PROJECT OVERVIEW · SEPTEMBER 2026

Astryum

Non-custodial capital control plane for XRP. Put your capital to work, or govern it together under a council the ledger itself enforces.

Your capital. Your control. Your signature.

THE TEAM

Two founders. The gap between them is the product.

Eric Guiral — architecture and protocol

Uses DeFi daily. The XRPL ↔ Flare tunnel (FAssets, Smart Accounts, the Data Connector) already existed as Flare infrastructure; he gave it a meaning and a use case, and built the rest: the contracts, the credential rail, the backend. Most of the ideas here started as things he was missing himself.

Guillem Esteban — product

Holds XRP and, until this existed, that was about all he did with it. Everything out there is made for people who live inside DeFi. He doesn't. The bar he sets: *if a screen needs explaining before it makes sense, it isn't finished.*

Two people, one repository, mainnet since July. Company wrapper in progress; XRPL Grants application in October.

THE PROBLEM

Three things people say about their XRP — and none of them is about crypto.

“My money is sitting still.”

Today XRP has two options: stay idle, or be handed to someone who custodies it. Every tool for a third option is built for people who already live inside DeFi.

“We are several, and one person has the keys.”

A family or a group holding capital together has a design flaw: one person can always move it alone — or a custodian holds it for all of them.

“I want someone to manage it without being able to rob me.”

Delegating the decision should not mean delegating possession. In DeFi, who the manager is lives off-chain; in TradFi, the custodian decides when you may leave.

All three answers need primitives only a ledger has: a quorum the network enforces, credentials with expiry anyone can check, contracts that refuse what does not fit the rules.

WHAT ASTRYUM IS

One web app. The wallet you already have. Rules the ledger enforces.

- **Connect** Xaman and see all your capital in one map — XRPL, Flare, EVM.
- **Put it to work** on Flare with one XRPL signature: no second wallet, no gas token, no bridge to operate.
- **Hold it together** under a council of real people, with a constitution anchored on the ledger.
- **Delegate its management** to someone whose credential is verified on the ledger, inside a contract that cannot pay principal to anyone.
- **Always sign yourself.** We prepare the transaction, show all of it — fees included — and stop.

The sentence that governs everything

A person decides.
The code decides what they can't.

Astryum never signs, never custodies, never executes with discretion. Not a policy — an absence of capability: there is no function to sign, broadcast or hold user funds in the system, and a boot guard enforces it.

THE MECHANISM

XRPL governs. Flare executes. The user signs.

XRP Ledger
Who has authority: a council with a quorum, a credential with expiry, a designation. The constitution. Settlement of XRP itself.

signed order, memo = commitment →

Flare Data Connector
The notary: attests that *this* XRPL account signed *this* commitment. Merkle proof.

proof →

Contract on Flare
Obeys exactly those bytes, or reverts. The "no" is given by the ledger, not by Astryum.

capital works →

Venues
Kinetic, Firelight, earnXRP, Monarq, FTSO — each behind its own scanner and switch.

Through the tunnel travels the order — never the capital

Except XRP itself, as FXRP, its legitimate representation on Flare. We never route capital that is not XRP through the XRP Ledger, and never bridge anything *to* XRPL first.

No custodian in the path

Astryum's own keys sign only Astryum's own transport transactions — attestation requests and proof deliveries. They carry bytes the user already committed to and cannot change a byte, an amount or a destination.

RAIL A — THE PERSONAL PATH

One XRPL signature. N actions on Flare. No FLR, no EVM wallet.

1 · The user composes a transaction
In the app, or with the help of an AI agent: “put 100 XRP to work in Kinetic.”

→

2 · Astryum prepares
Composes the batch the user’s Flare account will run, hashes it, reads protocol fees live, simulates. Returns an *unsigned* XRPL Payment to the FAssets Core Vault with a memo carrying the hash.

