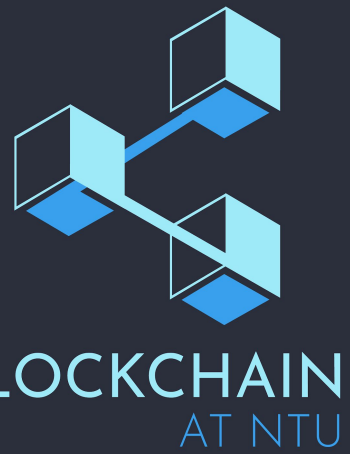
Gormos:

Sharded Plasma for scalable DEX

Alex Xiong



Agenda

- ❑ Preliminary
- ❑ Problem Statement
- ❑ Challenges → Gormos: Layer 2 Sharding → Open Problem
  - ↳ ACK: Joint work with Dr. Loi Luu
- ❑ **Design Space → Design Patterns/Lessons**

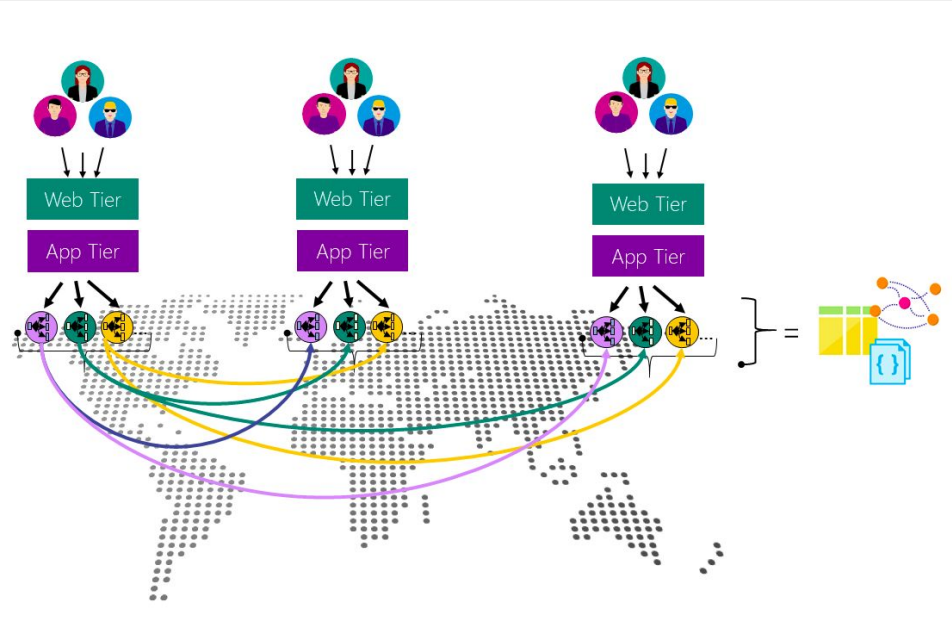
Preliminary

Blockchain & Smart Contract

Stateful distributed database,
with standardized execution engine,
without central control.

Blockchain & Smart Contract

Stateful distributed database,
with standardized execution engine,
without central control.



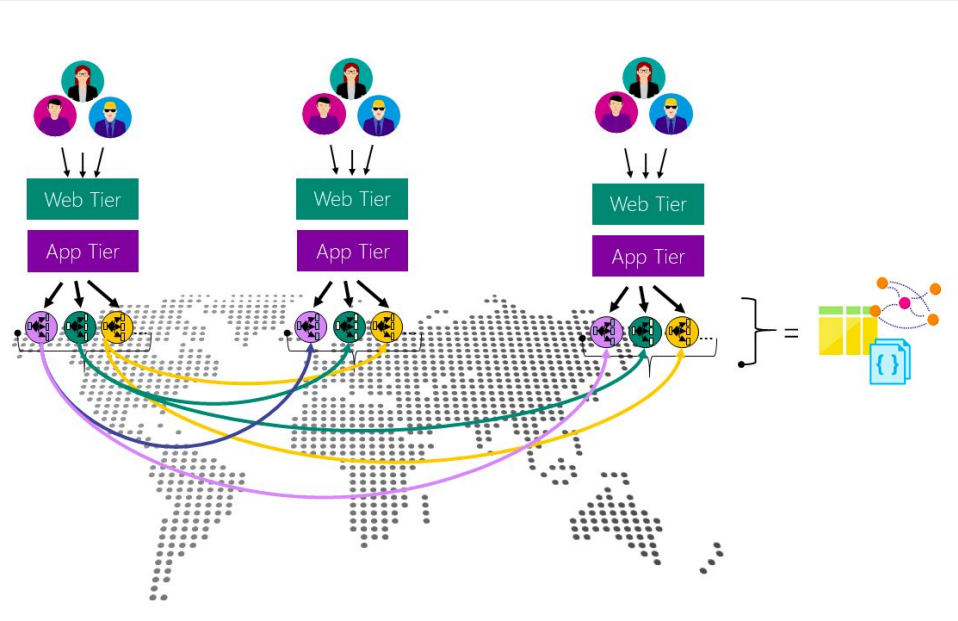
BLOCKCHAIN AT NTU

Blockchain & Smart Contract

Stateful distributed database,

with standardized execution engine,

without central control.



