Rollups

ETHEREUM 2.0

WITH ROLLUPS

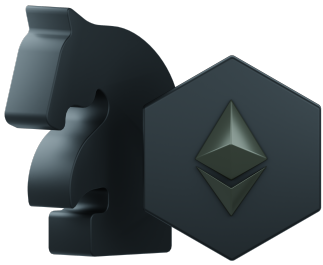
RAFAEL FUENTES RANGEL

ROLLUPS AS AN ALTERNATIVE TO MAKE BETTER THE SPEED AND LOW FEES IN ETHEREUM BLOCKCHAIN

INTRODUCTION

All of us know that Ethereum is an obstacle to make speed and low transactions, at the moment to buy an NFT, or make a swap in Uniswap, and is something annoying for the community pay until 300 USD dollars per transaction.

There are a lot alternatives to Ethereum, that accepts the Ethereum Virtual Machine, as Avalanche, xDAI, RSK, inclusive protocols as Polygon. In the other way, there are blockchain layer1, that wats replace Ethereum, as Polkadot, Cardano, Solana, Neo, EOS, etc. But we know that is not easy the communication between the smart contracts in all blockchains, that is called as multi-chain, so, there are a disruptive alternative for this, that is in Layer2, o also met as Ethereum 2.0, and are the Rollups.



ROLLUPS

Rollups is a collective term, made to help to scale the transaction into Ethereum. A simple way to explain a rollup is that execute a transaction out of the main chain from Ethereum, but is secure inside it.

There are 3 properties of this:

1. Execute transactions out of the mainnet
2. All transactions are in layer1
3. The smart contract execute with the layer1 data in the layer2.

Over all, exist 2 rollups types with security models:

Optimism Rollup - It knows that the transactions as correct for defect and just execute the calculate across a “Proof of Fraud” that in resume is package transactions in lots and send to Ethereum in one transaction. It suppose are valides, but can be rejected if suspected of fraud. Then PoF will execute the transaction to let it see if it made a fraud. This method increase the amount of transactions possibles while supports the security.

Zero-knowledge Rollup - It runs computation off-chain and make a “Proof of validity” to the chain. A “PoV” is a way to increase the speed of the transactions, that is the same into method than Optimism Rollup, but the calculate makes out of the chain, and then it will send to the mainnet (Ethereum) with PoV. This method increase the amount of transactions while is secured in the Layer1.

OPTIMISM TECHNICAL STUFF

The smart contracts can separate in some key-components.

Chain: Smart contracts in layer1, that contain the transactions order and the commitments with the state root in layer2.

Verification: Smart contracts in layer1 that make the process to get the result of one transaction.

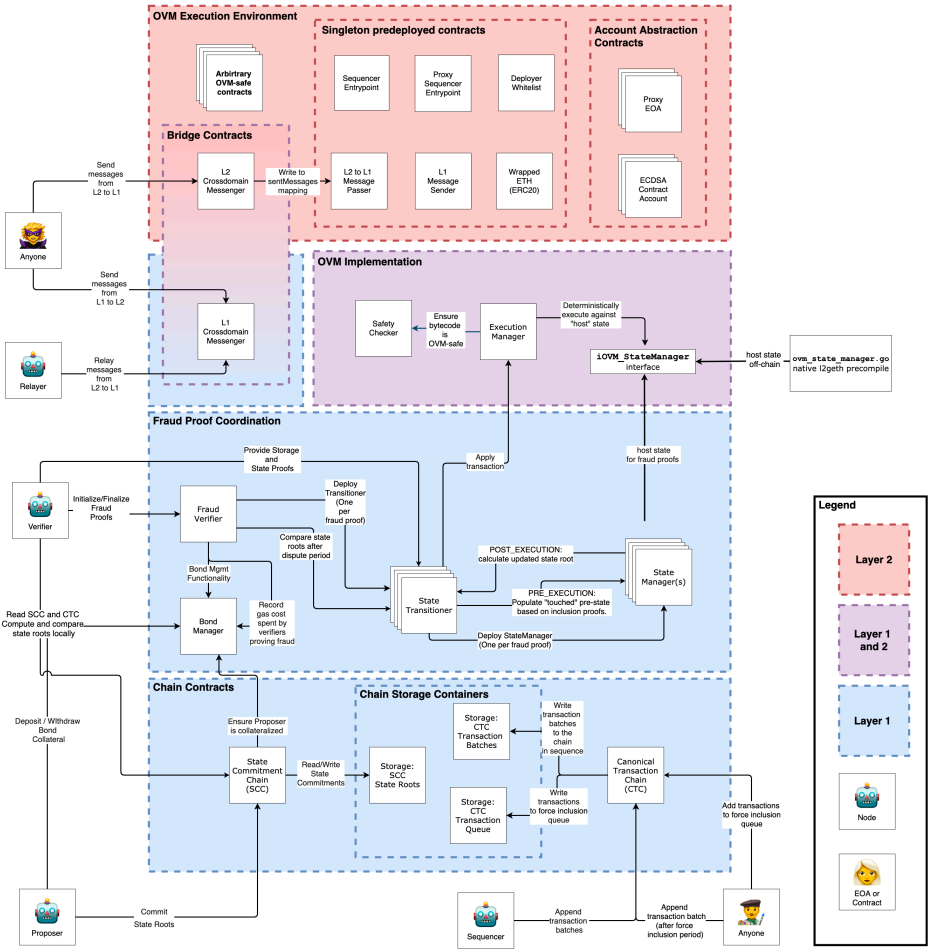
Bridge: Smart contracts that make easy the message step between layer1 and layer2.

