 단계별 접근 방식(phased approach)을 통해 순차적으로 도입되었다.

- **Phase 0 – 포럼 논의 단계(Forum Discussion):**
 -  모든 단계에서 지속 진행
- **Phase 1 – 수수료 조정(Fee Adjustments):**

아스타의 트랜잭션 수수료는 **네이티브(Substrate)** 수수료와 **EVM 수수료**로 구성된다. 두 시스템은 수수료 계산 방식이 다르지만, 토크노믹스 2.0에서는 동일한 리소스를 사용하는 트랜잭션이 **거래 유형(Native/EVM)에 상관없이 유사한 비용을 지불**하도록 수수료 체계를 통합하였다.

이 섹션에서는 **토크노믹스 2.0의 수수료 모델 계산 구조**를 자세히 설명한다.

네이티브 수수료(Native Fees)

네이티브 수수료는 서브스트레이트(Substrate)에 네이티브한 트랜잭션에 적용된다. 예를 들어 잔액 전송, dApp 스테이킹 사용, 멀티시그 생성, 국민 투표(referendum) 투표 등이 이에 해당한다.

수수료는 폴카닷(Polkadot)과 (아마도) 모든 패러체인에서 일반적으로 사용되는 **모델**을 사용해 계산된다:

$$native_fee = base_fee + c * weight_fee + length_fee + rent_fee + tip$$

여기서 다음이 적용된다:

$$weight_fee = weight_factor * \frac{transaction_weight}{base_weight}$$

$$length_fee = length_factor * transaction_length$$

$$rent_fee = storage_items * price_per_item + storage_bytes * price_per_byte$$

- **base_weight** - 블록에 포함되는 각 트랜잭션의 고정 기본 가중치.
- **base_fee** - 블록에 포함되는 모든 트랜잭션이 지불해야 하는 고정 수수료.
- **weight_fee** - 트랜잭션의 가중치에 관련된 수수료.
- **c** - 수수료 승수(fee multiplier); 네트워크 활용도가 이상치(ideal)보다 높으면 c가 증가해 더 많은 수수료를 지불하게 된다. 반대로 혼잡도가 낮으면 승수가 감소한다.
- **length_fee** - 트랜잭션 길이(바이트 수)에 관련된 수수료.
- **rent_fee** - 온체인에 데이터를 저장하기 위한 예치(렌트) 수수료. Wasm 트랜잭션의 렌트 수수료 계산에 대한 자세한 설명은 빌드 섹션에서 확인할 수 있다.
- **tip** - 트랜잭션이 더 빨리 블록에 포함되도록 제출자가 추가로 지불하는 금액.

네이티브 수수료는 본질적으로 동적이며, 다음 공식을 사용해 각 블록마다 계산되는 수수료 승수 c를 사용한다:

$$c_n = c_{n-1} * (1 + adjustment + \frac{adjustment^2}{2})$$

$$adjustment = v * (s - s^*)$$

$$s = \frac{block_weight}{max_block_normal_dispatch_weight}$$

다음은 여러 구성 파라미터이다:

- **s*** - 이상적인 블록 포화도(ideal block fullness); 장기적으로 원하는 평균 블록 포화도.
- **v** - 변동성 계수(variability factor); 조정 계수가 변하는 속도를 제어한다. 값이 작으면 천천히, 크면 빠르게 조정된다.
- **block_weight** - 이전 블록의 총 가중치.
- **c_{min}** - 수수료 승수 c의 최소값.
- **c_{max}** - 수수료 승수 c의 최대값.

그리고 s를 사용해 현재 블록 포화도를 설명하면:

- **s > s***이면, 블록 포화도가 이상치보다 **높다**는 의미이며, 조정 값은 양수가 된다.
- **s < s***이면, 블록 포화도가 이상치보다 **낮다**는 의미이며, 조정 값은 음수가 된다.

네트워크 사용량(혼잡도)에 따라 계수 c는 블록마다 증가하거나 감소한다. 네트워크가 크게 사용되고 블록이 가득 차면 c가 증가하여 가중치 수수료가 확대되고 트랜잭션 비용이 높아진다. 반대로 네트워크 혼잡도가 이상 수준보다 낮으면 수수료 승수가 감소하여 트랜잭션 비용이 낮아진다.

EVM 수수료(EVM Fees)

아스타(Astar)는 **이더리움(Ethereum)과 완전히 호환되는 네트워크**이다. 즉, 이더리움의 **가스(Gas) 개념**을 지원한다. 가스는 네이티브 수수료 모델에서 사용되는 weight(가중치)와 유사하지만 완전히 동일하지는 않다. 그 결과, 이더리움 트랜잭션의 수수료는 약간 다른 방식으로 계산된다. 단순화된 공식은 다음과 같다.

$$ethereum_fee = used_gas * (base_fee_per_gas + priority_fee_per_gas)$$

- **used_gas** - 트랜잭션 실행에 소모된 모든 리소스를 포함한다.
- **base_fee_per_gas** - 가스 단위당 사용자가 지불해야 하는 기본 수수료.
- **priority_fee_per_gas** - 사용자가 트랜잭션 우선 처리를 위해 가스 단위당 추가로 지불하는 팁 수수료.

이전 섹션에서 다룬 **네이티브 수수료 모델**과 비교하면, 이더리움의 트랜잭션은 **사용자에게 노출되는 정보가 적고 설정 가능성이 낮다**. 또한 네이티브 모델과 달리 **렌트 수수료(rent fee)**가 존재하지 않는다. 즉, 스토리지를 생성할 때 그 비용이 이미 가스 수수료에 포함되며, 나중에 스토리지를 삭제하더라도 사용자는 환불을 받지 않는다.