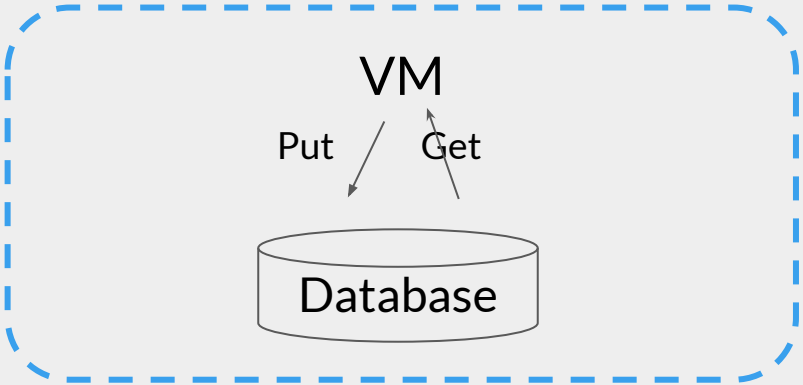
BLOCKCHAIN AT NTU

Blockchain & Smart Contract

Stateful distributed database,

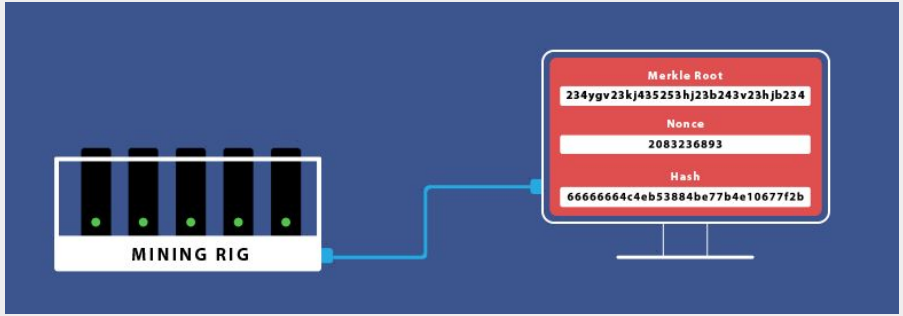
with standardized execution engine,

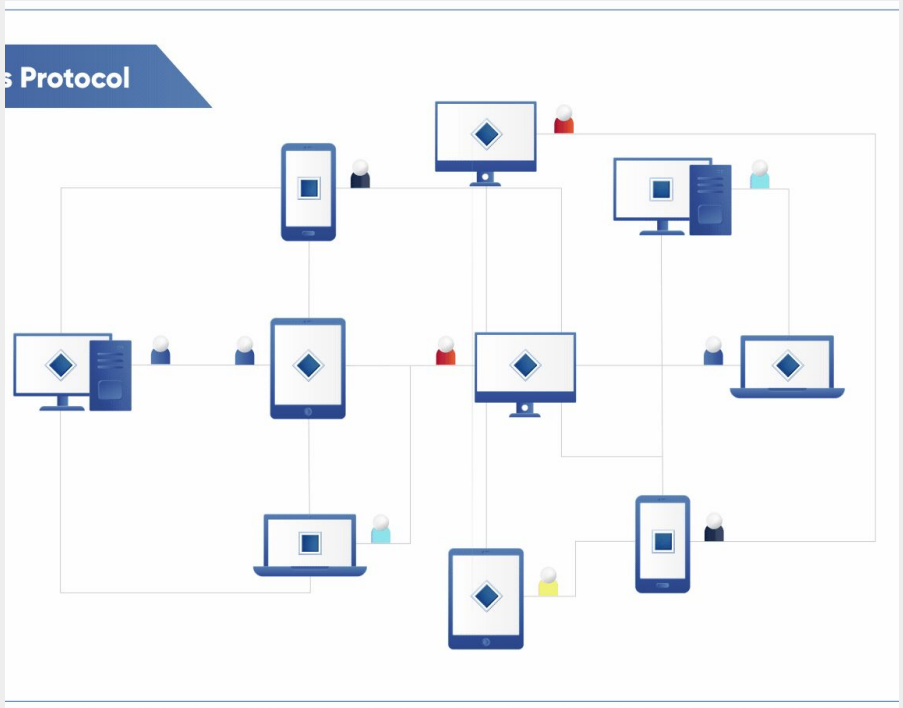
without central control.



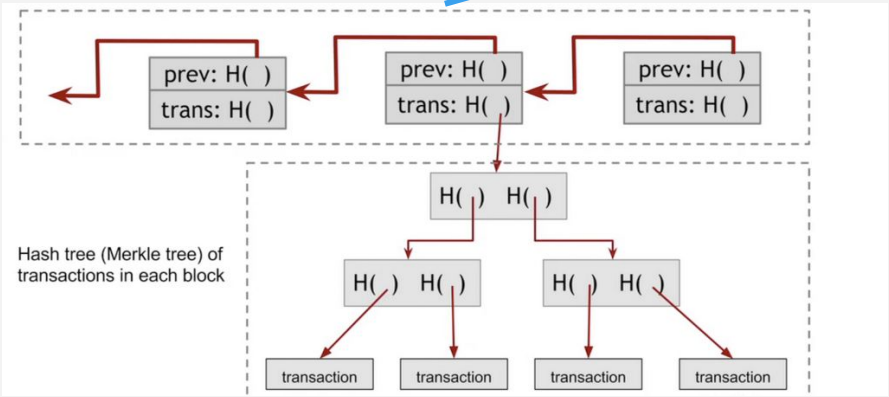
Blockchain & Smart Contract

Stateful distributed database,
with standardized execution engine,
without central control.

