

Astryum on mainnet, transaction by transaction

Every transaction Astryum's accounts signed on the XRP Ledger since July, followed by every transaction it caused on Flare — the attestation request, the execution, the agent's payout and its confirmation — grouped by product and in the order it happened, so each circuit can be followed on-chain, step by step.

XRPL transactions 204 Flare transactions 524 Circuits 112 Carrying the Source Tag 149 tagged tx · 16 accounts · 342.96 XRP

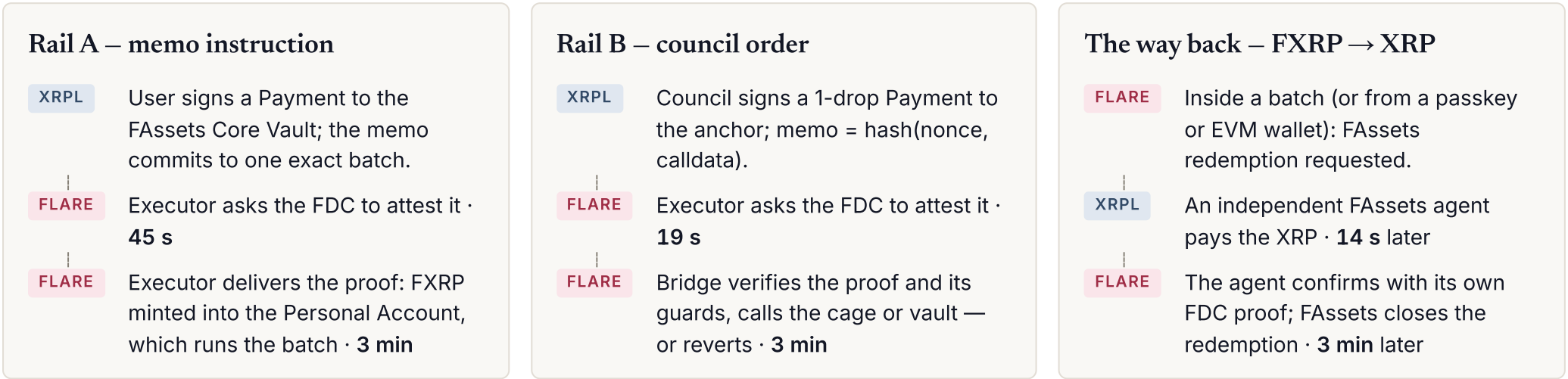
Metrics summary

MEASURE	READ FROM THE XRP LEDGER (19 SEPTEMBER 2026, 16:40 UTC)
Accounts that signed	16 since 14 Jul — multisig co-signers included: each member of a council signs with their own key
On-chain volume	342.96 XRP since 14 Jul
Transactions	149 validated XRPL transactions carrying the tag since 14 Jul

The Source Tag goes in the body of every transaction a user signs through Astryum, so anyone can recompute these figures from the ledger without asking us. An independent index of the same tag reported 7 accounts and 342.96 XRP (19 Sep 2026, 16:38 UTC): the volume matches our read to the cent, and the difference in accounts is method — it does not count multisig co-signers. Our operational keys never carry the tag and are excluded. Nobody was asked to transact “to support the project”. Flare transactions carry no tag — they are here because they are what the XRPL signatures did.

How a circuit reads

A circuit is one signature and everything it caused. It opens with the XRPL transaction the user or the council signed; the rows below it, slightly indented, are the transactions that followed on Flare — and, on the way back, on the XRP Ledger again. They are joined by what the chains carry, never by our word: the Flare execution holds the XRPL hash inside its FDC proof; the attestation request holds it in its request body; the agent's confirmation holds the hash of its XRPL payout. The small figure under each time is the delay from the signature. Medians measured on these circuits:



347 attestation requests appear below. 246 of them belong to three July payments that could never execute and were re-requested every hour until the 18 July guard — they are listed, not hidden, under their circuits. Rows marked **reverted** changed nothing; a reverted delivery is retried with the same proof, so the user never signs twice. Rows without a step number stand alone: ceremonies, credentials, deployments, funding.

Legacy — a family governs by quorum

Two councils exist on mainnet. Council #1 (rsmvJMhh..., 3-of-4, master key off since 15 July, constitution anchored 18 July) governs the first Legacy stack (bridge 0x02aE...b26F, vault 0xc837...E3b1, deployed and bound on 22 July) and has executed four orders end to end. Council #2 (rpM7wQNU..., 2-of-3, master off, constitution anchored 11 August) was born through LegacyStackFactory with one memo instruction — the factory only accepts the council's own Personal Account, so producing that payment *is* the quorum. What each circuit proves: the quorum is a fact of the XRP Ledger, the constitution cannot be rewritten quietly, and a contract on Flare only moves capital between the vault and listed venues when the council orders it — every order carries the constitution reference, and the bridge's nonce grows by exactly one per executed order. The two memo instructions of 14 July are the only ones that never executed: signed from the council account for a batch prepared for another Personal Account, they could not pass, and their repeated attestation requests are the incident that produced the executor's pay-once guard.

12 circuits · 24 XRPL transactions · 224 Flare transactions

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-14 17:25	XRPL	SignerListSet — SignerListSet — quorum 3 of 3 (no tag)	rsmvJMhhj...	The quorum becomes a fact of the network: from here on, this many signers must agree.
2026-07-14 17:50	XRPL	TicketCreate — TicketCreate — tickets for the asynchronous signing tray (no tag)	rsmvJMhhj... (multisig: 3 signers)	Tickets let each member sign their part asynchronously without blocking the account's sequence.
2026-07-14 20:11	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rsmvJMhhj...	Never executed: prepared for founder wallet A's Personal Account but signed from the council account, so the Personal Account can never accept it (InvalidSender). The attestation was re-requested hourly until the 18 July guard (below). Nothing moved on Flare; the transaction id is still unused in the MasterAccountController.
2026-07-14 20:12 +30 s	FLARE 2	↳ 0xfa24a51d... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR ► 101 requests, 2026-07-14 20:12 → 2026-07-18 09:10 · 2,020 FLR in fees	executor 0xd876...CecE (Jul-Aug key)	Re-requested 101 times between 07-14 20:12 and 07-18 09:10 — 2,020 FLR of attestation fees for a payment that could not execute. Fixed on 18 July: the executor now checks for free before paying, and pays one attestation per payment.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-14 20:14	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rsmvJMhhj...	Never executed: prepared for founder wallet A's Personal Account but signed from the council account, so the Personal Account can never accept it (InvalidSender). The attestation was re-requested hourly until the 18 July guard (below). Nothing moved on Flare; the transaction id is still unused in the MasterAccountController.
2026-07-14 20:15 +1 min 13 s	FLARE 2	↳ 0x38975a0d... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR ► 102 requests, 2026-07-14 20:15 → 2026-07-18 09:07 · 2,040 FLR in fees	executor 0xD876...CecE (Jul-Aug key)	Re-requested 102 times between 07-14 20:15 and 07-18 09:07 — 2,040 FLR of attestation fees for a payment that could not execute. Fixed on 18 July: the executor now checks for free before paying, and pays one attestation per payment.
2026-07-14 20:51	XRPL	EscrowCreate — EscrowCreate (multisig rehearsal) — every signer signs a real transaction before the door closes	rsmvJMhhj... (multisig: 3 signers)	Every signer signed a real multisig transaction before the door was closed — nobody locks a door they cannot open.
2026-07-15 19:58	XRPL	SignerListSet — SignerListSet — quorum 3 of 4 (no tag)	rsmvJMhhj... (multisig: 3 signers)	The quorum becomes a fact of the network: from here on, this many signers must agree.
2026-07-15 20:08	XRPL	EscrowCreate — EscrowCreate (multisig rehearsal) — every signer signs a real transaction before the door closes	rsmvJMhhj... (multisig: 3 signers)	Every signer signed a real multisig transaction before the door was closed — nobody locks a door they cannot open.
2026-07-15 21:39	XRPL	AccountSet — AccountSet asfDisableMaster — the original single key is switched off	rsmvJMhhj...	Irreversible without the quorum: the original single key can no longer move anything alone.
2026-07-18 14:04	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	rsmvJMhhj... (multisig: 3 signers)	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-07-22 18:03	FLARE	0xde33f4d1... · Contract deployment — XrpICouncilBridge #1 at 0x02ae...b26f	deployer 0x7D04...Ba8F	Public bytecode at a fixed address — the code the circuits in this document call.
2026-07-22 18:03	FLARE	0x98d0a892... · Contract deployment — LegacyVault #1 at 0xc837...e3b1	deployer 0x7D04...Ba8F	Public bytecode at a fixed address — the code the circuits in this document call.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-22 18:03	FLARE	0x5614c15e... · XrpIcouncilBridge.bind — the bridge is bound to the vault (one-shot, deployer only; the deployer keeps no power after it)	deployer 0x7D04...Ba8F	One-shot: after binding, the deployer has no power over the bridge or the vault.
2026-07-29 12:10	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 5 XRP	rsmvJMhhj... (multisig: 3 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-29 12:11 +39 s	FLARE 2	↳ 0xbf7649b4... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-29 12:13 +2 min 40 s	FLARE 3	↳ 0x7dd2f4ca... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rsmvJMhhj..., which runs the batch: FXRP.approve(LegacyVault #1, 4.6953 FXRP); LegacyVault #1.deposit(4695300)	executor 0xD876...CecE (Jul-Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-07-29 12:29	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rsmvJMhhj... (multisig: 3 signers)	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-07-29 12:43 +14 min	FLARE 2	↳ 0xd0283e70... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-29 12:46 +17 min	FLARE 3	↳ 0x1e7ca421... · XrpIcouncilBridge #1.execute — nonce 0 → directTo 4.6953 FXRP	executor 0xD876...CecE (Jul-Aug key)	The bridge verified the FDC proof and its guards and executed nonce 0. Capital directed to a registry-listed venue — the only kind of destination the cage accepts.
2026-08-01 21:29	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rsmvJMhhj... (multisig: 3 signers)	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-01 21:30 +13 s	FLARE 2	↳ 0xf45803e0... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-01 21:32 +2 min 49 s	FLARE 3	↳ 0x241c4efd... · XrpCouncilBridge #1.execute — nonce 1 → recall 4.6953 FXRP	executor 0xD876...CecE (Jul–Aug key)	The bridge verified the FDC proof and its guards and executed nonce 1. Capital recalled from the venue back into the vault.
2026-08-03 15:04	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rsmvJMhhj... (multisig: 3 signers)	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-08-03 17:38 +2 h 33 min	FLARE 2	↳ 0xab3420a4... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-03 17:40 +2 h 35 min	FLARE 3	↳ 0xfc77374a... · XrpCouncilBridge #1.execute — nonce 2 → directTo 0.1 FXRP	executor 0xD876...CecE (Jul–Aug key)	The bridge verified the FDC proof and its guards and executed nonce 2. Capital directed to a registry-listed venue — the only kind of destination the cage accepts.
2026-08-03 18:37	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rsmvJMhhj... (multisig: 3 signers)	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-08-03 18:37 +18 s	FLARE 2	↳ 0x85688efe... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-03 18:38 +1 min 37 s	FLARE 3	↳ 0x69687b74... · XrpCouncilBridge #1.execute — nonce 3 → recall 0.1 FXRP	executor 0xD876...CecE (Jul–Aug key)	The bridge verified the FDC proof and its guards and executed nonce 3. Capital recalled from the venue back into the vault.
2026-08-11 14:03	XRPL	SignerListSet — SignerListSet — quorum 2 of 3 (no tag)	rpM7wQNUm...	The quorum becomes a fact of the network: from here on, this many signers must agree.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-11 14:05	XRPL	EscrowCreate — EscrowCreate (multisig rehearsal) — every signer signs a real transaction before the door closes	rpM7wQNUm... (multisig: 3 signers)	Every signer signed a real multisig transaction before the door was closed — nobody locks a door they cannot open.
2026-08-11 14:06	XRPL	AccountSet — AccountSet asfDisableMaster — the original single key is switched off	rpM7wQNUm...	Irreversible without the quorum: the original single key can no longer move anything alone.
2026-08-11 14:13	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	rpM7wQNUm... (multisig: 3 signers)	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-08-11 14:24	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 4 XRP	rpM7wQNUm... (multisig: 3 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-11 14:25 +1 min 02 s	FLARE 2	↳ 0xba7a4f13... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-11 14:26 +2 min 27 s	FLARE 3	↳ 0xcaa6e342... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rpM7wQNUm..., which runs the batch: LegacyStackFactory.create(rpM7wQNUmAPSSb bb6J3yFJGnS9eoZthu8v); FXRP.approve(LegacyVault #2, 3.6963 FXRP); LegacyVault #2.deposit(3696300)	executor 0xD876...CecE (Jul–Aug key)	Birth of a stack by ONE signature: the factory accepts only the Personal Account of this root, so the payment itself is the authority; bridge (immutable council hash) and cage/vault are deployed and bound in this same transaction, with the first approve/deposit.
2026-08-23 17:06	XRPL	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	r1ypgoqtd... (multisig: 2 signers)	Council order signed on XRPL with no attestation and no execution: a duplicate submission, dropped before paying — the bridge only executes the exact next nonce.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-23 17:36	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	r1ypgoqtd... (multisig: 2 signers)	Order attested but never executed: superseded by a later signature of the same order, or refused before sending — the bridge only executes the exact next nonce.
2026-08-23 17:37 +22 s	FLARE 2	↳ 0xf9bac483... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-23 18:00	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	r1ypgoqtd... (multisig: 2 signers)	Order attested but never executed: superseded by a later signature of the same order, or refused before sending — the bridge only executes the exact next nonce.
2026-08-23 18:00 +17 s	FLARE 2	↳ 0x8764cf80... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR ► 2 requests, 2026-08-23 18:00 → 2026-08-23 18:35 · 40 FLR in fees	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash — requested 2 times before the delivery went through; every request is listed. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee.
2026-08-23 18:42	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	r1ypgoqtd... (multisig: 2 signers)	Order attested but never executed: superseded by a later signature of the same order, or refused before sending — the bridge only executes the exact next nonce.
2026-08-23 18:42 +18 s	FLARE 2	↳ 0xfdae1e16... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-23 19:17	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	r1ypgoqtd... (multisig: 2 signers)	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-08-23 19:18 +22 s	FLARE 2	↳ 0xa75bf92c... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-23 19:20 +3 min 07 s	FLARE 3	↳ 0xbb422126... · stack bridge of r1ypgoqtd....execute — nonce 0 → directTo 2 FXRP	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 0. Capital directed to a registry-listed venue — the only kind of destination the cage accepts.

