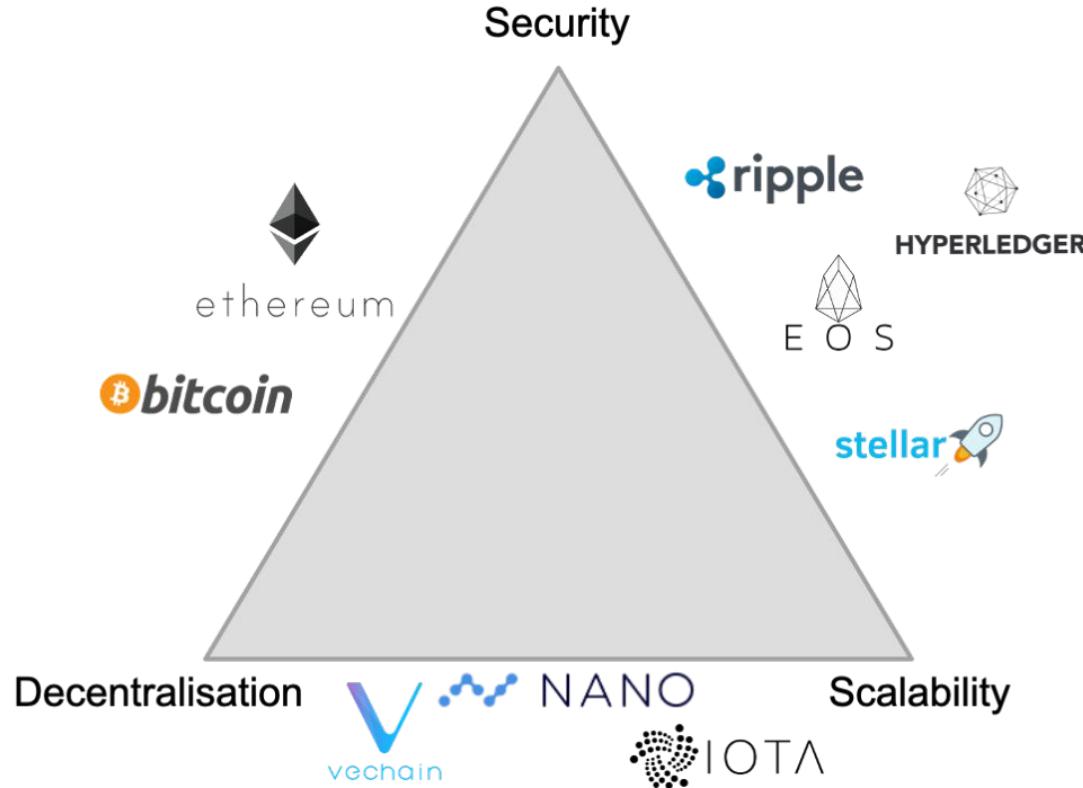


# Multivers<sup>x</sup>

## Case study

**Public?**  
Community

# Blockchain Trilemma



# Proof of Stake



Validating capacity depends on the stake in the network



Validators do not receive a block reward, instead, they collect transaction fees as reward