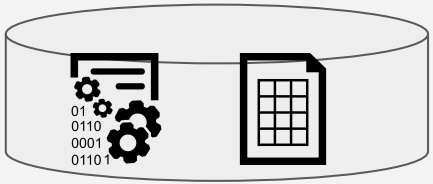




Blockchain



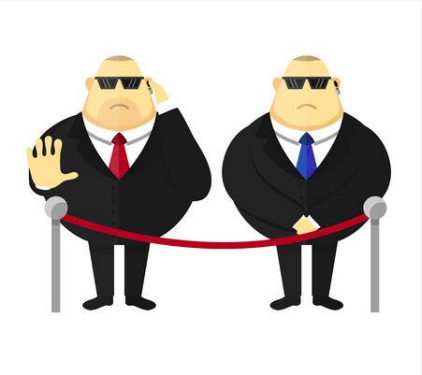
Hash tree (Merkle tree) of transactions in each block



BLOCKCHAIN AT NTU

Blockchain :: Permission(less/ed)

- Permission required to join & participate
- Potential problems of public blockchain:
  - Sybil attack
  - Communication overhead
  - Consensus delay
  - Byzantine failure



Public Blockchain :: Properties

- Secure & fault tolerant
  - Under “honest majority” assumption
  - Through replication & consensus
- Data availability & (somewhat) censorship resistant
  - Through replication & decentralized control
- Immutability w.h.p & enforced honest computing
  - Through authenticated, append-only data structure and public smart contract

Problems

Fact Check ...

VISA Peak	56,000 tps only 1/3 is used	VISA Average	4,000 tps	Ethereum	10 tps	Bitcoin	7 tps

BLOCKCHAIN AT NTU

Blockchain :: inherent limitation

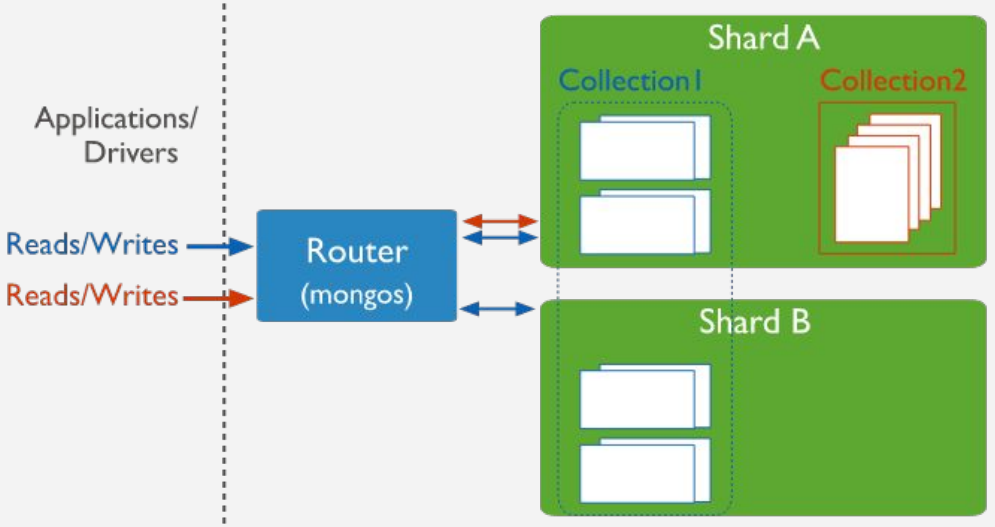
- Each node: **process all** TXs, **store all** TXs.
  - Network throughput = O (capacity of a single node)
  - Does not scale with more computation
- Throughput (tps) = tx per block / block interval
  - ↑ ↓
  - Centralization risk, bandwidth churn, more forks.

Key Idea:

Scale network throughput by
partitioning data processing & storage
& **parallelizing** their execution
within a **smaller committee**.

Sharding :: a good old friend

- ❑ Huge data volume
- ❑ Frequent data TX
- ❑ Higher throughput & faster response

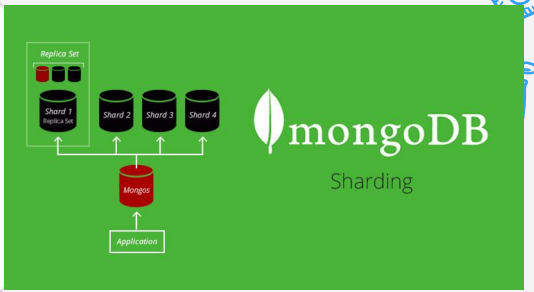


Challenges

Traditional Sharding

- Trusted Infrastructure
  - Master-slave, coordinated sharding
  - Non-byzantine behavior

Google Cloud Spanner



ORACLE DATABASE

BLOCKCHAIN AT NTU

Non-trivial for public blockchain

- Stronger adversarial assumption
  - ↳ Single shard takeover?
- Generic sharding
  - ↳ What's the shard key?
- Data availability
  - ↳ Shard fraud inspection/detection
  - ↳ Tradeoff: replication cost v.s. data availability
- Expensive cross-shard communication
  - ↳ Asynchronous, non-global → race condition, atomicity

