

에이브(Aave)

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.

Aave v4

Aave v4는 유동성 관리를 위해 허브 앤 스포크(Hub & Spoke) 모델을 사용한다. 유동성 허브(Liquidity Hub)는 프로토콜 전반의 유동성과 회계를 통합하고, 스포크(Spoke)는 격리된 리스크를 바탕으로 모듈형 차입 기능을 구현한다. 이는 Aave v3의 시장별 개별 풀(market-per-pool) 설계를 하나의 통합된 허브 앤 스포크(Hub & Spoke) 모델로 발전시킨 것으로, 거버넌스가 유동성을 이전하지 않고도 새로운 기능이나 시장을 도입할 수 있게 한다.

유동성 허브(Liquidity Hubs)

유동성 허브는 Aave v4의 중앙 유동성 공급원이다. 허브는 Spoke들을 감독하며, 각 Spoke에 차입을 위한 credit line과 공급을 위한 debit line을 부여한다. 또한 허브는 시스템 전반의 회계를 집행하고, Spoke가 인출하거나 추가할 수 있는 유동성의 규모를 제한한다.

- 유동성과 회계를 중앙화
- 스포크별 credit/debit 한도 부여
- 시스템 전반의 제약사항 집행
- 비상 정지(emergency stop) 제어 제공

허브 자산(Hub Assets)

허브 자산은 유동성 허브에 등록된 ERC-20 토큰이다. 이들은 스포크들이 차입하고 공급하는 공유 유동성을 구성하며, 스포크 간 효율적인 유동성 재사용을 가능하게 한다.

스포크(Spokes)

스포크는 특정 공급 및 차입 사용 사례, 자산 유형, 또는 리스크 프로파일을 관리하는 모듈이다. 사용자는 스포크와 직접 상호작용하며, 스포크는 허브와의 유동성 입출금을 중개한다.

- 로컬 제어(예: 리스크 파라미터, 오라클 상호작용, 비상 정지)
- 스포크별 독립 회계
- 모듈형 구조: 다른 스포크에 영향을 주지 않고 추가/제거 가능

리저브(Reserves)

등록되지 않음 ×

리저브는 특정 스포크 내에서 하나의 허브 자산이 어떻게 공급되고 차입되는지를 정의한다. 각 리저브는 자체적인 리스크 설정을 가지며, 동시에 허브는 credit line과 debit line을 통해 해당 리저브로 유입·유출될 수 있는 유동성의 양을 제어한다.

금리(Interest Rates)

차입 금리와 공급 금리는 활용률 곡선(utilization curve)에 의해 결정된다. 활용률은 특정 자산의 유동성 중 현재 얼마나 많이 차입되었는지를 나타내며, 다음과 같이 표현된다.

총 차입 금액(borrowed) / 총 유동성(total liquidity)

최적 사용 비율(optimal usage ratio) 이하에서는 차입 금리가 완만하게 증가한다. 이를 초과하면 금리가 더 가파르게 상승한다. 공급 금리는 프로토콜 수수료를 제외한 차입자 이자에서 도출되며, 해당 자산의 수요와 공급을 반영한다.

이 곡선은 기본 금리(base rate), 최적 사용 비율 이하의 기울기, 최적 사용 비율 이상의 기울기, 그리고 최적 사용 비율 자체로 구성되며, 각 허브 자산 별로 설정된다.

사용자 리스크 프리미엄(User Risk Premium)은 사용자의 담보 품질에 따라 차입 비용에 추가로 부과되는 이자 가산분을 의미한다. 이는 해당 자산의 공통 차입 금리에 더해지는 백분율 형태의 할증이며, 부채를 담보하기 위해 필요한 담보를 가중치로 반영해 계산한다. 또한 포지션이 변할 때마다 재계산된다.

사용자 포지션(User Positions)

사용자 포지션(user position)은 사용자가 하나의 스포크 내에서 공급하고 차입한 내역을 종합적으로 보여주는 뷰이다. 포지션은 내부적으로 리저브 별로 추적되며, 외부에서는 해당 스포크에서의 사용자 활동을 집계한 형태로 나타난다.

- User Supplies: 사용자가 스포크를 통해 허브에 예치한 자산이다. 시간이 지남에 따라 이자가 누적되며, 허브의 가용 유동성을 확대한다.
- User Borrows: 사용자가 스포크를 통해 허브에서 인출한 자산이다. 이는 상환될 때까지 이자가 누적되는 부채 잔액을 형성한다.

Health Factor는 포지션의 지급 여력(solvency)을 나타내며, 청산까지 얼마나 가까운지를 보여준다. 다음과 같이 계산한다.

적격 담보 가치 / 부채 가치

청산을 피하려면 반드시 1보다 커야 한다. 부채를 상환하거나 담보 가치가 상승하면 증가한다. 반대로 이자가 누적되거나 담보 가치가 하락하거나 새로운 부채를 부담하면 감소한다.

Liquidations

포지션의 health factor가 1.0 아래로 떨어지면 청산 대상이 된다. Aave v4는 v3보다 자본 효율성과 공정성을 개선한 새로운 청산 엔진을 도입한다.

- Target Health Factor: 고정된 close factor 대신, 청산자는 포지션을 스포크 수준에서 설정된 Target Health Factor까지 회복시키는 데 필요한 만큼의 부채만 상환한다. 이를 통해 과도한 청산(over-liquidation)을 방지한다.
- Variable Liquidation Bonus: 청산자에게 지급되는 보너스는 포지션의 health factor에 따라 달라진다. health factor가 낮을수록 더 높은 보너스를 제공하며, 이를 통해 가장 위험한 포지션이 우선 처리되도록 하는 더 치욕선(Dutch-auction) 방식의 인센티브를 형성한다.

- Dust Prevention: 청산 후 남게 되는 부채나 담보가 1,000달러 미만이 될 경우, 청산자는 해당 포지션을 전부 정리해야 한다. 이를 통해 소액 잔여분이 누적되는 것을 방지한다.

유동성 프로토콜(Liquidity Protocol)



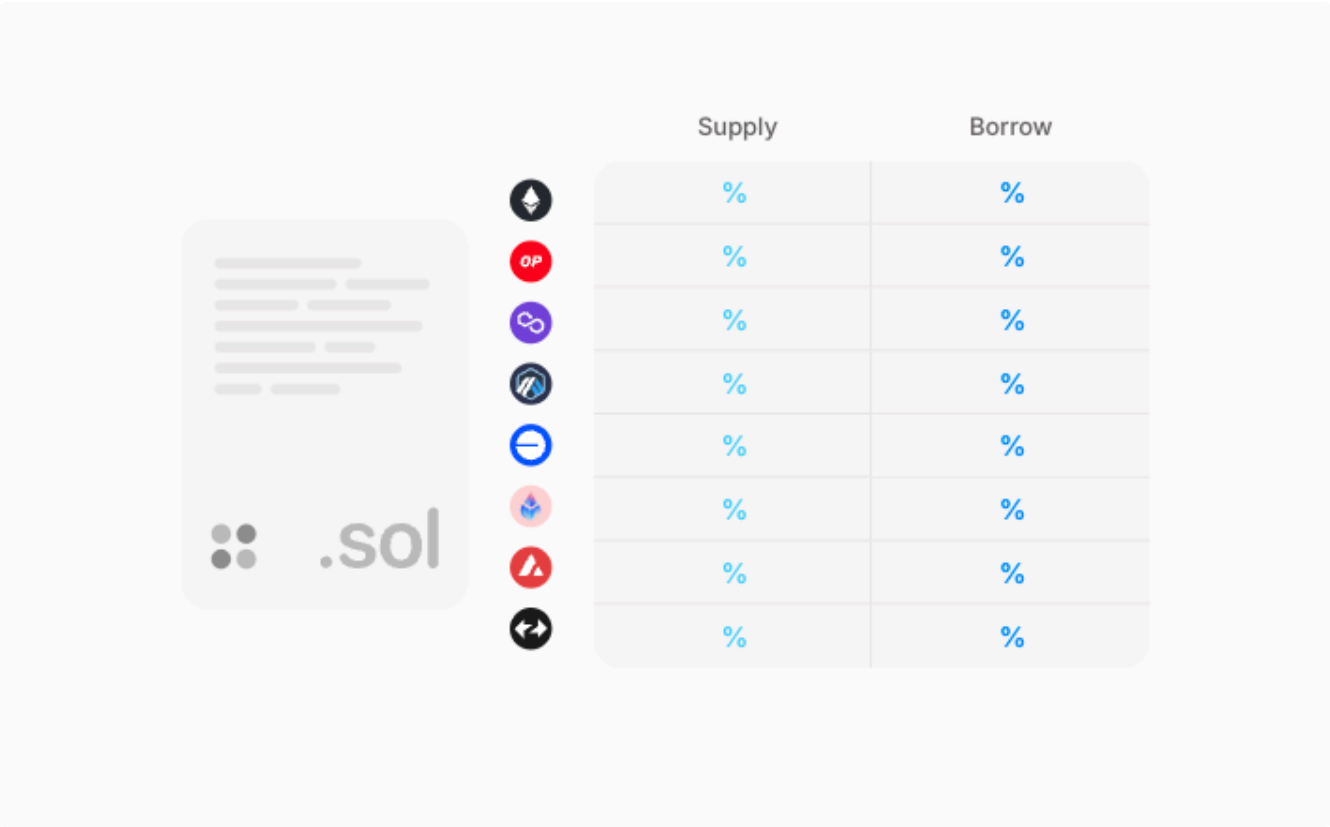
유동성 프로토콜은 디지털 자산의 전송을 용이하게 하는 탈중앙화된 스마트 컨트랙트 시스템이다. 공급(supply)과 차입(borrow) 모델을 기반으로 운영되는 선도적인 유동성 프로토콜인 에이브(Aave)는 사용자가 자산을 유동성 풀에 공급하게 하고, 그 대가로 다른 참여자가 자신들의 담보를 이용해 해당 풀에서 자금을 빌릴 수 있도록 허용한다. 이 프로토콜은 여러 블록체인 네트워크에서 운영되므로 다양한 생태계의 사용자가 접근하기 용이하다.

탈중앙화 유동성 프로토콜의 핵심 특징 중 하나는 비수탁(non-custodial) 성격으로, 사용자가 항상 자신의 자산에 대한 통제권을 유지한다는 점이다. 프로토콜과의 상호작용은 자기 수탁 지갑(self-custodial wallet)을 통해 이루어지며, 사용자는 중개자에 의존하지 않고 직접 자금을 공급하거나 차입할 수 있다. 이러한 모든 과정은 공개적으로 접근 가능하고 누구나 사용할 수 있는(permissionless) 스마트 컨트랙트를 통해 관리되며, 담보 비율(collateral ratios)이나 시장 매개변수와 같은 사전 정의된 조건에 따라 거래를 실행하고 검증한다. 이를 통해 투명하고 신뢰를 수반하지 않는(trustless) 경험을 제공한다.

에이브는 AAVE 토큰 보유자들에 의해 거버넌스가 이루어진다. 이러한 탈중앙화된 거버넌스 모델은 프로토콜의 적응력과 접근성을 더욱 강화한다. 거버넌스 토큰 보유자는 이자율, 담보 요건, 차입자 및 공급자 모두에게 영향을 미치는 주요 매개변수의 조정 등 프로토콜 변경 사항을 제안하고, 투표하며, 실행할 수 있다. 이러한 탈중앙화된 관리 구조는 프로토콜이 중앙집중적 통제 없이 커뮤니티의 요구에 따라 발전할 수 있도록 한다.

디파이(DeFi) 플랫폼으로서 에이브는 자체 스테이블코인인 GHO부터 자산 스왑이나 상환과 같은 복잡한 작업을 단순화하는 주변(periphery) 컨트랙트에 이르기까지 다양한 기능을 포함한다. 에이브와 같은 유동성 프로토콜의 유연성은 블록체인 네트워크에 접근할 수 있는 누구에게나 더욱 개방적이고, 투명하며, 공정한 금융 서비스를 제공할 수 있는 디파이의 잠재력을 보여준다.

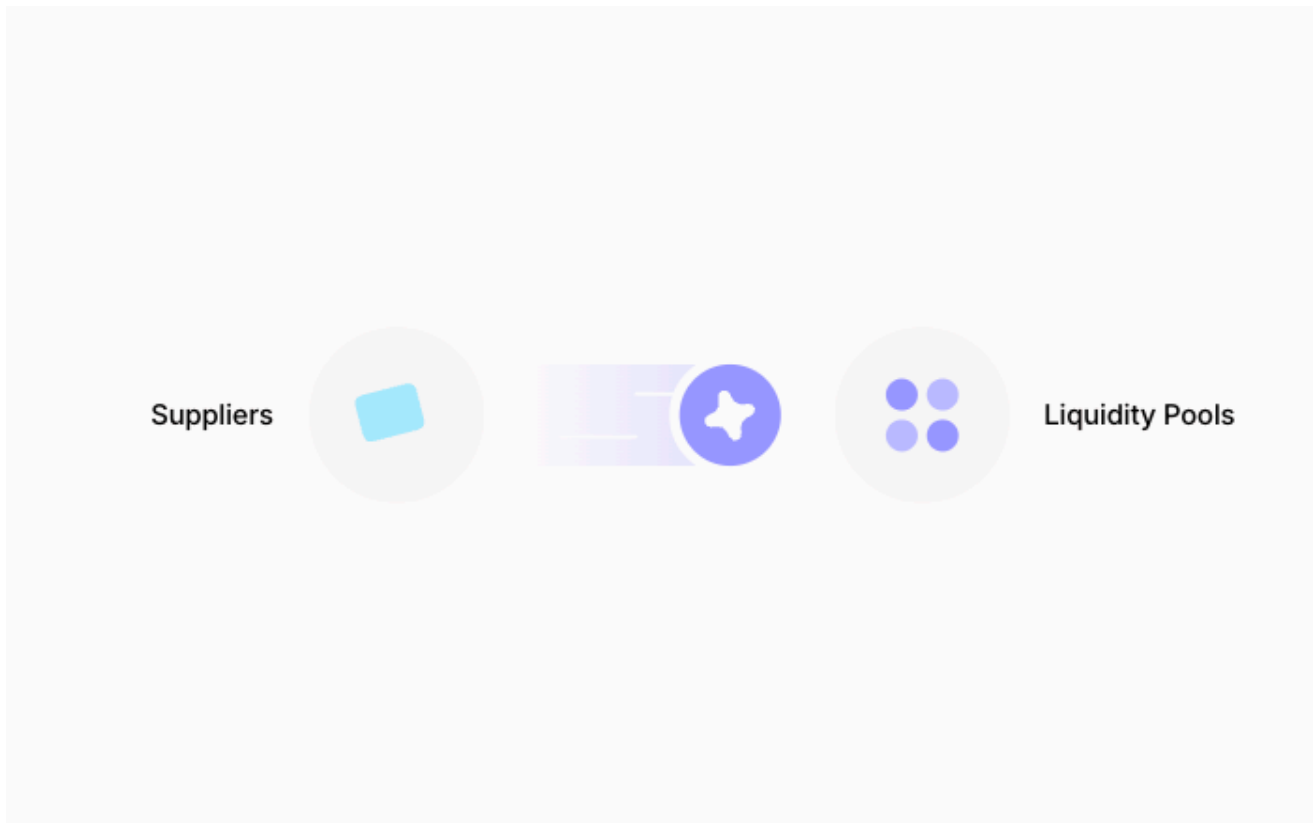
유동성 풀(Liquidity Pool)



유동성 풀은 사용자가 공급자(supplier) 또는 차입자(borrower)로 참여할 수 있게 하는 에이브의 마켓 인스턴스(instance)이다. 리저브 구성(reserve configurations), 담보화 임계치(collateralisation thresholds)와 같이 거버넌스에 의해 승인된 매개변수가 각 풀을 정의한다. 공급자는 차입자가 초과담보(overcollateralised) 포지션을 통해 접근할 수 있는 유동성을 풀에 제공한다. 그 대가로 공급자는 이자를 얻고, 차입자는 담보를 기반으로 유동성을 얻을 수 있으며, 이 모든 과정은 탈중앙화된 스마트 컨트랙트를 통해 이루어진다.

에이브의 유동성 풀은 블록체인 네트워크 상에서 운영되며, 체인과 준비금 매개변수를 정의하는 의사결정에 의해 관리된다. 매개변수 결정은 다양한 행위에 필요한 유동성 수요와 위험 관리 간 균형을 맞추어야 한다. 스마트 컨트랙트의 사용은 이러한 매개변수를 검증하고, 차입, 상환, 청산 절차를 중개자 없이 원활하게 실행한다. 이러한 탈중앙화 접근 방식은 풀 내 금융 상호작용의 투명성, 효율성, 보안을 강화한다.

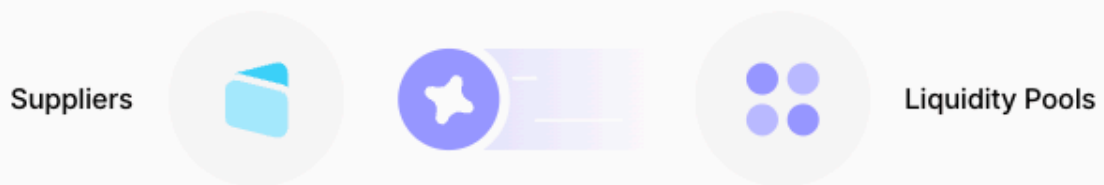
공급(Supply)



사용자가 에이브 프로토콜에 토큰을 공급하면 이에 대해 이자를 얻을 수 있을 뿐만 아니라 공급한 토큰을 담보로 활용할 수도 있다. 토큰이 공급되면 에이브 유동성 풀로 전송되는데, 이는 초과담보 차입을 가능하게 하는 스마트 컨트랙트 시스템이다. 에이브에 공급된 토큰은 현재 시장 공급률(market supply rate)에 따라 자동으로 이자를 축적한다. 공급된 토큰 잔액이 증가함에 따라 이자가 동적으로 축적되며, 이는 공급자에게 할당된 현재 이자율을 반영한다.

공급된 토큰의 이자율은 차입 활용률(borrow utilisation rate) — 현재 차입된 자산 비율을 풀에 공급된 총액 대비로 측정 — 및 커뮤니티 의사결정을 통해 조정 가능한 거버넌스 매개변수에 의해 결정된다. 이러한 매개변수에는 담보화 요건, 공급자 및 차입자에 대한 이자율이 포함되며, 이는 토큰 잔액, 오라클 가격, 차입 활용 비율과 같은 온체인 입력값에 의해 영향을 받는다. 유동성이 풀에 공급, 차입, 상환 또는 인출됨에 따라 이자율이 이에 맞게 갱신된다.

