A couple of weeks ago I decide to write down every proper scaling solution I was aware and see how the things might play out in the future. I'm also considering building something and I couldn't find a useful write up on the state of rollups and the options available to new startups.

Assuming that I not alone I thought to put those findings down myself and sharing them in case other founders or maybe people in the space are interested.

Hope you find it useful.

Introduction

Crypto networks build during the last decade have nonetwork effects. Blockchains are eitherexpensive, insecure or not decentralized and there can't be a single blockchain that can both run on consumer laptops and have cheap fees.

But if we can't have a single chain that's both cheap and decentralized, can we have multiple, inter-operating chains? That's what's proposed in the Ethereum roadmap almost 3 years ago.

When the rollup-centric roadmap was proposed, the exact details were unclear and having proper working systems was far away. It's been while now and I think we now know these systems might look like. This essay is about presenting and discussing these L2s, L3s and fractal scaling systems with their security characteristics, interoperability properties and finalization times.

Multiple L1s (Naive scaling)

Naive scaling means running multiple distinct blockchains in parallel. In such a consortium, every blockchain is validated by it's own validator set and interactions are facilitated with transaction proofs sent from one blockchain to the other (and back).

Even if blockchains' security budgets grows proportionally with their adoption, facilitating interactions between two or more chains is both ineffective and sometimes insecure. One the one hand, it's insecure since the receiving chain have to trust that the other chain's validators won't fork their chain (either for an upgrade or by acting maliciously). If they do, they may revert the transaction that could cause lost of user funds. One the other hand, there are high costs to validate these cross chain transactions. As these transaction require multiple cryptographic operations they need a lot of gas and they usually cost an order of magnitude (or two) more that a typical transaction.

Moreover, having multiple chains means that validators will have to be divided among the chains. The average chain will then have far fewer validators. Chains with less nodes have a higher chance that they offline which is bad as of itself and even worse in a multi-chain world that chains depend on each other.

Besides the drawbacks, naive scaling is simple to setup. L1s do not need cross chain operations that are expensive and complex to operate. Chains are simple, less fragile and have clear finality conditions.

Properties:

- Validity:

Each chain validated by its own nodes

- Data availability:

Data-availability on the L1

- Interoperability:

Fast but trusted

- Confirmation:

A few seconds (block time)

- Finality:

A few seconds to a few hours (depends on implementation)

More info: [Vitalik's on multi-chain

and cross-chain

](https://reddit.com/r/ethereum/comments/rwojtk/ama_we_are_the_efs_research_team_pt_7_07_january/hrngyk8/)

Multiple L2s with a single bridge (OP Superchain)

Optimism's Superchain is a good example of multiple L2s combined in a single unified system. A Superchain is a collective of layer 2 blockchains that share 1) a chain factory that creates new L2 chains and 2) a common L1 ↔ L2 bridge for deposit and withdraws.

The chain factory makes the deployment of new OP chains cheaper, more simple and configurable. It allows the new chains to pick choose their configuration (ie. chain id, gas limit, sequencer infrastructure, data-availability settings etc) easily and used a tested smart contract to deploy their own version of the L2. At the same time if the chain is deployed through the chain factory it easy for the Superchain to trust that the new chain code is safe. The shared common bridge allows chains on Superchain to utilize the security audits and fail-safes of the Optimism bridge by default as well.

Note:

Optimism made the first working L2 in 2019 and worked with with Geohot to get on-chain fault proofs working.

Cross chain transactions would work with L2s that share the same sequencer(s). More specifically, L2 transactions will not touch the L1 but go directly from one L2 to another so that transactions are cheap and fast but also atomic.

The Optimism collective is also building a framework to support multi-chain applications. The framework proposed cross-chain contract state management standards and a single RPC endpoint (called the Superchain RPC) that users can use to interact with all OP chains seamlessly. Chains that follow the standard would have the smart contracts that can move from one OP chain to another and have the same address.

Note:

As stated in the Superchain Explainer, once the ZK tech is more mature they will introduce ZK proofs for withdrawals that will be both fast and secure.

Properties:

- Validity:

At least 1 honest node (honest minority)

- Data availability:

On-chain guarantees posted on L1

- Interoperability:

Trust-minimized but slow (fast with shared sequencer)

- Confirmation:

A few seconds (sovereign)

- Finality:

A few days (on L1)

More info: Optimism Superchain Explainer

Multiple L2s, L3s etc with fractal scaling (Slush SDK)

In a fractal design there are multiple blockchains with childs and parents in a tree like formation. Child chains are anchored to parent chains up until reaching the L1 that is the only chain that doesn't have a parent. Slush is proposing a design that relies on fractal scaling and validiums (ie. chains with lower transaction costs that don't post their data on the L1) but have alternative ways to store their data.