Global enforcement on non-global data

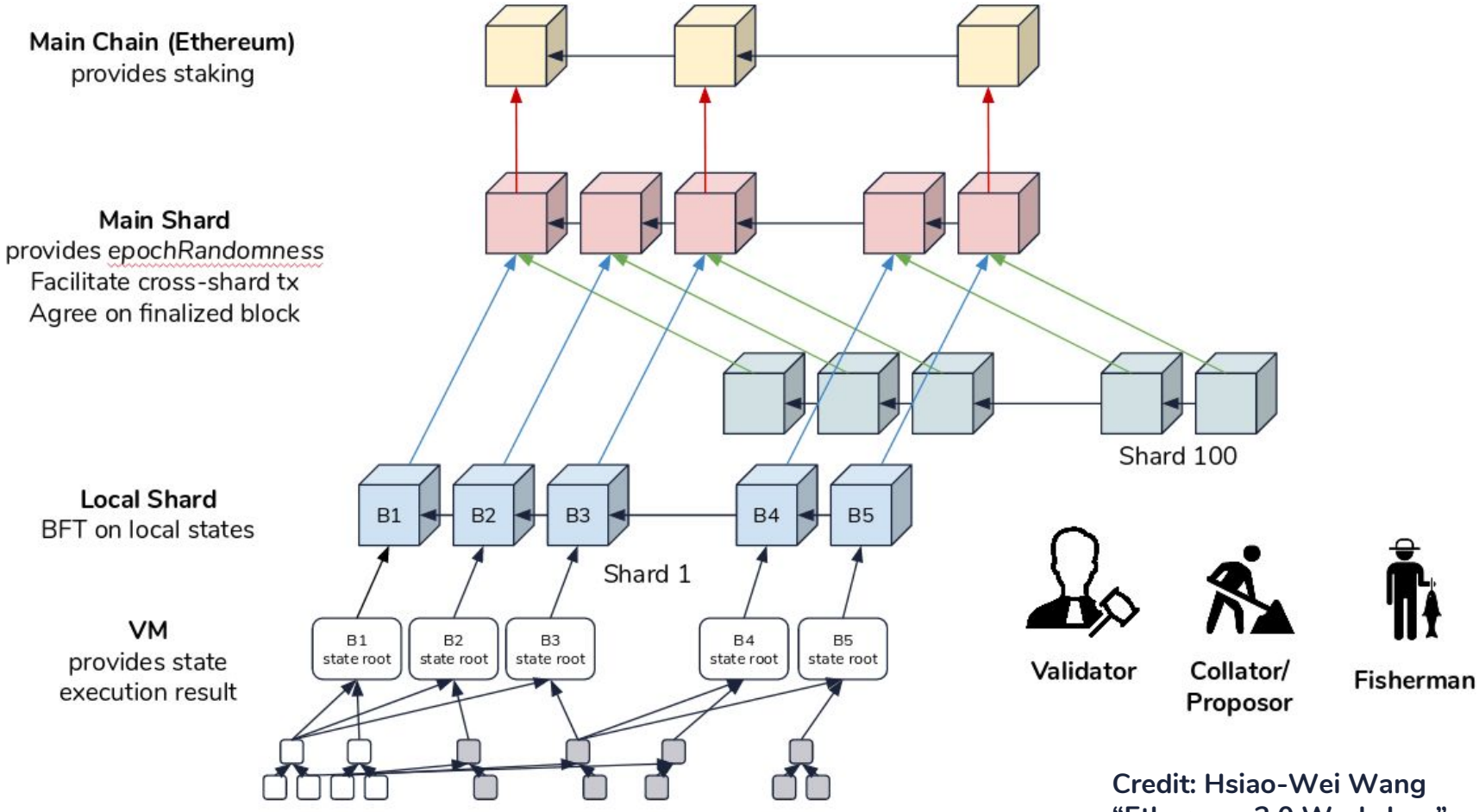
-- what's so hard about Blockchain Layer 2 approaches

Gormos

Key Observation & Insight

Sharded Plasma chain (layer 2 / off-chain)

1. with cryptononmically bounded validators **randomly assigned** to each shard to notarize new block.
2. **Shard key is token pair** (e.g. {ETH, OMG}) to minimize cross-shard TX.
3. **Fisherman** guarantee data availability through probabilistic challenge.



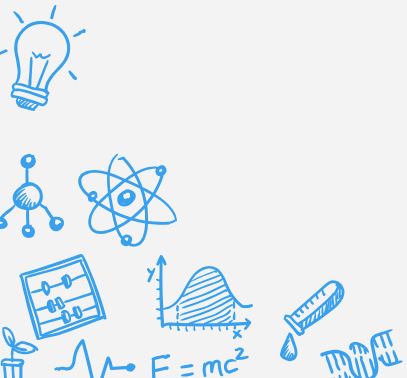
Credit: Hsiao-Wei Wang
"Ethereum 2.0 Workshop"

