=nil; Foundation

zkSharding: A New Dimension of Scaling L2 on Ethereum

Ilya Marozau

Who I am?

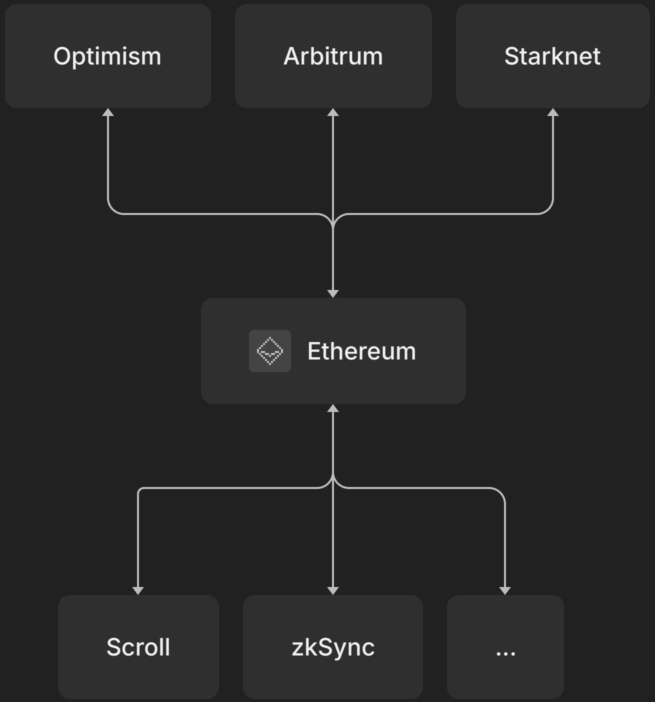
Hi!
I am Ilya

Protocol Researcher & Lead of Analytics at @nil_foundation

Security Researcher at @univienna

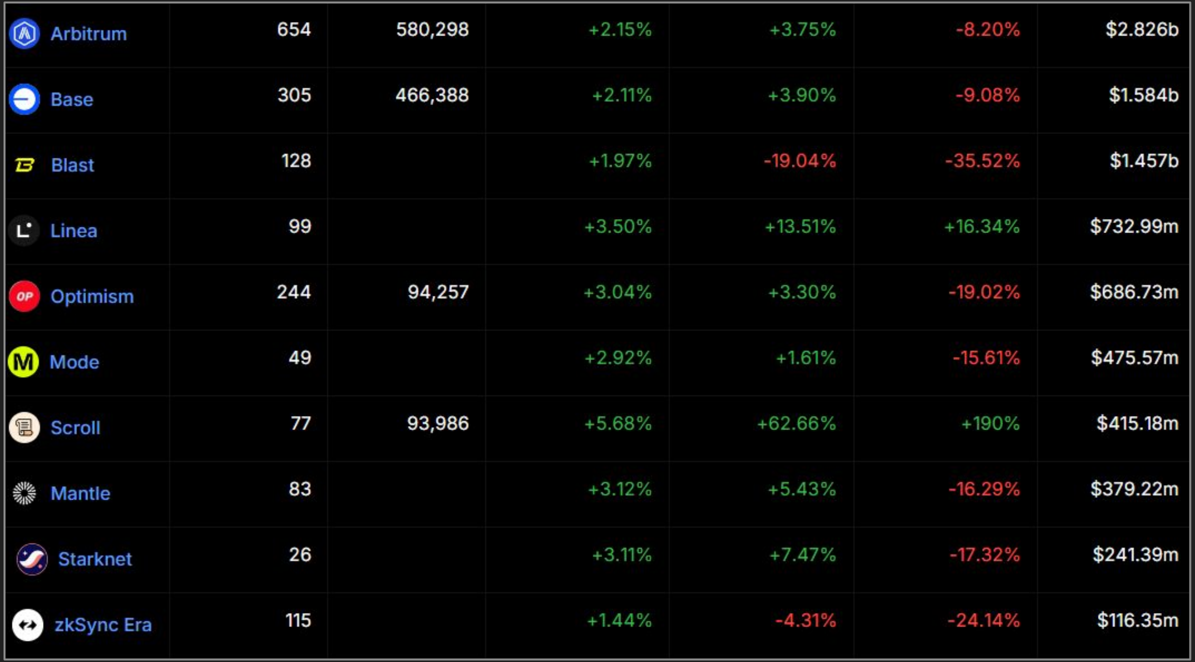
Rollup-Centric approach

“The Ethereum ecosystem is likely to be all-in on rollups (plus some plasma and channels) as a scaling strategy for the near and mid-term future.” – Vitalik Buterin, Oct. 2020



L2 state of art

- Over 30 in mainnet
- Over 15 known in testnet
- Many more planned (e.g. AAVE, ENS)



Arbitrum	654	580,298	+2.15%	+3.75%	-8.20%	$2.826b
Base	305	466,388	+2.11%	+3.90%	-9.08%	$1.584b
Blast	128		+1.97%	-19.04%	-35.52%	$1.457b
Linea	99		+3.50%	+13.51%	+16.34%	$732.99m
Optimism	244	94,257	+3.04%	+3.30%	-19.02%	$686.73m
Mode	49		+2.92%	+1.61%	-15.61%	$475.57m
Scroll	77	93,986	+5.68%	+62.66%	+190%	$415.18m
Mantle	83		+3.12%	+5.43%	-16.29%	$379.22m
Starknet	26		+3.11%	+7.47%	-17.32%	$241.39m
zkSync Era	115		+1.44%	-4.31%	-24.14%	$116.35m