A rollup provides many advantages to the application layer like having cheaper transaction fees, supporting different VMs and having private transactions. Most benefits of rollups apply to both horizontal and fractal rollups (ie. L2 rollups) fractal rollups but since horizontal rollups are much easier to setup and coordinate there isn't a lot of interest in fractal rollup so far. That's not the case with fractal validiums. Fractal validium designs could utilize their fractal topology to better coordinate the blockchain data storage. That's what's the Slush design is about.

In Slush draft design, each child chain is responsible for replicating the data availability of all its parent chain. Chains that are higher up and have many chains below them would naturally have higher data replication (and hence higher security) but higher fees. In contrast, chains lower down with less children would have less data replication (and less security) but lower transaction fees.

The design also proposes a mitigation to the L2s liveness problems. In short, if chains higher up start to go offline or fail to provide proofs that their data is available, chains lower down can replace their parents so other chains in the fractal leaf can

continue to operate smoothly. That possible since children have their parent's data and they could be incentived to take that role higher up the hierarchy to better benefit from the parent chain demand.

Properties:

- Validity:

Wallets (light clients) validating directly (ZKPs)

- Data availability:

Off-chain guarantees by child chains

- Interoperability:

Fast and trust-minimized

- Confirmation:

A few seconds (sovereign)

- Finality:

A few hours (consensus)

More info: Slush, a proposal for Fractal scaling

Multiple L2s with horizontal scaling (Sovereign Labs)

A sovereign rollup design utilizes off-chain computations to get to consensus. These off-chain computations could happen on the user wallet in the background and right before sending a transaction. But to calculate the state of the blockchain is very computationally demanding and requires hours of compute and a lot of disk space which is couldn't be done within a browser extension or a mobile app. That means that sovereign works can work if there's a way to compute cheaply the state of the blockchain.

One such solution are zero knowledge proofs that are a computationally cheap way to validate that a blockchain has progressed correctly. One node can generate a proof and the other nodes can then verify the blockchain cheaply without executing the transactions again

. That's very useful in general as processing transactions is one of the two core function of a blockchain (the other is storing the blockchain data) but it's especially neat for cross-chain transactions.

Traditional zk rollups, usually called smart contract rollups, have these zero knowledge proofs posted on the L1 for L2 nodes to use and get to consensus from. But these L1 proofs are still computationally

expensive to execute on-chain and L2s only post them every 6-12 hours to save on transaction costs. That means that smart contract rollups finalize every few hours and as the consequence cross-chain transactions need a similar time to go from one L2 chain to another.

Sovereign rollups achieve a similar result through a different method. They only post the proof data on-chain, which are much cheaper than executing the whole proof on-chain and share the proof directly to user wallets that have the responsibility to verify the proof. To achieve this safely they have to store another proof on-chain that demonstrates which transactions are included in the L1. This second proof proves L2 transactions are included in the L1 and combining the two proofs shows that L2 transactions are both valid and posted on the L1.

In summary, a sovereign zk rollup needs two proofs to operate, one for the state of the underlying L1 and one for the state of the L2. Light clients can validate these two proofs to know if a transaction went through in the the L2. Since these proofs are cheap to post to the L2 they can post them frequently which enables instant cross-chain transactions.

Properties:

- Validity:

Wallets (light clients) validating directly (ZKPs)

- Data availability:

On-chain on L1 or off-chain

- Interoperability:

Fast and trust-minimized

- Confirmation:

A few seconds (sovereign)

- Finality:

A few hours (consensus)

More info: How the Sovereign SDK works

Multiple L3s on a single L2 (Arbitrum Orbits)

Arbitrum Orbit chains are L3 validiums that settle on the Arbitrum L2. These L3 chains are very similar to an L2 chain settling on the L1 but use Abtritrum's off-chain data availability layer instead. Since they don't post transaction data on the more expensive L1, they have lower transaction fees than a L2 rollup.

Details on the exact interoperability features are not released yet. One way they might work is that cross chain transactions will have to go through a shared sequencer (splitting L3s into interoperable clusters). Alternatively they will have to go through the L2 to go from one L3 to another, similarly to how OP chains use the L1 for cross-chain transactions.

Properties:

- Validity:

At least 1 honest node (honest minority) and Arbitrum off-chain committee

- Data availability:

Off-chain data availability committee (Arbitrum Nova)

- Interoperability:

Trust-minimized but slow (?)

- Confirmation:

A few seconds (sequencer)

- Finality:

A few days (on L1)

More info: A gentle introduction: Orbit chains

Conclusion

After years of research, we are now entering the implementation phase. The first implementations of the rollup-centric roadmap are now in beta or in production and we should see working systems with multiple L2s, multiple L3s and various other combinations in the next few months.

When these cross chain systems become more mature there would be a few more challenges to solve and that's mainly with user experience. We will need new wallets that would accommodate cross-chain transactions without making sacrifices on simplicity and usability and I can't wait to see these cross-chain systems working in the wild.