Xaman signs and broadcasts →

3 · XRP Ledger
The Payment settles. Our Source Tag is on it. No DestinationTag — it would misroute the mint.

attestation (~2–5 min) →

4 · FDC + executor
Astryum’s executor requests the attestation (~40 FLR per operation in practice) and delivers the proof to the AssetManager. Zero discretion.

→

5 · FXRP minted into the user’s Personal Account
Derived from the XRPL address (CREATE2), no sign-up. In the same transaction it executes exactly the batch whose hash was in the memo.

Fees, all read live from the protocol

Minting 10 bps (min 0.1 XRP) · executor 0.2 XRP · redemption 18 bps. Shown before the QR. Never ours, never hardcoded.

The way back

“Convert to XRP” redeems FXRP to XRP on the user’s own XRPL address — inside the same memo-authorized batch. Exit queues counted in real time.

Status

LIVE mint, deposit, withdraw, repay on mainnet; redemption end to end 31 Jul; full rail end to end 22 Aug.

RAIL B — THE COUNCIL ORDER

A group governs a contract without anyone holding a key to it.

Root account on XRPL
A family council (weighted SignerList with a quorum, master key off) or an accredited root with credentials.

1-drop Payment, memo = hash(nonce, calldata) →

Anchor account
rLcoFM…YDn7 — can become a door closed by consensus (DepositAuth + credential preauth).

FDC proof, relayed by Astryum →

Bridge (XrplCouncilBridge v2)
Nine guards: proof verifies · type & source · source hash = council · tx succeeded · memo = hash · txId unused · nonce next · anchor matches · then vault.call(bytes).

fits the rules →

Cage / vault
Direct capital to a listed venue, recall it. Nothing else exists.

does not fit →

Revert
The ledger says no. Astryum has no vote.

Each member signs their part, when they can

Asynchronous signing tray in Xaman; Tickets keep the sequence free; the browser combines and broadcasts. The backend never touches a signature.

Proven

LIVE Four council orders executed end to end on the Legacy stack (bridge nonce = 4); the last one in 1 min 40 s. Cage v2 bridges at nonces 3 / 1 / 3.

WHY THE XRP LEDGER IS LOAD-BEARING

Remove the ledger — which capability disappears?

PRIMITIVE	WHAT DISAPPEARS WITHOUT IT	STATE
Weighted SignerList + master key disabled	The council, the heirs, the reinforced account. “3 of 4 agreed” is a fact of the network, not a row we could edit.	LIVE
Payment with Memo	One signature in the wallet you already have orders N actions on another chain. The transporter cannot change a byte.	LIVE
Credentials (XLS-70)	Licence and identity as ledger objects with expiry, accepted by the subject, checkable by anyone. The credential *is* the account type.	LIVE gate
Designation by credential	Hierarchy between accounts as a ledger object; revoking beheads the subordinate without touching any client’s exit.	LIVE
DepositAuth + credential preauth	The door closed by consensus itself — no server of ours in the middle.	BUILT
DIDSet	The constitution anchored in the account’s own record; cannot be rewritten quietly.	LIVE
Escrow	Scheduled payments that settle themselves; finish is permissionless.	LIVE
Source Tag	Verifiable attribution; survives multisig. Our operational keys never carry it — on purpose.	LIVE

Not used, and not announced as ours: Batch (not on mainnet) · Permission delegation XLS-75 (not active; not one line of code until it is — and then transport, never the security boundary) · Permissioned Domains (composed, no consumer yet) · Zero-knowledge credentials (announced for XRPL; on our radar, no code).

WHY IT TAKES BOTH NETWORKS

The product exists only in the overlap.

Take XRPL away

Governance goes back to being a promise on somebody's server. The quorum, the credential, the constitution and the settlement of XRP are jobs the ledger does that a company would otherwise do.

Take Flare away

The rules go back to being a setting in an app. "Capital can only go to this list, the manager may direct but never extract, the client leaves whenever they want" must live as a contract, and the XRP Ledger is not a general-purpose state machine.

