Sharding

and other things
About us

Bowen

- Core engineer at NEAR (built a lot of stuff we’re covering)

Peter

- Developer experience and developer tooling
- Working on sustainable behavior on the side
About NEAR Protocol

- Public Sharded Blockchain
- Emphasis on usability,
- (specifically Developer Usability)
- A bunch of ex-MemSQL and ex-Google
- A group of ACM ICPC gold medalists
- A grip of previous founders
What we’re talking about

1. Sharding, (in blockchains)
2. 10,000 ft view of crypto
3. Behavior (it relates to blockchain)
Sharding Overview
Sharding Overview
Sharding Overview

**Main Chain**
- a.k.a. Beacon Chain
- Relay Chain
- Hub

**Shard Chains**
- a.k.a. Parachains
- Zones
Corrupting Validators

X validators building one chain.
Need to corrupt 0.51x

X validators building 10 chains
Need to corrupt 0.051x
Sampling Validators
Sampling Validators

You ➔ Current Validators ➔ Previous Validators ➔ First Validators ➔ Genesis Block

- Trust Block
- Assume < 1/3 Malicious
- Ask Last Block
- Ask Last Block
- Ask Last Block
- Assume < 1/3 Malicious
- Assume < 1/3 Malicious
- Assume < 1/3 Malicious
- Assume < 1/3 Malicious
- Genesis Block
Malicious Behavior

Forking

Invalid State Transitions
## Invalid State Transition

**Transaction X**

<table>
<thead>
<tr>
<th>From:</th>
<th>Alice</th>
</tr>
</thead>
<tbody>
<tr>
<td>To:</td>
<td>Bob</td>
</tr>
<tr>
<td>Amt:</td>
<td>10</td>
</tr>
</tbody>
</table>

### Block A (Valid)

<table>
<thead>
<tr>
<th>State Before:</th>
<th>Alice: 10, Bob: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transactions:</td>
<td>X</td>
</tr>
<tr>
<td>State After:</td>
<td>Alice: 0, Bob: 10</td>
</tr>
</tbody>
</table>

### Block A’ (Invalid)

<table>
<thead>
<tr>
<th>State Before:</th>
<th>Alice: 10, Bob: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transactions:</td>
<td>X</td>
</tr>
<tr>
<td>State After:</td>
<td>Alice: 0, Bob: 1000</td>
</tr>
</tbody>
</table>
Malicious Behavior *without* Sharding

- [✓] Forking
- [✗] Invalid State Transitions
Malicious Behavior **without** Sharding
Malicious Behavior with Cross-Shard Transactions

✔ Forking

✔ Invalid State Transitions
Cross-shard Communication

Main Chain
a.k.a.
Beacon Chain
Relay Chain
Hub

Shard Chains
a.k.a.
Parachains
Zones
Cross-shard Communication

Main Chain
a.k.a.
Beacon Chain
Relay Chain
Hub

Shard Chains
a.k.a.
Parachains
Zones
Cross-shard Communication: Receipts

- Alice from Shard#1 sends money to Bob on Shard#2;
- A tx that debits Alice’s account is executed on Shard#1;
- A proof of execution (Receipt) is created and sent to Shard#2;
- A tx that credits Bob’s account is executed on Shard#2.
State Validity

Shard #1

Shard #2

CROSS-SHARD TRANSACTION
State Validity
State Validity

Shard #1

Shard #2

Shard #3

CROSS-SHARD TRANSACTION

CROSS-SHARD TRANSACTION
Fisherman

Shard #1

A ← B ← C ← D

Shard #2

V ← W ← X ← Y ← Z

CHALLENGE

CROSS-SHARD TRANSACTION

CHALLENGE PERIOD
Data Availability

Main Chain

**Light Client**
Can't download all the blocks from shard chains

Shard Chains

**Full Nodes**
Data Availability

Merkle Root

Original Data

Extended Data

Any n out on 2n are sufficient to reconstruct
Proposed Protocols
Cosmos
Cosmos

 Validators do not rotate between Zones

 IBC to transfer assets between Zones
Polkadot
Polkadot

Parachain runtime is on-chain

In principle possible to rotate validators

Fisherman
Zilliqa

Only shard processing, do not shard state

Only parallelize single shard transactions

Stop all shards to batch-process cross-shard transactions
Ethereum Serenity

Main Chain provides staking

Beacon Chain provides random numbers

Shard Chain provides data

VM provides state execution result

PoW

Casper / PoS

Shard N
Near Protocol

Main Chain

Shard Chains
Near Protocol vs Ethereum Serenity

**Main Chain**

**Ethereum:** GHOST + Casper FFG among all validators

**Near:** Doomslug, validators rotate every epoch, (it’s a setting)
Near Protocol vs Ethereum Serenity

**Ethereum:** Proposers + Attesters + Cross-linking

**Near:** Fast finality (roughly 3 seconds)
Crypto/business
Why are we doing any of this in the first place?

- PoW → PoS → DPoS → Sharded DPoS
- Locking value? Insurance? Supply chain?
- Interesting promises from early Eth
  - DAOs
  - Programmable Money
  - New business models
Behavior

As it relates to crypto
Behavior

As it doesn’t relate to crypto
Thank You

Check out code

- [http://near.dev](http://near.dev) -- example apps
- [http://github.com/nearcore](http://github.com/nearcore) -- core chain code

Whiteboard Series (Cosmos, Solana, Ontology, more to come...)

- [http://near.ai/youtube](http://near.ai/youtube)

Code is open, all the discussions are public

- [http://near.chat](http://near.chat)