Rollup-Centric approach issues

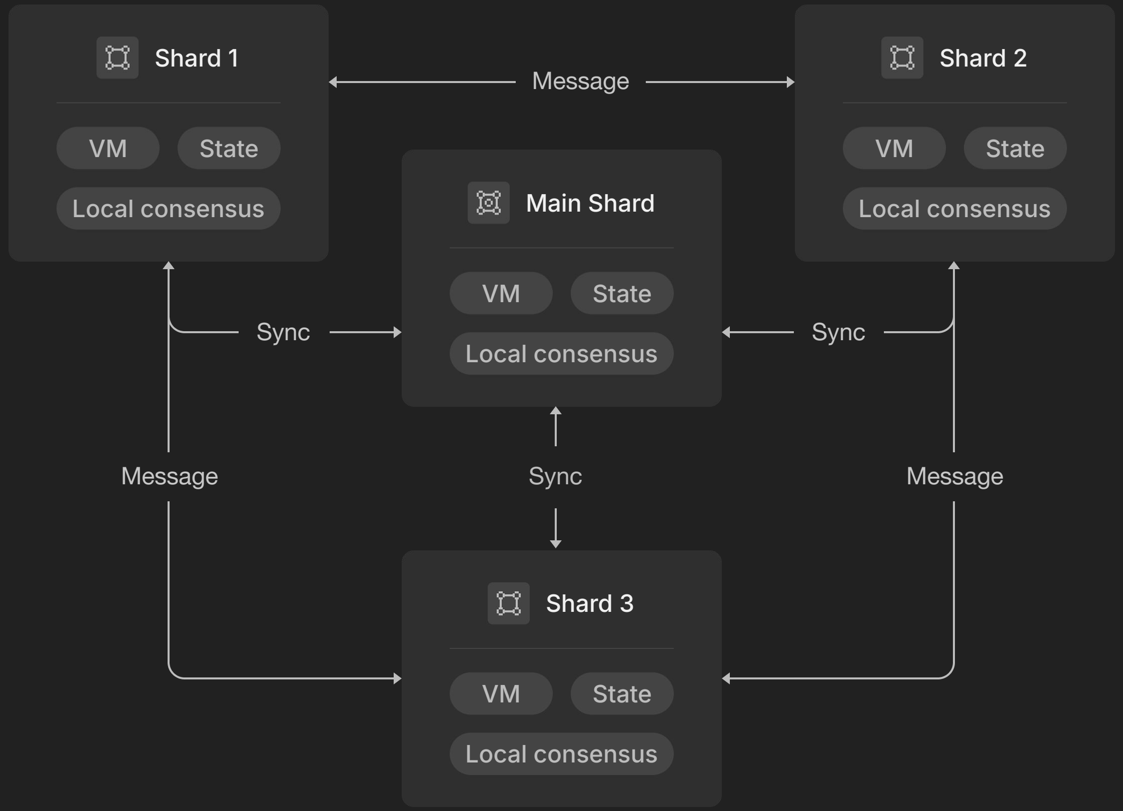
- Too many L2 brings fragmentation of liquidity among solutions
- Quite easy to build the new solution (fork) – leads to unclear security and sustainability
- No scalability of applications
- Limited possibilities for decentralization due to low liquidity
- Limited potential for scalability and performance improvement – L3/L4, VM optimizations, EIP improvements (e.g. 4844)

Sharding – true parallelism over decentralized network

“A database shard, or simply a shard, is a horizontal partition of data in a database or search engine. Each shard is held on a separate database server instance, to spread load” – Wiki

Decentralized Ledger Shard – is a partition of global data, with non-blocking state transition.

Sharding concept

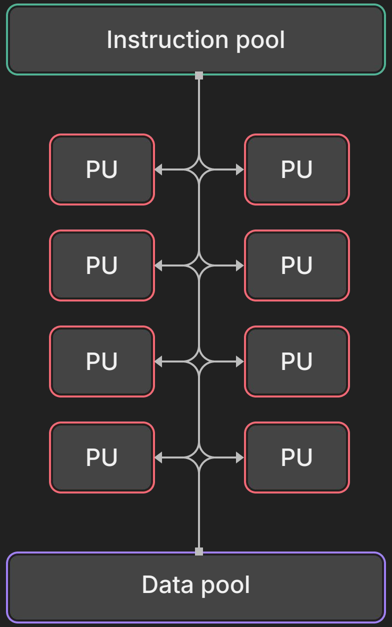


Sharding – MIMD computations

- Single execution at a time is very old concept, defined by Flynn's taxonomy as SISD – single instruction single data
- Later it was replaced with MIMD computations, that stands for multiple instructions multiple data – number of processors that function asynchronously and independently
- Sharding defines each node as computational unit that runs asynchronous, where instruction poll is transactions and data pool – “shared” state

