=nil; Foundation

Overcoming Security Risks in zkSharding

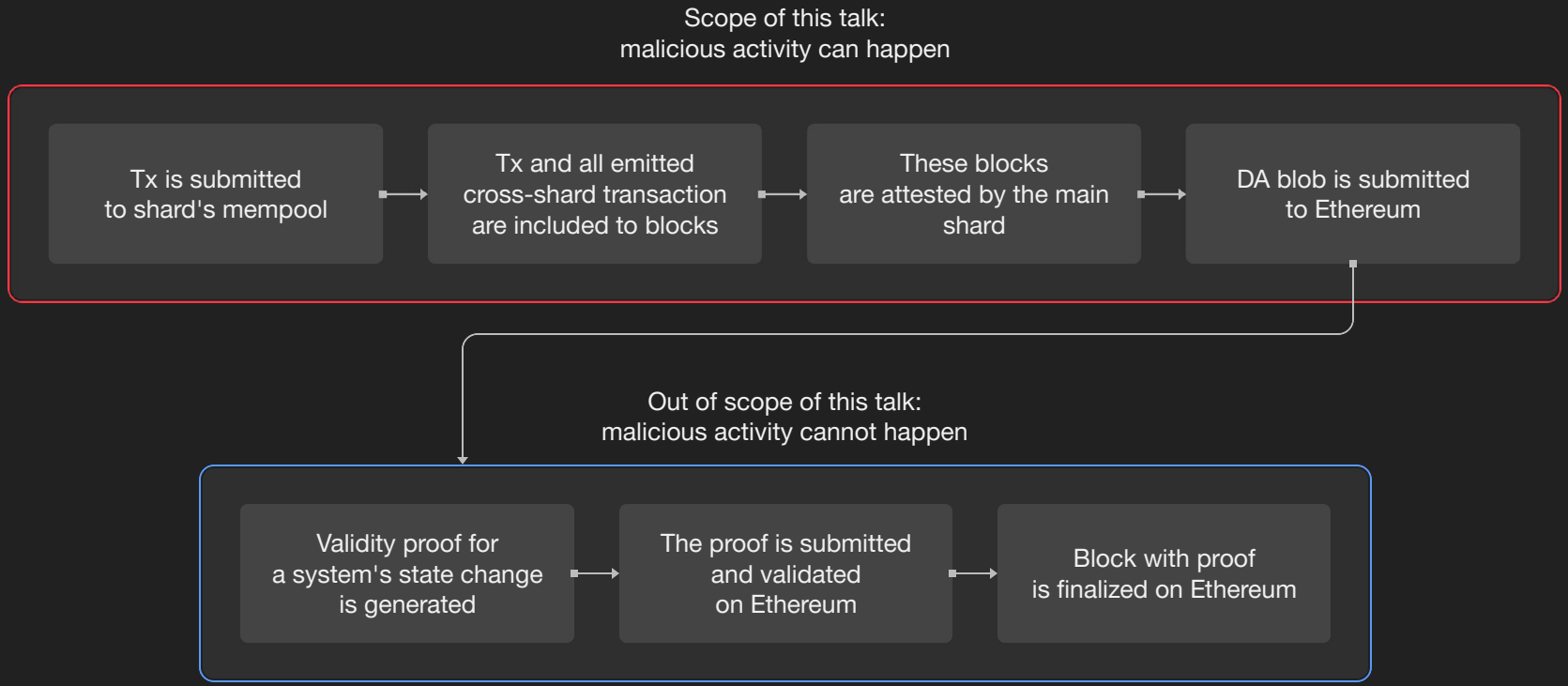
Vitaly Kuznetsov

What we'll cover

- Outline of the problem and the scope
- Possible approaches
- Proposed solution
- Further directions