Reinforced account — a personal wallet no single key can move

Account rP49LEKa... was turned into a 2-of-3 governed account on 21 August: SignerListSet, a multisig rehearsal (EscrowCreate signed by the quorum), asfDisableMaster, and a constitution anchored with DIDSet. It then acted as a normal user: it minted FXRP to Kinetic with one multisigned XRPL Payment (22 August — the full memo rail end to end), became a client of a managed vault (deposit 8 September, redeem request the same day, claim 12 September), moved FXRP by its own signature, and on 18 September took the way back: one multisigned Payment redeemed 14.46 FXRP, an independent agent paid 14.43 XRP to the account three minutes later and confirmed it on Flare. What it proves: the same ceremony that protects a family protects a person, and a governed account can use every product — including the exit to XRP — without any special path.

5 circuits · 12 XRPL transactions · 12 Flare transactions

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-21 16:18	XRPL	SignerListSet — SignerListSet — quorum 2 of 3 (no tag)	rP49LEKat...	The quorum becomes a fact of the network: from here on, this many signers must agree.
2026-08-21 16:20	XRPL	EscrowCreate — EscrowCreate (multisig rehearsal) — every signer signs a real transaction before the door closes	rP49LEKat... (multisig: 3 signers)	Every signer signed a real multisig transaction before the door was closed — nobody locks a door they cannot open.
2026-08-21 16:20	XRPL	AccountSet — AccountSet asfDisableMaster — the original single key is switched off	rP49LEKat...	Irreversible without the quorum: the original single key can no longer move anything alone.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-21 16:23	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	rP49LEKat... (multisig: 3 signers)	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-08-22 10:38	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 8 XRP	rP49LEKat... (multisig: 3 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-22 10:38 +36 s	FLARE 2	↳ 0x8e86184b... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR ► 3 requests, 2026-08-22 10:38 → 2026-08-22 10:39 · 60 FLR in fees	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash — requested 3 times before the delivery went through; every request is listed. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee.
2026-08-22 10:40 +1 min 41 s	FLARE 3	↳ 0xaffdbe72... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rP49LEKat..., which runs the batch: FXRP.approve(Kinetic isoFXRP, 7.6923 FXRP); Kinetic isoFXRP.mint (supply)	executor 0xD876...CecE (Jul-Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-08-22 14:29	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rP49LEKat... (multisig: 3 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-22 14:30 +49 s	FLARE 2	↳ 0xcd492d5c... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-22 14:32 +3 min 19 s	FLARE 3	↳ 0xe4606c03... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rP49LEKat..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw)	executor 0xD876...CecE (Jul-Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-05 09:44	XRPL	Payment — Tagged XRP transfer between Astryum-governed accounts · 2 XRP	rP49LEKat... (multisig: 2 signers)	Tagged transfer between governed accounts (capital allocation for a ceremony or a test).
2026-09-11 16:14	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rP49LEKat... (multisig: 2 signers)	Attested, but never executed: the executor did not send the execution (its checks refused the batch). No FXRP was minted for it — the transaction id is still unused in the MasterAccountController (read 19 Sep).
2026-09-11 16:17 +3 min 44 s	FLARE 2	↳ 0xc5b65c08... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-14 16:22	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rP49LEKat... (multisig: 2 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-14 16:27 +5 min 31 s	FLARE 2	↳ 0xae74c54a... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-14 16:29 +7 min 22 s	FLARE 3	↳ 0x4b222311... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rP49LEKat..., which runs the batch: FXRP.transfer(Personal Account of rNaFfVgbq..., 9.0000 FXRP)	executor 0xBC43...2821	FXRP moved by the account's own XRPL signature — the Personal Account only executes what the memo hash committed to.
2026-09-18 17:31	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rP49LEKat... (multisig: 2 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-18 17:32 +20 s	FLARE 2	↳ 0xb3b918e8... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-18 17:34 +2 min 33 s	FLARE 3	↳ 0x1b454c74... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rP49LEKat..., which runs the batch: AssetManagerFXRP.redeemAmount(14455525, rP49LEKattxJ9ppioRZVVRoZ7QeWvzYyuG)	executor 0xBC43...2821	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-09-18 17:34 +2 min 49 s	XRPL 4	↳ Payment — an independent FAssets agent pays 14.4266 XRP to rP49LEKat... · redemption #13373736	rU3KcE1fq... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-09-18 17:37 +5 min 30 s	FLARE 5	↳ 0xc15c8530... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #13373736 · 14.4526 XRP)	FAssets agent 0x6751...554b	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-09-18 17:44	XRPL	Payment — Tagged XRP transfer between Astryum-governed accounts · 13 XRP	rP49LEKat... (multisig: 2 signers)	Tagged transfer between governed accounts (capital allocation for a ceremony or a test).

Personal account — one XRPL signature, N actions on Flare

The founders' own wallets (rNaFfVgb..., rBGVzjE5..., rGb1VJkx...) and rNyrefqu... (before it became a manager root on 9 September) exercised the memo rail against every live venue: Kinetic (supply, withdraw, the carry that borrows USDT0 and repays it — with a SparkDEX swap-fill inside the same batch), Firelight (deposit, redeem request, claim), earnXRP and Monarq (deposit, instant redeem), and the way back to XRP (redeemAmount), whose circuit ends with an independent FAssets agent paying XRP on the XRP Ledger and confirming it on Flare. Two early paths are also here, in full: on 12 July the founder's EVM wallet acted as the executor (the very first delivery, into Firelight), and five exits to XRP in July were redeemed straight from that EVM wallet (the MetaMask path, no XRPL signature). On 18 September founder wallet A used the way back to fund the second exchange run (FXRP redeemed to its root and to its omnibus). What it proves: the user holds no FLR and no EVM key; the Personal Account is derived from the XRPL address; the executor delivers a proof it cannot alter; and a failed delivery is retried with the same proof — the user never signs twice.

53 circuits · 76 XRPL transactions · 167 Flare transactions

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-12 10:05	FLARE 1	0x557e75ff... · AssetManagerFXRP redemption requested directly from founder EVM wallet 0xEAbc...462c (MetaMask path — no XRPL signature, no Personal Account)	founder EVM wallet 0xEAbc...462c	The same exit without Astryum's rail: FXRP held in an EVM wallet is redeemed directly at FAssets; an independent agent pays the XRP on the XRP Ledger and confirms on Flare (next rows).
2026-07-12 10:06 +14 s	XRPL 2	↳ Payment — an independent FAssets agent pays 10.3807 XRP to rNaFfVgbq... · redemption #11145692	raDZBXEKs... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-07-12 10:08 +2 min 55 s	FLARE 3	↳ 0x2d7c56cf... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #11145692 · 10.4015 XRP)	FAssets agent 0xbf95...1384	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-07-12 16:17	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 5 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-12 18:46 +2 h 29 min	FLARE 2	↳ 0xc034d623... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	founder EVM wallet 0xEAbc...462c	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-12 18:47 +2 h 30 min	FLARE 3	↳ 0x64d3bdb7... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(Firelight stXRP, 4.6953 FXRP); Firelight stXRP.deposit(4695300, Personal Account of rNaFfVgbq...)	founder EVM wallet 0xEAbc...462c	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-07-12 20:11	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 5 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-12 21:31 +1 h 19 min	FLARE 2	↳ 0x8f51e7c9... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-12 21:34 +1 h 22 min	FLARE 3	↳ 0x16124670... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(earnXRP (Upshift), 4.6953 FXRP); earnXRP (Upshift).deposit	executor 0xD876...CecE (Jul–Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-07-12 20:21	XRPL	TrustSet — TrustSet (no tag)	rNaFfVgbq...	
2026-07-12 20:23	XRPL	TrustSet — TrustSet (no tag)	rNaFfVgbq...	
2026-07-12 20:23	XRPL	TrustSet — TrustSet (no tag)	rNaFfVgbq...	
2026-07-14 16:50	XRPL	SignerListSet — SignerListSet — quorum 3 of 3 (no tag)	rGb1VJkxm...	The quorum becomes a fact of the network: from here on, this many signers must agree.
2026-07-15 20:23	XRPL	EscrowFinish — EscrowFinish — permissionless settlement of an escrow (no tag)	rPw6JAdM9...	Scheduled payment: the ledger fixes destination and dates; finishing is permissionless.
2026-07-16 17:21	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-16 17:22 +14 s	FLARE 2	↳ 0x14c2cd9d... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-16 17:23 +1 min 50 s	FLARE 3	↳ 0xefc1a7ca... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: earnXRP (Upshift).instantRedeem	executor 0xD876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-07-16 17:22	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	Never executed: its twin 877430... (signed 1 min earlier for the same nonce) was delivered first, so this one can never pass (InvalidNonce). The attestation was re-requested until the 18 July guard (below). Nothing moved on Flare.
2026-07-16 17:24 +2 min 08 s	FLARE 2	↳ 0x9aa1b89a... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR ► 43 requests, 2026-07-16 17:24 → 2026-07-18 09:03 · 860 FLR in fees	executor 0xD876...CecE (Jul–Aug key)	Re-requested 43 times between 07-16 17:24 and 07-18 09:03 — 860 FLR of attestation fees for a payment that could not execute. Fixed on 18 July: the executor now checks for free before paying, and pays one attestation per payment.
2026-07-16 20:00	XRPL	EscrowFinish — EscrowFinish — permissionless settlement of an escrow (no tag)	rPw6JAdM9...	Scheduled payment: the ledger fixes destination and dates; finishing is permissionless.
2026-07-19 10:14	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-19 10:14 +25 s	FLARE 2	↳ 0xd5bc5b93... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-19 10:16 +2 min 18 s	FLARE 3	↳ 0x8f1e3080... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: Firelight stXRP.redeem(4694962, Personal Account of rNaFfVgbq...)	executor 0xD876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-20 15:16	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-20 15:21 +5 min 25 s	FLARE 2	↳ 0xb010792f... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-20 15:23 +7 min 40 s	FLARE 3	↳ 0xfb5120da... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: Firelight stXRP.claimWithdraw	executor 0xD876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-07-29 08:34	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 6 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-29 08:34 +20 s	FLARE 2	↳ 0x5c9339d3... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR ► 2 requests, 2026-07-29 08:34 → 2026-07-29 08:35 · 40 FLR in fees	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash — requested 2 times before the delivery went through; every request is listed. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee.
2026-07-29 08:37 +3 min 09 s	FLARE 3	↳ 0x5e831e78... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(Kinetic isoFXRP, 5.6943 FXRP); Kinetic isoFXRP.mint (supply)	executor 0xD876...CecE (Jul–Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-29 08:37 +3 min 11 s	FLARE 4	↳ 0xdfd13d35... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(Kinetic isoFXRP, 5.6943 FXRP); Kinetic isoFXRP.mint (supply) reverted	executor 0xD876...CecE (Jul–Aug key)	Delivery reverted on Flare — nothing moved. The executor retried with the same proof (next row): the user never signed twice.
2026-07-29 08:44	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-29 08:44 +47 s	FLARE 2	↳ 0x06aac9ea... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-29 08:46 +2 min 08 s	FLARE 3	↳ 0xf4b382c5... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: AssetManagerFXRP.redeemAmount(12202317, rNaFfVgbqWYotrR3Jz4RorirDUZgLfxxXt)	executor 0xD876...CecE (Jul–Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-07-29 08:46 +2 min 21 s	XRPL 4	↳ Payment — an independent FAssets agent pays 12.1779 XRP to rNaFfVgbq... · redemption #11683390	raDZBXEKs... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-07-29 08:49 +5 min 17 s	FLARE 5	↳ 0xfa2d0d12... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #11683390 · 12.2023 XRP)	FAssets agent 0xbf95...1384	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-07-29 10:25	FLARE 1	0x59128300... · AssetManagerFXRP redemption requested directly from founder EVM wallet 0xEAbc...462c (MetaMask path — no XRPL signature, no Personal Account)	founder EVM wallet 0xEAbc...462c	The same exit without Astryum's rail: FXRP held in an EVM wallet is redeemed directly at FAssets; an independent agent pays the XRP on the XRP Ledger and confirms on Flare (next rows).

