Abstract

The aim of this post is to sketch a possible Anoma proof-of-stake system, which for this post I will call “heterogeneous proof-of-stake”, although this name should be considered provisional.

This design is a continuation of previous thoughtshere, and has also been inspired by discussions with Barnabe Monnot from the Ethereum Foundation.

To start, I think it is necessary to describe the design context and goals.

1.1: What should proof-of-stake provide?

Proof-of-stake was originally understood as an alternative to proof-of-work to be used with Nakamoto consensus. It also enabled BFT consensus algorithms to be operated in a more open manner, particularly influenced by the launch of Cosmos. At this time, proof-of-stake was primarily considered a voting power allocation mechanism for consensus

, and was concerned only with the service of consensus (often referred to as “validation”, although this is a bit of a misnomer).

I understand here the goals of proof-of-stake in a broader sense - in particular, two, which are typically coupled but fundamentally quite distinct:

- Credible long-term alignment

with an asset (and the associated network), which can be achieved by locking the asset for some duration, with a longer duration providing longer-term alignment. This mechanism considered alone is not dissimilar to a “bond” issued by corporations or governments - it is, in effect, debt sold by the protocol (although it also does not feature many specific legal features present in that market, and we should probably find a different term).

- Credible service commitments

which can be entered into by various parties to provide various services such as attestation (validators), storage provisioning, compute provisioning, bandwidth provisioning, etc. These commitments typically require: * The ability to punish

safety violations (e.g. double-signing), which often requires association of the commitment with a “safety deposit” that cannot be withdrawn at will. This is why these commitments (particularly to attestation) are often associated with the locked bonds of native tokens.

- The ability to reward

correct behaviour (e.g. signing votes), which often requires the ability to mint native tokens and distribute them to correctly-behaving parties.

- The ability to punish

safety violations (e.g. double-signing), which often requires association of the commitment with a “safety deposit” that cannot be withdrawn at will. This is why these commitments (particularly to attestation) are often associated with the locked bonds of native tokens.

- The ability to reward

correct behaviour (e.g. signing votes), which often requires the ability to mint native tokens and distribute them to correctly-behaving parties.

1.2: What do users want?

- Network operator diversity (decentralization, preference entropy).
- The ability to prove long-term alignment and to make service commitments.
- Returns for committing to long-term alignment.
- As much liquidity as is compatible with the protocol’s security requirements.

2.1: Abstract Proposal

Abstractly, the proposed proof-of-stake mechanism would consist of:

- A bond mechanism

, with which the protocol can buy and sell (from and to users), and users can buy and sell (from and to each other and the protocol), transferable bond objects, each with a duration, interest rate, and other such parameters. Details such as variable/fixed interest, auto-rollover, etc. are yet to be determined.

- A target debt profile

, describing the protocol's preferred debt distribution across time and bond types, which is used to set prices at which the protocol will buy and sell (AMM-style), and to set interest rates for variable-interest bonds.

- A general language for service commitments

, each of which consists of: * A safety property

, typically what the operator will not

do (e.g. double-sign), logic to detect and attribute faults when this property is violated, and logic to punish (slash) faulty operators.

- A liveness property

, typically what the operator will

do (e.g. sign), and logic to detect correct behavour.

- A safety property

, typically what the operator will not

do (e.g. double-sign), logic to detect and attribute faults when this property is violated, and logic to punish (slash) faulty operators.

- A liveness property

, typically what the operator will

do (e.g. sign), and logic to detect correct behavour.

- A service bonding mechanism

to associate service commitments

with bonds

. In general, services must be deactivated before bonds can be resold to the protocol. Most service deactivations are immediate, but some (e.g. attestation) will require cooldown periods (in order to allow for faults to be detected).

- A service payout function

(mutable) which calculates what the protocol will pay out for different service commitments, on the basis of the type, the liveness, the operator's identity, and the other service commitments made.

2.2: Concrete Proposal

- Most of the mechanisms and functions can be implemented on the Anoma Resource Machine.
- The service commitment language

should be in the form of predicates over incoming and outgoing messages. This must be legible to the whole system. These commitments can then be registered

with particular controllers.

3: Alternative Options

- These two goals can be completely decoupled by making the choice about service commitment payments collective (via governance) rather than individual. This means that losses are socialized, though - need to think about incentives here. Perhaps a hybrid approach (~ delegator DAOs) is possible.