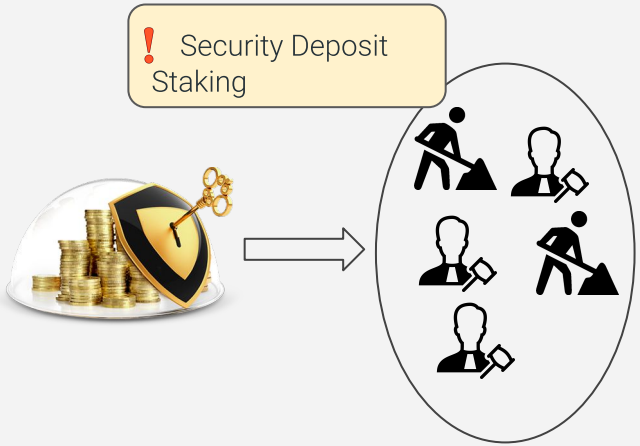
Gormos :: 5 steps

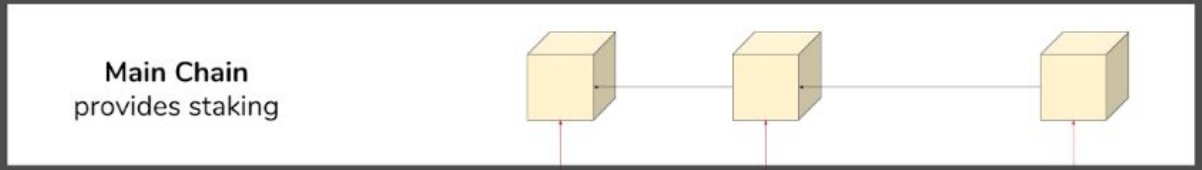
1. Validator pool formation
2. Random assignment of validators to shards
3. Intra-committee BFT consensus
4. Final-committee /main shard consensus and root-chain commit
5. Generate randomness for next epoch

Gormos :: 5 steps

1. Validator pool formation
2. Random assignment of validators to shards
3. Intra-committee BFT consensus
4. Final-committee /main shard consensus and root-chain commit
5. Generate randomness for next epoch

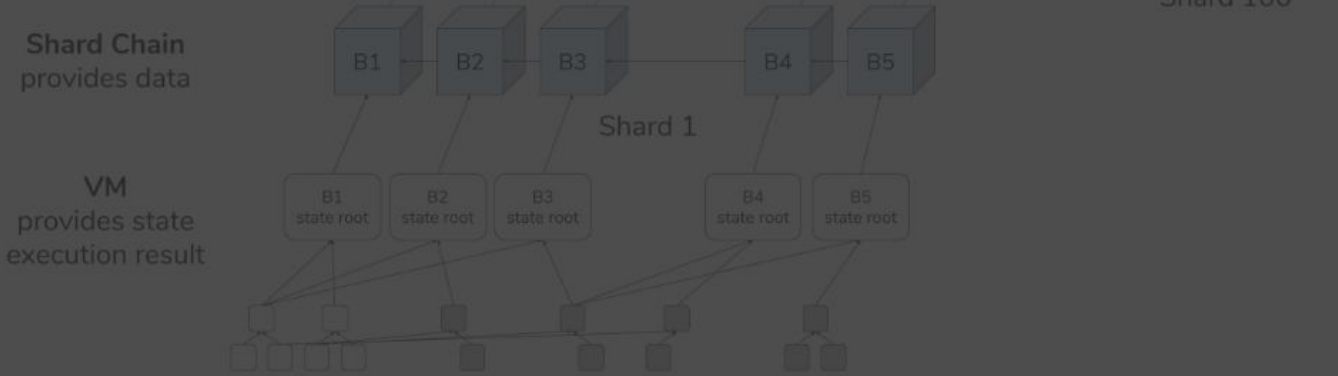



Security Deposit
Staking



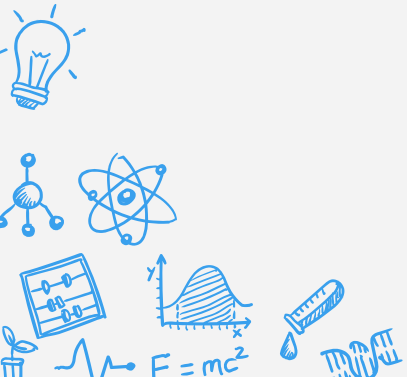
Staking - Sharding Manager Contract (SMC)

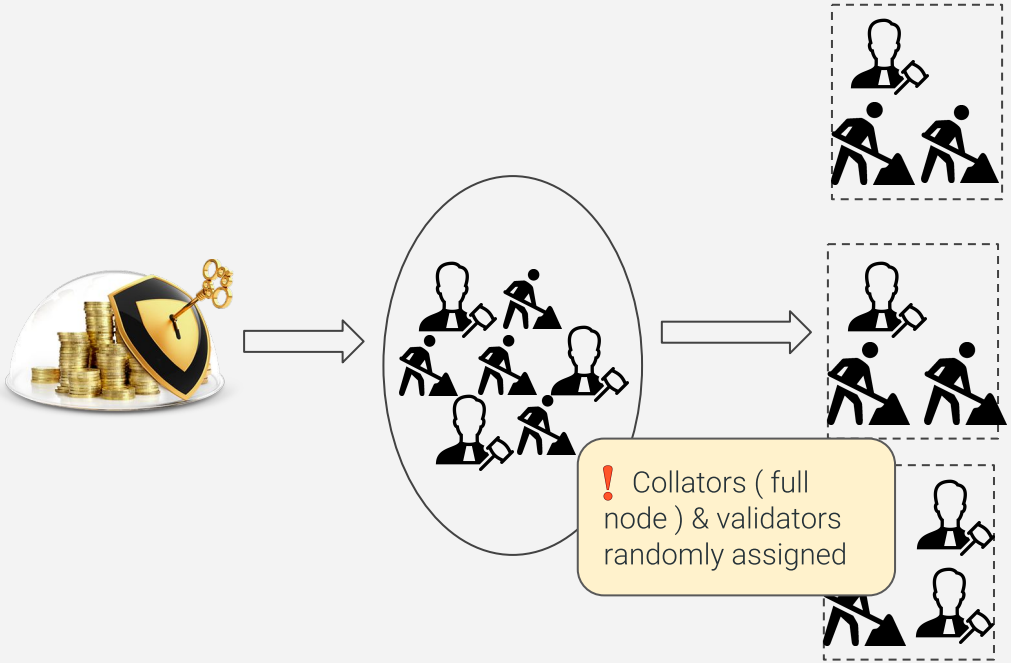
- ⟠ Security deposit
- ⟠ Validator & Collator pool