XRPL is where *who* is decided. Flare is where *what may happen* is enforced. The Data Connector is the notary between them — and there is no custodian in the path.

Honest about the edge: FDC's EVM attestation reaches only Ethereum, Flare and Songbird today, so the common layer across chains is not a chain — it is the signature. A fact goes on a chain only when a contract has to obey it.

THE PRODUCTS

One kernel, audited once. Each product is a small rule module.

Personal LIVE

Self-custody. Your Xaman commands your account on Flare. Capital map; one-signature strategies; the way back to XRP.

Reinforced LIVE

Self-custody. A personal wallet where no single key moves anything: 2-of-3 of your own keys, master off. Same ceremony as Legacy, no cage.

Legacy LIVE

Self-custody by the council. A family governs by quorum: council, anchored constitution, a vault that cannot pay principal to anyone. *Separate deck.*

Managed vaults LIVE

Self-custody: the client's shares are theirs. A manager with a credential verified on the ledger directs capital inside a cage. Client exits, nobody gates. *Separate deck.*

Exchange LIVE

Custodied by the exchange — with the client's shares in their own name. The structure of an exchange, adopted by an operator: root, designated omnibus, clients by tag. *Separate deck.*

Credentials · MoneyFlows · /proof LIVE

The access layer; rules that prepare and never sign; the page that verifies our claims in your browser.

Kernel shared by all: FDC verification · binding to one XRPL account · anti-replay · canonical receipt · exit never gated. If a piece has to ask "which account type are you?", it is misplaced.

PRODUCT 1 OF 6 · PERSONAL ACCOUNT LIVE

Your XRPL wallet commands your account on Flare. Self-custody, one signature.

What the user lives

Logs in with Xaman — the wallet is the identity. Sees all their capital in one map (XRPL, Flare, EVM in read-only). Puts XRP to work on Flare with one signature: no FLR, no EVM wallet, no bridge to operate. Leaves from the same screen: "Convert to XRP".

How it works

1 The user composes the operation in the app (or with the AI agent's help); Astryum prepares an *unsigned* XRPL Payment to the FAssets Core Vault whose memo commits to the batch. **2** The user signs in Xaman; FDC attests; the executor delivers the proof. **3** FXRP is minted into the user's Personal Account (derived from the XRPL address) and the account executes exactly that batch — supply, borrow, repay, redeem.

Custody · status · proof

Self-custody: no key of the user touches our servers; the executor cannot change a byte. Live venues: Kinetic (supply, the carry), Firelight, earnXRP, Monarq, FTSO delegation. **68 transactions** on mainnet in the inventory, including redemption payouts by independent FAssets agents — the unmint completed on XRPL.

Fees read live from the protocol and shown before the QR: minting 10 bps (min 0.1 XRP) · executor 0.2 XRP · redemption 18 bps. Never ours, never hardcoded.

PRODUCT 2 OF 6 · REINFORCED ACCOUNT LIVE

A personal wallet where no single key moves anything.

What the user lives

Their own account with the dials turned up: two of three of *their own* keys (two phones and a backup, for example) must sign, and the original single key is switched off. It stays a personal wallet: no constitution, no cage, no inheritance. The middle step between simple (1-of-1) and council (3-of-4).

How it works

1 `SignerListSet` with weights and quorum. **2** A rehearsal: every signer signs a real multisig transaction (an `EscrowCreate`) so nobody closes a door they cannot open. **3** `AccountSet` `asfDisableMaster` — irreversible without the quorum. From then on every Payment, including the memo instruction to Flare, is multisigned; the Source Tag survives it, so each signer counts.

Custody · status · proof

Self-custody, by a quorum of the same person's keys. Landmines refused and documented: a RegularKey is an OR, a pure 2-of-2 is a freezer, closing the master needs the master. On mainnet: `rP49LEKa…`, 2-of-3 since 21 Aug, later a client of a managed vault with a multisigned deposit and exit — **10 transactions** in the inventory.

