Smart Contract overview

This guide provides an overview of the functionality of the smart contract components. You can also findcontract addresses on OP Mainnet.

Layer 1 contracts

The layer 1 contracts of the OP Stack are deployed on Ethereum. Their primary purpose is to facilitate the cross domain message passing and maintain the valid state root of the layer 2.

Official releases

The full smart contract release process is documented in themonorepo(opens in a new tab) . All production releases are always tagged, versioned as/v . Contract releases have a component name ofop-contracts and therefore are tagged asop-contract/vX.Y.Z .

⚠ For contract releases, refer to the GitHub release notes for a given release, which will list the specific contracts being released—not all contracts are considered production ready within a release , and many are under active development. These release pages are linked below.

Releases or tags of the formvwithout a component name, such asv1.1.4 , indicate releases of all Go code only, andDO NOT include smart contracts. DO NOT use these releases for deploying smart contracts—only deploy fromop-contracts/vX.Y.Z

op-contracts/v1.6.0 - Fault proof fixes

The release fixes security vulnerabilities found in Fault Proof contracts. They were made in response to security vulnerabilities identified during a series of third-party security audits by Spearbit, Cantina, and Code4rena. None of the vulnerabilities have been exploited, and user assets are not and were never at risk.

The upgrade was coupled with theGranite network upgrade to improve the stability and performance of the Fault Proof System. In addition, the capabilities of the Guardian and DeputyGuardian have been extended to set the anchor state for the Fault Proof System in order to prevent referencing invalid anchor states.

- Official - Fault Proof Fixes Release(opens in a new tab)
- Governance Post(opens in a new tab)

Changelog Added:

- MIPS:1.1.0(opens in a new tab)
- PreimageOracle:1.1.2(opens in a new tab)
- FaultDisputeGame:1.3.0(opens in a new tab)
- PermissionedDisputeGame:1.3.0(opens in a new tab)
- DelayedWETH:1.1.0(opens in a new tab)
- DeputyGuardianModule:2.0.0(opens in a new tab)
- AnchorStateRegistry:2.0.0(opens in a new tab)

No change:

- AddressManager:Latest (this has no version)(opens in a new tab)
- L1CrossDomainMessenger:2.3.0(opens in a new tab)
- L1ERC721Bridge:2.1.0(opens in a new tab)
- L1StandardBridge:2.1.0(opens in a new tab)
- OptimismMintableERC20Factory:1.9.0(opens in a new tab)
- OptimismPortal:3.10.0(opens in a new tab)
- SystemConfig:2.2.0(opens in a new tab)
- DisputeGameFactory:1.0.0(opens in a new tab)
- SuperchainConfig:1.1.0(opens in a new tab)
- ProtocolVersions:1.0.0(opens in a new tab)

op-contracts/v1.5.0 - Safe extensions

The Safe Extensions protocol upgrade is intended to increase the security and decentralization of the Superchain by:

1. Increasing the Security Council Safe's signing threshold, from 4 to 10, out
2. of 13 owners. This meets the 75% threshold requirement for a Stage 1 rollup
3. outlined inL2Beat's Stages framework(opens in a new tab)
4. Reassigning the role of Guardian from the Foundation to a new Guardian Safe

5. with the Security Council Safe as its sole owner. This moves the Superchain
6. closer to satisfying the 1 week exit window 5 requirement for Stage 1.* Additionally the Foundation is appointed to the new DeputyGuardian role
7.
    - which is able to act as Guardian through the Guardian Safe. This
8.
    - appointment can be revoked by the Security Council Safe at any time.
9. Reassigning the owner of the L2ProxyAdmin contract from the Foundation to
10. the Security Council. This ensures the Security Council Safe has a blocking
11. vote for L2 predeploy upgrades and is a requirement for Stage 1.
12. Official - Safe Extensions Release(opens in a new tab)
13. Governance Post(opens in a new tab)

Changelog Added:

- LivenessGuard:1.0.0(opens in a new tab)
- LivenessModule:1.2.0(opens in a new tab)
- DeputyGuardianModule:1.1.0(opens in a new tab)