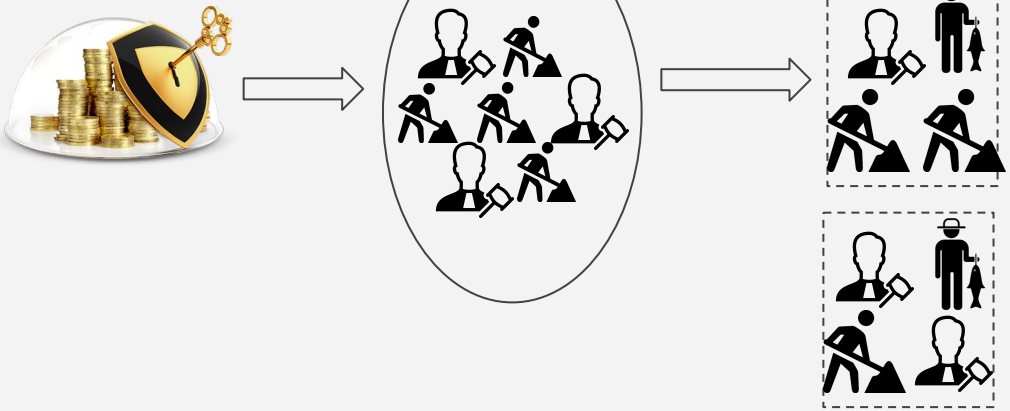


Gormos :: 5 steps

1. Validator pool formation
2. Random assignment of validators to shards
3. Intra-committee BFT consensus
4. Final-committee /main shard consensus and root-chain commit
5. Generate randomness for next epoch



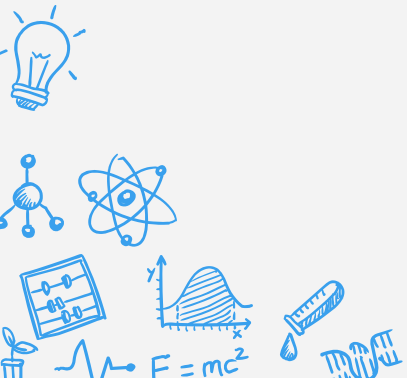

❗ Collators (full node) & validators randomly assigned

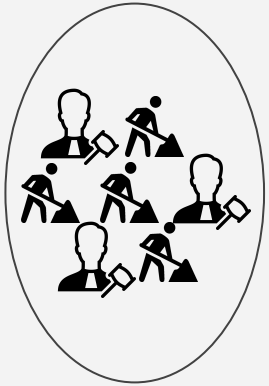


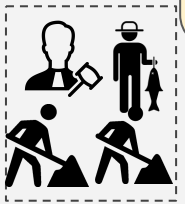
❗ Fisherman (small bond) challenge collator, ensure data availability

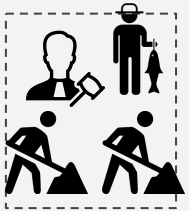
Gormos :: 5 steps

1. Validator pool formation
2. Random assignment of validators to shards
3. Intra-committee BFT consensus
4. Final-committee /main shard consensus and root-chain commit
5. Generate randomness for next epoch







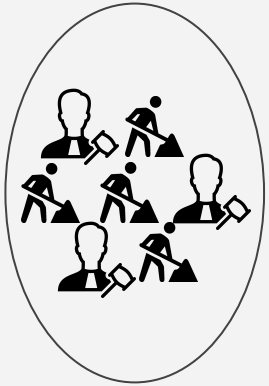


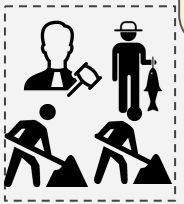
! Collator propose, validator rectify and sign using BFT