PRODUCT 3 OF 6 · LEGACY LIVE

A family governs by quorum. Nobody moves the capital alone — not a member, not us.

What the user lives

A council of real people (3 of 4) holds the family's capital. They write the constitution in plain language, anchor it on the ledger, and close the door: the original single key is switched off. From then on every decision needs the quorum. The heirs do not wait for an event — they already are the quorum. Scheduled payments to beneficiaries settle themselves.

How it works

1 SignerList + rehearsal + master off + `DIDSet` (the constitution's hash). **2** One memo instruction from the council deploys bridge + vault through the factory — which only accepts the council's own Personal Account. **3** Orders travel as 1-drop Payments whose memo is the hash of (nonce, calldata); the bridge verifies the FDC proof and nine guards and calls the vault, or reverts. The vault has no `withdrawPrincipal`: capital only goes to listed venues and comes back; only yield leaves to people.

Custody · status · proof

Self-custody by the council. Two councils on mainnet: 3-of-4 since July (four orders executed end to end, the last in 1 min 40 s) and a 2-of-3 born through the factory on 11 Aug. No external audit: own capital, 5 XRP cap per cage. **24 transactions** in the inventory. *Separate deck.*

PRODUCT 4 OF 6 · MANAGED VAULTS LIVE

Delegate the decision, not the possession.

What the two users live

The **manager** opens their vault from their Xaman, picks venues from a governed list and directs capital: send it to work, recall it. The **client** deposits with their own signature, receives shares in their own name, and leaves whenever they want — the exit is owner-only and no credential, manager or Astryum key can gate it.

How it works

1 The manager's root on XRPL carries credentials issued by third parties (a sector credential and an identity credential, with expiry, accepted by the subject); without valid credentials the cage is not born. **2** One memo instruction deploys bridge + cage + first pote through `AstryumCageFactory`. **3** Every order (create pote, direct, recall) is a council order verified by the bridge; the pote only accepts venues from `AstryumRegistry`; the manager's cut is a payee capped by the cage at 20 % of yield; the pote has no fee hook.

Custody · status · proof

Self-custody: the client's shares are theirs. Four cages and five potes on mainnet; three manager roots with credentials; the full cycle create → deposit → direct → recall → redeem executed with two different clients; the venue registry with Kinetic and Firelight live. 168/168 Foundry tests, 62 on the cage v2. **21 transactions** in the inventory. *Separate deck.*

PRODUCT 5 OF 6 · EXCHANGE LIVE

The structure of an exchange, on the ledger. Custodied by the exchange — shares in the client's name.

What the two users live

The **operator** is born by stations from its Xaman: accredited root → constitution → cage → pote → KYC door → designation of its omnibus with two signatures; then a multi-client desk with caps, typed refusals and receipts verified on chain. The **client** needs no XRPL account, no reserves, no gas: a DestinationTag to enter, Face ID to leave, and shares in their own name from the first block.

How it works

1 Two accounts, always: K1 (the root, with custody-sector and company credentials and its constitution) and K2 (the omnibus), bound by an OMNIBUS designation credential. **2** The client deposits XRP into K2 with their tag; K2 signs the memo instruction that mints FXRP and deposits it into the pote in the name of the client's passkey account. **3** The client redeems by Face ID — a relayer pays the gas and cannot decide; XRP payouts from K2 are signed by the exchange inside its caps.

Custody · status · proof

Custodied by the exchange: the XRP in K2 is the operator's custody by design; the shares in the pote are the client's and nobody can block their exit. The per-box KYC record is enforced by the backend, not by the ledger. On mainnet: K1 with credentials, the designation, the cage with two potes, three tagged deposits, two omnibus instructions, a passkey exit — **11 transactions** in the inventory. Rehearsed with our own accounts; no third-party exchange yet. *Separate deck.*

PRODUCT 6 OF 6 · CREDENTIALS · MONEYFLOWS · /PROOF LIVE

The layer every product shares: who may act, rules that never sign, and proof anyone can check.

