Razor

Technical summary

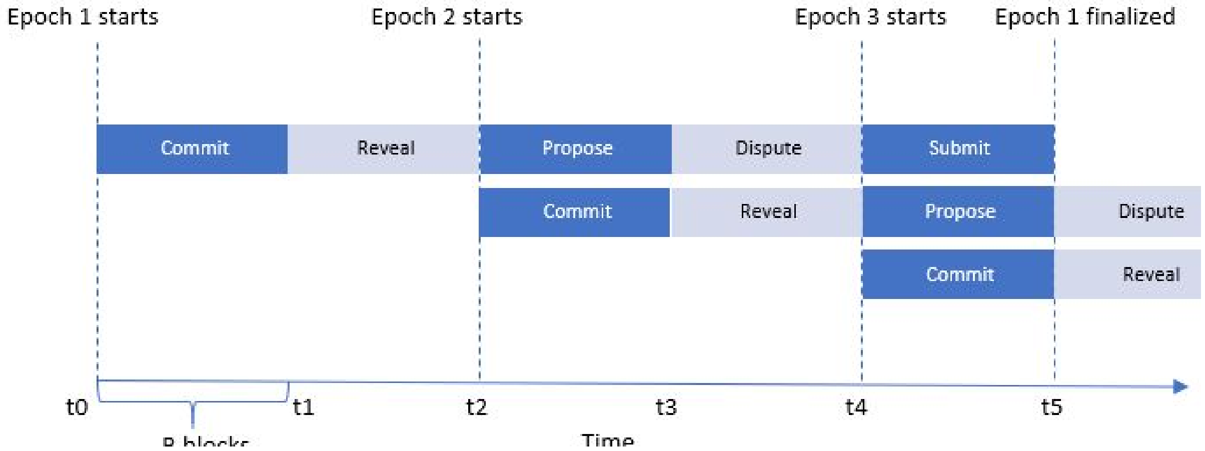
Razor network is a general purpose decentralized oracle network with the goal of offering high economic security. It’s a Proof of Stake protocol. We use commit reveal scheme to keep the individual votes secret. All stakers commit to results of ALL the jobs, but reveal only the ones assigned to them. This makes the protocol fully decentralized and extremely resilient to collusion and bribing attacks.

Stakers are pseudo-randomly selected to aggregate the results and produce a "Block". Fraud is proven by doing calculations on chain.

There is also an optional offchain trustless staking pool protocol using BLS threshold signatures.

And lastly an onchain governance protocol to protect stakers from malicious clients, and to provide URL-less long term price feeds (even if some of the datasources go down)

