Glossary

The glossary provides definitions of important terminology used throughout the Optimism Developer Documentation. To make revisions to this page, pleasesubmit an issue(opens in a new tab).

Table of Contents

General Terms

Address Aliasing

When a contract submits adeposit from L1 to L2, it's address (as returned byORIGIN andCALLER) will be aliased with a modified representation of the address of a contract.

Block

A sequential list of transactions, along with a couple of properties stored in the "'header"' of the block; can refer to al1 block, or to anL2 block, which are structured similarly. It is useful to distinguish between input block properties, which are known before executing the transactions in the block, and output block properties, which are derived after executing the block's transactions (e.g.,Merkle Patricia Trie roots).

Block Time

L2 block time is 2 seconds, meaning there is an L2 block at every 2stime slot .Post-merge , it could be said the that L1 block time is 12s as that is the L1time slot . However, in reality the block time is variable as some time slots might be skipped. Pre-merge, the L1 block time is variable, though it is on average 13s.

EOA or Externally Owned Account

Ethereum term to designate addresses operated by users, as opposed to contract addresses.

Execution Engine

Responsible for executing transactions in blocks and computing the resulting state roots, receipts roots, and block hash. Both L1 (post-merge) and L2 have an execution engine. On L1, the executed blocks can come from L1 block synchronization or from a block freshly minted by the execution engine (using transactions from the L1 mempool), at the request of the L1 consensus layer. On L2, the executed blocks are freshly minted by the execution engine at the request of therollup node , using transactionsderived from L1 blocks .

Optimism Collective

The Optimism Collective is a band of people, projects, and companies working together to build a better economy for everyone, united by a mutually beneficial pact to adhere to the axiom of impact=profit — the principle that positive impact to the collective should be rewarded with profit to the individual. New model of digital democratic governance optimized to drive rapid and sustained growth of a decentralized ecosystem.

OP Token

A governance token, referred to as "OP." Content should not discuss the token price or speculate on the price, and content that refers to OP incorrectly will be removed from Optimism platforms and will not be eligible for promotion.

Layer 1 (L1)

Refers the Ethereum blockchain, used in contrast tolayer 2 , which refers to Optimism.

Layer 2 (L2)

Refers to the Optimism blockchain and is used in contrast tolayer 1 , which refers to the Ethereum blockchain.

L2 Output Root

32 byte value which serves as a commitment to the current state of the L2 chain.

L2 Output Oracle Contract

L1 contract to which L2 output roots are posted by the sequencer .

Predeployed Contract

A contract placed in the L2 genesis state (i.e. at the start of the chain).

PGA or Priority Gas Auction

Transactions in ethereum are ordered by the price that the transaction pays to the miner. Priority Gas Auctions (PGAs) occur when multiple parties are competing to be the first transaction in a block. Each party continuously updates the gas price of their transaction. PGAs occur when there is value in submitting a transaction before other parties (like being the first deposit or submitting a deposit before there is not more guaranteed gas remaining). PGAs tend to have negative externalities on the network due to a large amount of transactions being submitted in a very short amount of time.

Receipt

The output generated by a transaction, comprising a status code, the amount of gas used, a list of log entries, and a bloom filter(opens in a new tab) indexing these entries. Log entries are most notably used to encode Solidity events. Receipts are not stored in blocks, but blocks store a Merkle Patricia Trie root for a tree containing the receipt for every transaction in the block.

Rollup Node

The node responsible for deriving the L2 chain from the L1 chain (L1 blocks and their associated receipts . The rollup node can run either in validator or sequencer mode.

Sequencer mode

The rollup node receives L2 transactions from users, which it uses to create L2 blocks. These are then submitted to a data availability provider via batch submission . The L2 chain derivation then acts as a sanity check and a way to detect L1 chain re-orgs.

Time Slot

On L2, there is a block every 2 second (this duration is known as the block time). We say that there is a "time slot" every multiple of 2s after the timestamp of the L2 genesis block . On L1, post-merge, the time slots are every 12s. However, an L1 block may not be produced for every time slot, in case of even benign consensus issues.

Transaction Type

Ethereum provides a mechanism as described in EIP-2718(opens in a new tab) for defining different transaction types. Different transaction types can contain different payloads and be handled differently by the protocol.

Validator

An entity (individual or organization) that runs a rollup node in validator mode. Doing so grants a lot of benefits similar to running an Ethereum node, such as the ability to simulate L2 transactions locally, without rate limiting. It also lets the validator verify the work of the sequencer , by re-deriving output roots and comparing them against those submitted by the sequencer. In case of a mismatch, the validator can perform a fault proof .

Validator mode

The rollup node performs derivation as indicated in sequencer mode, but is also able to "run ahead" of the L1 chain by getting blocks directly from the sequencer, in which case derivation serves to validate the sequencer's behavior. A rollup node running in validator mode is sometimes calleda replica .

