



WHITEPAPER · VERSION 2.0

Programmable staking and the trust layer for agentic commerce

Revolution Network V2 on ZKsync OS. Carries forward the V1 economics of Libertas (V1 network) and specifies the V2 system.

Contents



Abstract

01 Introduction

- 1.1 The first problem: who captures value
- 1.2 The second problem: who stands behind an agent
- 1.3 The Revolution answer
- 1.4 The third market: real-world assets

02 Network overview

- 2.1 Design principles
- 2.2 The vocabulary
- 2.3 Network lineage

03 V1: Libertas and the programmable staking economy

- 3.1 What V1 introduced
- 3.2 Creator Pool use cases
- 3.3 What Libertas proved and where it fell short

04 Why V2: from Libertas to Virtus

- 4.1 Platform changes
- 4.2 Economic changes
- 4.3 Reasons

05 V2 architecture: the chain

- 5.1 Off chain decides, on chain verifies
- 5.2 Execution
- 5.3 Proving
- 5.4 Settlement and data availability
- 5.5 Verifying nodes
- 5.6 Traffic handling
- 5.7 Performance and capacity
- 5.8 Operations and cost

06 Identity: Nomen (name), Sigillum (verified identity), and Agens (agent)

- 6.1 Nomen (name)
- 6.2 Sigillum (verified identity) levels
- 6.3 Agens (agent) identities
- 6.4 Facets and zero-knowledge proofs
- 6.5 Identity in AI assistants
- 6.6 Compliance for real-world assets

07 Accounts, Potestas (permissions), and Mandatum (delegated authority)

- 7.1 EntryPoint v0.8 and modular accounts
- 7.2 Account classes and Potestas (permissions)
- 7.3 Passkeys
- 7.4 Mandatum (delegated authority)
- 7.5 Cornerstone services
- 7.6 The Cornerstone Budget

08 Payments and ACommerce

- 8.1 The stack

- 8.2 The Settlement contract
- 8.3 Netting for micro-payments
- 8.4 The Mandate Anchor and proof of intent
- 8.5 Escrow and splits
- 8.6 Payments on every rail
- 8.7 The Intent Book
- 8.8 Reputation, bonds, and disputes
- 8.9 Interoperability
- 8.10 Fees

09 Programmable staking and the creator economy

- 9.1 Roles
- 9.2 Node lifecycle
- 9.3 The split
- 9.4 Accounting
- 9.5 Emissions by time
- 9.6 Settlement fees to nodes
- 9.7 Claims
- 9.8 Parameters
- 9.9 Creator Pools in V2

10 Token economics

- 10.1 Supply and allocation
- 10.2 Token creation and the base token
- 10.3 Uses of REVO
- 10.4 Value flows
- 10.5 Proposed parameters

11 Developer platform

- 11.1 Products
- 11.2 SDK vocabulary
- 11.3 Gateway rules
- 11.4 Running a node
- 11.5 AI assistants
- 11.6 Merchants

12 Migration from Libertas (V1 network)

- 12.1 Steps
- 12.2 What carries over
- 12.3 Decisions before the freeze

13 Security, compliance, and governance

- 13.1 Threat model
- 13.2 Compliance posture
- 13.3 Governance

14 Roadmap

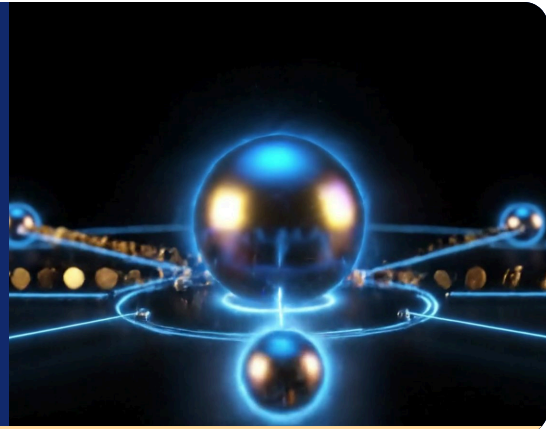
- 14.1 Phase 1 status

15 Open decisions, glossary, and disclosures

- 15.1 Open decisions before mainnet
- 15.2 Glossary
- 15.3 Disclosures
- 15.4 References

OVERVIEW

Abstract



Revolution is an Ethereum Layer 2 built for two things: programmable staking and the trust layer for agentic commerce. This whitepaper replaces the Revolution Network Whitepaper (2025) and Addendum A (ACommerce, September 2026). It carries forward the V1 economic design and specifies the V2 system.