Credentials — the access layer

Licence and identity as XLS-70 objects on the root account: the issuer signs `CredentialCreate` with a mandatory expiry (and a URI to the attestation — the identity credentials carry a Coinbase attestation); the subject signs `CredentialAccept`, which is the consent. The gate reads them on the ledger before a cage is born and at every order — never at an exit. Astryum never issues a licence: it chooses which issuers it accepts. **30 transactions** in the inventory.

MoneyFlows — rules that prepare, never sign

Standing orders and automations inside signed limits: XRPL Escrow fixes destination and dates, finish is permissionless. When a rule triggers, it prepares the transaction and brings it to the wallet that owns the position for that person to sign. The keeper that pushes time-based escrows signs with an account of our own — never user capital, never with the Source Tag.

/proof — the ledger's word, not ours

Every claim is checked in the visitor's browser against XRPL and Flare: transactions, memos, credential objects, the state of registry, factory, cage and pote. The 176-transaction inventory extends it: each row is a signature on one ledger and its consequence on the other, linked by the hash inside the FDC proof.

ACCOUNTS, CREDENTIALS, LEGAL VEHICLES

One root per legal vehicle. The credential on the root decides the account type.

Legal vehicle (off-chain)

Person · family or foundation · company managing capital · company custodying for clients · company operating a protocol

one root each →

Root on XRPL

Single-sig, or SignerList = the board. Carries the credentials the law already requires: sector licence + identity. Anchors its by-laws (DID). Its book of acts: hash + pointer, never content.

obey by binding →

Arms on Flare

Personal Account, bridge, cage, potes, passkey accounts. No credential of their own — the chain of immutable bindings proves obedience. Gating the root gates the tree.

Issued by third parties, never by us

Issuer signs CredentialCreate off-ledger-verified; the subject signs CredentialAccept — the consent. Mandatory expiry is the revocation. URI points to the attestation; no personal data on the ledger.

The corporate jewel

When the root is the board's SignerList, a multisigned CredentialAccept *is* the board's resolution — native on the ledger, with a real quorum. (Designed; the route signs single-subject today.)

Entry gated, exit never

Revoking a credential freezes governance. It never traps a client. The ledger carries the YES; the NO is never published.

THE INFRASTRUCTURE WE BUILD PRODUCTS OUT OF

Nothing here is ours except the contracts and the backend. That is the point.

XRPL · GOVERNS	Payment + Memo · SignerList · DisableMaster · Tickets · Credentials XLS-70 · DIDSet · Escrow · DepositAuth · Preauth · Source / Destination Tag
FLARE · EXECUTES	FDC XRPPayment attestation · Smart Accounts → Personal Account · FAssets FXRP mint / redeem · FTSO (cost at price) · RIP-7212 passkeys
OUR CONTRACTS	XrplCouncilBridge v1 / v2 · LegacyVault · LegacyStackFactory · AstryumCage · CageFactory · Registry · AstryumVaultV2 (potes) · PasskeyAccount
OUR BACKEND	intent preparation · policy guard · simulation · memo composer · executor · relayer · credential ceremony · notary · gate · exchange desk · omnibus watcher · Source Tag metrics · /proof

No proxy, no pause, no upgrade

Zero `upgradeTo`, `delegatecall`, `whenNotPaused` or `Initializable` in our contracts. The owner is the user's own council, never Astryum.

If Astryum disappears tomorrow

Keys were never ours; capital never sat at our address; rules live in contracts the quorum controls; the exit needs no permission from us. This test is applied to every new piece before it is merged.

TRACTION · WHAT IS LIVE ON MAINNET

Not a prototype. Real councils, real cages, real capital — small on purpose.