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-29 10:25 +9 s	XRPL 2	↳ Payment — an independent FAssets agent pays 5.988 XRP to rNaFfVgbq... · redemption #11686788	raDZBXEKs... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-07-29 10:28 +2 min 31 s	FLARE 3	↳ 0x73663a3b... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #11686788 · 6 XRP)	FAssets agent 0xbf95...1384	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-07-29 10:40	FLARE 1	0x1c35b04a... · AssetManagerFXRP redemption requested directly from founder EVM wallet 0xEAbc...462c (MetaMask path — no XRPL signature, no Personal Account)	founder EVM wallet 0xEAbc...462c	The same exit without Astryum's rail: FXRP held in an EVM wallet is redeemed directly at FAssets; an independent agent pays the XRP on the XRP Ledger and confirms on Flare (next rows).
2026-07-29 10:40 +13 s	XRPL 2	↳ Payment — an independent FAssets agent pays 5.988 XRP to rNaFfVgbq... · redemption #11688454	raDZBXEKs... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-07-29 10:43 +2 min 38 s	FLARE 3	↳ 0xfecdeb31... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #11688454 · 6 XRP)	FAssets agent 0xbf95...1384	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-07-29 10:59	FLARE 1	0x38c761a3... · AssetManagerFXRP redemption requested directly from founder EVM wallet 0xEAbc...462c (MetaMask path — no XRPL signature, no Personal Account)	founder EVM wallet 0xEAbc...462c	The same exit without Astryum's rail: FXRP held in an EVM wallet is redeemed directly at FAssets; an independent agent pays the XRP on the XRP Ledger and confirms on Flare (next rows).
2026-07-29 10:59 +11 s	XRPL 2	↳ Payment — an independent FAssets agent pays 5.988 XRP to rNaFfVgbq... · redemption #11689596	raDZBXEKs... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-07-29 11:01 +1 min 54 s	FLARE 3	↳ 0xf7d9bace... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #11689596 · 6 XRP)	FAssets agent 0xbf95...1384	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-29 11:01	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-29 11:01 +21 s	FLARE 2	↳ 0x308628ea... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-29 11:04 +2 min 40 s	FLARE 3	↳ 0x730aeeb4... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw); AssetManagerFXRP.redeemAmount(6394301, rNaFfVgbqWYotrR3Jz4RorirDUZgLfxxXt)	executor 0xD876...CecE (Jul-Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-07-29 11:04 +2 min 50 s	XRPL 4	↳ Payment — an independent FAssets agent pays 6.38151 XRP to rNaFfVgbq... · redemption #11690430	raDZBXEKs... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-07-29 11:07 +5 min 33 s	FLARE 5	↳ 0xf7e9c429... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #11690430 · 6.3943 XRP)	FAssets agent 0xbf95...1384	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-07-29 12:06	XRPL	Payment — Tagged XRP transfer between Astryum-governed accounts · 6 XRP	rNaFfVgbq...	Tagged transfer between governed accounts (capital allocation for a ceremony or a test).
2026-07-29 22:12	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 49 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-29 22:12 +11 s	FLARE 2	↳ 0x41838ac4... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-29 22:14 +2 min 41 s	FLARE 3	↳ 0xf7d17022... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(Kinetic isoFXRP, 48.6513 FXRP); Kinetic isoFXRP.mint (supply); Kinetic comptroller.enterMarkets; Kinetic kUSDT0.borrow	executor 0xd876...CecE (Jul–Aug key)	The carry: supply FXRP to Kinetic, enter the market, borrow USDT0 — all in one Flare transaction authorized by one XRPL Payment.
2026-07-30 09:43	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 48.2038 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-30 09:43 +31 s	FLARE 2	↳ 0xf5cb15cb... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-30 09:46 +3 min 08 s	FLARE 3	↳ 0x64d0c0b5... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(Kinetic isoFXRP, 47.8559 FXRP); Kinetic isoFXRP.mint (supply)	executor 0xd876...CecE (Jul–Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-07-30 20:10	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-30 20:11 +48 s	FLARE 2	↳ 0x17314e42... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-30 20:13 +2 min 53 s	FLARE 3	↳ 0x851ffa9c... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw)	executor 0xD876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-07-31 06:35	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-31 08:08 +1 h 33 min	FLARE 2	↳ 0x72d3e4bf... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-31 08:10 +1 h 34 min	FLARE 3	↳ 0xfbf8bcda... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: AssetManagerFXRP.redeemAmount(10000000, rNaFfVgbqWYotrR3Jz4RorirDUZgLfxxXt)	executor 0xD876...CecE (Jul–Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-07-31 08:10 +1 h 34 min	XRPL 4	↳ Payment — an independent FAssets agent pays 9.98 XRP to rNaFfVgbq... · redemption #11762576	rU3KcE1fq... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-07-31 08:13 +1 h 37 min	FLARE 5	↳ 0x0f82bfc0... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #11762576 · 10 XRP)	FAssets agent 0x6751...554b	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-07-31 08:31	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 6 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-31 08:31 +21 s	FLARE 2	↳ 0xa94bf89d... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-31 08:34 +3 min 07 s	FLARE 3	↳ 0x9d18cbc6... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(Kinetic isoFXRP, 5.6943 FXRP); Kinetic isoFXRP.mint (supply); Kinetic comptroller.enterMarkets; Kinetic kUSDT0.borrow	executor 0xD876...CecE (Jul-Aug key)	The carry: supply FXRP to Kinetic, enter the market, borrow USDT0 — all in one Flare transaction authorized by one XRPL Payment.
2026-07-31 09:18	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-31 09:18 +21 s	FLARE 2	↳ 0x26247e8d... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-31 09:20 +2 min 28 s	FLARE 3	↳ 0xb27422f1... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: USDT0.approve(Kinetic kUSDT0, 1288251); Kinetic kUSDT0.repayBorrow	executor 0xD876...CecE (Jul-Aug key)	The loan repaid from the Personal Account, authorized by one XRPL Payment.
2026-07-31 09:21	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-31 09:21 +14 s	FLARE 2	↳ 0x52d96197... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-31 09:23 +2 min 07 s	FLARE 3	↳ 0xf40d4a8d... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(SparkDEX router, 0.0001 FXRP); SparkDEX router.exactOutputSingle (swap-fill); USDT0.approve(Kinetic kUSDT0, 112); Kinetic kUSDT0.repayBorrow	executor 0xD876...CecE (Jul–Aug key)	A swap-fill inside the same batch: SparkDEX buys the exact amount needed to repay, then the loan is repaid — all in one execution, one signature.
2026-07-31 10:12	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-07-31 10:13 +20 s	FLARE 2	↳ 0x3164cf64... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-07-31 10:14 +2 min 16 s	FLARE 3	↳ 0x6f52eda0... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw) reverted	executor 0xD876...CecE (Jul–Aug key)	Delivery reverted on Flare — nothing moved. The executor retried with the same proof (next row): the user never signed twice.
2026-07-31 10:20 +7 min 27 s	FLARE 4	↳ 0xb9e4fd4d... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw)	executor 0xD876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-31 10:14	FLARE 1	0x60fbcf75... · AssetManagerFXRP redemption requested directly from founder EVM wallet 0xEAbc...462c (MetaMask path — no XRPL signature, no Personal Account)	founder EVM wallet 0xEAbc...462c	The same exit without Astryum's rail: FXRP held in an EVM wallet is redeemed directly at FAssets; an independent agent pays the XRP on the XRP Ledger and confirms on Flare (next rows).
2026-07-31 10:14 +13 s	XRPL 2	↳ Payment — an independent FAssets agent pays 12.2569 XRP to rNaFfVgbq... · redemption #11767164	rU3KcE1fq... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-07-31 10:17 +3 min 10 s	FLARE 3	↳ 0x4b939a19... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #11767164 · 12.2814 XRP)	FAssets agent 0x6751...554b	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-08-01 09:51	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 6 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-01 09:52 +1 min 07 s	FLARE 2	↳ 0xb2524952... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-01 09:55 +3 min 27 s	FLARE 3	↳ 0x206c230d... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(Firelight stXRP, 5.6943 FXRP); Firelight stXRP.deposit(5694300, Personal Account of rNaFfVgbq...)	executor 0xD876...CecE (Jul-Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-08-01 10:02	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-01 10:02 +44 s	FLARE 2	↳ 0x619d273d... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-01 10:04 +2 min 14 s	FLARE 3	↳ 0xa03bed99... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: Firelight stXRP.redeem(5693890, Personal Account of rNaFfVgbq...)	executor 0xd876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-08-03 09:33	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-03 09:34 +1 min 08 s	FLARE 2	↳ 0x559ab52c... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-03 09:35 +2 min 33 s	FLARE 3	↳ 0xd85526e9... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: Firelight stXRP.claimWithdraw	executor 0xd876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-08-03 14:40	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-03 14:41 +56 s	FLARE 2	↳ 0x56305e70... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-03 14:43 +2 min 48 s	FLARE 3	↳ 0x62d6ddc7... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(SparkDEX router, 0.0215 FXRP); SparkDEX router.exactOutputSingle (swap-fill); USDT0.approve(Kinetic kUSDT0, 11006181); Kinetic kUSDT0.repayBorrow	executor 0xD876...CecE (Jul– Aug key)	A swap-fill inside the same batch: SparkDEX buys the exact amount needed to repay, then the loan is repaid — all in one execution, one signature.
2026-08-03 15:13	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-03 15:14 +1 min 03 s	FLARE 2	↳ 0xd9761f79... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul– Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-03 15:16 +2 min 39 s	FLARE 3	↳ 0x46895275... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw); FXRP.transfer(0xDFABF0a7..., 48.6518 FXRP) reverted	executor 0xD876...CecE (Jul– Aug key)	Delivery reverted on Flare — nothing moved. The executor retried with the same proof (next row): the user never signed twice.
2026-08-03 15:21 +8 min 12 s	FLARE 4	↳ 0x64ec0fdc... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw); FXRP.transfer(0xDFABF0a7..., 48.6518 FXRP)	executor 0xD876...CecE (Jul– Aug key)	FXRP moved by the account's own XRPL signature — the Personal Account only executes what the memo hash committed to.
2026-08-04 17:22	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-04 17:23 +1 min 01 s	FLARE 2	↳ 0xd2e9f5bb... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-04 17:25 +2 min 52 s	FLARE 3	↳ 0xa12e4ac2... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw)	executor 0xd876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-08-08 16:39	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 5.98493 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-08 16:40 +49 s	FLARE 2	↳ 0x3d201ee1... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-08 16:41 +2 min 16 s	FLARE 3	↳ 0x980c8029... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(earnXRP (Upshift), 5.6792 FXRP); earnXRP (Upshift).deposit	executor 0xd876...CecE (Jul–Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-08-08 16:45	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-08 16:46 +27 s	FLARE 2	↳ 0x2e2802c9... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-08 16:47 +1 min 47 s	FLARE 3	↳ 0x47e5ca60... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: earnXRP (Upshift).instantRedeem	executor 0xD876...CecE (Jul–Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-08-08 17:13	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-08 17:13 +32 s	FLARE 2	↳ 0xb99fd5bb... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-08 17:16 +3 min 07 s	FLARE 3	↳ 0x2aa19460... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: AssetManagerFXRP.redeemAmount(8506678, rNyrefquhQfqFYwHojT8aHwVPmKtqYZtg8)	executor 0xD876...CecE (Jul–Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-08-08 17:16 +3 min 19 s	XRPL 4	↳ Payment — an independent FAssets agent pays 8.48967 XRP to rNyrefquh... · redemption #12037254	rs6q5K4Rr... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-08-08 17:19 +6 min 12 s	FLARE 5	↳ 0x84936b1e... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #12037254 · 8.50668 XRP)	FAssets agent 0xc7d6...8e19	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-08-08 18:40	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 8.4497 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-08 18:46 +5 min 29 s	FLARE 2	↳ 0x5399ee6f... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-08 18:47 +7 min 04 s	FLARE 3	↳ 0x8449ce32... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(earnXRP (Upshift), 8.1415 FXRP); earnXRP (Upshift).deposit	executor 0xD876...CecE (Jul-Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-08-08 18:48	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-08 18:49 +1 min 07 s	FLARE 2	↳ 0x5b4337b0... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-08 18:52 +3 min 37 s	FLARE 3	↳ 0xacbb839c... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: earnXRP (Upshift).instantRedeem	executor 0xD876...CecE (Jul-Aug key)	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-08-12 11:21	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-12 11:21 +10 s	FLARE 2	↳ 0xa554d8db... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-12 11:23 +2 min 01 s	FLARE 3	↳ 0x4d91de7a... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: AssetManagerFXRP.redeemAmount(9541554, rNyrefquhQfqFYwHojT8aHwVPmKtqYZtg8)	executor 0xd876...CecE (Jul–Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-08-12 11:23 +2 min 12 s	XRPL 4	↳ Payment — an independent FAssets agent pays 9.52247 XRP to rNyrefquh... · redemption #12140708	raDZBXEk... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-08-12 11:25 +3 min 46 s	FLARE 5	↳ 0xa4c492dd... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #12140708 · 9.54155 XRP)	FAssets agent 0xbf95...1384	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-08-12 11:25	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 8.47017 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-12 11:26 +1 min 07 s	FLARE 2	↳ 0xc5b60d62... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xd876...CecE (Jul–Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-12 11:28 +2 min 37 s	FLARE 3	↳ 0x2c8372ac... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(Kinetic isoFXRP, 8.1620 FXRP); Kinetic isoFXRP.mint (supply); Kinetic comptroller.enterMarkets; Kinetic kUSDT0.borrow	executor 0xd876...CecE (Jul–Aug key)	The carry: supply FXRP to Kinetic, enter the market, borrow USDT0 — all in one Flare transaction authorized by one XRPL Payment.