Scope

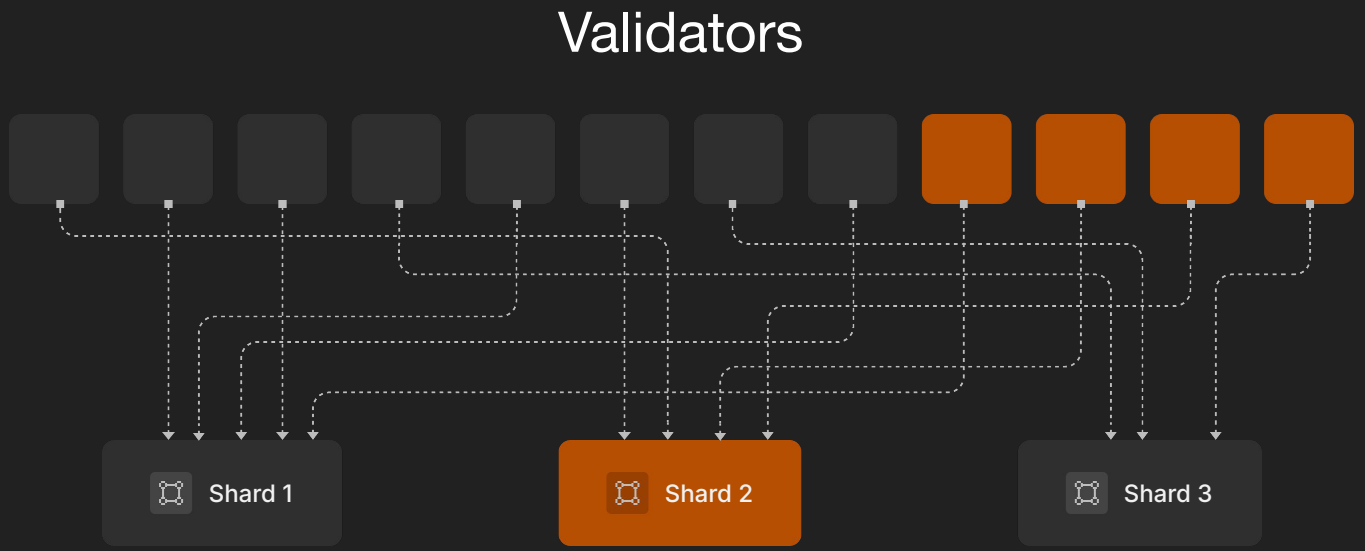
Time window before the validity proof is generated



Probabilistic Security

Understanding risks before L1 settlement

- Randomness Origins
- Consensus Safety Guarantees
- 1% Attack



The Need for Detection and Correction

- Mechanism to ensure integrity across shards
- Scope: State correction
- Out of Scope: Detection

Approach 1 – “Ignoring” the Problem

Configuring parameters to make risks negligible

- X – number of malicious validators on a shard
- n – number of validators on a shard
- f – fraction of malicious validators that could corrupt a shard
- N – number of validators
- t – number of malicious validators, up to N/3

