

# Preparing for Fault Proofs Breaking Changes

This page outlines breaking changes related to Fault Proof upgrade for bridges, centralized exchanges, and custom solutions that use withdrawals. This page outlines changes for OP Mainnet and OP Sepolia only, and details for other OP Chains are forthcoming. If you experience difficulty at any stage of this process, please reach out to [developer support \(opens in a new tab\)](#).

⚠️ \* Fault Proofs are scheduled to launch on testnet on Tuesday, March 19, 2024 \* . \* Withdrawals between OP Sepolia and Sepolia will no longer be instant, since they are using the Fault Proofs mechanism.

## Overview of Changes

The `L2OutputOracle` is being entirely removed and replaced by the `OptimismPortal` and `DisputeGameFactory`. The `L2OutputOracle` smart contract is currently used by the trusted Proposer role to store L2 state output proposals. Presently, developers use these outputs to prove that their withdrawals actually happened on L2. But with fault proofs, developers will have to change how their client software proves withdrawals in the first step of the two-step withdrawal process.

### L2OutputOracle

The `OptimismPortal` is changing slightly with Fault Proof Mainnet because it now points to the `DisputeGameFactory` contract instead of the `L2OutputOracle` contract. Most notable change for developers is that withdrawals will have to be proven against the root claim of some dispute game rather than referencing an output in the `L2OutputOracle` contract.

### OptimismPortal

The `OptimismPortal` smart contract is the low-level contract on L1 used to execute deposits and withdrawals. Previously, developers would look at the `L2OutputOracle` to find the exact next output that included their withdrawal. Now, developers must look at the `OptimismPortal` contract to determine the respected game type and then use this information to query the `DisputeGameFactory` for a list of recent `DisputeGame` contracts with the correct game type.

### DisputeGameFactory

It is recommended that developers search for a reasonable number of recent games, say 100 games, and pick the first proposal with a sufficient block number. Developers should then verify this proposal locally as the default game type will allow for permissionless proposals and there is no longer a strong guarantee that proposals will be valid.

## For Bridges and Centralized Exchanges

The proposed Fault Proof upgrade requires developers to enable Fault Proof changes before the Sepolia release. This impacts bridges, centralized exchanges, and custom solutions that use withdrawals.

### Update Logic to Support Fault Proofs

Most teams use the Optimism SDK or Viem to handle this logic under the hood and will simply need to update their software version ahead of the upgrade. However, any project performing withdrawals programmatically will need to update their client code (see [OptimismPortal](#) for details).

- \* Option 1: Optimism SDK Update. \* If you use OptimismSDK for bridging, simply update to version 3.2.0 or higher. \* The Optimism SDK changes do not break the API and require no changes other than updating to the correct software version to support the new `OptimismPortal` logic. The Optimism SDK will automatically begin to use the new logic once it detects that the FPM update has gone live.
- \* Option 2: Viem Update. \* Viem changes will break the API and require both updating to latest version and replacing use of the currently used decorator with the experimental `experimental` decorator that supports fault proofs. When fault proofs are on Mainnet in the future, `publicActionsL2` will be updated to support fault proofs by default, and it will be recommended that developers switch to the stable API.

### Update Withdrawal Monitor

The Withdrawal Monitor service is an important part of the two-step withdrawal system that checks if anyone has been able to prove withdrawals that do not actually appear on L2. The Withdrawal Monitor is now slightly slower at startup time but is more reliable, simpler, and compatible with more infrastructure to more easily support any OP Stack chain. Changes to the Withdrawal Monitor are entirely backwards compatible.

- Option 1: Withdrawal from source.
- If building or using withdrawal-monitor from source, make sure that you are using the latest develop branch. All withdrawal monitor changes are fully backwards compatible.
- Option 2: Docker image.
- If using docker, use the latest chain-mon docker image.

## Update Dispute Monitor

The Dispute Monitor service detects when invalid outputs have been proposed. Given the large number of changes to the output proposal system, the current Fault Monitor is being replaced in favor of a new Dispute Monitor purposely built for the fault proof monitor system.

⚠ Any team running the current Fault Monitor will see the monitor stop functioning when the Fault Proof Monitor upgrade goes live. These teams should run the new service and update their alerting accordingly.

## Next Steps

- See the [Fault Proofs Overview](#)
- to learn more about the three main components of the Fault Proof System.
- You can also read more about [Cannon FPVM](#)
- and [Mips.sol](#)
- , the on-chain smart contract implementation of Cannon.

[Preparing for Ecotone Breaking Changes Superchain App Quick Start](#)