두 시스템 간 수수료를 정렬(alignment)하기 위해, 아스타 네트워크의 EVM 수수료 공식은 다음과 같이 조정된다. 즉, **base_fee_per_gas**가 각 블록마다 동적으로 계산되는 파라미터로 변경된다.

$$EVM_fee = used_gas * (base_fee_per_gas + priority_fee_per_gas)$$

$$base_fee_per_gas_n = base_fee_per_gas_{n-1} * (1 + adjustment + \frac{adjustment^2}{2})$$

다음은 구성 파라미터(configuration parameters)이다.

- $base_fee_per_gas_{min}$ - $base_fee_per_gas$ 의 최소값.
- $base_fee_per_gas_{max}$ - $base_fee_per_gas$ 의 최대값.

수수료 모델 파라미터(Fee Model Parameters)

아래 표는 수수료 모델에서 사용되는 모든 파라미터의 값을 정리한 것이다.

파라미터 이름(Parameter name)	시부야 값 (Shibuya)	시덴 값 (Shiden)	아스타 값 (Astar)
$base_weight$	98974	98974	98974
$weight_factor$ (per byte)	0.030855 SBY	0.00030855 SDN	0.030855 ASTR
$length_factor$ (per byte)	0.0000235 SBY	0.000000235 SDN	0.0000235 ASTR
$max_block_normal_dispatch_weight$	375,000,000,000	375,000,000,000	375,000,000,000
s^*	0.25	0.25	0.25
v	0.000015	0.000015	0.000015
c_{min}	0.1	0.1	0.1
c_{max}	10	10	10
$price_per_item$	0.00004 SBY	0.0000004 SDN	0.00004 ASTR
$price_per_byte$	0.000001 SBY	0.00000001 SDN	0.000001 ASTR
$base_fee_per_gas_{min}$	0.0000008 SBY	0.000000008 SDN	0.0000008 ASTR
$base_fee_per_gas_{max}$	0.00008 SBY	0.0000008 SDN	0.00008 ASTR

위의 파라미터 값은 **EVM 수수료와 네이티브 수수료가 동일하도록 조정**되어 있으며, 평균적인 트랜잭션(중간 수준의 weight 및 length, rent fee 없음)의 경우 약 **0.5 ASTR** 또는 **0.005 SDN** 수준의 수수료가 발생하도록 설정되어 있다.

렌트 수수료(Rent Fee)

온체인(On-chain)에서 특정 작업을 실행하면 **추가적인 스토리지(storage)**가 생성될 수 있다. 예를 들어, 어떤 계정이 자신의 **아이덴티티(identity)**를 생성하기로 하면, 이 정보는 온체인에 저장되어야 하며, 노드를 운영하는 모든 참여자는 이 데이터를 저장해야 한다. 이러한 과도한 데이터 생성을 방지하기 위해(스팸 방지 목적), 일반적으로 해당 작업에는 일정한 **렌트(rent) 수수료**가 부과되며, 이 금액은 **예치(deposit)** 형태로 지불된다.

일반적인 트랜잭션 수수료와 달리, 저장된 항목이 삭제되면 **렌트 수수료도 계정에 반환(refund)**된다. 각 항목당 및 바이트당 구체적인 수치는 앞선 표를 참고하면 된다.

렌트 수수료가 부과되는 주요 작업은 다음과 같다:

- 아이덴티티 생성(하위 계정 포함)
- 멀티시그(multisig) 호출 생성
- 자산 및 메타데이터(asset & metadata) 생성 (**참고: 아래 표 참조**)
- 프록시(proxy) 생성 및 프록시 호출 공지(announce)
- WASM 스마트 컨트랙트와의 상호작용

pallet-assets의 **자산(asset)**을 생성할 때는 일반적인 작업보다 **더 높은 비용**이 부과된다. 이는 자산 생성이 단순히 프록시를 생성하는 것보다 훨씬 더 높은 **가중치(weight)**를 갖기 때문이다. 또한 생성된 자산은 **크로스체인 자산(cross-chain asset)**으로 등록될 수 있으며, **프리컴파일(precompile)**을 통해 상호작용이 가능하므로 모든 계정에서 사용할 수 있다.

작업(Action)	시부야 가격 (Shibuya)	시덴 가격 (Shiden)	아스타 가격 (Astar)
자산 생성(Asset creation)	10 SBY	10 SDN	1000 ASTR

인플레이션(Inflation)

아스타 네트워크는 연간 인플레이션율이 **소프트 캡(soft cap)**으로 설정되어 있으며, **최대 공급량에는 상한이 없다(uncapped max supply)**. 인플레이션으로 발생한 신규 발행 토큰은 네트워크 내 여러 참여자에게 비율에 따라 분배된다.

이 참여자에는 다음이 포함된다:

- 스테이커(stakers)

- dApp 소유자(dApp owners)
- 콜레이터(collators)
- 트레저리(treasury)

기본 시간 단위(Basic Time Units)

사이클(Cycles)

사이클은 아스타 네트워크에서의 '연도(year)'에 해당하는 개념이다. 새로운 사이클이 시작될 때마다, 당시의 총 발행량(total issuance)에 따라 새로운 인플레이션 설정이 계산된다. 예를 들어 연간 인플레이션율이 7%이고, 총 발행량이 1,000,000 ASTR이라면, 해당 사이클의 소프트 캡 인플레이션은 70,000 ASTR이 된다.

사이클 길이는 설정 가능하지만, 실제로는 오프체인 기준으로 약 1년에 해당하는 기간과 유사하다.

$$inflation_soft_cap = inflation_rate * total_issuance$$

기간(Periods)

각 사이클(cycle)은 하나 이상의 기간(period)으로 구성된다. 기간(period)은 dApp 스테이킹 프로토콜의 핵심 시간 단위이며, 각 기간은 Voting과 Build&Earn 두 개의 하위 기간(subperiod)으로 나뉜다.

Voting 하위 기간 동안에는 스테이커(staker)나 dApp 소유자는 보상을 받을 수 없으며, 콜레이터(collators)와 트레저리(treasury)만이 계속 보상을 받는다. 반면 Build&Earn 하위 기간에서는 스테이커가 각 에라(era)마다 스테이킹 보상을 받으며, 기간이 끝날 때 보너스 리워드(bonus reward)를 받을 수 있다. dApp 소유자 또한 dApp 스테이킹 성과에 따라 각 에라 종료 시점마다 보상을 받는다.