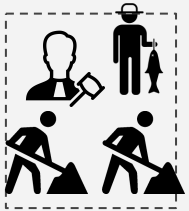
Block H1

Block H2

Block H3







❗ Collator propose, validator rectify and sign using BFT

Block H1

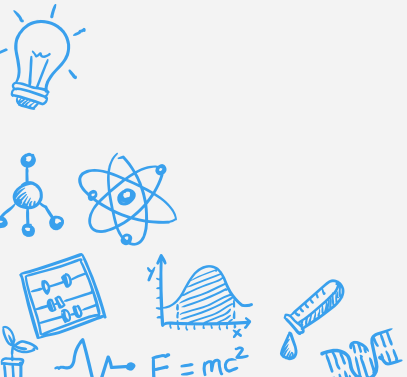
❗ Use aggregated signatures

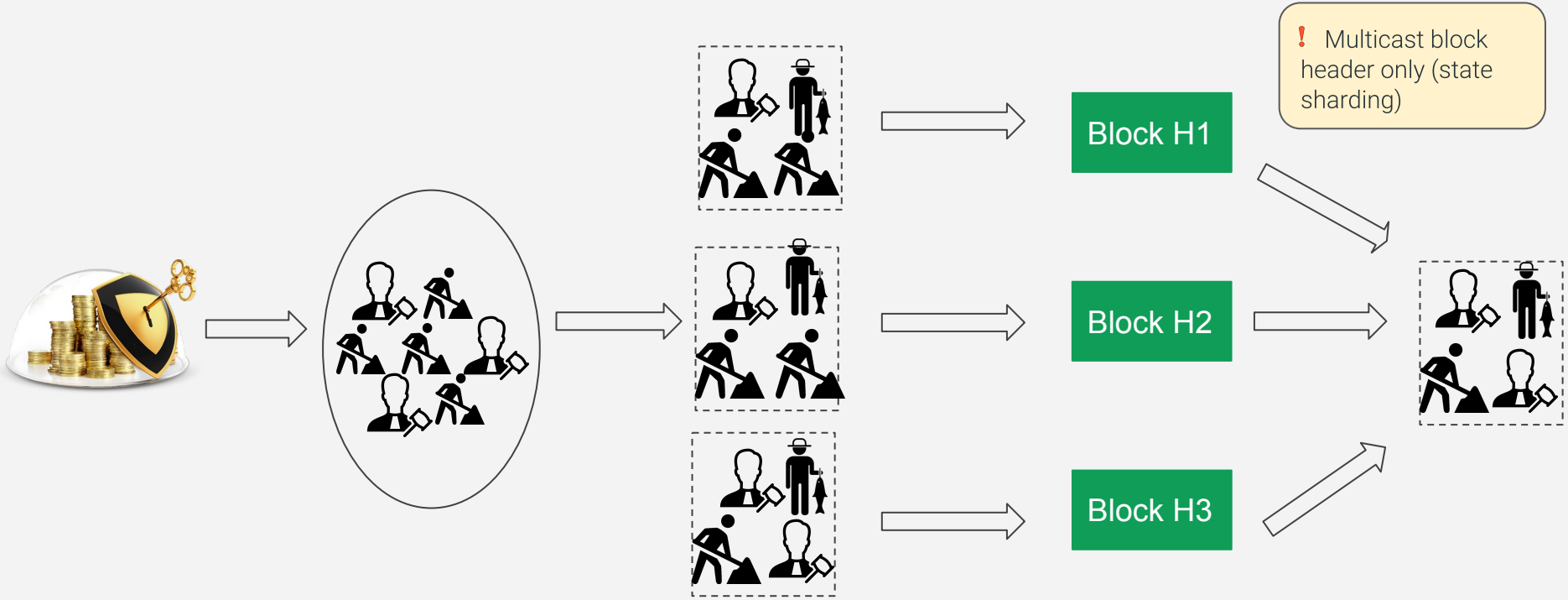
Block H2

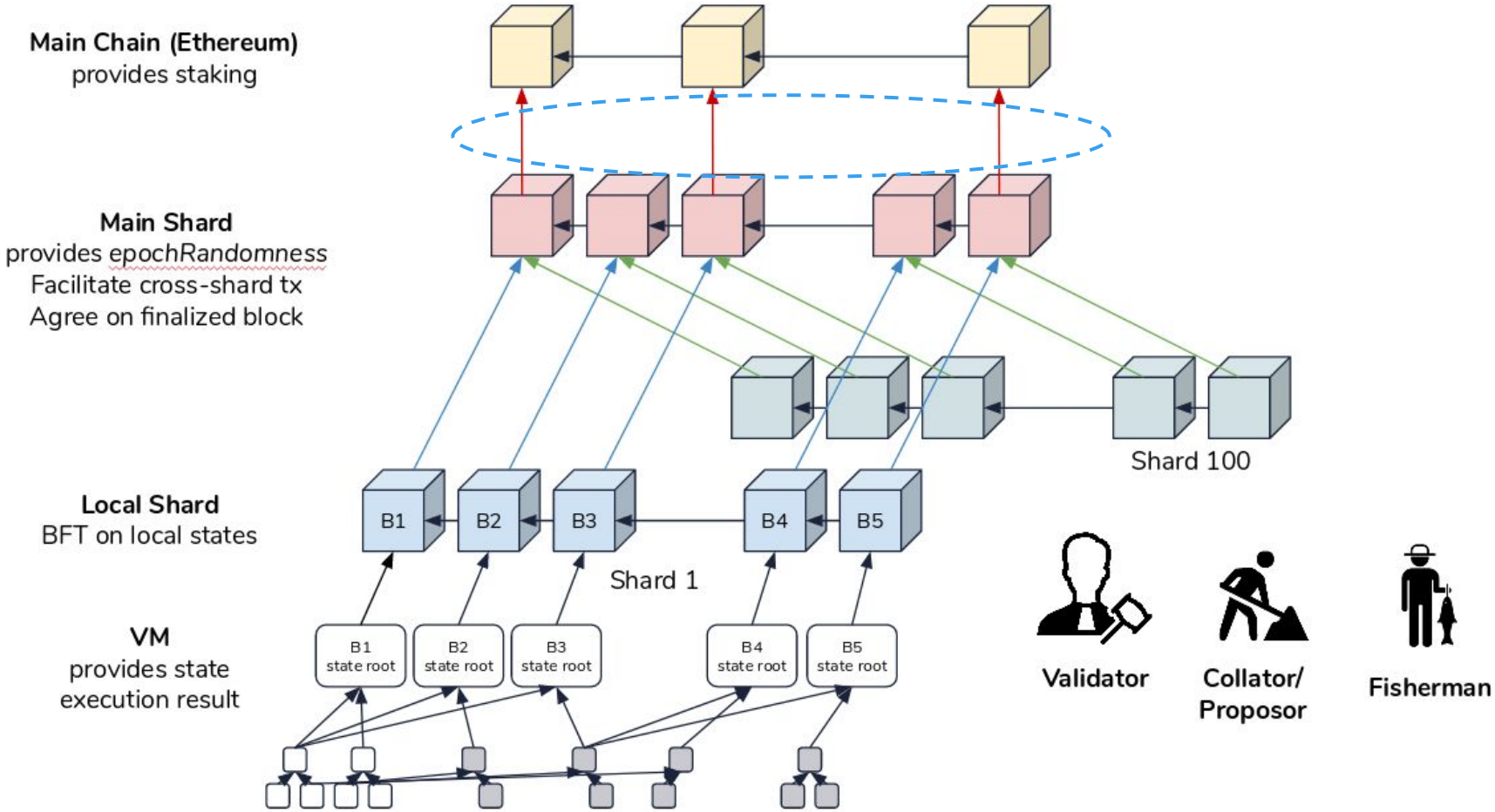
Block H3

Gormos :: 5 steps

1. Validator pool formation
2. Random assignment of validators to shards
3. Intra-committee BFT consensus
4. Final-committee /main shard consensus and root-chain commit
5. Generate randomness for next epoch

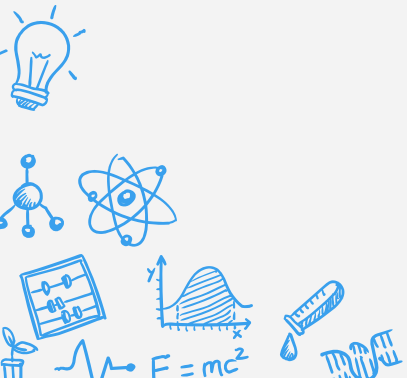



Block H1
Block H2
Block H3
! Multicast block header only (state sharding)


Main Chain (Ethereum)
provides staking
Main Shard
provides epochRandomness
Facilitate cross-shard tx
Agree on finalized block
Local Shard
BFT on local states
VM
provides state
execution result
B1
B2
B3
B4
B5
Shard 1
Shard 100
B1 state root
B2 state root
B3 state root
B4 state root
B5 state root
Validator
Collator/
Proposor
Fisherman

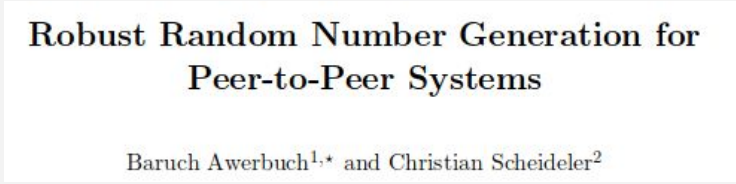
Gormos :: 5 steps

