Decentralized Data Model

This page describes how data aggregation is applied to produce Chainlink Data Feeds and provides more insight as to how Data Feeds are updated.

Data aggregation

Each data feed is updated by multiple, independent Chainlink oracle operators. TheAccessControlledOffchainAggregator aggregates the data onchain.

Offchain Reporting (OCR) further enhances the aggregation process. To learn more about OCR and how it works, see theOffchain Reporting page.

Shared data resource

Each data feed is built and funded by the community of users who rely on accurate, up-to-date data in their smart contracts. As more users rely on and contribute to a data feed, the quality of the data feed improves. For this reason, each data feed has its own properties depending on the needs of its community of users.

Decentralized Oracle Network

Each data feed is updated by a decentralized oracle network. Each oracle operator is rewarded for publishing data. The number of oracles contributing to each feed varies. In order for an update to take place, the data feed aggregator contract must receive responses from a minimum number of oracles or the latest answer will not be updated. You can see the minimum number of oracles for the corresponding feed atdata.chain.link .

Each oracle in the set publishes data during an aggregation round. That data is validated and aggregated by a smart contract, which forms the feed's latest and trusted answer.

Components of a Decentralized Oracle Network

Data Feeds are an example of a decentralized oracle network, and include the following components:

- A consumer contract
- A proxy contract
- An aggregator contract

To learn how to create a consumer contract that uses an existing data feed, read theUsing Data Feeds documentation.

Consumer

A Consumer contract is any contract that uses Chainlink Data Feeds to consume aggregated data. Consumer contracts must reference the correctAggregatorV3Interface contract and call one of the exposed functions.

...AggregatorV3Interface feed=AggregatorV3Interface(address);returnfeed.latestRoundData(); Offchain applications can also consume data feeds. See the Javascript and Python example code on theUsing Data Feeds page to learn more.

Proxy

Proxy contracts are onchain proxies that point to the aggregator for a particular data feed. Using proxies enables the underlying aggregator to be upgraded without any service interruption to consuming contracts.

Proxy contracts can vary from one data feed to another, but theAggregatorProxy.solcontract on Github is a common example.

Aggregator

An aggregator is the contract that receives periodic data updates from the oracle network. Aggregators store aggregated data onchain so that consumers can retrieve it and act upon it within the same transaction.

You can access this data using the Data Feed address and theAggregatorV3Interfacecontract .

Aggregators receive updates from the oracle network only when theDeviation ThresholdorHeartbeat Thresholdtriggers an update during an aggregation round. The first condition that is met triggers an update to the data.

- Deviation Threshold: A new aggregation round starts when a node identifies that the off-chain values deviate by more

than the defined deviation threshold from the onchain value. Individual nodes monitor one or more data providers for each feed.
- Heartbeat Threshold: A new aggregation round starts after a specified amount of time from the last update.