TL;DR

it's a decentralized network where ephemeral blockchains (i.e. app-chains mostly) can be created and destroyed. This can be a L2 or L3 solution to the Ethereum ecosystem. Basically it's a generalized L2 solution for all EVM-compatible blockchains.

Problems

If smart-contract blockchains are like decentralized world computers, it's bizarre that a computer cannot delete old applications and data. Append-only ledgers can be very useful and we already saw that. But there can also be ledgers not append-only but still useful and even more propriate for certain use cases. Think about the experimental, toy and dead contracts that were deployed in the early days to the Ethereum blockchain. Do people still interact with those contracts and need access to the storage? I doubt that. Most of them are probably dead. Ethereum has a long-term roadmap so that clients do not need to keep old data. But there is a long way to go and we need a way to slow down the space-growing speed. The solution is to give people another option if they don't need on-chain persistent data.

How it works:

- it's a permissionless network with decentralized nodes. The nodes are idle when there is no demand. The only thing they do at idle times is to ping each other to keep the network alive

- anyone can make a requst to the network to create an ephemeral blockchain with a configuration set (e.g. {name, id, max_ttl}) and a deposit locked somewhere to cover the miners' base maintainence fees during the period its alive. The deposit should be big enough and calculated using the max_ttl.

- miners will also collect normal gas fees paid by people who use the blockchain during the period. we might also add a feature to allow chain proposer to claim (0%-100%) of the gas fees to themselves. Note that even at 0%, miners are still incentived by the initial deposit of the chain proposer.

- one the chain was set up, anyone can start to bridge ETHs to the chain. The bridged ETH and the initial deposit by the chain proposer is locked in a smart contract on Ethereum or other L2 chains.

- the ephemeral blockchain's native token will be the bridged ETH (i.e. bETH). So gas fees will be paid in bETH.

- when the max_ttl is reached or the chain proposer decides to end the chain early, the chain is scheduled to be destroyed by the network (in amout a week. determined by the protocol. initial deposit should also cover this period. this is for full transparency so people can audit what has happened on the chain is fair and legit) and a snapshot will be taken. The snapshot is critical to determine the amount of ETH people can withdraw from the bridge contract. There should be a mechanism to automatically distribute the fund instead of asking people to do it themselves. For example, it can allow anyone to do the distribution and earn some fees as a return.

Peripheral features

- note that we might enable a feature to forfeit the txs on the ephemeral blockchain if there is a hack. For example, we might add a function in the bridge contract to allow the chain proposer to return all the locked fund back to everyone at the same amount they initially locked.

- there can also be a feature to "top-up" the chain as long as you still need it. Another feature that combined will be powerful is to "cap" the chain at a certain size. For example, you can set a rule to create a chain which only keep the latest 3 months' records.

- the chain should be customizable. for example, the proposer can specify that only himself can deploy contracts on the chain.

Use cases

- This can solve the gas war problem we saw at the time of Otherside's public mint event. In the indicent, lots of people got their txs reverted and lost a lot in gas fees. If using this L2 solution, the Otherside company will first initiate an ephemeral blockchain and deploy a smart contract for the NFT public sell. Then they announce the plan publicly and the chain address and the bridge address. Then people start to brige fund to the chain. Then when the public begins, those people compete in the ephemeral chain to try to mint the NFTs. The competition should be much less costly then done in Ethereum directly. After the sell, when the chain is destroyed, users' unused fund will be claimable via the bridge contract. The NFTs should also be bridged to the Ethereum chain. This might need to be done in a centralized way by the Otherside company. But maybe there can be a better way of doing it. Nevertheless, at least this

design achieved full transparency.

- another potential use case is the Web3 games and this can be very exciting. Think about that you play some PVP cards games such as Hearthstone. If it's a web3 game, it might be decentralized and thus requires transparent game execution on-chain, especially when GameFi element is involved in the game. But any existing solutions, even the high-speed chains such as Solana, won't be a good fit because it's still too expensive. Basically, any generalized smart-contract blockchains won't be a good fit for such use cases and it's also a waste of public resources. But let's take a step back and rethink about what the game really needs. The PVP game does not really need the data for each match to be stored forever on-chain. A reasonable amount of time for auditing purpose is more than enough. So it will be a perfect fit to use the "cap" and "top-up" features of the network to create a chain that will only keep the most recent records.

- in general, any use cases that requires a transparent and decentralized smart-contract execution environment but does not need persistent data will find this network useful.

Questions

- does this idea make sense? are there any better solutions?

- If I want to figure out the percentage of contract code and storage data on a Ethereum node that are untouched for more than X years (i.e. dead contracts), how can I do that?

- what's the timeline for Ethereum to implement similar features? are there any teams working on it?