No change:

- AddressManager:Latest (this has no version)(opens in a new tab)
- L1CrossDomainMessenger:2.3.0(opens in a new tab)
- L1ERC721Bridge:2.1.0(opens in a new tab)
- L1StandardBridge:2.1.0(opens in a new tab)
- OptimismMintableERC20Factory:1.9.0(opens in a new tab)
- OptimismPortal:3.10.0(opens in a new tab)
- SystemConfig:2.2.0(opens in a new tab)
- FaultDisputeGame:1.2.0(opens in a new tab)
- PermissionedDisputeGame:1.2.0(opens in a new tab)
- DisputeGameFactory:1.0.0(opens in a new tab)
- AnchorStateRegistry:1.0.0(opens in a new tab)
- DelayedWETH:1.0.0(opens in a new tab)
- MIPS:1.0.1(opens in a new tab)
- PreimageOracle:1.0.0(opens in a new tab)
- SuperchainConfig:1.1.0(opens in a new tab)
- ProtocolVersions:1.0.0(opens in a new tab)

op-contracts/v1.4.0 - Fault proofs

This protocol upgrade reduces the trust assumptions for users of the OP Stack by enabling permissionless output proposals and a permissionless fault proof system. As part of a responsible and safe rollout of Fault Proofs, it preserves the ability for the guardian to override if necessary to maintain security.

As a result, withdrawals no longer depend on the privileged proposer role posting an output root, allowing the entire withdrawal process to be completed without any privileged actions. The trust assumption is reduced to requiring only that the guardian role does not act to intervene.

Combined with the Guardian, Security Council Threshold and L2 ProxyAdmin Ownership changes 23 proposals, this satisfies the criteria to have OP Chains reach Stage 1 status.

- Official - Fault Proof Release(opens in a new tab)
- Governance Post(opens in a new tab)

Changelog Added:

- FaultDisputeGame:1.2.0(opens in a new tab)
- PermissionedDisputeGame:1.2.0(opens in a new tab)
- DisputeGameFactory:1.0.0(opens in a new tab)
- AnchorStateRegistry:1.0.0(opens in a new tab)
- DelayedWETH:1.0.0(opens in a new tab)
- MIPS:1.0.1(opens in a new tab)
- PreimageOracle:1.0.0(opens in a new tab)

Upgraded:

- OptimismPortal:3.10.0(opens in a new tab)

- SystemConfig:2.2.0(opens in a new tab)

Deprecated:

- L2OutputOracle: 1.8.0

No change:

- AddressManager:Latest (this has no version)(opens in a new tab)
- L1CrossDomainMessenger:2.3.0(opens in a new tab)
- L1ERC721Bridge:2.1.0(opens in a new tab)
- L1StandardBridge:2.1.0(opens in a new tab)
- OptimismMintableERC20Factory:1.9.0(opens in a new tab)
- SuperchainConfig:1.1.0(opens in a new tab)
- ProtocolVersions:1.0.0(opens in a new tab)

op-contracts/v1.3.0 - Multi-Chain Prep (MCP)

This is the current recommended contract release for new production chains. This protocol upgrade strengthens the security and upgradeability of the Superchain by enabling L1 contracts to be upgraded atomically across multiple chains in a single transaction. This upgrade also extends theSystemConfig to contain the addresses of the contracts in the network, allowing users to discover the system's contract addresses programmatically.

- Official - MCP Release(opens in a new tab)
- Governance Post(opens in a new tab)

Changelog Upgraded:

- OptimismPortal:2.5.0(opens in a new tab)
- L1CrossDomainMessenger:2.3.0(opens in a new tab)
- L1StandardBridge:2.1.0(opens in a new tab)
- L1ERC721Bridge:2.1.0(opens in a new tab)
- OptimismMintableERC20Factory:1.9.0(opens in a new tab)
- L2OutputOracle:1.8.0(opens in a new tab)
- SystemConfig:1.12.0(opens in a new tab)

No change:

- AddressManager:Latest (this has no version)(opens in a new tab)
- SuperchainConfig:1.1.0(opens in a new tab)
- ProtocolVersions:1.0.0(opens in a new tab)

op-contracts/v1.2.0 - SuperchainConfig with Extended Pause Functionality

The SuperchainConfig contract is used to manage global configuration values for multiple OP Chains within a single Superchain network.

