ZK Rollup: scaling with zero-knowledge proofs

Alex Gluchowski
Ethereum Foundation
Matter Labs

ZK Rollup is a L2 sidechain resembling Plasma, but

- Data availability is solved by broadcasting data to Ethereum
- Fraud challenges are replaced with zero-knowledge proofs

Problems ZK Rollup solves

- No liveness assumption
- Same security as L1
- Capital efficiency
- Censorship-resistance via decentralized operators
- No need for mass exits
- No operational security risks

Current limitations

- Throughput limit of 500 TPS

  => Ethereum does 5 TPS, Paypal 160 TPS

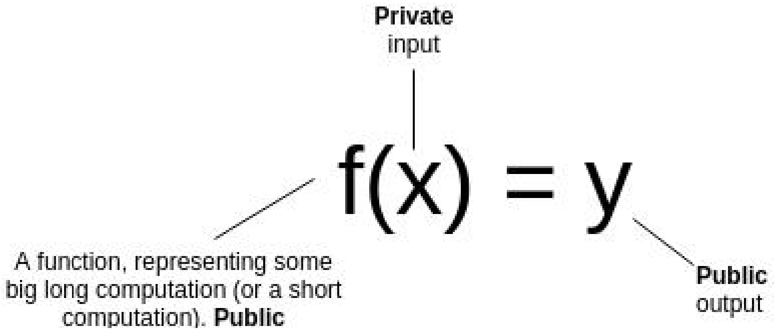
  => More possible in the future

- Fixed functionality: ERC20 transfers, swaps, state channels

  => Smart contracts possible after EIP1829

Zero Knowledge Proofs

“Zero-knowledge” proofs allow one party (the prover) to prove to another (the verifier) that a statement is true, without revealing any information beyond the validity of the statement itself.


Private input
f(x) = y
A function, representing some big long computation (or a short computation). **Public**
Public output

Succint ZKP for scaling: SNARKs vs. STARKs

	Prover complexity	Verifier complexity	Communicational complexity	Proof size
SNARKs	O(N log N)	O(1)	O(1)	288 bytes (sic!)
STARKs	O(N polylog N)	O(polylog N)	O(polylog N)	~100 KB

SNARKS tradeoff: trusted setup

MPC ceremony:

- N participants
- If everyone retains “toxic waste”, proofs can be counterfeited
- Thus: single honest one is sufficient
- Secure logistics is difficult

Solution: Universal Trusted Setup

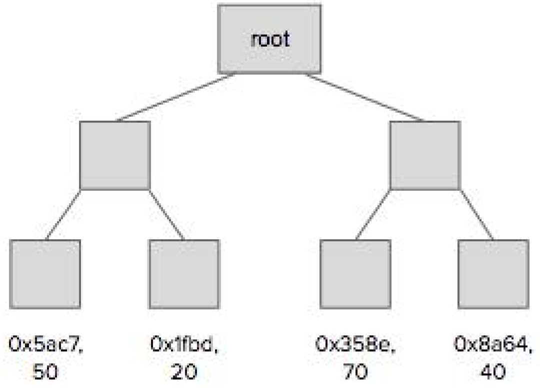
Sonic SNARK for smart contracts

- Same asymptotic scaling characteristics
- Universal trusted setup for all circuits
- Implementation under development

