Feature Proposal Program

The Feature Proposal Program provides a workflow for activation of Solana network features through community vote based on validator stake weight.

Community voting is accomplished usingSPL Tokens . Tokens are minted that represent the total active stake on the network, and distributed to all validators based on their stake. Validators vote for feature activation by transferring their vote tokens to a predetermined address. Once the vote threshold is met the feature is activated.

Background

The Solana validator software supports runtime feature activation through the built-inFeature program. This program ensures that features are activated simultaneously across all validators to avoid divergent behavior that would cause hard forks or otherwise break consensus.

Thefeature andfeature_set Rust modules are the primitives for this facility, and thesolana feature command-line subcommands allow for easy feature status inspection and feature activation.

Thesolana feature activate workflow was designed for use by the core Solana developers to allow for low-overhead addition of non-controversial network features over time.

The Feature Proposal Program provides an additional mechanism over these runtime feature activation primitives to permit feature activation by community vote when appropriate.

Source

The Feature Proposal Program's source is available onGitHub

Interface

The Feature Proposal Program is written in Rust and available oncrates.io anddocs.rs .

Command-line Utility

Thespl-feature-proposal command-line utility can be used to manage feature proposal. Once you haveRust installed , run:

cargo install spl-feature-proposal-cli Runspl-feature-proposal --help for a full description of available commands.

Configuration

Thespl-feature-proposal configuration is shared with thesolana command-line tool.

Feature Proposal Life Cycle

This section describes the life cycle of a feature proposal.

Implement the Feature

The first step is to conceive of the new feature and realize it in the Solana code base, working with the core Solana developers athttps://github.com/solana-labs/solana .

During the implementation, afeature id will be required to identify the new feature in the code base to avoid the new functionality until its activation. Thefeature id for a feature proposal is derived by running the following commands.

First create a keypair for the proposal:

solana-keygen new --outfile feature-proposal.json --silent --no-passphrase Wrote new keypair to feature-proposal.json Now run thespl-feature-proposal program to derive thefeature id :

spl-feature-proposal address feature-proposal.json Feature Id: HQ3baDfNU7WKCyWvtMYZmi51YPs7vhSiLn1ESYp3jhiA Token Mint Address: ALvA7Lv9jbo8JFhxqnRpjWWuR3aD12uCb5KBJst4uc3d Acceptance Token Address: AdqKm3mSJf8AtTWjfpA5ZbJszWQPcwyLA2XkRyLbf3Di which in this case isHQ3baDfNU7WKCyWvtMYZmi51YPs7vhSiLn1ESYp3jhiA .

HQ3baDfNU7WKCyWvtMYZmi51YPs7vhSiLn1ESYp3jhiA is the identifier that will be used in the code base and eventually will be visible in thesolana feature status command.

Note however that it is not possible to usesolana feature activate to activate this feature, as there is no private key forHQ3baDfNU7WKCyWvtMYZmi51YPs7vhSiLn1ESYp3jhiA . Activation of this feature is only possible by the Feature Proposal Program.

Initiate the Feature Proposal

After the feature is implemented and deployed to the Solana cluster, thefeature id will be visible insolana feature status and thefeature proposer may initiate the community proposal process.

This is done by running:

spl-feature-proposal propose feature-proposal.json Feature Id: HQ3baDfNU7WKCyWvtMYZmi51YPs7vhSiLn1ESYp3jhiA Token Mint Address: ALvA7Lv9jbo8JFhxqnRpjWWuR3aD12uCb5KBJst4uc3d Distributor Token Address: GK55hNft4TGc3Hg4KzbjEmju8VfaNuXK8jQNDTZKcsNF Acceptance Token Address: AdqKm3mSJf8AtTWjfpA5ZbJszWQPcwyLA2XkRyLbf3Di Number of validators: 376 Tokens to be minted: 134575791.53064314 Tokens required for acceptance: 90165780.3255309 (67%) Token distribution file: feature-proposal.csv JSON RPC URL: http://api.mainnet-beta.solana.com

Distribute the proposal tokens to all validators by running: solana-tokens distribute-spl-tokens --from GK55hNft4TGc3Hg4KzbjEmju8VfaNuXK8jQNDTZKcsNF --input-csv feature-proposal.csv --db-path db.8CyUVvio --fee-payer ~/.config/solana/id.json --owner solana-tokens spl-token-balances --mint ALvA7Lv9jbo8JFhxqnRpjWWuR3aD12uCb5KBJst4uc3d --input-csv feature-proposal.csv

Once the distribution is complete, request validators vote for the proposal. To vote, validators should first look up their token account address: spl-token --owner ~/validator-keypair.json accounts ALvA7Lv9jbo8JFhxqnRpjWWuR3aD12uCb5KBJst4uc3d and then submit their vote by running: spl-token --owner ~/validator-keypair.json transfer ALL AdqKm3mSJf8AtTWjfpA5ZbJszWQPcwyLA2XkRyLbf3Di

Periodically the votes must be tallied by running: spl-feature-proposal tally 8CyUVvio2oYAP28ZkMBPHq88ikhRgWet6i4NYsCW5Cxa Tallying is permissionless and may be run by anybody. Once this feature proposal is accepted, the HQ3baDfNU7WKCyWvtMYZmi51YPs7vhSiLn1ESYp3jhiA feature will be activated at the next epoch.

Add --confirm flag to initiate the feature proposal If the output looks good run the command again with the--confirm flag to continue, and then follow the remaining steps in the output to distribute the vote tokens to all the validators.

COST: As a part of token distribution, thefeature proposer will be financing the creation of SPL Token accounts for each of the validators. A SPL Token account requires 0.00203928 SOL at creation, so the cost for initiating a feature proposal on a network with 500 validators is approximately 1 SOL.

Tally the Votes

After advertising to the validators that a feature proposal is pending their acceptance, the votes are tallied by running:

spl-feature-proposal tally 8CyUVvio2oYAP28ZkMBPHq88ikhRgWet6i4NYsCW5Cxa Anybody may tally the vote. Once the required number of votes is tallied, the feature will be automatically activated at the start of the next epoch.

Upon a successful activation the feature will now show as activated bysolana feature status as well.Edit this page Previous Single-Validator Stake Pool Next Introduction * Background * Source * Interface * Command-line Utility * * Configuration * Feature Proposal Life Cycle * * Implement the Feature * * Initiate the Feature Proposal * * Tally the Votes