- Official - SuperchainConfig and Extended Pause Release(opens in a new tab)
- Governance post(opens in a new tab)

Changelog Added:

- SuperchainConfig:1.1.0(opens in a new tab)

Upgraded:

- AddressManager:Latest (this has no version)(opens in a new tab)
- L1CrossDomainMessenger:2.2.0(opens in a new tab)
- L1ERC721Bridge:2.0.0(opens in a new tab)
- L1StandardBridge:2.0.0(opens in a new tab)
- L2OutputOracle:1.7.0(opens in a new tab)
- OptimismMintableERC20Factory:1.8.0(opens in a new tab)
- OptimismPortal:2.4.0(opens in a new tab)
- SystemConfig:1.11.0(opens in a new tab)
- ProtocolVersions:1.0.0(opens in a new tab)

No change:

- AddressManager: Latest (this has no version)

op-contracts/v1.1.0 - ProtocolVersions

The Protocol Version documents the progression of the total set of canonical OP Stack specifications. Components of the OP Stack implement the subset of their respective protocol component domain, up to a given Protocol Version of the OP Stack.

The Protocol Version is NOT a hardfork identifier, but rather indicates software-support for a well-defined set of features introduced in past and future hardforks, not the activation of said hardforks.

The Protocol Version only applies to the Protocol specifications with the Superchain Targets(opens in a new tab) specified within. This versioning is independent of the Semver versioning used in OP Stack smart contracts, and the Semver-versioned reference software of the OP-Stack. This is an optional feature. * Official ProtocolVersions Release(opens in a new tab)

Changelog Added:

- ProtocolVersions: 1.0.0

No change:

- AddressManager:Latest (this has no version)(opens in a new tab)
- L1CrossDomainMessenger:1.4.0(opens in a new tab)
- L1ERC721Bridge:1.1.1(opens in a new tab)
- L1StandardBridge:1.1.0(opens in a new tab)
- L2OutputOracle:1.3.0(opens in a new tab)
- OptimismMintableERC20Factory:1.1.0(opens in a new tab)
- OptimismPortal:1.6.0(opens in a new tab)
- SystemConfig:1.3.0(opens in a new tab)

op-contracts/v1.0.0 - Bedrock

The Bedrock protocol upgrade was designed to minimize the amount of code in the OP Stack, pushes it as close as possible to Ethereum-Equivalence, and most importantly making the stack modular.

- Official - Bedrock Release(opens in a new tab)
- ...
- Governance post(opens in a new tab)

Changelog Added:

- AddressManager:Latest (this has no version)(opens in a new tab)
- L1CrossDomainMessenger:1.4.0(opens in a new tab)
- L1ERC721Bridge:1.1.1(opens in a new tab)
- L1StandardBridge:1.1.0(opens in a new tab)
- L2OutputOracle:1.3.0(opens in a new tab)
- OptimismMintableERC20Factory:1.1.0(opens in a new tab)
- OptimismPortal:1.6.0(opens in a new tab)
- SystemConfig:1.3.0(opens in a new tab)

L1 Contract Details

AddressManager

AddressManager is a legacy contract that was used in the old version of the Optimism system to manage a registry of string names to addresses. We now use a more standard proxy system for most contracts, but this contract is still used for the L1CrossDomainMessenger, via ResolvedDelegateProxy

SuperchainConfig

TheSuperchainConfig contract is used to manage configuration of global superchain values. It has the ability to pause and unpause all withdrawals in the Superchain.

L1CrossDomainMessenger

TheL1CrossDomainMessenger is a message passing interface between L1 and L2 responsible for sending and receiving data on the L1 side. Users are encouraged to use this interface instead of interacting with lower-level contracts directly.