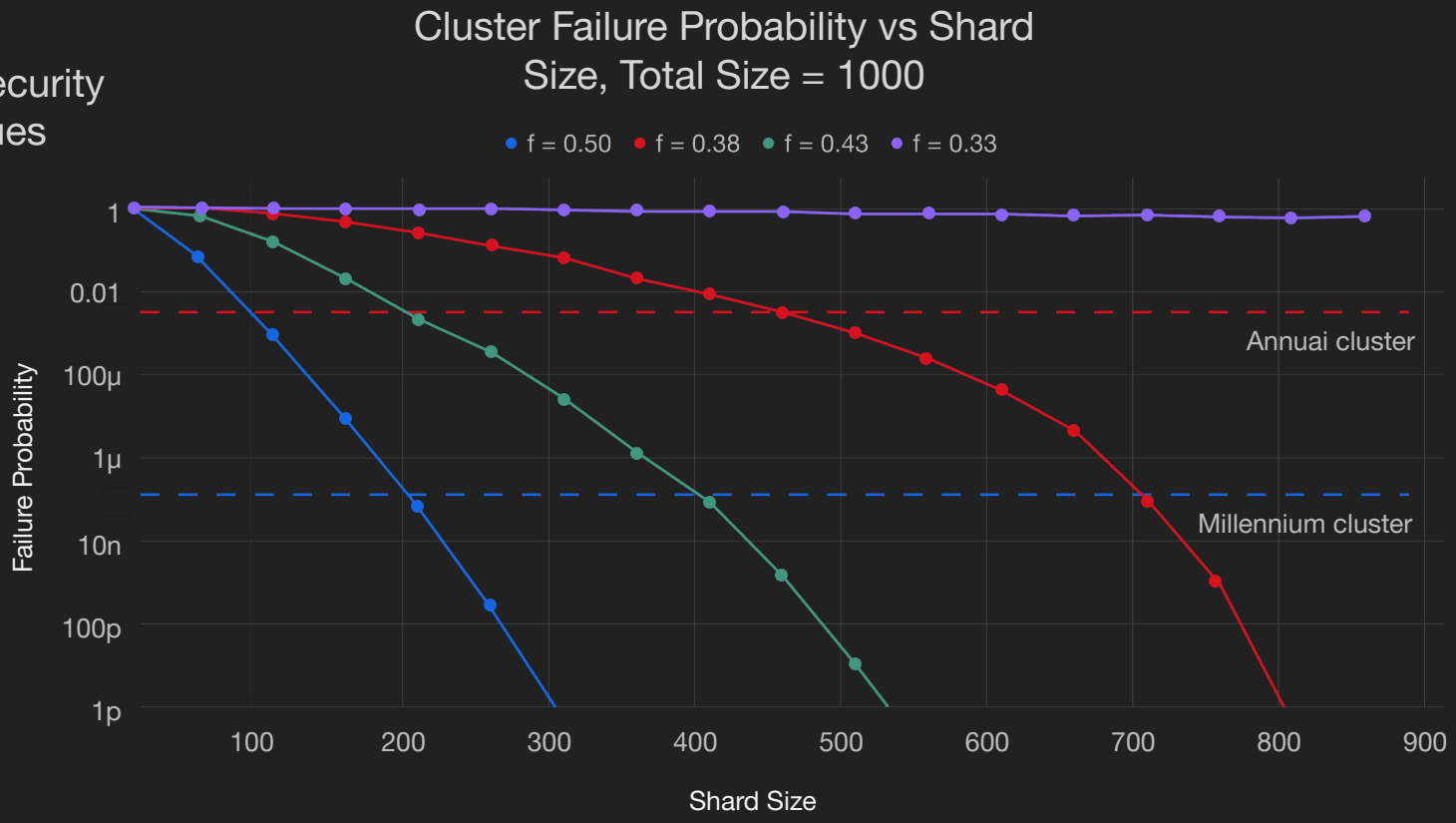
$$p_{\texttt{local\_fail}} \coloneqq \mathbb{P}\left(X \geq \lceil n \cdot f \rceil\right) = \sum_{x=\lceil n \cdot f \rceil}^{n} \frac{\binom{t}{x}\binom{N-t}{n-x}}{\binom{N}{n}}$$

$$p_{\texttt{fail}} = 1 - \left(1 - p_{\texttt{local\_fail}}\right)^{\lfloor \frac{N}{n} \rfloor}$$

