Tr

Haym @SalomonCrypto

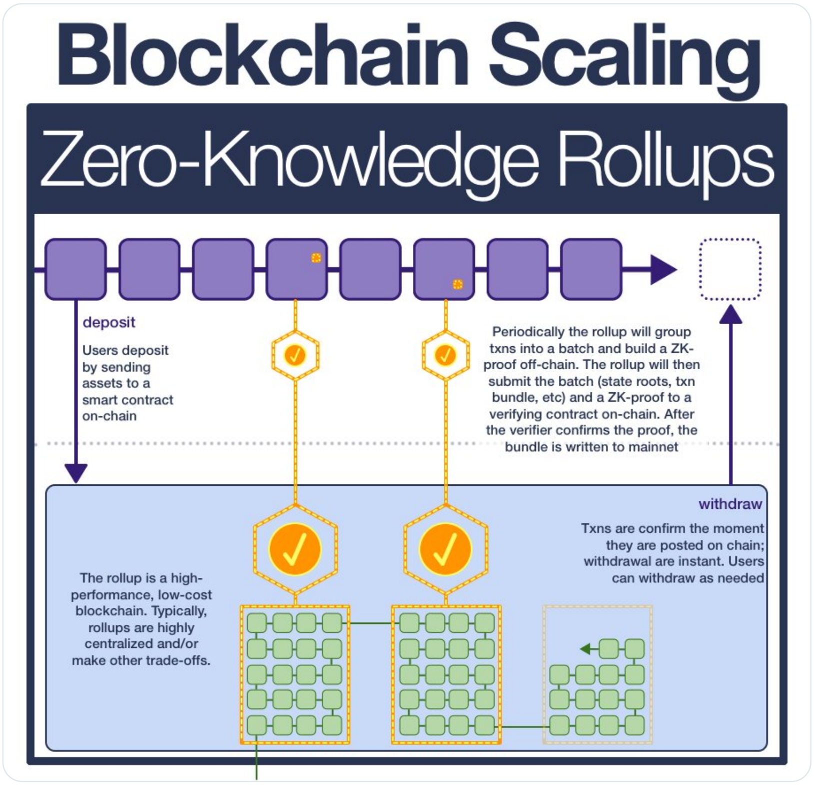
Sep 12 • 24 tweets • SalomonCrypto/status/1569334975614582784

(1/23) Blockchain Scaling: Zero-Knowledge Rollups

State Channels → Plasma → Optimistic Rollups → ZK-Rollups

This is how @ethereum scales to 100k txn/sec. This is how Ethereum becomes the World Computer.

Your guide to the future of blockchain scaling technology.


Blockchain Scaling
Zero-Knowledge Rollups
deposit
Users deposit by sending assets to a smart contract on-chain
Periodically the rollup will group txns into a batch and build a ZK-proof off-chain. The rollup will then submit the batch (state roots, txn bundle, etc) and a ZK-proof to a verifying contract on-chain. After the verifier confirms the proof, the bundle is written to mainnet
withdraw
Txns are confirm the moment they are posted on chain; withdrawal are instant. Users can withdraw as needed
The rollup is a high-performance, low-cost blockchain. Typically, rollups are highly centralized and/or make other trade-offs.

(2/23) In 2008, we all got a taste when Satoshi Nakamoto released @Bitcoin out into the world. But in 2015 @VitalikButerin served us the whole meal: @ethereum was born.

Unfortunately, the World Computer is SLOW...

...for now.

(3/23) First came State Channels, which allow users to lock funds in a smart contract on @ethereum and transact freely off-chain.

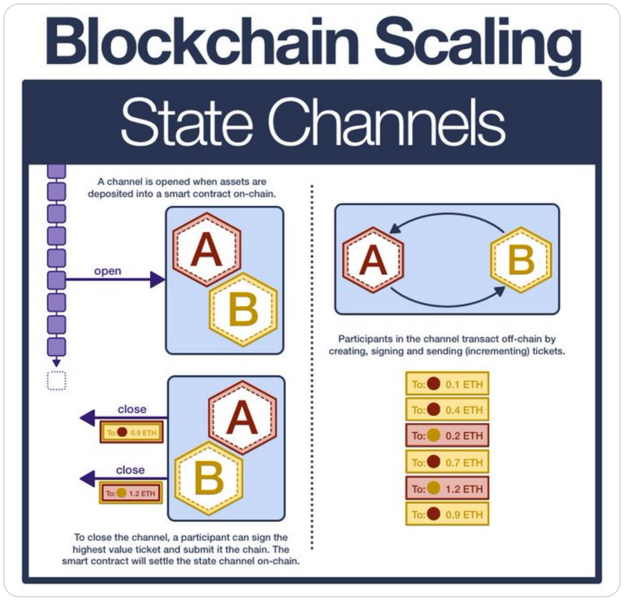
This enabled users to send assets back and forth MUCH cheaper and faster than before while still (ultimately) settling to Ethereum.

