

SQUAD for Decentralised Sequencing

Why Decentralized Sequencing? A central component to any rollup is the sequencer. Sequencers are nodes in the rollup network that are tasked with the following operations of a rollup: 1. 1. 2. Aggregate transactions: Accept transactions from the end users via a mempool. 3. 2. 4. Order transactions: Select transactions from the mempool and order them according to some policy. 5. 3. 6. Execute transactions: In most rollups, the sequencer is also tasked with executing the transactions according to the underlying VM. 7. 4. 8. Produce pre-confirmation: As finality for a rollup transaction ultimately comes from the underlying Layer 1, sequencers often produce pre-confirmation for a user transaction that allows clients and applications to proceed while waiting for finality from the Layer 1. 9. 5. 10. Send ordered transaction data to the DA Layer: As the last step, sequencer also commits the ordered transaction data to a DA layer, for example the underlying Layer 1. Once posted on Layer 1, transactions have Layer 1-level finality. Sequencers are therefore the heart of a rollup. Now, imagine a rollup with a single sequencer. What would happen if the sequencer were to become unavailable? Well, the network would then effectively stop processing transactions. This is not entirely a hypothetical question as there have been instances where rollups stopped processing transactions when the sequencer was down. While one could argue that even when a sequencer goes down, the network is still operational as users could instead send their Layer 2 transaction directly to Layer 1. However, those transactions can take up to 24 hours to be considered final and will cost the same fees as the underlying Layer 1 and hence would result in poor user experience. Also, a 24 hour delay may not be ideal for time-sensitive transactions like liquidation of a debt position. More generally, a centralized sequencer may: * Censor user transactions * Become a choke-point or in the worst-case could go completely unavailable * Extract excessive rent * Frontrun user transactions or more broadly create bad MEV and adversely affect users It is therefore imperative to decentralize the sequencer. Unfortunately, none of the live rollups have decentralized sequencing enabled today. Decentralized Sequencing via SQUAD AltLayer is designed in a way that application developers who wish to have a rollup can request one via the rollup launchpad and specify the number of sequencers they wish to have for their rollup. The application developers themselves do not have to worry about finding and bootstrapping the sequencers or hosting any component of the rollup themselves, thereby making the entire process of setting up a rollup with decentralized sequencers a straightforward and seamless experience. This is achieved by leveraging SQUAD which allows nodes run by EigenLayer AVS operators to register their interest to be a sequencer for rollups instantiated by our RaaS no-code dashboard. When an end user makes a request for a rollup via our RaaS platform, the user specifies the minimum and the maximum number of sequencers needed to operate the rollup and the minimum amount of economic collateral needed from each sequencer and a list of tokens in which the collateral could come from. The collateral needs to be restaked via EigenLayer and is later used to slash any misbehaving sequencer. Once the required number of sequencers have put the minimum collateral, these sequencers can then start sequencing transactions for the rollup. To facilitate discovery of sequencers, SQUAD is designed as a network of nodes open for any EigenLayer AVS operators to join and in fact a full-fledged blockchain in itself with WASM as well as EVM runtimes. The nodes in the SQUAD are called validators, typical of any PoS network. Once a validator becomes a part of the SQUAD network, it can start registering sequencing needs from the rollups and match them with sequencers. Registering as a Sequencer with the SQUAD Sequencers bring in the minimum required LST stake themselves or implement a delegated staking mechanism allowing LST holders to stake with the sequencer. Staking on the Squad is implemented natively at AVS contract. Registering a Rollup Request with the SQUAD If a developer wishes to deploy his own rollup of a certain type (a flash layer or a persistent layer) via AltLayer's rollup launchpad, he would start by registering his execution layer request to SQUAD in the form of a smart contract call to a contract named ExecutionLayerRegistryContract .

The registration request has the following JSON format. Basically, the developer specifies the settlement and the DA layer for the execution layer, the runtime (i.e., WASM, EVM, etc.) and the type of execution layer (optimistic execution with fraud proofs or ZK with validity proofs) and the purpose of the execution-layer (i.e., general purpose or app-specific). Note that the request also specifies the number of sequencers required for this execution layer. "execution_layer_req" : { "settlement_layer" : "ethereum", "da_layer": "celestia", "vm": "wasm", "type": "optimistic", "purpose": "app-specific", "nb_sequencers": "3", "allowed_tokens": [{"sym": "ALT"}, {"sym": "stSOL"}], "min_stake_per_seq": [{"sym": "ALT", val: "100"}, {"sym": "stSOL", val: "1000"}] } Selecting Sequencers for a Rollup Upon receiving a rollup request, SQUAD selects validators from those that have committed the minimum number of LSTs based on their stake and some randomness generated from a VRF. These nodes are then allowed to become sequencers for the given rollup.

Once a rollup has been instantiated, it will become possible to obtain the network information such as chainID, RPC endpoint etc. At this stage the network is fully operational and the sequencers are expected to start processing transactions. [Previous Interoperability via MACH Next- Restaked Rollups Staking of dApp Token](#) Last modified 2mo ago