2026-08-16 16:52	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.4 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-16 16:53 +1 min 07 s	FLARE 2	↳ 0x7753d4ff... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-16 16:55 +2 min 42 s	FLARE 3	↳ 0x7cf4b90b... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(earnXRP (Upshift), 10.0999 FXRP); earnXRP (Upshift).deposit	executor 0xD876...CecE (Jul-Aug key)	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-08-21 18:08	XRPL	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	Not delivered: a duplicate of the neighbouring instruction, refused by the executor's free pre-check before paying any attestation. No FXRP was minted for it — the transaction id is still unused in the MasterAccountController (read 19 Sep).
2026-08-21 18:09	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-21 19:01 +52 min	FLARE 2	↳ 0x05414b10... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-21 19:04 +55 min	FLARE 3	↳ 0x5739712e... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: AssetManagerFXRP.redeemAmount(10000000, rP49LEKattxJ9ppioRZVVRoZ7QeWvzYyuG)	executor 0xD876...CecE (Jul-Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-08-21 19:04 +55 min	XRPL 4	↳ Payment — an independent FAssets agent pays 9.98 XRP to rP49LEKat... · redemption #12461498	rU3KcE1fq... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-21 19:07 +58 min	FLARE 5	↳ 0x7ac56a37... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #12461498 · 10 XRP)	FAssets agent 0x6751...554b	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-08-22 16:18	XRPL	EscrowFinish — EscrowFinish — permissionless settlement of an escrow (no tag)	rPw6JAdM9...	Scheduled payment: the ledger fixes destination and dates; finishing is permissionless.
2026-08-23 08:04	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-23 08:09 +5 min 28 s	FLARE 2	↳ 0xe6afa5d7... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-23 08:11 +7 min 28 s	FLARE 3	↳ 0x140661cc... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: AssetManagerFXRP.redeemAmount(10000000, rGb1VJkxmMzZLbWRFfu9Xh5GVjZGYR4d3x)	executor 0xD876...CecE (Jul-Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-08-23 08:12 +7 min 40 s	XRPL 4	↳ Payment — an independent FAssets agent pays 9.98 XRP to rGb1VJkxm... · redemption #12568254	rEUvL6uJ1... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-08-23 08:14 +10 min	FLARE 5	↳ 0x997ce0d5... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #12568254 · 10 XRP)	FAssets agent 0x47e1...ea1f	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-08-23 12:41	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-23 12:42 +52 s	FLARE 2	↳ 0x750c7626... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-23 12:44 +3 min 26 s	FLARE 3	↳ 0x70824da4... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: AssetManagerFXRP.redeemAmount(5000000, rHKxjrGRrCegQhLrdnXEPaGyeJ1JR4Hae)	executor 0xD876...CecE (Jul-Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-08-23 12:45 +3 min 40 s	XRPL 4	↳ Payment — an independent FAssets agent pays 4.99 XRP to rHKxjrGRr... · redemption #12574364	rEUvL6uJ1... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-08-23 12:47 +6 min 22 s	FLARE 5	↳ 0xb64ab4c7... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #12574364 · 5 XRP)	FAssets agent 0x47e1...ea1f	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-08-23 12:52	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-23 12:52 +14 s	FLARE 2	↳ 0xb8202c7e... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xD876...CecE (Jul-Aug key)	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-23 12:55 +2 min 36 s	FLARE 3	↳ 0x8d2b878b... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: AssetManagerFXRP.redeemAmount(20000000, r1ypgoqtdQG71bMK7g2KdjReekZH1MuoG)	executor 0xD876...CecE (Jul-Aug key)	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-23 12:55 +2 min 50 s	XRPL 4	↳ Payment — an independent FAssets agent pays 19.96 XRP to r1ypgoqtd... · redemption #12575066	rEUvL6uJ1... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-08-23 12:58 +5 min 31 s	FLARE 5	↳ 0x4c0d31da... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #12575066 · 20 XRP)	FAssets agent 0x47e1...ea1f	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-08-24 13:51	XRPL	EscrowFinish — EscrowFinish — permissionless settlement of an escrow (no tag)	rPw6JAdM9...	Scheduled payment: the ledger fixes destination and dates; finishing is permissionless.
2026-08-25 08:16	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-25 08:17 +51 s	FLARE 2	↳ 0x945b0445... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-25 08:19 +2 min 56 s	FLARE 3	↳ 0x9b4dd0d6... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(SparkDEX router, 0.0018 FXRP); SparkDEX router.exactOutputSingle (swap-fill); USDT0.approve(Kinetic kUSDT0, 1757866); Kinetic kUSDT0.repayBorrow	executor 0xBC43...2821	A swap-fill inside the same batch: SparkDEX buys the exact amount needed to repay, then the loan is repaid — all in one execution, one signature.
2026-08-26 16:25	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-26 16:26 +1 min 00 s	FLARE 2	↳ 0xb22ec53c... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-26 16:28 +3 min 13 s	FLARE 3	↳ 0x106d6a7e... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: Kinetic isoFXRP.redeemUnderlying (withdraw)	executor 0xBC43...2821	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-08-28 15:07	FLARE	0x361291a1... · 10.00 FLR sent to the Personal Account of rNaFfVgbq... (plain transfer; not needed for any circuit — the user's rail never spends FLR)	founder EVM wallet 0xEAbc...462c	Not needed by any circuit: the Personal Account pays nothing — the executor carries the gas.
2026-09-05 10:24	FLARE	0x50588b3d... · 300.00 FLR sent to the Personal Account of rNaFfVgbq... (plain transfer; not needed for any circuit — the user's rail never spends FLR)	founder EVM wallet 0xc8be...e23f	Not needed by any circuit: the Personal Account pays nothing — the executor carries the gas.
2026-09-05 10:31	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-05 10:32 +34 s	FLARE 2	↳ 0xb06b34fa... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-05 10:34 +2 min 34 s	FLARE 3	↳ 0x0d7ca294... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: founder EVM wallet 0xEAbc...462c ← send 310 FLR	executor 0xBC43...2821	A plain value transfer executed from the Personal Account, exactly as the memo hash committed.
2026-09-08 14:35	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-08 14:35 +31 s	FLARE 2	↳ 0x4914d6f7... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-08 14:37 +2 min 16 s	FLARE 3	↳ 0xe630458f... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: earnXRP (Upshift).instantRedeem	executor 0xBC43...2821	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-09-09 17:59	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-09 18:00 +56 s	FLARE 2	↳ 0x7c57768e... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-09 18:02 +3 min 17 s	FLARE 3	↳ 0x03b4cf01... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(earnXRP (Upshift), 20.3323 FXRP); earnXRP (Upshift).deposit	executor 0xBC43...2821	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-09-09 18:25	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-09 18:25 +42 s	FLARE 2	↳ 0x40e7b607... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-09 18:28 +3 min 07 s	FLARE 3	↳ 0x867d1d4b... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: earnXRP (Upshift).instantRedeem reverted	executor 0xBC43...2821	Delivery reverted on Flare — nothing moved. The executor retried with the same proof (next row): the user never signed twice.
2026-09-09 18:33 +8 min 32 s	FLARE 4	↳ 0x80a955d7... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: earnXRP (Upshift).instantRedeem	executor 0xBC43...2821	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-09-09 18:44	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-09 18:45 +42 s	FLARE 2	↳ 0x243ffe60... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-09 18:47 +2 min 46 s	FLARE 3	↳ 0x8e875b41... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(earnXRP (Upshift), 20.4264 FXRP); earnXRP (Upshift).deposit	executor 0xBC43...2821	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-09-10 14:11	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-10 14:12 +40 s	FLARE 2	↳ 0x799b8fea... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-10 14:14 +3 min 10 s	FLARE 3	↳ 0x13c62137... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: earnXRP (Upshift).instantRedeem	executor 0xBC43...2821	Unwinding a position with one XRPL signature: the venue returns FXRP to the Personal Account.
2026-09-10 14:26	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-10 14:27 +1 min 04 s	FLARE 2	↳ 0xfd734066... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-10 14:29 +3 min 20 s	FLARE 3	↳ 0x1454e902... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(Monarq (Upshift), 8.9561 FXRP); Monarq (Upshift).deposit	executor 0xBC43...2821	FAssets minted FXRP into the user's Personal Account and, in the same transaction, the account supplied it to the venue. The executor carried a proof it cannot alter; the user holds no FLR and no EVM key.
2026-09-14 20:47	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-14 20:48 +44 s	FLARE 2	↳ 0xdfea45bf... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-14 20:50 +3 min 14 s	FLARE 3	↳ 0x90ce4f10... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.transfer(Personal Account of rBGVzjE5J..., 9.0000 FXRP)	executor 0xBC43...2821	FXRP moved by the account's own XRPL signature — the Personal Account only executes what the memo hash committed to.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-14 21:12	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rBGVzjE5J...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-14 21:12 +25 s	FLARE 2	↳ 0x22049690... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-14 21:14 +2 min 37 s	FLARE 3	↳ 0x65b8e471... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rBGVzjE5J..., which runs the batch: AssetManagerFXRP.redeemAmount(9000000, rNaFfVgbqWYotrR3Jz4RorirDUZgLfXhXt)	executor 0xBC43...2821	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-09-14 21:15 +2 min 49 s	XRPL 4	↳ Payment — an independent FAssets agent pays 8.982 XRP to rNaFfVgbq... · redemption #13245074	rU3KcE1fq... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-09-14 21:17 +5 min 38 s	FLARE 5	↳ 0xf6cee116... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #13245074 · 8.9982 XRP)	FAssets agent 0x6751...554b	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-09-15 12:51	FLARE	0x6ce37aa6... · 301,000.00 FLR sent to the Personal Account of rNaFfVgbq... (plain transfer; not needed for any circuit — the user's rail never spends FLR)	0xa03c5fbb...	Not needed by any circuit: the Personal Account pays nothing — the executor carries the gas.
2026-09-18 12:58	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-18 12:59 +1 min 13 s	FLARE 2	↳ 0x37aed905... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-18 13:01 +2 min 33 s	FLARE 3	↳ 0x4d3f3cb7... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: AssetManagerFXRP.redeemAmount(5000000, rEvD7wxmbwng8skmqNu8R6wAUQQVVXXACv)	executor 0xBC43...2821	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.
2026-09-18 13:01 +2 min 49 s	XRPL 4	↳ Payment — an independent FAssets agent pays 4.99 XRP to rEvD7wxmb... · redemption #13362366	rU3KcE1fq... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-09-18 13:04 +5 min 41 s	FLARE 5	↳ 0x6e974ac1... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #13362366 · 4.999 XRP)	FAssets agent 0x6751...554b	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-09-18 13:00	XRPL	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	Not delivered: a duplicate of the neighbouring instruction, refused by the executor's free pre-check before paying any attestation. No FXRP was minted for it — the transaction id is still unused in the MasterAccountController (read 19 Sep).
2026-09-18 13:37	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-18 13:38 +40 s	FLARE 2	↳ 0x41067b00... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-18 13:40 +2 min 45 s	FLARE 3	↳ 0xca895d61... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: AssetManagerFXRP.redeemAmount(5553867, r4yp47QPbB3XcVurs8pbE75k7EwQ1GEBAA)	executor 0xBC43...2821	The way back starts here: the Personal Account asks FAssets to redeem FXRP for XRP (RedemptionRequested). An independent agent must now pay on the XRP Ledger — see the next rows. Authorized by the XRPL signature alone.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-18 13:40 +2 min 59 s	XRPL 4	↳ Payment — an independent FAssets agent pays 5.54276 XRP to r4yp47QPb... · redemption #13363524	rU3KcE1fq... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-09-18 13:43 +5 min 40 s	FLARE 5	↳ 0x8b3e4890... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #13363524 · 5.55276 XRP)	FAssets agent 0x6751...554b	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.

