A Rhythm of Rollups

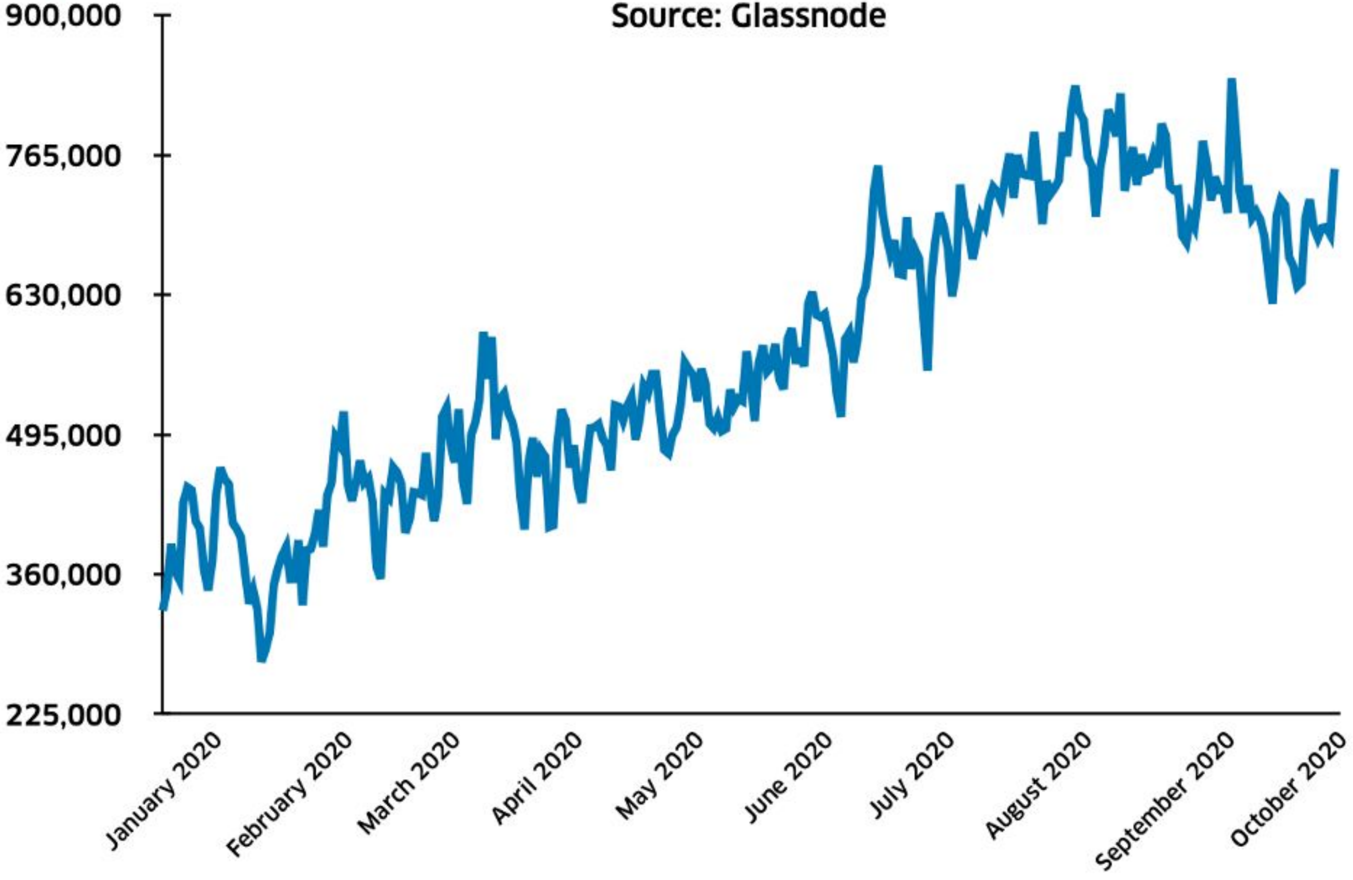
Gokul Alex - Founder of Semiott Systems & Gigamesh Garages

