

Hyperscale Weekly

Split the work, not the world

MILESTONE 1 PROGRESS

76%

Previously: flightofthefox had completed a full halt-recovery system and was ready to wire validator-shuffling logic into production.

This was a big, busy week. flightofthefox hardened the exact moment shards split and merge — the trickiest part of adaptive sharding — while also building a complete fork-detection and recovery system, overhauling the project's cryptographic foundations, and standing up a live browser demo of the simulation.

125

+108% vs last wk

COMMITTS

+11k -2.1k

-5% vs last wk

LINES CHANGED

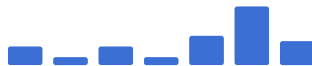
300

FILES TOUCHED

34+

DAY COMMIT STREAK

COMMITTS / DAY



peak 48/day · 7/7 days active

WHAT FLIGHTOFTHEFOX BUILT

Resharding cut-over, nailed down

The moment a shard splits or merges is the hardest part of adaptive sharding. flightofthefox simplified how each new shard's starting state is derived, fixed committee handoff timing so it flips at the cut not an epoch later, and proved the lifecycle in formal models.

'see "The Library" on [hyperscale.rs](#)

Forks detected, fenced, and punished

A complete fork-detection system now catches validators who vote for rival blocks, gossips the evidence to the beacon, fences the offending shard until recovery completes, and jails the culprits — replacing soft penalties with key revocation and pool-wide conviction.

'see "The Overlap" on [hyperscale.rs](#)

Crypto rebuilt from scratch

flightofthefox created a dedicated crypto interface with separate BLS and mock implementations, routing all signing and verification through clean traits — making the system safer to run and easier to test deterministically.

'see "The System" on [hyperscale.rs](#)

THE BOTTOM LINE

STANDOUT CHANGE

The exact moment a shard splits or merges is now derived through a single, tested, formally modeled path — the keystone of adaptive sharding.

WHAT'S NEXT

Likely next: completing the validator-shuffling integration into the live system, now that shuffle intervals are derived from timing parameters and the beacon's conviction and revocation machinery is in place.

From the community

What people are saying, and the words behind the work.

HEARD IN THE CHAT

The mood: A heated exchange about marketing and communication, plus interest in community testing and a new sharding FAQ comparison page on the website.

“The main issue with the design at the moment is it just slaps byzantine nodes with a wet bus ticket.”

— FLIGHTOFTHEFOX, IN THE COMMUNITY TELEGRAM

THIS WEEK’S CONCEPT

The Library — *The moving day that never happens.* The reshape cut-over — the precise instant a shard’s state subtree is handed to its children — is the heart of adaptive sharding, and this week it was simplified, tested, and formally modeled end to end.

hyperscale.rs/library

JARGON, DECODED

Equivocation — When a validator signs conflicting votes for the same round, which is self-proving evidence of dishonesty.

Key revocation — Permanently disabling a validator’s identity as punishment for proven cheating, replacing a temporary jail sentence.

Cut-over — The exact moment a shard split or merge takes effect, when committees and state hand off to the new arrangement.

The bigger picture

Where the work landed, and the map to read along on hyperscale.rs.

MOMENTUM • LINES CHANGED / WEEK



last 8 weeks • newest right

THIS WEEK'S WORK



93% Rust 7% Config

WHERE THE WORK LANDED

types • [The System](#)

crypto newtypes, wave registry, transfer manifest consolidation

beacon • [The Lottery](#)

jailing, conviction tiers, shuffle interval derivation

shard • [The Overlap](#)

fork detection, fork fences, equivocation evidence

THE FULL PICTURE

The System

The whole sharded design at a glance.

The Census

The leaderless beacon of validators, stake and shards.

The Archive

Every published byte is preserved or provably expired.

The Lottery

Random, ever-moving committees; proven cheats are jailed.

The Crash Lab

The simulator that replays any failure byte-for-byte.

The Journey

One transaction's trip through the network.

The Overlap

Why two conflicting blocks can never both commit.

The Library

All state in one merkle tree; a shard is a subtree.

The Triage

Under overload, urgent traffic never waits for bulk.

The Proof

Proving safety with maths, not just tests.

The Clock

Consensus-attested time across independent shards.

The Generals

All-or-nothing cross-shard commits, computed not voted.

The Will

How in-flight transactions settle when a shard dies.

The Governor

Validator entry is re-priced every epoch, like a market.

The Asterisks

Every design's trade-offs — including Hyperscale's own.

THE ROAD TO MAINNET



Now: Milestone 1 • Adaptive Sharding — Week 13 • ~month 3 of 4 • 76%. Since day one: 2,527 commits — one developer, fully in public.

LINKS

WEBSITE

hyperscale.rs

DISCUSS THIS ISSUE ON X

x.com/i/status/2081605978785456342

CODE (PUBLIC)

github.com/hyperscalers/hyperscale-rs

MILESTONE PROPOSAL (PDF)

hyperscale.rs/xian_proposal.pdf

COMMUNITY (TELEGRAM)

t.me/hyperscale_rs

AUTHOR

[flightofthefox](#)



scan for

hyperscale.rs

Questions? [flightofthefox](#) is always happy to discuss in the community Telegram — t.me/hyperscale_rs.