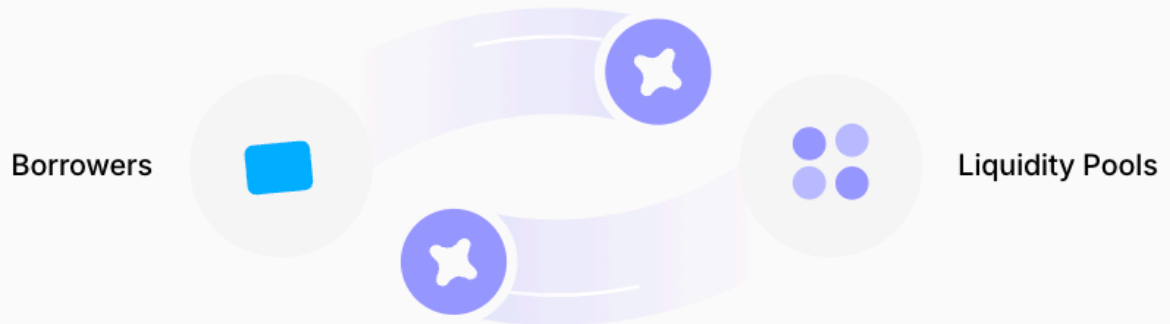
인출(Withdraw)



에이브 프로토콜은 리저브에 충분한 미차입 유동성이 존재하는 한 공급자가 공급한 토큰과 누적 이자를 인출할 수 있도록 허용한다. 인출 금액은 가용 기초 자산에 의해 제한되며, 인출 후 사용자가 자신의 차입 포지션에 대해 충분한 담보 비율을 유지할 수 있어야 한다. 인출 및 변환(Switch)과 같은 기능을 가진 주변(periphery) 컨트랙트는 사용자가 다른 토큰으로 공급된 유동성을 상환할 수 있도록 하여, 보다 효율적인 자산 관리 옵션을 제공한다.

활성 차입 포지션이 있는 상태에서 인출할 경우, 청산을 피하기 위해 건전한 담보화 비율을 유지하는 것이 중요하다. 담보를 줄이면 건전성 지수(health factor)가 낮아지고 청산 위험이 커진다. 안전을 유지하려면, 인출 이후 계정이 청산 임계치 매개변수 이상을 유지해야 한다. 따라서 인출 시 청산을 피하기 위해서는 전체 차입 포지션을 고려한 신중한 관리가 필요하다.

차입(Borrow)



에이브 프로토콜에서 토큰을 차입하면, 사용자는 예치한 토큰을 담보로 활용하여 자산을 매도하지 않고도 유동성에 접근할 수 있다. 그러나 담보의 가치가 요구되는 기준 이하로 떨어질 경우, 차입자는 청산 위험에 직면하게 된다. 이자율은 프로토콜 요인과 거버넌스 결정에 의해 동적으로 산정되며, 커뮤니티의 의견에 따라 시간이 지남에 따라 변할 수 있다. 이자는 차입된 유동성의 비율을 나타내는 활용률(utilisation rate)에 기반하여 발생한다. 높은 활용률은 더 높은 이자율로 이어지며, 수요에 따라 조정된다. 각 리저브는 차입자와 공급자 모두에게 적절한 유인을 제공하기 위해 설계된 특정 매개변수를 가진다.

건전한 비율을 유지하고 청산 위험을 피하기 위해, 차입자는 자신의 담보화 수준을 적극적으로 모니터링하고 건전성 지수를 점검해야 하며, 이를 통해 차입 포지션이 초과담보 상태를 유지하도록 보장해야 한다.

상환(Repay)

Borrowers



Liquidity Pools

에이브 프로토콜에서 차입한 토큰을 상환하는 것은 차입 포지션을 관리하는 중요한 단계이다. 차입자는 동일한 토큰으로 상환하거나, 동일한 기초 자산의 aToken(담보 토큰)으로 상환할 수 있다. 또한 주변(periphery) 컨트랙트를 활용하면, 다른 담보 자산 등 다른 토큰으로도 상환이 가능해 사전에 수동으로 변환할 필요가 없다. 이러한 유연성은 차입자가 필요할 때 포지션을 보다 쉽게 관리하고 종료할 수 있도록 돕는다.

상환은 담보화 비율을 높여 적절한 담보 상태를 보장하고 청산을 방지한다. 상환을 통해 차입된 금액 대비 담보를 강화함으로써 자산이 청산되는 것을 막고, 차입자가 안전하게 담보 일부를 인출할 수 있도록 한다.

리저브(Reserve)

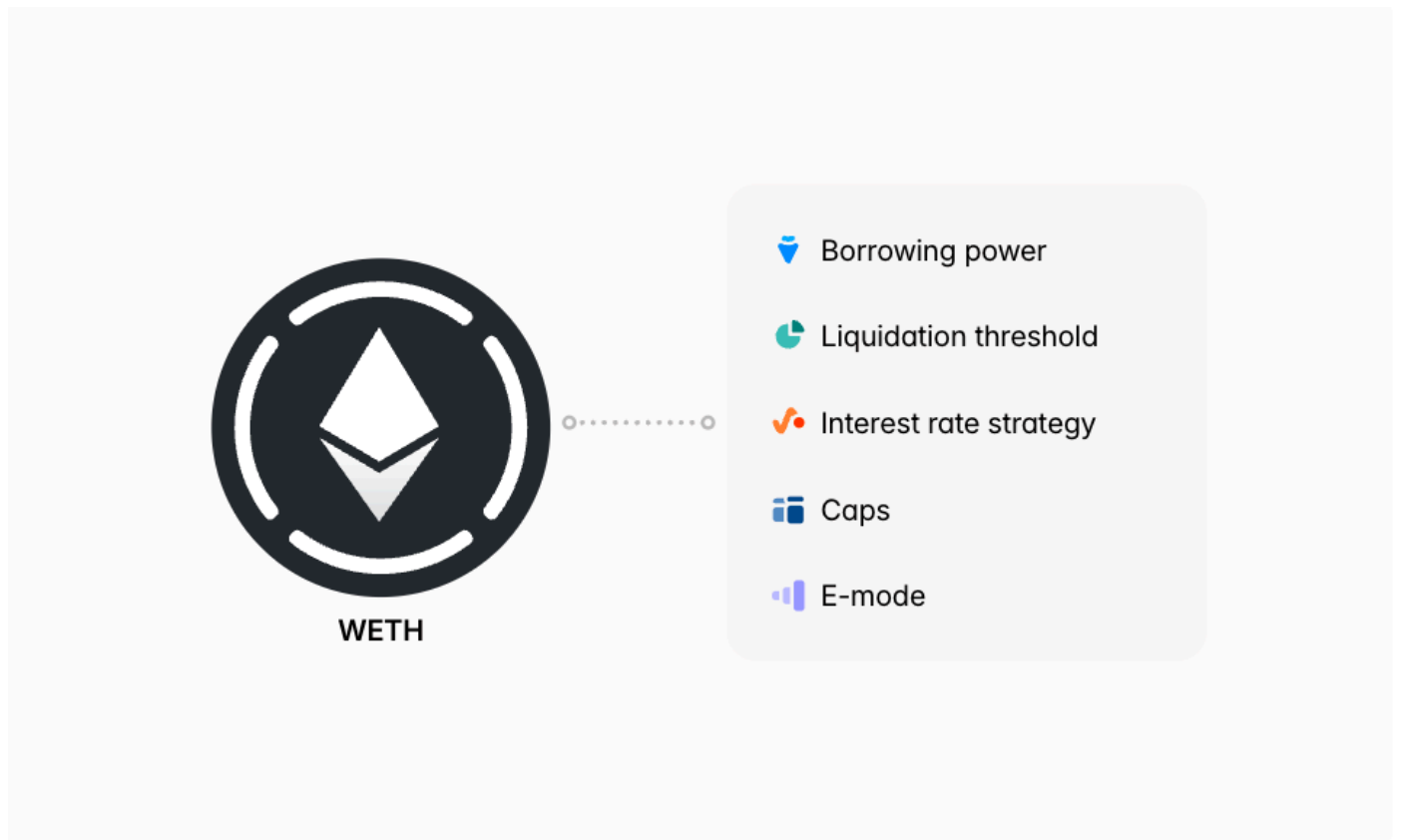
리저브는 에이브의 유동성 풀 내 특정 토큰의 인스턴스를 의미한다. 각 리저브는 리스크를 관리하고 유동성을 최적화하기 위한 일련의 매개변수에 의해 운영된다. 이러한 매개변수는 동일한 기초 자산이라 하더라도 서로 다른 마켓마다 달라질 수 있으며, 이를 통해 에이브는 다양한 네트워크와 풀 조건에 적응할 수 있다.

주요 리저브 매개변수(Key Reserve Parameters)

1. 담보 인정 비율(Loan-to-Value, LTV): 담보 가치 대비 차입할 수 있는 최대 금액을 의미한다. 예를 들어, 75% LTV는 담보 가치의 75%까지 차입할 수 있음을 의미한다. LTV가 0%인 자산은 담보로 활용할 수 없다.
2. 청산 임계치(Liquidation Threshold): 포지션이 청산 위험에 노출되는 시점을 정의한다. 담보 가치가 임계치를 초과하면, 포지션은 차입자의 부채를 상환하기 위해 청산될 수 있다.
3. 차입 가능 여부(Borrowing Enabled): 특정 리저브의 유동성이 차입 가능한지를 결정한다.
4. 한도(Caps): 공급 한도(Supply Cap)와 차입 한도(Borrow Cap)는 리저브에서 특정 토큰이 공급되거나 차입될 수 있는 총량을 제한한다. 이러한 한도는 유동성을 유지하고 변동성이 큰 시장 상황에서 과도한 익스포저(exposure)를 방지하는 데 필수적이다.
5. 이자율 모델(Interest Rate Model): 에이브의 이자율은 유동성 풀의 활용도에 따라 동적으로 조정된다. 더 많은 유동성이 차입될수록 자산의 가용성이 줄어들고, 이에 따라 이자율이 상승하여 출금과 청산에 필요한 충분한 유동성

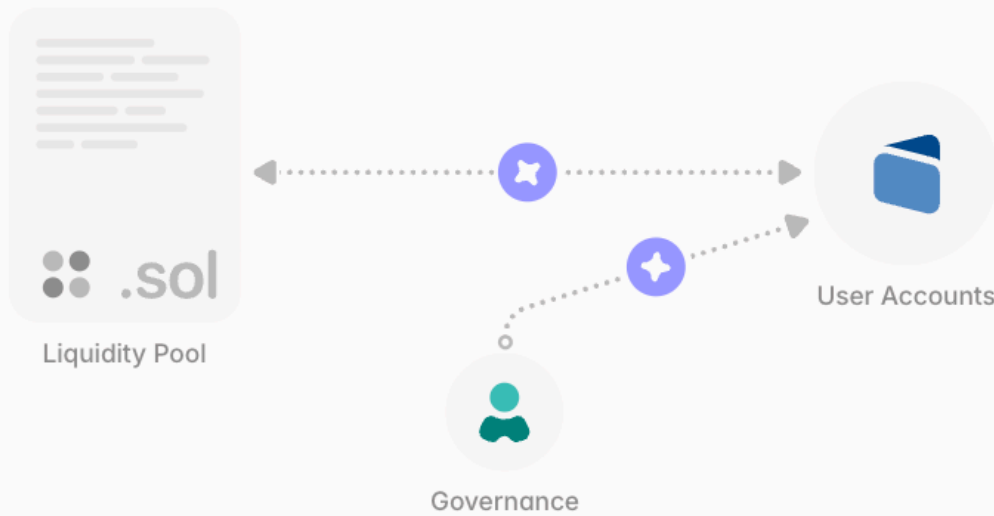
이 유지되도록 한다. 이자율은 기본 금리(base rate)와 활용도에 따른 기울기(slope)를 설정하는 매개변수에 의해 제어된다.

동적 파라미터와 거버넌스(Dynamic Parameters and Governance)



각 리저브의 파라미터는 고정되어 있지 않고, 시장마다 다르며 시간이 지나면서 변할 수 있다. 이는 에이브 거버넌스가 시장 상황을 모니터링하고 그에 맞게 설정을 조정하기 때문이다. 예를 들어, 동일한 기초 자산(e.g. ETH)이라 할지라도 이더리움 시장과 폴리곤 시장에서는 서로 다른 담보 인정 비율이나 이자율을 가질 수 있다. 이러한 조정은 각 시장의 유동성과 리스크 관리를 최적화하기 위해 이루어진다. 거버넌스 제안을 통해 커뮤니티는 차입 한도 상향이나 담보 인정 비율 조정과 같은 변경 사항에 투표할 수 있으며, 이를 통해 현재 시장 역학이 리저브에 지속적으로 반영될 수 있다.

인센티브(Incentives)



에이브 프로토콜의 인센티브는 공급자와 차입자의 적극적인 참여를 장려하여 유동성과 프로토콜 전반의 효율성을 높인다. 다양한 인센티브 이니셔티브가 에이브 다오(Aave DAO) 트레저리(treasury)뿐만 아니라 특정 리저브의 유동성을 촉진 하려는 외부 주체 등 여러 출처에서 비롯될 수 있다는 점을 유의해야 한다.

AAVE 토큰 보유자의 제안과 투표로 운영되는 에이브 다오는 프로토콜 내 특정 활동을 장려하기 위해 다오 트레저리의 자금을 배정할 수 있다. 다오가 자금을 지원하는 인센티브 프로그램은 에이브 거버넌스 프로세스를 통해 제안, 논의, 승인되며, 커뮤니티 참여와 투명성을 보장한다.

유동성 풀 인센티브(Liquidity Pool Incentives)

인센티브는 Aave 유동성 풀의 공급 측이나 차입 측에도 적용될 수 있으며, 인센티브가 부여된 리저브의 활동을 촉진한다. Aave에서 특정 자산의 공급자와 차입자에게 보상을 제공함으로써 토큰의 가시성과 채택을 높일 수 있다. 이러한 외부 인센티브는 거버넌스 승인이 필요하다.

승인된 인센티브는 사용자가 공급하거나 차입한 유동성의 양에 비례하여 시간이 지남에 따라 지속적으로 분배된다. 사용자는 인센티브의 할당과 분배를 관리하는 프로토콜의 인센티브 컨트롤러를 통해 이러한 보상을 청구할 수 있다. 이 시스템은 프로토콜에 적극적으로 참여하는 사용자에게 가치를 제공하는 동시에, 사용자 이해관계를 에이브 생태계의 건전성과 안정성과 정렬시킨다.

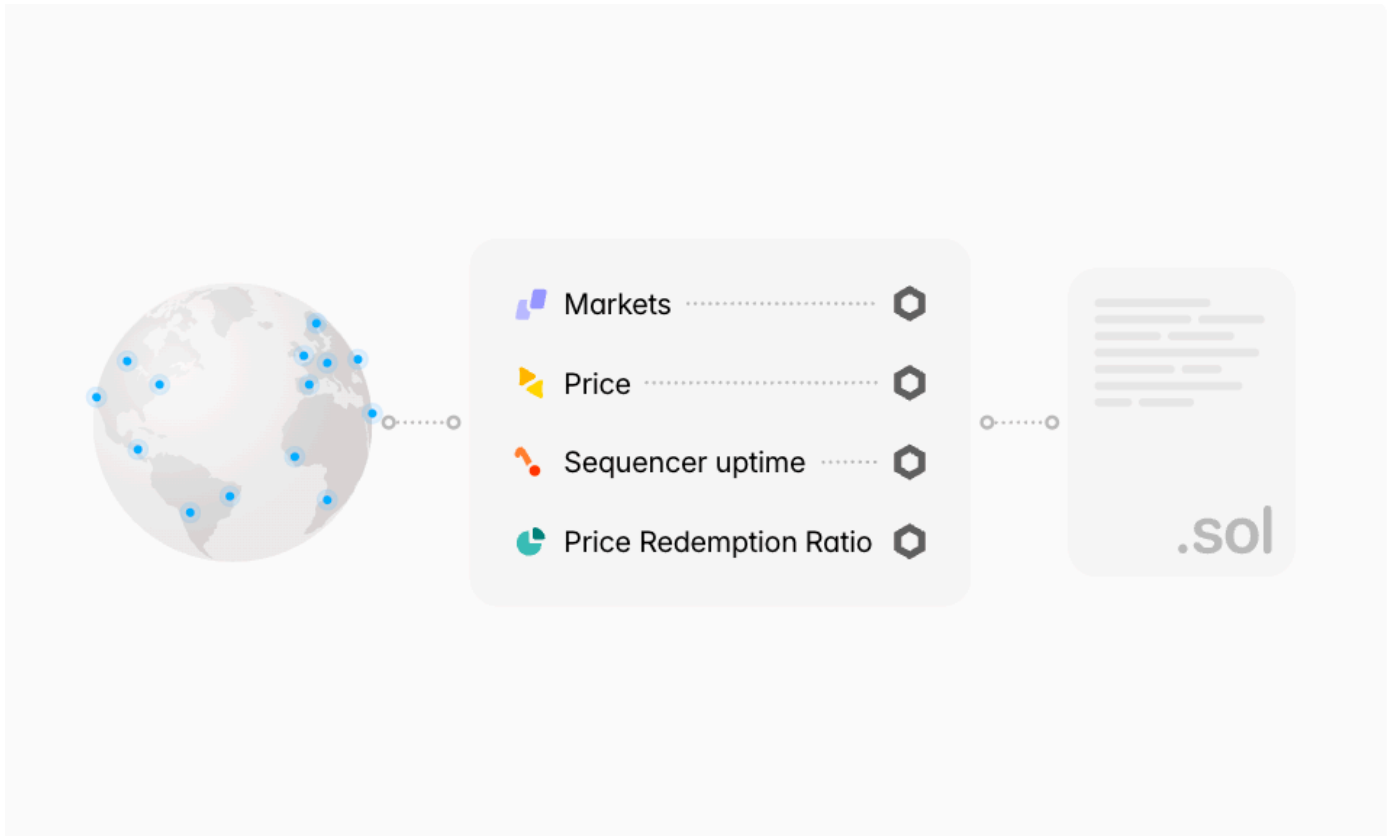
안전 모듈(Safety Module)

AAVE 보유자는 안전 모듈에 토큰을 스테이킹할 수 있으며, 이는 프로토콜을 예상치 못한 유동성 부족으로부터 보호하기 위해 설계된 리저브이다. AAVE 토큰을 스테이킹하고 관련된 리스크를 떠안는 대가로, 참여자는 보상을 받게 되며, 이는 일반적으로 추가적인 AAVE 토큰이나 거버넌스에서 승인된 기타 인센티브 형태로 지급된다. 이러한 보상은 스테이커를 보상할 뿐만 아니라, 잠재적 결손을 충당하기 위해 충분한 유동성이 확보되도록 하여 프로토콜의 보안을 강화한다.

