WEEK #28
16–22 Nov 2026

Hyperscale Weekly

Split the work, not the world

THIS WEEK'S WORK

Pausing this week, with a pending fork-safety test scenario waiting in the simulator

Previously: flightofthefox had just tightened the shard chains' certificate rules — including judging certificates by their signed epoch instead of any clock — and left a pending fork scenario to test.

A quiet week on the code front: no commits landed between 16 and 22 November. The community chat was the centre of gravity instead, with a candid conversation about funding, timelines, and one sharp technical question about how hot contracts behave on a sharded network. flightofthefox was as open as ever about all three.

0	+0 −0	0	0
-100% vs last wk	-100% vs last wk		
COMMITS	LINES CHANGED	FILES TOUCHED	DAY COMMIT STREAK

WHAT FLIGHTOFTHEFOX BUILT

Funding, on his own terms

flightofthefox confirmed he still won't apply for more grants, saying applications tend to invite critics rather than help. He remains sole-funded while the community openly wishes it could support him.

The hot-contract question

A member asked whether a contract with one shared record — like an AMM pool — handles only about one transaction per second. It's a fair question about how the mempool gate serialises competing transactions, and a good window into the design.

'see "The Journey" on hyperscale.rs

THE BOTTOM LINE

STANDOUT CHANGE

No code landed this week — the story was flightofthefox candidly explaining why he won't take grants and why he won't promise dates.

WHAT'S NEXT

Likely next: resuming work on the pending fork scenario and fetch-grace rules for adopted certificates, whenever his available time allows.

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 members pressed flightofthefox on timelines and funding; he answered both candidly, and the chat debated whether a DAO offer could keep him working 'in hypermode'.

> *"there's no timelines because there's no way to predict how much time i'll have over the next months"*
>
> — FLIGHTOFTHEFOX, IN THE COMMUNITY TELEGRAM

FLIGHTOFTHEFOX EXPLAINED

Why no timelines

There are no timelines because there's no way to predict how much time he'll have over the coming months.

Grants still off the table

He still doesn't want to apply for grants — applications invite nitpickers — and he's miffed the RDX foundation never followed through with its matching.

THIS WEEK'S CONCEPT

The Journey — *One payment crosses a sharded world.* The week's only real technical discussion was about the transaction path — how one hot account forces transactions to queue, which is exactly what The Journey's mempool gate governs.

hyperscale.rs/journey

JARGON, DECODED

Mempool — The waiting room where transactions sit before the network processes them.

In-flight transaction — A transaction that has started but not yet finished its journey across shards.

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

WHERE THE WORK LANDED

A quiet week in the codebase.

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 FOUR PARTS

done	–	–	–	–
Adaptive Sharding done Aug 2026	The new VM	Indexing & APIs	Validator tooling & testnet	Foundations & reliability

Share of lines changed this week, by part; shared groundwork counts as Foundations & reliability. Since day one: 6,138 commits — one developer, fully in public.

LINKS

WEBSITE

hyperscale.rs

DISCUSS THIS ISSUE ON X

x.com/i/status/2124729940155285696

CODE (PUBLIC)

github.com/hyperscalers/hyperscale-rs

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.