Approach 1 – “Ignoring” the Problem

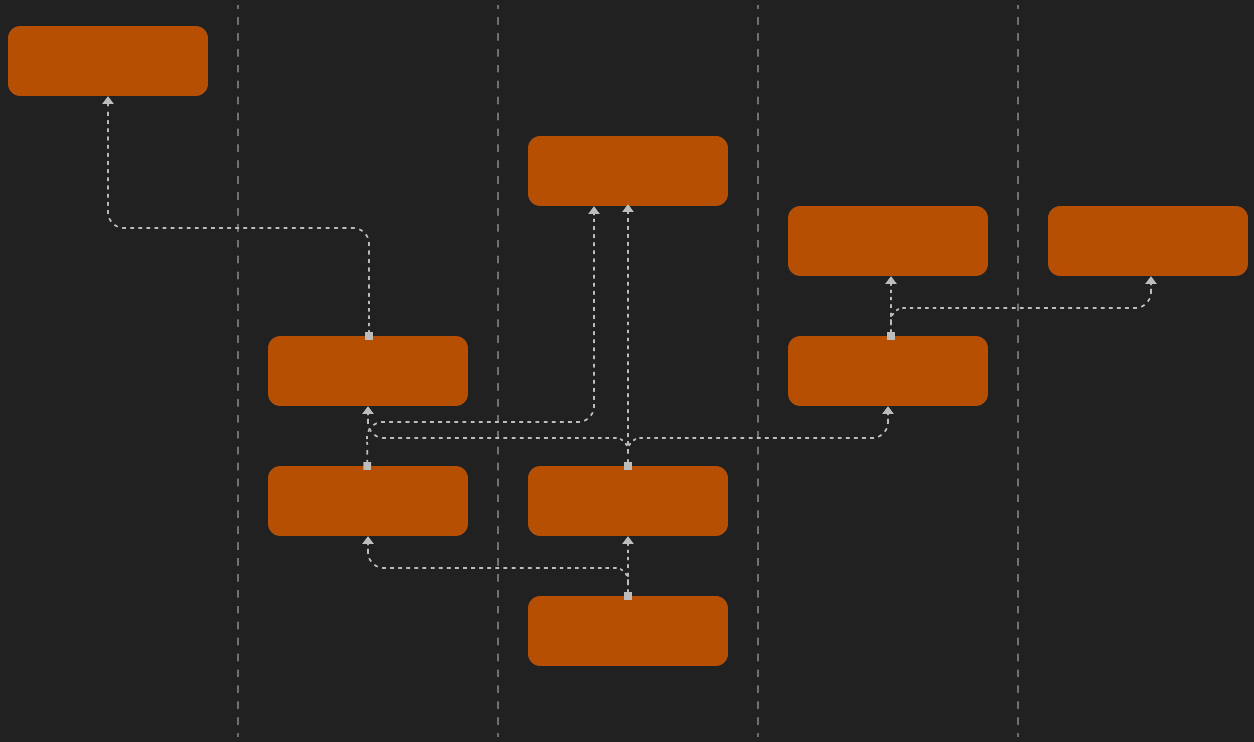
Limitations

- Undesirable security
- Scalability issues



Approach 2 – Partial Fixes

Tracing error propagation in the system

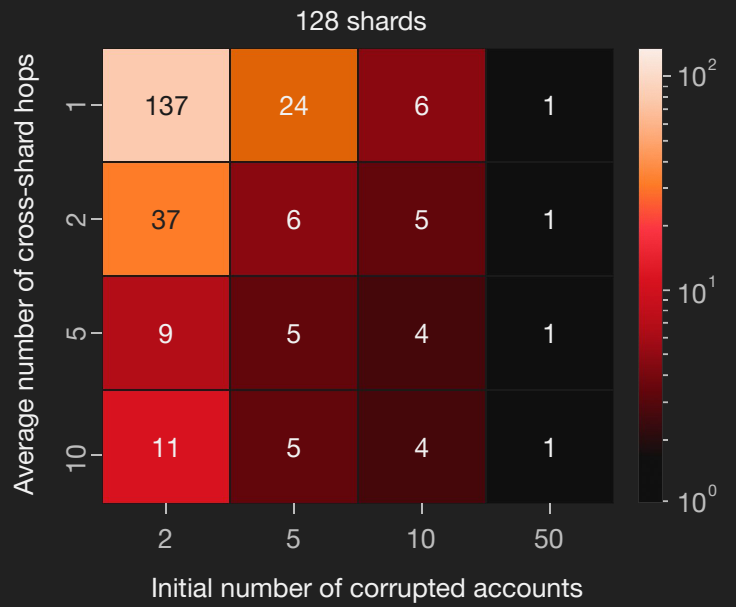


