

Hyperscale Weekly

Split the work, not the world

MILESTONE 1 PROGRESS

59%

Previously: flightofthefox had just folded the simulator and production into one shared test-scenario framework, with a clean reshape orchestrator driving both runners.

Building on last week's test consolidation, flightofthefox turned to safety and uncovered the biggest bug yet — a rare way the beacon could finalise the same epoch two different ways across a network split. He reproduced it in the simulator, closed it with a new pool-ratified commit path, and proved the fix in five new mathematical models. A landmark week for the beacon's trustworthiness.

82

-15% vs last wk

COMMITTS

+11k -6.1k

+16% vs last wk

LINES CHANGED

166

FILES TOUCHED

34+

DAY COMMIT STREAK

COMMITTS / DAY



peak 17/day · 7/7 days active

WHAT FLIGHTOFTHEFOX BUILT

Beacon double-commit, found and fixed

flightofthefox reproduced a rare safety bug where the beacon could finalise the same epoch two different ways across a network split, then built a pool-ratified commit path that mutually excludes the old route — closing the door for good.

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

Straddlers and reshapes, more strongly fenced

Transactions caught in a splitting shard can no longer be evicted while a gate still holds them, and the handoff between old and new committees now waits until the successor is actually live.

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

Five new proofs in math

Five new formal models now prove core safety rules in math: straddler settlement, cross-shard atomic commit, shard consensus, the weighted-time clock, and the dual-commit fix itself.

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

THE BOTTOM LINE

STANDOUT CHANGE

The beacon's rare double-commit safety gap is now closed, mutually excluded, and proven in math.

WHAT'S NEXT

Likely next: the lottery-style validator shuffling that's the other big piece of M1 still to land, now with a much firmer beacon underneath it.

From the community

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

THIS WEEK'S CONCEPT

The Census — *The beacon that tracks validators & topology.* The central event was the beacon's double-commit safety gap being reproduced, fixed with pool ratification, and mathematically proven. That's a major step forward for the chain that tracks validators and topology.

hyperscale.rs/census

JARGON, DECODED

Pool ratification — A way for validator pools to vote on and approve a beacon block before it counts as final, replacing a less safe older path.

Dual commit — A safety bug where two conflicting versions of the same block could be finalised at the same time across a split network.

The bigger picture

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

MOMENTUM • LINES CHANGED / WEEK



THIS WEEK'S WORK



WHERE THE WORK LANDED

beacon • The Census

dual-commit gap closed with pool ratification

simulation • The Crash Lab

reproduced the partition bug end-to-end

types • The Library

new ratification and straddler data types

THE FULL PICTURE

The System

The whole sharded design at a glance.

The Journey

One transaction's trip through the network.

The Clock

Consensus-attested time across independent shards.

The Census

The leaderless beacon of validators, stake and shards.

The Overlap

Why two conflicting blocks can never both commit.

The Generals

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

The Archive

Every published byte is preserved or provably expired.

The Library

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

The Will

How in-flight transactions settle when a shard dies.

The Lottery

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

The Triage

Under overload, urgent traffic never waits for bulk.

The Governor

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

The Crash Lab

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

The Proof

Proving safety with maths, not just tests.

The Asterisks

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

THE ROAD TO MAINNET



Now: Milestone 1 • Adaptive Sharding — Week 10 • ~month 2 of 4 • 59%. Since day one: 2,324 commits — one developer, fully in public.

LINKS

WEBSITE

hyperscale.rs

CODE (PUBLIC)

github.com/hyperscalers/hyperscale-rs

MILESTONE PROPOSAL (PDF)

hyperscale.rs/xian_proposal.pdf

COMMUNITY (TELEGRAM)

t.me/hyperscale_rs

AUTHOR

[flightofthefox](https://github.com/flightofthefox)



scan for
hyperscale.rs

Questions? *flightofthefox* is always happy to discuss in the community Telegram — t.me/hyperscale_rs.