메리트 프로그램(Merit Program)

메리트는 에이브 친화적 사용자 보상 시스템으로, 머클 트리(merkle tree) 기반의 주기적 에어드롭 방식으로 설계되어 에이브에 유익한 행동을 장려하고 에이브 프로토콜의 경쟁력을 강화한다. 메리트 분배에 대한 더 많은 정보와 접근 인터페이스는 [여기](#)에서 확인할 수 있다.

오라클(Oracle)



에이브 프로토콜 내 각 [리저브](#)는 오라클 컨트랙트와 연결되어 있다. 이 오라클 컨트랙트는 프로토콜 내 자산의 시장 가격을 알려주는 역할을 하며, 이는 담보 요건을 결정하는 데 필수적이다.

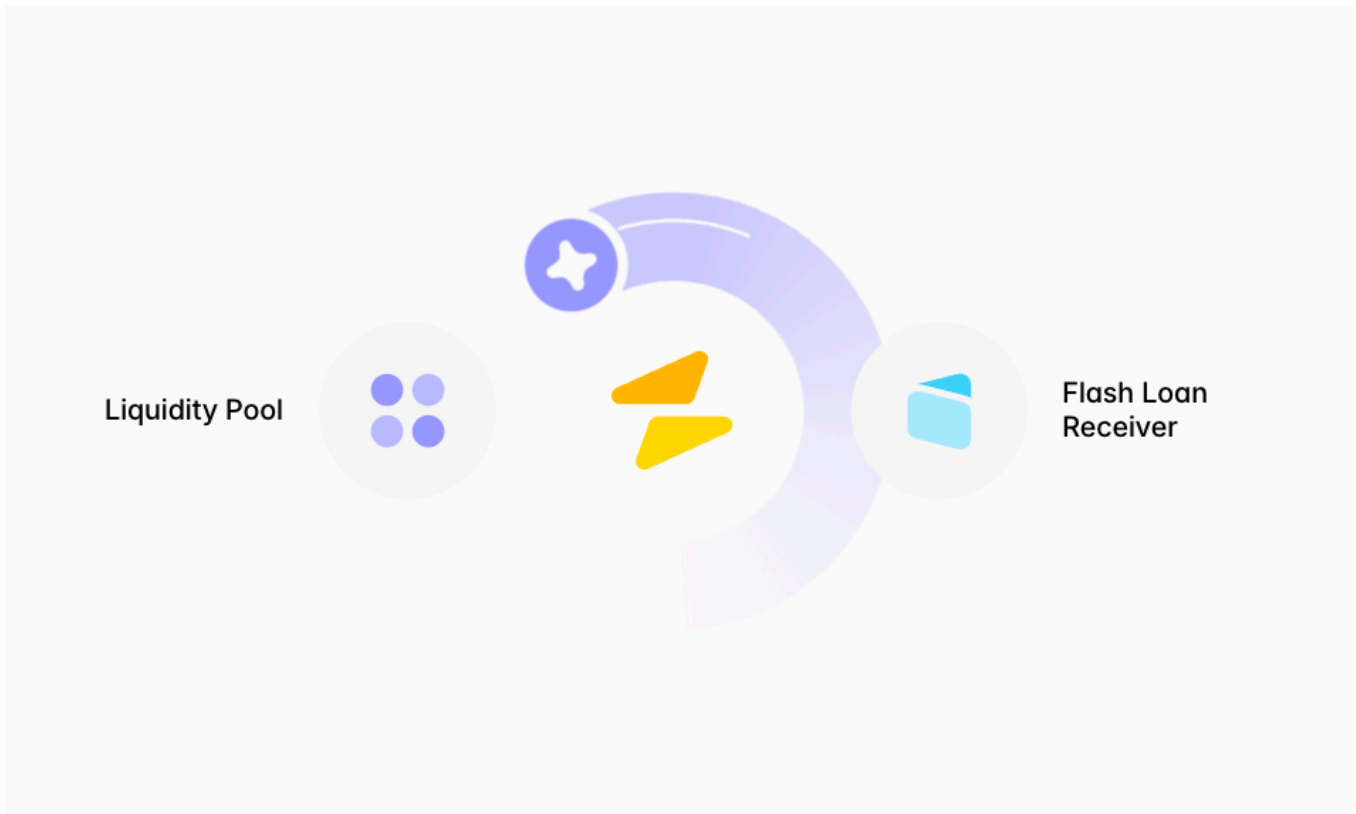
각 리저브의 오라클은 에이브 거버넌스를 통해 선택되며, 커뮤니티가 어떤 데이터 소스를 사용할지 결정하는 탈중앙화 모델을 구축한다. 선택이 이루어지면, 오라클 컨트랙트는 자체 운영 로직(시간 기반, 편차 기반 등)에 따라 가격 피드 업데이트를 제출한다.

사용 중인 오라클 유형

현재 에이브에서 활용되는 주요 오라클 컨트랙트 유형은 두 가지이다:

1. 체인링크(Chainlink) 가격 피드: 체인링크 오라클은 다양한 자산에 대해 신뢰성이 높고 탈중앙화된 가격 데이터를 제공한다. 이 가격 피드는 여러 출처에서 데이터를 수집하고 이를 집계하여 조작이나 장애의 위험을 최소화한다.
2. 상관 자산 가격 오라클(Correlated Assets Price Oracle, CAPO): 상관 자산 가격 오라클은 다른 자산의 가격과 강한 상관관계를 가진 자산을 위해 설계되었다. 예를 들어, 래핑된 토큰은 해당 오라클을 사용하여 기초 자산의 가격을 반영할 수 있다. 상관 자산 가격 오라클은 이러한 상관 자산의 움직임을 따라가는 가격을 조정하고 제출하기 위해 특수화된 로직을 활용한다. 구현에 대한 더 많은 세부 사항은 [관련 깃허브](#)에서 확인할 수 있다.

플래시 론(Flash Loans)



플래시 론은 특정 자산을 빌릴 수 있게 해주는 특별한 트랜잭션으로, 차용한 금액(및 수수료)이 트랜잭션이 끝나기 전에 반환되기만 하면 된다(이를 'One Block Borrows'라고도 한다). 이러한 트랜잭션은 사용자가 담보를 제공하지 않고도 실행할 수 있다. 플래시 론에는 현실 세계의 직접적인 비유가 없으므로, 블록체인에서 블록 단위로 상태가 어떻게 관리되는지에 대한 기본적인 이해가 필요하다.

플래시 론은 개발자를 위한 고급 개념이다. 이를 활용하려면 EVM, 프로그래밍, 스마트 컨트랙트에 대한 충분한 이해가 필요하다.

개요

플래시 론은 트랜잭션 종료 시점까지 사용자가 해당 금액과 수수료를 반환하거나 (허용될 경우) 상응하는 부채 포지션을 개설하는 한 풀의 유동성(대출이 허용된 리저브에 한정)에 접근하여 단일 트랜잭션에서 자금을 사용할 수 있도록 한다.

에이브 V3는 플래시 론을 위해 두 가지 옵션을 제공한다:

- `flashLoan()`: 차용자가 단일 플래시 론 트랜잭션에서 여러 리저브의 유동성에 접근할 수 있도록 한다. 또한, 차용자는 제공한 담보나 크레딧 위임을 바탕으로 변동 금리 대출 포지션을 개설할 수 있다.
참고: 승인된 `flashBorrowers`([ACLManager](#)에 의해 관리됨)의 경우 플래시 론 수수료는 면제된다.
- `flashLoanSimple()`: 차용자가 단일 트랜잭션에서 단일 리저브의 유동성에 접근할 수 있도록 한다. 이 경우 플래시 론 수수료는 면제되지 않으며, 차용자가 트랜잭션 종료 시점에 부채 포지션을 개설할 수도 없다. 단일 리저브 자산에 대한 간단한 플래시 론을 활용하려는 사용자에게 가스 효율적이다.

실행 흐름

개발자가 솔루션을 설계할 때 고려할 수 있는 유용한 사고 모델은 다음과 같다:

1. 당신의 컨트랙트가 Pool 컨트랙트를 호출하여, `flashLoanSimple()` 또는 `flashLoan()`을 통해 특정 `reserve(s)`의 특정 `amount(s)`에 상응하는 플래시 론을 요청한다.
2. 기본적인 검증 후, Pool은 요청된 `amounts`를 당신의 컨트랙트로 전송한 뒤, 수신자 컨트랙트의 `executeOperation()`을 호출한다.
3. 당신의 컨트랙트는 플래시 론된 `amount(s)`를 보유한 상태에서 코드 내 임의의 연산을 실행한다.
 - `flashLoanSimple`을 수행하는 경우, 코드 실행이 끝난 후 플래시 론 금액 + 수수료에 대해 Pool을 승인해야 한다.
 - `flashLoan`을 수행하는 경우, 각 리저브에 대해 자산에 대해 전달된 `interestRateMode`에 따라 Pool이 플래시 론 금액 + 수수료에 대한 승인을 받아야 하거나, 부채 포지션 개선을 위한 충분한 담보나 크레딧이 제공 혹은 위임되어야 한다.
 - 차입 금액을 상환할 수 없는 경우(잔액 부족, 승인 부족, 부채에 대한 담보 부족 등), 트랜잭션은 취소된다.
4. 위 모든 과정은 단일 트랜잭션에서 일어난다(즉, 단일 이더리움 블록 내에서 발생한다).

플래시 론의 활용 사례

에이브의 플래시 론은 이미 에이브 V3에서 유동성 변환 기능에 사용되고 있다. 그 밖의 실제 활용 사례는 다음과 같다:

- 원금을 보유하지 않고도 수행하는 자산 간 차익거래.
- 부채를 직접 상환하지 않고 할인된 담보를 활용하여 플래시 론 금액 + 수수료를 상환하는 차용 포지션 청산.

플래시 론 수수료

플래시 론 수수료는 배포 시 기본값이 0.05%로 설정되며, 거버넌스 투표를 통해 업데이트될 수 있다. 현재 값은 `FLASHLOAN_PREMIUM_TOTAL`로 확인할 수 있다.

플래시 론 수수료는 유동성 공급자와 프로토콜 트레저리 간에 분배될 수 있다. `FLASHLOAN_PREMIUM_TOTAL`은 차용자가 지불하는 총 수수료를 나타내며, 이는 다음과 같이 나뉜다:

- 유동성 공급자 수수료: `FLASHLOAN_PREMIUM_TOTAL - FLASHLOAN_PREMIUM_TO_PROTOCOL`
- 프로토콜 수수료: `FLASHLOAN_PREMIUM_TO_PROTOCOL`

초기화 시점에서 `FLASHLOAN_PREMIUM_TO_PROTOCOL` 값은 0으로 설정된다.

단계별 절차

1. 설정하기

플래시 론 금액을 수신하는 컨트랙트는 반드시 `executeOperation()` 함수를 구현하여 `IFlashLoanSimpleReceiver` 또는 `IFlashLoanReceiver` 인터페이스를 준수해야 한다.

또한, 차용 금액은 컨트랙트에서 *pull* 방식으로 회수되므로, 컨트랙트는 Pool이 플래시 론 금액 + 프리미엄을 회수할 수 있도록 승인해야 한다.

2. `flashLoan()` 또는 `flashLoanSimple()` 호출하기

Pool에서 플래시 론 메서드 중 하나를 호출하려면 관련 매개변수를 전달해야 한다. 이를 수행하는 방법은 세 가지다.

1. EOA(일반 이더리움 계정)에서 호출
EOA를 사용할 경우, 해당 Pool에 트랜잭션을 보내 `flashLoan()` 또는 `flashLoanSimple()` 함수를 호출한다. 이때 매개변수는 Pool 문서를 참고하되, `receiverAddress`에는 1단계에서 준비한 컨트랙트 주소를 사용해야 한다.
2. 다른 컨트랙트에서 호출
EOA에서 호출하는 방식과 유사하되, `receiverAddress`를 1단계에서 준비한 컨트랙트 주소로 설정해야 한다.
3. 같은 컨트랙트에서 호출
1단계에서 사용한 동일한 컨트랙트를 사용하려면, 플래시 론 메서드의 `receiverAddress` 매개변수에 `address(this)`를 사용한다.

⚠ *FlashLoanReceiverBase* 컨트랙트에 자금을 영구적으로 보관하지 않아야 한다. 공격자가 저장된 자금을 악용하는 그리핑(griefing) 공격에 노출될 수 있기 때문이다.

플래시 론 마무리하기

플래시 론 자산을 활용한 로직(`executeOperation()` 함수 내)을 실행한 후, `flashLoanSimple()`을 사용했거나 `flashLoan()`에서 `interestRateModes = 0`으로 지정된 자산이 있다면 해당 자산에 대해 플래시 론 금액을 상환해야 한다.

- 플래시 론 자산 상환하기
컨트랙트가 차입 자산을 상환하기 위한 금액 + 프리미엄을 보유하고 있는지 확인해야 한다. 이는 `executeOperation()` 함수에 전달된 `amounts`와 `premiums` 배열의 해당 항목을 합산하여 계산할 수 있다. 차입금을 직접 Pool로 전송할 필요는 없다. 연산이 종료되면 자금은 자동으로 *pull* 방식으로 회수된다.
- 부채 발생시키기(즉, 즉시 상환하지 않기)
초기 호출 시 `modes` 매개변수에서 특정 자산에 대해 `mode=1` 또는 `mode=2`를 지정한 경우, `onBehalfOf`로 전달된 주소가 부채를 부담하게 된다. 단, 해당 주소가 사전에 `msg.sender`에게 대리 부채 발생을 승인했어야 한다. 이는 일부 자산은 즉시 상환하고, 다른 자산은 부채로 남길 수 있음을 의미한다.

크레딧 위임(Credit Delegation)

크레딧 위임은 공급자가 에이브 프로토콜에 유동성을 제공하여 이자를 얻는 동시에, 차입 권한(즉, 크레딧)을 다른 사용자에게 위임할 수 있도록 한다. 차입 포지션의 집행과 조건은 공급자와 차입자 간에 합의되며, 이는 오프체인 법적 계약을 통해서나 온체인 스마트 컨트랙트를 통해 이루어질 수 있다.

이를 통해 다음이 가능하다:

- 공급자(위임자)는 프로토콜에서 이미 얻고 있는 수익 외에 추가적인 수익을 얻을 수 있다.
- 차입자(수임자)는 무담보 유동성에 접근할 수 있다.

`delegatee`의 차입은 반드시 `delegator`의 `eMode` 카테고리과 일치해야 한다. 예를 들어, 위임자의 `eMode` 카테고리가 `STABLECOINS`라면:

- 위임자는 `STABLECOINS` `eMode` 카테고리에 속한 자산만 차입할 수 있다.
- 만약 위임자가 `delegatee`에게 `STABLECOINS` 카테고리가 아닌 자산(예: `WETH`)에 대한 크레딧을 승인하면,

해당 차입은 취소된다.

delegatee는 delegator의 포지션을 청산할 목적으로 크레딧 승인을 남용할 수 없다. 즉, 차입이 위임자의 포지션을 $HF < HEALTH_FACTOR_LIQUIDATION_THRESHOLD$ 상태로 만든다면, 차입이 이루어지지 않는다.

위임 승인

VariableDebtToken 컨트랙트의 approveDelegation 또는 delegationWithSig 함수가 공급자(위임자)에 의해 호출되어, 특정 금액을 차입자(수임자)에게 승인해야 한다.

이 과정은 위임해야 하는 각 부채 토큰마다 수행되어야 한다.

위임자가 approveDelegation을 위해 반드시 이미 자금을 프로토콜에 예치해 둘 필요는 없다. 그러나 차입자가 borrow를 실행하기 이전에, 위임자는 프로토콜에 충분한 담보를 공급해 두어야 한다.

크레딧 차입

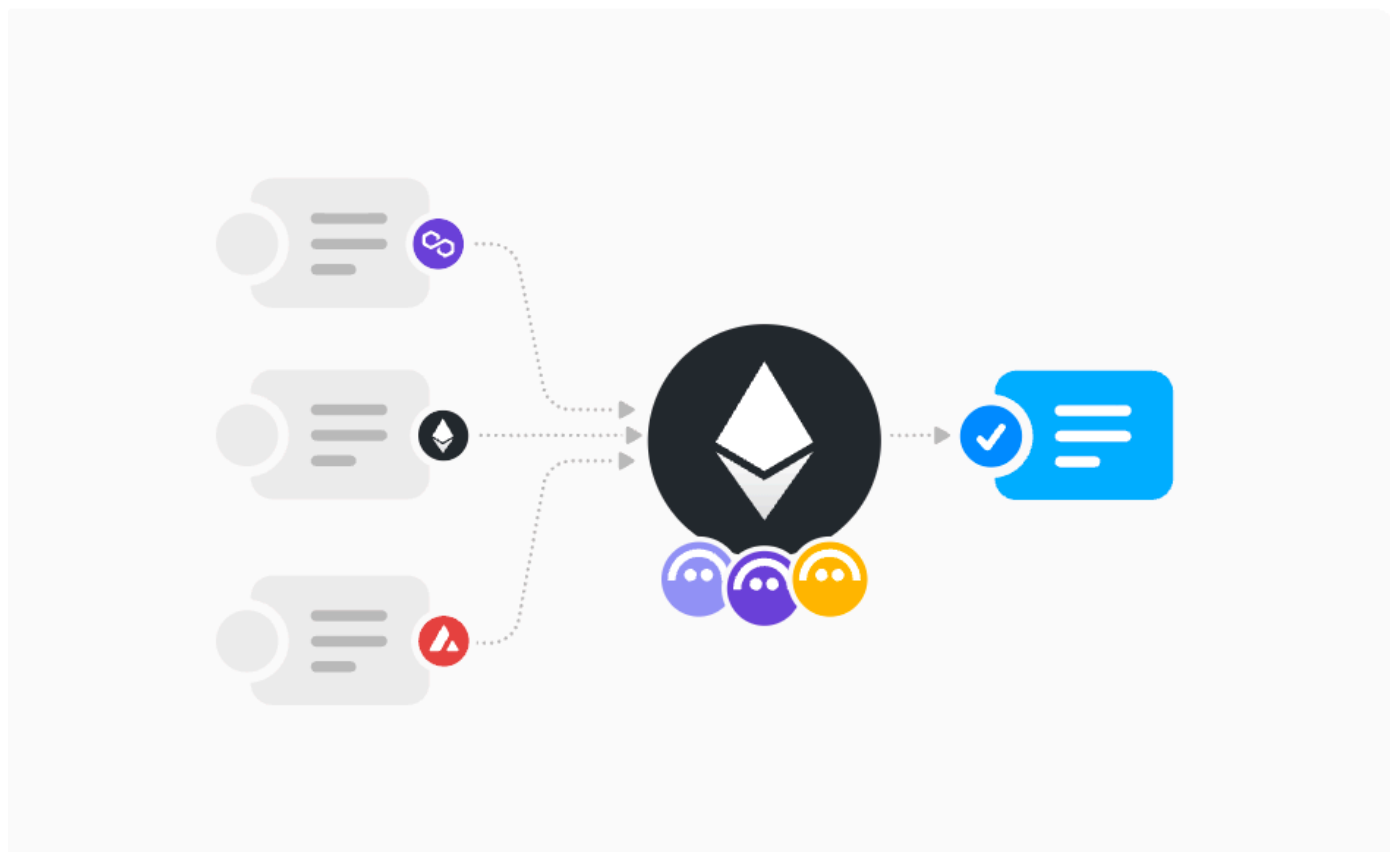
차입자(수임자)는 Pool의 borrow 함수를 호출하면서, 마지막 매개변수 onBehalfOf에 공급자(위임자)의 주소를 사용한다.