에라(Eras)

각 기간(period)은 여러 개의 에라(era)로 구성된다. 에라는 dApp 스테이킹의 기본 시간 단위이며, 길이는 블록 수로 측정된다.

Build&Earn 하위 기간은 하나 이상의 표준 에라(standard era)로 구성되며, 각 에라는 고정된 길이를 가진다.

Voting 하위 기간은 항상 하나의 에라로 구성되며, 이 에라의 블록 길이는 표준 에라보다 길 수 있지만, 항상 표준 에라 길이의 배수로 설정된다.

예시(Example)

- 블록 생성 간격: 12초
- 에라 길이: 7200 블록 (24시간, 즉 1일 — 표준 에라 길이)
- Voting 하위 기간 길이: 10 에라 (단일 투표 에라, 총 10 × 7200 블록)
- Build&Earn 하위 기간 길이: 81 에라
- 사이클 길이: 4 기간(Periods)

이 구성을 기준으로 하면, 하나의 사이클은 약 364일(약 1년)에 해당하며, 각 기간은 약 3개월 정도의 길이를 갖는다.

재계산(Recalculation)

새로운 사이클(cycle)이 시작되면, 소프트 인플레이션 캡(soft-inflation cap)이 다시 계산되며, 그 결과에 따라 네트워크의 모든 참여자(reward actors)에 대한 보상 비율이 조정된다.

이 인플레이션 상한이 하드캡(hard-cap)이 아닌 소프트캡(soft-cap)인 이유는 스테이커(staker)와 dApp 보상이 분배되는 방식 때문이다. 보상은 실제로 청구될 때(mint on claim)에만 발행되므로, 재계산 시점에 일부 보상이 아직 청구되지 않은 상태일 수 있다. 따라서 이론적으로는 인플레이션이 소프트캡을 초과할 가능성이 존재하지만, 실제 운영 환경에서는 지연 발행(lazy minting) 및 수수료 소각(fee burn) 메커니즘으로 인해 이러한 상황은 극히 드물다.

인플레이션 분배(Inflation Distribution)

계산된 소프트캡을 기준으로, 네트워크의 모든 참여자에게 보상이 조정되어 분배된다.

콜레이터(Collators)

콜레이터는 사이클의 소프트캡 인플레이션 중 일정 비율을 고정(fixed amount)으로 받는다. 이 금액은 해당 사이클의 전체 블록 수로 균등하게 나누어 분배된다.

$$collator_reward_per_block = \frac{total_collator_cycle_reward}{blocks_per_cycle}$$

트레저리(Treasury)

콜레이터와 마찬가지로, 트레저리 또한 사이클의 소프트캡 인플레이션 중 일정 비율을 고정적으로 받으며, 사이클 내 모든 블록에 걸쳐 균등하게 분배된다.

$$treasury_reward_per_block = \frac{total_treasury_cycle_reward}{blocks_per_cycle}$$

dApp

dApp 보상은 Build&Earn 하위 기간(subperiod)의 각 에라(era) 가 종료될 때마다 할당(assign) 된다. 따라서 사이클 전체의 dApp 보상 금액은 사이클 내의 모든 Build&Earn 에라 수로 균등 분배된다.

$$dapp_reward_pool_per_era = \frac{total_dapp_cycle_reward}{periods_per_cycle * eras_per_build_and_earn}$$

dApp 스테이킹 프로토콜은 각 dApp에 스테이킹된 금액을 기준으로, 각 dApp이 받아야 할 보상 금액을 자동으로 계산한다.

스테이커(Stakers)

스테이커 보상은 두 부분으로 구성된다 —

1. 일반적인 스테이킹 보상(regular staking rewards) ,
2. 보너스 보상(bonus rewards) .

일반 스테이킹 보상(Regular Staker Rewards)

일반 스테이킹 보상은 dApp에 네이티브 토큰인 **ASTR**을 스테이킹한 사용자에게 지급된다. 이 보상은 **기본(base)** 보상과 **가변(adjustable)** 보상으로 나뉜다.

기본 보상은 각 에라 종료 시점에, 스테이킹 총액과 관계없이 보상 풀에 일정 금액이 할당된다.

$$base_staker_reward_pool_per_era = \frac{total_base_staker_cycle_reward}{number_of_cycles * eras_per_build_and_earn}$$

가변 보상은 네트워크의 총 스테이킹 금액(total value staked) 과 목표 스테이킹 값(target stake value) 에 따라 동적으로 결정된다. 이 금액은 스테이킹 총액이 증가할수록 선형적으로(linearly) 증가하며, 목표치에 도달하면 포화(saturate)된다. 이 구조를 통해 스테이킹 보상은 **제로섬(zero-sum)** 형태가 아니다.

$$max_adjustable_staker_reward_pool_per_era = \frac{total_adjustable_staker_cycle_reward}{number_of_cycles * eras_per_build_and_earn}$$

가변 보상은 각 에라 종료 시점에 다음의 수식을 사용해 계산된다.

$$adjustable_factor = \min(1, \frac{total_value_staked_percent}{ideal_staking_percent})$$

이를 바탕으로 가변 보상 풀은 다음과 같다.

$$adjustable_staker_reward_pool = max_adjustable_staker_reward_pool_per_era * adjustable_factor$$

adjustable_factor가 1 미만일 경우, 남은 부분은 발행되지 않으며, 이는 전체 인플레이션을 줄이는 효과를 가져온다.