Haym
@SalomonCrypto · **Follow**

(1/14) Blockchain Scaling: State Channels

@Bitcoin, **@ethereum** and all (good) blockchain computers share one important quality: they are SLOW. State channels are the first attempt at changing this and bringing blockchain to scale.

Your guide to the original scaling tech.


Blockchain Scaling
State Channels
A channel is opened when assets are deposited into a smart contract on-chain.
open
A
B
close
To: 0.9 ETH
close
To: 1.2 ETH
To close the channel, a participant can sign the highest value ticket and submit it the chain. The smart contract will settle the state channel on-chain.
Participants in the channel transact off-chain by creating, signing and sending (incrementing) tickets.
To: 0.1 ETH
To: 0.4 ETH
To: 0.2 ETH
To: 0.7 ETH
To: 1.2 ETH
To: 0.9 ETH

3:25 PM · Sep 10, 2022

Read the full conversation on Twitter

164 Reply Copy link

Read 9 replies

(4/23) While state channels are powerful they are limited:

- participants must opt-in, cannot send funds to people who are not already in the channel
- requires large amount of capital to be locked
- cannot represent objects without a clear owner (eg @Uniswap)

(5/23) Plasma were created in order to address (some of) these issues.

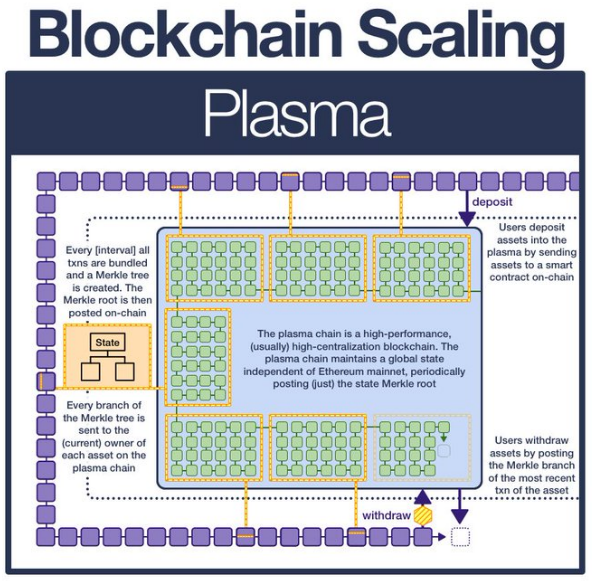
Plasma chains are independent blockchains that anchor themselves to @ethereum mainnet.

Haym
@SalomonCrypto · **Follow**

(1/19) Blockchain Scaling: Plasma

First there were state channels. There there was Plasma, the first persistent-state scaling solution that settled to **@ethereum**.

Your guide to the precursor to modern blockchain scaling.


Blockchain Scaling
Plasma
deposit
Users deposit assets into the plasma by sending assets to a smart contract on-chain
Every [interval] all txns are bundled and a Merkle tree is created. The Merkle root is then posted on-chain
State
The plasma chain is a high-performance, (usually) high-centralization blockchain. The plasma chain maintains a global state independent of Ethereum mainnet, periodically posting (just) the state Merkle root
Every branch of the Merkle tree is sent to the (current) owner of each asset on the plasma chain
Users withdraw assets by posting the Merkle branch of the most recent txn of the asset
withdraw

1:52 AM · Sep 11, 2022

Read the full conversation on Twitter

192 See the latest COVID-19 information on Twitter

Read 17 replies

(6/23) Just like @ethereum, a plasma chain has its own virtual machine state (the status/balance of every user & smart contract)

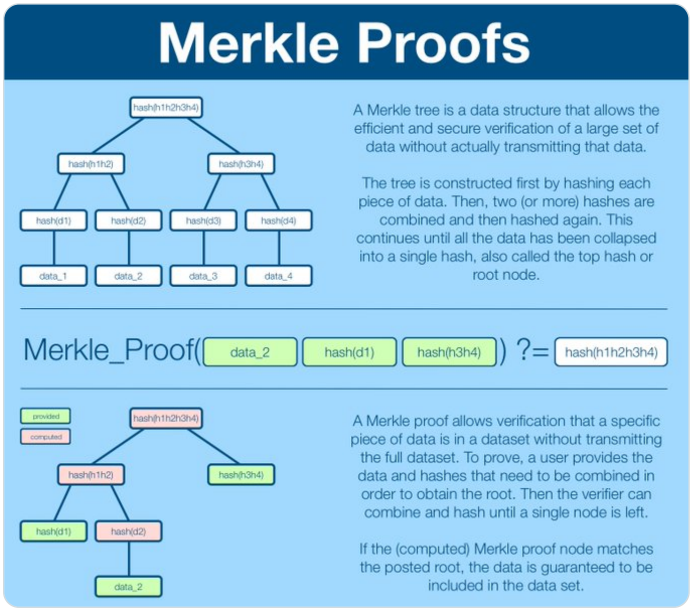
This state can be represented as a Merkle tree - a data structure that allows a huge amount of data to be compressed into a single line (Merkle root)

Haym
@SalomonCrypto · **Follow**

Replying to @SalomonCrypto

(9/13) A Merkle proof is established by providing the specific data and the intermediate hashes which allow a verifier to recreate the Merkle tree.