차입자의 사용 가능한 크레딧은 차입된 금액만큼 감소한다.

크레딧 상환

누구든지 사용자를 대신(OnBehalf) 하여 차입 포지션을 상환할 수 있으며, 이를 위해 Pool의 repay 또는 repayWithPermit 함수 중 하나를 호출한다. 또한 공급자(채권자)는 동일한 풀에서 해당 기초 자산의 aTokens를 사용해 repayWithATokens 함수를 통해 차입 포지션을 상환할 수도 있다.

거버넌스(Governance)



에이브 다오는 AAVE 토큰 보유자와 기여자로 이루어진 탈중앙화된 집합체로, 구조화된 거버넌스 프로세스를 통해 프로토콜의 미래를 함께 만들어간다. 이 커뮤니티 주도형 모델은 참가자들이 프로토콜 변경 사항을 제안하고, 토론하며, 투표할 수 있도록 하여 프로토콜의 발전을 이끌고, 이러한 과정이 구성원들의 공동 목표와 부합하도록 한다.

에이브 프로토콜은 이더리움 메인넷의 AAVE, stkAAVE, aAAVE 토큰 보유자 커뮤니티에 의해 절차, 투표, 스마트 컨트랙트 실행을 통해 운영되며, 이를 통틀어 에이브 거버넌스라고 한다.

BGD Labs가 개발한 [Aave Governance V3](#)는 이더리움 메인넷에 토큰 잔고를 유지한 채 폴리곤 PoS와 아발란체 C-Chain과 같이 수수료가 낮은 네트워크에서 투표할 수 있는 기능을 도입했다.

구조(Architecture)

에이브 거버넌스 V3는 에이브와 같은 온체인 다오의 포괄적인 요구사항을 충족하기 위해 모듈형 아키텍처로 설계되었다. 시스템은 제안의 생애 주기 단계에 따라 세 가지 주요 구성 요소로 나뉜다:

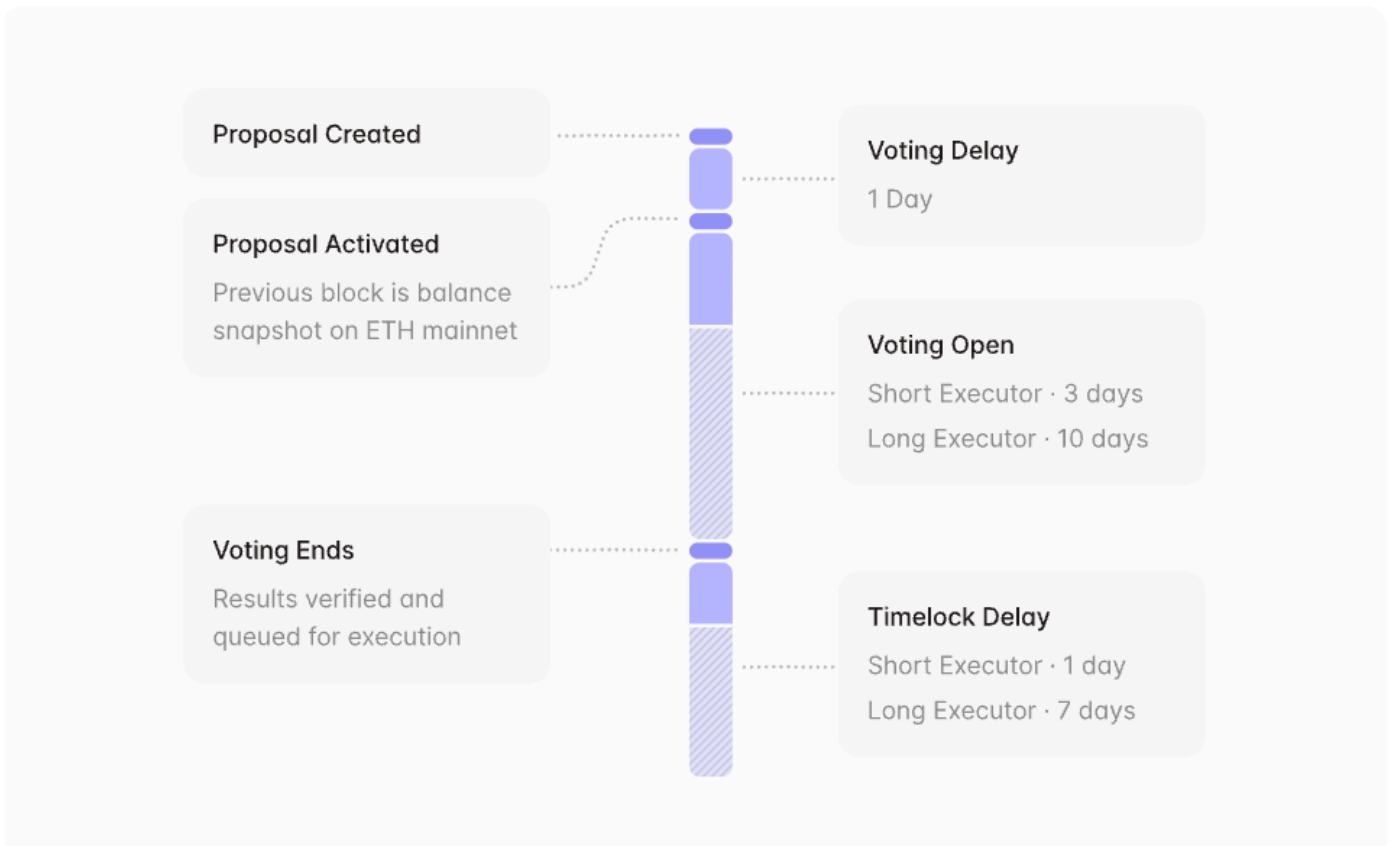
- 코어 네트워크(Core Networks): 투표 권한이 존재하는 정산 및 보안 계층 역할을 한다. 제안에 대한 상위 수준의 검증을 수행하고 거버넌스 토큰 잔고를 관리한다. 이더리움 메인넷이 코어 네트워크 역할을 한다.
- 투표 네트워크(Voting Networks): 스토리지 증명(storage proof)을 기반으로 투표가 이루어지는 환경이다. 사용자가 자금을 코어 네트워크에서 옮기지 않은 채 폴리곤 POS, 아발란체 C-Chain과 같은 저비용 네트워크에서 투표할 수 있도록 한다.
- 실행 네트워크(Execution Networks): 제안 페이로드(payloads)가 등록되고 실행되는 네트워크이다. 승인된 제안의 실행을 다양한 네트워크(모든 투표 네트워크 및 코어 네트워크 포함)에서 처리한다.

이러한 구성 요소 간의 통신은 [Aave Delivery Infrastructure\(a.DI\)](#)에 의해 지원되며, 안전하고 효율적인 크로스체인 상호작용을 제공한다. 또한 [Aave Robot](#)은 제안 대기열 등록 및 실행과 같은 에이브 거버넌스의 무허가(permissionless) 작업을 자동화한다.

프로세스(Processes)

에이브의 거버넌스 프로세스는 프로토콜이 탈중앙화, 보안성, 적응성을 유지하도록 구조화되어 있다. 제안의 생애 주기는 커뮤니티 구성원이 아이디어를 제시하고, 이에 대해 투표하며, 승인된 변경 사항을 투명하고 체계적인 프로세스를 통해 실행할 수 있도록 설계되어 있다.

제안 생애 주기(Proposal Lifecycle)



1. 거버넌스 포럼: 제안 프로세스는 [Aave Governance Forum](#)에 아이디어가 제시되면서 시작된다. 여기서 초기 논의가 이루어지고 피드백이 수집된다. 커뮤니티는 제안이 다음 단계로 넘어가기 전에 이를 구체화하는 데 도움을 준다.
2. 스냅샷 투표를 통한 Temp Check: 논의 이후, 제안은 [Aave Snapshot Space](#)에서 'TEMP CHECK' 단계로 넘어간다. 이 비공식 투표는 커뮤니티의 분위기를 파악하기 위함이다. TEMP CHECK 투표는 구속력이 없지만, 제안이 충분한 관심을 얻고 있는지 판단하는 데 도움이 된다. 이 단계의 제안은 구체적인 기술적 세부사항을 요구하지 않는다.
3. ARFC(Aave Request for Final Comments): TEMP CHECK를 통과하면 제안은 ARFC 단계로 이동하여 더 공식적인 검토를 받는다. AIP 단계로 나아가기 위해 서비스 제공자와 커뮤니티 구성원이 제안이 프로토콜에 미칠 영향에 대해 상세한 피드백을 제공한다. ARFC 투표 역시 [Aave Snapshot Space](#)에서 이루어진다.
4. AIP(Aave Improvement Proposal): AIP 단계는 제안이 공식적인 온체인 제출로 전환되는 단계이다. 이는 메타데이터(IPFS에 저장됨)와 컨트랙트 페이로드, 두 부분으로 구성된다. 이들은 주로 코어 메인넷의 에이브 거버넌스 컨트랙트를 통해 제출된다.
5. 온체인 투표: AIP는 'PENDING' 상태를 거쳐 투표를 위해 'ACTIVE' 상태가 된다. 투표는 에이브 거버넌스 컨트랙트를 통해 온체인에서 진행된다. 제안은 정족수와 기타 차등 요건을 충족하면 승인된다. 이 기준을 충족하지 못하면 제안은 승인되지 않는다.
6. 실행: 승인된 제안은 실행 단계로 이동하며, 에이브의 거버넌스 인프라를 통해 시행된다. 제안 유형에 따라 변경 사항이 적용되기 전 1일 또는 7일의 타임락 지연이 부과된다. 크로스체인 제안은 에이브의 'Delivery Infrastructure(a.DI)'를 사용하여 실행된다.

제안 프레임워크(Proposal Frameworks)

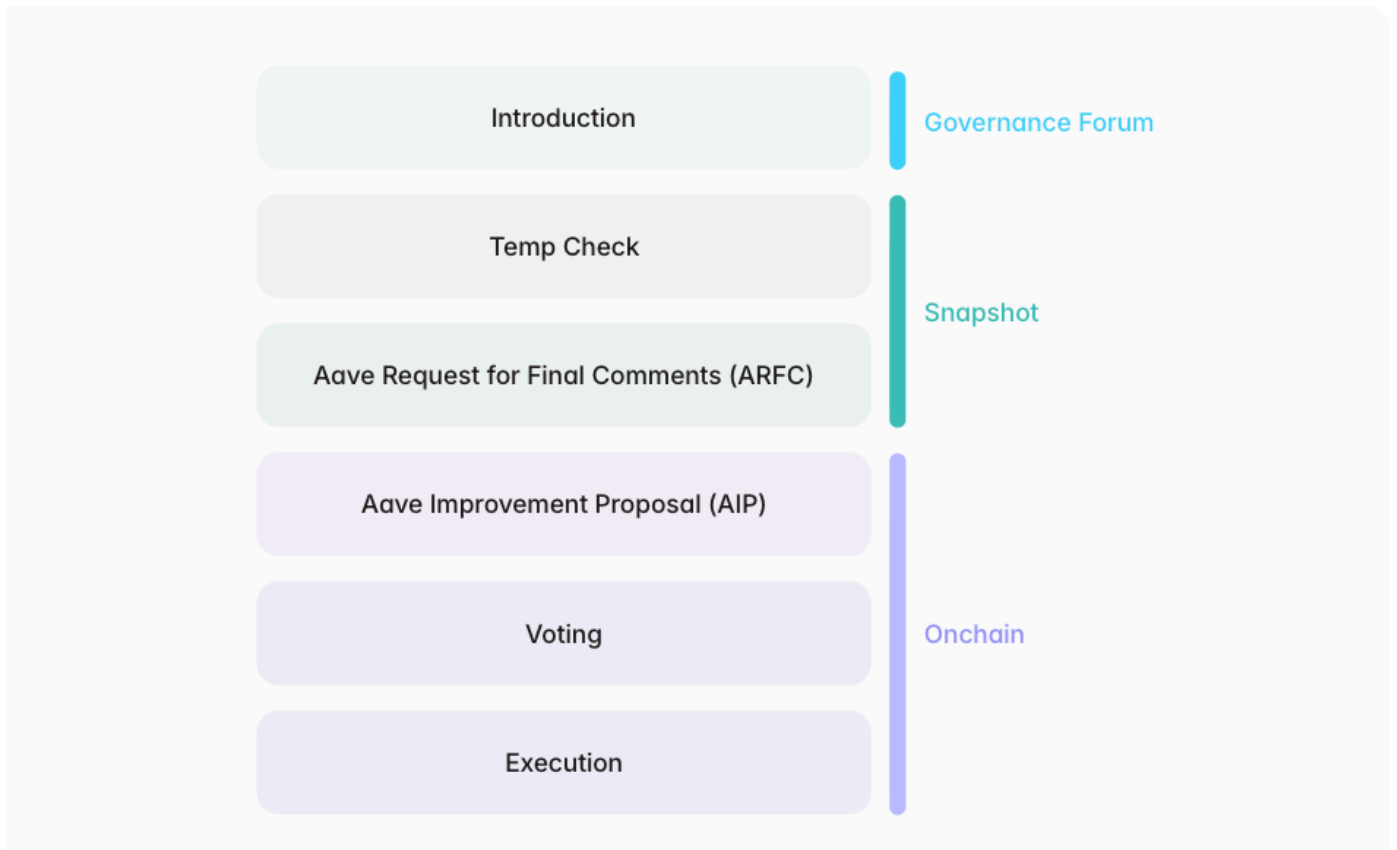
에이브는 일반적인 유형의 제안을 위해 사전에 정의된 프레임워크를 제공하여 거버넌스 프로세스를 단순화한다:

- [자산 온보딩 프레임워크\(Asset Onboarding Framework\)](#): 새로운 자산을 프로토콜에 온보딩하기 위한 표준화된 생애 주기이다. 위험 평가와 커뮤니티 토론을 위한 구조화된 프로세스를 제공한다.

- [신규 체인 배포 프레임워크\(New Chain Deployment Framework\)](#): 에이브를 새로운 블록체인에 배포하기 위한 가이드라인이다. 보안 감사, 유동성 고려사항, 거버넌스 승인을 포함하여 안전하고 효과적인 멀티체인 확장을 지원한다.
- [이미션 매니저 프레임워크\(Emission Manager Framework\)](#): 리저브에 이미션 관리자(emission admin)를 추가하기 위한 단순화된 프로세스로, 유동성 인센티브의 지속적인 관리를 가능하게 한다.
- [한도 업데이트 프레임워크\(Caps Update Framework\)](#): 한도(cap) 조정이나 리저브 동결을 위한 'Direct-to-AIP' 프로세스로, 프로토콜의 안정성과 유동성 제약을 유지하면서 시장 위험에 더 빠르게 대응할 수 있도록 한다.
- [AIP 패스트트랙 프레임워크\(Direct to AIP Framework\)](#): 한도 업데이트 프레임워크와 유사하나 더 광범위하게 적용되며, 특정 거버넌스 활동이 TEMP CHECK 및 ARFC 단계를 건너뛸 수 있도록 허용한다. 이 경우 AIP 생성 전에 ARFC 포럼 게시글만 필요하다.

이러한 프레임워크는 거버넌스를 간소화하고, 커뮤니티가 복잡한 이슈에 집중할 수 있도록 하면서도 효율성과 보안을 유지할 수 있게 한다.

투표(Voting)



오프체인 투표(스냅샷)

오프체인 투표는 제안 초기 단계(Temp Check와 ARFC)에서 커뮤니티의 의견을 수렴하기 위해 사용된다. 이러한 비구속적(non-binding) 투표는 스냅샷에서 3일 동안 지속되는데, 토큰 보유자가 거래 수수료 없이 참여할 수 있도록 한다.

온체인 투표(거버넌스 인터페이스)

제안이 AIP 단계에 도달하면 온체인 투표가 필요하다. 토큰 보유자, 대리인(delegate), 위임자(delegator)는 AAVE, stkAAVE, aAAVE 토큰을 사용하여 제안에 투표한다. 이 과정은 [Aave Labs](#), [BGD Labs](#), [Boardroom](#) 등 다양한 거버넌스 인터페이스를 통해 보안이 보장된다.

거버넌스 V3에서는 각 제안이 특정 투표 네트워크를 지정하며, 해당 네트워크에서 모든 투표가 진행된다. 거버넌스 V3 활성화에 따른 초기 투표 네트워크는 폴리곤 POS, 아발란체 C-Chain, 이더리움 메인넷(백업)이다. 여러 네트워크에서 투표가 가능하기 때문에, 특정 네트워크에만 존재하는 스마트 컨트랙트 지갑에는 제약이 생길 수 있다. 이를 보완하기 위해 거버넌스 인터페이스는 투표 대리인(voting representatives) 설정 기능을 제공하며, 네트워크별로 다른 투표 주소를 지정할 수 있다.

투표자는 자금을 투표 네트워크로 이동할 필요가 없으며, 모든 거버넌스 토큰 잔고와 위임은 여전히 이더리움 메인넷에 저장되고, 스토리지 증명을 통해 투표 네트워크에서 검증된다.

투표는 제안에 대응되는 투표 네트워크에 있는 VotingMachine에서 submitVote() 함수를 호출하여 수행된다.

필요한 스토리지 증명 파라미터를 생성하기 위해 [aave-cli](#)가 개발되었다. [GovernanceV3Helpers](#) Foundry 스크립트를 통해 투표를 진행할 수도 있다.

투표 권한(Voting Power)

AAVE, stkAAVE(프로토콜 안전 모듈에 스테이킹된 AAVE), aAAVE(이더리움 V3 마켓에 예치된 AAVE를 나타내는 aToken) 토큰 보유자는 이더리움 메인넷에 있는 잔고 합계에 비례하여 거버넌스 권한을 부여받는다.