L1ERC721Bridge

TheL1ERC721bridge is a contract which works together with theL2ERC721Bridge to make it possible to transfer ERC721 tokens from Ethereum to OP Mainnet. This contract acts as an escrow for ERC721 tokens deposited into L2.

L1StandardBridge

⚠ This contract is not intended to support all variations of ERC20 tokens. Examples of some token types that may not be properly supported by this contract include, but are not limited to: tokens with transfer fees, rebasing tokens, and tokens with blocklists. TheL1StandardBridge is responsible for transferring ETH and ERC20 tokens between L1 and L2. In the case that an ERC20 token is native to L1, it will be escrowed within this contract. If the ERC20 token is native to L2, it will be burnt.

OptimismPortal

TheOptimismPortal is a low-level contract responsible for passing messages between L1 and L2. Messages sent directly to theOptimismPortal have no form of replayability. Users are encouraged to use theL1CrossDomainMessenger for a higher-level interface.

ProtocolVersions

TheProtocolVersions contract is used to manage Superchain protocol version information. It exposes a recommended and required version for node operators. The recommended/required versions are changed with each hard fork. Nodes can optionally halt if using the wrong version.

SystemConfig

TheSystemConfig contract is helps manage configuration of an OP Stack network. Much of the network's configuration is stored on L1 and picked up by L2 as part of the derivation of the L2 chain. The contract also contains references to all other contract addresses for the chain.

DisputeGameFactory

TheDisputeGameFactory contract generatesFaultDisputeGame contract instances.

FaultDisputeGame

TheFaultDisputeGame contract is deployed by theDisputeGameFactory . Each deployed instance of the contract acts as a host to a proposal about the state of the OP Stack chain at a given block number.

PermissionedDisputeGame

TheGuardian can also choose to revert to aPermissionedDisputeGame contract that only allows specific roles to submit and challenge proposals.

AnchorStateRegistry

TheAnchorStateRegistry is a contract that stores the latest "anchor" state for each availableFaultDisputeGame type. The anchor state is the latest state that has been proposed on L1 and was not challenged within the challenge period. By using stored anchor states, newFaultDisputeGame instances can be initialized with a more recent starting state which reduces the amount of required offchain computation.

DelayedWETH

TheDelayedWETH contract has been set up to hold the bonds and only allow payouts after a delay, so that bonds can be redirected towards the correct recipient in the case that a game resolves incorrectly.

MIPS

TheMIPS smart contract is an onchain implementation of a virtual machine (VM) that encompasses the 32-bit, Big-Endian, MIPS III Instruction Set Architecture (ISA).

PreimageOracle

ThePreimageOracle contract for storing permissioned pre-images.

DEPRECATED - L2OutputOracle

TheL2OutputOracle contains an array of L2 state outputs, where each output is a commitment to the state of the L2 chain. Other contracts like theOptimismPortal use these outputs to verify information about the state of L2.

Layer 2 Contracts (Predeploys)

Predeployed smart contracts exist at predetermined addresses in the genesis state. They are similar to precompiles but instead run directly in the EVM instead of running native code outside the EVM.

Predeploys are used instead of precompiles to make it easier for multiclient implementations as well as allowing for more integration with hardhat/foundry network forking.

WETH9

WETH9 is the standard implementation of Wrapped Ether. It is a commonly used contract and is placed as a predeploy so that it is at a deterministic address.

- Address:
- 0x4200000000000000000000000000000000000006
- Introduced:
- Legacy
- Deprecated:
- no
- Proxied:
- no

L2CrossDomainMessenger

TheL2CrossDomainMessenger is a high-level interface for message passing between L1 and L2 on the L2 side. Users are generally encouraged to use this contract instead of lower level message passing contracts.

- Address:
- 0x4200000000000000000000000000000000000007
- Introduced:
- Legacy
- Deprecated:
- no
- Proxied:
- yes

L2StandardBridge

⚠ This contract is not intended to support all variations of ERC20 tokens. Examples of some token types that may not be properly supported by this contract include, but are not limited to: tokens with transfer fees, rebasing tokens, and tokens with blocklists. TheL2StandardBridge is responsible for transferring ETH and ERC20 tokens between L1 and L2. In the case that an ERC20 token is native to L2, it will be escrowed within this contract. If the ERC20 token is native to L1, it will be burnt.