1. Validator pool formation
2. Random assignment of validators to shards
3. Intra-committee BFT consensus
4. Final-committee /main shard consensus and root-chain commit
5. Generate randomness for next epoch



Epoch Randomness

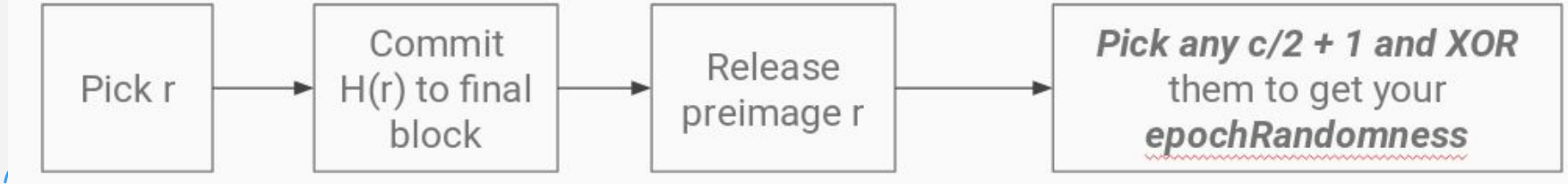
- Unpredictable by adversaries
  - ↳ Avoid targeted attack & single shard takeover
- RNG is hard
  - ↳ Up to ⅙ faulty nodes
  - ↳ Excessive message complexity


Robust Random Number Generation for Peer-to-Peer Systems

Baruch Awerbuch[1,⋆] and Christian Scheideler[2]

Epoch Randomness

- VRF [Micali, S. et al.]
- Or RANDAO



Open Problems

Open Problems

- ❑ Compatible incentive design for fisherman and collators
  - ↳ TX fee sharing scheme
- ❑ Formalization of error bound on data availability guarantee
  - ↳ Probabilistic probing/challenging
- ❑ Fast sync for newly assigned validators
  - ↳ Recursive SNARK ? Cryptonomic checkpointing?
- ❑ Efficient, secure cross-shard communication specs
- ❑ Front-running attack

Thank you!

🔨 with 💙 by

👷 Alex Xiong

💬 https://alexxiong.xyz
hello@alexxiong.xyz
@ALuoyuan