Institutional pote A (v1) — the first cage on the stack factory

Council r1ypgoqt... (2-of-3, master off, constitution anchored, 23 August) created the first institutional stack with one memo instruction (AstryumStackFactory.create + approve + deposit, all in the same Flare transaction), deposited again, redeemed, and then ordered directTo through its own bridge. The one **refused** row matters most: an EVM wallet tried to call directTo directly on the pote and the contract reverted — only the council's bridge can order it. The client-side passkey flow was rehearsed here for the first time: a P-256 account was created on 23 August and, signing on the device, redeemed its FXRP to XRP — the agent paid 7.64 XRP to the council account and confirmed on Flare.

5 circuits · 11 XRPL transactions · 13 Flare transactions

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-23 13:50	XRPL	SignerListSet — SignerListSet — quorum 2 of 3 (no tag)	r1ypgoqtd...	The quorum becomes a fact of the network: from here on, this many signers must agree.
2026-08-23 13:52	XRPL	EscrowCreate — EscrowCreate (multisig rehearsal) — every signer signs a real transaction before the door closes	r1ypgoqtd... (multisig: 3 signers)	Every signer signed a real multisig transaction before the door was closed — nobody locks a door they cannot open.
2026-08-23 13:53	XRPL	AccountSet — AccountSet asfDisableMaster — the original single key is switched off	r1ypgoqtd...	Irreversible without the quorum: the original single key can no longer move anything alone.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-23 13:55	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	r1ypgoqtd... (multisig: 3 signers)	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-08-23 13:57	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	r1ypgoqtd... (multisig: 3 signers)	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-08-23 13:58	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 5 XRP	r1ypgoqtd... (multisig: 3 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-23 13:59 +1 min 01 s	FLARE 2	↳ 0xa75f9fe1... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-23 14:01 +3 min 01 s	FLARE 3	↳ 0x9f3a4c71... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of r1ypgoqtd..., which runs the batch: AstryumStackFactory.create(r1ypgoqtdQG71bMK7g2KdjReekZH1MuoG); FXRP.approve(pote A (v1), 4.6953 FXRP); pote A (v1).deposit(4695300, Personal Account of r1ypgoqtd...)	executor 0xBC43...2821	Birth of a stack by ONE signature: the factory accepts only the Personal Account of this root, so the payment itself is the authority; bridge (immutable council hash) and cage/vault are deployed and bound in this same transaction, with the first approve/deposit.
2026-08-23 15:17	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 8 XRP	r1ypgoqtd... (multisig: 3 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-23 15:23 +5 min 21 s	FLARE 2	↳ 0x86b0a875... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-23 15:25 +7 min 37 s	FLARE 3	↳ 0x7401fed2... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of r1ypgoqtd..., which runs the batch: FXRP.approve(pote A (v1), 7.6923 FXRP); pote A (v1).deposit(7692300, passkey account 0x4011...33C8)	executor 0xBC43...2821	A client deposits into a managed pote with their own XRPL signature: shares are minted to the client's account, never to the manager; no approve to the operator exists anywhere in the batch.
2026-08-23 15:30	FLARE	0xfeab6ca... · pote A.directTo called directly by an EOA — REVERTED: only the council (bridge) may order the pote reverted	founder EVM wallet 0xEAbc... 462c	The pote rejects every caller except the council's bridge — even an EVM wallet of the founders. The revert is the proof.
2026-08-23 16:26	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 1 XRP	r1ypgoqtd... (multisig: 2 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-08-23 16:27 +21 s	FLARE 2	↳ 0x0c42b7c9... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-08-23 16:29 +2 min 58 s	FLARE 3	↳ 0x37947c06... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of r1ypgoqtd..., which runs the batch: pote A (v1).redeem(4695300000, Personal Account of r1ypgoqtd...)	executor 0xBC43...2821	Instant exit from a pote with no cooldown: shares burned, FXRP back to the holder — owner-only, no manager in the path.
2026-08-23 19:52	FLARE	0xf093954e... · PasskeyAccountFactory.create — a P-256 (Face ID) account deployed for a client; the relayer only pays gas	executor 0xBC43...2821 (as relayer)	The client's account is a contract controlled by a P- 256 passkey (Face ID) on their device; the relayer pays gas and cannot sign for it.
2026-08-23 19:52	FLARE	0xdb0ccba0... · PasskeyAccount.executeBatch on passkey account 0x4011...33C8 — the client signs with their passkey; the relayer only pays gas	executor 0xBC43...2821 (as relayer)	Signed by the client's passkey on their device; the relayer only carries it.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-23 20:05	FLARE 1	0x29199961... · PasskeyAccount.executeBatch on passkey account 0x4011...33C8 — the client signs with their passkey; the relayer only pays gas	executor 0xBC43...2821 (as relayer)	Signed by the client's passkey on their device; the relayer only carries it. Here the batch requested a redemption to XRP — the agent's payout and confirmation follow.
2026-08-23 20:06 +12 s	XRPL 2	↳ Payment — an independent FAssets agent pays 7.63849 XRP to r1ypgoqtd... · redemption #12591706	r6K7w4oK5... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it.
2026-08-23 20:08 +2 min 37 s	FLARE 3	↳ 0x5a45f181... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #12591706 · 7.6538 XRP)	FAssets agent 0xb2ba...0c53	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.
2026-09-12 19:55	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	r1ypgoqtd... (multisig: 2 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-12 20:01 +5 min 32 s	FLARE 2	↳ 0xdf0310ed... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-12 20:02 +7 min 13 s	FLARE 3	↳ 0x1dc3bc3f... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of r1ypgoqtd..., which runs the batch: FXRP.transfer(Personal Account of rP49LEKat..., 5.4077 FXRP)	executor 0xBC43...2821	FXRP moved by the account's own XRPL signature — the Personal Account only executes what the memo hash committed to.
2026-09-12 19:57	XRPL	Payment — Tagged XRP transfer between Astryum-governed accounts · 10.8479 XRP	r1ypgoqtd... (multisig: 2 signers)	Tagged transfer between governed accounts (capital allocation for a ceremony or a test).

Managed vaults (cage v2) — a manager with a credential directs inside a cage

Three manager roots (rDcohNUm... 5–8 September, rNyrefqu... 10–11 September, raqZZbqW... 15 September) went through the same sequence: credentials accepted on the root → cage born by memo instruction through AstryumCageFactory (the factory accepts only the root's Personal Account) → potes created by council order → capital directed to registry venues by council order. Clients (rP49LEKa..., rNaFfVgb...) deposited into the first pote with their own signatures, requested redemption and claimed. A demo vault (raqZZbqW..., no cooldown) was then used end to end: rNyrefqu... deposited and redeemed twice (15 and 17 September) and the root directed 4.23 FXRP to a registry venue by council order (17 September). The registry rows show the two venues (Kinetic isoFXRP, Firelight stXRP) proposed and activated after the timelock. What it proves: without valid credentials on the root the cage is not born; the manager can only direct and recall; the client's exit is owner-only; the venue list is a governed contract.

20 circuits · 26 XRPL transactions · 44 Flare transactions

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-06 08:32	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	rDcohNUmB...	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-09-06 18:14	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.5 XRP	rDcohNUmB...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-06 18:15 +56 s	FLARE 2	↳ 0x58dff14a... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-06 18:17 +3 min 27 s	FLARE 3	↳ 0xf3cd61fa... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rDcohNUmB..., which runs the batch: AstryumCageFactory.create(rDcohNUmBApE8b SPP695GkUsSMfBeM5Czj); FXRP.approve(cage of rDcohNUmB..., 0.1998 FXRP)	executor 0xBC43...2821	Birth of a stack by ONE signature: the factory accepts only the Personal Account of this root, so the payment itself is the authority; bridge (immutable council hash) and cage/vault are deployed and bound in this same transaction, with the first approve/deposit.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-06 18:51	FLARE	0x9bf76e23... · AstryumRegistry.proposeVenue — Kinetic isoFXRP	executor 0xBC43...2821	The venue list is a governed contract: a venue is proposed, waits the timelock, then is activated. A cage can only send capital to active venues.
2026-09-06 18:51	FLARE	0x4d29558d... · AstryumRegistry.proposeVenue — Firelight stXRP	executor 0xBC43...2821	The venue list is a governed contract: a venue is proposed, waits the timelock, then is activated. A cage can only send capital to active venues.
2026-09-06 20:12	FLARE	0x69fbbaf4... · AstryumRegistry.activateVenue — Kinetic isoFXRP	executor 0xBC43...2821	The venue list is a governed contract: a venue is proposed, waits the timelock, then is activated. A cage can only send capital to active venues.
2026-09-06 20:12	FLARE	0xd4ee5e3a... · AstryumRegistry.activateVenue — Firelight stXRP	executor 0xBC43...2821	The venue list is a governed contract: a venue is proposed, waits the timelock, then is activated. A cage can only send capital to active venues.
2026-09-06 20:32	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rDcohNUmB...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-06 20:32 +13 s	FLARE 2	↳ 0xf7727deb... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-06 20:34 +1 min 49 s	FLARE 3	↳ 0x81d698d8... · bridge v2 of rDcohNUmB....execute — nonce 0 → createPote “Easy FXRP Yield” · cooldown 12h · payee cap 1000 bps	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 0. A pote was created inside the cage (name, cooldown, payee cap): the manager can only create inside the rules of the cage.
2026-09-08 16:04	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rP49LEKat... (multisig: 3 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-08 16:04 +13 s	FLARE 2	↳ 0x4e182edc... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-08 16:07 +2 min 47 s	FLARE 3	↳ 0x94f6f068... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rP49LEKat..., which runs the batch: FXRP.approve(pote in cage of rDcohNUmB..., 15.0500 FXRP); pote in cage of rDcohNUmB....deposit(15049950, Personal Account of rP49LEKat...)	executor 0xBC43...2821	A client deposits into a managed pote with their own XRPL signature: shares are minted to the client's account, never to the manager; no approve to the operator exists anywhere in the batch.
2026-09-08 16:46	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 10.5 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-08 16:48 +2 min 38 s	FLARE 2	↳ 0x53c9ce52... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-08 16:50 +4 min 23 s	FLARE 3	↳ 0x4802ca58... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: FXRP.approve(pote in cage of rDcohNUmB..., 10.1898 FXRP); pote in cage of rDcohNUmB....deposit(10189800, Personal Account of rNaFfVgbq...)	executor 0xBC43...2821	A client deposits into a managed pote with their own XRPL signature: shares are minted to the client's account, never to the manager; no approve to the operator exists anywhere in the batch.
2026-09-08 18:09	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rDcohNUmB...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-08 18:09 +18 s	FLARE 2	↳ 0x806fdab9... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-08 18:11 +2 min 48 s	FLARE 3	↳ 0xaf07d395... · bridge v2 of rDcohNUMB....execute — nonce 1 → directTo 11 FXRP → pote in cage of rDcohNUMB...	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 1. Capital directed to a registry-listed venue — the only kind of destination the cage accepts.
2026-09-08 18:15	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rDcohNUMB...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-08 18:15 +15 s	FLARE 2	↳ 0x87974e09... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-08 18:17 +2 min 26 s	FLARE 3	↳ 0x373f2db0... · bridge v2 of rDcohNUMB....execute — nonce 2 → directTo 11.7158 FXRP → pote in cage of rDcohNUMB...	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 2. Capital directed to a registry-listed venue — the only kind of destination the cage accepts.
2026-09-08 18:23	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-08 18:24 +59 s	FLARE 2	↳ 0x39977008... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-08 18:26 +3 min 48 s	FLARE 3	↳ 0x64232661... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: pote in cage of rDcohNUMB....requestRedeem(10189800000, Personal Account of rNaFfVgbq...) reverted	executor 0xBC43...2821	Delivery reverted on Flare — nothing moved. The executor retried with the same proof (next row): the user never signed twice.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-08 18:32 +9 min 01 s	FLARE 4	↳ 0x5986d645... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: pote in cage of rDcohNUmB....requestRedeem(10189800000, Personal Account of rNaFfVgbq...)	executor 0xBC43...2821	Client exit, step 1: requestRedeem is owner-only in the pote — no manager, credential or Astryum key can gate it.
2026-09-08 18:24	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rP49LEKat... (multisig: 2 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-08 18:27 +3 min 26 s	FLARE 2	↳ 0x6b0d1a9b... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-08 18:29 +5 min 21 s	FLARE 3	↳ 0x785159a0... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rP49LEKat..., which runs the batch: pote in cage of rDcohNUmB....requestRedeem(15049950000, Personal Account of rP49LEKat...)	executor 0xBC43...2821	Client exit, step 1: requestRedeem is owner-only in the pote — no manager, credential or Astryum key can gate it.
2026-09-10 20:40	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	rNyrefquh...	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-09-10 20:52	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	rNyrefquh...	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-09-10 21:14	XRPL	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.5 XRP	rNyrefquh...	Not delivered: a duplicate of the neighbouring instruction, refused by the executor's free pre-check before paying any attestation. No FXRP was minted for it — the transaction id is still unused in the MasterAccountController (read 19 Sep).