각 거버넌스 토큰에는 두 가지 권한이 있다:

- 제안을 생성할 수 있는 제안 권한(proposal power).
- 활성 제안에 찬반 투표할 수 있는 투표 권한(voting power).

거버넌스 권한은 공동으로든 별도로든 임의의 이더리움 주소에 위임될 수 있다.

투표 대리인(Voting Representatives)

스마트 컨트랙트 지갑은 스토리지 증명 기반 투표를 위한 메시지 서명 기능이 없으며, 모든 네트워크에 존재하지 않을 수도 있다. 이로 인해 크로스체인 투표 컨트랙트와의 상호작용에 어려움이 발생한다.

이를 해결하고 모든 거버넌스 참여자에게 더 큰 유연성을 제공하기 위해, 지갑은 다른 체인에서 대리인 주소를 연결할 수 있다.

대리인을 선택하려면, 이더리움 주소에서 GovernanceCore 스마트 컨트랙트의 updateRepresentativesPerChain() 함수를 호출해야 한다.

대리인 주소는 이더리움 메인넷에서 위임자의 잔고 및 받은 위임을 사용하여 지정된 네트워크에서 투표할 수 있다. 하나의 주소가 여러 다른 주소의 대리인이 될 수도 있다.

스토리지 증명(Storage Proofs)

에이브 거버넌스 V3는 코어 네트워크(이더리움 메인넷)의 블록 해시를 투표 잔고의 정보 출처로 사용하며, 스토리지 증명을 검증의 핵심 메커니즘으로 활용한다.

모든 제안에 대해, 시스템은 activateVote 메서드가 호출되기 직전 블록의 해시를 기준으로 코어 네트워크에서 투표 토큰 잔고/위임 상태의 "스냅샷"을 생성한다. 이 정보는 에이브 투표 네트워크로 전달되며, 해당 네트워크에서 주소가 투표를 제출할 때 검증의 기준으로 사용된다.

이더리움 블록 해시는 네트워크의 상태 트리를 포함하며, 여기에는 해당 시점의 모든 스마트 컨트랙트 데이터가 담겨 있다. 그 중 AAVE, stkAAVE, 기타 투표 자산 컨트랙트가 포함되며, 이 안에는 모든 이더리움 주소의 잔고와 위임 정보가 들어 있다.

스토리지 증명은 특정 데이터가 트리 내에 존재함을 암호학적으로 증명하는 메커니즘이다. 이 경우, 특정 주소가 이더리움 블록의 상태 트리에서 투표 및 위임 권한을 가지고 있음을 증명할 수 있다.

실제로 온체인 투표가 이루어질 때는, 투표자가 제안의 이더리움 블록 시점에서 보유한 투표 자산의 잔고와 해당 블록 해시에 대한 스토리지 증명을 함께 제출한다. 이후 투표 네트워크는 이 스토리지 증명을 이더리움 블록 해시와 대조해 암호학적으로 검증하고, 추가적인 검증(예: 이중 투표 방지)을 거쳐 결과를 저장한다.

제안 생애 주기(Proposal Lifecycle)

1. 페이로드 등록(Payload Registration): 실행 가능한 로직을 포함하는 제안 페이로드는 대상 실행 네트워크(Execution Network)의 `PayloadsController`에 배포 및 등록된다. 이 과정은 무허가(permissionless) 방식이다.
2. 제안 생성(Proposal Creation): 충분한 제안 권한을 가진 제안자가 Governance 컨트랙트와 상호작용하여 코어 네트워크(이더리움 메인넷)에서 제안을 생성한다. 제안자는 페이로드 ID와 대상 네트워크를 지정한다.
3. 제안 활성화(Proposal Activation): 일정 기간(`cooldownBeforeVotingStart`) 이후 제안이 활성화된다. 이 과정에서는 코어 네트워크 상태의 스냅샷을 생성하고 a.DI를 통해 지정된 투표 네트워크로 전달한다.
4. 투표 준비(Voting Setup): 투표 네트워크에서 필요한 데이터(이더리움 블록 해시, 상태 트리, 투표 자산 루트)가 DataWarehouse 컨트랙트에 저장된다. 이 단계는 무허가 방식이며, 누구나 수행할 수 있다.
5. 투표 기간(Voting Period): 투표는 투표 네트워크에서 진행된다. 투표자는 스냅샷 블록에서의 토큰 잔고와 스토리지 증명을 함께 제출하여 VotingMachine의 `submitVote()`를 호출한다.
6. 투표 종료(Vote Closure): 투표 기간(`votingDuration`)이 끝나면 투표가 종료된다. VotingMachine 컨트랙트는 투표 결과(찬성 및 반대 수)를 a.DI를 통해 코어 네트워크로 전송한다.
7. 결과 검증(Result Validation): 코어 네트워크의 Governance 컨트랙트가 최소 기준 및 투표 차등과 같은 승인 지표에 따라 투표 결과를 검증한다.
8. 제안 대기열 등록(Proposal Queuing): 제안이 승인 기준을 충족하면 실행 네트워크의 `PayloadsController`로 전달되어 타임락 메커니즘에 대기열로 등록된다.
9. 제안 실행(Proposal Execution): 타임락이 만료된 후, 누구나 `PayloadsController`의 `executePayload()`를 호출하여 제안을 실행할 수 있으며, 이는 적절한 Executor로 전달되어 실행된다.

커뮤니티(Community)

대리인(Delegates)

대리인은 다른 커뮤니티 구성원 또는 자기 위임을 통해 투표 권한을 위임받는다. 이들은 위임받은 사람을 대신하여 제안에 투표함으로써 거버넌스에 적극적으로 참여한다. 일부 대리인은 오빗(Orbit) 프로그램을 통해 기여에 대한 보상을 받는다.

위임자(Delegators)

위임자는 AAVE, stkAAVE, aAAVE 토큰을 보유하지만, 자신의 투표 권한을 다른 개인이나 단체에 위임하기로 선택한 커뮤니티 구성원이다. 이를 통해 직접 모든 투표에 참여하지 않고도 자신의 이해가 거버넌스에 반영되게끔 할 수 있다.

기여자(Contributors)

기여자는 작업 그룹 참여, 바운티 수행, 프로토콜 위 개발, 또는 보조금(grant)을 통해 에이브 다오에 시간과 자원을 투자한다. 이러한 노력은 에이브 생태계를 유지하고 개선하는 데 기여한다.

서비스 제공자(Service Providers)

서비스 제공자는 프로토콜에 필수적인 서비스를 제공하며, 여기에는 리스크 관리, 재무 감독, 보안, 개발 등이 포함된다. 대표적인 서비스 제공자는 다음과 같다:

- Chaos Labs(리스크)
- Lllamarisk(리스크)
- Karpatkey(재무)
- Certora(보안)
- Tokenlogic(재무)
- BGD Labs(개발)
- ACI(성장 및 사업 개발)
- Aave Labs(개발)

가디언 (Guardians)

에이브 커뮤니티 가디언은 커뮤니티가 선출한 서명자 그룹으로, 제한적인 긴급 보호 조치를 실행할 권한을 가진다.

에이브 가디언은 두 개의 다중 서명 지갑으로 역할이 나뉘며, 역할과 서명자는 아래와 같다.

프로토콜 긴급 가디언(Protocol Emergency Guardian)

프로토콜 긴급 가디언은 에이브 프로토콜 마켓에 대한 EMERGENCY__ADMIN 권한을 가지며, 비상 모드에서 크로스체인 메시지를 위한 안전 장치를 실행한다.

프로토콜 긴급 가디언은 5/9 다중 서명 지갑이며, 다음의 커뮤니티 선출 서명자로 구성된다:

- Chaos Labs(리스크 서비스 제공자)
- Lllamarisk(리스크 서비스 제공자)
- Karpatkey(재무 서비스 제공자)
- Certora(보안 서비스 제공자)
- Tokenlogic(재무 서비스 제공자)
- BGD Labs(개발 서비스 제공자)
- ACI(성장 및 사업 개발 서비스 제공자)
- Erz3al(에이브 다오 대리인)
- Stable Labs(에이브 다오 대리인)

거버넌스 긴급 가디언(Governance Emergency Guardian)

거버넌스 긴급 가디언은 잠재적 거버넌스 장악으로부터 에이브 프로토콜을 보호하기 위해, 악의적인 온체인 페이로드가 발견될 경우 이를 “거부(veto)”할 수 있는 권한을 가진다.

거버넌스 긴급 가디언은 5/9 다중 서명 지갑이며, 다음의 커뮤니티 선출 서명자로 구성된다:

- Seb(재퍼, Zapper)
- Mounir(파라스왑, Paraswap)
- Gavi Galloway(스탠다드 크립토, Standard Crypto)
- Nenad(디파이 세이버, Defi Saver)
- Fernando(밸런서, Balancer)
- Roger(체인링크 커뮤니티)
- Mariano Conti(디파이 OG)
- Marin(리도, Lido)
- Certora(보안 서비스 제공자)

스튜어드(Stewards)

스튜어드는 특정 프로토콜 파라미터에 대한 책임을 위임받아, 다오가 시장 변화에 신속하게 대응할 수 있도록 한다. 스튜어드는 GHO 스테이블코인과 유동성 파라미터와 같은 영역을 관리한다. 이 시스템은 사소한 조정에 대한 잦은 투표 필요성을 줄여 효율성을 높인다. GHO 스튜어드 컨트랙트의 소스 코드는 [GitHub](#)에서 확인할 수 있다.

GHO 버킷 스튜어드(GHO Bucket Steward)

파라미터	설명
Facilitator Bucket Capacity	최대 100% 증가

GHO 에이브 스튜어드(GHO Aave Steward)

파라미터	설명
GHO Borrow Cap	최대 100% 증가 또는 감소
GHO Borrow Rate	최대 25% 이내에서 optimalUsageRatio, baseVariableBorrowRate, variableRateSlope1, variableRateSlope2를 최대 5% 변경
GHO Supply Cap	최대 100% 증가

GHO CCIP 스튜어드(GHO CCIP Steward)

파라미터	설명
Bridge Limit	최대 100% 증가 또는 감소
Rate Limit	최대 100% 증가 또는 감소

GHO 안정성 모듈 스튜어드(GHO Stability Module Steward)

파라미터	설명
GSM Exposure Cap	최대 100% 증가
GSM Buy/Sell Fees	최대 0.5% 증가 또는 감소

리스크 스튜어드(Risk Steward)

설명	값
변경 주기	5일
최대 공급 한도 증가	50%
최대 차입 한도 증가	50%

엄브렐라(Umbrella)



에이브의 엄브렐라는 에이브 v3 풀의 불량 부채(bad debt) 커버리지를 자동화하는 모듈형 온체인 리스크 관리 시스템이다. BGD Labs에 의해 개발되고 에이브 거버넌스의 승인을 받은 엄브렐라는 거버넌스 개입 없이 프로토콜을 보호할 수 있는 보다 자본 효율적이고 자동화된 방식을 도입한다.

엄브렐라는 사용자가 aTokens(aUSDC, aUSDT, aWETH 등 예치된 토큰)와 GHO(에이브의 네이티브 스테이블코인), 이렇게 두 가지 유형의 자산을 스테이킹할 수 있도록 한다. 이와 같이 프로토콜 자산을 스테이킹함으로써 사용자는 리스크 관리에 적극적으로 기여하고 보상을 얻는다.

엄브렐라는 기존 안전 모듈을 자동화된 스테이킹 시스템으로 대체함으로써 에이브 프로토콜의 회복력을 강화한다. 특정 자산에서 결손이 발생할 경우 엄브렐라는 해당 스테이킹된 자산을 소각하여 부채를 상쇄할 수 있으며, 이 과정에서 거버넌스 결정이나 수동 개입이 필요하지 않다. 이를 통해 프로토콜과 사용자에게 더 효율적이고, 반응적이며, 예측 가능한 안전 메커니즘을 제공한다.

엄브렐라는 [Aave Labs 인터페이스](#) 또는 [BGD Labs 인터페이스](#)를 통해 이용할 수 있다.

주요 이점(Key Benefits)

자동화된 커버리지: 온체인 결손 데이터를 기반으로 직접 대응하여 불량 부채 해결을 위한 거버넌스 의존을 제거한다.

aToken 스테이킹: 사용자가 이미 보유한 aToken을 스테이킹하여, 기초 수익을 계속 얻으면서 리스크 관리에도 참여할 수 있다.

효율성 향상: aToken은 결손 커버리지를 위한 가장 자본 효율적인 자산이다. 예를 들어, aUSDT 소각은 시장 매각 없이 USDT 부채를 직접 커버한다.

보안 강화: 자동화 메커니즘을 통해 커버리지를 차입 자산과 직접 연결함으로써 더 회복력 있는 프로토콜을 구축한다.

보상 및 안전 인센티브(Rewards and Safety Incentives)

엄브렐라에서는 aToken을 스테이킹한 사용자가 예치 자산에 대한 수익을 계속 얻는 동시에, 프로토콜을 보호한 대가로 추가 보상을 받는다. aToken 스테이킹(USDC, USDT, WETH)의 경우 이중 수익 흐름을 얻을 수 있다. 기본이 되는 aToken 수익은 자동으로 스테이킹 포지션에 누적되며, 추가적인 'Safety Incentives'는 삭감 위험을 감수한 대가로 지급된다.

GHO 스테이킹의 경우, aToken과 달리 기초 수익이 발생하지 않기 때문에 직접 Safety Incentives를 얻게 된다. 보상은 거버넌스 설정에 따라 AAVE, GHO, USDC 등 다양한 토큰으로 지급될 수 있다. 자동으로 누적되는 aToken 수익과 달리, Safety Incentives는 블록체인 트랜잭션을 통해 직접 청구해야 한다.

보상 시스템은 수학적으로 모델링된 이미션 커브(Emission Curve)를 사용하여 최적 스테이킹 수준을 목표로 한다. 시스템은 목표 스테이킹 자산량을 정의하고, 총 스테이킹이 정확히 목표에 도달하면 최대 보상을 할당한다. 총 스테이킹이 목표에 미달하면 참여를 유도하기 위해 보상이 비례적으로 증가하며, 목표를 초과하면 과도한 스테이킹을 방지하기 위해 보상이 감소한다.

이 메커니즘은 극단적인 APY 변동을 방지하는 균형을 제공한다. 일반적으로 최소 자산만 스테이킹될 경우의 최대 APY는 목표 유동성에서의 APY의 두 배이지만, 참여 증가 시 사라지는 과도한 세 자릿수 APY 시나리오는 방지한다.

삭감 리스크와 결손 보호(Slashing Risk and Deficit Protection)

보상을 얻는 대가로, 스테이커는 자신이 스테이킹한 특정 풀과 자산에서 결손이 발생할 경우 삭감 위험을 감수해야 한다. 스테이킹 리스크는 스테이킹된 aToken의 특정 자산과 네트워크에 국한된다. 예를 들어, aUSDC 스테이킹은 USDC 부채만 커버한다.

시스템에는 초기 손실 보호(first-loss protection)를 제공하는 중요한 결손 상쇄 메커니즘이 포함된다. 예를 들어, USDT 스테이킹은 100,000 USDT 상쇄 한도를 가지므로, 에이브 다오가 첫 100,000 USDT의 부채를 커버한 이후에만 스테이커 자산이 영향을 받는다. 극단적 상황에서 스테이킹 자산 전부가 삭감될 수 있지만, 이러한 상쇄 메커니즘은 일반적인 결손 상황에서 삭감 가능성을 크게 줄인다.

예를 들어, 에이브 v3.3 운영 첫 달 동안 모든 풀에서 약 \$400 규모의 총 결손이 발생했으며, 이는 약 95억 달러의 미상환 차입에 대비한 수치이다. 이는 미상환 차입의 약 0.000004%에 해당한다.

네트워크 커버리지와 초기 활성화 (Network Coverage and Initial Activation)

초기 활성화는 이더리움에서 시작되며, USDC, USDT, WETH, GH0와 같은 차입 수요가 높은 자산으로 확장된다. 보상율과 목표 유동성 수준을 포함한 활성화 파라미터는 에이브 다오에 의해 설정되며, 지속적인 관리는 에이브 재무 위원회 (Aave Finance Committee)에 위임된다.

각 엠브렐라 배포는 스테이킹된 특정 자산과 네트워크만 보호하므로, 정밀한 리스크 분리가 가능하다. 시스템은 기존 안전 모듈과 동일하게 20일 쿨다운(cooldown) 기간과 2일 출금 과정을 유지한다.

기존 안전 모듈에서의 전환 (Transition from the Legacy Safety Module)

엠브렐라는 점진적으로 기존 안전 모듈을 대체한다. 현재 참여자는 스테이킹된 자산에 따라 서로 다른 마이그레이션 경로를 가진다.

stkAAVE와 stkABPT는 전환 기간 동안 활성 상태를 유지하며, 엠브렐라가 충분한 규모에 도달하면 삭감이 비활성화된다. 이러한 포지션에 대해 즉각적인 조치는 필요하지 않다.

스테이킹된 GH0(stkGH0)는 쿨다운 기간과 삭감 위험을 제거한 새로운 파라미터를 가진 'savings GH0(sGH0)'로 전환된다. 이를 통해 사용자는 에이브 스테이킹 자산에 일반적으로 수반되는 리스크 없이 GH0에 대한 수익을 얻을 수 있다. 증가하는 삭감 리스크에도 불구하고 더 높은 보상을 원할 경우, 사용자는 레거시 stkGH0에서 쿨다운을 활성화하고 GH0를 언스테이킹한 후, 이를 새로운 엠브렐라 시스템의 stkGH0로 스테이킹할 수 있다.

이를 통해 차입하지 않는 공급자도 리스크 관리에 참여하면서 보상을 얻을 수 있는데, 이는 프로토콜 보안에 대한 참여를 확대하는 한편 인센티브 구조가 프로토콜 건전성에 부합하도록 한다.

더 많은 질문은 [Umbrella FAQ](#)에서 확인할 수 있다.

구조(Architecture)

엠브렐라는 세 가지 주요 컨트랙트 유형으로 구성된다:

- UmbrellaCore: 각 에이브 v3 풀의 결손 모니터링, 슬래싱, 자산 커버리지를 조율한다.
- StakeToken: 자산 및 네트워크별 ERC4626 Vault로, 스테이킹된 자산을 보관하고, 지분을 발행하며, 보상 컨트롤러와 통합된다.
- RewardsController: 다중 토큰 보상, 발행 곡선, 사용자 보상 회계를 관리한다.
- UmbrellaBatchHelper: 여러 작업을 단일 트랜잭션으로 묶을 수 있도록 하는 보조 컨트랙트이다.

스테이킹(Staking)

엠브렐라에 참여하려면 사용자는 지원되는 자산(aToken 또는 GH0)을 해당 StakeToken 컨트랙트에 예치해야 한다. 이 작업은 사용자의 포지션을 나타내는 지분을 발행한다. StakeToken은 1:1에서 시작하는 환율을 사용하며, 삭감이 발생할 경우에만 감소한다.