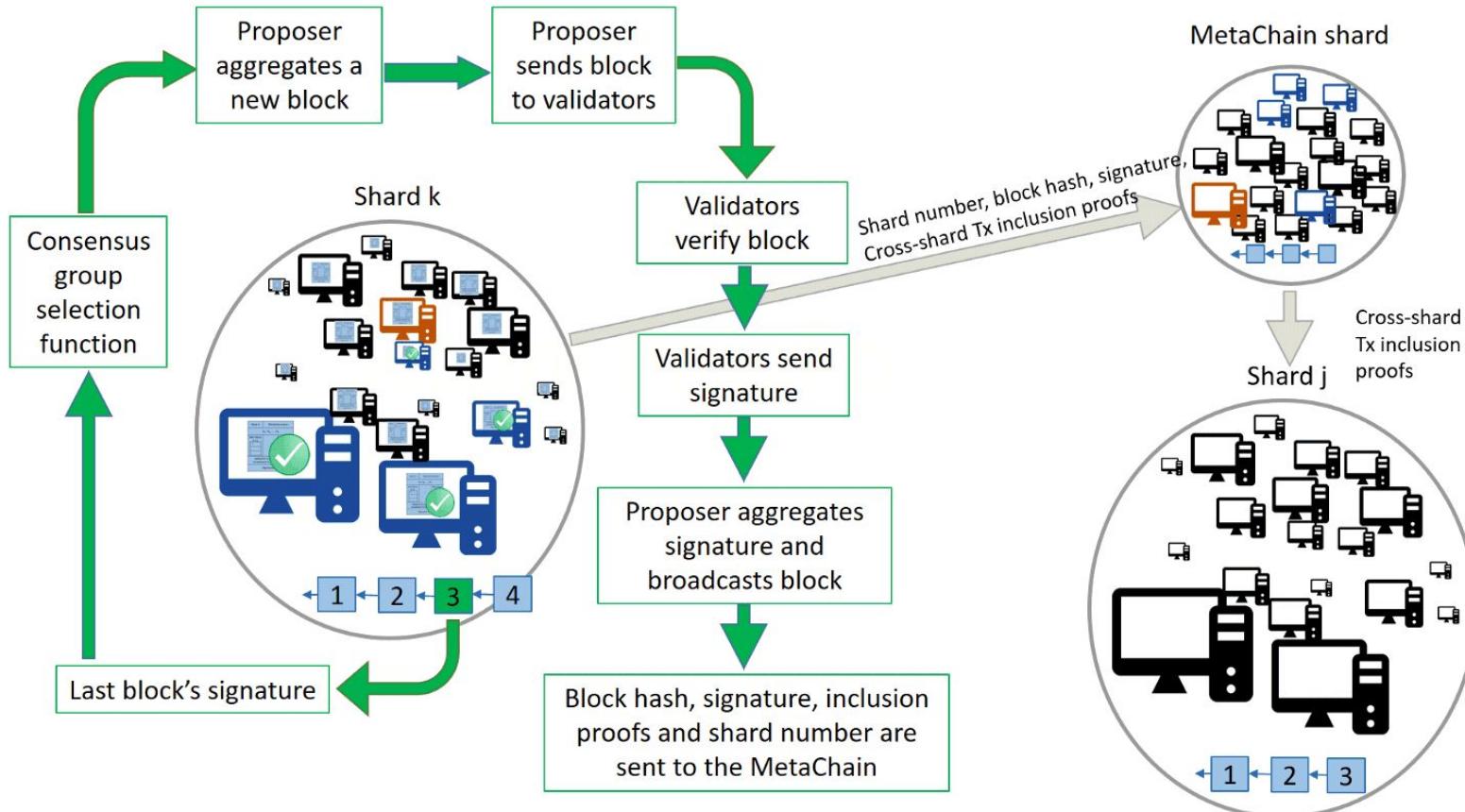
Hacker would need to own 51% of all the cryptocurrency on the network, which is practically impossible and therefore, makes 51% attacks impossible