If the newly computed root node matches root node of the dataset, the verifier can be certain the data is in the dataset.


Merkle Proofs

A Merkle tree is a data structure that allows the efficient and secure verification of a large set of data without actually transmitting that data.

The tree is constructed first by hashing each piece of data. Then, two (or more) hashes are combined and then hashed again. This continues until all the data has been collapsed into a single hash, also called the top hash or root node.

Merkle_Proof(data_2 hash(d1) hash(h3h4)) ?= hash(h1h2h3h4)

A Merkle proof allows verification that a specific piece of data is in a dataset without transmitting the full dataset. To prove, a user provides the data and hashes that need to be combined in order to obtain the root. Then the verifier can combine and hash until a single node is left.

If the (computed) Merkle proof node matches the posted root, the data is guaranteed to be included in the data set.

10:17 PM · Sep 7, 2022

10 Reply Copy link

Read 1 reply

(7/23) Once every [interval] the plasma chain will build the state Merkle tree and post the Merkle root to mainnet. Thus, a (compressed) record of the entire plasma chain exists on mainnet.

In order to withdraw assets, a user submits the Merkle proof for that asset.

(8/23) Plasma chains are a huge improvement; they can send assets to users who have not yet opted in and are much more capital efficient than state channels.

However, plasma chains still have one glaring weakness: data availability.

(9/23) Data availability - the guarantee that all txn data (in the block) will be available for network participants.

Plasma doesn't post the txn data to mainnet, it only posts the state root. If the operator decided to stop sending Merkle proofs, users can't withdraw assets.

(10/23) And so, the solution is to post the txn data along with the state root.

The solution is a Rollup.

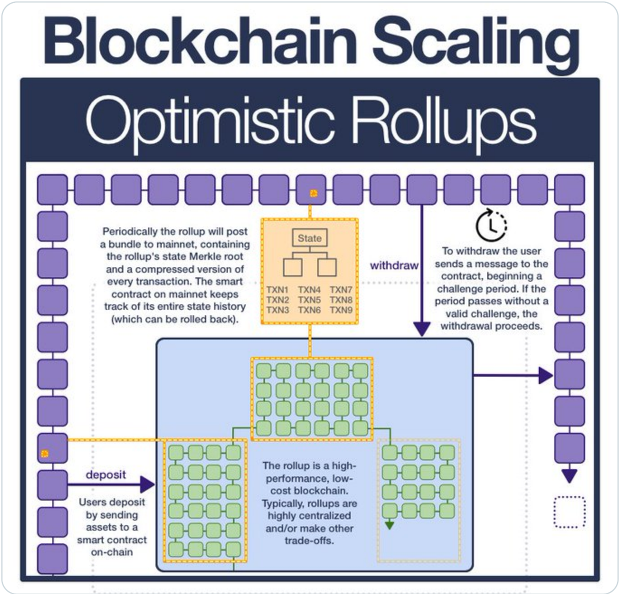
The first group of rollups are called Optimistic Rollups, which assumes there are no malicious actors (while providing a window to dispute any invalid txns).

Haym
@SalomonCrypto · **Follow**

(1/24) Blockchain Scaling: Optimistic Rollups

Wondering why folks are comfortable bridging assets to **@arbitrum**? Curious what's going on under **@optimismFND**'s hood? Need to understand how **@MetisDAO** secures your **$ETH**?

Your guide to the today's premier blockchain scaling solutions.


Blockchain Scaling
Optimistic Rollups
Periodically the rollup will post a bundle to mainnet, containing the rollup's state Merkle root and a compressed version of every transaction. The smart contract on mainnet keeps track of its entire state history (which can be rolled back).
State
TXN1 TXN4 TXN7
TXN2 TXN5 TXN8
TXN3 TXN6 TXN9
withdraw
To withdraw the user sends a message to the contract, beginning a challenge period. If the period passes without a valid challenge, the withdrawal proceeds.
deposit
Users deposit by sending assets to a smart contract on-chain
The rollup is a high-performance, low-cost blockchain. Typically, rollups are highly centralized and/or make other trade-offs.

