XI'AN · HYPERSCALE FOR RADIX

WEEK 29 · 13 JUL 2026
13–19 Jul 2026

Hyperscale Weekly

Split the work, not the world

CURRENT MILESTONE

Milestone 1 · Adaptive Sharding

Teaching the network to split and merge its shards on its own

Week 12

Previously: flightofthefox expanded formal proofs with Model F covering the full reshape lifecycle, ran adversarial Monte Carlo analysis on validator shuffling, and moved safe-vote and ratification structures to durable

If last week was about proving the reshape math, this week was about making shards survive the worst. flightofthefox built an entire halt-recovery system — the machinery to detect a frozen shard, redraw its committee, and bring it back to life without losing or duplicating a single transaction. It was a big, dense week with 60 commits across nearly every part of the codebase.

60	+12k −1.8k	142	34+
+9% vs last wk	-10% vs last wk		
COMMITS	LINES CHANGED	FILES TOUCHED	DAY COMMIT STREAK

COMMITS / DAY  peak 18/day · 7/7 days active

WHAT FLIGHTOFTHEFOX BUILT

Shards that heal themselves

flightofthefox built a full halt-recovery pipeline: detecting stalled shards, re-drawing their entire committee, bridging consensus over a proven parent, and freezing cross-shard transactions until the shard is safe again.

'see "The Will" on hyperscale.rs

Beacon punishes silence, rewards recency

The beacon now jails validators who withhold randomness proposals, weights committee re-samples by how recently each validator served, and evicts the longest-tenured member each shuffle — keeping the lottery fair and alive.

'see "The Lottery" on hyperscale.rs

Randomness by folding, proven by math

Each shard now publishes a per-block randomness reveal that gets folded into the epoch's shared randomness — the seed for committee draws. A new Quint model proves the fold is correct.

'see "The Proof" on hyperscale.rs

THE BOTTOM LINE

STANDOUT CHANGE

A complete halt-recovery system now lets a frozen shard be detected, re-staffed, and restored without losing or duplicating cross-shard transactions.

WHAT'S NEXT

Likely next: wiring the validator-shuffling logic into production now that halt recovery and the beacon's committee-management are in place.

Written by AI from public commits, the community chat and hyperscale.rs — may contain mistakes.

From the community

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

HEARD IN THE CHAT

The mood: Community discussion about visibility on X and the radix.wiki site, plus an off-topic drama that got wiped by mods. flightofthefox clarified consensus naming and stressed the value of combining simula

> *“Seems like we're having a week every 2 days 😅”*
>
> — FLIGHTOFTHEFOX, IN THE COMMUNITY TELEGRAM

THIS WEEK’S CONCEPT

The Will — *The dead shard's will.* The heart of this week was giving dead shards a clean path back: detecting the stall, freezing cross-shard work, rebuilding the committee, and ensuring straddlers resolve by the fence — not by luck.

hyperscale.rs/testament

JARGON, DECODED

Halt recovery — The process of detecting a frozen shard, assigning it a new committee, and bringing it back online without losing transactions.

Randomness reveal — A value each shard publishes every block that gets blended together to create the unpredictable seed for the next committee draw.

Fence freeze — Pausing all cross-shard transactions involving a halted shard until its fate is settled.

Written by AI from public commits, the community chat and hyperscale.rs — may contain mistakes.

ZOOM OUT

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



89% Rust · 4% Docs · 4% Specs · 3% Config

WHERE THE WORK LANDED

node · The Will
halt-recovery pipeline and committee rebuild

types · The Will
recovery data structures and fence types

beacon · The Lottery
jailing, recency weighting, committee caps

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

M1	M2	M3	M4	M5	M6
Adaptive Sharding ~4 mo	Radix Engine ~5 mo	Gateway & API ~3 mo	Validator GUI ~3 mo	Migration ~3 mo	Live support 12 mo

Now: Milestone 1 · Adaptive Sharding — Week 12. Since day one: 2,402 commits — one developer, fully in public.

LINKS

WEBSITE
hyperscale.rs

DISCUSS THIS ISSUE ON X
x.com/i/status/2079069081660002690

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.

Written by AI from public commits, the community chat and hyperscale.rs — may contain mistakes.