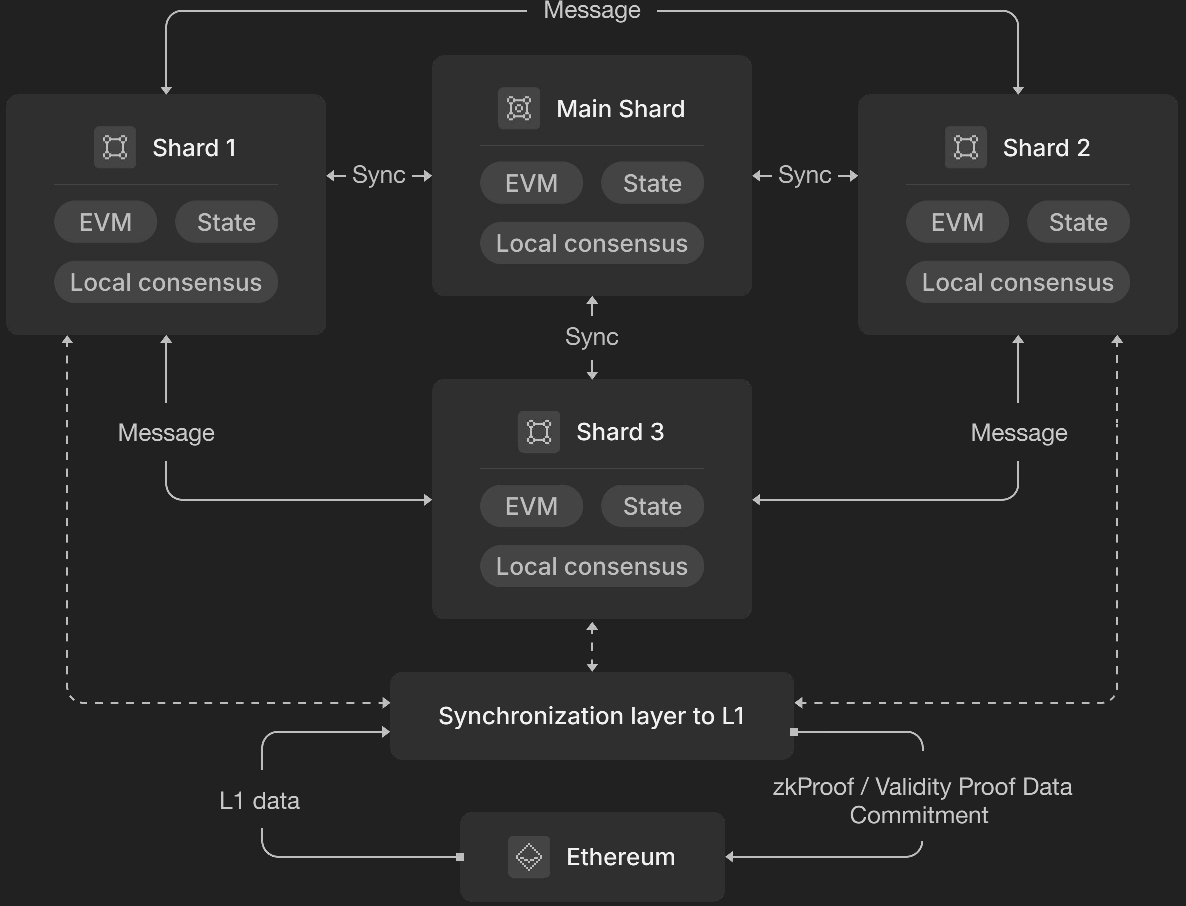
From Amdahl's law we know that parallelization is limited with sequential computations. To address the questions of load distribution sharding is separated into static and dynamic.

MIMD



L2 Sharding

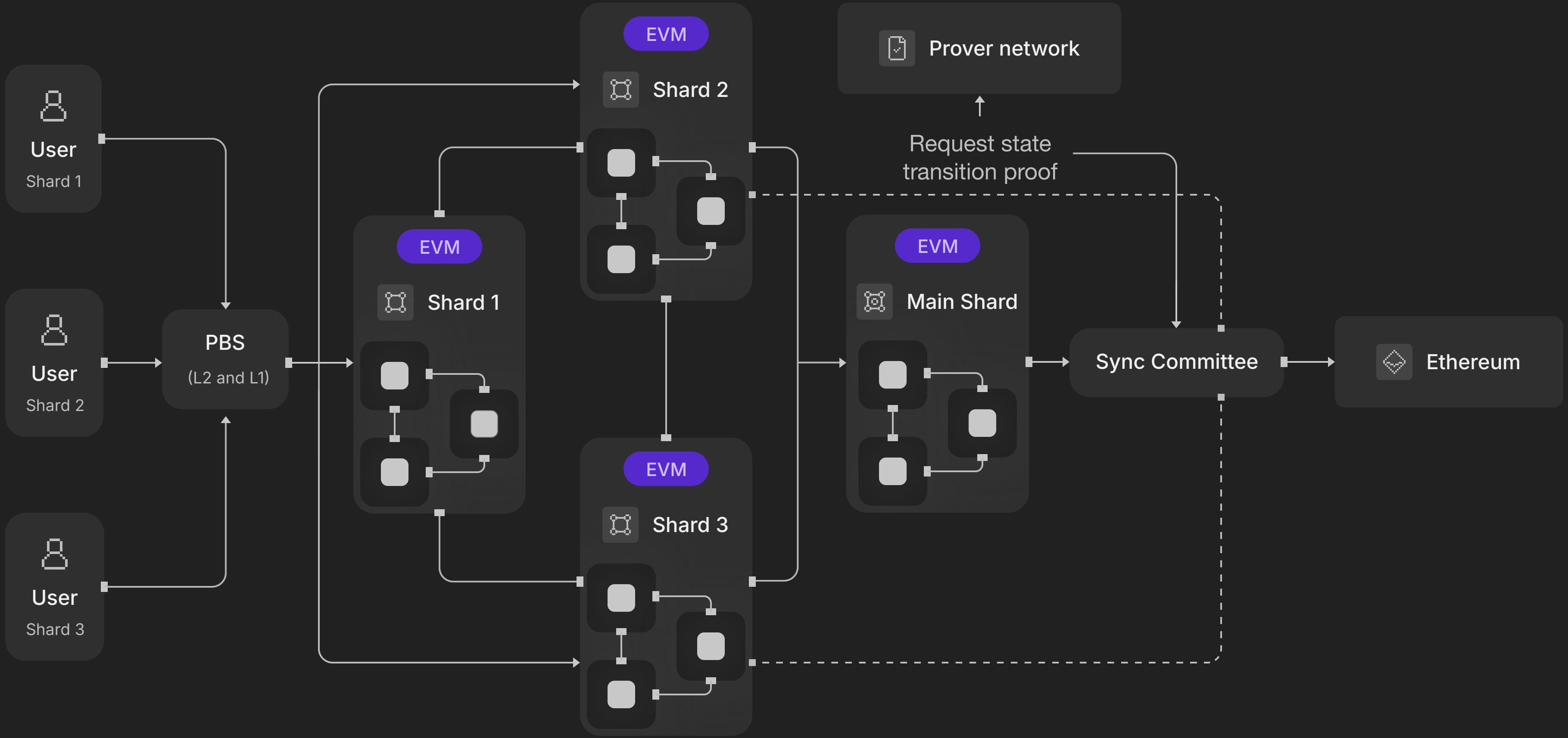
Ethereum is the settlement and data availability layer