6:02 PM · Sep 11, 2022

Read the full conversation on Twitter

423 Reply Copy link

Read 11 replies

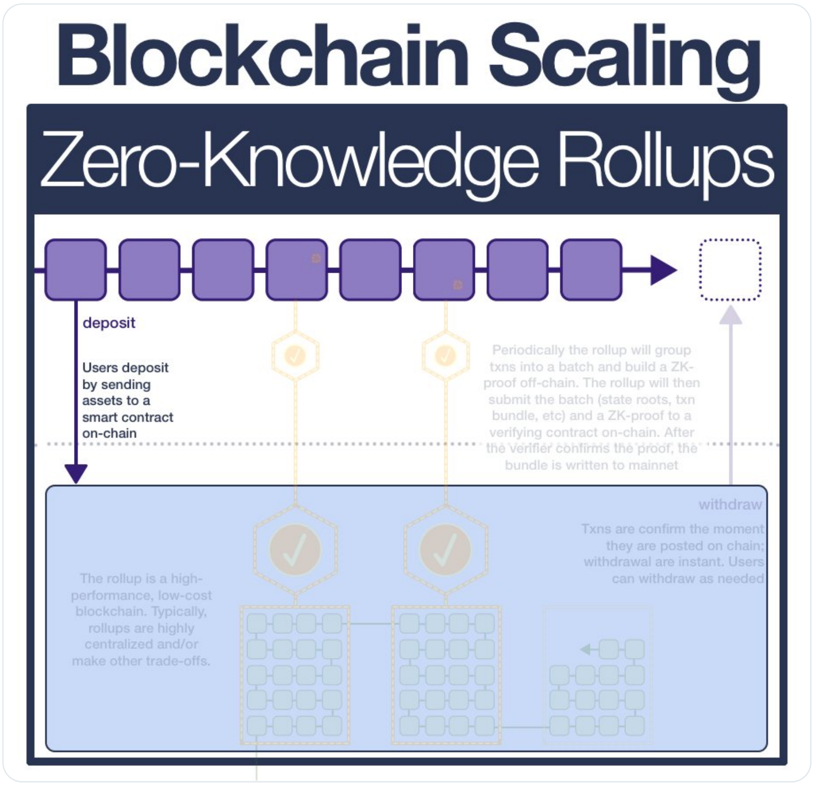
(11/23) Optimistic rollups are very powerful, delivering the performance of a centralized system while still settling to @ethereum

However, their optimistic nature requires a big trade-off: a HUGE challenge window must pass before settlement is finalized

ZK-Rollups solve this

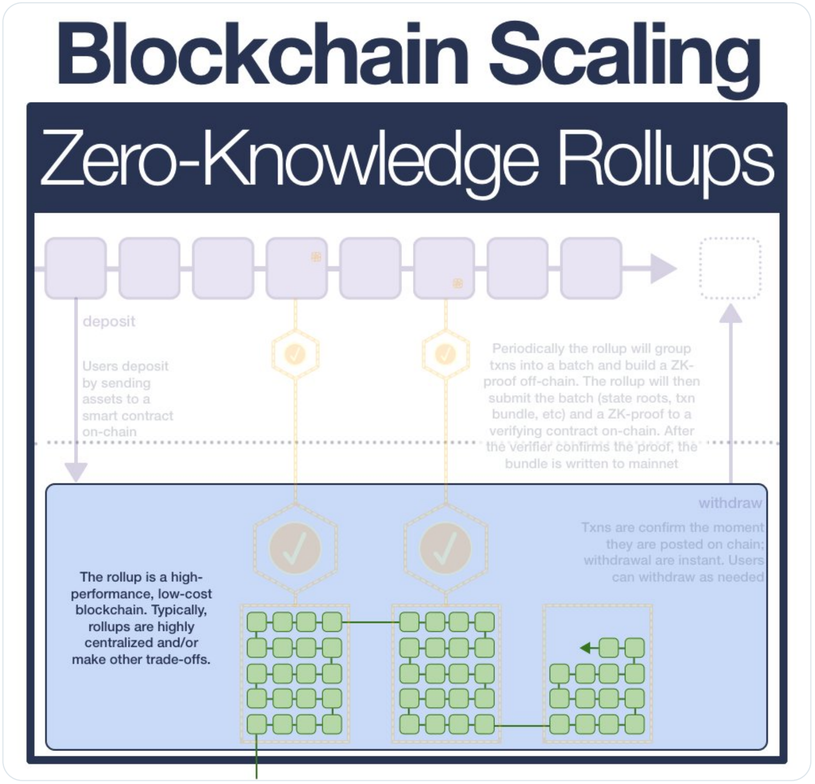
(12/23) Similar to optimistic rollups, first users deposit assets into a smart contract on mainnet. The ZK-rollup operator then mints an equivalent amount of assets on the rollup chain and gives it to the depositor.

On mainnet, the assets remain in escrow.


Blockchain Scaling
Zero-Knowledge Rollups
deposit
Users deposit by sending assets to a smart contract on-chain
Periodically the rollup will group txns into a batch and build a ZK-proof off-chain. The rollup will then submit the batch (state roots, txn bundle, etc) and a ZK-proof to a verifying contract on-chain. After the verifier confirms the proof, the bundle is written to mainnet
withdraw
Txns are confirm the moment they are posted on chain; withdrawal are instant. Users can withdraw as needed
The rollup is a high-performance, low-cost blockchain. Typically, rollups are highly centralized and/or make other trade-offs.