1B

REVO fixed supply

250M

REVO emitted to stakers
over about ten years

40

emission eras of 90 days

0

personal data written on
chain

V1 introduced Creator Pools. Fans stake behind creators, creators stake behind nodes, and a fixed emission of 250 million REVO is split by public rules enforced in code. The V1 network, Libertas (V1 network), proved the model and exposed its limits. It ran on a superseded EraVM stack, produced no real proofs, and stopped settling to Ethereum in December 2025.

V2 is a regeneration on ZKsync OS. It keeps the token supply, the emission curve, and the staking economics. It replaces custom VM features with open Ethereum standards: ERC-4337 accounts, ERC-7579 modules, passkeys, Airbender validity proofs, and blob data on Ethereum. It adds a trust layer for software agents that buy and sell for people and businesses. A merchant can verify, at settlement, that a verified human stands behind the buyer agent, that the agent acts inside its mandate, and that the funds are real. No personal data goes on chain.

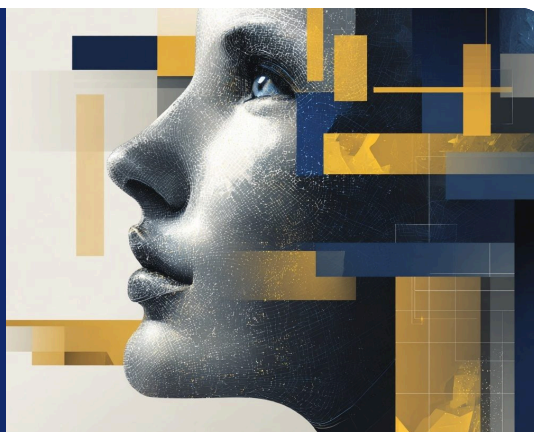
V2 is tested on Virtus (V2 testnet) before mainnet. The governing design rule is simple. Off chain decides. On chain verifies.

This document is a draft for review. It is not an offer of any token or security. Parameters marked *governance-set* are proposals. Section 15 lists the decisions still open.

01

THE PROBLEM

Introduction



The internet is built for extraction. Platforms take 20% to 30% of what creators and merchants earn, and they own the relationship with the customer. Revolution replaces that model with one in which participants own the network's revenue and the rules that split it.

1.1 The first problem: who captures value

Creators, fans, developers, and brands produce the value on today's platforms. Subscriptions, ad models, and paywalls route it to intermediaries. Revolution makes staking a protocol primitive. A fan stakes behind a creator instead of paying a subscription. The stake is reversible. The rewards flow by public rules that no platform can change after the fact.

1.2 The second problem: who stands behind an agent

Software agents now buy and sell on behalf of people and businesses. Payment rails exist. Catalogs exist. What is missing is trust at the point of settlement:

- **Identity.** ERC-8004, the Ethereum agent identity standard, does not bind an agent to a human or legal owner. Its identities are transferable. A June 2026 study (Xiong et al., arXiv) found coordinated Sybil behavior among 59% to 91% of reviewers and measured the median cost of manipulating an ERC-8004 reputation at under six cents.
- **Privacy.** Payment mandates in the market today are plaintext. A merchant that reads one learns the buyer's budget. US regulators treat undisclosed personalized pricing as a compliance risk.
- **Discovery.** Buyers pull from catalogs. Merchants pay platforms to be found. No live system lets a buyer post funded demand for merchants to bid on.
- **Enforcement.** When two agents agree, something must hold both to the agreement.

1.3 The Revolution answer

Revolution is the record of trust between parties that do not know each other. It binds every agent to a verified human or entity. It proves facts about them without revealing them. It holds the mandate, the escrow, and the settlement in one auditable place. It routes the fees to the operators who secure the network.

The two problems share one answer. Both need a neutral ledger that enforces who may do what, pays participants by public rules, and keeps personal data off chain. That ledger is Revolution.

■ **1.4 The third market: real-world assets**

Regulated issuers, funds, and creators need to launch, trade, and hold tokenized assets with compliance built in. Today each application repeats its own identity checks. On Revolution, a Sigillum (verified identity) facet proves accreditation or jurisdiction once and travels across applications. Section 6.6 sets out how.

02

ARCHITECTURE

Network overview



Revolution V2 has five build layers: chain, accounts, economics, identity, and ACommerce. Each layer has one job and a published interface. The diagram in this section shows how they connect.