[return to top]

OP Profile & Identity

Attestations

Statements or evidence of information made by anyone about anything.

Attestation recipient

The entity or individual that receives theattestation from the attester. In decentralized identity, the attestation recipient can be a service provider, website, or any other entity that requires verification of a person's identity or personal information.

Attestation issuer

The entity or individual that performs the attestation process and issues theattestation . In decentralized identity, the attestation issuer can be a government agency, financial institution, or any other trusted entity that is authorized to verify a person's identity or personal information.

Attestation verifier

The entity or individual that verifies theattestation and ensures that it is valid and accurate. In decentralized identity, the attestation verifier can be a service provider, website, or any other entity that requires verification of a person's identity or personal information.

Decentralized identity

A system that enables individuals to have greater control and ownership over their personal data and identity. In decentralized identity, personal information is stored on a blockchain or other decentralized system, and individuals have the ability to grant or revoke access to their data as they see fit. This allows for greater privacy, security, and control over personal information.

Ethereum Attestation Service (EAS)

An Ethereum infrastructure public good for making attestations on or off-chain about anything.

Sybil-resistance

A defense mechanism that prevents someone from creating multiple fake identities in a decentralized identity system.

Web of trust

A decentralized model used to establish trust among participants in a network. It relies on the concept of participants vouching for the authenticity or trustworthiness of other participants. In a web of trust, trust is built through the accumulation of attestations from trusted individuals.

[return to top]

OP Stack & OP Superchain

Attestation-Based Fault Proof

A fault proof where challenges can be successfully made by supplying anattestation proof which disagrees with the original withdrawal claim.

Attestation-Based Validity Proof

A validity proof which can be verified by supplying anattestation proof which agrees with the withdrawal claim.

Attestation Proof

A proof which consists of some number of signatures from a pre-agreed upon set of chain attestors.

Cannon Fault Proof

A fault proof where challenges are evaluated using an onchain game which is guaranteed to result in a truthful outcome, given economic rationality assumptions.

Chain

A statetransition system(opens in a new tab) consisting of an initial state, a state transition function, and a list of inputs (transactions)—which is cryptographically committed to and can be independently replicated with commodity computer hardware and internet connection.

Chain Proof

Difficult to forge evidence of the validity of a particular withdrawal claim. Proofs are commonly used to enable chains to communicate with each other.

Challenge Period

The window of time in which a challenge can be made to disprove a fault proof.

Fault Proof

A proof which relies on the absence of counter-evidence to prove correctness.

L1 Origin

the L1 origin of an L2 block is the L1 block corresponding to itssequencing epoch .

Merkle Patricia Trie

sparsetrie(opens in a new tab) , which is a tree-like structure that maps keys to values. The root hash of a MPT is a commitment to the contents of the tree, which allows a proof to be constructed for any key-value mapping encoded in the tree. Such a proof is called a Merkle proof, and can be verified against the Merkle root.

Modular Proof

The ability to use multiple proof systems for the same OP Chain. For instance, it should be possible to prove an OP Chain using a fault proof or a validity proof.

Modular Sequencing

The ability to configure the sequencer address during OP Chain deployment. This value can be configured by the OP Chain deployer.

OP Chain

An individual chain within the Optimism Superchain. All chains, regardless of their specific properties are considered OP Chains if they are officially governed by the Optimism Collective, and therefore part of the Superchain.

OP Mainnet

Layer 2 blockchain powered by the OP Stack. Previously known as just "Optimism,"OP Mainnet is where it all started, and the first chain to commit to the Superchain vision.

OP Stack

the modular, open source, MIT-licensed development stack that powers the OP Mainnet, OP Chains, and, in the future, the Superchain. The OP Stack is maintained by the Optimism Collective.

OP Stack Fork

Layer 2 blockchain that has been built using the MIT-licensed OP Stack, but is not governed by Optimism's governance or contributing sequencer revenue back to the Collective (and therefore is not on track to become part of the Superchain). This means OP Stack Forks won't necessarily share security or interoperability with OP Chains in the Superchain.

Plasma Chain

A chain where transaction data is committed to on L1 but not supplied to L1 directly, with a data availability challenge fallback.

Rollup Chain

A chain where all transaction data is submitted to L1.

Rollup transactions

Rollup transactions can be included in two ways: through a deposited transaction enforced by the system and through a regular transaction embedded in a sequencer batch . Submitting transactions for inclusion in a batch saves costs by reducing overhead, and enables the sequencer to pre-confirm the transactions before the L1 confirms the data.