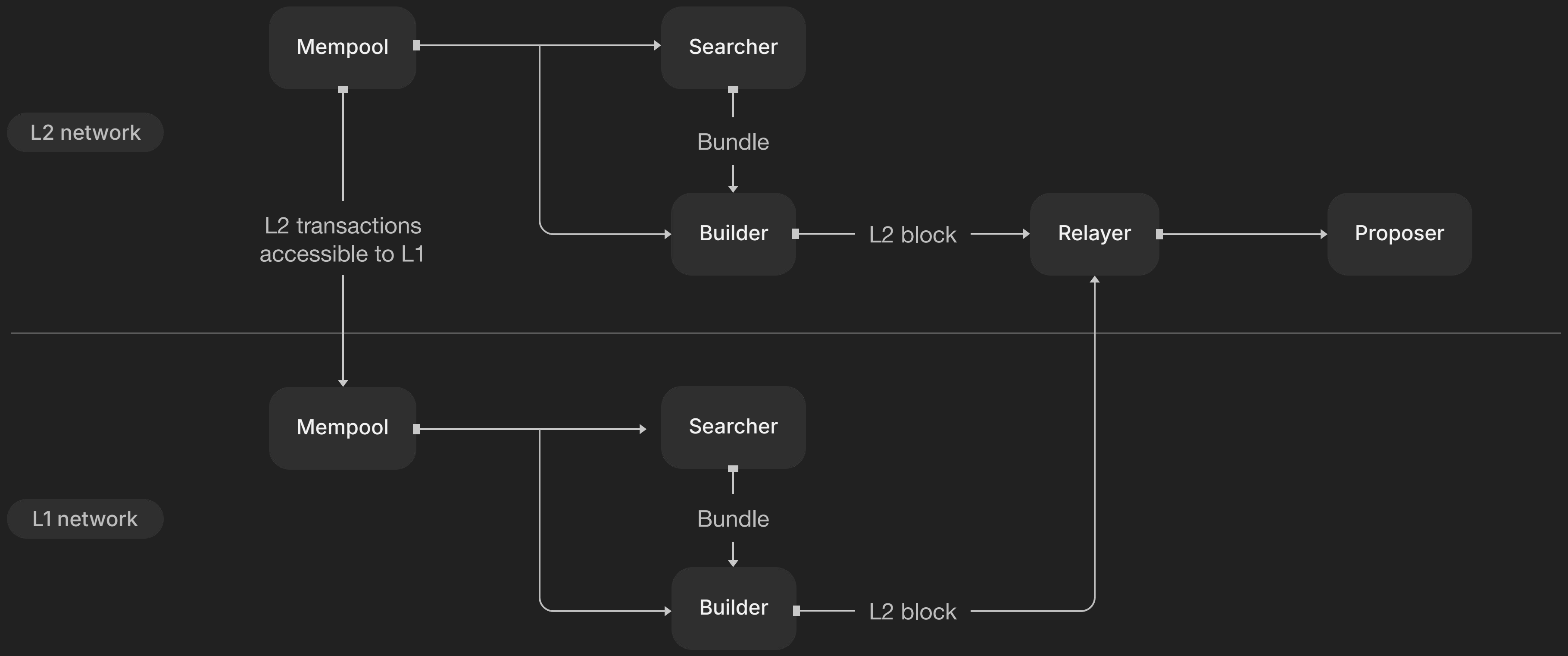
L2 Sharding + ZK EVM => zkSharding

- No fragmentation of liquidity
- Single validator set rotated between shards
- Unlimited horizontal scalability
- Scalability of applications
- Seamless execution environment with parallel processing and storage
- Fast messaging protocol
- Compatibility with and full utilization of zkEVM power