이제 위의 수식을 통해 특정 스테이커(예: Alice)가 에라 n에서 받는 스테이킹 보상은 다음과 같이 표현된다.

$$staker_reward_per_era_{Alice} = \frac{staked_value_{Alice,n}}{total_staked_value_n} * (base_staker_reward_pool_per_era_n + adjustable_staker_r$$

보너스 보상(Bonus Rewards)

스테이커가 Voting 하위 기간 동안 스테이킹을 시작하고, Build&Earn 하위 기간 동안 스테이킹 금액을 감소시키지 않았다면, 해당 스테이커는 **보너스 보상(bonus reward)** 을 받을 자격이 주어진다.

보너스 보상 풀은 **기간(period)** 단위로 할당되며, 다음과 같이 계산된다.

$$bonus_reward_pool_per_period = \frac{total_bonus_cycle_reward}{periods_per_cycle}$$

스테이커 Alice의 보너스 보상은 다음과 같이 표현된다.

$$bonus_staker_reward_{Alice} = \frac{voting_subperiod_staked_value_{Alice}}{total_voting_subperiod_staked_value} * (bonus_reward_pool_per_period)$$

지연 발행(Lazy Minting)

스테이커(staker) 와 dApp의 보상은 모두 **지연 발행(lazy minting)** 방식으로, 즉 필요한 시점에만 발행된다. 반면 **콜레이터(collator)** 와 **트레저리(treasury)** 의 보상은 블록 단위로 발행된다.

가변 스테이킹 보상(adjustable staker reward)과 dApp 스테이킹의 **티어 시스템(tier system)** 구조로 인해, 실제 인플레이션은 계산된 **소프트캡(soft-cap)** 보다 훨씬 낮게 유지된다. 가변 보상의 최대치가 발행되는 것은 **이상적인 스테이킹 비율(ideal staking rate)** 이 충족되거나 초과될 때뿐이다. 또한 dApp 보상의 경우 모든 티어가 완전히 채워질 가능성은 낮다. 티어의 용량(capacity)이 실제 수요보다 크거나, 일부 dApp이 충분한 지지를 얻지 못할 경우 해당 보상은 아예 발행되지 않으며, 이는 결과적으로 **실질 인플레이션을 낮추는 요인**이 된다.

보상은 영구적으로 유지되지 않으며, **만료(expire)** 되기 전에 반드시 **청구(claim)** 해야 한다. 이 만료 조건은 비교적 느슨하게 설정되어 있지만, 청구하지 않아 만료되는 경우도 발생할 수 있다.

앞서 언급한 메커니즘들은 **소각(burn)** 방식이 아니라, 단지 **발행 시점을 지연시키는 방식**이라는 점에 주의해야 한다. 아스타 네트워크의 주요 소각 메커니즘은 **수수료 체계(fee system)** 내에 존재하며, 수수료의 상당 부분이 소각된다. 이로 인해, 실제로는 단일 사이클 내에서 발행 및 할당되는 보상 기준으로 볼 때 **소프트캡 인플레이션 한도**에 도달하는 것은 사실상 불가능하다.

파라미터(Parameters)

파라미터(Parameter)	아스타(Astar)	시덴(Shiden)	시부야(Shibuya)
Periods Per Cycle	3	6	2
Eras Per Voting Subperiod	11	6	8
Eras Per Build&Earn Subperiod	111	55	20
Blocks Per Era	7200 (~24시간)	7200 (~24시간)	1800 (~6시간)
Cycle Inflation Rate	7%	7%	1%
Treasury Part	5%	5%	5%
Collators Part	3.2%	3.2%	3%
dApps Part	13%	13%	20%
Base Staker Part	10%	10%	25%
Adjustable Staker Part	55%	58.8%	35%
Bonus Part	13.8%	10%	12%
Ideal Staking Rate	50%	50%	20%

크로스체인 메시징(Cross-Chain Messaging, XCM)

XCM(Cross-Consensus Message)은 서로 다른 **합의 메커니즘(consensus model)** 을 사용하는 블록체인의 네트워크 간의 통신을 가능하게 하는 **메시징 프로토콜**이다. 폴카닷(Polkadot) 패러체인(parachain) 간 통신뿐만 아니라, 다른 합의 구조를 가진 블록체인 간에도 메시지를 교환할 수 있다. XCM은 **임의의 메시지 전송(arbitrary message passing)** , **자산 전송(asset transfer)** , **원격 함수 호출(remote function call)** 등을 지원하며, 이로써 서로 다른 체인 간의 **상호운용성(interoperability)** 을 실현한다.

이를 통해 개발자는 패러체인 전용 dApp뿐만 아니라, 여러 블록체인에 걸쳐 작동하는 **멀티체인 dApp**을 구축할 수 있다. XCM을 활용하면 서로 다른 네트워크의 기능을 연동해, 사용자 경험을 향상시키고 프로젝트 간 협력과 통합을 가능하게 한다. 예를 들어 XCM은 **크로스체인 DEX**와 같이 여러 블록체인 간 통신이 필요한 응용 프로그램 개발을 지원한다.

폴카닷 네트워크는 XCM 외에도 다양한 메시징 포맷을 지원하지만, XCM은 그중에서도 특히 강력한 장점을 제공한다. 예를 들어 **XCMP**(Cross-Chain Message Passing, 때로는 HRMP로도 불림)는 패러체인 간의 수평적 메시지 전달(horizontal message passing)을 위한 **고도로 확장 가능하고 프로토콜 비의존적(protocol-agnostic)** 전송 시스템으로, 모든 임의 메시지를 지원한다.

반면 **VMP**(Vertical Message Passing)는 **릴레이체인(Relay Chain)** 과 패러체인 간 메시지 전달 방식으로, 모든 메시지 데이터가 릴레이체인 상에 존재한다. 이로 인해 VMP는 XCMP보다 상대적으로 **확장성이 낮은 방식**으로 평가된다.

거버넌스(Governance)

상태(Status)

이 거버넌스 솔루션은 현재 **시부야(Shibuya)** 및 **아스타(Astar)** 네트워크에서 활성화되어 있다. **시덴(Shiden)** 네트워크에는 배포할 계획이 없다.