Sequencer

The specific entity or smart contract which has priority when submitting transactions to an OP Chain, can be either a rollup node ran in sequencer mode or the operator of this rollup node. The sequencer receives L2 transactions from L2 users, creates L2 blocks using them, which it then submits to data availability provider (via a batcher). It also submits output roots to L1.

Sequencing Window

Range of L1 blocks from which a sequencing epoch can be derived. A sequencing window whose first L1 block has number N contains batcher transactions for epoch N . The window contains blocks (N, N + SWS) where SWS is the sequencer window size. Additionally, the first block in the window defines the depositing transactions which determine the deposits to be included in the first L2 block of the epoch.

Sequencing Epoch

sequential range of L2 blocks derived from a sequencing window of L1 blocks. Each epoch is identified by an epoch number, which is equal to the block number of the first L1 block in the sequencing window. Epochs can have variable size, subject to some constraints.

Superchain

A decentralized blockchain platform which consists of many chains that share security and a technology stack (OP Stack). The interoperability and standardization enables individual chains to be treated identically by tools and wallets.

Superchain Bridge

The L1 bridge contracts which govern all OP Chains in the Superchain. This bridge can be upgraded by the Optimism Collective.

Validity Proof

A proof of a withdrawal claim which can be immediately validated, without a challenge period.

Withdrawal Claim

A claim about the state of one chain made on another chain. For instance, I can claim that in OP Mainnet I have burned my tokens with the intent to withdraw those tokens back to L1.

Zero Knowledge Proof

A validity proof which relies on cryptographic properties and low error margins.

[return to top]

Protocol

Batcher

Software component (independent program) that is responsible to make channels available on a data availability provider. The batcher communicates with the rollup node in order to retrieve the channels. The channels are then made available using batcher transactions .

Batcher Transaction

Transaction submitted by a batcher to a data availability provider, in order to make channels available. These transactions carry one or more full frames, which may belong to different channels. A channel's frame may be split between multiple batcher transactions. When submitted to Ethereum calldata, the batcher transaction's receiver must be the sequencer inbox address. The transaction must also be signed by a recognized batch submitter account.

Channel

Sequence ofsequencer batches (for sequential blocks) compressed together; uniquely identified by its timestamp (UNIX time at which the channel was created) and a random value. The reason to group multiple batches together is simply to obtain a better compression rate, hence reducing data availability costs.

- A channel can be split inframes
- in order to be transmitted viabatcher transactions
- . The reason to split a channel
- into frames is that a channel might be too large to include in a single batcher transaction.
- On the side of therollup node
- (which is the consumer of channels), a channel is considered to beopened
- if its final frame (explicitly marked as such) has not been read, or closed otherwise.

Channel Frame

Chunk of data belonging to achannel .Batcher transactions carry one or multiple frames. The reason to split a channel into frames is that a channel might too large to include in a single batcher transaction.

Channel Timeout

Duration (in L1 blocks) during whichchannel frames may land on L1 withinbatcher transactions . The acceptable time range for the frames of achannel is[channel_id.timestamp, channel_id.timestamp + CHANNEL_TIMEOUT] . The acceptable L1 block range for these frames are any L1 block whose timestamp falls inside this time range. (Note thatchannel_id.timestamp must be lower than the L1 block timestamp of any L1 block in which frames of the channel are seen, or else these frames are ignored.)

The purpose of channel timeouts is dual:

- Avoid keeping old unclosed channel data around forever (an unclosed channel is a channel whose final frame was not
- sent).
- Bound the number of L1 blocks we have to look back in order to decodesequencer batches
- from
- channels. This is particularly relevant during L1 re-orgs.

Data Availability

Guarantee that some data will be "available" (i.e.retrievable) during a reasonably long time window. In Optimism's case, the data in question aresequencer batches thatvalidators needs in order to verify the sequencer's work and validate the L2 chain. Thefinalization period should be taken as the lower bound on the availability window, since that is when data availability is the most crucial, as it is needed to perform afault proof . "Availability"does not mean guaranteed long-term storage of the data.

Data Availability Provider

Service that can be used to makedata available . Ideally, a good data availability provider provides strongverifiable guarantees of data availability, such as Ethereum call data and EIP4844.

Deposit

L2 transaction derived from an L1 block (by the rollup driver). While transaction deposits are notably (but not only) used to "deposit" (bridge) ETH and tokens to L2, the worddeposit should be understood as "a transactiondeposited to L2 from L1".

Deposit contract

L1 contract to whichEOAs and contracts may send deposits. The deposits are emitted as log records (in Solidity, these are calledevents) for consumption byrollup nodes .

⚠ The deposits are not stored in calldata because they can be sent by contracts, in which case the calldata is part of theinternal execution between contracts, and this intermediate calldata is not captured in one of theMerkle Patricia Trie roots included in the L1 block.