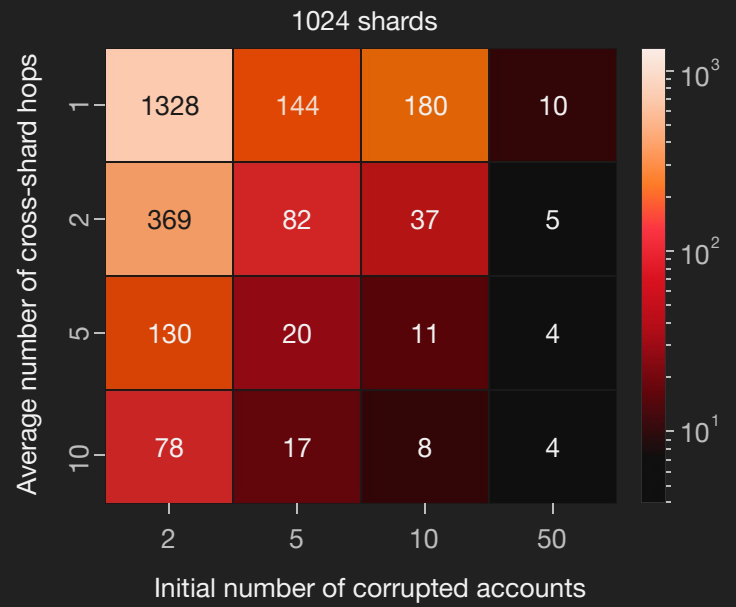
Approach 2 – Partial Fixes

Limitations

- High complexity
- Fragility
- Still poor UX

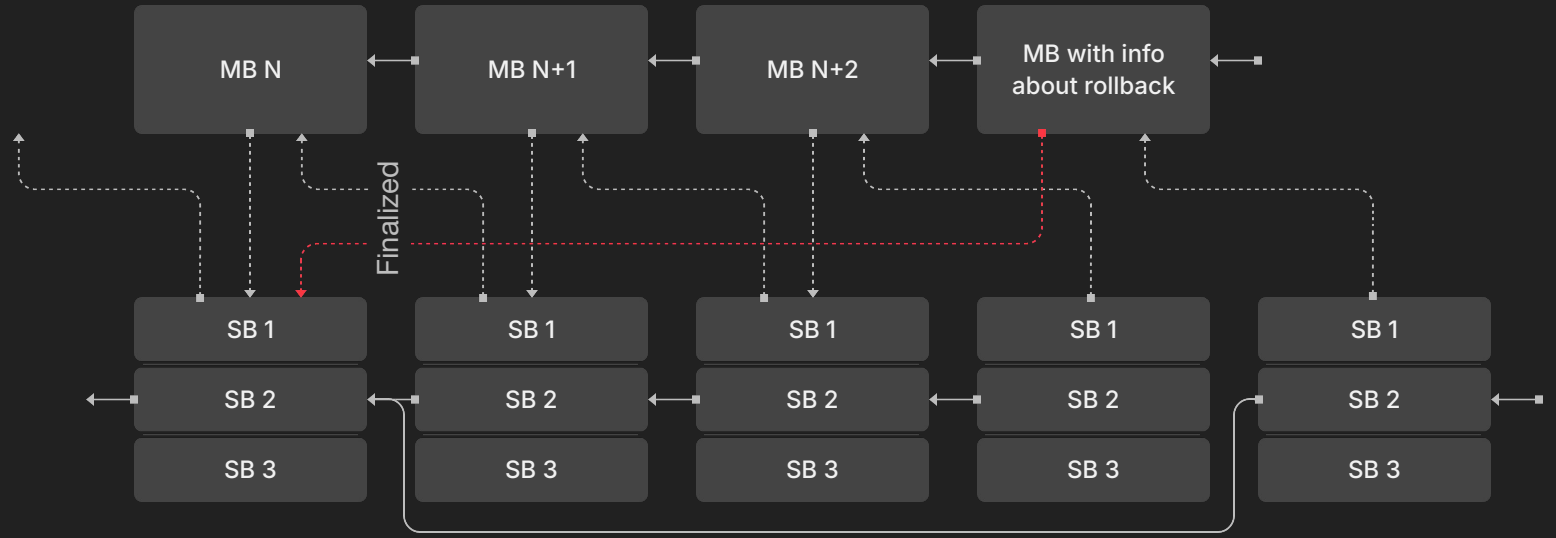
Number of steps (seconds) to corrupt half of the shards





