How does AltLayer's in-house rollup stack work?

AltLayer is designed as a modular scaling framework to fit the various application needs with configurations for different requirements. The system is compatible with EVM as well as WASM and supports all contracts originating from Ethereum, or EVM/WASM-compatible chains. AltLayer decouples the life-cycle of a transaction execution into multiple stages, i.e., transaction aggregation, block production, and block verification.

Aggregators

The aggregators receive transactions from different channels, e.g., browser extensions, mobile wallets and dApp backends. They are organized in a decentralized manner to avoid single point of failures. They aggregate transactions and send them in batches to different block producers (like load balancers) instead of overwhelming a single block producer. Depending on the requirements from dApps, it is also possible to customize the configuration of aggregators. For instance, if a dApp developer does not wish to enforce load balancing, the system setup can be simplified by letting block producers accept transactions directly from end users and thereby make aggregators optional. Aggregators also timestamp the transactions they receive and sequence them before forwarding to the block producers for execution. If there is more than one aggregator, then, instead of attempting to synchronize a global sequence across all the aggregators, we instead assume that each individual aggregator has a source of trusted time. The sequence numbers generated by each aggregator can then be used instead to identify if there are any missing transactions. As a result, there is no need to have a global consensus on what the sequence is, rather, each node only needs to synchronize the time at appropriate checkpoints and then proceed to keep track of it individually.

Block Producers

Block producers' main job is to collaboratively select the order of the transaction using non-repudiable information from the aggregators. They are also responsible for executing the transactions and updating the network state. A new block produced by the block producer represents an optimistic view of block progression akin to optimistic execution in optimistic rollups. By default, AltLayer will run with a single producer architecture to minimize transaction latency. However, if a rollup owner prefers to have a more decentralized system and is willing to make the trade-off in terms of transaction latency, more block producers can be added. Block producers run GRANDPA consensus protocol.

Verifiers

Verifiers' main job is to verify the validity of blocks generated by the block producers. Additionally, they also update the underlying Layer 1 of the block progression by periodically creating a pack . Before a pack can be created, the verifier needs to produce additional information to be included in the pack. The additional information is to enable optimizations that can be made for on-chain verification. For example to reduce the verification work from a block to a transaction, Merkle root hashes for the required states will have to be generated for each of the transaction rather than just for the entire block. While producing a block, the block producers generate the minimum amount of information needed by the verifiers to work with. This is to ensure the timeliness of the blocks that are generated and to improve the overall throughput. Any additional information (e.g., instruction level state hashes) that is needed for on-chain verification is left to the verifier to produce. Similarly, to be able to reduce the verification to a single instruction, the root hashes have to be generated for each instruction rather than per transaction. The inclusion of these intermediate state hashes allows the pack to be bisected to the appropriate level of granularity. The instruction state Merkle root hash will include all instructions that need to be executed for a particular pack.

Accountability of Verifiers

In order to detect and penalize a malicious verifier that submits an invalid state root hash on the underlying chain, AltLayer comes with a fraud proof and dispute resolution mechanism. For fraud proof and dispute resolution, on-chain dispute resolution is done via the bisection approach by locating the exact contentious instruction, which leads to a minimum on-chain cost to verify the fraud proof. After the challenge period has elapsed with no successful disputes made to it, the block will be considered finalized. Note that instead of waiting for the challenge period to be over, individual users can proceed to verify the block themselves to persuade themselves of the finality of the block without having to wait for the challenge period to be over.

Block Finalization

Definitions of a finalized block might differ across blockchains. For some, a finalized block is deterministic, or truly final. That is, a finalized block will not be reverted unless due to interventions external to the system like a hard fork. For others, a finalized block might simply be probabilistic, where any finalized block will always have a chance of being reverted (however small) when a longer or heavier chain comes along, though the probability of that happening grows increasingly smaller the longer the chain is. Blockchains typically only differentiate between blocks that have been finalized and blocks that have not. AltLayer on the other hand, uses a tiered methodology for block finalization. The tiered methodology allows an end user to decide on the finality status of a transaction give her security budget . Finalization consists of three different tiers: Execution-

level (for low security budget, Verification-level (for medium security budget) and Rollup-level (for high security budget). As the block progresses through the different tiers of finalization, there is an increasing level of confidence on the finality of the block. The level of confidence is lowest at the execution-level and is the highest at the rollup-level. Execution-level Finalization: As discussed earlier, transactions get collected from the transaction pool and packed into blocks by block producers. Block are then accepted by the consensus protocol that the block producers participate in. Blocks and transactions therein that have been accepted by the consensus protocol are said to be finalized at the execution-level . Verification-level Finalization : After blocks have been produced by block producers and have undergone the consensus protocol, they are then verified by the verifiers. Blocks that have been verified via this verification process are considered to be finalized at the verification-level . A stronger finalization is also achieved when additional verifiers can proceed to either affirm or reject the update committed by another verfier. The more affirmations a block has, the more likely it is to be correct. Note that consensus on verified blocks via a quourm of verifiers is an optional feature in AltLayer. Rollup-Level Finalization: Blocks that have passed the quorum consensus will go through a period of time open for challenges. This allows anyone to run their own verifier and participate in the verification process if they so wish to. In the event that a discrepancy is found between the challenger and the proposal on chain, the challenger may choose to make a challenge on the block. If the challenge is successful, any block found after the point of challenge will be deemed invalid. AltLayer would also have to perform a state revert up till the point of the last valid block. In the case where the challenge is unsuccessful, the stake provided by the challenger will be slashed. Previous cBridge SDK Next- AltLayer's In-House Rollup Stack in Depth Decentralized Sequencer Set Last modified5mo ago On this page Aggregators Block Producers Verifiers Accountability of Verifiers Block Finalization