First of all, I do entirely understand that everything is processed 1 by 1 on layer 1. I'm thinking about a layer 2 network architecture though where unconnected smart contracts can run in parallel, but some contracts that are connected would need to concentrate resources in some part of the network at time of high volume.

With money involved, a valid concern is who get priority on their transaction? In my imagination, I believe this can be settled by this layer 2 network as it is all under the governance of assets on layer 1.

I'll need to formulate my thoughts a bit more clear. I feel I can only do that by brute-forcing myself through some discussion.

Let's assume there are computers in the real world that go untouched because of watchful eyes from layer 1. In other words, a DAO from layer 1 ensures several computers remain untouched. Let's assume the DAO is fair. These computers don't need to mine blocks with proof-of-work or proof-of-stake.

Between these computers needs to exist a language for developers to create smart contracts. I believe we'll eventually be dealing with some sort of "Chinese Postman Problem". Trying to optimize routes between machines that involve smart contracts that refer to each other.

What critical features to a programming language need to exist in a network like this? Where a smart contract can still refer to any other smart contract, but it must do so asynchronously and of course gas prices are calculated based on the current concentration of demand.

And when I think about tokens, something interesting pops into mind. I would think a very simplistic token "contract" (if it would even remain that with what I'm about to describe) could be sharded across the network. There shouldn't need to be a bottle neck around keeping track of simple "points" or some type of simple "token". Balances could be stored in different regions of the network and there should never have to be many different other smart contract trying to funnel into a single token contract to get accurate balances on different users.

Perhaps my wording is a bit cloudy. I appreciate any contribution to this idea. I'm trying to clear things up for myself.