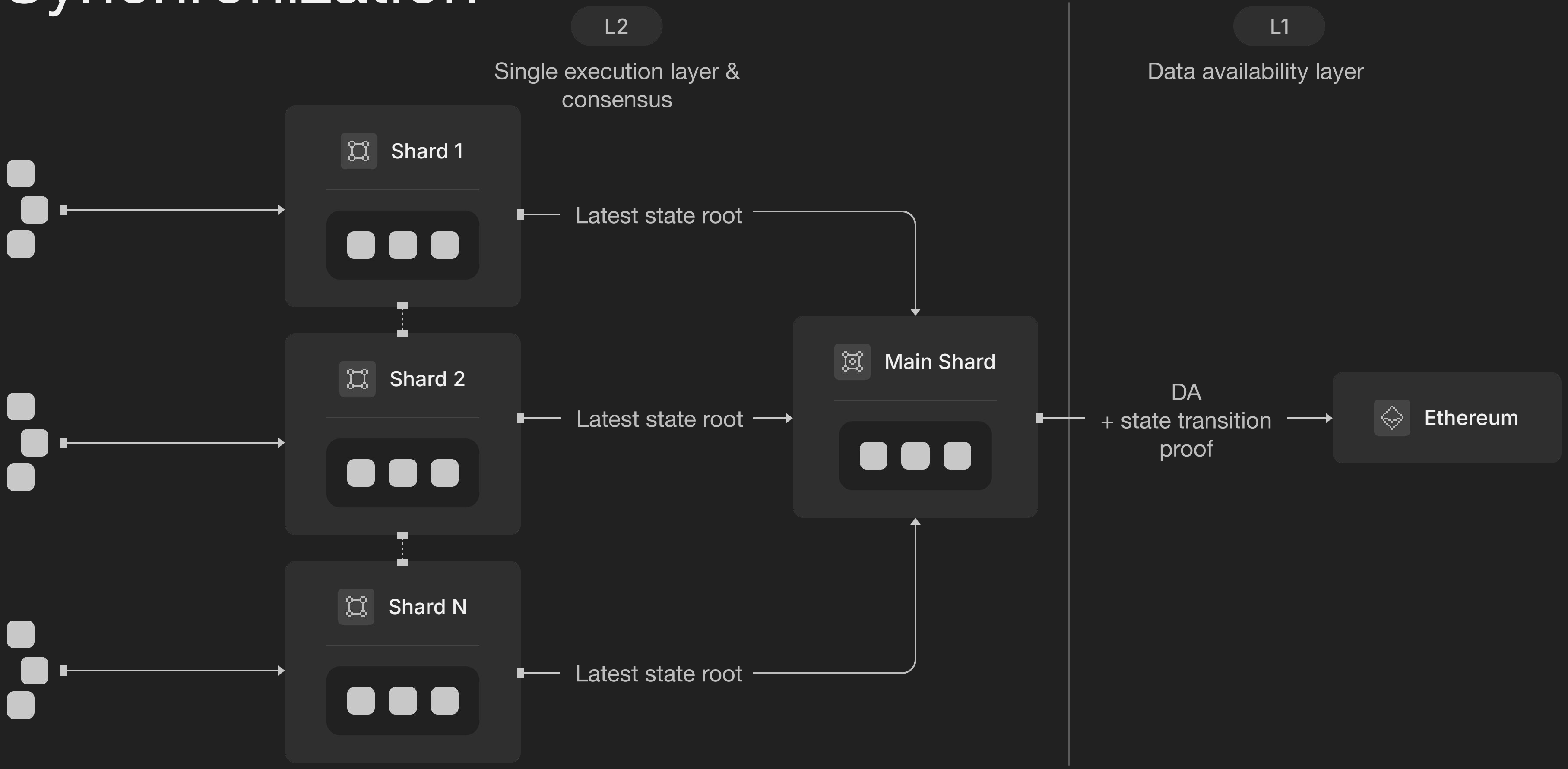
zkSharding step by step



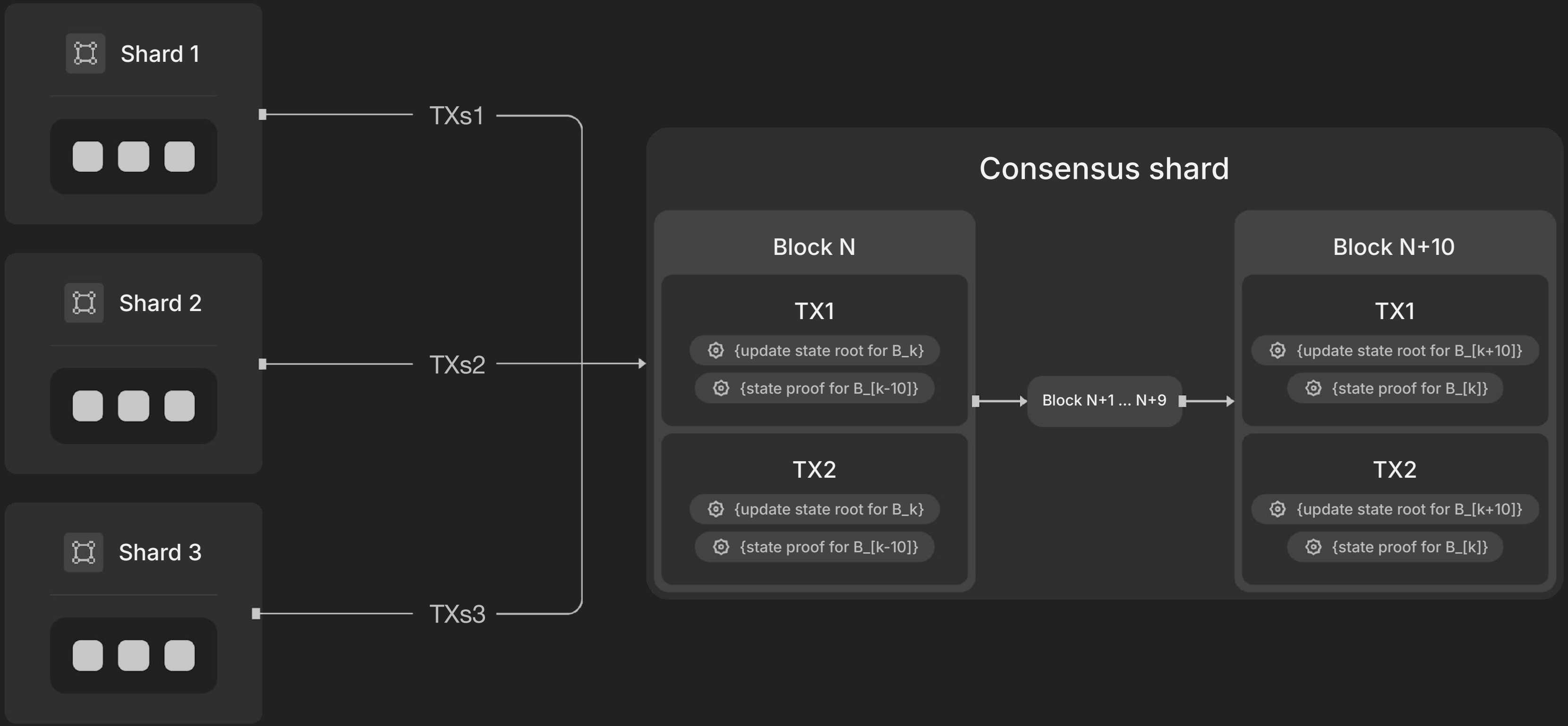
Sequencing


L2 network
Mempool
Searcher
Bundle
Builder
L2 block
Relayer
Proposer
L2 transactions accessible to L1
L1 network
Mempool
Searcher
Bundle
Builder
L2 block

Synchronization



Main shard


Shard 1
Shard 2
Shard 3
TXs1
TXs2
TXs3
Consensus shard
Block N
TX1
{update state root for B_k}
{state proof for B_[k-10]}
TX2
{update state root for B_k}
{state proof for B_[k-10]}
Block N+1 ... N+9
Block N+10