스테이킹은 표준 ERC20 approve와 deposit으로 수행할 수 있으며, EIP-2612 허가 서명 (depositWithPermit)으로도 가능하다. 사용자는 쿨다운을 관리할 '쿨다운 운영자'를 설정할 수도 있다.

```
IERC20 underlying = IERC20(stakeToken.asset());
underlying.approve(address(stakeToken), amount);
stakeToken.deposit(amount, msg.sender);`
```

언스테이킹(Unstaking)

출금하려면 사용자가 먼저 `cooldown()` 을 호출하여 쿨다운을 활성화해야 한다. 쿨다운 기간은 20일(거버넌스에 의해 설정 가능)이며, 그 후 2일간의 출금 기간이 존재한다. 쿨다운 동안에도 보상은 계속 누적되며, 자산은 여전히 삭감 대상이 될 수 있다. 사용자당 동시에 하나의 쿨다운 기록만 허용된다. 사용자가 쿨다운을 활성화한 후 지분을 전송하면 출금 가능한 금액이 줄어든다. 쿨다운 이후 추가 입금은 쿨다운된 금액에 영향을 주지 않는다.

쿨다운 기간이 끝나면, 사용자는 출금 기간 내에서 `redeem` 또는 `withdraw`를 사용해 자산을 출금할 수 있다. 만약 출금 기간을 놓치면 쿨다운을 다시 시작해야 한다. 출금은 항상 현재 환율을 기준으로 하며, 이는 삭감으로 인해 변동되었을 수 있다.

```
stakeToken.cooldown();  
// ...wait for cooldown period...  
stakeToken.redeem(shares, msg.sender, msg.sender);
```

삭감(Slashing)

삭감은 해당 에이브 풀에서 발생한 결손이 설정된 상쇄 한도를 초과할 때 'UmbrellaCore'에 의해 자동으로 발동된다. 삭감은 StakeToken의 총 자산을 감소시켜 모든 지분의 가치를 낮춘다. 컨트랙트는 전체 자산이 고갈되는 것을 방지하기 위해 최소 자산 한도를 강제한다. 오직 UmbrellaCore(소유자)만이 `slash`를 호출할 수 있으며, 삭감된 자산은 결손 보전을 위해 Aave Collector로 전송된다.

```
function slash(uint256 amount) external onlyOwner;
```

보상(Rewards)

보상은 RewardsController에 의해 관리되며, StakeToken당 최대 8개의 보상 토큰을 지원한다. 보상은 구간별 선형 곡선(piecewise linear curve)에 의해 조정되며, 이는 최적의 유동성을 목표로 한다. 목표 유동성보다 낮으면 스테이커를 유치하기 위해 보상이 증가해 목표 지점에서는 최대화된다. 목표치를 초과하면 과도한 스테이킹을 억제하기 위해 보상이 점진적으로 감소한다. 거버넌스는 새로운 자산과 보상을 설정하며, Rewards Admin은 보상률과 분배 종료 시점을 조정할 수 있다.

사용자는 언제든지 자신의 자산에 대해 모든 이용 가능한 리워드를 청구할 수 있으며, 또는 대리 청구자를 지정하여 대신 청구하도록 할 수 있다.

```
rewardsController.claimAllRewards([address(stakeToken)], msg.sender);
```

특정 보상을 청구하려면:

```
rewardsController.claimRewards(  
    [address(stakeToken)],  
    amount,  
    msg.sender,  
    rewardToken  
);
```

대리 청구자를 설정하려면:

```
rewardsController.setClaimer(user, claimer);
```

호라이즌(Horizon)

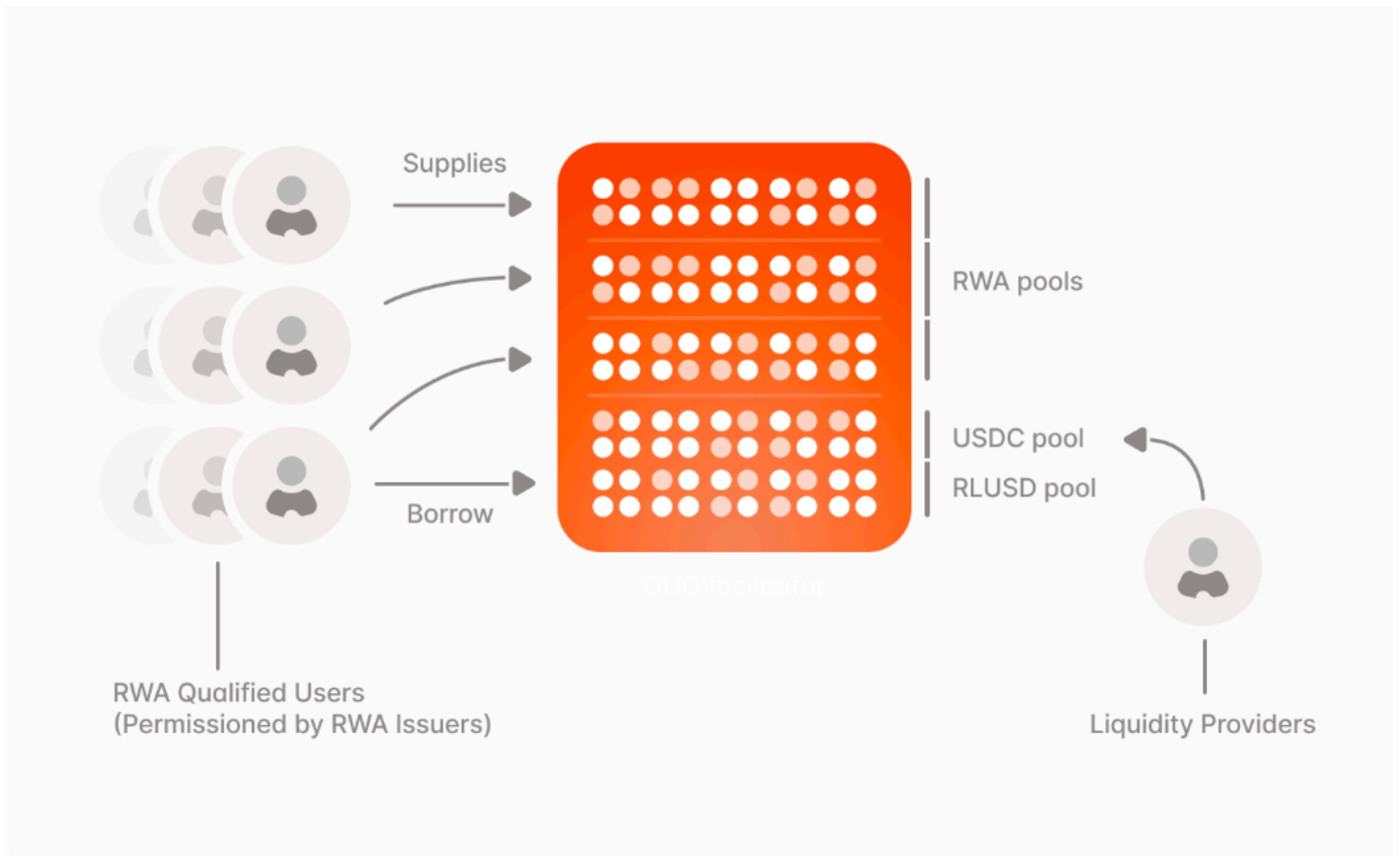


호라이즌은 기관 및 적격 참여자에 맞게 설계된 실물자산(RWAs) 마켓플레이스를 구축하기 위해 에이브 랩스(Aave Labs)가 추진하는 이니셔티브다. 이는 RWA 상품과 관련된 규제 및 컴플라이언스 요구사항을 충족하기 위해 v3.3을 포크하여 라이선스를 가진 별도의 에이브 프로토콜 인스턴스로 운영된다. 호라이즌 마켓의 핵심 기능은 사용자가 허가 기반의 토큰화된 RWA를 담보로 제공하면서 동시에 무허가형 스테이블코인을 대출받을 수 있도록 하는 것이다. 이러한 구조는 기관이 규제 준수를 유지하면서도 디파이의 자본 효율성과 유동성에 접근할 수 있게 한다.

호라이즌 RWA 마켓

호라이즌 RWA 마켓은 기관 금융과 디파이의 무허가 속성을 연결하기 위해 설계된 특수한 유동성 프로토콜이다. 이 구조는 다양한 자산 유형과 사용자 역할을 명확히 분리하여 컴플라이언스를 유지하면서 자본 효율성을 극대화한다. 이러한 분리는 전통 금융 기관이 규제 의무를 훼손하지 않고 디파이에 참여할 수 있는 안전한 환경을 만든다.

이 마켓은 토큰화된 실물자산(RWAs)을 담보로 사용하는 것을 지원한다. 이 부분은 엄격한 허가 기반으로, RWA 발행자에 의해 화이트리스트에 오른 적격 사용자에게만 한정된다. 컴플라이언스는 RWA 토큰의 네이티브 스마트 컨트랙트를 통해 자산 단위에서 집행되며, 승인된 참여자만이 규제 자산과 상호작용할 수 있도록 보장한다. 반대로, USDC, RLUSD, GHO와 같은 스테이블코인에 대한 유동성 풀은 무허가 방식으로 통합되어, 유동성 공급자에게 접근 제한 없이 운영된다.



호라이즌 RWA 시장의 핵심 혁신은 이 두 영역의 상호작용 방식에 있다. 즉, 무허가 스테이블코인 풀에서 대출할 수 있는 권한은 허가 기반의 RWA 담보를 제공한 사용자에게만 부여된다. 이 모델은 규제된 자산을 활용하여 디파이 생태계의 유동성에 접근할 수 있도록 전략적으로 설계되었다. 결과적으로, 전통 금융의 보안성과 컴플라이언스를 유지하면서도 탈중앙화된 프로토콜의 효율성을 활용할 수 있는 독특한 하이브리드 시스템을 구현한다.

통합(Integrations)

호라이즌은 에이브 API 및 SDK 메서드와 완벽히 호환된다. 애플리케이션에 에이브를 프로그래밍 방식으로 통합하려면 [getting started](#) 가이드를 참고하고, 아래 쿼리 예시처럼 Horizon Pool 주소를 지정하면 된다.

```
import { AaveClient, chainId, evmAddress } from "@aave/client";
import { market } from "@aave/client/actions";

const client = AaveClient.create();

// Query Horizon market data
const horizonMarketData = await market(client, {
  address: evmAddress("0xAe05Cd22df81871bc7cC2a04BeCfb516bFe332C8"), // Horizon
  chainId: chainId(1),
});
```

RWA 담보 공급(Supplying RWA Collateral)

호라이즌 시장 참여는 토큰화된 실물자산(RWAs)을 담보로 제공하는 것에서 시작된다. 이 과정은 특정 RWA 발행자에 의해 적격 심사를 거쳐 화이트리스트에 등록된 사용자에게만 허용된다. 적격 심사는 일반적으로 기관 온보딩 절차를 포함하며, 신원 확인, 컴플라이언스 점검, 관련 약관 동의가 포함된다.

공급된 RWA는 오직 담보 용도로만 지정되며 다른 참여자가 대출할 수 없다. 사용자가 자산을 공급하면 담보 포지션을 나타내는 aToken을 받게 된다. 이 시장의 중요한 특징은 해당 수취 토큰이 양도 불가능하다는 점이다. 이는 담보 포지션이 원래 공급자에게만 유지되도록 강제하여 보안성과 컴플라이언스를 강화한다. 이러한 비양도성은 담보 포지션의 2차 시장 거래를 방지하며, 권한 시스템의 무결성을 유지한다.

담보 가치는 각 RWA 유형의 특성을 반영하는 정립된 가격 메커니즘을 통해 산출된다. 담보 인정 비율과 청산 한도를 포함한 리스크 매개변수는 기초 자산의 위험 프로필과 규제 요구사항에 맞게 설정된다.

스테이블코인 대출(Borrowing Stablecoins)

호라이즌 시장에서 주요 대출 가능 자산은 USDC, RLUSD, GHO와 같은 무허가 스테이블코인이다. 이러한 스테이블코인은 누구나 완전히 무허가 방식으로 시장에 공급하여 수익을 얻을 수 있다. 스테이블코인은 자체적으로 담보로 사용할 수 없으며, 이는 허가 기반 담보 영역과 무허가 유동성 영역 간의 명확한 구분을 유지한다.

따라서 대출은 반드시 RWA 담보를 먼저 공급하고 활성화한 사용자에게만 허용된다. 이는 규제된 허가 기반 자산을 활용하여 무허가 유동성에 접근할 수 있는 구조를 만든다. 대출 과정은 에이브 프로토콜의 표준 메커니즘을 따르며, 이자율은 이용률 곡선과 시장 역학에 따라 결정된다.

차입자는 공급한 RWA 담보의 리스크 파라미터에 의해 정해진 충분한 담보 비율을 유지해야 한다. 시스템은 이러한 비율을 지속적으로 모니터링하며, 포지션이 과소 담보 상태가 될 경우 청산을 진행할 수 있다. 단, 청산 메커니즘은 RWA 담보의 고유한 특성을 반영하도록 조정된다.

RWA 가격 오라클 안전 장치(RWA Price Oracle Safeguards)

크립토 네이티브 자산이 탈중앙화 거래소에서 가격이 형성되는 것과 달리, RWA는 발행자가 직접 보고하는 순자산가치(Net Asset Value, NAV)에 의존한다. 여기에는 지연, 부정확, 악의적 가격 보고 가능성 등과 같은 중대한 위험 요소들이 존재한다. 단순한 운영 실수나 악의적 행위로 잘못된 순자산가치가 온체인에 게시되면, 시장 전반에 걸쳐 잘못된 청산을 촉발시킬 수 있다. 물론 긴급 멀티시그를 통해 리저브를 동결할 수 있지만, 불가피한 운영 지연이 발생해 화이트리스트 청산인이 잘못된 가격을 악용하여 차입자에게 치명적 피해를 줄 수 있는 시간이 생긴다.

호라이즌은 이러한 위험을 완화하기 위해 오라클 수준에서 강력한 자동화 안전장치를 구현한다. 체인링크와 라마리스크(LlamaRisk)와 협력하여 각 RWA 가격 피드에 대해 사전에 정의된 상한/하한을 분산 오라클 네트워크(Decentralized Oracle Network, DON) 수준에서 구성한다. 발행자가 새로운 순자산가치를 보고하면, 체인링크의 분산 오라클 네트워크는 이를 상한 및 하한과 대조하여 온체인 게시 여부를 결정한다. 순자산범위가 사전에 합의된 범위 내에 있으면 업데이트가 승인되고, 범위를 벗어나면 업데이트는 거부되며 오라클은 마지막으로 유효한 가격을 계속 보고한다. 이 선제적 메커니즘은 잘못된 가격이 호라이즌 프로토콜에 도달하는 것을 차단하여 잘못된 청산이 발생할 위험을 제거한다. 또한 이러한 가격 범위 이탈이 발생하면 호라이즌의 긴급 멀티시그가 신속히 대응해 원인 분석이 완료될 때까지 신규 대출 발생과 추가 익스포저를 차단한다.

관리 기능 및 예외 상황 처리(Administrative Functions and Edge Case Management)

호라이즌 시장 구조는 RWA 발행자에 의해 지원되는 계정 복구 및 예외 상황 관리 메커니즘을 제공한다. 이러한 관리 기능은 기관 참여에 필수적이며, 규제 환경에서 발생할 수 있는 복잡한 상황에 대한 해결 경로를 마련한다.

예를 들어 사용자가 개인 키 접근 권한을 잃은 경우, 자산 발행자는 관리 권한을 통해 사용자의 전체 담보 및 대출 포지션을 새로 생성한 좀 더 안전한 지갑으로 마이그레이션하는 것을 지원할 수 있다. 이 복구 절차는 시장의 무결성을 유지하면서 기관 사용자에게 필수적인 운영 유연성을 제공한다.

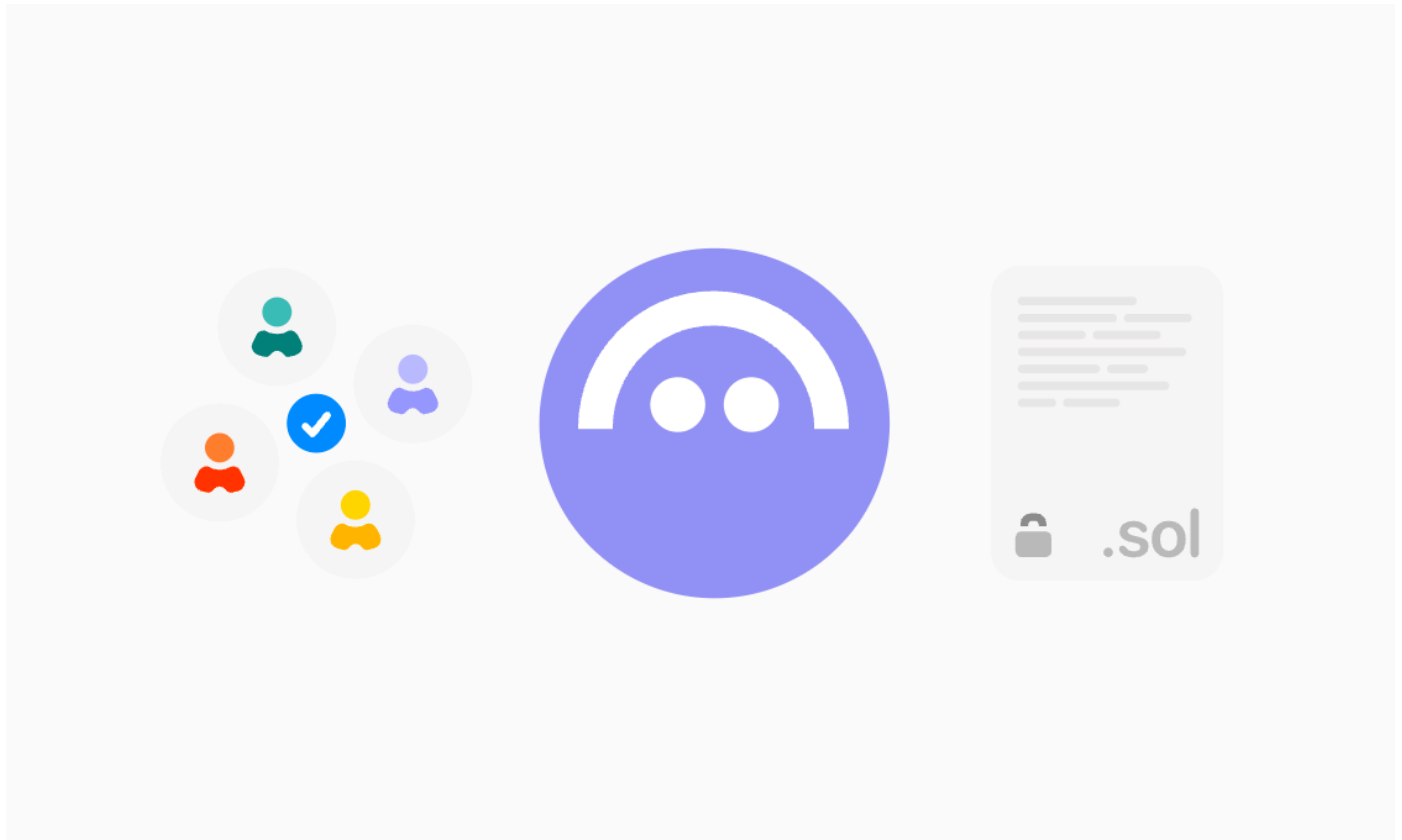
또한 발행자가 법적으로 포지션 보호 조치를 취해야 하는 상황이 발생할 수 있다. 예를 들어, 발행자는 관련 당사자와 협력하여 사용자의 대출 포지션 확대를 방지하는 기능을 수행할 수 있으며, 이는 시장의 담보화와 컴플라이언스를 보존한다.

