

# Contracts Overview

## Contracts Overview

The Aave Protocol V3 contracts are divided into two repositories:

- [aave-v3-core](#)
- : Hosts core protocol V3 contracts that contains the logic for - supply, borrow, liquidation, flashloan, a/s/v tokens, portal, pool configuration, oracles and interest rate strategies.
- [aave-v3-periphery](#)
- : rewards, ui data provider, incentive data provider, wallet balance provider and WETH gateway.
- 

## Core Contracts

Core protocol contracts fall in following 4 categories:

- Configuration
- Pool logic
- Tokenization
- Misc
- 

## Configuration

### [ACLManager](#)

Aave Protocol V3 implements an access control list to segregate powers and/or benefits that can be allocated to different entities on the protocol. The roles and holders are managed in the [ACLManager.sol](#), which keeps track of the individual roles and its holders.

### [PoolAddressesProvider](#)

The main addresses register of the protocol, containing address of core protocol contracts and ACL admin. It acts as factory of proxies and admin of those. The owner of this contract has the right to set/update implementation of the upgradable contracts. The latest contract addresses should be retrieved from this contract by making the appropriate calls.

### [PoolAddressesProviderRegistry](#)

Contains a list of active PoolAddressProvider addresses, for different markets. It is used for indexing all Aave protocol's markets.

### [PriceOracleSentinel](#)

Oracle Sentinel validates if operations are allowed depending on the PriceOracle health. Once the PriceOracle gets up after an outage/downtime, users can make their positions healthy during a grace period.

## Pool Logic

### [DefaultReserveInterestRateStrategy](#)

Implements the calculation of the interest rates depending on the reserve state. This contract holds the information needed to calculate and update the yield relating to specific liquidity pools.

Each contract stores the optimised base curves using the corresponding parameters of each asset. This means that there is a mathematical function which determines the yield of each liquidity pool, with the yield changing based on the amount of borrowed funds and the total liquidity (i.e. utilisation) of the pool.

### [Pool](#)

The main entry point into the Aave Protocol. Most user interactions with the Aave Protocol occur via the Pool contract. Pool is owned by the PoolAddressesProvider of the specific market. All admin functions are callable by the PoolConfigurator contract, which is defined in PoolAddressesProvider.

### [PoolConfigurator](#)

Provider configuration methods for the Pool contract. The write methods of this contract can only be called by addresses with corresponding permission-ed system roles that are managed by ACLManager.

Tokenization

### [AToken](#)

Yield-generating tokens that are minted and burnt upon supply and withdraw of assets to Aave Pool

### [DelegationAwareAToken](#)

The special type of aToken that are minted and burnt upon supply and withdraw of assets that has voting power associated (which can be delegated) with them.

### [StableDebtToken](#)

The non-transferable interest accruing, stable rate tokenised borrows.

### [VariableDebtToken](#)

The non-transferable interest accruing, variable rate tokenised borrows.

Misc

### [AaveOracle](#)

Contract to get asset prices, manage price sources and update the fallback oracle.

### [AaveProtocolDataProvider](#)

Peripheral contract to collect and pre-process information from the Pool.

Peripheral Contracts

Rewards

### [RewardsController](#)

This contract is responsible for configuring the different rewards and the claim process.

Transfer Strategies

These are isolated contracts that manages the procedure of the rewards transfer at claim. This allows the RewardsController to support any custom rewards like Staked Aave, common ERC20 or even NFT.

Misc

### [UiIncentiveDataProviderV3](#)

Helper contract to fetch Incentives Data. It is used by Aave UI for reward balance info.

### [UiPoolDataProviderV3](#)

Helper contract, used by Aave UI, to fetch Pool Data such reserves list, all reserves data like liquidity, token addresses, rate strategy etc.

### [WETHGateway](#)

Only ERC20 tokens can be supplied & borrowed from the Aave Protocol V3 pools. To facilitate using token native to the chain (such ETH on Ethereum or Matic on polygon) a helper contract is used to support single tx supply, borrow, repay and withdraw.

### [WalletBalanceProvider](#)

Used for getting multiple tokens balance for one user address. This contract is used by Aave backend to reduce the number of blockchain calls for fetching user balance.

[Previous V3 Overview](#) [Next L2 Optimization](#) Last updated 6 months ago On this page \* [Contracts Overview](#) \* [Core Contracts](#) \* [Configuration](#) \* [Pool Logic](#) \* [Tokenization](#) \* [Misc](#) \* [Peripheral Contracts](#) \* [Rewards](#) \* [Misc](#)

Was this helpful?