Applications and agents

Creator and fan apps, buyer and merchant agents, passkey wallets, Claude and ChatGPT

Access (off chain)

Node SDK, developer APIs, revo-mcp, merchant SDK, explorer, bridge portal

Phases 1 to 3

ACommerce

Mandate Anchor, Settlement, Escrow, Splits, Intent Book, Reputation, Disputes, Z402

Specified

Identity: Nomen (name), Sigillum (verified identity), Agens (agent)

RNS .revo names, Agent Identity Registry, Facet Registry, Proof Verifier

Specified

Accounts

EntryPoint v0.8, ERC-7579 Nexus accounts, Cornerstone Paymaster, bundlers

Built

Economics

Staking for nodes, creators and fans; emission schedule; Cornerstone Budget

Built

Chain

ZKsync OS sequencer, Airbender provers, verifying nodes that co-sign batches

Devnet running

Ethereum L1

Batch commit, validity proof verification, execution, blob data

☒ The five Revolution build layers ☐ Off-chain access and applications

Figure 1. Revolution V2 architecture. The five build layers sit between applications and Ethereum.

Each layer depends only on the layers below it. Applications reach the chain through the access layer and ERC-4337 accounts. Ethereum holds the final record.

Revolution also describes itself in three layers. The base layer is the chain and accounts. The value layer is programmable staking. The trust layer is identity and ACommerce.

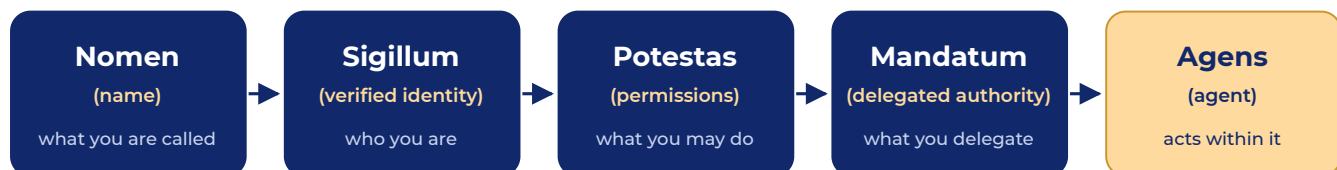
2.1 Design principles

- 1. Off chain decides. On chain verifies.** The chain holds money, stake, commitments, and revocation. Decisions that need data, policy, or judgment run off chain and return a signed or proven result.
- 2. Prove, do not store.** The chain holds commitments and verification results. It never holds personal attributes or documents.
- 3. Free to prepare, paid to settle.** Identity, proofs, anchoring, intent posting, escrow creation, and staking claims cost the caller nothing. Settlement pays a fee.
- 4. Accept every rail.** Stablecoin settlement is native. Card settlement is recorded and referenced. Revolution is the record of trust, not the only mover of money.
- 5. Bond the seller.** Merchant reputation is backed by stake.
- 6. Parent controls child.** Every agent has a verified parent that can revoke it.
- 7. Standards first.** Where an Ethereum standard exists, Revolution uses it.

2.2 The vocabulary

Revolution names its identity and agent stack in Latin, after the Roman legal order it echoes. Every Latin term appears with its plain-English meaning in parentheses.

Nomen (name) tells the network what you're called. **Sigillum (verified identity)** proves who you are. **Potestas (permissions)** determines what you're permitted to do. **Mandatum (delegated authority)** defines the authority you delegate. **Agens (agent)** acts within it.



Each layer depends on the one before it. A Mandatum (delegated authority) never exceeds the grantor's Potestas (permissions).

Figure 2. The identity and authority stack. Each term pairs its Latin name with its plain-English meaning.

Term	Plain English	Components	Standards
Nomen	name	Revolution Name Service (RNS), .revo names, resolver records	ENS-style naming
Sigillum	verified identity	Verification levels, facet root, Facet Registry, Proof Verifier	Verifiable credentials, ZK proofs
Potestas	permissions	Level unlocks, account class controls, policy hooks	ERC-7579 hooks
Mandatum	delegated authority	Standing Mandatum (signed parent policy); transaction Mandatum (anchored AP2 mandate)	ERC-7710, AP2

Term	Plain English	Components	Standards
Agens	agent	Agent identities, session keys, agent smart accounts	ERC-4337, ERC-8004 mirror

The terms encode four rules. A Mandatum (delegated authority) never exceeds the Potestas (permissions) of the principal who grants it. A Mandatum (delegated authority) is always revocable. An Agens (agent) has no authority of its own; every action traces back to a Sigillum (verified identity). Agents cannot create agents.

Credentials belong to Sigillum (verified identity). Potestas (permissions) is the set of actions that the Sigillum (verified identity) level and the account class unlock. Policy hooks enforce it on every call.

Contract names stay descriptive (Agent Identity Registry, Mandate Anchor, Facet Registry) so auditors read them without translation. The SDK and developer documentation use the vocabulary.

2.3 Network lineage



Figure 3. Network lineage. Libertas (V1 network) to Virtus (V2 testnet) to Revolution mainnet.