Ethereum Engineering – From an Argonaut to a Jinxified Juggernaut

INTERDAX

External Smart Contract Calls on Ethereum

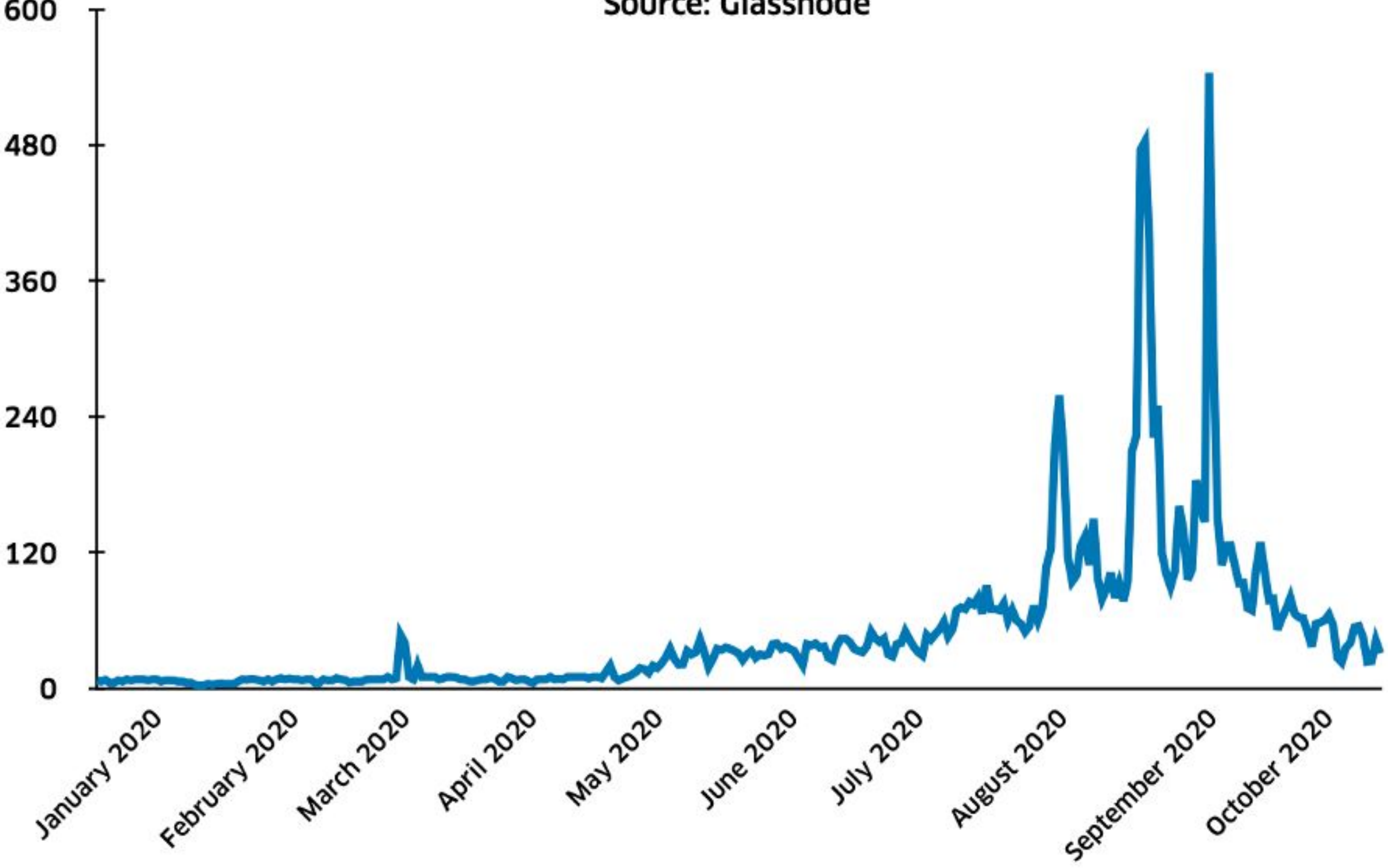
Source: Glassnode



INTERDAX

Ethereum’s Median Gas Price (Gwei)

Source: Glassnode



Emerging Approaches to Ethereum Scalability

First, you can make the blockchain itself have a higher transaction capacity.

The main challenge with this technique is that blockchains with "bigger blocks" are inherently more difficult to verify and likely to become more centralized.