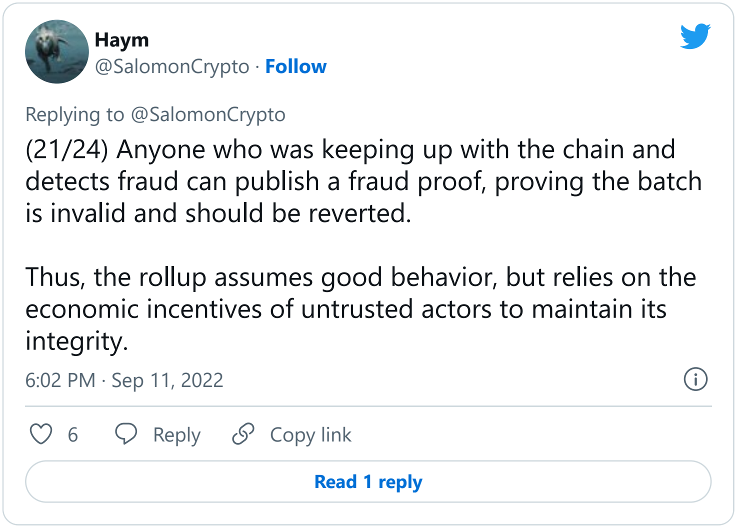
(13/23) Also like optimistic rollups, ZK-rollups derive settlement from @ethereum. This allows the rollup chain to be much more centralized without sacrificing trustlessness.

From a user perspective, execution times and gas costs are SIGNIFICANTLY cheaper than using mainnet.


Blockchain Scaling
Zero-Knowledge Rollups
deposit
Users deposit by sending assets to a smart contract on-chain
Periodically the rollup will group txns into a batch and build a ZK-proof off-chain. The rollup will then submit the batch (state roots, txn bundle, etc) and a ZK-proof to a verifying contract on-chain. After the verifier confirms the proof, the bundle is written to mainnet
withdraw
Txns are confirm the moment they are posted on chain; withdrawal are instant. Users can withdraw as needed
The rollup is a high-performance, low-cost blockchain. Typically, rollups are highly centralized and/or make other trade-offs.

(14/23) From here, both optimistic and zk-rollups bundle groups of transactions together and post a compressed version down to mainnet.

Optimistic rollups naively accept these bundles, assuming that the txns within are valid.



Haym
@SalomonCrypto · **Follow**

Replying to @SalomonCrypto

(21/24) Anyone who was keeping up with the chain and detects fraud can publish a fraud proof, proving the batch is invalid and should be reverted.

Thus, the rollup assumes good behavior, but relies on the economic incentives of untrusted actors to maintain its integrity.

6:02 PM · Sep 11, 2022

6 Reply Copy link

Read 1 reply

(15/23) On the rollup chain itself, you might never notice the challenge period... but it becomes a major issue when crossing between on and off-chain.

Modern optimistic rollups require a 7 day challenge period. Even in Trad-Fi, 7 day settlement would be considered unworkable.

(16/23) ZK-rollups seek to solve the finality question: how can we finalize the batch the MOMENT it is accepted on-chain?

The answer: Zero-Knowledge Proofs.

Haym
@SalomonCrypto · **Follow**

(1/13) Cryptography Basics: Zero-Knowledge Proofs

To keep **@ethereum** decentralized & fair, we keep it slow enough for even the most humble nodes to keep up. But what if they didn't have to keep up with the whole network? What if you could just trust the summaries...

Trustlessly.

12:24 AM · Sep 12, 2022

Read the full conversation on Twitter

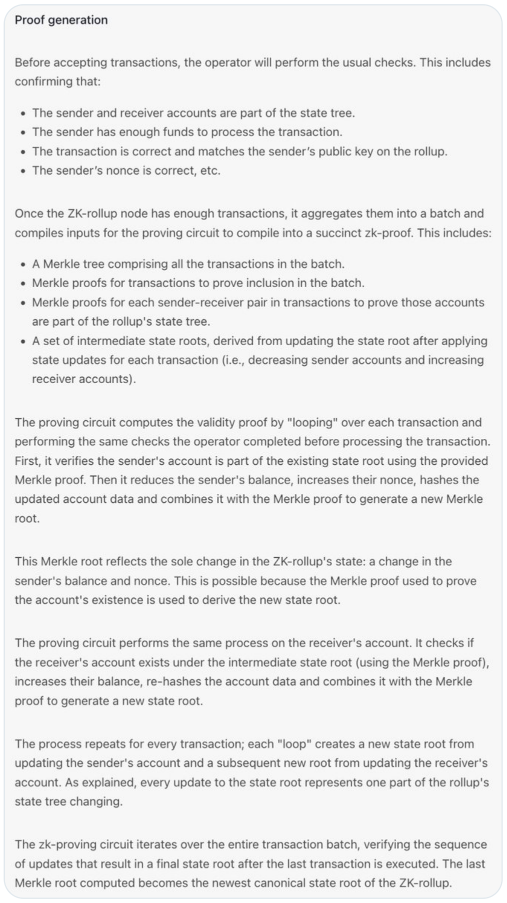
339 Reply Copy link

Read 7 replies

(17/23) ZK-proofs are a category of mathematical proofs that allows one party (prover) to prove to another party (verifier) that a statement is true while also ensuring that the prover does not give the verifier any info that the verifier didn't already have.

(18/23) ZK-proofs are the cutting edge of cryptographic research, details are for another time (maybe).

Here's what you need to know: proof generation is long and hard, but proof verification is quick and easy.