관리 프레임워크는 투명성과 책임성을 바탕으로 설계되었으며, 명확한 거버넌스 구조에 따라 이러한 예외적 권한의 행사 방식이 결정된다. 이를 통해 기관 참여자는 자신의 이익이 보호된다는 확신을 가지면서도 기반 프로토콜의 탈중앙성을 유지할 수 있다.

RwaAToken 기능 및 역할 구현에 대한 자세한 내용은 [tokenization documentation](#) 참고하길 바란다.

Horizon RWA 마켓 거버넌스에 대한 자세한 내용은 [access controls](#) 참고하길 바란다.

AAVE



AAVE 토큰은 에이브 프로토콜의 네이티브 거버넌스 토큰이다. AAVE 토큰은 보유자가 프로토콜 거버넌스를 통해 의사결정에 참여할 수 있도록 권한을 부여한다. AAVE는 이더리움 블록체인에 배포된 ERC-20 토큰이며, 다양한 중앙화 및 탈중앙화 거래소에서 폭넓게 접근 가능하다.

거버넌스(Governance)

AAVE 토큰 보유자는 거버넌스를 통해 에이브 프로토콜의 미래에 영향을 미칠 수 있다. AAVE 및 안전 모듈에 스테이킹된 AAVE(stkAAVE) 보유자는 제안에 투표하거나 투표 권한을 다른 사람에게 위임할 수 있다. 거버넌스 투표에 참여함으로써 보유자는 프로토콜 배포, 매개변수 조정, 기능 추가 등 다양한 의사결정에 기여한다.

유동성 풀(Liquidity Pools)

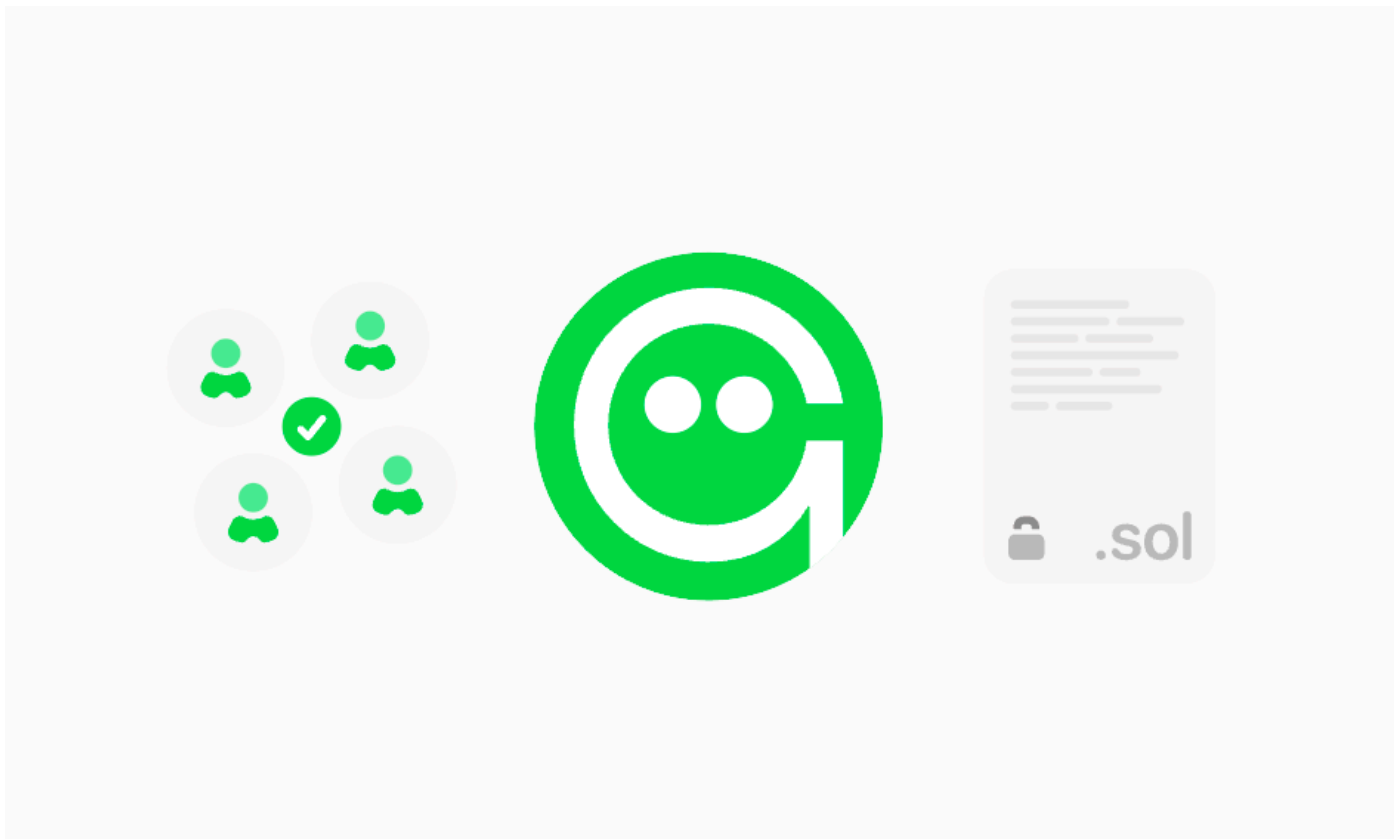
AAVE 토큰은 에이브 프로토콜 내 유동성 풀이나 탈중앙화 거래소와 같은 외부 풀에 공급되어 사용자가 수익을 얻을 수 있다.

보유자는 에이브 안전 모듈에 토큰을 스테이킹하여 시스템 보안을 강화할 수 있는데, 이를 통해 채무 불이행 발생 시 백스톱 (backstop) 역할을 제공하고 그 대가로 보상을 받는다. AAVE 스테이킹은 또한 GH0 대출 금리 인하 혜택을 제공하여 추가적인 참여 유인을 만든다. 더 나아가, AAVE 토큰은 에이브 마켓에 담보로 제공될 수 있어 사용자가 다른 자산을 대출 하고 자본 효율성을 높일 수 있다.

크로스체인(Cross-Chain)

또한 AAVE는 여러 네트워크에서 정규 네트워크 메시징 브리지를 사용한 크로스체인 특성을 지니고 있다. 이를 통해 사용자는 다양한 블록체인 네트워크에서 AAVE에 접근할 수 있다. 이더리움과 다른 네트워크 간 토큰 브리징 능력은 네트워크 브리지의 가용성에 따라 달라진다.

GH0



GH0(발음: "고")는 완전 담보 및 투명성을 원칙으로 에이브 프로토콜에 네이티브로 구현된 초과 담보 기반의 탈중앙화 스테이블코인이다. 미국 달러에 페깅된 가치를 유지하도록 설계되었다. 사용자는 에이브 거버넌스에 의해 설정된 발행 한도 (mint cap) 내에서 필요에 따라 GH0를 발행할 수 있다. 그 안정성은 시장 효율성과 에이브 프로토콜에 내재된 초과 담보 화 메커니즘을 통해 유지된다.

GH0는 이더리움에 배포된 ERC-20 토큰으로, 퍼실리테이터(facilitator) 모델을 기반으로 운영된다. 퍼실리테이터는 에이브 거버넌스에 의해 승인된 컨트랙트로, 거버넌스가 정의한 발행 한도에 따라 GH0를 발행하고 소각할 수 있다. 이 모델은 공급에 대한 탈중앙적 제어를 유지하면서도 GH0 기능 확장에 유연성을 제공한다.

GH0 토큰

GH0 토큰 컨트랙트는 퍼실리테이터와 그 발행 한도를 관리하기 위한 특정 역할을 포함한다:

- `FACILITATOR_MANAGER_ROLE`: 퍼실리테이터 추가 및 제거를 담당한다.
- `BUCKET_MANAGER_ROLE`: 기존 퍼실리테이터의 `bucketCapacity`를 설정하는 역할을 담당한다.

각 퍼실리테이터는 다음을 포함하는 `Facilitator` 구조체로 표현된다:

- `label`: 퍼실리테이터의 문자열 식별자
- `bucketCapacity`: 퍼실리테이터가 발행할 수 있는 GHO의 최대치
- `bucketLevel`: 퍼실리테이터가 현재까지 발행한 GHO의 양

transfer

가장 단순한 ERC-20 토큰 전송 방법은 `transfer`이며, 이는 사전 토큰 승인 없이 어떤 주소로든 토큰을 전송할 수 있다. 단, 이 방식은 반드시 토큰 보유자가 직접 실행해야 하므로, 스마트 컨트랙트 함수 호출 중 사용자(EOA)로부터 자금을 가져오는 데에는 사용할 수 없다.

transferFrom

스마트 컨트랙트 함수 내에서 토큰을 전송하려면 `transferFrom` 메서드를 사용한다. `transferFrom`을 실행하려면 송신자가 해당 `spender` 주소에 대해 최소 전송 `amount`만큼 미리 승인해야 한다. 승인을 수행하는 방법은 두 가지가 있다:

approve

표준 ERC-20 `approve`는 토큰 보유자가 온체인 트랜잭션을 통해 특정 `spender`와 `amount`를 승인하도록 요구한다.

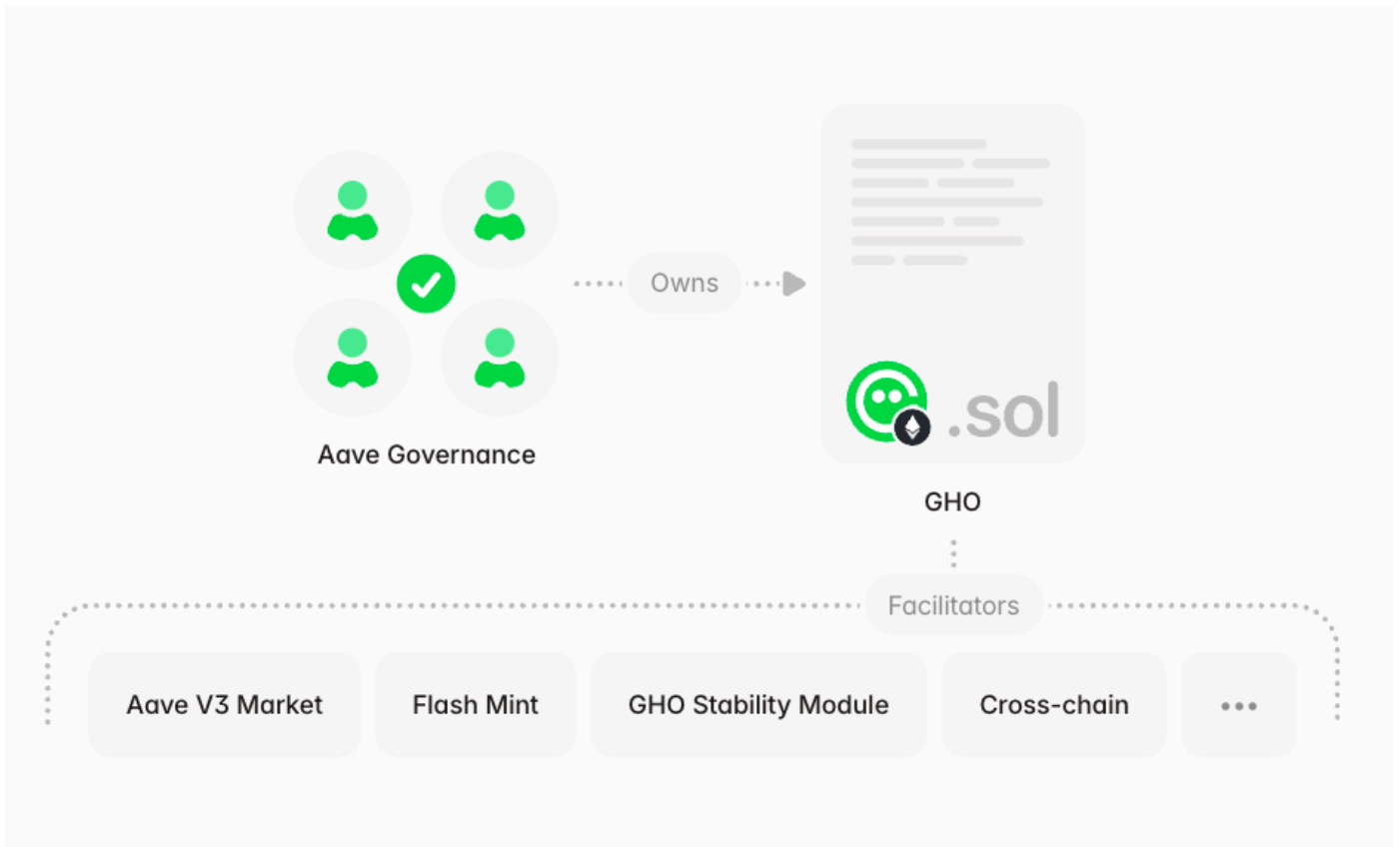
permit

EIP-2612 `permit`은 다음 두 가지 요소를 요구하는 승인 방식이다:

- 토큰 보유자가 서명한 승인 메시지 (`owner`, `spender`, `amount`, `nonce`, `deadline`, `DOMAIN_SEPARATOR`를 포함)
- 어떤 주소에서든 실행 가능한 온체인 `permit` 트랜잭션

`permit`을 `approve` 대신 사용할 때의 장점은 트랜잭션 가스를 토큰 보유자가 아닌 다른 주소가 지불할 수 있으며, `permit` 호출을 다른 액션과 함께 배치하여 트랜잭션 수를 줄일 수 있다는 점이다. 예시로, 에이브 프로토콜 V3의 [supplyWithPermit](#)이 있다.

퍼실리테이터(Facilitators)



퍼실리테이터는 에이브 거버넌스에 의해 승인된 컨트랙트 주소로, `FACILITATOR_MANAGER_ROLE` 및 `BUCKET_MANAGER_ROLE` 권한을 가지며 GHOSol 토큰을 발행(`mint`)하고 소각(`burn`)할 수 있다. 각 퍼실리테이터는 특정 `bucketCapacity` 하에서 운영되며, 이는 해당 퍼실리테이터가 생성할 수 있는 GHOSol의 최대치를 제어한다. 퍼실리테이터는 GHOSol의 안정성을 유지하고 다양한 플랫폼 및 사용 사례에 통합하는 데 핵심적인 역할을 한다.

퍼실리테이터용 GHOSol 토큰 함수

퍼실리테이터는 다음과 같은 함수들을 사용하여 GHOSol 토큰과 상호작용한다:

- `mint(address account, uint256 amount)`: 퍼실리테이터가 `account`로 `amount`만큼의 GHOSol 토큰을 발행한다. `amount`는 0보다 커야 하며, 새로운 `bucketLevel`(현재 `bucketLevel` + `amount`)이 퍼실리테이터의 `bucketCapacity`를 초과해서는 안 된다. 이 함수는 퍼실리테이터의 `bucketLevel`을 갱신한다.
- `burn(uint256 amount)`: 퍼실리테이터가 자신의 잔고에서 `amount`만큼의 GHOSol 토큰을 소각한다. `amount`는 0보다 커야 하며, 퍼실리테이터의 `bucketLevel`은 소각된 `amount`만큼 감소한다.

퍼실리테이터 관리 함수

GHOSol 토큰 컨트랙트는 퍼실리테이터 관리를 위한 함수들도 제공한다:

- `addFacilitator(address facilitatorAddress, string calldata facilitatorLabel, uint128 bucketCapacity)`: `FACILITATOR_MANAGER_ROLE` 권한을 가진 주소가 호출 가능하며, 지정된 `label`과 `bucketCapacity`를 가진 새로운 퍼실리테이터를 추가한다. 퍼실리테이터가 이미 존재하거나 `label`이 비어 있으면 추가할 수 없다.
- `removeFacilitator(address facilitatorAddress)`: `FACILITATOR_MANAGER_ROLE` 권한을 가진 주소가 호출 가능하며, 기존 퍼실리테이터를 제거한다. 퍼실리테이터는 `bucketLevel`이 0이어야만 제거할 수 있다.
- `setFacilitatorBucketCapacity(address facilitator, uint128 newCapacity)`:

BUCKET_MANAGER_ROLE 권한을 가진 주소가 호출 가능하며, 기존 퍼실리테이터의 bucketCapacity를 갱신한다.

퍼실리테이터 조회 함수

퍼실리테이터에 대한 정보는 다음 함수를 통해 조회할 수 있다:

- `getFacilitatorsList()`: 모든 등록된 퍼실리테이터 주소 목록을 반환한다.
- `getFacilitator(address facilitator)`: 지정된 `facilitator` 주소에 대한 `Facilitator` 구조체 상세 정보(`label`, `bucketCapacity`, `bucketLevel`)를 반환한다.
- `getFacilitatorBucket(address facilitator)`: 특정 퍼실리테이터의 `bucketCapacity`와 `bucketLevel`을 반환한다.

에이브 V3 이더리움 마켓

에이브 V3 이더리움 마켓은 GH0의 주요 퍼실리테이터 역할을 한다. 사용자는 승인된 담보 자산을 에이브 프로토콜에 공급하고 이를 담보로 GH0을 대출(발행)할 수 있다. 이 과정은 표준 초과 담보화 관행을 따르며, 프로토콜의 보안과 스테이블코인의 신뢰성을 보장한다.

