XI'AN · HYPERSCALE FOR RADIX

WEEK 22 · 25 MAY 2026

25–31 May 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 5

Previously: flightofthefox had scaffolded the beacon crate — the slow coordination chain that will track validators and the shard map — and reshaped the production node around a thread-per-shard model, with The Crash Lab and The Pr

This was a turning-point week. flightofthefox pushed a compiler-enforced safety system across the entire codebase, so unverified blocks or votes literally cannot be used by accident. He also fully wired the beacon into the production node and replaced an older recovery path with a cleaner skip flow.

184	+15k −11k	297	15
-27% vs last wk	+22% vs last wk		
COMMITS	LINES CHANGED	FILES TOUCHED	DAY COMMIT STREAK

COMMITS / DAY



peak 45/day · 7/7 days active

WHAT FLIGHTOFTHEFOX BUILT

Compiler-enforced safety everywhere

A new typestate wrapper now forces the compiler to reject any code that tries to use a block, vote, or certificate before it has been cryptographically checked. Unverified data can no longer slip through by accident — a deep structural safety win for The Overlap.

'see "The Overlap" on hyperscale.rs

The beacon wakes up

The beacon was fully plugged into the production node: sync, gossip, fetch pools, timers, and the SPC state machine all came online. An older 'recovery' path was retired in favour of a simpler skip flow for missed blocks.

'see "The Census" on hyperscale.rs

The Crash Lab grows teeth

The simulator absorbed multi-coordinator, Byzantine, and topology-change scenarios, plus cross-network signature rejection tests, so a wider range of attacks can now be replayed byte-for-byte in The Crash Lab.

'see "The Crash Lab" on hyperscale.rs

THE BOTTOM LINE

STANDOUT CHANGE

A compiler-enforced 'verified' wrapper now covers nearly every consensus type, making it structurally impossible to trust an unverified block, vote, or certificate.

WHAT'S NEXT

Likely next: driving a full beacon epoch transition end-to-end through the simulator, including the new skip flow.

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: Quiet week in chat — flightofthefox heads-down on the codebase.

THIS WEEK'S CONCEPT

> **The Overlap** — *Provably fork-free.* Nearly every consensus type now has to be cryptographically verified before the compiler will let the code touch it, making it structurally impossible for unverified data to influence a decision. This is fork-safety baked into the type system.
>
> hyperscale.rs/overlap

JARGON, DECODED

Typestate — A coding pattern where the type itself records whether a step has happened, so the compiler refuses to let you use something before that ste

Gossip — The way nodes spread messages peer-to-peer, each one passing the news to a few neighbours who pass it on, like a rumour.

Unicast — Sending a message to one specific node, the opposite of broadcasting to everyone.

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



96% Rust 4% Config

WHERE THE WORK LANDED

types · The Overlap
typestate wrappers for verified consensus types

beacon · The Census
wired into node, sync, fetch, skip flow

node
BeaconCoordinator wired into the runner

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	Radix Engine	Gateway & API	Validator GUI	Migration	Live support
~4 mo	~5 mo	~3 mo	~3 mo	~3 mo	12 mo

Now: Milestone 1 · Adaptive Sharding — Week 5. 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

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.