XI'AN · HYPERSCALE FOR RADIX

Hyperscale Weekly

Split the work, not the world

WEEK #15
17–23 Aug 2026
Milestone 2 · Radix Engine
Week 3 · ~month 1 of 5

MILESTONE 2 PROGRESS 14%

Previously: flightofthefox was in week 2 of Milestone 2, overhauling the authorization model to use badges and proofs, adding post-quantum signatures, and hardening cross-shard straddler rules.

Week 3 of Milestone 2 was a deep dive into the virtual machine's bones. flightofthefox rebuilt how the VM describes and stores component state, how resources like tokens identify themselves, and how nodes fetch records from each other across shards — all while musing about a privacy future for the network.

409
+66% vs last wk
COMMITS

+31k −11k
+46% vs last wk
LINES CHANGED

444
FILES TOUCHED

8
DAY COMMIT STREAK

COMMITS / DAY  peak 74/day · 7/7 days active

WHAT FLIGHTOFTHEFOX BUILT

The VM learns to describe itself

flightofthefox gave the VM a formal vocabulary for component layouts, field sizes, and event payloads — so the engine can read and refuse malformed data before it ever runs, making smart contracts safer by construction.

'see "The Crash Lab" on hyperscale.rs

Resources find their own addresses

Tokens and other resources now derive their identity from the issuer that created them rather than an external label, making ownership and minting rules self-consistent and tamper-resistant across every shard.

'see "The Library" on hyperscale.rs

Each node fetches its own truth

Nodes now pull component records from the shard that sealed them instead of keeping a global copy, and each simulated node gets its own engine world — a major step toward realistic multi-shard testing.

'see "The Journey" on hyperscale.rs

THE BOTTOM LINE

STANDOUT CHANGE

The VM can now formally describe and validate its own component layouts, event payloads, and resource identities — turning runtime crashes into compile-time refusals.

WHAT'S NEXT

Likely next: continuing to wire resource behaviors (minting, burning, collections) into the newly shaped VM, and pushing the per-node derivation path deeper into cross-shard record fetching.

From the community

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

HEARD IN THE CHAT

The mood: A lively week — flightofthefox floated the idea of optional privacy (Zcash-style shielding) built natively into the sharded VM, sparking debate about regulation and whether Radix itself should be shie

> *"what if the best sharded blockchain, can also be the best privacy blockchain"*
>
> — FLIGHTOFTHEFOX, IN THE COMMUNITY TELEGRAM

THIS WEEK'S CONCEPT

> **The Crash Lab** — *Rehearsing the rarest day.* Most of the week was about making the VM self-describing — giving every record, field, and event a formal shape the engine checks — which is the foundation for deterministic, replayable multi-shard testing.
>
> hyperscale.rs/crashlab

JARGON, DECODED

Seal — A cryptographic stamp that proves a component or resource was created by the right authority.

Cell — A storage slot in the VM that holds a piece of a component's state, like a field in a record.

Wasmtime — The engine that actually runs compiled smart contracts; flightofthefox upgraded it to the latest version.

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

83% Rust · 15% Config · 1% Docs

WHERE THE WORK LANDED

vm/sdk · The Crash Lab
VM shape descriptions and record layout overhaul

vm/harness · The Crash Lab
Per-node engine worlds and fuzz lane rebuild

node · The Journey
Cross-shard record fetching and derivation

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
done	~5 mo	~3 mo	~3 mo	~3 mo	12 mo

Now: Milestone 2 · Radix Engine — Week 3 · ~month 1 of 5 · 14%. Since day one: 3,623 commits — one developer, fully in public.

LINKS

WEBSITE
hyperscale.rs

DISCUSS THIS ISSUE ON X
x.com/i/status/2091752933410935053

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.