에이브 풀 퍼실리테이터를 통해 GH0과 상호작용하는 것은 일반적인 에이브 리저브 자산과 유사하지만 두 가지 주요 차이가 있다:

- GH0은 공급되는 것이 아니라 발행되므로, 이자율 및 이용 가능 유동성 계산은 맞춤형 이자율 전략과 퍼실리테이터 한도에 기반한다.

아래는 모든 GH0 동작에 대한 기술 가이드와 해당 컨트랙트 참조다.

발행(Minting)

발행은 에이브 V3 이더리움 마켓의 `borrow` 함수를 통해 이루어진다. GH0을 발행하는 과정은 다른 리저브 자산을 대출하는 것과 거의 동일하다. 발행하려면 주소가 충분한 담보를 보유해야 하며, 이를 위해 담보 자산을 승인한 뒤 에이브 풀에 `supply` 호출을 수행해야 한다. 충분한 담보를 확보한 주소는 담보 자산 구성에 의해 결정된 최대 담보 인자(`maximum collateral factor`)까지 대출할 수 있다.

GH0은 공급자에게서 빌리는 것이 아니라 새롭게 생성되므로, 이용 가능한 유동성 제한을 받지 않는다. 대신 퍼실리테이터 한도와 담보화 요건이 GH0 발행 한계를 정의하며, 다음과 같이 계산된다:

```
availableFacilitatorCap = ghoReserveData.aaveFacilitatorBucketMaxCapacity -  
ghoReserveData.aaveFacilitatorBucketLevel
```

에이브 `borrow` 기능 통합에 대한 더 많은 정보는 [core functions](#) 참고하길 바란다.

상환(Repay)

GH0 상환은 다른 자산과 동일하게 수행되며, Pool 컨트랙트가 GH0 토큰을 사용할 수 있도록 승인(승인 트랜잭션 또는 서명된 `permit`과 `repayWithPermit` 사용)을 거쳐 진행된다.

에이브 `repay` 기능 통합에 대한 더 많은 정보는 [core functions](#) 참고하길 바란다.

청산(Liquidation)

특정 주소가 GHO 차입 포지션을 보유할 경우, 다른 담보화 주소와 동일한 조건에서 청산 대상이 될 수 있다. GHO 차입의 건전성 지수가 1 미만으로 떨어지면(차입 가치의 합이 담보 자산의 청산 한도 가중 평균을 초과할 때 발생), 누구나 Pool 컨트랙트에 대해 `liquidationCall`을 실행할 수 있다.

`liquidationCall`은 GHO 차입 포지션의 최대 100%를 상환하는 대신, 동등한 USD 가치에 상응하는 담보와 청산 보너스를 함께 획득한다.

플래시 민트(Flash Mint)

GHO는 일반적인 에이브 리저브처럼 차입되는 방식이 아니므로, 에이브 풀의 플래시 론 기능을 대체하기 위해 별도의 퍼실리테이터가 사용된다.

FlashMinter 퍼실리테이터는 에이브 풀과는 별도의 발행 한도를 가진다. 모든 FlashMint 거래는 단일 트랜잭션 내에서 반환되므로, 이 퍼실리테이터를 통해 GHO가 실제로 발행되는 일은 없으며, 한도는 각 트랜잭션에 적용된다.

FlashMint는 청산, 부채 변환, 페그 차익거래 등 다양한 활용 사례에서 유용하다. GhoFlashMinter 스마트 컨트랙트는 다음 함수를 구현한다:

```
function maxFlashLoan(address token) external view override returns (uint256)

function getFee() external view override returns (uint256)

function flashLoan(
    IERC3156FlashBorrower receiver,
    address token,
    uint256 amount,
    bytes calldata data
) external override returns (bool)
```

플래시 론 통합 개발에 대한 더 많은 정보는 [개발자 플래시 론 가이드](#)를 참고하기 바란다.

안정화 모듈(Stability Module)

페그 안정화 모듈(Peg Stability Module, PSM)은 두 토큰을 사전에 정해진 비율로 변환할 수 있게 하는 컨트랙트다. GHO 안정화 모듈(GHO Stability Module, GSM)은 기존 PSM 모델의 장점을 활용하면서도 여러 혁신을 도입해 GHO 페그 유지에 기여한다. GSM은 거버넌스에 의해 승인된 토큰과 GHO 간의 변환을 촉진하도록 설계되었으며, 유연한 운영과 리스크 관리를 위한 다양한 기능을 기반으로 한다.

GSMRegistry

GSMRegistry는 모든 GSM 인스턴스의 목록을 저장하는 스마트 컨트랙트다. 이 컨트랙트는 에이브 거버넌스 Short Executor가 소유한다.

GSM

안정화 모듈의 각 토큰 페어링은 매수·매도 기능을 위한 진입점이자 GHO 퍼실리테이터 역할을 하는 GSM 또는 GSM4626 컨트랙트 인스턴스를 가진다.

GSM4626 컨트랙트는 ERC-4626 토큰화된 볼트 지분을 외생(exogenous) 토큰으로 지원하는 특별한 GSM 인스턴스다.

가격 전략(Price Strategy)

GSM은 유연한 가격 전략 프레임워크를 도입하여 시장 상황이나 전략적 목표에 따라 가격 메커니즘을 조정할 수 있도록 한다. 이 시스템은 고정 및 동적 가격 전략을 모두 지원하며, 실시간 상황과 사전에 정의된 조건에 따라 조정된다. 초기 구현은 매수-매도 전환에 대해 1:1 고정 비율을 중심으로 하며, 향후 다오 거버넌스에 따라 동적 전략으로 확장될 수 있다.

수수료 전략(Fee Strategy)

각 GSM 인스턴스는 매수-매도 전환에 대한 수수료 비율을 결정하는 FeeStrategy 컨트랙트를 가지며, 이 수수료는 에이브 다오 트레저리로 할당된다.

익스포저 한도 (Exposure Cap)

익스포저 한도는 안정화 모듈이 보유할 수 있는 외생(exogenous) 토큰의 최대량을 설정하는 파라미터이며, Aave DAO 거버넌스에 의해 결정된다.

변환 동결 및 오라클 가격 범위 (Conversion Freezes and Oracle Price Bounds)

외생 토큰의 가격이 설정된 비율에서 벗어날 경우, 에이브 다오는 동결 역할을 직접 수행하거나 특정 주체(자율 에이전트 또는 컨트랙트)에 할당하여 대응하고 변환을 중단시킬 수 있다.

동결 역할의 구현체는 OracleSwapFreezer 컨트랙트다. 이 컨트랙트는 체인링크 오라클과 에이브 거버넌스가 정한 가격 범위를 활용하여, 오라클 조건에 따라 동결 또는 해제를 수행한다.

최후 청산(Last Resort Liquidations)

외생 토큰의 리스크가 급격히 증가하는 경우, GSM은 최후 청산 기능을 제공하여 외생 토큰을 청산한다. 이 컨트랙트의 역할은 최악의 상황에서 다오가 GSM 기능을 중단하고 외생 토큰의 잔액을 청산할 수 있도록 한다.

크로스체인(Cross-Chain)

모든 GHO 토큰은 이더리움 메인넷에서 기원한다. GHO는 크로스체인 메시징 인프라를 통해 다른 네트워크에서도 접근 가능하다.

체인링크 CCIP 프로토콜은 네트워크 간 GHO 전송을 지원하는 메시징 브리지로 에이브 거버넌스에 의해 승인되었다.

GHO는 네트워크 간 전송 시, 출발지 네트워크에서 잠금(Ethereum 메인넷) 또는 소각(기타 네트워크)을 실행하고, 크로스체인 메시지가 검증된 후 목적지 체인에서 해제(Ethereum 메인넷) 또는 발행(기타 네트워크)이 이루어진다.

GhoCCTPTokenPoolEthereum 컨트랙트는 이더리움 및 기타 체인에서 GHO의 잠금과 소각을 지원하여, GHO가 다양한 디파이 생태계에 존재할 수 있게 한다.

GHO 유동성 위원회(GHO Liquidity Committee)

GHO 유동성 위원회(GLC)는 2023년 10월에 GHO 스테이블코인의 유동성에만 집중하기 위해 설립되었다. 위원회는 [거버넌스 제안](#)을 통해 소규모 팀으로 구성되었으며, 초기 3개월 성공적인 활동 후 [에이브 유동성 위원회\(Aave Liquidity Committee, ALC\)](#)로 통합되었다.

ALC의 GHO 관련 주요 역할은 다음과 같다:

- 유동성 전략 분석 및 모델링 제공
- GHO 유동성을 호스팅하는 프로토콜 지원팀과 협력
- 위원회의 주간 활동 주도 및 조정
- 전략 개선을 위한 핵심 피드백 제공

- 트랜잭션 검증 및 서명 수행

GHO에 대한 ALC의 성과 지표와 유동성 목표는 TokenLogic에서 제공하는 [GHO Analytics platform](#)에서 확인할 수 있다.

GHO 유동성 위원회의 역할에 대한 추가 정보는 [에이브 거버넌스 포럼](#)에서 확인할 수 있다.

GHO 스튜어드(GHO Stewards)

GHO 스튜어드는 2024년 4월에 설립된 조직으로, GHO 시장 매개 변수를 보다 유연하게 관리하여 시장 상황에 따라 GHO 생태계를 확장할 수 있도록 한다. GHO 스튜어드 컨트랙트의 소스 코드는 [관련 깃허브](#)에서 확인할 수 있다.

GHO 스튜어드는 사전에 정의되고 거버넌스에 의해 승인된 임계값 내에서 다음 항목들의 조정 여부와 조정 범위를 결정한다:

- GHO 차입 한도(GHO Borrow Cap)
- GHO 차입 금리(GHO Borrow Rate)
- GSM 익스포저 한도(GSM Exposure Cap)
- GSM 버킷 용량(GSM Bucket Capacity)
- GSM 가격 전략(GSM Price Strategy)
- GSM 수수료 전략(GSM Fee Strategy)
- GSM 가격 범위(GSM Price Range)

다수의 유동성 풀이 생성되고 보상이 분배되는 상황에서, 다오가 신속하게 GHO 차입 한도를 늘려 GHO가 1달러 이상에서 거래되는 상황을 해소하는 것이 중요하다. GHO 스튜어드는 GHO 페그 상회 거래를 해소하기 위해 빠르게 차입 한도를 늘릴 수 있으며, 50M 단위까지 증액하여 총 차입 한도를 100M까지 확장할 수 있다.

차입 금리는 생태계가 안전하게 확장될 수 있도록 점진적으로 조정되어야 한다. GHO의 30일 이동 평균 가격이 \$0.995~\$1.005 범위를 벗어날 경우, GHO 스튜어드는 2일 주기로 최대 500bps까지만 차입 금리를 조정할 수 있으며, 최대 25% APR까지 허용된다.

GHO 스튜어드는 Growth(ACI), Risk(ChaosLabs), Finance(TokenLogic + karpatkey) 서비스 제공자 멤버로 구성되며, 3/4 멀티시그를 사용한다.

sGHO



sGHO는 GH0 스테이블코인을 위한 에이브 프로토콜의 네이티브 예치 메커니즘으로, 이더리움 메인넷에 배포되어 있다. sGHO는 GH0 보유자가 자신의 토큰을 스마트 컨트랙트에 공급하고, GH0 형태의 보상이 축적되는 sGHO를 받도록 한다. 이 메커니즘에는 sGHO를 상환하고 GH0를 받을 때 쿨다운이 없으며, 삭감 리스크도 없고, 공급된 자금의 재사용 (rehypothecation)도 없다. sGHO 보유자에 대한 모든 보상은 [Merit 프로그램](#)을 통해 분배된다.

테스트 및 디버깅(Testing & Debugging)

이 가이드는 에이브 프로토콜 통합을 테스트하고 디버깅하는 방법에 대한 종합적인 지침을 제공한다.

테스트 (Testing)

에이브 프로토콜을 테스트하는 주요 방법은 테스트넷(testnet)을 사용하거나 포크 네트워크(fork networks)를 생성하는 것이다.

테스트넷 (Testnet)

에이브 프로토콜은 Sepolia 테스트넷에서 테스트할 수 있다. app.aave.com에 접속하여 오른쪽 상단에서 테스트넷 모드를 활성화하면 Sepolia V3 테스트넷 마켓에 접근할 수 있다.

시작 방법:

1. app.aave.com에서 테스트 마켓 방문
2. Faucet 탭을 사용하여 테스트 마켓에서 제공되는 테스트 토큰 획득
3. 해당 토큰을 프로토콜에 공급
4. 대출, 상환, 인출 등 다양한 기능을 수행하여 테스트

포크 네트워크(Fork Networks)

포크 네트워크는 블록체인의 상태를 복제하여 샌드박스 환경에서 상호작용할 수 있게 한다. 이는 메인넷에 존재하는 컨트랙트 및 상태에 의존하는 시나리오를 테스트하는 데 특히 유용하다.

포크 네트워크 생성

여러 도구를 사용해 포크 네트워크를 생성할 수 있다:

- [Tenderly](#): 포크 네트워크, 시뮬레이션, 온라인 대시보드를 제공하는 디버깅 툴킷
- [Foundry](#): 스마트 컨트랙트 개발 프레임워크
- [Hardhat](#): 스마트 컨트랙트 개발 프레임워크

포크 네트워크에서 토큰을 얻기 위해, 원하는 토큰을 보유한 주소(블록 탐색기에서 확인 가능)를 가장(impersonate)하여 자신의 주소로 토큰을 전송할 수 있다.

포크 네트워크를 UI에 연결

app.aave.com 인터페이스는 포크 네트워크에 연결되도록 설정할 수 있다. 브라우저 콘솔을 통해 특정 변수를 설정하여 이 기능을 활성화할 수 있다. 자세한 설정 방법은 [이 가이드](#)를 참고하기 바란다.

디버깅 (Debugging)

에이브 프로토콜과 상호작용할 때 스마트 컨트랙트 에러가 발생할 수 있다. 이 섹션은 에러 코드를 해석하고, 디버깅 도구를 사용하는 방법 및 일반적인 문제를 해결하는 방법을 다룬다.

에러 코드 (Error Codes)

에이브 프로토콜은 트랜잭션 실패 원인을 나타내기 위해 특정 에러 코드를 사용한다. 이러한 코드를 이해하는 것은 디버깅에 필수적이다. 각 에이브 프로토콜 버전의 에러 코드는 해당 `Errors.sol` 파일에 정의되어 있다:

- 에이브 V3 에러 코드: [Errors.sol](#)
- 에이브 V2 에러 코드: [Errors.sol](#)

디버깅 도구(Debugging Tools)

트랜잭션을 효과적으로 디버깅하려면 Tenderly, Hardhat, Foundry와 같은 도구를 사용할 수 있다. 이 도구들은 트랜잭션 시뮬레이션, 상태 변화 검사, 실행 단계별 추적을 통해 문제를 식별할 수 있게 한다.

Tenderly

Tenderly는 트랜잭션을 시뮬레이션하고 단계별로 디버깅할 수 있다. 트랜잭션 세부 정보(to, from, data, value)를 입력하면 호출을 시뮬레이션하고, 오류 코드와 스택 트레이스를 포함해 revert 원인을 식별할 수 있다.

Hardhat

Hardhat은 테스트 및 디버깅을 위한 로컬 이더리움 환경을 제공한다. 솔리디티 콘솔 로그, 상세 스택 트레이스, 메인넷 상태 포킹 등 다양한 기능을 지원한다.

Foundry

Foundry는 빠르고 휴대 가능하며 모듈화된 이더리움 애플리케이션 개발 툴킷이다. Forge와 같은 툴을 포함하여 스마트 컨트랙트 테스트 및 디버깅을 지원하며, 트랜잭션을 효과적으로 시뮬레이션하고 디버깅할 수 있다.

"Gas Estimation Failed" 이해하기

"Gas estimation failed" 에러는 트랜잭션 실행 시 실패할 것임을 나타내는 일반적인 메시지다. 이 문제를 진단하려면:

1. 트랜잭션의 to, from, data, value 필드를 확인한다.
2. 위에서 언급한 디버깅 도구를 사용해 트랜잭션을 시뮬레이션한다.
3. 실패를 유발한 특정 에러 코드나 예외를 식별한다.
4. 스택 트레이스를 검사하여 문제를 정확히 찾아낸다.

일반적인 오류(Common Errors)

에이브 V3 풀(Aave V3 Pool)

에이브 V3 풀과 상호작용할 때 토큰 승인 문제, 담보 부족, 프로토콜 한도 도달 등과 관련된 오류가 발생할 수 있다.

공급 및 상환을 위한 ERC-20 토큰 승인(ERC-20 Token Approval for Supply and Repay)

토큰을 공급하거나 상환하기 전에, 에이브 Pool 컨트랙트가 해당 토큰을 전송할 수 있도록 허용해야 한다. 이를 위해 ERC-20 토큰 컨트랙트의 approve 함수를 호출하여 Pool을 spender로 승인해야 한다.

담보 부족(Insufficient Collateral)

자산을 차입할 수 없다면, 담보 부족이나 요구되는 담보 인정 비율을 충족하지 못했을 수 있다. 충분한 담보를 공급했는지, 그리고 그것이 프로토콜의 요구사항을 충족하는지 확인해야 한다.

차입 한도 도달(Borrow Cap Reached)

특정 자산을 차입할 수 없는 경우, 해당 자산의 차입 한도가 도달했을 수 있다. 이 경우 차입자가 상환하거나 풀에 더 많은 유동성이 공급될 때까지 기다려야 한다.

잠재적인 revert 케이스에 대한 더 자세한 설명은 [디파이 통합 가이드](#)를 참고하기 바란다.

담보 인정 비율 = 0%(Loan To Value = 0%)

에이브 [리저브](#)의 담보 인정 비율 파라미터가 0%로 설정된 경우(LTV0라 불림), 리저브 상호작용에 다음과 같은 영향을 미친다:

- 차입 능력: 해당 자산을 담보로 보유해도 더 이상 차입 능력이 제공되지 않는다.
- 담보 상태: 해당 자산은 더 이상 담보로 활성화될 수 없다.
- 전송: 전송을 통해 자산을 받을 경우 자동으로 담보로 활성화되지 않는다.
- 출금: LTV0 자산은 담보 상태를 비활성화(Pool.setUserUseReserveAsCollateral 사용 또는 공급 잔액 전액 출금)해야만 다른 자산을 출금할 수 있다.

WrappedTokenGateway

에이브 프로토콜에서 이더(ETH)와 상호작용할 때는 WrappedTokenGateway가 ETH를 WETH로 감싸거나(Wrap), 반대로 해제(Unwrap)하는 데 사용된다. 일반적인 문제로는 트랜잭션에 ETH를 전송하지 않은 경우, 또는 입금/출금 시 잘못된 함수를 사용하는 경우가 있다.

자세한 사용법 및 잠재적인 revert 원인에 대한 내용은 [WrappedTokenGateway 문서](#)를 참고하기 바란다.