Proof generation

Before accepting transactions, the operator will perform the usual checks. This includes confirming that:

- The sender and receiver accounts are part of the state tree.
- The sender has enough funds to process the transaction.
- The transaction is correct and matches the sender's public key on the rollup.
- The sender's nonce is correct, etc.

Once the ZK-rollup node has enough transactions, it aggregates them into a batch and compiles inputs for the proving circuit to compile into a succinct zk-proof. This includes:

- A Merkle tree comprising all the transactions in the batch.
- Merkle proofs for transactions to prove inclusion in the batch.
- Merkle proofs for each sender-receiver pair in transactions to prove those accounts are part of the rollup's state tree.
- A set of intermediate state roots, derived from updating the state root after applying state updates for each transaction (i.e., decreasing sender accounts and increasing receiver accounts).

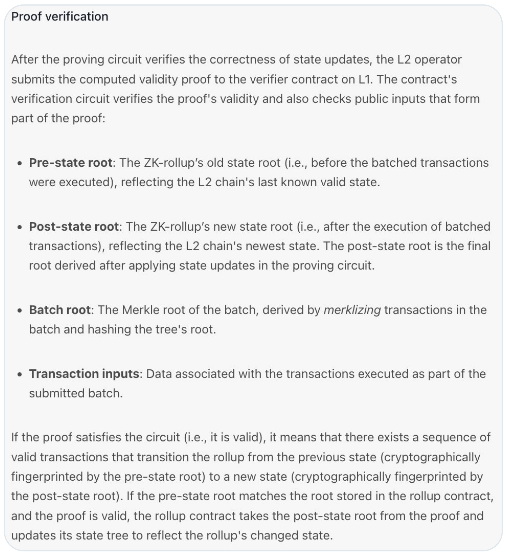
The proving circuit computes the validity proof by "looping" over each transaction and performing the same checks the operator completed before processing the transaction. First, it verifies the sender's account is part of the existing state root using the provided Merkle proof. Then it reduces the sender's balance, increases their nonce, hashes the updated account data and combines it with the Merkle proof to generate a new Merkle root.

This Merkle root reflects the sole change in the ZK-rollup's state: a change in the sender's balance and nonce. This is possible because the Merkle proof used to prove the account's existence is used to derive the new state root.

The proving circuit performs the same process on the receiver's account. It checks if the receiver's account exists under the intermediate state root (using the Merkle proof), increases their balance, re-hashes the account data and combines it with the Merkle proof to generate a new state root.

The process repeats for every transaction; each "loop" creates a new state root from updating the sender's account and a subsequent new root from updating the receiver's account. As explained, every update to the state root represents one part of the rollup's state tree changing.

The zk-proving circuit iterates over the entire transaction batch, verifying the sequence of updates that result in a final state root after the last transaction is executed. The last Merkle root computed becomes the newest canonical state root of the ZK-rollup.



Proof verification

After the proving circuit verifies the correctness of state updates, the L2 operator submits the computed validity proof to the verifier contract on L1. The contract's verification circuit verifies the proof's validity and also checks public inputs that form part of the proof:

- **Pre-state root**: The ZK-rollup's old state root (i.e., before the batched transactions were executed), reflecting the L2 chain's last known valid state.

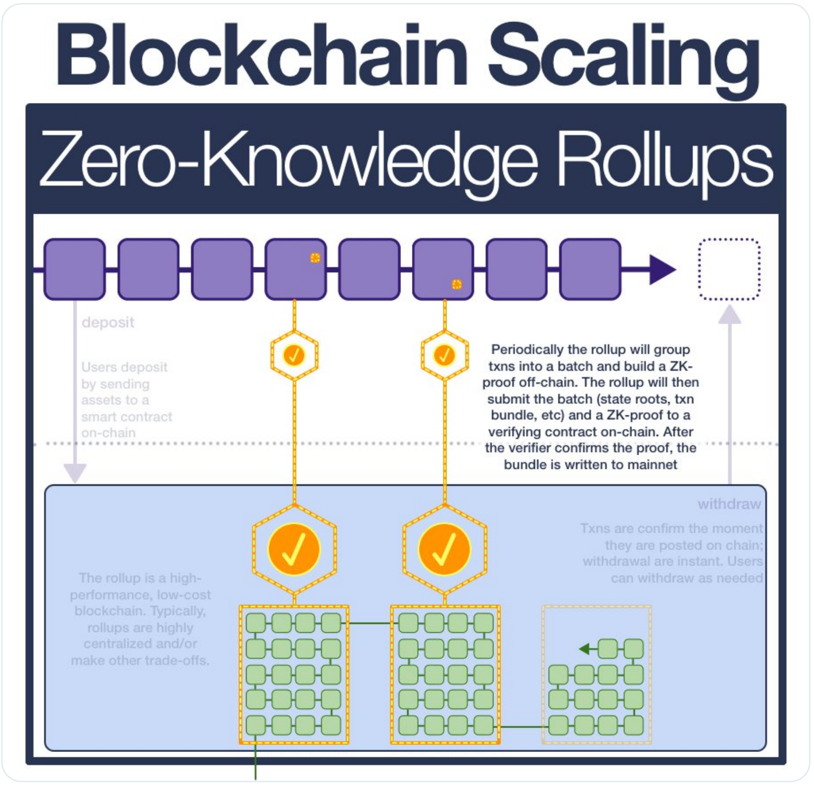
- **Post-state root**: The ZK-rollup's new state root (i.e., after the execution of batched transactions), reflecting the L2 chain's newest state. The post-state root is the final root derived after applying state updates in the proving circuit.