Deposited Transaction

L2 transaction that was derived from L1 and included in a L2 block. There are two kinds of deposited transactions: L1 attributes deposited transaction (submits the L1 block's attributes to the L1 Attributes Predeployed Contract) and User deposited transactions (transactions derived from an L1 call to the deposit contract).

Depositor

L1 address (contract or [EOA]) that makes (is themsg.sender of) thedepositing call . Thedepositor isNOT the originator of thedepositing transaction (i.e.tx.origin).

Depositing Call

L1 call to the deposit contract, which will be derived to auser-deposited transaction by the rollup driver. This call specifies all the data (destination, value, calldata, ...) for the deposited transaction.

Depositing Transaction

L1 transaction that makes one or moredepositing calls .

Deposited transaction type

EIP-2718 transaction type, which specifies the input fields and correct handling of a deposited transaction.

Event or log data

Data generated by thedepositing call and read by the rollup driver to derive thedeposited transaction .

Finalized L2 Head

Highest L2 block that can be derived fromfinalized L1 blocks — i.e. L1 blocks older than two L1 epochs (64 L1time slots).

Finalization Period

The minimum amount of time (in seconds) that must elapse before a withdrawal can be finalized, sometimes calledwithdrawal delay . The finalization period is necessary to afford sufficient time for validators to make a fault proof.

L1 Attributes Deposited Transaction

Adeposited transaction that is used to register the L1 block attributes (number, timestamp, ...) on L2 via a call to theL1 Attributes Predeployed Contract . That contract can then be used to read the attributes of the L1 block corresponding to the current L2 block.

L1 Attributes Predeployed Contract

Apredeployed contract on L2 that can be used to retrieve the L1 block attributes of L1 blocks with a given block number or a given block hash.

L2 Chain Derivation

The process that readsL2 derivation inputs from L1 in order to derive the L2 chain.

L2 Chain Inception

L1 block number at which the output roots for thegenesis block were proposed on theoutput oracle contract. In the current implementation, this is the L1 block number at which the output oracle contract was deployed or upgraded.

L2 Derivation Inputs

Data that is found in L1 blocks and is read by therollup node to constructpayload attributes . L2 derivation inputs include: L1 block attributes (block number, timestamp, basefee), deposits (as log data), sequencer batches (as transaction data), and system configuration updates (as log data).

Payload Attributes

Object that can be derived fromL2 chain derivation inputs found on L1, which are then passed to theexecution engine to construct L2 blocks. The payload attributes object essentially encodes ablock without output properties.

Relayer

EOA on L1 which finalizes a withdrawal by submitting the data necessary to verify its inclusion on L2.

Safe L2 Block

L2 block that can be derived entirely from L1 by a rollup node . This can vary between different nodes, based on their view of the L1 chain.

Safe L2 Head

Highest safe L2 block that a rollup node knows about.

Sequencer Batch

List of L2 transactions (that were submitted to a sequencer) tagged with an epoch number and an L2 block timestamp (which can trivially be converted to a block number, given our block time is constant). Sequencer batches are part of the L2 derivation inputs . Each batch represents the inputs needed to build one L2 block (given the existing L2 chain state) — except for the first block of each epoch, which also needs information about deposits (cf. the section on L2 derivation inputs).

System Configuration

Collection of dynamically configurable rollup parameters maintained by the SystemConfig (opens in a new tab) contract on L1 and read by the L2 derivation process. These parameters enable keys to be rotated regularly and external cost parameters to be adjusted without the network upgrade overhead of a hardfork.

Unsafe L2 Block

L2 block that a rollup node knows about, but which was not derived from the L1 chain. In sequencer mode, this will be a block sequenced by the sequencer itself. In validator mode, this will be a block acquired from the sequencer via unsafe sync .

Unsafe L2 Head

Highest unsafe L2 block that a rollup node knows about.

Unsafe Block Consolidation

Process through which the rollup node attempts to move the safe L2 head a block forward, so that the oldest unsafe L2 block becomes the new safe L2 head. In order to perform consolidation, the node verifies that the payload attributes derived from the L1 chain match the oldest unsafe L2 block exactly.

Unsafe Sync

Process through which a validator learns about unsafe L2 blocks from the sequencer . These unsafe blocks will later need to be confirmed by the L1 chain (via unsafe block consolidation).

User-Deposited Transaction

Deposited transaction which is derived from an L1 call to the deposit contract and depositing call ; explicitly excludes L1 attributes deposited transactions .

Withdrawal transaction

Sent from L2 to L1 that may transfer data and/or value; these "transactions" exist at multiple levels (see withdrawal initiating transaction and withdrawal finalizing transaction for details).

Withdrawal initiating transaction

A specific transaction on L2 sent to the Withdrawals predeploy.

Withdrawal finalizing transaction

A specific L1 transaction which finalizes and relays the withdrawal.

[return to top]