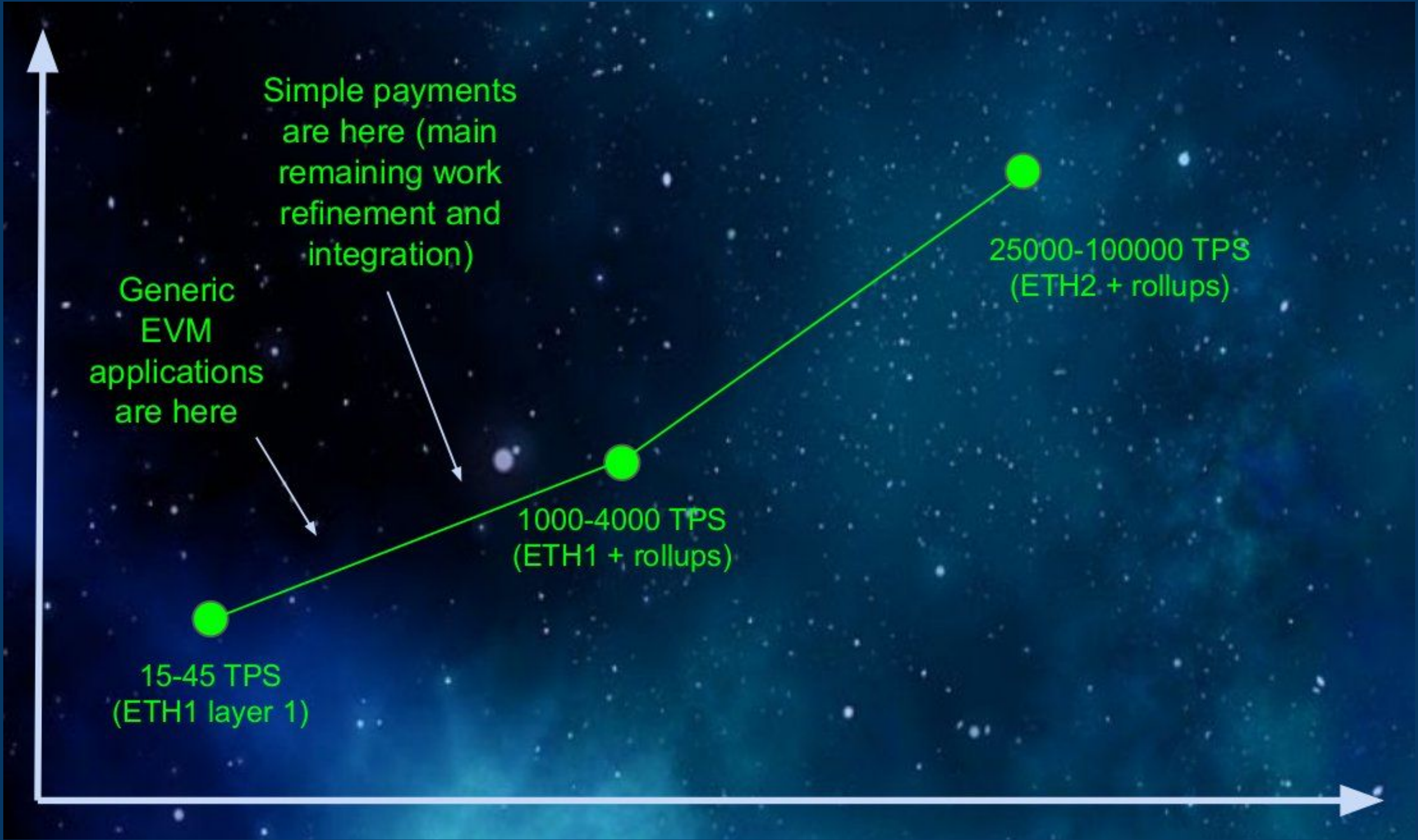
To avoid such risks, developers can either increase the efficiency of client software or, more sustainably, use techniques such as sharding to allow the work of building and verifying the chain to be split up across many nodes; the effort known as "eth2" is currently building this upgrade to Ethereum.

Second, you can change the way that you use the blockchain.

Instead of putting all activity on the blockchain directly, users perform the bulk of their activity off-chain in a "layer 2" protocol.

There is a smart contract on-chain, which only has two tasks: processing deposits and withdrawals, and verifying proofs that everything happening off-chain is following the rules.

There are multiple ways to do these proofs, but they all share the property that verifying the proofs on-chain is much cheaper than doing the original computation off-chain.



Simple payments are here (main remaining work refinement and integration)

Generic EVM applications are here

25000-100000 TPS
(ETH2 + rollups)

1000-4000 TPS
(ETH1 + rollups)

15-45 TPS
(ETH1 layer 1)

What is a Rollup ?

Rollups are solutions that bundle or "roll up" sidechain transactions into a single transaction and generate a cryptographic proof

Rollups in Perspective

A Rollup is an off-chain aggregation of transactions inside an Ethereum smart contract, which reduces fees and congestion by increasing the throughput of the blockchain from its current 15 tps to more than 1,000 tps.

Within the smart contract, users can transact with security guarantees their transactions will not be misused and they will settle to the main chain at some point in the future.