참여자자는 일반적인 **서브스트레이트 스타일(SS58)** 계정뿐만 아니라 **이더리움 스타일(H160)** 계정(dispatch 프리컴파일을 통해 접근 가능)도 거버넌스 참여에 사용할 수 있다.

접근 방식(Approach)

Astar는 폴카닷(Polkadot)의 **v1 거버넌스 모델(Gov v1)**을 재활용한다. 이 모델은 실용성을 유지하면서 네트워크의 탈중앙화를 점진적으로 확대하기 위한 것이다. 즉, 완전히 새로운 시스템을 만드는 대신, 검증된 구조를 발전시키는 접근을 취한다.

사용자는 [공식 문서](#)를 참고하여 Gov v1의 세부 작동 방식을 학습할 수 있다.

메인 카운슬(Main Council)과 **기술 위원회(Technical Committee)**는 공식 문서에 설명된 그대로 동일하게 사용된다.

새로운 구성 요소로는 **커뮤니티 카운슬(Community Council)**이 도입된다. 이 위원회는 **커뮤니티 트레저리(community treasury)**와 관련된 의사결정을 담당하며, 커뮤니티 트레저리 자금의 사용 요청을 승인 또는 거절할 수 있다. 또한 트레저리 자금으로 어떤 dApp에 스테이킹할지를 결정할 수 있고, 특정 dApp을 dApp 스테이킹 시스템에 **신속 등록(fast-track)**시킬 권한도 갖는다.

커뮤니티 트레저리는 **온체인 트레저리**와는 **별도의 독립 계정으로**, 주로 dApp 스테이킹 보상을 통해 자금을 확보한다.

커뮤니티 카운슬은 이 자금을 현명하게 관리하여 수익성을 유지하면서도 프로젝트 및 앰배서더를 지원해야 한다.

확신 투표(conviction voting) 역시 적용된다. 이는 폴카닷 Gov v1의 기본 구조와 동일하되, 파라미터 설정만 다르다.

ASTR 토큰 보유자는 누구나 제안을 **발의(propose)**, **지지(endorse)** 또는 **투표(vote)**할 수 있다.

⚠️ 중요:

dApp 스테이킹에 참여 있는 토큰도 **거버넌스 참여에 사용 가능하며**, 반드시 그렇게 활용해야 한다.
사용자는 보상을 받으면서 동시에 네트워크의 방향을 결정할 수 있다.

간략 개요(Quick Overview)

[공식 폴카닷 Gov v1 문서](#)를 중복하지 않기 위해, 아스타의 거버넌스 시스템을 간략히 요약하면 다음과 같다.

- 제안을 제출하기 전에, 실행하고자 하는 호출(call)의 **프리이미지(preimage)**를 온체인에 등록해야 한다. 이는 실행 논리(예: 런타임 업그레이드, 자금 이체 등)를 미리 제출하는 절차이다.
- 공개 제안(public proposal)**은 누구나 제출할 수 있으며, 다른 토큰 보유자의 **지지(endorsement)**를 받을 수 있다. 동시에 여러 제안이 존재할 수 있다.
- 외부 제안(external proposal)**은 **메인 카운슬(Main Council)**만 제출할 수 있으며, 한 번에 하나의 외부 제안만 존재할 수 있다.
- 각 **런치 기간(Launch Period)**의 끝에서, 가장 많은 지지를 받은 공개 제안 또는 외부 제안 중 하나가 **국민투표(referendum)**로 승격된다. (두 개의 “트랙(track)” 존재)
 - 런치 기간은 **공개 제안 라운드** → **외부 제안 라운드**로 번갈아 진행된다.
 - 해당 라운드에 적합한 제안이 없을 경우, 다른 트랙의 제안을 사용한다.
- 제안이 국민투표로 승격되면, **투표 기간(Voting Period)** 동안 토큰 보유자들이 투표한다. 투표가 통과되려면, 투표 종료 시점에 **정족수(quorum)** 요건이 충족되어야 한다.
- 투표가 통과되면, **시행 기간(Enactment Period)**이 지난 후 온체인에서 실행된다. 이는 변경 사항에 대비할 수 있도록 하는 **지연 실행(delay execution)** 메커니즘이다.
- 확신 투표(conviction voting)**를 통해, 사용자는 자신의 토큰을 일정 기간 **잠금(lock)**함으로써 투표 가중치를 높일 수 있다. 예를 들어 **1000 ASTR**를 **확신 계수 3(conviction = 3)**으로 투표하면 **3000 ASTR**으로 투표한 것과 동일한 효과를 얻지만, 잠금 기간은 더 길어진다.
- 투표자가 승리한 측에 속할 경우, 그들의 토큰은 일정 기간 잠금 상태를 유지한다. 반대로 패배한 측의 투표자는 토큰이 즉시 잠금 해제된다. 토큰은 계정을 떠나지 않으며, 단지 **이동 불가(non-transferable)** 상태일 뿐이다.
- 잠금 기간이 종료되면, 사용자는 **unlock** 호출을 실행하여 토큰을 다시 **유동화(liquid)**할 수 있다.
- 토큰 보유자는 자신의 **투표 권한을 위임(delegate)**할 수도 있다. 단, 토큰 자체는 전송되지 않으며, 오직 **투표력만 위임**된다.
- 위임과 직접 투표는 동시에 수행할 수 없으며, **상호 배타적(mutually exclusive)**이다.
- 세 가지 유형의 **정족수(quorum)**가 존재한다. (자세한 내용은 공식 문서를 참고)
 - 단순 과반(Simple Majority):** 전체 투표의 50% 이상이 찬성(aye)일 때 통과. 메인 카운슬의 **다수 동의(majority agreement)**로 제출된 외부 제안에 적용된다.
 - 초과 과반 승인(Super Majority Approve):** 투표 참여율(turnout)에 따라 필요한 찬성 비율이 조정된다. 참여율이 낮으면 더 높은 찬성 비율이 요구되고, 참여율이 높아질수록 낮아진다. 이는 일반 **공개 제안(public proposal)**에 적용된다.
 - 초과 과반 반대(Super Majority Against):** 마찬가지로 참여율에 따라 조정되지만, 이는 메인 카운슬의 **만장일치(unanimous agreement)**로 제출된 외부 제안에 적용된다. 참여율이 높아질수록 요구되는 찬성 비율이 증가한다.
- 메인 카운슬(Main Council)**은 네트워크에 해롭다고 판단될 경우, **진행 중인 국민투표를 취소(cancel)**할 수 있다. (긴급 상황에서만 사용)
- 기술 위원회(Technical Committee)**는 긴급 상황에서 외부 제안을 **신속 처리(fast-track)**할 수 있다. 이를 통해 즉시 국민투표로 전환하고, 투표 및 시행 기간을 짧게 설정할 수 있다.
- 기술 위원회**는 네트워크에 유해하다고 판단되는 **공개 제안(public proposal)**을 **취소(cancel)**할 수도 있다.
- 또한, **기술 위원회**는 메인 카운슬이 제출한 외부 제안을 **거부(veto)**하여 일시적으로 연기시킬 수 있다.