TX1
{update state root for B_[k+10]}
{state proof for B_[k]}
TX2
{update state root for B_[k+10]}
{state proof for B_[k]}

Consensus

Local Consensus

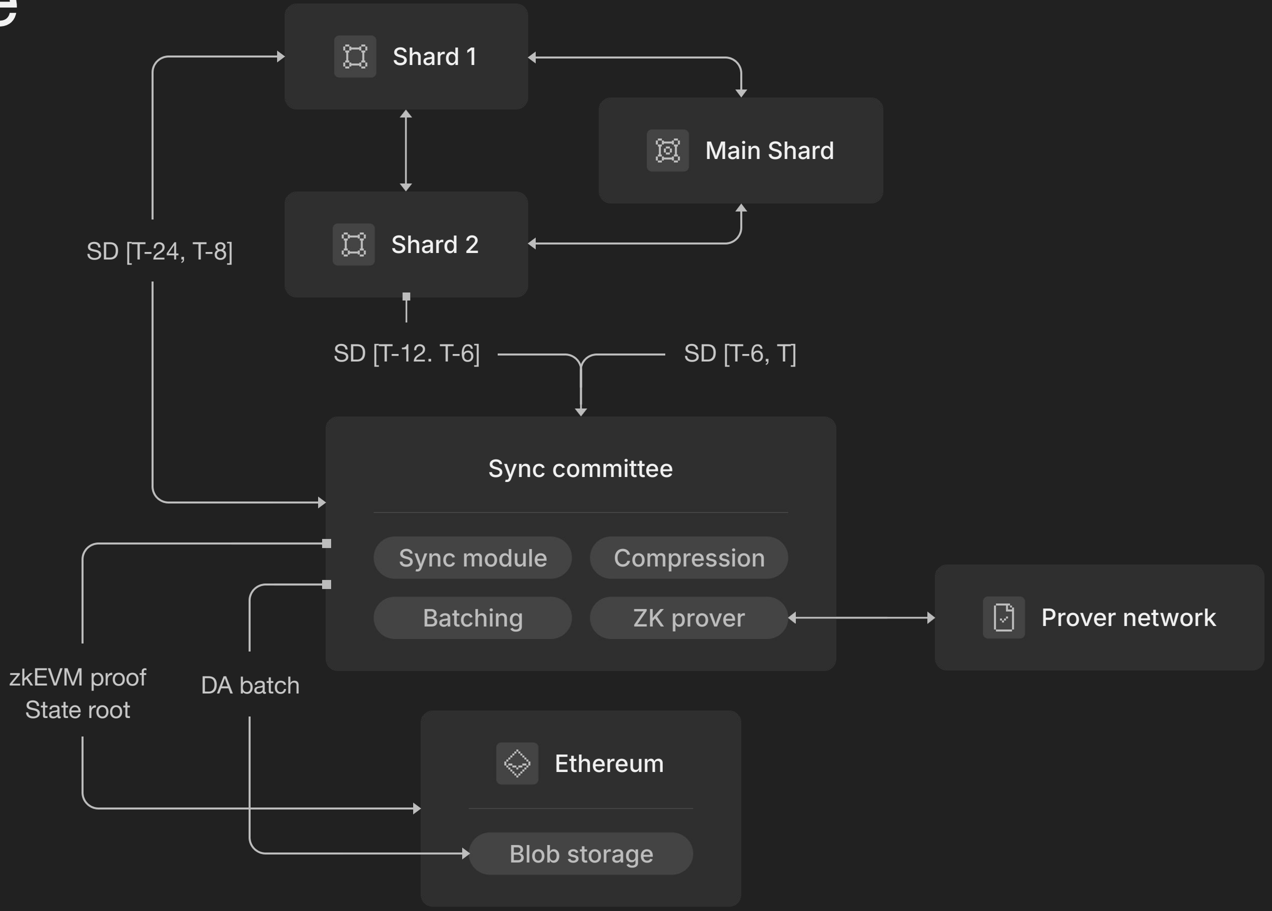
- Each shard – is a standalone network (blockchain). It runs its own consensus called "Local"
- Local consensus has not much specific to sharding, other than inclusion validation of cross-shard messages
- Operates over PBFT mechanism based on Hotstuff 2. As number of validators is rather limited not much load on communication