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-10 21:16	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.5 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-10 21:19 +2 min 41 s	FLARE 2	↳ 0x809d90c9... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-10 21:22 +5 min 42 s	FLARE 3	↳ 0x8eccb881... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: AstryumCageFactory.create(rNyrefquhQfqFYwHojT8aHwVPmKtqYZtg8); FXRP.approve(cage of rNyrefquh..., 0.1998 FXRP)	executor 0xBC43...2821	Birth of a stack by ONE signature: the factory accepts only the Personal Account of this root, so the payment itself is the authority; bridge (immutable council hash) and cage/vault are deployed and bound in this same transaction, with the first approve/deposit.
2026-09-10 21:56	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rNyrefquh...	Order attested but never executed: superseded by a later signature of the same order, or refused before sending — the bridge only executes the exact next nonce.
2026-09-10 22:02 +5 min 39 s	FLARE 2	↳ 0x0257123e... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-11 08:21	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rNyrefquh...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-11 08:21 +17 s	FLARE 2	↳ 0x4e0f7c47... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-11 08:23 +2 min 38 s	FLARE 3	↳ 0x5e05f56b... · bridge v2 of rNyrefquh....execute — nonce 0 → createPote “Wlja” · cooldown 24h · payee cap 1000 bps	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 0. A pote was created inside the cage (name, cooldown, payee cap): the manager can only create inside the rules of the cage.
2026-09-11 16:14	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNaFfVgbq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-11 16:15 +10 s	FLARE 2	↳ 0x0b565e43... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-11 16:17 +2 min 59 s	FLARE 3	↳ 0x5b439e54... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNaFfVgbq..., which runs the batch: pote in cage of rDcohNumB....claimRedeem(1, [['1', '281']])	executor 0xBC43...2821	Client exit, step 2: the claim after the cooldown; the receiver is fixed by the holder.
2026-09-12 19:39	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rP49LEKat... (multisig: 2 signers)	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-12 19:39 +25 s	FLARE 2	↳ 0x99348a71... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-12 19:41 +2 min 20 s	FLARE 3	↳ 0x201418cb... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rP49LEKat..., which runs the batch: pote in cage of rDcohNumB....claimRedeem(0, [])	executor 0xBC43...2821	Client exit, step 2: the claim after the cooldown; the receiver is fixed by the holder.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-15 20:51	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	raqZZbqWq...	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-09-15 20:52	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.5 XRP	raqZZbqWq...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-15 20:52 +11 s	FLARE 2	↳ 0x18abf102... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBc43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-15 20:55 +2 min 32 s	FLARE 3	↳ 0x2d457985... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of raqZZbqWq..., which runs the batch: AstryumCageFactory.create(raqZZbqWqAr5cJcwp9EjGYCWDtTy2eJKvM); FXRP.approve(cage of raqZZbqWq..., 0.1998 FXRP)	executor 0xBc43...2821	Birth of a stack by ONE signature: the factory accepts only the Personal Account of this root, so the payment itself is the authority; bridge (immutable council hash) and cage/vault are deployed and bound in this same transaction, with the first approve/deposit.
2026-09-15 20:54	XRPL	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.5 XRP	raqZZbqWq...	Not delivered: a duplicate of the neighbouring instruction, refused by the executor's free pre-check before paying any attestation. No FXRP was minted for it — the transaction id is still unused in the MasterAccountController (read 19 Sep).
2026-09-15 21:00	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	raqZZbqWq...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-15 21:00 +17 s	FLARE 2	↳ 0xb76bf722... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBc43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-15 21:02 +2 min 19 s	FLARE 3	↳ 0x20437f75... · bridge v2 of raqZZbqWq....execute — nonce 0 → createPote (demo vault) · cooldown 0s · payee cap 1700 bps	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 0. A pote was created inside the cage (name, cooldown, payee cap): the manager can only create inside the rules of the cage.
2026-09-15 22:03	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-15 22:03 +44 s	FLARE 2	↳ 0x6e2428d9... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-15 22:05 +2 min 18 s	FLARE 3	↳ 0x2cd8eac2... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(pote in cage of raqZZbqWq..., 5.0499 FXRP); pote in cage of raqZZbqWq....deposit(5049950, Personal Account of rNyrefquh...)	executor 0xBC43...2821	A client deposits into a managed pote with their own XRPL signature: shares are minted to the client's account, never to the manager; no approve to the operator exists anywhere in the batch.
2026-09-15 22:10	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-15 22:10 +39 s	FLARE 2	↳ 0x90d8604b... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-15 22:13 +3 min 03 s	FLARE 3	↳ 0x2a726055... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: pote in cage of raqZZbqWq....redeem(5049950000, Personal Account of rNyrefquh...)	executor 0xBC43...2821	Instant exit from a pote with no cooldown: shares burned, FXRP back to the holder — owner-only, no manager in the path.
2026-09-17 15:40	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-17 15:40 +45 s	FLARE 2	↳ 0xa6d87732... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-17 15:43 +2 min 51 s	FLARE 3	↳ 0x51dab113... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: FXRP.approve(pote in cage of raqZZbqWq..., 5.2924 FXRP); pote in cage of raqZZbqWq....deposit(5292361, Personal Account of rNyrefquh...)	executor 0xBC43...2821	A client deposits into a managed pote with their own XRPL signature: shares are minted to the client's account, never to the manager; no approve to the operator exists anywhere in the batch.
2026-09-17 15:46	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	raqZZbqWq...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-17 15:46 +22 s	FLARE 2	↳ 0xfc92975e... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-17 15:49 +2 min 42 s	FLARE 3	↳ 0x579d2533... · bridge v2 of raqZZbqWq....execute — nonce 1 → directTo 4.23389 FXRP → pote in cage of raqZZbqWq...	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 1. Capital directed to a registry-listed venue — the only kind of destination the cage accepts.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-17 17:41	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.35 XRP	rNyrefquh...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-17 17:41 +36 s	FLARE 2	↳ 0xc87b8457... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-17 17:43 +2 min 06 s	FLARE 3	↳ 0x607fe1ad... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rNyrefquh..., which runs the batch: pote in cage of raqZZbqWq....redeem(5292361000, Personal Account of rNyrefquh...)	executor 0xBC43...2821	Instant exit from a pote with no cooldown: shares burned, FXRP back to the holder — owner-only, no manager in the path.

Exchange — the structure of an exchange, rehearsed with our own accounts

First run (13–15 September). Root K1 (rwc9Dqir...) received custody-sector and company credentials and accepted them, anchored its constitution, gave birth to its cage by memo instruction, created two potes ("Exchange pote B" with a 72-hour cooldown and "easy yield") and directed capital by council order. The omnibus K2 (rMB8xNE6...) accepted the OMNIBUS designation issued by K1 and the per-box KYC records. Clients (the founders' own wallets, DestinationTag 101) deposited XRP into K2 three times; K2 signed the memo instruction that minted FXRP and deposited it into the pote **in the name of the client's passkey account** (0x0c8F...), which was deployed and exercised by the relayer paying gas. **Second run (18–19 September, self-serve).** Root rEvD7wxm... received CASP and KYB, anchored its constitution, gave birth to its cage, created "Exchange pote A" and directed 11.42 FXRP to a venue; its omnibus r4yp47QP... accepted OMNIBUS and KYC-201; a client deposited 13 XRP with tag 201; the omnibus minted and deposited in the name of passkey account 0x8B41...; and the next morning that passkey signed the exit — an independent agent paid 12.63 XRP back **into box 201 of the omnibus** and confirmed on Flare. What it proves and what it does not: the structure, the shares-in-the-client's-name and the client-signed exit are on chain; the KYC record is enforced by the backend, not by the ledger; the XRP in the omnibus is the exchange's custody by design; no third-party exchange has run on it.

12 circuits · 18 XRPL transactions · 26 Flare transactions

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-13 16:37	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	rw9Dqire...	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-09-13 16:38	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.5 XRP	rw9Dqire...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-13 16:39 +58 s	FLARE 2	↳ 0xe4f9c24c... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-13 16:41 +3 min 17 s	FLARE 3	↳ 0x01069e13... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rw9Dqire..., which runs the batch: AstryumCageFactory.create(rw9DqireGfwcEecBJD8QF41vmA18QN9rR); FXRP.approve(cage of rw9Dqire..., 0.1998 FXRP)	executor 0xBC43...2821	Birth of a stack by ONE signature: the factory accepts only the Personal Account of this root, so the payment itself is the authority; bridge (immutable council hash) and cage/vault are deployed and bound in this same transaction, with the first approve/deposit.
2026-09-13 16:50	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rw9Dqire...	Order attested but never executed: superseded by a later signature of the same order, or refused before sending — the bridge only executes the exact next nonce.
2026-09-13 16:51 +16 s	FLARE 2	↳ 0x414131b6... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-13 16:51	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rw9Dqire...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-13 16:51 +15 s	FLARE 2	↳ 0x350fad9d... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-13 16:53 +2 min 31 s	FLARE 3	↳ 0x807ade48... · bridge v2 of rwc9Dqire....execute — nonce 0 → createPote “Exchange pote B” · cooldown 72h · payee cap 2000 bps	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 0. A pote was created inside the cage (name, cooldown, payee cap): the manager can only create inside the rules of the cage.
2026-09-14 08:52	XRPL	Payment — Client deposit into the omnibus with DestinationTag 101 · 7 XRP	rNaFfVgbq...	The client needs no XRPL account of their own beyond this wallet: a tag is the box. The omnibus watcher credits it.
2026-09-14 08:52	XRPL	Payment — Client deposit into the omnibus with DestinationTag 101 · 4 XRP	rGb1VJkxm...	The client needs no XRPL account of their own beyond this wallet: a tag is the box. The omnibus watcher credits it.
2026-09-14 15:02	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 11 XRP (no tag: operational)	rMB8xNE6X...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-14 16:12 +1 h 09 min	FLARE 2	↳ 0x6d8a7b4c... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-14 16:14 +1 h 12 min	FLARE 3	↳ 0x1625666a... · AssetManagerFXRP.executeDirectMintingWithD ata — FXRP minted into Personal Account of rMB8xNE6X..., which runs the batch: FXRP.approve(pote in cage of rwc9Dqire..., 10.6893 FXRP); pote in cage of rwc9Dqire....deposit(10689300, passkey account 0x0c8F...D9CB)	executor 0xBC43...2821	A client deposits into a managed pote with their own XRPL signature: shares are minted to the client's account, never to the manager; no approve to the operator exists anywhere in the batch.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-14 21:28	XRPL	Payment — Client deposit into the omnibus with DestinationTag 101 · 9 XRP	rNaFfVgbq...	The client needs no XRPL account of their own beyond this wallet: a tag is the box. The omnibus watcher credits it.
2026-09-14 21:35	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 9 XRP (no tag: operational)	rMB8xNE6X...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-14 21:36 +29 s	FLARE 2	↳ 0xd1e6d09a... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-14 21:38 +2 min 49 s	FLARE 3	↳ 0x1cf6344f... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rMB8xNE6X..., which runs the batch: FXRP.approve(pote in cage of rwc9Dqire..., 8.6913 FXRP); pote in cage of rwc9Dqire....deposit(8691300, passkey account 0x0c8F...D9CB)	executor 0xBC43...2821	A client deposits into a managed pote with their own XRPL signature: shares are minted to the client's account, never to the manager; no approve to the operator exists anywhere in the batch.
2026-09-14 22:18	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rwc9Dqire...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-14 22:19 +20 s	FLARE 2	↳ 0xcc473658... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-14 22:20 +1 min 53 s	FLARE 3	↳ 0x08f2743c... · bridge v2 of rwc9Dqire....execute — nonce 1 → createPote "easy yield" · cooldown 0s · payee cap 1000 bps	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 1. A pote was created inside the cage (name, cooldown, payee cap): the manager can only create inside the rules of the cage.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-14 22:27	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rw9Dqire...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-14 22:27 +20 s	FLARE 2	↳ 0xee646326... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-14 22:29 +2 min 21 s	FLARE 3	↳ 0x1616e339... · bridge v2 of rw9Dqire....execute — nonce 2 → directTo 17.4425 FXRP → pote in cage of rw9Dqire...	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 2. Capital directed to a registry-listed venue — the only kind of destination the cage accepts.
2026-09-14 23:19	FLARE	0x3f4e1a87... · PasskeyAccountFactory.create — a P-256 (Face ID) account deployed for a client; the relayer only pays gas	executor 0xBC43...2821 (as relayer)	The client's account is a contract controlled by a P-256 passkey (Face ID) on their device; the relayer pays gas and cannot sign for it.
2026-09-14 23:19	FLARE	0xff8f7ef3... · PasskeyAccount.executeBatch on passkey account 0x0c8F...D9CB — the client signs with their passkey; the relayer only pays gas	executor 0xBC43...2821 (as relayer)	Signed by the client's passkey on their device; the relayer only carries it.
2026-09-18 13:53	XRPL	DIDSet — DIDSet — constitution anchored (hash + URI)	rEvD7wxmb...	The constitution is anchored in the account's own record; the contract references this hash for every governing call.
2026-09-18 13:54	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 0.5 XRP	rEvD7wxmb...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-18 13:54 +16 s	FLARE 2	↳ 0x1d27f3c7... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-18 13:56 +2 min 16 s	FLARE 3	↳ 0xe6473e1b... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of rEvD7wxmb..., which runs the batch: AstryumCageFactory.create(rEvD7wxmbwng8skmqNu8R6wAUQQVVXXACv); FXRP.approve(cage of rEvD7wxmb..., 0.1998 FXRP)	executor 0xBC43...2821	Birth of a stack by ONE signature: the factory accepts only the Personal Account of this root, so the payment itself is the authority; bridge (immutable council hash) and cage/vault are deployed and bound in this same transaction, with the first approve/deposit.
2026-09-18 14:02	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rEvD7wxmb...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-18 14:03 +21 s	FLARE 2	↳ 0xe9323b06... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-18 14:05 +2 min 56 s	FLARE 3	↳ 0x7cde4bc2... · bridge v2 of rEvD7wxmb....execute — nonce 0 → createPote “Exchange pote A” · cooldown 0s · payee cap 2000 bps	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 0. A pote was created inside the cage (name, cooldown, payee cap): the manager can only create inside the rules of the cage.
2026-09-18 17:45	XRPL	Payment — Client deposit into the omnibus with DestinationTag 201 · 13 XRP	rNaFfVgbq...	The client needs no XRPL account of their own beyond this wallet: a tag is the box. The omnibus watcher credits it.
2026-09-18 18:51	XRPL 1	Payment — Memo instruction (rail A) — XRP → FXRP → batch on Flare · 13 XRP (no tag: operational)	r4yp47QPb...	The only signature in this circuit: an XRPL Payment to the FAssets Core Vault whose memo commits to one exact batch. Every Flare row below is that commitment being carried out — nothing else can be.
2026-09-18 18:51 +42 s	FLARE 2	↳ 0x50b15738... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-18 18:53 +2 min 53 s	FLARE 3	↳ 0xc5d8c330... · AssetManagerFXRP.executeDirectMintingWithData — FXRP minted into Personal Account of r4yp47QPb..., which runs the batch: FXRP.approve(pote in cage of rEvD7wxmb..., 12.6873 FXRP); pote in cage of rEvD7wxmb....deposit(12687300, passkey account 0x8B41...537A)	executor 0xBC43...2821	A client deposits into a managed pote with their own XRPL signature: shares are minted to the client's account, never to the manager; no approve to the operator exists anywhere in the batch.
2026-09-18 19:17	XRPL 1	Payment — Council order (rail B) — 1-drop Payment, memo = hash(nonce, calldata) · 0.200001 XRP	rEvD7wxmb...	The council's only signature: a 1-drop Payment to the anchor whose memo is the hash of (nonce, calldata). The bridge on Flare accepts nothing else.
2026-09-18 19:18 +24 s	FLARE 2	↳ 0x48cd33dd... · FdcHub.requestAttestation — XRPPayment attestation of the XRPL transaction above · fee 20 FLR	executor 0xBC43...2821	Flare's Data Connector is asked to attest this XRPL transaction by its hash. The attestation comes from Flare's data providers, not from Astryum; the executor pays the fee — the user holds no FLR.
2026-09-18 19:20 +2 min 55 s	FLARE 3	↳ 0x8fcd48fb... · bridge v2 of rEvD7wxmb....execute — nonce 1 → directTo 11.4186 FXRP → pote in cage of rEvD7wxmb...	executor 0xBC43...2821	The bridge verified the FDC proof and its guards and executed nonce 1. Capital directed to a registry-listed venue — the only kind of destination the cage accepts.
2026-09-19 07:29	FLARE	0x4faaed84... · PasskeyAccountFactory.create — a P-256 (Face ID) account deployed for a client; the relayer only pays gas	executor 0xBC43...2821 (as relayer)	The client's account is a contract controlled by a P-256 passkey (Face ID) on their device; the relayer pays gas and cannot sign for it.
2026-09-19 07:29	FLARE 1	0xf89c0210... · PasskeyAccount.executeBatch on passkey account 0x8B41...537A — the client signs with their passkey; the relayer only pays gas	executor 0xBC43...2821 (as relayer)	Signed by the client's passkey on their device; the relayer only carries it. Here the batch requested a redemption to XRP — the agent's payout and confirmation follow.
2026-09-19 07:30 +27 s	XRPL 2	↳ Payment — an independent FAssets agent pays 12.6303 XRP to r4yp47QPb... into box 201 of the omnibus · redemption #13387954	rU3KcE1fq... (FAssets agent)	The agent pays the XRP on the XRP Ledger — Astryum does not sign it. It lands in the client's box of the omnibus: the exit needs no XRPL account of the client's own.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-19 07:32 +3 min 07 s	FLARE 3	↳ 0x967c8b94... · AssetManagerFXRP — the agent confirms the payment with its own FDC proof (RedemptionPerformed #13387954 · 12.6531 XRP)	FAssets agent 0x6751...554b	FAssets checks the agent's proof of the XRPL payment above and closes the redemption, releasing the agent's collateral. Neither Astryum nor the user signs this step — the circuit closes on its own.