2

Legacy councils on XRPL (3-of-4 and 2-of-3): quorum enforced by the network, master key switched off, constitution anchored (DIDSet)

4

council orders executed end to end through FDC to Flare — the last one in 1 min 40 s

4 · 5

cages and potes born on the managed-vault stack v2 (registry, factory, anchor live since 6–8 Sep) — one of them the exchange's

168 / 168

Foundry tests green, 62 on the cage v2 against real venues on a mainnet fork — incl. a theft reproduced in v1 and blocked in v2

Managed vault, full cycle on mainnet

Create pote → deposit → direct to Kinetic / Firelight by council order → recall → redeem. Manager gate fail-closed at birth: without valid credentials on the root, the cage is not born.

Personal rail, end to end

XRP → FXRP → Flare venue with one XRPL signature (22 Aug); FXRP → XRP redemption (31 Jul); the executor refuels itself at cost (25 Jul). Five XRPL transaction types signed by users or quorums on mainnet: Payment, SignerListSet, DIDSet, EscrowCreate, Credentials.

TRACTION · SOURCE TAG 2607090002

Few signatures — each one a ceremony or a movement of real capital.

128

validated XRPL transactions carrying our Source Tag, since the first one on 14 Jul — our own ledger read, 16 Sep 2026

15

signing accounts in those transactions, multisig co-signers included — same read

313.06 XRP

delivered by tagged Payments since 14 Jul — same read

176

mainnet transactions in the per-product inventory, each linked to its Flare execution

How we count, and what we refuse

Active user = signer of ≥1 validated `tesSUCCESS` transaction carrying the tag, multisig signers included, de-duplicated by hash. Our operational keys (executor, relayer, escrow keeper, notary, exchange omnibus) never carry the tag — our own code excludes self-attribution. We never ask anyone to transact "to support the project".

Why the numbers are small

On purpose. Until the external audit, capital is capped per cage and per account, in writing. What the numbers show is that the mechanism works on mainnet with real money. An account here is a person who signed a council ceremony or moved capital under rules.

WHAT ANYONE CAN CHECK IN FIVE MINUTES

Open the app, walk the three doors, then read the ledger.

1 · Log in
astryum.xyz/app with Xaman or XRP Identity — the wallet is the identity; nothing else is asked.

→

2 · The capital map
XRPL, Flare and EVM positions in one place; "your Astryum account" folded inside the wallet that governs it.

→

3 · Legacy
The councils, the constitution, the signing tray — and the four executed orders.

→

4 · Managed vaults
The manager's desk, the directory, the credential status read from the ledger; the exchange's front page.

→

5 · /proof + the inventory
Every claim verified in your browser; 176 transactions grouped by product, each linked to its Flare execution.

To sign something yourself

Any XRPL wallet works: put a small amount of XRP to work in Kinetic and watch the same memo instruction, attestation and execution happen for you — fees shown before the QR, receipt read from the chain after. Caps are small and written down.

To refute us

Read the anchors, the councils and the cages on xrpscan and flarescan with the addresses on the next slide. Nothing in this deck asks for trust: the quorum, the credential objects, the bridge nonces and the potes' shares are ledger state.

EVIDENCE · CHECK IT WITHOUT ASKING US

Every claim is on a ledger. /proof verifies it in your browser.

XRP Ledger (xrpscan)

Core Vault rfkXSaCZKTg1EZzec2rLDyrWHxRVJdtVXj
Legacy anchor rK4tsuGhmbhaNQuvucL8n1RKLtARBCp3qm
v2 anchor rLcoFM9XF8CL5GoFyMACguhdnDtwkNYDn7
Notary issuer rHKxjrGRrCegQhLrdnXEPAeGyeJ1JR4Hae

Flare, chain 14 (flarescan)

XrplCouncilBridge 0x02aE9fcB76768e42b8D3Ed9FE842238A6616b26F
LegacyVault 0xc8379c79779cCE3B738424892709fe0D4339E3b1
AstryumRegistry 0x18c807d107EF9d337aFA5BdD39b190C42Dc50Aec
AstryumCageFactory 0xE897fFef10F950Abc2d9c759107410F713e941D9
PasskeyAccountFactory 0xfa559cE04135835Cb8f7d3CFad2b8d14525ad932

Product · every transaction

astryum.xyz · /proof
176 mainnet transactions grouped by product, each linked to its Flare execution by the hash inside the FDC proof.