Global Consensus

- Sharding needs rotation of validators between shards and updates for main shard – this is where it comes to play
- The whole validator set is responsible for operation of main shard to mitigate bottleneck and attacks risks

Sync committee

Extra layer to off-load validators and provide synchronization for DA and Settlement commitments.

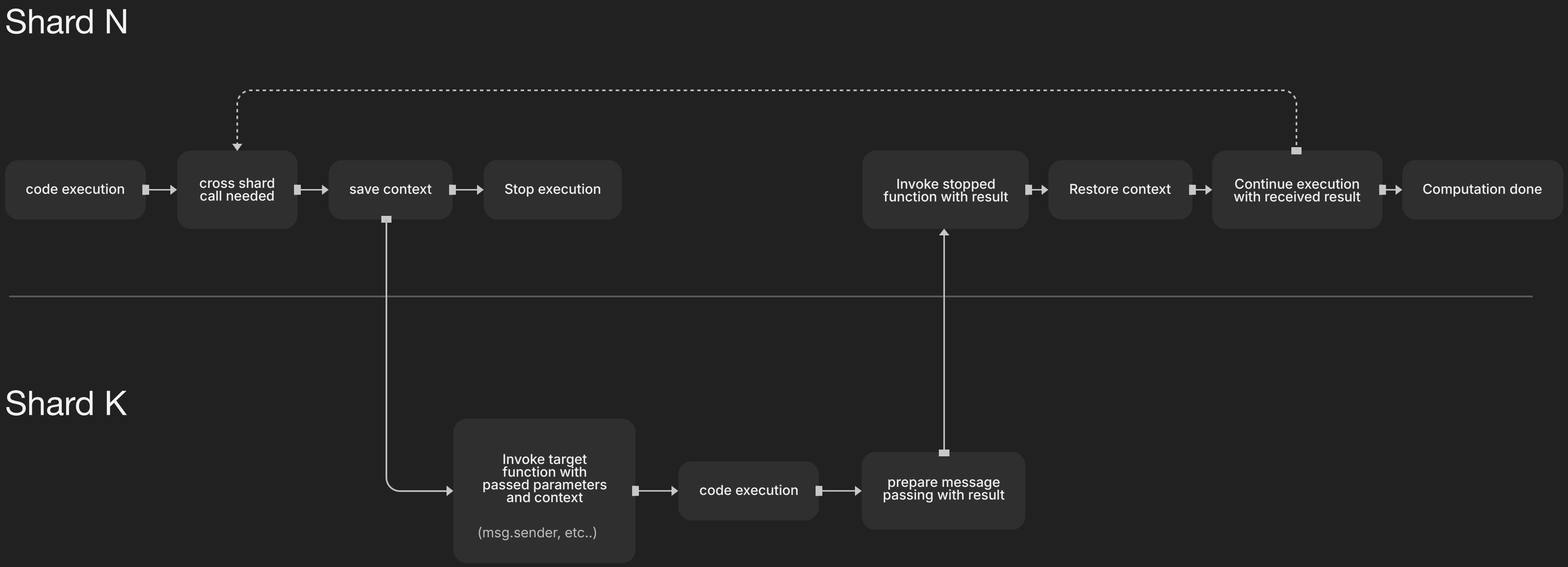


Cross-shard communication

- Important protocol mechanism that provides message transferring from one shard to another
- It provides context saving and delivery guarantees (exp. 1 second) for the message
- Each message saved on source and destination chain as part of block
- Messages emitted during transaction validation and “sent” when block accepted by chain
- Protocol has address resolution to quickly resolve address<->shard requests