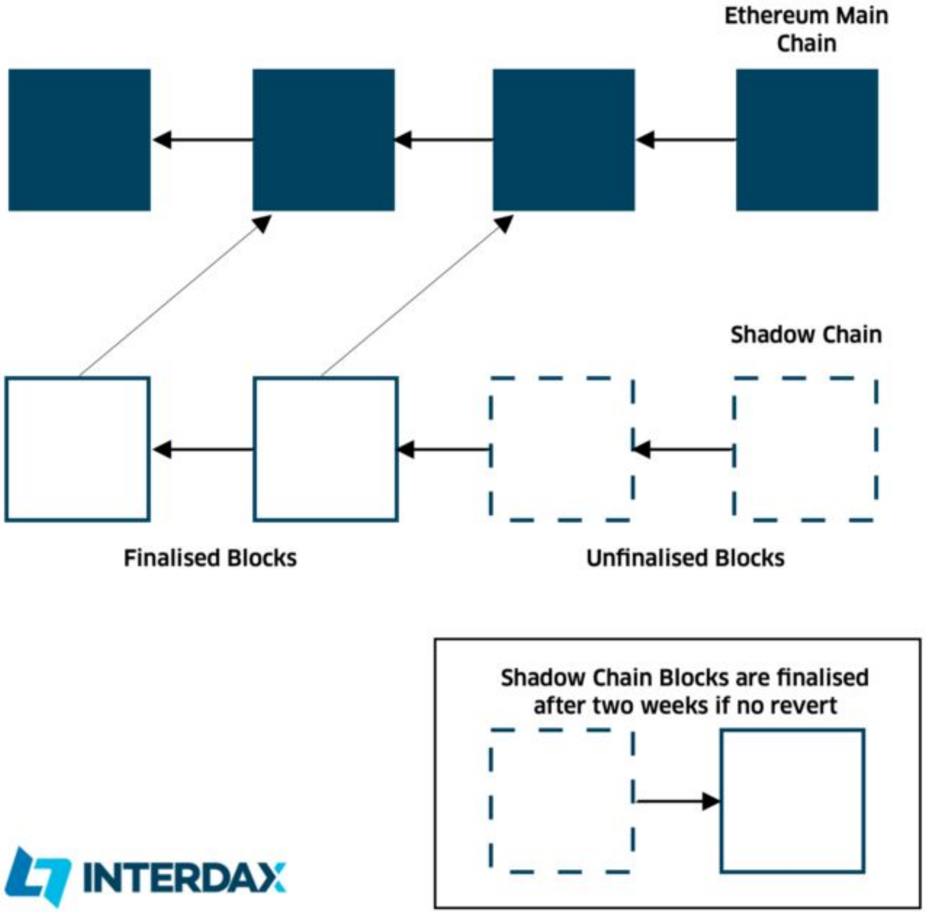
It publishes just enough data on-chain so that any observer can reconstruct the state (i.e., account balances) and detect invalidity.

Rollups History

The concept of Rollups dates back to 2014, described as “shadow chains” by Ethereum co-founder Vitalik Buterin.

The failures of solutions like Plasma and state channels led developers to revisit Buterin’s shadow chains — now known as Rollups.

While Plasma and state channels can scale millions of transactions per second, they are not compatible with the smart contracts that power many of the DeFi applications which have surged in popularity.



Ethereum Main Chain

Shadow Chain

Finalised Blocks

Unfinalised Blocks

Shadow Chain Blocks are finalised after two weeks if no revert

INTERDAX

Shadow Chain Concept in Detail

Participants of a shadow chain do not verify blocks by default. Blocks are finalised after two weeks if there is no revert and the chain runs in parallel to the main chain, but does not shadow all the data.

The main idea is to separate the data (which is relatively cheap to store) and the execution of smart contracts, which uses more gas.

Shadow Chain Concept in Detail

The shadow chain concept takes all the computations occur off-chain while state transitions are committed back to the main chain.

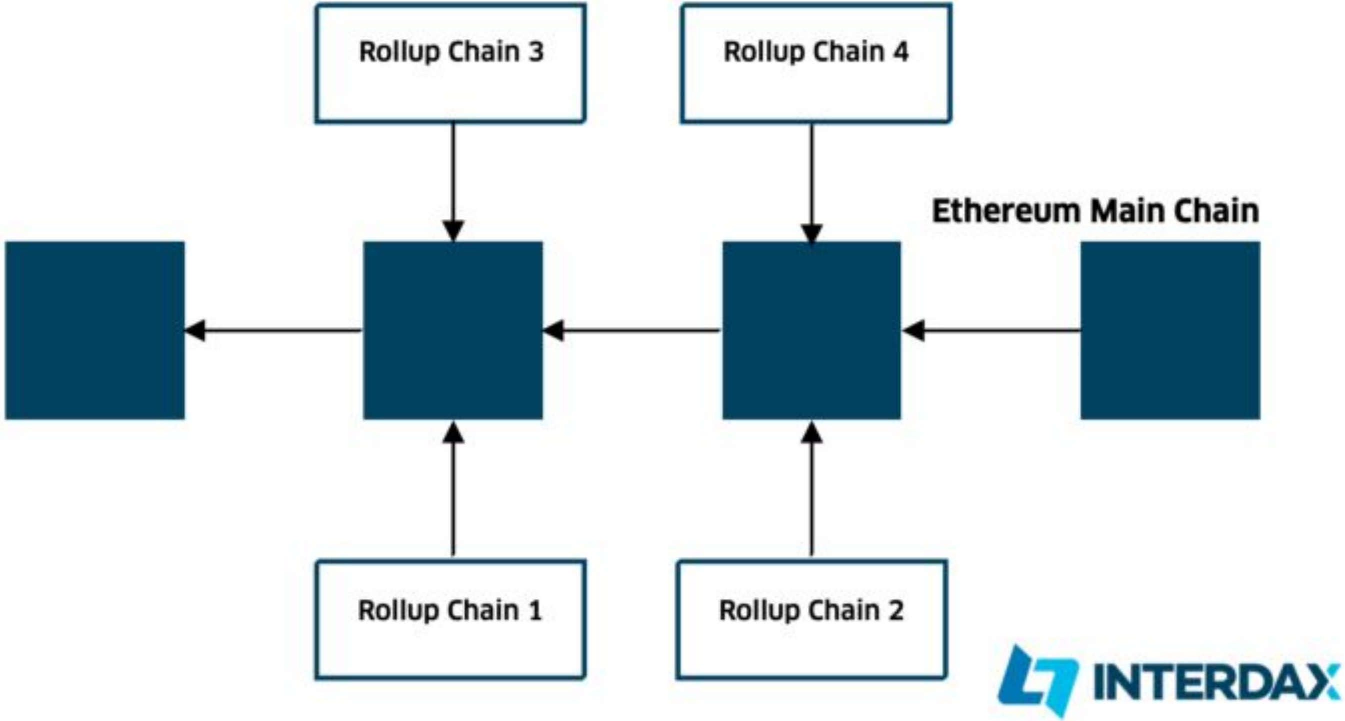
The state transitions are only applied automatically to the main chain if a block is unchallenged after a particular period of time (or number of blocks).