Implementation: A smart contracts set that implement and are available in genesis system. This contracts are similars to the pre-compiles of Ethereum, written in solidity and can find in address that starts in 0x42.

The Optimism blocks production is manage for the “sequencer”. It gives transaction confirmed instantly and state updates, build and execution of block in L2, and sending transactions of user to L1.

Is just there are a disadvantage, that we need to wait 1 week to complete the transaction, for the main reason that it need to make the PoF, if the transaction is rejected in the range of 7 days, it means that Optimism cannot process the transaction. But if complete the 7 days. It will process it.

Optimistic Ethereum High Level Architecture



ZK ROLLUP TECHNICAL STUFF

In resume ZK Rollup, make the transactions out of the chain, and make a cryptography test. This test can came in SNARK way (succinct non-interactive argument of knowledge) or STARKs (scalable transparent argument of knowledge). Are better knowledge as PoV and it publish in L1.

The smart contract of ZK-Rollup keeps the state of the transactions in L2. And this just needs the PoV and not the transaction data. And with ZK-Rollup, validate a block is quickly and lower because include a few data. Here there are a greater difference in this data, about the size and time in this 2 Zero-Knowledge Rollup.

	Proof Size	Prover Time	Verification Time
SNARKs (has trusted setup)	288 bytes	2.3s	10ms
STARKs	45KB-200KB	1.6s	16ms
Bulletproofs	~1.3KB	30s	1100ms

There are some disadvantages with this Rollup, the first one is that some of this have not EVM support, the second one is that the PoV are intense of calculate, because it tries to figure it out, or better say, calculate the exact data that the transaction needs, and then customize it, for example, reduce bytes32 to bytes4.

We can difference SNARKs and STARKs viewing this simple table.

	SNARKs	STARKs	Bulletproofs
Algoritmic complexity: prover	O(N * log(N))	O(N * poly-log(N))	O(N * log(N))
Algoritmic complexity: verifier	~O(1)	O(poly-log(N))	O(N)
Communication complexity (proof size)	~O(1)	O(poly-log(N))	O(log(N))
- size estimate for 1 TX	Tx: 200 bytes, Key: 50 MB	45 kB	1.5 kb
- size estimate for 10.000 TX	Tx: 200 bytes, Key: 500 GB	135 kb	2.5 kb
Ethereum/EVM verification gas cost	~600k (Groth16)	~2.5M (estimate, no impl.)	N/A
Trusted setup required?	YES 😒	NO 😄	NO 😄
Post-quantum secure	NO 😒	YES 😄	NO 😒
Crypto assumptions	Strong 😒	Collision resistant hashes 😄	Discrete log 😒

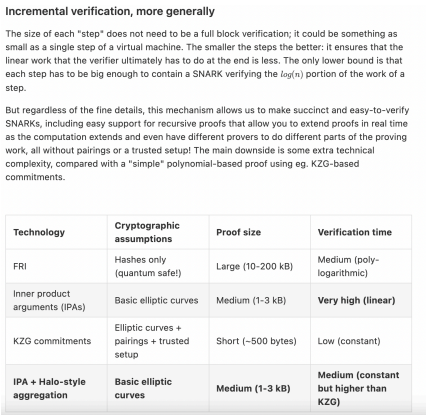
Something interesting about the 2 Rollups, is that the protocol and framework Polygon (Matic) uses both, takes the better of each one, and apply with it. With another tools as well.

CONCLUSION
WAR FOR THE LAYER2

As we had seen, there are solutions there, solutions here, but is an advance on blockchain technology, specifically in EVM compatible solutions, fighting to get the main solution for the community to let their use perfectly and comfortably the price and the speed with the operations we want execute in the EVM.

As a perfect value for both projects. Definitely is not mature to choose one, actually, the ecosystem see more the next hype, the multi-chain functions between blockchains that accepts EVM, as a L1 projects as the L2 projects. But, we are the public that sees a battles between all of them, more for the technology than for the price, as a point of view from a blockchain developer.

To let finish you read this simple article. Let you see this wonderful article from Vitalik Buterin talking about the **increase verify and SNARK without pairing**. Click on!



Incremental verification, more generally

The size of each "step" does not need to be a full block verification; it could be something as small as a single step of a virtual machine. The smaller the steps the better: it ensures that the linear work that the verifier ultimately has to do at the end is less. The only lower bound is that each step has to be big enough to contain a SNARK verifying the $log(n)$ portion of the work of a step.

But regardless of the fine details, this mechanism allows us to make succinct and easy-to-verify SNARKs, including easy support for recursive proofs that allow you to extend proofs in real time as the computation extends and even have different provers to do different parts of the proving work, all without pairings or a trusted setup! The main downside is some extra technical complexity, compared with a "simple" polynomial-based proof using eg. KZG-based commitments.

Technology	Cryptographic assumptions	Proof size	Verification time
FRI	Hashes only (quantum safe!)	Large (10-200 kB)	Medium (poly-logarithmic)
Inner product arguments (IPAs)	Basic elliptic curves	Medium (1-3 kB)	**Very high (linear)**
KZG commitments	Elliptic curves + pairings + trusted setup	Short (~500 bytes)	Low (constant)
IPA + Halo-style aggregation	**Basic elliptic curves**	**Medium (1-3 kB)**	**Medium (constant but higher than KZG)**