Credentials — the access layer, issued by the notary, accepted by the subject

Every credential is an XLS-70 object: the issuer signs CredentialCreate with a mandatory expiry (and a URI pointing at the attestation when one exists — the identity credentials carry the Coinbase attestation on Base); the subject signs CredentialAccept, which is the consent. Two CredentialDelete + re-Create pairs on 13 September are a re-issue with corrected fields. Self-issued AIFM/KYC on rNyrefqu... (10 September) are the demo path where the subject issues to itself — they are listed and not counted as third-party attestations. The second exchange run added CASP and KYB for rEvD7wxm... and OMNIBUS + KYC-201 for its omnibus (18 September). What it proves: the manager gate reads these objects on the ledger before letting a cage be born; revocation is expiry; nothing here is issued by Astryum as a licence — the notary re-checks public facts and brings them on chain. Credentials live only on the XRP Ledger, so this section has no Flare rows.

0 circuits · 37 XRPL transactions · 0 Flare transactions

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-05 09:24	XRPL	CredentialCreate — CredentialCreate · type KYC · issuer rHKxjrGRrC... → subject rNaFfVgbqW... · expires 2026-12-04 09:23 · URI https://base.easscan.org/attestation/view/0xe8d6b58ee9c121e4 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-05 09:45	XRPL	CredentialCreate — CredentialCreate · type KYC · issuer rHKxjrGRrC... → subject rDcohNUmBA... · expires 2026-12-04 09:45 · URI https://base.easscan.org/attestation/view/0xe8d6b58ee9c121e4 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-05 09:46	XRPL	CredentialAccept — CredentialAccept · type KYC · issuer rHKxjrGRrC... → subject rDcohNUmBA... (no tag)	rDcohNUmB...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-05 15:06	XRPL	CredentialCreate — CredentialCreate · type AIFM · issuer rHKxjrGRrC... → subject rDcohNUmBA... · expires 2027-03-04 15:05 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-05 15:06	XRPL	CredentialAccept — CredentialAccept · type AIFM · issuer rHKxjrGRrC... → subject rDcohNUmBA... (no tag)	rDcohNUmB...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-09 17:41	XRPL	CredentialCreate — CredentialCreate · type KYC · issuer rHKxjrGRrC... → subject rNyrefquhQ... · expires 2026-12-08 17:41 · URI https://base.easscan.org/attestation/view/0xf6b765f4e8562d92 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-09 17:44	XRPL	CredentialAccept — CredentialAccept · type KYC · issuer rHKxjrGRrC... → subject rNyrefquhQ... (no tag)	rNyrefquh...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-10 18:28	XRPL	CredentialCreate — CredentialCreate · type AIFM · issuer rNyrefquhQ... → subject rNyrefquhQ... · expires 2027-03-09 18:28 (no tag)	rNyrefquh...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-10 21:02	XRPL	CredentialCreate — CredentialCreate · type KYC · issuer rNyrefquhQ... → subject rNyrefquhQ... · expires 2027-03-09 21:02 (no tag)	rNyrefquh...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-10 21:08	XRPL	CredentialCreate — CredentialCreate · type AIFM · issuer rHKxjrGRrC... → subject rNyrefquhQ... · expires 2026-12-09 21:08 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-10 21:09	XRPL	CredentialAccept — CredentialAccept · type AIFM · issuer rHKxjrGRrC... → subject rNyrefquhQ... (no tag)	rNyrefquh...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-13 15:26	XRPL	CredentialCreate — CredentialCreate · type CASP · issuer rHKxjrGRrC... → subject rwc9DqireG... · expires 2026-12-12 15:26 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-13 15:26	XRPL	CredentialDelete — CredentialDelete · type CASP · issuer rHKxjrGRrC... → subject rwc9DqireG... (no tag: operational)	rHKxjrGRr...	Revocation/re-issue by the issuer.
2026-09-13 15:27	XRPL	CredentialCreate — CredentialCreate · type CASP · issuer rHKxjrGRrC... → subject rwc9DqireG... · expires 2026-12-12 15:26 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-13 15:27	XRPL	CredentialAccept — CredentialAccept · type CASP · issuer rHKxjrGRrC... → subject rwc9DqireG... (no tag)	rwc9Dqire...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-13 16:36	XRPL	CredentialCreate — CredentialCreate · type KYB · issuer rHKxjrGRrC... → subject rwc9DqireG... · expires 2026-12-12 16:36 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-13 16:36	XRPL	CredentialDelete — CredentialDelete · type KYB · issuer rHKxjrGRrC... → subject rwc9DqireG... (no tag: operational)	rHKxjrGRr...	Revocation/re-issue by the issuer.
2026-09-13 16:36	XRPL	CredentialCreate — CredentialCreate · type KYB · issuer rHKxjrGRrC... → subject rwc9DqireG... · expires 2026-12-12 16:36 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-13 16:37	XRPL	CredentialAccept — CredentialAccept · type KYB · issuer rHKxjrGRrC... → subject rwc9DqireG... (no tag)	rwc9Dqire...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-13 16:53	XRPL	CredentialCreate — CredentialCreate · type OMNIBUS · issuer rwc9DqireG... → subject rMB8xNE6XY... · expires 2027-03-12 16:53 (no tag)	rwc9Dqire...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-13 16:54	XRPL	CredentialAccept — CredentialAccept · type OMNIBUS · issuer rwc9DqireG... → subject rMB8xNE6XY... (no tag: operational)	rMB8xNE6X...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-14 14:38	XRPL	CredentialCreate — CredentialCreate · type KYC-101 · issuer rwc9DqireG... → subject rMB8xNE6XY... · expires 2027-03-13 14:37	rwc9Dqire...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-14 14:38	XRPL	CredentialAccept — CredentialAccept · type KYC-101 · issuer rwc9DqireG... → subject rMB8xNE6XY... (no tag: operational)	rMB8xNE6X...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-14 21:27	XRPL	CredentialCreate — CredentialCreate · type KYC-102 · issuer rwc9DqireG... → subject rMB8xNE6XY... · expires 2027-09-14 21:26	rwc9Dqire...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-14 21:27	XRPL	CredentialAccept — CredentialAccept · type KYC-102 · issuer rwc9DqireG... → subject rMB8xNE6XY... (no tag: operational)	rMB8xNE6X...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-15 20:30	XRPL	CredentialCreate — CredentialCreate · type AIFM · issuer rHKxjrGRrC... → subject raqZZbqWqA... · expires 2026-12-14 20:30 · URI https://base.easscan.org/attestation/view/0x8d21d155fde21eb6 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-15 20:31	XRPL	CredentialAccept — CredentialAccept · type AIFM · issuer rHKxjrGRrC... → subject raqZZbqWqA...	raqZZbqWq...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-15 20:50	XRPL	CredentialCreate — CredentialCreate · type KYC · issuer rHKxjrGRrC... → subject raqZZbqWqA... · expires 2026-12-14 20:50 · URI https://base.easscan.org/attestation/view/0x8d21d155fde21eb6 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-15 20:50	XRPL	CredentialAccept — CredentialAccept · type KYC · issuer rHKxjrGRrC... → subject raqZZbqWqA...	raqZZbqWq...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-18 13:52	XRPL	CredentialCreate — CredentialCreate · type CASP · issuer rHKxjrGRrC... → subject rEvD7wxmbw... · expires 2026-12-17 13:52 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-18 13:52	XRPL	CredentialAccept — CredentialAccept · type CASP · issuer rHKxjrGRrC... → subject rEvD7wxmbw...	rEvD7wxmb...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-18 13:52	XRPL	CredentialCreate — CredentialCreate · type KYB · issuer rHKxjrGRrC... → subject rEvD7wxmbw... · expires 2026-12-17 13:52 (no tag: operational)	rHKxjrGRr...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-18 13:53	XRPL	CredentialAccept — CredentialAccept · type KYB · issuer rHKxjrGRrC... → subject rEvD7wxmbw...	rEvD7wxmb...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-18 14:05	XRPL	CredentialCreate — CredentialCreate · type OMNIBUS · issuer rEvD7wxmbw... → subject r4yp47QPbB... · expires 2027-03-17 14:05	rEvD7wxmb...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.
2026-09-18 14:06	XRPL	CredentialAccept — CredentialAccept · type OMNIBUS · issuer rEvD7wxmbw... → subject r4yp47QPbB...	r4yp47QPb...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.
2026-09-18 16:28	XRPL	CredentialCreate — CredentialCreate · type KYC-201 · issuer rEvD7wxmbw... → subject r4yp47QPbB... · expires 2027-09-18 16:27	rEvD7wxmb...	Issuer attests off-ledger and writes only type + expiry (+ URI); no personal data on the ledger.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-18 17:23	XRPL	CredentialAccept — CredentialAccept · type KYC-201 · issuer rEvD7wxmbw... → subject r4yp47QPbB...	r4yp47QPb...	The subject signs the acceptance: consent recorded on the ledger; the credential becomes readable by the gate.

Infrastructure — deployments, funding, the executor economy

Contract deployments (22 July bridge + vault, listed under Legacy; 5 August Legacy factory; 23 August stack + passkey factories; 6 September registry, cage factory, pote deployer), operational funding (the executor's first FLR on 12 July; on 23 August the Jul–Aug executor key handed its FLR and fee FXRP to the current key), and the executor's refuel cycles: the FXRP collected as executor fee is swapped to FLR on SparkDEX and unwrapped — the economy of the personal rail closes itself, at cost. These rows are Astryum's own operational transactions (no Source Tag, on purpose).