행위자(Actors)

거버넌스 시스템에는 여러 유형의 **행위자(actor)**가 존재하며, 각 행위자는 서로 다른 역할과 권한을 가진다. 거버넌스 관련 호출(call)이 실행될 때, 해당 호출의 **기원(origin)**이 어떤 행위자인지에 따라 권한 수준이 결정된다.

예를 들어, 일반 사용자가 민주주의 제안(democracy proposal)을 제출할 경우, 그 호출의 기원은 사용자의 개인 계정(개인 키로 제어됨)이 된다. 반면, Main Council이 호출을 제출하는 경우, 그 기원은 **집합체(collective)**로서의 Main Council이며, 예를 들어 2/3 다수 동의(majority agreement)가 요구된다. 이 경우, 일반 사용자의 호출보다 높은 권한을 가지지만, 카운슬 구성원 간의 합의가 필요하다.

특수한 형태의 **origin**으로 **root**가 있으며, 이는 시스템 내에서 가장 높은 권한을 가진다.

메인 카운슬(Main Council)

Main Council은 초기 단계에서 **Astar 재단(Astar Foundation)**의 리더와, 네트워크 및 생태계에 깊이 관여한 인물들로 구성된다.

Main Council의 권한은 다음과 같다:

- Main Council, Technical Committee, Community Council의 구성원을 **2/3 다수 동의**로 변경할 수 있다.
- Main Treasury의 지출 제안을 **승인(approve)**하거나 **거절(reject)**할 수 있다.
- **2/3 다수 동의**로 다음 유형의 외부 제안을 제출할 수 있다.
 - Super Majority Approve
 - Simple Majority
- **만장일치(unanimous agreement)**로 Super Majority Against 유형의 외부 제안을 제출할 수 있다.
- **2/3 다수 동의**로 이미 통과된 국민투표를 **취소(cancel)**할 수 있다.
(이것은 긴급 상황에서만 가능한 **응급 조치(emergency action)**이다.)

기술 위원회(Technical Committee)

Technical Committee는 초기에는 **Astar 런타임 개발자**들로 구성된다. 이 역할은 체인의 기술적 구조에 대한 깊은 이해가 필요하기 때문이다.

Technical Committee의 권한은 다음과 같다:

- **외부 제안(external proposal)**을 신속 처리(fast-track)할 수 있다.
 - **표준 신속 처리(standard fast-track)**는 2/3 다수 동의가 필요하며, 투표 및 시행 기간의 길이를 수정할 수 있다.
 - **즉시 신속 처리(instant fast-track)**는 만장일치 동의가 필요하며, 투표 및 시행 기간을 임의로 짧게 설정할 수 있다. (긴급 상황용)
- 네트워크에 유해하다고 판단되는 **공개 제안(public proposal)**을 2/3 다수 동의로 **취소(cancel)**할 수 있다. (국민투표는 취소 불가)
- **dApp 스테이킹(dApp staking)** 팔렛의 유지보수 모드(maintenance mode)를 2/3 다수 동의로 활성화(enable)하거나 비활성화(disable)할 수 있다.
- 아스타 네트워크의 긴급 유지보수 기능을 2/3 다수 동의로 활성화 또는 비활성화할 수 있다.
예:
 - dApp staking의 유지보수 모드 전환
 - 특정 손상된 트랜잭션 일시 중단(pause/unpause)
 - 전체 체인을 **세이프 모드(Safe Mode)**로 전환 또는 해제

또한, Technical Committee의 어떤 구성원(member)도 Main Council이 제출한 외부 제안을 **거부(veto)**할 수 있다.

커뮤니티 카운슬(Community Council)

Community Council은 Astar 재단 구성원과 커뮤니티 앰배서더(ambassador) 등 활발히 활동하는 커뮤니티 인원으로 구성된다.

Community Council의 권한은 다음과 같다:

- dApp 스테이킹 프로토콜에 dApp을 등록(register)할 수 있으며, 2/3 다수 동의가 필요하다.
- dApp 스테이킹 프로토콜에서 dApp을 해지(unregister)할 수 있으며, 4/5 다수 동의가 필요하다.
- 커뮤니티 트레저리(Community Treasury)를 대신하여 dApp 스테이킹을 수행할 수 있으며, 2/3 다수 동의가 필요하다.
- Community Treasury의 지출 요청을 **승인(approve)** 또는 **거절(reject)**할 수 있으며, 2/3 다수 동의가 필요하다.

토큰 보유자(Token Holders)

ASTR 토큰을 보유한 모든 사용자는 거버넌스 시스템에 참여할 수 있다. 토큰 보유자는 다음을 수행할 수 있다:

- **공개 제안(public proposal)**을 제출하며, 일부 ASTR을 예치금(deposit)으로 납부해야 한다.
 - 제안이 취소될 경우, 예치금은 **소각(slash)**된다.
- 기존 제안에 **지지(endorse)**할 수 있으며, 일정량의 ASTR 예치금을 추가로 납부해 제안이 국민투표로 승격될 가능성을 높인다.
- 진행 중인 국민투표(referendum)에 투표(vote)할 수 있다.
- **트레저리 지출 요청(treasury payout request)**을 제출할 수 있다.