- **Batch root**: The Merkle root of the batch, derived by *merklizing* transactions in the batch and hashing the tree's root.

- **Transaction inputs**: Data associated with the transactions executed as part of the submitted batch.

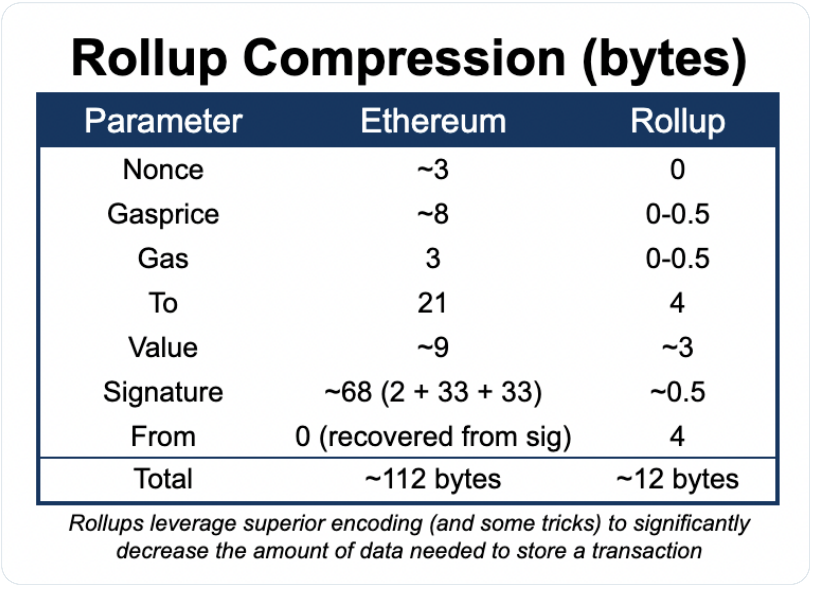
If the proof satisfies the circuit (i.e., it is valid), it means that there exists a sequence of valid transactions that transition the rollup from the previous state (cryptographically fingerprinted by the pre-state root) to a new state (cryptographically fingerprinted by the post-state root). If the pre-state root matches the root stored in the rollup contract, and the proof is valid, the rollup contract takes the post-state root from the proof and updates its state tree to reflect the rollup's changed state.

(19/23) Once every [interval], the rollup operator will bundle the previous transactions and build a ZK-proof (off-chain). It will then submit the batch and the proof to a verifying contract on-chain. If the contract accepts the proof, the batch is finalized instantly.


Blockchain Scaling
Zero-Knowledge Rollups
deposit
Users deposit by sending assets to a smart contract on-chain
Periodically the rollup will group txns into a batch and build a ZK-proof off-chain. The rollup will then submit the batch (state roots, txn bundle, etc) and a ZK-proof to a verifying contract on-chain. After the verifier confirms the proof, the bundle is written to mainnet
withdraw
Txns are confirm the moment they are posted on chain; withdrawal are instant. Users can withdraw as needed
The rollup is a high-performance, low-cost blockchain. Typically, rollups are highly centralized and/or make other trade-offs.

(20/23) ZK-rollups have all the data compression gains of optimistic rollups, and more.

The introduction of the ZK-proof brings mathematically certainty that the batch is valid; much of the block data needed to confirm cryptographic integrity of the chain can be left off.