A block is reverted if someone can successfully challenge it before being committed.

Shadow Chain to RollUp Transition

Rollups build on the shadow chains idea by taking execution of the state off-chain and only using the Ethereum blockchain for data availability.

Rollups post blocks or state updates, only publishing some data to the main chain for each transaction via tx CALLDATA, providing an improvement in throughput and overcoming a major hurdle for sidechains: data withholding attacks.



RollUp Chains as Shards

What is a Sidechain?

Sidechain is a Scalability Technique used by Blockchain Protocols

Let us understand Sidechain in Ethereum Perspective ...

Plasma is the name given to the construction scalability method that places layer 2 blocks on top of the Ethereum blockchain in the form of a side chain.

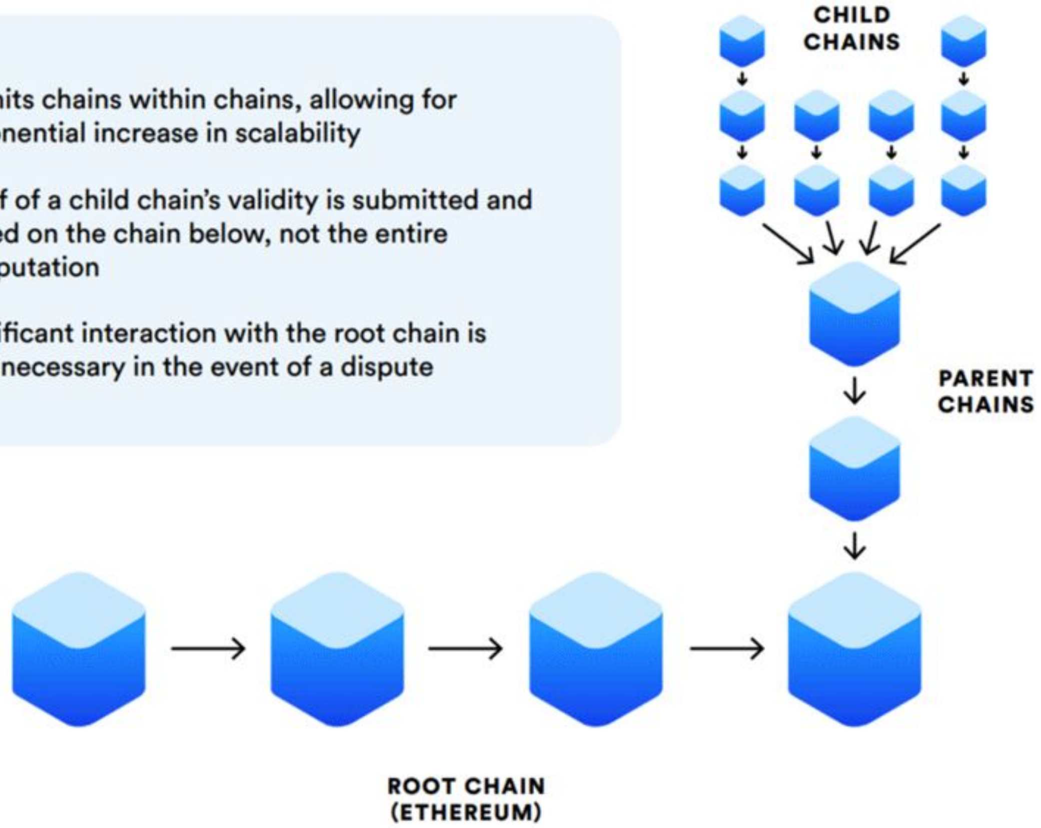
The implementation of Plasma gives the ability of hundreds of side chain transactions to be processed offline with only a single hash of the side chain block being added to the Ethereum blockchain.