파라미터(Parameters)

민주주의(Democracy)

이 파라미터들은 거버넌스 로직의 핵심인 pallet-democracy 기능과 관련되어 있다.

런치 기간(Launch Period) 은 토큰 보유자들이 공개 제안을 지지(endorse)하거나, Main Council이 외부 제안을 제출할 수 있는 기간을 의미한다. 각 라운드의 유형에 따라, 가장 많은 지지를 받은 공개 제안 혹은 외부 제안 중 하나가 국민투표(referendum)로 승격된다.

투표 기간(Voting Period) 은 국민투표가 진행되는 기간으로, 이 기간 동안 토큰 보유자들은 투표할 수 있다. 만약 투표가 통과되어 정족수(quorum)가 충족되면, **시행 기간(Enactment Period)** 이 끝난 뒤 제안이 온체인에서 실제로 실행된다. 이 시행 기간은 제안 실행 전에 준비할 시간을 제공하는 **지연(delayed)** 단계이다.

기술 위원회(Technical Committee) 는 외부 제안을 **신속 처리(fast-track)** 할 수 있다. 일반적인 신속 처리에는 최소 투표 기간이 존재하지만, **즉시 신속 처리(instant fast-track)** 의 경우 투표 기간을 임의로 짧게 설정할 수 있다. 이 기능은 주로 **긴급 상황(emergency situations)** 에 사용된다.

새로운 제안을 제출할 때, 토큰 보유자는 일정량의 네이티브 토큰을 **예치(deposit)** 해야 한다. 이는 스팸 제안을 방지하기 위한 **보험금(insurance)** 의 역할을 한다. 이 예치금은 임의로 설정할 수 있지만, 최소 금액이 존재한다. 또한 예치금은 **지지자(endorser)** 가 동일 금액을 매칭해야 하므로, 너무 큰 금액을 설정할 경우 주의가 필요하다.

투표자가 승리한 측에 속할 경우, 해당 토큰은 일정 기간 동안 **잠금(lock)** 상태가 되며, 이 잠금 기간은 **확신(conviction)** 에 따라 배수로 조정된다.

파라미터 이름(Parameter Name)	시부야(Shibuya)	아스타(Astar)
Launch Period	4일	7일
Voting Period	4일	7일
Enactment Period	2일	2일
Minimum Fast Track Voting Period	1시간	2시간
Vetoed External Proposal Cooloff Period	1일	7일
Minimum Public Referendum Deposit	10 SBY	1000 ASTR
Simple Majority Proposal Origin	Main Council의 1/2 동의	Main Council의 2/3 동의
Super Majority Against Proposal Origin	Main Council 만장일치	Main Council 만장일치
Fast Track Origin	Tech Committee의 1/2 동의	Tech Committee의 2/3 동의
Instant Track Origin	Tech Committee 만장일치	Tech Committee 만장일치
Referendum Cancellation Origin	Main Council의 1/2 동의	Main Council의 2/3 동의
Public Proposal Cancellation Origin	Tech Committee의 1/2 동의	Tech Committee의 2/3 동의

확신 투표(Conviction Voting)

확신 투표는 사용자가 자신의 토큰을 일정 기간 잠금(lock) 상태로 유지함으로써 투표의 **가중치(weight)** 를 높이는 메커니즘이다. 잠금 기간이 길수록 투표의 영향력이 커진다.

확신 단계(Conviction)	기본 잠금 배수(Base Lock Period Multiplier)	잠금 기간(시부야)	잠금 기간(아스타)
0.1	0x (잠금 없음)	0일	0일
1	1x	1일	9일
2	2x	2일	18일
3	4x	4일	36일
4	8x	8일	56일
5	16x	16일	144일
6	32x	32일	288일

트레저리(Treasuries)

네이티브 토큰을 보유한 모든 사용자는 **트레저리 지출(treasury payout)** 을 요청할 수 있다. 다만, 제안을 제출할 때는 일정량의 토큰을 **예치(deposit)** 해야 한다. 예치금은 요청 금액의 일정 비율로 계산되며, 이는 스텝성 제안으로 네트워크를 과부하시키는 것을 방지하기 위한 것이다.

예치금의 최소 및 최대 한도는 별도의 파라미터로 설정되어, 예치금이 지나치게 높거나 낮지 않도록 조정된다.

지출 주기(Spend Period) 는 트레저리 자금이 실제로 지출되는 주기를 의미한다. 즉, 제안이 승인되더라도, 실제 자금 이전은 **지출 주기 경과 후에만** 이루어진다.

트레저리 지출 요청의 **승인(approval)** 과 **거절(rejection)** 은 각 트레저리를 관리하는 **해당 카운슬(council)** 이 담당한다.

메인 트레저리(Main Treasury)

이 파라미터들은 **메인 트레저리(Main Treasury)** 의 운영 로직과 관련되어 있다.

파라미터 이름(Parameter Name)	시부야(Shibuya)	아스타(Astar)
Approve Origin	Main Council의 1/2 동의	Main Council의 2/3 동의
Reject Origin	Main Council의 1/2 동의	Main Council의 2/3 동의
Bond Percentage	5%	5%
Minimum Proposal Bond	100 SBY	100 ASTR
Maximum Proposal Bond	10000 SBY	1000 ASTR
Spend Period	3일	7일

메인 트레저리 주소:

YQnbw3oWxBnCUarnbePrjFcrSgVPP2jqTZYzWcccmN8fXhd

커뮤니티 트레저리(Community Treasury)

이 파라미터들은 **커뮤니티 트레저리(Community Treasury)** 의 운영 로직과 관련되어 있다.

