The v2 specs must include a simple multisignature proof-of-stake application, which we will use to manage validator sets on the decentralized testnets.

Basic requirements:

- Validators can register and deregister
- Validators can have associated voting weights
- New validators must be approved by a quorum of existing validators
- Validators can unilaterally deregister

Resource kinds:

- A Validator

resource has: * a consensusIdentity

used in consensus

- a operatorIdentity

used to authorize operational changes

- a consensusIdentity

used in consensus

- a operatorIdentity

used to authorize operational changes

- The Validator

resource logic checks: * if the consensusIdentity

or operatorIdentity

changes, the operatorIdentity

must sign

- if the consensusIdentity

or operatorIdentity

changes, the operatorIdentity

must sign

- A ValidatorSet

resource has: * a map of Validator

references to voting weights (unsigned integers)

- a map of Validator

references to voting weights (unsigned integers)

- The ValidatorSet

resource logic checks: * if a Validator

reference is removed, the validator’s operator identity must sign

- if a Validator

reference is added, or the voting weight changed, a 2/3 quorum of existing validators by weight must sign

- if a Validator

reference is removed, the validator’s operator identity must sign

- if a Validator

reference is added, or the voting weight changed, a 2/3 quorum of existing validators by weight must sign

There’s also a read-only transaction to retrieve the current validator set which simply iterates over the map in ValidatorSet and normalizes the weights. This validator set can then be used by the consensus engine to produce the next block.

Discuss!