Plasma: The Basics

Permits chains within chains, allowing for exponential increase in scalability

Proof of a child chain's validity is submitted and stored on the chain below, not the entire computation

Significant interaction with the root chain is only necessary in the event of a dispute



Challenges with Plasma Model

Game Theoretic Constraints
Engineering Constraints

An exit game must be played for a user to withdraw from the side chain.

It requires side chain users to retain a high amount of data so that enough exists for validation.

Also, a lengthy challenge period requires users to stay online or lose reward.

Benefits of Rollups over Plasma Model

Where Plasma creates one transaction per transfer, Rollups bundle hundreds of transfers into a single transaction. The smart contract will deconstruct and verify all of the transfers held in a single transaction.

The two broad approaches to Rollups (Optimistic Rollups and ZK-Rollups) differ in the way they ensure validity of the sidechain blocks.

For Optimistic Rollups, validity is ensured by a fraud proof and the synchrony assumption, while for ZK-Rollups, validity is ensured by zero-knowledge proofs.

zkRollups in a nutshell

A "zero knowledge proof" approach is used to present and publicly record the validity of the block on the Ethereum blockchain.

ZK reduces computing and storage resources for validating the block by reducing the amount of data held in a transaction; zero knowledge of the entire data is needed.

Zero Knowledge Proofs

History of Zero Knowledge Proofs

The concept of zero-knowledge proofs was first introduced in 1985 by Shafi Goldwasser, Charles Rackoff, and Silvio Micali and actually appeared in The New York Times in 1987.

They designed the notion of knowledge complexity, a metric for the amount of knowledge that is needed to transfer from a prover to a verifier for it to be considered valid.

https://www.nytimes.com/1987/02/17/science/a-new-approach-to-protecting-secrets-is-discovered.html

A NEW APPROACH TO PROTECTING SECRETS IS DISCOVERED

Blending pure logic with computer technology, the researchers are developing an area of mathematics called zero-knowledge proof. Where a conventional proof conveys information, a zero-knowledge proof is meant to convey only the assurance that the information is in hand. The goal is to convince a second party without providing any of the knowledge that would allow him, or an eavesdropper, to convince a third party.

Report Links Acid Rain to 30-Year Decline of Black Ducks

'An extensive regional fish and wildlife resource may be put at great risk' by legislative delay.

Brief U.S. Suppression Of Proof Stirs Anger

Mathematicians see a threat to the free exchange of ideas.

New Approach to Guarding Secrets Developed

Continued From Page C1

"The whole notion of what constitutes a mathematical proof is broadening."

SCIENCE WATCH

Meteors as Communication Links

The essence of zero-knowledge proof lies in an interactive exchange of information between the "prover" and the "verifier."

That is a break with the traditional practice of simply writing a proof down, once and for all – or simply revealing one's password.

''What I find most exciting is that the whole notion of what constitutes a mathematical proof is broadening,''

Manuel Blum of the University of California at Berkeley.

Dr. Micali, Shafi Goldwasser, Charles Rackoff, Oded Goldreich and Avi Wigderson, have shown that graph colorings can be proved in ways that convey zero knowledge of the proof.

Some mathematicians speculate about automated procedures for signing contracts or communicating orders to stockbrokers.

Milestones in ZKP

- 1985- The first zero-knowledge proofs were written about in a paper called "The Knowledge Complexity of Interactive Proof-Systems" by Shafi Goldwasser, Silvio Micali, and Charles Rackoff.
- 2012- Alessandro Chiesa and a team of researchers coin the term zk-SNARKs.
- 2016- Zcash is released and becomes the most used privacy-focused cryptocurrency to use zk-SNARKs .

ZCash and ZKP

Zcash is the first widespread use case and application of zero-knowledge proofs in the crypto world.

The privacy coin uses a form of zero-knowledge proofs called zk-SNARKs which stands for “Zero-Knowledge Succinct Non-Interactive Argument of Knowledge”.

In the basic zero-knowledge proofs we’ve been discussing, provers and verifiers must interact for a few rounds for the verifier to the be convinced of the prover’s honesty.

zkSNARKS

In a zk-SNARKs based protocol, there must be a “trusted setup” to start the system.

The information used at start up—if it fell into the wrong hands—can be used to jeopardize and corrupt the entire system after it is deployed.

In Zcash, the private keys used at launch and computers that processed them were destroyed in a special ceremony.

zkSTARKs

The trusted setup phase is considered a security vulnerability because people need to trust that the information used during the setup was destroyed properly.