Code

github.com/iaserveraims
README + licence (audit yes, exploit no)

Tests

Foundry 168 · backend 4,275 · frontend 2,169 — green on 15 Sep. Tests prove the pieces; the ledger proves the chain.

WHAT WE DO NOT CLAIM

The honesty is the argument.

- No external audit of the contracts yet — so our own policy is written down: our own capital, small amounts, caps in configuration.
- The beta is fully open to everyone — with small capital caps, written down.
- No yield promises, ever. Rates are live protocol data with their source.
- No “licensed”, “regulated”, “audited”. In public: *a credential verified on the ledger.*
- No swap product, planner, fiat ramp, card, autonomous agent, protocol-managed wallet or confidential compute in code today — all of them on the roadmap.

Open blockers, in order

1. External audit — the number-one blocker; the main line of our grant applications.
2. The company wrapper — in progress; needed for provider contracts.
3. Key concentration: the venue registry’s governor is the treasury/executor key — moving to a multisig is first in the next wave.
4. The on-ledger credential door is built and not yet switched on.

Failures we already had, and what they built: “signing is not executing” (no screen turns green before settlement) · fail-safe retry classification (never a second signature on an unknown outcome) · what production shows lives in code, never in an environment variable.

NEXT, AND AFTER

One web, one kernel, N account types — and two things we wait for.

Next — the V2 plan NEXT

- **Wave 0** — fixes from our own audit, before anything new.
- **Wave 1** — the verification machine: one venue class end to end, a fork battery with a dated certificate, a card the explainer reads.
- **Wave 2** — the enterprise account: KYB provider, the anchor door on, the kit (API keys + remote MCP), goal compartments, the distribution fee paid by venues.
- **Wave 3** — composition: several actions per step, the worst case sealed in the signature.
- **In parallel** — contract v3 + external audit; the legal work that is not code.

After — on primitives not yet on mainnet LATER

- **Protocol-managed wallets + confidential compute** as the cage's remote arm: TEE k-of-n signers on native XRPL/BTC multisig, instructions only from the registered cage. Frame exists; the arm reverts on purpose. Isolated module, jurisdiction-gated.
- **The single root**: one XRPL account governing the family's, the company's and the reinforced accounts — one human key, N seats — each governing wallets on other chains.
- **The agent** as a sixth account type: revocable designation, its pote is its limit, the root is always human.
- **Fiat in and out; a card that cannot freeze your balance.**

HOW ASTRYUM EARNS

Whoever gains from the flow pays. Never the person who puts in the capital.

Distribution fee, paid by the venue

On the capital that reaches it from Astryum. The user deposits 100, the venue receives 100. Same rate for the whole class; never affects order or visibility; always disclosed. The principal source — decided, not built.

Fixed fees per act between companies

Creating a pote beyond the three free ones (1 FXRP, live in the contract), designating a manager, joining the directory. Collected by the code, never intercepted in the path of other people's money.

Transport at cost — later

Not charged today. When blockchain abstraction is fully adopted in Astryum — the canonical language doing the hard work, not the user — each XRPL → Flare order will be charged at cost, inside the payment the user signs and disclosed before signing. Built, switched off.

What is never charged

Nothing on shares or on the client's yield — the pote has no fee hook; zero enforced by absence. The manager's cut is capped at 20 % of yield by the cage. No prepaid balances, ever. No token.

Today

Nothing is charged. The machinery is wired and dormant until a public fee schedule, a disclosure row in the UI, a test that the ranking cannot see the fee, and a legal opinion exist.

ASTRYUM · 2026

Your capital.
Your control.
Your signature.

We are not asking you to trust us. The ledger does that work.

astryum.xyz · /proof · github.com/iaserveraims · Source Tag 2607090002 · Product decks: Legacy · Managed vaults · Exchange