Objective

Explain Anoma over a cup of Coffee. The explanation should provide the listener a strong foundation to proceed with more inquiry.

Please critique this, tell me why and how it is wrong. Also, if you feel so inspired, I would encourage you to try this exercise as well.

Caveats

I recognize I'm excluding many details. The goal is to explain to the listener without too many Proper nouns. For example, I decided not to include machine groupings but thought about it.

Anoma

Anoma is a generalized architecture designed for intents. Intents are commitments to preferences over a shared state space. Intents let a user specify what they want by invoking a set of constraints. Intents allow users to control the amount of information they reveal - what to whom.

In the Anoma architecture, a user outsources the task of computation to a solver

. A solver is a specialized agent that figures out how to satisfy a users' intent. Solvers match intents, which are represented in the form of partial transactions. Partial transactions are combined to form a balanced transaction, which can contain many

intents.

Solvers send fully balanced transactions to Typhon, Anoma's ordering protocol, which provides the mempool, consensus, and execution engines. Taiga, Anoma's shielded state transition protocol, checks the validity of the transactions, ensuring all application rules and predicates are satisfied.

Anoma realizes intents with a resource model. Resources can be represented as a UTXOs. Intents allow for arbitrary constraints which are represented as resource logic or as side information sent to the solver. This architecture requires an intent gossip network (p2p layer), a marketplace of solvers offering solutions, and a place to send matched intents.

Anoma scales with fractal instances. Nodes running the Anoma network can spin up an instance on-demand, or a particular set of nodes could operate a persistent instance within the Anoma network. Typhon allows for instances to interoperate and create atomic bundles that settle to multiple instances given enough overlap of Anoma nodes (validators). This feature is known as Chimera Chains.

Anoma allows for various runtime configurations that are up to the user and application developers to determine. For example, there could be a fairness or welfare criteria that needs to be achieved. In the former, a solver could send an encrypted batch of intents to Typhon which comes to consensus on the batch then decrypts the batch after x blocks for solvers to then match. This type of configuration is often referenced when describing Ferveo, a threshold decryption DKG protocol.

Anoma can be instantiated as a blockchain, albeit at any layer. It can settle to any other blockchain or act as a settlement layer. Anoma's components can be used in isolation or more beneficially as one homogenous stack. It is up to users to choose the constraints they want satisfied and the security domain they wish to settle their intents to. Anoma applications are not married to any one particular instance and can be portable across any of Anoma's fractal instances. Anoma application developers write programs in Juvix, a high-level functional language.