Gelato Network Contracts

Intro to Gelato's use of the Diamond Proxy pattern Gelato Network is designed to handle the rapidly evolving multi-chain landscape with its diversity of L1, L2 and sidechain designs. Significant differences can be found across aspects such as consensus mechanisms, block times, likelihood of block re-orgs and transaction fee models.

To handle the nuances that each blockchain design entails, GelatoV2 smart contracts follow a modular upgradeability standard:EIP-2535 Diamond Proxy .

Diamond Proxy Pattern

A Diamond contains:

- Proxy contract
- that holds all state variables.
- Facets
- which are smart contracts that implement any desired functionality and can be replaced at any time.
- Libraries
- , which can be used to share state variables and utility functions across all Facets.
-

Benefits of Diamond Proxy Pattern

- Having a single smart contract holding all the state, while not running into issues of exceeding bytecode size.
- Ability to share state variables and functions between multiple Facets.
- Fine-grained control in terms of which components of the protocol to upgrade. This means that protocol upgrades are gas-efficient.
- Optionality of changing GelatoV2 from upgradeable to immutable at any time, simply by revoking rights to upgrade Facets.
-

Implementing GelatoV2 as a Diamond means that we can easily accommodate new use cases by eliminating integration friction with users and developers, adapt to lower level changes such as a chain changing from the legacy transaction fee model to EIP-1559, and simply adding or removing features as needed without enforcing strong opinions at the application interface level.

GelatoV2 on Ethereum mainnet can be found here:https://louper.dev/diamond/0x3CACa7b48D0573D793d3b0279b5F0029180E83b6

Facets of GelatoV2

Brief guide to the facets that make up GelatoV2 on Ethereum mainnet:

Facet Function DiamondCutFacet Used to make upgrades to the Diamond, such as removing and adding new Facets, and possibly initialising their state variables upon deployment. DiamondLoupeFacet Helper smart contract which allows one to inspect all Facets in the Diamond at any given time. OwnershipFacet Manage the ownership of GelatoV2, whose owner is currently the Gelato Multisig smart contract. AddressFacet Manage Gelato specific utility smart contracts such as the gas price oracle and oracle aggregator. GelatoV1Facet Backward compatibility with GelatoV1. ConcurrentCanExecFacet Coordination algorithm between multiple Gelato executors. ExecAccessFacet Manage Gelato executors. ExecAccessFlashbotsFacet Manage Gelato executors that submit their tasks as Flashbots bundles. ExecAccountingFacet Manage accounting of transaction fees, and soon also payroll to Gelato executors. ExecFacet Main Facet. All tasks submitted to Gelato will be routed through theexec method. PrepaidExecFacet Facet where some use-case specific calls will be routed. Currently deprecated. UniswapV2SwapFacet Similar to ExecFacet, but automatically handles token swaps upon payment. Currently deprecated. TransferFacet Fee withdrawal from GelatoV2, managed by Gelato Multisig. Implementations of GelatoV2 on other chains can also be found onLouper .

Previous Legacy Automate Migration Guide Next Teams Last updated1 month ago On this page *Diamond Proxy Pattern * Benefits of Diamond Proxy Pattern * Facets of GelatoV2