- Address:
- 0x4200000000000000000000000000000000000010
- Introduced:
- Legacy
- Deprecated:
- no
- Proxied:
- yes

SequencerFeeVault

TheSequencerFeeVault is the contract that holds any fees paid to the Sequencer during transaction processing and block production.

- Address:
- 0x4200000000000000000000000000000000000011
- Introduced:
- Legacy
- Deprecated:
- no
- Proxied:

- yes

OptimismMintableERC20Factory

TheOptimismMintableERC20Factory is responsible for creating ERC20 contracts on L2 that can be used for depositing native L1 tokens into. These ERC20 contracts can be created permissionlessly and implement the interface required by the StandardBridge to just work with deposits and withdrawals.

Each ERC20 contract that is created by theOptimismMintableERC20Factory allows for the L2StandardBridge to mint and burn tokens, depending on if the user is depositing from L1 to L2 or withdrawing from L2 to L1.

- Address:
- 0x4200000000000000000000000000000000000012
- Introduced:
- Legacy
- Deprecated:
- no
- Proxied:
- yes

GasPriceOracle

TheGasPriceOracle is no longer a permissioned contract like in the legacy system. It only exists to preserve the API for offchain gas estimation. The functiongetL1Fee(bytes) accepts an unsigned RLP transaction and will return the L1 portion of the fee. This fee pays for using L1 as a data availability layer and should be added to the L2 portion of the fee, which pays for execution, to compute the total transaction fee.

Following the Ecotone upgrade, the values used for L1 fee computation are:

- baseFeeScalar
- blobBaseFeeScalar
- decimals

These new scalar values are managed by theSystemConfig contract on the L1 by introducing a backwards compatible versioned encoding scheme of its scalars storage slot. The decimals remain hardcoded to 6, and the overhead value is ignored.

- Address:
- 0x420000000000000000000000000000000000000F
- Introduced:
- Legacy
- Deprecated:
- no
- Proxied:
- yes

GovernanceToken

The OP token used in governance and supporting voting and delegation. Implements EIP 2612 allowing signed approvals. Contract is "owned" by aMintManager instance with permission to themint function only, for the purposes of enforcing the token inflation schedule.

- Address:
- 0x4200000000000000000000000000000000000042
- Introduced:
- Legacy
- Deprecated:
- no
- Proxied:
- no

L1Block

TheL1Block predeploy gives users access to information about the last known L1 block. Values within this contract are updated once per epoch (every L1 block) and can only be set by the "depositor" account, a special system address. Depositor account transactions are created by the protocol whenever we move to a new epoch.

- Address:
- 0x4200000000000000000000000000000000000015

- Introduced:
- Bedrock
- Deprecated:
- no
- Proxied:
- yes

L2ToL1MessagePasser

TheL2ToL1MessagePasser is a dedicated contract where messages that are being sent from L2 to L1 can be stored. The storage root of this contract is pulled up to the top level of the L2 output to reduce the cost of proving the existence of sent messages.

- Address:
- 0x4200000000000000000000000000000000000016
- Introduced:
- Bedrock
- Deprecated:
- no
- Proxied:
- yes

L2ERC721Bridge

⚠ Do not bridge an ERC721 that was originally deployed on Optimism. This bridge ONLY supports ERC721s originally deployed on Ethereum. Users will need to wait for the one-week challenge period to elapse before their Optimism-native NFT can be refunded on L2. TheL2ERC721Bridge is a contract which works together with theL1ERC721Bridge to make it possible to transfer ERC721 tokens from Ethereum to Optimism. This contract acts as a minter for new tokens when it hears about deposits into theL1ERC721Bridge . This contract also acts as a burner for tokens being withdrawn.

- Address:
- 0x4200000000000000000000000000000000000014
- Introduced:
- Legacy
- Deprecated:
- no
- Proxied:
- yes

OptimismMintableERC721Factory

Factory contract for creatingOptimismMintableERC721 contracts.