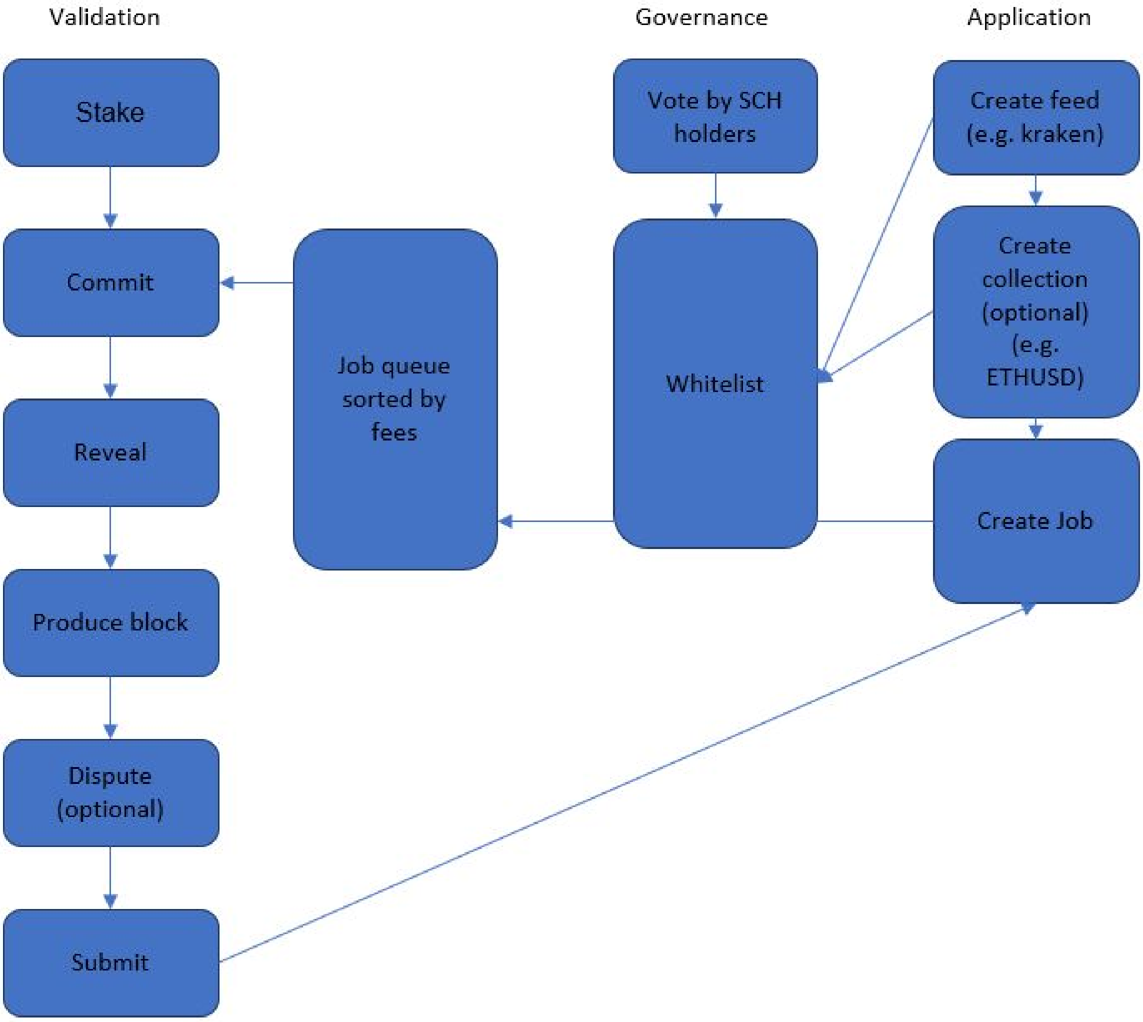
Oracle platform



1. Validators stake their schelling coins (SCH)
2. Each epoch consists of 50 blocks (12.5 minutes) (subject to change)
3. First 25 blocks are commit periods and next 25 are reveal period.
4. To stake, F schells must be burned. And minimum of S_{min} schells must be staked.
5. Reputation: 1 when you stake. Increases logarithmically.
6. If withdrawn, Reputation becomes 0.
7. Influence = Reputation * stake
8. Reputation = Log(maturity)

9. Maturity is the age of stake of deposit in epochs, but may decrease on penalty.
10. During commit period, following actions can be performed: stake, commit vote, unstake, withdraw, propose block for epoch (N-1), submit block for epoch (N-2)
11. During reveal period, following actions can be performed: reveal vote, dispute block proposed in epoch (N-1)
12. Incentives (schells): get block reward (e.g. 5 schells)
13. Penalties (schells) producing incorrect block: 100%, revealing secret in commit period: 100%.
14. Incentives and Penalties: 100% maturity penalty for voting 0*M or 2*M+. In between a quadratic curve will be used. For +/-1% for weighted median, no penalty & will be awarded reputation cut from those not in consensus.
15. Median is a weighted median of votes. Weight = influence
16. Probability of becoming block producer =influence/(influence of staker with max influence)