Proposal

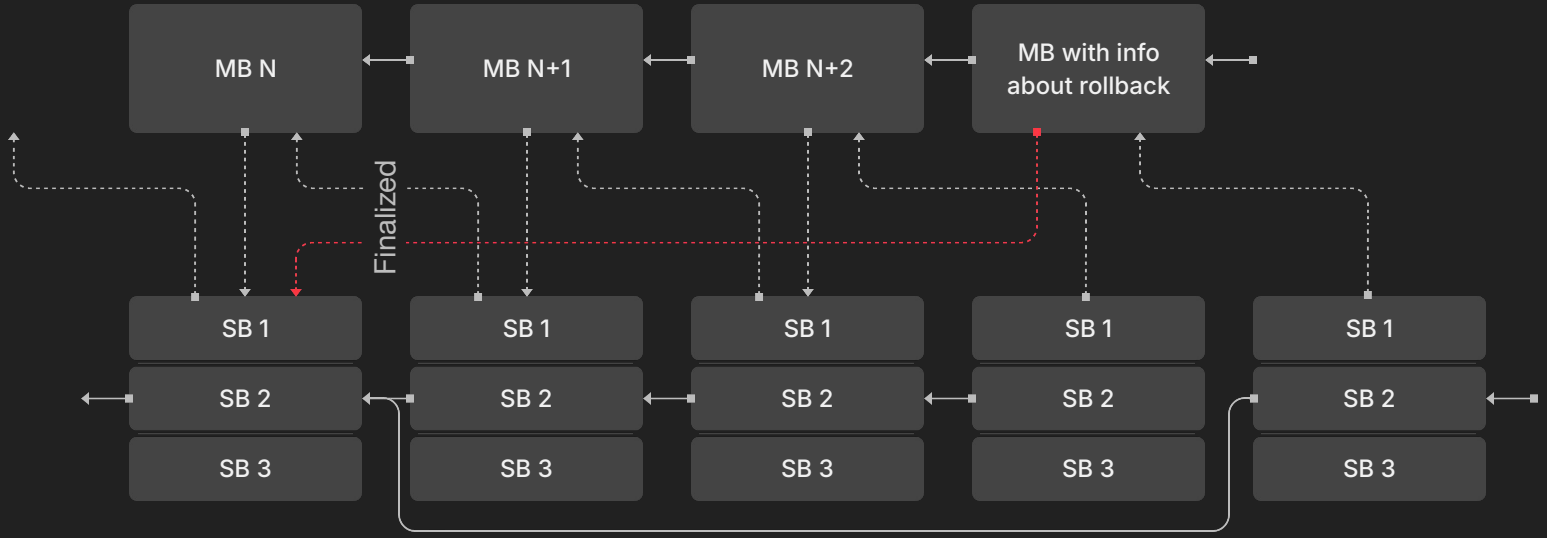
A simple approach



Proposal

Detailed look

1. Initiation step: Check the fraud proof
2. Get the info about the last finalized point
3. Set the main chain state root to the last finalized state root
4. Reference finalized shard blocks
5. Slash malicious actors — nodes who signed the fraudulent block
6. Update consensus parameters and reassign validators



Proposal

Benefits

- Smaller committee size
- Straightforward and robust approach

Further Directions

Ensuring compatibility with other parts of zkSharding

1. L1-L2 communication and Sync Committee
2. ShardDAG
3. Detection mechanism
4. Fraud proving

Your questions

=nil; Foundation

Thank you!