

Hyperscale Weekly

Split the work, not the world

MILESTONE 2 PROGRESS

18%

Previously: flightofthefox had reshaped the VM's internals — formal component layouts, resource identity, and per-node shard record fetching — pushing toward feature parity with Babylon.

This was a week of deep, grinding VM hardening: flightofthefox made the Radix Engine refuse invalid constructions at every seam, formalized how vaults and resources are described, and restructured large swaths of code into cleaner modules. In the community, a lively discussion about formal verification and privacy showed just how much the project's testing-first philosophy resonates.

349

-21% vs last wk

COMMITTS

+17k -6.9k

-44% vs last wk

LINES CHANGED

355

FILES TOUCHED

0

DAY COMMIT STREAK

COMMITTS / DAY



peak 134/day · 5/7 days active

WHAT FLIGHTOFTHEFOX BUILT

The VM learns to say no

Dozens of commits make the engine reject bad inputs — duplicate fields, half-applied supply changes, invalid shapes — at build time instead of crashing at runtime, making the whole system safer by construction.

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

Vaults and resources take form

Vaults are now named state fields with explicit mint-burn pairs, supply changes must apply whole or not at all, and token denominations are pinned to their resources — giving the token system a precise, checkable structure.

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

Proofs get provenance and structure

Authority proofs now carry traceable provenance, threshold sign-in gets a formal presenting path, and badge-gating ties to protocol-level evidence — making access control verifiable rather than assumed.

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

THE BOTTOM LINE

STANDOUT CHANGE

The Radix Engine now rejects an enormous range of invalid constructions — bad shapes, duplicate fields, half-applied token changes — before code ever runs, making smart-contract failures a build-time problem rather than a runtime crash.

WHAT'S NEXT

Likely next: continuing to wire resource operations like minting, burning, and vault movements into the hardened VM, and finishing the proof-and-authorization presenting paths that are clearly mid-flight.

From the community

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

HEARD IN THE CHAT

The mood: Chat buzzed about formal verification after a Zcash bug made headlines, with flightofthefox explaining how privacy relays could work and noting he hasn't chased M1 payment admin yet — too busy buildin

“the better integrated robust testing is from the start - the more it unlocks being able to move at ridiculous velocity - and move with confidence”

— FLIGHTOFTHEFOX, IN THE COMMUNITY TELEGRAM

THIS WEEK'S CONCEPT

The System — *Split the work, not the world.* Nearly every commit this week drilled into the Radix Engine's core architecture — the execution layer that runs every smart contract — hardening its type system, resource model, and authorization against entire classes of failure before they can ever occur.

hyperscale.rs

JARGON, DECODED

Vault — A named field inside a smart-contract component that holds a specific kind of token, tracking how much of that resource it contains.

Provenance — A record of where a proof of authority came from, so the system can verify that the right person signed off on an action.

Manifest — A blueprint file that describes what a smart-contract package contains and how it should be published to the 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

vm/effects • The System

63 files: resource-operation refusals and validation

vm/sdk • The System

52 files: manifest builder and construction-time projections

vm/harness • The Crash Lab

38 files: test consolidation and in-repo guest scaffold

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

M1

Adaptive Sharding
done

M2

Radix Engine
~5 mo

M3

Gateway & API
~3 mo

M4

Validator GUI
~3 mo

M5

Migration
~3 mo

M6

Live support
12 mo

Now: Milestone 2 • Radix Engine — Week 4 • ~month 1 of 5 • 18%. Since day one: 4,003 commits — one developer, fully in public.

LINKS

WEBSITE

hyperscale.rs

DISCUSS THIS ISSUE ON X

x.com/i/status/2094289473504784666

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](https://t.me/hyperscale_rs) is always happy to discuss in the community Telegram — t.me/hyperscale_rs.