Rollup architecture

Sidechain state: account model

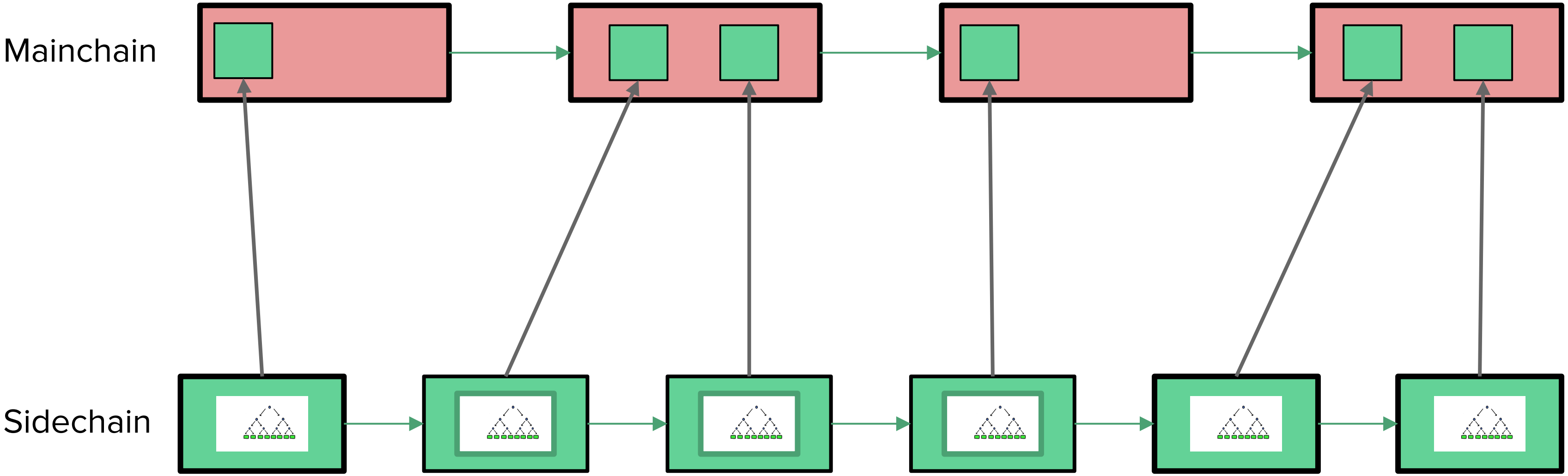
Contract stores state as Merkle root of `{account ID => (pubkey, balance)}`



Merkle root

Account data

Sidechain workflow



Rollup SNARK circuit

Relayer gathers set of transactions t[1]...t[n], creates ZK SNARK:

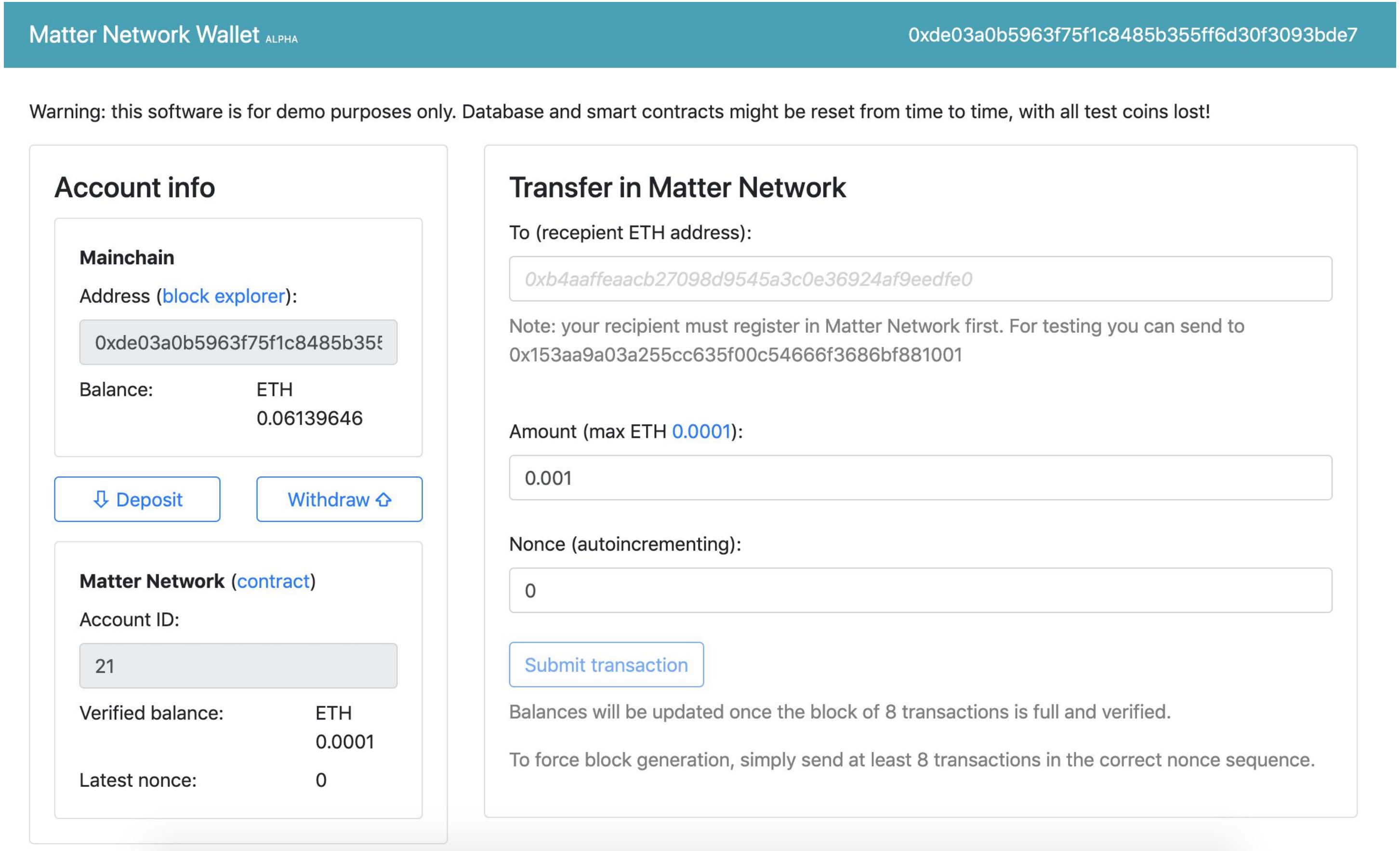
- STF(PRE_STATE, t[1] .... t[n]) = POST_STATE
- Each t[i] has valid signature
- root(PRE_STATE) = r1
- root(POST_STATE) = r2

Data availability in rollup

- Posting small part of the tx data on-chain (CALLDATA)
- For simple token transfers 9 bytes is enough:
  - From (3 bytes)
  - To (3 bytes)
  - Amount (2 bytes, floating point)
  - Fee (1 byte, floating point)

Limits throughput on Ethereum today to 500 TPS :(

PoC demo https://demo.matter-labs.io



Matter Network Wallet ALPHA

0xde03a0b5963f75f1c8485b355ff6d30f3093bde7

Warning: this software is for demo purposes only. Database and smart contracts might be reset from time to time, with all test coins lost!

Account info

Mainchain

Address (block explorer):

0xde03a0b5963f75f1c8485b35!

Balance: ETH 0.06139646

Matter Network (contract)

Account ID:

21

Verified balance: ETH 0.0001

Latest nonce: 0

Transfer in Matter Network

To (recepient ETH address):

0xb4aaffeaacb27098d9545a3c0e36924af9eedfe0

Note: your recipient must register in Matter Network first. For testing you can send to 0x153aa9a03a255cc635f00c54666f3686bf881001

Amount (max ETH 0.0001):

0.001

Nonce (autoincrementing):

0

Balances will be updated once the block of 8 transactions is full and verified.

To force block generation, simply send at least 8 transactions in the correct nonce sequence.

Roadmap for v1.0

[✓] Bellman community edition (Ethereum-compatible)

[✓] PoC live on testnet (ETH transfers)

[✓] Powers of tau MPC test run at ETH Paris

[✓] Production-grade multi-server prover

[   ] *Circuits & contracts for atomic swaps + state channels*

[   ] Security audit

Thank you!

stay tuned

@the_matter_labs