파라미터 이름(Parameter Name)	시부야(Shibuya)	아스타(Astar)
Approve Origin	Community Council의 1/2 동의	Community Council의 2/3 동의
Reject Origin	Community Council의 1/2 동의	Community Council의 2/3 동의
Bond Percentage	5%	5%
Minimum Proposal Bond	100 SBY	100 ASTR
Maximum Proposal Bond	10000 SBY	1000 ASTR
Spend Period	3일	7일

커뮤니티 트레저리 주소:

YQnbw3oWxBju7z5CRVoq1K6JzwDaj6DNePwdc2R2fG7jdix

집합체(Collectives)

이 파라미터들은 Main Council, Technical Committee, Community Council 등 각 **집합체(collective)** 에 해당하는 모듈의 작동 방식과 관련되어 있다.

모든 **집합체 구성원(member)** 은 **모션(motion)** 을 제안할 수 있으며, 이는 네트워크 내에서 특정 권한이 필요한 행동(예: 구성원 변경, 승인 등)을 실행하는 절차이다. 다른 구성원들은 해당 모션에 대해 **찬성(aye)** 또는 **반대(nay)** 투표를 할 수 있다.

모션에는 **지속 기간(duration)** 이 있으며, 이 기간 내에 정족수(quorum)를 충족하지 못하면 모션은 **실패(failed)** 로 간주된다. 정족수가 충족되면 모션은 **통과(passed)** 로 처리된다.

집합체는 0명 이상의 구성원을 가질 수 있으며, 구성원 수는 미리 정의된 상한(limit)을 넘을 수 없다.

현재 구성원 수를 확인하려면, polkadot-js app 또는 전용 거버넌스 UI를 통해 온체인 데이터를 조회하는 것이 가장 정확하다.

파라미터 이름(Parameter Name)	시부야(Shibuya)	아스타(Astar)
Main Council Motion Duration	5일	5일
Technical Committee Motion Duration	5일	5일
Community Council Motion Duration	5일	5일
Main Council Member Change Origin	Main Council의 1/2 동의	Main Council의 2/3 동의
Technical Committee Member Change Origin	Main Council의 1/2 동의	Main Council의 2/3 동의
Community Council Member Change Origin	Main Council의 1/2 동의	Main Council의 2/3 동의
Main Council Member Limit	16명	16명
Technical Committee Member Limit	8명	8명
Community Council Member Limit	16명	32명

dApp 스테이킹(dApp Staking)

이 파라미터들은 **dApp 스테이킹(dApp staking)** 로직과 관련되어 있다.

파라미터 이름(Parameter Name)	시부야(Shibuya)	아스타(Astar)
dApp Register Origin	Community Council의 1/2 동의	Community Council의 2/3 동의
dApp Unregister Origin	Community Council의 4/5 동의	Community Council의 4/5 동의
Community Council Staking Operations	Community Council의 1/2 동의	Community Council의 2/3 동의
dApp Staking Maintenance Mode Origin	Tech Committee의 1/2 동의	Tech Committee의 1/2 또는 Main Council의 2/3 동의

긴급 유지보수 팔렛(Emergency Maintenance Pallets)

세이프 모드(Safe Mode)

SafeMode 팔렛은 체인 전체에 대한 긴급 중단(STOP) 메커니즘을 제공한다. 이 모드에서는 사전에 허용된 제한된 호출만 실행할 수 있으며, 잔액 전송(balance transfer), 자산 전송(asset transfer), 스마트 컨트랙트 호출, dApp 스테이킹 관련 트랜잭션 등은 모두 **금지(disabled)** 된다.

이 메커니즘은 **심각한 버그나 보안 공격**과 같은 긴급 상황에서 체인을 보호하기 위해 설계되었다. Technical Committee와 Main Council 모두가 **세이프 모드(Safe Mode)** 진입 또는 해제를 강제로 실행할 수 있다. 세이프 모드에 진입하거나 해제할 때, **dApp 스테이킹의 유지보수 모드** 또한 자동으로 활성화/비활성화된다.

세이프 모드는 일정 시간 동안만 유지되며, 필요 시 여러 차례 연장할 수 있다. 단, 체인을 **무기한(infinite)** 으로 세이프 모드 상태로 유지하는 것은 불가능하다.

다음은 **세이프 모드(Safe Mode)** 관련 파라미터이다.

파라미터 이름(Parameter Name)	시부야(Shibuya)	아스타(Astar)
Enter Duration	4시간	12시간
Extend Duration	2시간	4시간
Force Enter Origin	Tech Committee의 1/2 또는 Main Council의 2/3 동의	Tech Committee의 1/2 또는 Main Council의 2/3 동의
Force Exit Origin	Tech Committee의 1/2 또는 Main Council의 2/3 동의	Tech Committee의 1/2 또는 Main Council의 2/3 동의

트랜잭션 일시정지(Tx Pause)

TxPause 팔렛은 개별 트랜잭션 호출을 **동적으로 일시 중지(pause)** 할 수 있는 기능을 제공한다. 이 기능은 특정 모듈이나 호출에서 문제가 발생했을 때, 체인 전체를 멈추지 않고 해당 기능만 제한함으로써 문제를 빠르게 완화할 수 있다. 대부분의 경우, 이는 세이프 모드(Safe Mode)보다 더 정밀하고 유연한 대응책으로 선호된다.

Technical Committee와 Main Council은 필요에 따라 개별 호출을 동적으로 **일시 중지(pause)** 하거나 **재개(unpause)** 할 수 있다.

다음은 **TxPause** 로직 관련 파라미터이다.

파라미터 이름(Parameter Name)	시부야(Shibuya)	아스타(Astar)
Pause Origin	Tech Committee의 1/2 또는 Main Council의 2/3 동의	Tech Committee의 1/2 또는 Main Council의 2/3 동의
Unpause Origin	Tech Committee의 1/2 또는 Main Council의 2/3 동의	Tech Committee의 1/2 또는 Main Council의 2/3 동의