Requires less energy

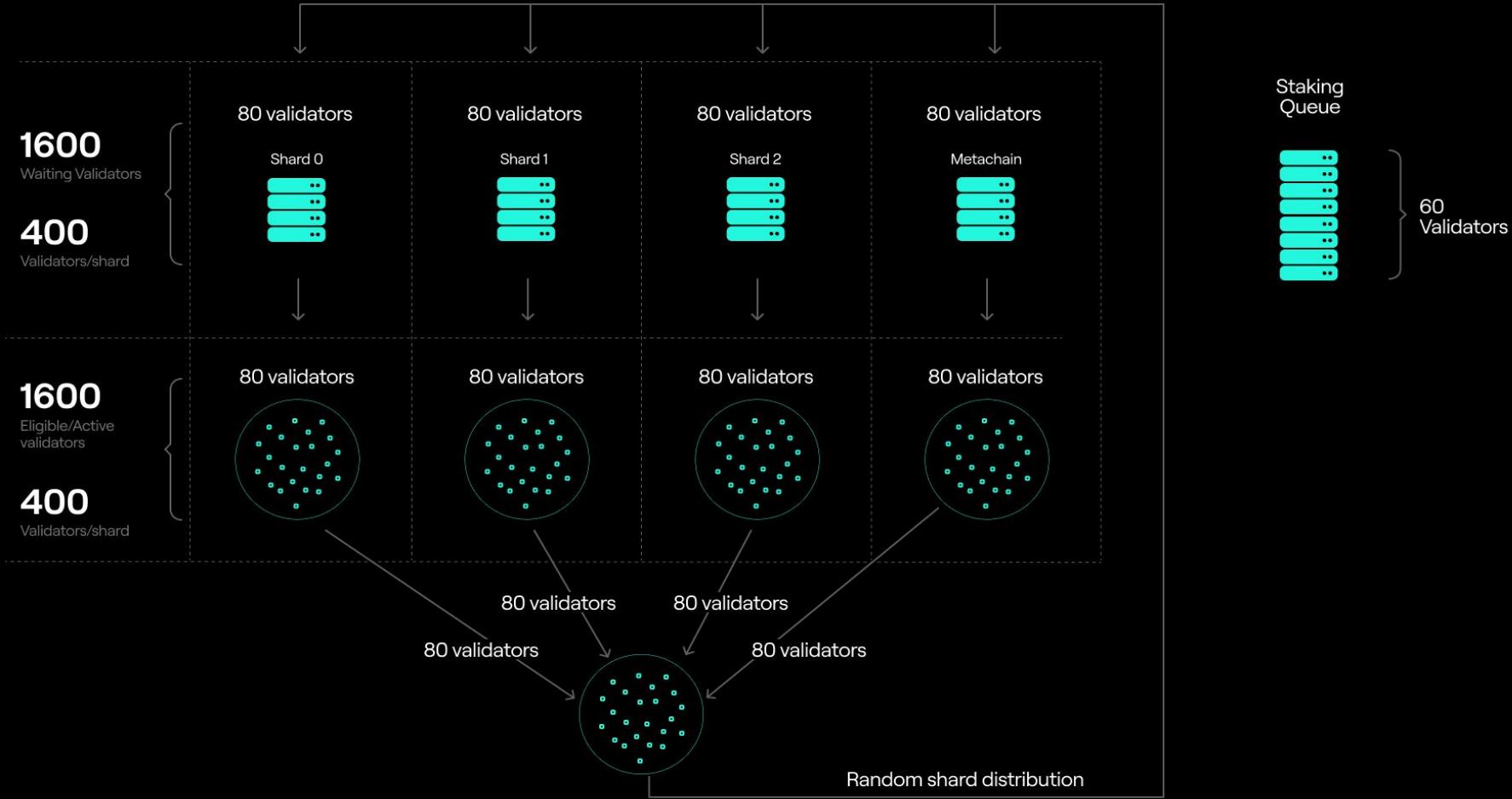
# Secure Proof of Stake

Secure Proof of Stake



# Network

- Permissionless
- 5000+ nodes (3200+ validator nodes)
- 30,000 current TPS
- 6s block time
- consensus: 63 validators out of 400
- $\frac{2}{3} + 1$  for successful block
- \$0.002 tx cost
- 30% SC gas returned to SC authors



