

Fault Proofs Mainnet Security

Source code for Fault Proof Mainnet contracts approved by Optimism Governance can be found [here \(opens in a new tab\)](#) . This page details changes to the security model of the OP Stack with the introduction of the Fault Proof Mainnet upgrade. The most significant change introduced by the Fault Proof Mainnet upgrade is the modification of the `OptimismPortal` to reference the `DisputeGameFactory` instead of the `permissionedL2OutputOracle` .

- `TheDisputeGameFactory`
- contract generates `FaultDisputeGame`
- contract instances that each act as a host to a proposal about the state of the OP Stack chain at a given block number.
- Unlike the `L2OutputOracle`
- , the `DisputeGameFactory`
- contract offers users the ability to permissionlessly play "fault dispute games" in which the correctness of the proposal is determined programmatically.

Security Model

Fault Proof Mainnet is a large contract upgrade that introduces a number of novel components. Given the relative complexity of these novel components, the approach to security for FPM has been to limit the blast radius of potential bugs to very specific contracts and fallback mechanisms that can be easily audited.

Handling Invalid Game Results

All of the security mechanisms put in place generally revolve around the possibility that a `FaultDisputeGame` contract may incorrectly finalize an invalid game result. There are two variations of this:

1. Resolving that an invalid proposal is valid potentially leading to stolen funds, and
2. Resolving that a valid proposal is invalid causing liveness delays or failures.

Both cases would cause honest challengers to lose bonds (unless the `Guardian` stepped in). Potential impact is managed through the introduction of a number of safeguards within the `OptimismPortal` and `FaultDisputeGame` contracts.

Safeguards Within `OptimismPortal`

The `OptimismPortal` contract includes various security mechanisms that allow the `Guardian` and `SystemOwner` roles to collaborate to prevent invalid proposals from impacting withdrawals.

- `TheSystemOwner`
- can replace the `Guardian`
- address.
- `TheGuardian`
- can trigger the global pause mechanism found in the original system.
- `TheGuardian`
- can "blacklist" specific `FaultDisputeGame`
- contracts that resolve incorrectly.
- `TheGuardian`
- can change the respected type of `FaultDisputeGame`
- contract in the case that an entire class of `FaultDisputeGame`
- contracts is found to have critical bugs. If desired, the `Guardian`
- can also choose to revert to a `PermissionedDisputeGame`
- contract that only allows specific roles to submit and challenge proposals.

Safeguards Within `FaultDisputeGame`

The `FaultDisputeGame` contracts store bonds within a `DelayedWETH` contract that is managed by the `SystemOwner` . Withdrawals from the `DelayedWETH` contract are delayed which gives the `SystemOwner` the ability to manually recover from situations in which bonds would be incorrectly distributed. This delay is set to 7 days on OP Mainnet to give the `SystemOwner` or `Guardian` sufficient time to respond to potential security concerns.

Safeguards Within `DelayedWETH`

- `TheSystemOwner`
- can replace the `Guardian`
- address.
- `TheSystemOwner`
- can hold funds from any specific `DisputeGame`
- contract.

- TheSystemOwner
- can remove funds from theDelayedWETH
- contract if the issue extends to so manyDisputeGame
- contracts that holding funds from specific contracts is not viable.
- TheGuardian
- can trigger the global pause mechanism to halt WETH withdrawals.

Cumulative Security Impact

As with the original system, the cumulative effect of these security capabilities is that theGuardian role provides fast response capabilities while theSystemOwner can always step in to resolve all classes of bugs that could result in a loss of funds. The most significant change in security model with the introduction of Fault Proof Mainnet is thatSystemOwner can take a more passive role as invalid proposals will generally be rejected by the Fault Proof system while theGuardian can act as a backstop only in case of a failure in the fault proof game.

Next Steps

- See the[FP Components](#)
- for an overview of FP system components and how they work together to enhance decentralization in the Optimism ecosystem.
- See the[specs\(opens in a new tab\)](#)
- for detailed information about the entire FP program, FP virtual machine, and dispute game.

[MIPS.sol Custom Gas Token](#)