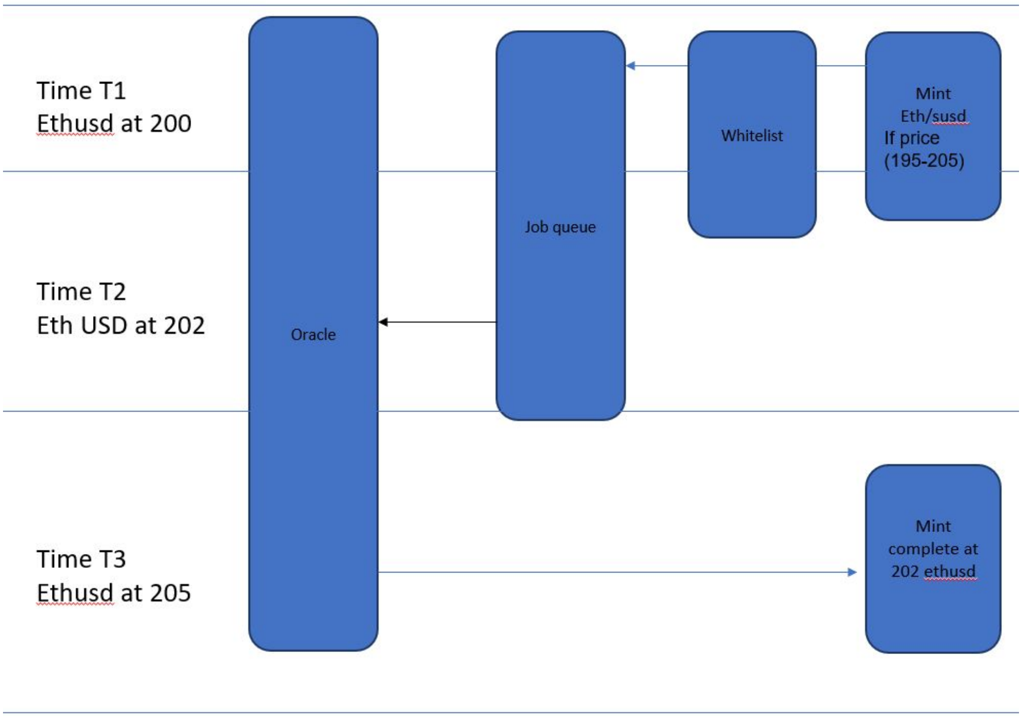
Governance platform



1. Governance platform is necessary because malicious jobs can be created to hurt validators. E.g. creating a job for a url which gives random values. Since there would be no consensus, lot of validators will be penalized.
2. To use oracle, the URL must be whitelisted through governance process.

3. A vote will be taken using schell tokens to whitelist or blacklist a URL.
4. SCH holders should whitelist a URL if it is: reputed, consistent, can handle high load, if it is exchange, it is not having withdraw/deposit/trading/regulatory issues, response is not too big, is free, not hidden in darknet, not geo-restricted, etc.
5. Another feature of this platform will be to use collections. E.g. ETHUSD collection can contain 5 exchange URLs. If any of those exchanges are compromised or become defunct, governance process can remove the URLs from the collection and add new ones.
6. Decentralized assets can be minted using this collective feed and they don't have to rely on a specific feed.
7. In beginning this process will be centralized and decentralized over time.

Application example: Synthetic assets platform



We will explore how one can develop a Synthetic assets platform utilizing Razor network as an oracle service provider. A synthetic assets platform (Also known as a Delta one platform) provide a way to speculate on the value of any asset without actually trading that asset. A synthetic assets platform can be built using Razor network in following way:

1. The application developer can propose various data feeds and collections, as required by the application, to the governance layer.
2. The governance layer approves the data feeds and collections as long as they are valid and follow certain guidelines.
3. Users can provide collateral to mint new assets according to data-feed values. collateral can be SCH, ETH, etc.
4. Users can burn assets anytime according to price-feed values to get back their collateral.

5. As an example of an asset that can be created using the application, consider sUSD, a stablecoin pegged to the value of USD. Ether can be used as a collateral and ETH/USD price-feed can be served through Razor network as a reference for the minting/burning process.
6. When assets are requested to be minted/burned the next future available price-point will be taken as reference.
    a. E.g. if Alice requests to mint sUSD at 10am, the last traded price at the beginning of the next epoch will be used as reference.
7. When a position is under collateralized, anyone can liquidate a position by creating an update job for the oracle.
8. To long, buy a synthetic asset off the market. To short, mint it and sell it on the market.