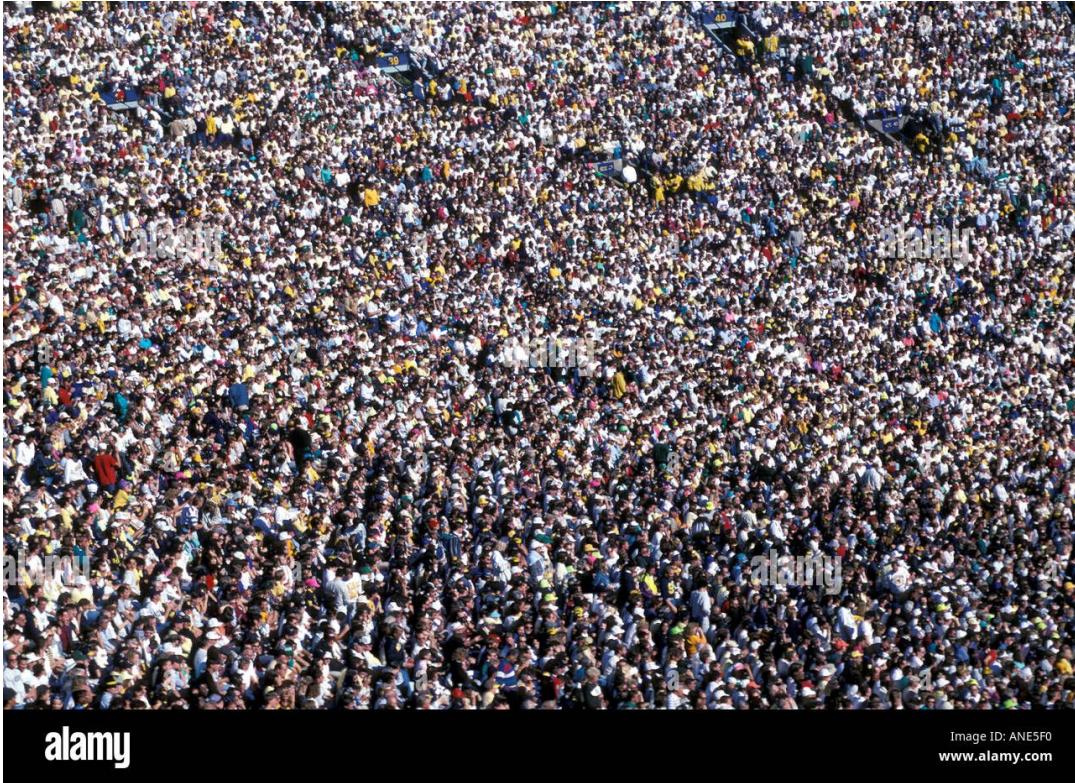
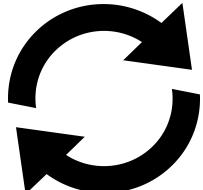
# Auction

# Waiting

# Eligible



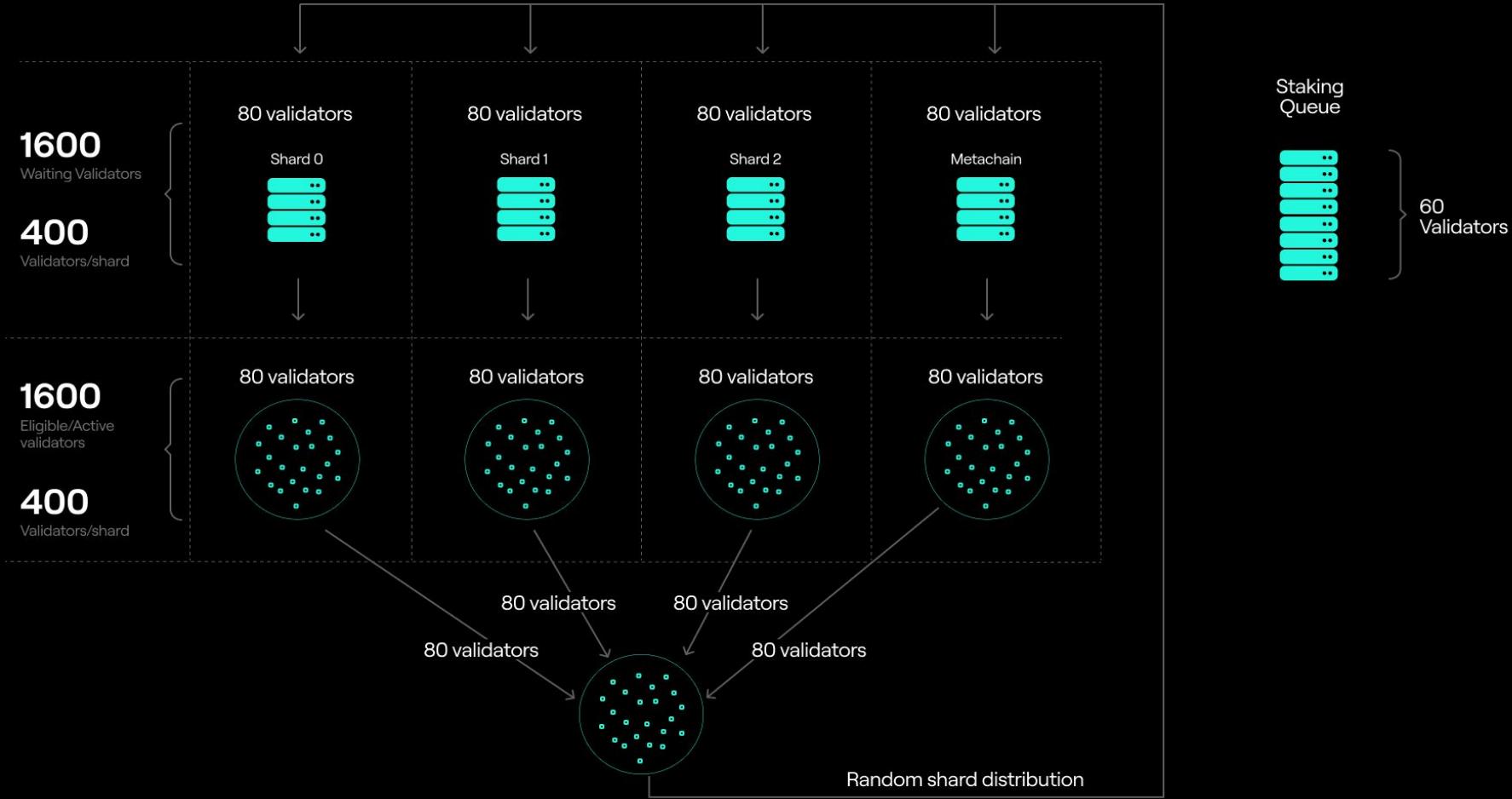
# Why sharding?