Asynchronous environment

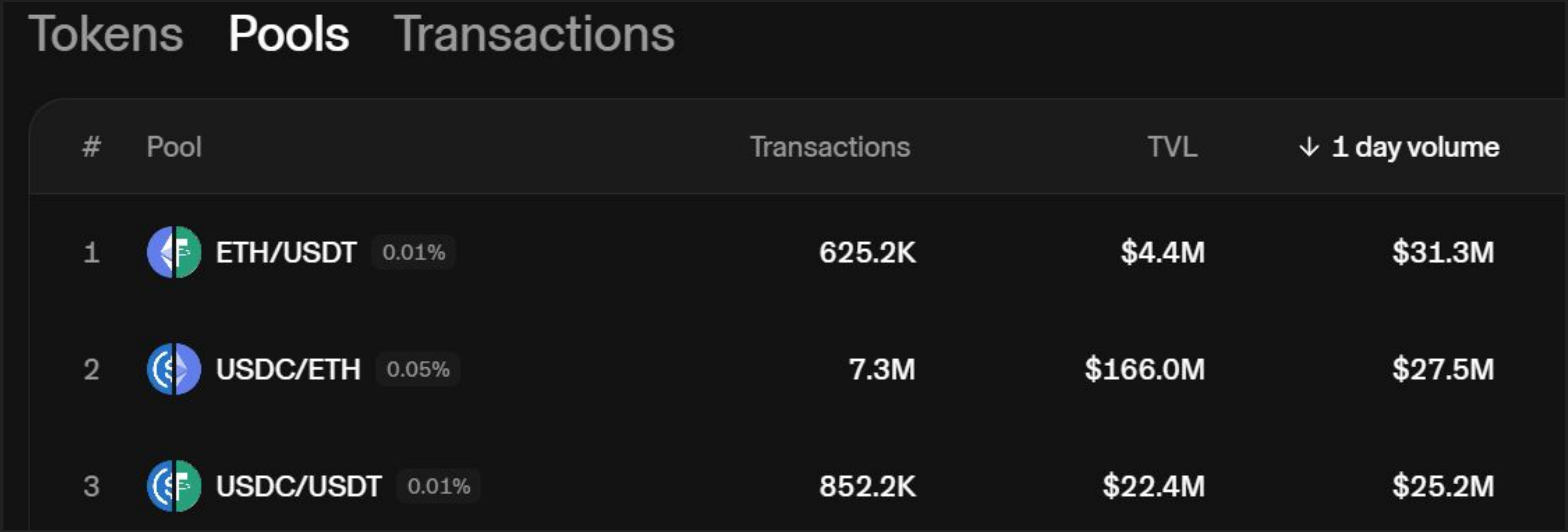
- Sharding introduce new possibilities one of them is asynchronous execution
- Cross-shard calls/messages are non blocking of execution.


Shard N
code execution
cross shard call needed
save context
Stop execution
Invoke stopped function with result
Restore context
Continue execution with received result
Computation done
Shard K
Invoke target function with passed parameters and context
(msg.sender, etc..)
code execution
prepare message passing with result

Application scaling

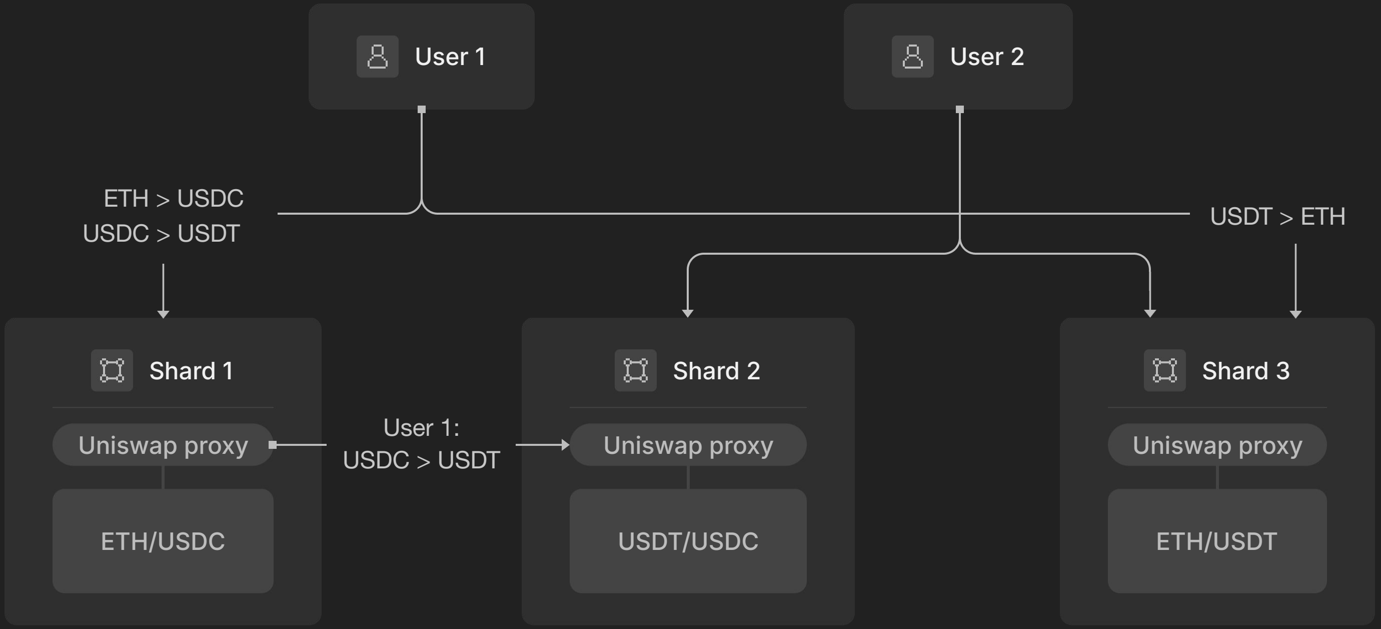
zkSharding introduce horizontal scaling of applications as well. Unique and flexible environment allows development of novel on-chain solutions.

Let’s identify possibility for DEX scaling:



#	Pool	Transactions	TVL	↓ 1 day volume
1	ETH/USDT 0.01%	625.2K	$4.4M	$31.3M
2	USDC/ETH 0.05%	7.3M	$166.0M	$27.5M
3	USDC/USDT 0.01%	852.2K	$22.4M	$25.2M

Example of DEX scaling with sharding



Final thoughts and conclusion

- Sharding opens a new dimension (horizontal) in scaling applications and networks
- Horizontal scaling has formally unlimited potential for performance improvement

- Despite obvious benefits sharding introduce very new exciting challenges starting from network to execution layer
- The technology was successfully applied on number of L1 project, and now it finds it’s apply on L2

Thank you for your attention!

My X profile
@cm0o0cky

=nil; Foundation on X
@nil_foundation

Devnet Launch