- Address:
- 0x4200000000000000000000000000000000000017
- Introduced:
- Bedrock
- Deprecated:
- no
- Proxied:
- yes

ProxyAdmin

TheProxyAdmin is the owner of all of the proxy contracts set at the predeploys. It is itself behind a proxy. The owner of theProxyAdmin will have the ability to upgrade any of the other predeploy contracts.

- Address:
- 0x4200000000000000000000000000000000000018
- Introduced:
- Bedrock
- Deprecated:
- no
- Proxied:
- yes

BaseFeeVault

TheBaseFeeVault predeploy receives the base fees on L2. The base fee is not burnt on L2 like it is on L1. Once the contract

has received a certain amount of fees, the ETH can be withdrawn to an immutable address on L1.

- Address:
- 0x4200000000000000000000000000000000000019
- Introduced:
- Bedrock
- Deprecated:
- no
- Proxied:
- yes

L1FeeVault

TheL1FeeVault predeploy receives the L1 portion of the transaction fees. Once the contract has received a certain amount of fees, the ETH can be withdrawn to an immutable address on L1.

- Address:
- 0x420000000000000000000000000000000000001a
- Introduced:
- Bedrock
- Deprecated:
- no
- Proxied:
- yes

SchemaRegistry

TheSchemaRegistry predeploy implements the global attestation schemas for the Ethereum Attestation Service protocol.

- Address:
- 0x4200000000000000000000000000000000000020
- Introduced:
- Bedrock
- Deprecated:
- no
- Proxied:
- yes

EAS

TheEAS predeploy implements the Ethereum Attestation Service protocol.

- Address:
- 0x4200000000000000000000000000000000000021
- Introduced:
- Bedrock
- Deprecated:
- no
- Proxied:
- yes

BeaconBlockRoot

TheBeaconBlockRoot predeploy provides access to the L1 beacon block roots. This was added during the Ecotone network upgrade and is specified inEIP-4788(opens in a new tab).

- Address:
- 0x000F3df6D732807Ef1319fB7B8bB8522d0Beac02
- Introduced:
- Ecotone
- Deprecated:
- no
- Proxied:
- no

DEPRECATED Contracts

DEPRECATED - LegacyMessagePasser

TheLegacyMessagePasser was the low-level mechanism used to send messages from L2 to L1 before the Bedrock upgrade.

- Address:
- 0x4200000000000000000000000000000000000000
- Introduced:
- Legacy
- Deprecated:
- yes
- Proxied:
- yes

DEPRECATED - DeployerWhitelist

DeployerWhitelist is a legacy contract that was originally used to act as a whitelist of addresses allowed to the Optimism network. TheDeployerWhitelist has since been disabled, but the code is kept in state for the sake of full backwards compatibility. As of the Bedrock upgrade, theDeployerWhitelist is completely unused by the Optimism system and could, in theory, be removed entirely.

- Address:
- 0x4200000000000000000000000000000000000002
- Introduced:
- Legacy
- Deprecated:
- yes
- Proxied:
- yes

DEPRECATED - LegacyERC20ETH

LegacyERC20ETH is a legacy contract that held ETH balances before the Bedrock upgrade. All ETH balances held within this contract were migrated to the state trie as part of the Bedrock upgrade. Functions within this contract that mutate state were already disabled as part of the EVM equivalence upgrade.

- Address:
- 0xDeadDeAddeAddEAddeadDEaDDEAdDeaDDeAD0000
- Introduced:
- Legacy
- Deprecated:
- yes
- Proxied:
- yes

DEPRECATED - L1BlockNumber

L1BlockNumber is a legacy contract that fills the role of theOVM_L1BlockNumber contract in the old version of the Optimism system. Only necessary for backwards compatibility. If you want to access the L1 block number going forward, you should use theL1Block contract instead.

- Address:
- 0x4200000000000000000000000000000000000013
- Introduced:
- Legacy
- Deprecated:
- yes
- Proxied:
- yes