# Adaptive State Sharding

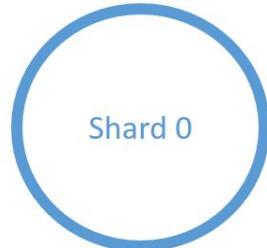
1. Network sharding
2. Transaction sharding
3. State sharding

- 👉 allows parallel processing
- 👉 nodes process only a part of all TXs



Step 1:  
Node to shard assignment

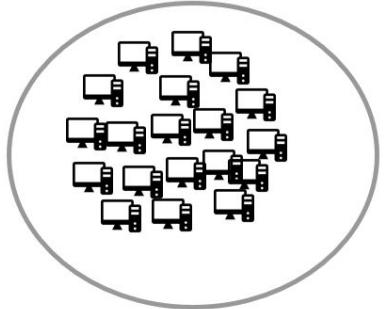
1 Shard



3bit Addresses	Shard
0	0
0	0
0	0
0	0
1	0
1	0
1	0
1	0

400 < Total Nodes < 800

Node Pool



# Smart Contracts



contracts > examples > crypto-kitties > kitty-auction > src > lib.rs > KittyAuction > create\_and\_auction\_gen\_zero\_kitty

```
1  #![no_std]
2
3  use multiversx_sc::imports::*;
4
5  pub mod auction;
6  use auction::*;

7  pub mod kitty_owner
8
9  #[multiversx_sc::co
2 implementations
10 pub trait KittyAuct
11     #[init]
12     fn init(
13         &self,
14         gen_zero_ki
15         gen_zero_ki
16         gen_zero_ki
17         opt_kitty_o
18     ) {
19         self.gen_ze
20             .set(ge
21         self.gen_ze
22             .set(ge
23         self.gen_ze
24             .set(ge
25
26         if let Opti
27             self.ki
28     }
29
30
31 // endpoints - owner-only
32
```



tract\_address {

# VM overview

Smart contract (Rust/C/C++/...)

Wasmer

SC primitives

Execution contexts

mx-chain-wasm-vm (VM)

mx-chain-vm-common (VM adapter)

mx-chain-go node (the protocol)