0 circuits · 0 XRPL transactions · 37 Flare transactions

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-12 19:56	FLARE	0xa25a28827... · 10.00 FLR → executor 0xD876...CecE (Jul–Aug key) (operational funding)	founder EVM wallet 0xc8be...e23f	Operational FLR for Astryum's own keys. The user's rails never require FLR.
2026-07-12 20:08	FLARE	0xa2983c5b... · 6,321.54 FLR → executor 0xD876...CecE (Jul–Aug key) (operational funding)	founder EVM wallet 0xc8be...e23f	Operational FLR for Astryum's own keys. The user's rails never require FLR.
2026-07-22 17:58	FLARE	0xe2bb3368... · 20.00 FLR → deployer 0x7D04...Ba8F (operational funding)	executor 0xD876...CecE (Jul–Aug key)	Operational FLR for Astryum's own keys. The user's rails never require FLR.
2026-07-25 15:07	FLARE	0x39fe5c43... · Executor refuel — approve on FXRP	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-07-25 15:07	FLARE	0x2ddab2e7... · Executor refuel — exactInputSingle (swap) on SparkDEX router	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-07-25 15:07	FLARE	0x9bd004d0... · Executor refuel — withdraw (unwrap) on WFLR	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-07-25 15:09	FLARE	0x2bbd041d... · Executor refuel — approve on FXRP	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-07-25 15:09	FLARE	0xaae5829c... · Executor refuel — exactInputSingle (swap) on SparkDEX router	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-07-25 15:09	FLARE	0x55cb5a0d... · Executor refuel — withdraw (unwrap) on WFLR	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-03 14:44	FLARE	0x0ec58ca6... · Executor refuel — approve on FXRP	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-03 14:44	FLARE	0xed2c8d08... · Executor refuel — exactInputSingle (swap) on SparkDEX router	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-03 14:44	FLARE	0x19bb3ffd... · Executor refuel — withdraw (unwrap) on WFLR	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-05 13:18	FLARE	0x65b3cc8e... · Contract deployment — LegacyStackFactory at 0xf93a...3e0a	deployer 0x7D04...Ba8F	Public bytecode at a fixed address — the code the circuits in this document call.
2026-08-22 10:39	FLARE	0xd79569c7... · Executor refuel — approve on FXRP	executor 0xD876...CecE (Jul–Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-08-22 10:39	FLARE	0x4e0a8b6e... · Executor refuel — exactInputSingle (swap) on SparkDEX router	executor 0xD876...CecE (Jul-Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-22 10:39	FLARE	0xcad452a0... · Executor refuel — withdraw (unwrap) on WFLR reverted	executor 0xD876...CecE (Jul-Aug key)	Reverted: nothing moved; the cycle was retried two minutes later.
2026-08-22 10:41	FLARE	0xbaf88543... · Executor refuel — approve on FXRP	executor 0xD876...CecE (Jul-Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-22 10:41	FLARE	0x6a998d87... · Executor refuel — exactInputSingle (swap) on SparkDEX router	executor 0xD876...CecE (Jul-Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-22 10:41	FLARE	0x54237c90... · Executor refuel — withdraw (unwrap) on WFLR	executor 0xD876...CecE (Jul-Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-23 09:46	FLARE	0x93ce0bce... · 40.00 FLR → deployer 0x7D04...Ba8F (operational funding)	founder EVM wallet 0xEAbc...462c	Operational FLR for Astryum's own keys. The user's rails never require FLR.
2026-08-23 09:52	FLARE	0x0e034a76... · Contract deployment — AstryumStackFactory at 0xc221...3783	deployer 0x7D04...Ba8F	Public bytecode at a fixed address — the code the circuits in this document call.
2026-08-23 09:52	FLARE	0xb16d3579... · Contract deployment — PasskeyAccountFactory at 0xfa55...d932	deployer 0x7D04...Ba8F	Public bytecode at a fixed address — the code the circuits in this document call.
2026-08-23 13:02	FLARE	0x9214af38... · FXRP.transfer(0x0A5f5ea7..., 1.6797 FXRP) — executor fee FXRP moved out of the retiring key	executor 0xD876...CecE (Jul-Aug key)	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-08-23 13:02	FLARE	0xf950d19d... · 1,535.99 FLR from executor 0xD876...CecE (Jul-Aug key) → 0x0a5f5ea7... (operational: key rotation / gas top-up)	executor 0xD876...CecE (Jul-Aug key)	Operational FLR for Astryum's own keys. The user's rails never require FLR.
2026-08-23 13:08	FLARE	0xa340fa2d... · 1,535.79 FLR → executor 0xBC43...2821 (operational funding)	0x0a5f5ea7...	Operational FLR for Astryum's own keys. The user's rails never require FLR.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-06 18:06	FLARE	0x6c1606e0... · Contract deployment — PoteDeployer at 0x5f84...7e9e	executor 0xBC43...2821	Public bytecode at a fixed address — the code the circuits in this document call.
2026-09-06 18:06	FLARE	0xcfc0c186b... · Contract deployment — AstryumRegistry at 0x18c8...0aec	executor 0xBC43...2821	Public bytecode at a fixed address — the code the circuits in this document call.
2026-09-06 18:06	FLARE	0x99708d84... · Contract deployment — AstryumCageFactory at 0xe897...41d9	executor 0xBC43...2821	Public bytecode at a fixed address — the code the circuits in this document call.
2026-09-10 14:28	FLARE	0xe3129c74... · Executor refuel — approve on FXRP	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-09-10 14:28	FLARE	0x7aef58cf... · Executor refuel — exactInputSingle (swap) on SparkDEX router	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-09-10 14:28	FLARE	0x43bd9a91... · Executor refuel — withdraw (unwrap) on WFLR	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-09-14 21:37	FLARE	0xd77dfb89... · Executor refuel — approve on FXRP	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-09-14 21:37	FLARE	0xc56114af... · Executor refuel — exactInputSingle (swap) on SparkDEX router	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-09-14 21:37	FLARE	0x5f990f7d... · Executor refuel — withdraw (unwrap) on WFLR	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-09-18 18:52	FLARE	0xf63dd14e... · Executor refuel — approve on FXRP	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.

WHEN (UTC)	LEDGER	TRANSACTION — WHAT IT DID	BY	WHY IT IS PROOF
2026-09-18 18:52	FLARE	0x66f9d8f3... · Executor refuel — exactInputSingle (swap) on SparkDEX router	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.
2026-09-18 18:52	FLARE	0xec0dd29b... · Executor refuel — withdraw (unwrap) on WFLR	executor 0xBC43...2821	The executor fee is collected in FXRP; the executor swaps it to FLR on SparkDEX and unwraps it to pay gas — the personal rail pays for itself, at cost.

Accounts & contracts

XRPL accounts and their Personal Accounts on Flare

XRPL ACCOUNT	ROLE	PERSONAL ACCOUNT ON FLARE
rNaFfVgbqWYotrR3Jz4RorirDUZgLfxxTt	founder wallet A	0xe7A124A08933d246398382be0Ce246157D9750a6
rBGVzjE5JcbZs2toRRLLrE6z62HmGNSxa	founder wallet B	0x0e4A88B694f8a48695Ff76d0a753B643AE688820
rGb1VJkxmMzZLbWRFfu9Xh5GVjZGYR4d3x	founder wallet C	0x0387537292208d59dd0CC164ef749fF6488Df7C9
rPbwqPUBRkDtThuGMittT3LCJs7PVfvnTwV	Legacy 4th signer	0x359c86511615BD6318f07e393425c8d4c72829E2
rsmvJMhhjn6L3oCf29UZE2mtw9kcsKDmrF	Legacy council #1 (3-of-4, master off)	0xe3030A6B8b567f4755A03791A91F51Cd57855697
rpM7wQNumAPSSbbb6J3yFJGnS9eoZthu8v	Legacy council #2 (2-of-3, master off; factory-born)	0x7e74084719FbA3E0618051c21d806aB4550D6531
rpPdcQ94UDUJiJryLG24wWZvXAFfmPbh23	Legacy #2 signer	—

rEPyRp9Y9qSwfKGf5Nxyg1Hf7yKzorzxWW	Legacy #2 signer	—
rJEUuTx7oEFTam3aDsrgsMcFLxmDyivtgk	Legacy #2 signer	—
rP49LEKattxJ9ppioRZVVRoZ7QeWvzYyuG	reinforced account (2-of-3, master off, constitution)	0xBD5709ff49d88Fa49A9D603895F5c615938F3ac3
r1ypgoqtdQ671bMK7g2KdjReekZH1MuoG	institutional council (2-of-3, master off) — pote A v1	0xE1ACf4F038C872F5e9C67b22F0d2ef7141Ab8749
rNyrefquhQfqFYwHojT8aHwVPmKtqYZtg8	manager root #1 (personal rail in Jul–Aug; manager from 9 Sep)	0xdE9554D04Ee85324C6530ffc4d0a0284E2A6521d
rDcohNUMBApE8bSPP695GkUsSMfBeM5Czj	manager root #2	0xF0eD0BaBc2c3d3CFE1346263Bcb1b19d51169F55
raqZZbqWqAr5cJcwP9EjGYCWDTty2eJKvM	manager root #3 (demo vault)	0x3772dc418cFeCB64aB9EFB8c2D6EEE6d2e8a64ec
rwc9DqireGfwcEecBJD8QF41vmA18QN9rR	exchange root K1	0x1F3F6e9091b8b11D91c8cF606f6AC0b5875F5ecF
rMB8xNE6XYyg9f6D7vB4LeZLwdVSCQwREf	exchange omnibus K2 (operational key of the simulated exchange)	0x640A904d3F54a27B7C480c50086CD390Fe0d06A0
rEvD7wxmbwng8skmqNu8R6wAUQQVVXXACv	exchange root K1, second run (self-serve, 18 Sep)	0x777000fC7844D817a3718da1b0D3178c89fbbbd4
r4yp47QPbB3XcVurs8pbE75k7EwQ1GEBAA	exchange omnibus K2, second run (operational key)	0x76e69348f586b1b688CcC4cC65cCBA5519f38055
rHKxjr6RrCegQhLrdnXEP AeGyeJ1JR4Hae	Astryum notary issuer (operational)	—
rK4tsuGhmbhaNQvvucL8n1RKLTARBCp3qm	Legacy order anchor (operational)	—
rLcoFM9XF8CL5GoFyMACguhdnDtwkNYDn7	v2 order anchor (operational)	—

Flare contracts and operational keys

ADDRESS	WHAT IT IS
0xbc439340ceb361f55f81d65a0c6cdfb1e7672821	executor 0xBC43...2821

0xd8767c3c4dc0a1e13f23368b172a5ff78b54cece	executor 0xD876...CecE (Jul–Aug key)
0x7d0441b6063cbce8315c3408327b5bdd51d8ba8f	deployer 0x7D04...Ba8F
0xeabcd745598916b0131ece397c8d6a332088462c	founder EVM wallet 0xEAbc...462c
0xc8be83f3ab02b35ddb5e5856c687240aa32ee23f	founder EVM wallet 0xc8be...e23f
0xf93a8a0bd93e95514ff02285349b0b1c1a5a3e0a	LegacyStackFactory
0xc221e9f447e65edae16a4af64d3a7a67e2ee3783	AstrumStackFactory
0xfa559ce04135835cb8f7d3cfad2b8d14525ad932	PasskeyAccountFactory
0x18c807d107ef9d337afa5bdd39b190c42dc50aec	AstrumRegistry
0xe897ffef10f950abc2d9c759107410f713e941d9	AstrumCageFactory
0x5f84c5f4805c29298af71fc6657d3e780ef27e9e	PoteDeployer
0x02ae9fcb76768e42b8d3ed9fe842238a6616b26f	XrplCouncilBridge #1
0x340b5990b23ccd2e17164a3043885f83603674b1	XrplCouncilBridge #2
0xc8379c79779cce3b738424892709fe0d4339e3b1	LegacyVault #1
0xc8b256e7a1e59e088ee38b8ee60079dfbfff0a55d	LegacyVault #2
0x2754381585646163cb72d025c2d46e780d0c5641	stack bridge of r1ypgoqtd...
0x21d4ccf29eeb61e573c2037f2b29b727168c3da9	pote A (v1)
0x11ce75810c2e353010011d938acce6361079b033	cage of rDcohNUmB...
0x2f4d0a8ae28334de05eee820f0e97fdb5620717c	cage of rEvD7wxmb...

0x6b8316fcd9aea8faba46e6c625ab6b5fc8808335	cage of rNyrefquh...
0xebc5228ecc915ad5014d2712c18c75724068bbb3	cage of raqZZbqWq...
0x8600851d10c318a088d0a6c964175a72b880b236	cage of rwc9Dqire...
0xe2f3db26dee655ba1d251a06cf038a77e42c7e71	bridge v2 of rDcohNUmB...
0x6e165872116d036b3c72b7d9883c8fd733d988e5	bridge v2 of rEvD7wxmb...
0x3abd1811e8174f7a65bdc1df2fa0865fc1cce3e9	bridge v2 of rNyrefquh...
0x7a42c825e5730135da8619363280e0ed47bd4866	bridge v2 of raqZZbqWq...
0xbf91bb5ab9f483bb40d6b1bc286ec5d7755fa58e	bridge v2 of rwc9Dqire...
0x1a5a3711a5663135b5855a869aebad249ffe11b9	pote in cage of rDcohNUmB...
0x7412d72f71bec96f76b3abbe2cef738f3d2b3c7c	pote in cage of rEvD7wxmb...
0xb5756570113fc9e1608473b9630e45d81eec66c8	pote in cage of rNyrefquh...
0x443e9cddb99fb9ca3f0551a47ca28f46ea6e509	pote in cage of raqZZbqWq...
0x52b2a6bc24de483e24dbd0bae90723ab580af53c	pote in cage of rwc9Dqire...
0x45f248e95fad466eb0089071c51a07bef72983eb	pote in cage of rwc9Dqire...
0x0c8f37934cdaf340460e8f42059ca257becbd9cb	passkey account 0x0c8F...D9CB
0x4011015268644de37061d6c9b734b1738a8933c8	passkey account 0x4011...33C8
0x8b416cd12d71254d9793e45627047d09933e537a	passkey account 0x8B41...537A

A row is never our word. It is a signature on one ledger and its consequence on the other.