To fix zk-SNARK's shortcomings, zk-STARKs was created. Zk-STARKs is a type of zero-knowledge proof that does not require the vulnerable trusted setup phase and also claims to be more scalable and efficient than zk-SNARKs.

AZTEC Protocol

Aztec is a project seeking to bring zero-knowledge proofs to the existing Ethereum network by building a stack of privacy-focused smart contracts.

These fully private smart contracts could be used to create private Ethereum tokens and decentralized organizations (DAOs).

The Ethereum upgrade, codenamed Istanbul, was specifically designed to reduce the costs of zero-knowledge proofs like the ones used by Aztec.

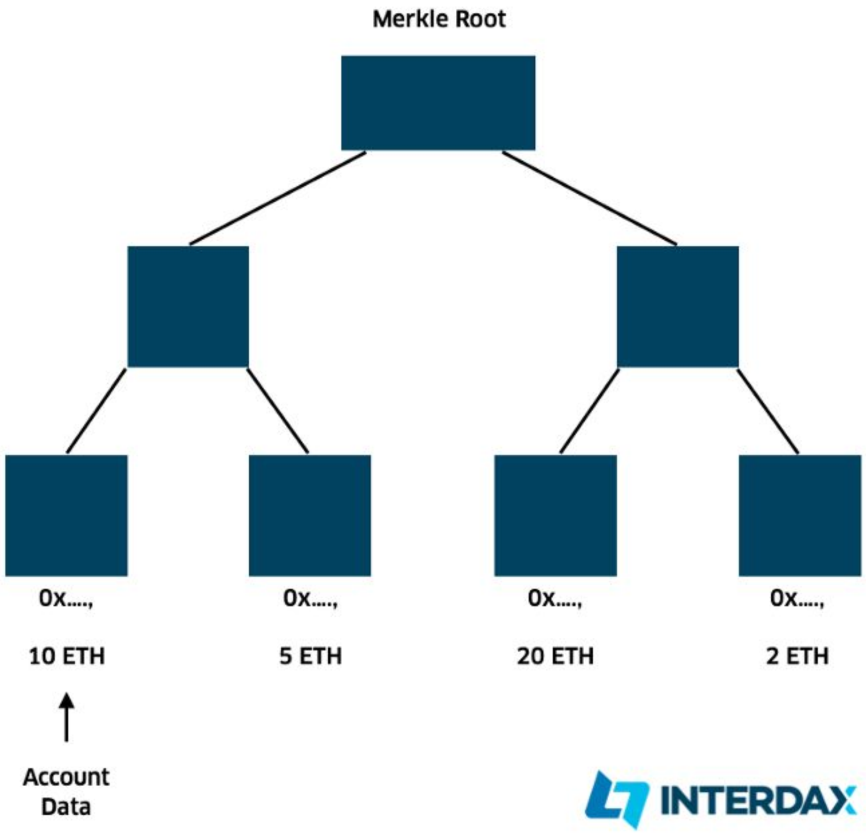
Deep Dive into zkRollups

History of zkRollups

Initially proposed by Barry Whitehat in 2018, ZK-Rollups have the same security guarantees as the underlying L1 (i.e., Ethereum), the potential to produce blocks in under a minute and increase throughput to as high as 2,000 tps. Projects working on ZK-Rollup implementations include Matter Labs and Starkware.

Instead of waiting two weeks for a block in the Shadow Chain to become finalised, ZK-Rollups replace fraud challenges with zero-knowledge proofs.

Accounts and balances are represented by separate Merkle Trees. These Merkle Tree roots ensure no one can fake the data. The roots of each Merkle Tree (one for accounts, the other for balances) are both stored in a smart contract on Ethereum which provides a succinct representation of the state of the sidechain. All other data is stored off-chain.



Merkle Tree in RollUps

zkRollups Workflow

- The ZK-Rollup scheme consists of two types of users: transactors and relayers.
- Transactors create their transfer and broadcast the transfer to the network.
- The transfer data consists of an indexed "to" and "from" address, a value to transact, the network fee, and nonce.
- A shortened 3 byte indexed version of the addresses reduces processing resource needs.
- The value of the transaction being greater than or less than zero creates a deposit or withdrawal respectively.
- The smart contract records the data in two Merkle Trees; addresses in one Merkle Tree and transfer amounts in another.

zkRollup Components

Relayers collect a large amount of transfers to create a rollup. It is the relayers job to generate the SNARK proof.

The SNARK proof is a hash that represents the delta of the blockchain state. State refers to "state of being."

SNARK proof compares a snapshot of the blockchain before the transfers to a snapshot of the blockchain after the transfers (i.e. wallet values) and reports only the changes in a verifiable hash to the mainnet.

It is worth noting that anyone can become a relayer so long as they have staked the required bond in the smart contract. This incentivises the relayer not to tamper with or withhold a rollup.

zkRollup Lifecycle

Transactions are aggregated together, then signed for and committed to the main chain with just the header. Therefore, the amount of data stored on the Ethereum chain is reduced.