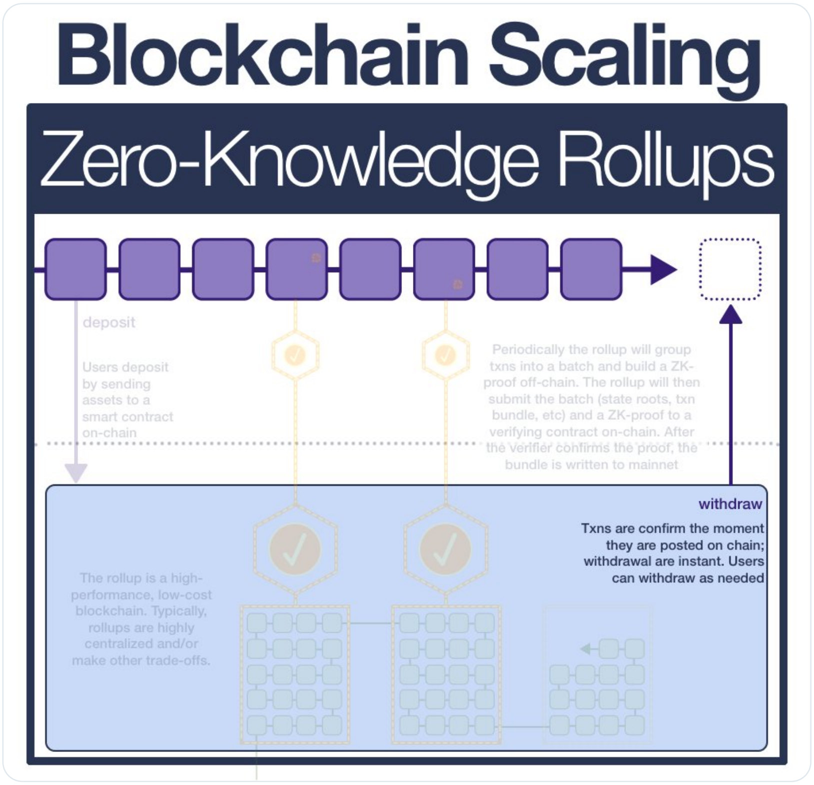


Rollup Compression (bytes)

Parameter	Ethereum	Rollup
Nonce	~3	0
Gasprice	~8	0-0.5
Gas	3	0-0.5
To	21	4
Value	~9	~3
Signature	~68 (2 + 33 + 33)	~0.5
From	0 (recovered from sig)	4
Total	~112 bytes	~12 bytes

Rollups leverage superior encoding (and some tricks) to significantly decrease the amount of data needed to store a transaction

(21/23) Posting a batch requires ZK-proof verification and delivers instant settlement. The rollup can therefore process withdraws instantly (no challenge window required).


Blockchain Scaling
Zero-Knowledge Rollups
deposit
Users deposit by sending assets to a smart contract on-chain
Periodically the rollup will group txns into a batch and build a ZK-proof off-chain. The rollup will then submit the batch (state roots, txn bundle, etc) and a ZK-proof to a verifying contract on-chain. After the verifier confirms the proof, the bundle is written to mainnet
withdraw
Txns are confirm the moment they are posted on chain; withdrawal are instant. Users can withdraw as needed
The rollup is a high-performance, low-cost blockchain. Typically, rollups are highly centralized and/or make other trade-offs.

(22/23) If you take a step back and look at the evolution of @ethereum scaling solutions, you see a theme: offload execution while anchoring settlement to mainnet.

From state channels (a single use, purpose) to ZK-rollups (persistent state, general purpose, instant settlement).

(23/23) In just a couple days, the execution chain will Merge with the Beacon Chain and @ethereum will change forever.

The World Computer is well on its way to 100k transactions per second... you just need to know where to look.

Come join the rest of us with zero-knowledge!

Like what you read? Help me spread the word by retweeting the thread (linked below).

Follow me for more explainers and as much alpha as I can possibly serve.

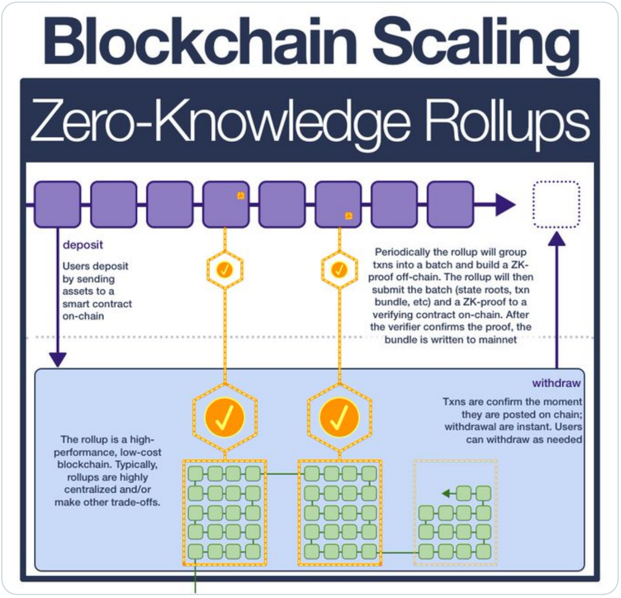
Haym
@SalomonCrypto · **Follow**

(1/23) Blockchain Scaling: Zero-Knowledge Rollups

State Channels → Plasma → Optimistic Rollups → ZK-Rollups

This is how **@ethereum** scales to 100k txn/sec. This is how Ethereum becomes the World Computer.

Your guide to the future of blockchain scaling technology.


Blockchain Scaling
Zero-Knowledge Rollups
deposit
Users deposit by sending assets to a smart contract on-chain
Periodically the rollup will group txns into a batch and build a ZK-proof off-chain. The rollup will then submit the batch (state roots, txn bundle, etc) and a ZK-proof to a verifying contract on-chain. After the verifier confirms the proof, the bundle is written to mainnet
withdraw
Txns are confirm the moment they are posted on chain; withdrawal are instant. Users can withdraw as needed
The rollup is a high-performance, low-cost blockchain. Typically, rollups are highly centralized and/or make other trade-offs.

2:39 PM · Sep 12, 2022

Read the full conversation on Twitter

5 See the latest COVID-19 information on Twitter

Read 2 replies

• • •