All signatures are replaced by a zero-knowledge proof known as ZK-SNARK, which enables the compression of the aggregated transactions.Computation is also improved since the verification of each signature is replaced with a single SNARK.

The only little bits of information that are added to the main chain are the two Merkle roots of the address book and balances/nonces, both of which are 32-byte fields, amounting to a small part of the transaction data that is published on-chain via CALLDATA.

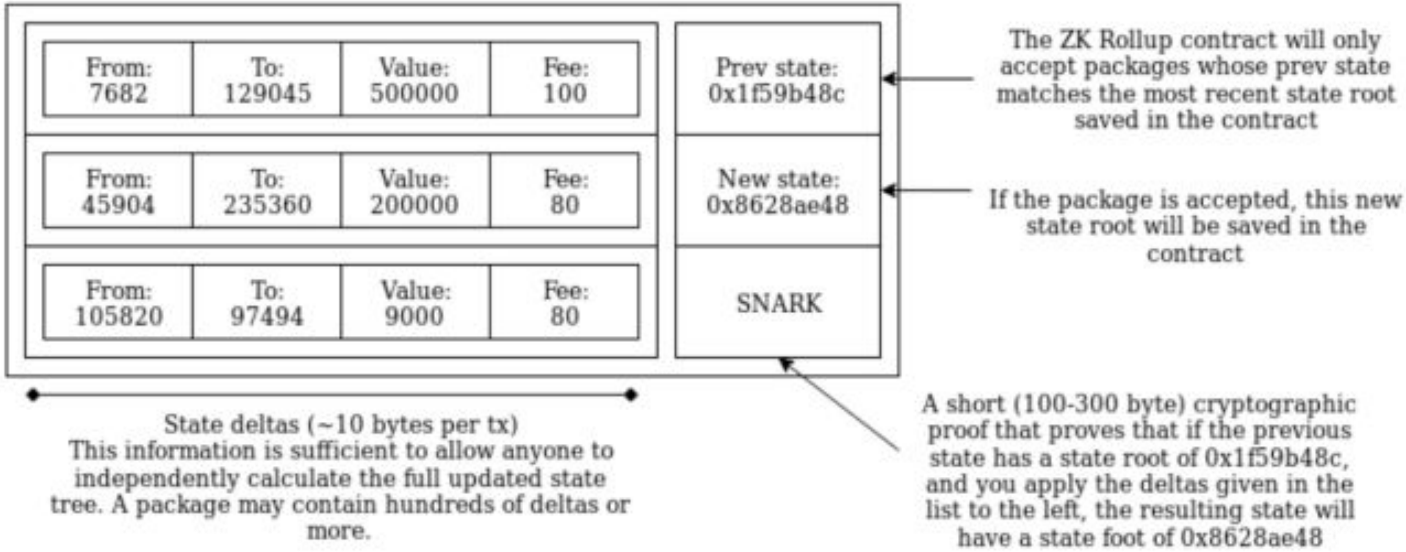
Once the block or state update is submitted, users can check the zero-knowledge proof for validity.

Comparison to Other Techniques

Scaling solutions like Optimistic Rollups and Plasma rely on a challenge to an incorrect state or root hash, where a challenging transaction tells the smart contract that some data is incorrect, and the block is invalidated as a result.

These challenges are replaced in ZK-Rollups with ZK-SNARKs, where it is impossible for the relayers to submit an invalid or incorrect state.

The SNARK proves that the series of transactions were correctly signed by owners, and that the updates to the account balances were correct, leading from the old Merkle root to the new Merkle root. Consequently, it is impossible for the relayers to commit an invalid or manipulated state.



From: 7682	To: 129045	Value: 500000	Fee: 100	Prev state: 0x1f59b48c
From: 45904	To: 235360	Value: 200000	Fee: 80	New state: 0x8628ae48
From: 105820	To: 97494	Value: 9000	Fee: 80	SNARK

The ZK Rollup contract will only accept packages whose prev state matches the most recent state root saved in the contract

If the package is accepted, this new state root will be saved in the contract

State deltas (~10 bytes per tx)
This information is sufficient to allow anyone to independently calculate the full updated state tree. A package may contain hundreds of deltas or more.

A short (100-300 byte) cryptographic proof that proves that if the previous state has a state root of 0x1f59b48c, and you apply the deltas given in the list to the left, the resulting state will have a state foot of 0x8628ae48

Anatomy of a ZK Rollup package that is published on-chain. Hundreds of "internal transactions" that affect the state (ie. account balances) of the ZK Rollup system are compressed into a package that contains ~10 bytes per internal transaction that specifies the state transitions, plus a ~100-300 byte SNARK proving that the transitions are all valid.

zkRollup Contract Model