Name	Role	Meaning
Libertas (V1 network)	The V1 testnet on EraVM	Freedom. The founding principle.
Virtus (V2 testnet)	The short-lived V2 public testnet	Strength proven by action. Freedom earns its strength by being tested.
Revolution	The network brand and mainnet	The order built on both.

03

HERITAGE

V1: Libertas and the programmable staking economy



V1 established the economic core that V2 keeps: Creator Pools, a two-level delegation cap, a fixed 250 million REVO emission over a 40-era decay curve, and public split rules. It ran as Libertas (V1 network), a sovereign ZK Stack chain on EraVM.

3.1 What V1 introduced

Creator Pools. A Creator Pool is a creator's own stake plus its fans' stake, delegated to one node. Reward logic sits at the protocol level, not in a platform's database. The pool is an interface for staking. The distribution is deterministic and on chain.

A block-reward economy for an L2. Most L2s pay operators only from sequencer fees and MEV. Fee-only models struggle to attract operators before volume exists. V1 solved the cold start with explicit rewards on a predictable decay curve, so nodes and creators could commit before fee revenue scaled.

Soft consensus. V1 separated transaction ordering from reward distribution. A stake-weighted priority counter picked one winning node per batch. Each node's counter rose by its stake each round. The highest counter won the batch reward and then fell by total network stake. Over time each node won in proportion to its stake.

Bounded topology. Fan to creator to node, never deeper. At most 250 nodes and 200 creators per node. A 10-minute unstake cooldown. These caps kept distribution loops tractable at L2 gas prices.

3.2 Creator Pool use cases

The V1 whitepaper set out four markets. They remain the demand thesis for programmable staking in V2.

Market	Staking replaces	Examples
Consumer and social	Subscriptions, tips, follower counts	Stake for access to a creator; stake-tiered memberships; stake as public backing; stake-backed project funding; stake-based loyalty and product drops
DeFi	Static collateral and governance-token gating	Staked positions as collateral; stake-tiered fees and rates; stake-weighted credit terms; liquidity for AMMs and lending markets
Gaming and metaverse	Microtransactions and one-time purchases	Stake for content and game modes; stake as guild reputation; stake-funded expansions and tournaments

Market	Staking replaces	Examples
AI and agents	Fixed API keys, prepaid credits, closed licensing	Stake for model access and priority; stake-tiered agent services; stake-weighted compute allocation; stake-weighted model governance

V2 extends the AI and agents market into a full trust layer for agentic commerce (section 8).

3.3 What Libertas proved and where it fell short

Libertas proved that staking rewards can be split across nodes, creators, and fans by public, on-chain rules. It ran the REVO token, Creator Pools, a NodeContract, the Revolution Name Service, and the Amped.Bio faucet and creator experience.

It also exposed limits that V2 must fix:

Finding	Effect
EraVM is superseded upstream by ZKsync OS	A dead-end execution stack and custom compiler
The verifier was a placeholder	No real validity proofs were ever produced
Settlement to Ethereum stopped on 3 December 2025 (last commit batch 4,386; last execute batch 3,103)	The chain ran without L1 finality
Batches averaged 108 seconds, not the assumed 40	About 20.8M REVO emitted against about 53.0M intended over the same period
About one transaction per batch, a sequencer limited to 0.3 CPU, database on network storage	Throughput far below design
Reward distribution ran inside block production	Chain liveness was coupled to reward accounting
The NodeContract and the gas waiver were not in any repository	Unrecoverable dependencies
Rewards were paid under the Amplify technical draft split, not the published rule	A gap between what holders were told and what the code did
EigenDA was described but not found in the code	A public claim with no implementation

V2 corrects every item on this list. Section 4 explains how.

04

THE UPGRADE

Why V2: from Libertas to Virtus



V2 is a regenesis, not an upgrade. Revolution starts a new chain on ZKsync OS, keeps the V1 economics, and carries V1 state across by snapshot (section 12).

4.1 Platform changes

Area	V1 (Libertas)	V2
Stack	zkSync Era, EraVM, protocol v29	ZKsync OS, protocol v31, EVM equivalent
Compiler	zksolc	solc and standard Ethereum tooling
Accounts	EraVM native account abstraction	ERC-4337 EntryPoint v0.8 plus ERC-7579 modular accounts
Free cornerstone calls	Planned bootloader gas waiver	ERC-4337 verifying paymaster funded from emissions
Proofs	Placeholder verifier	Airbender validity proofs from public testnet onward
Data availability	EigenDA described, not implemented	Rollup with blob data on Ethereum
Settlement to L1	Stopped 3 December 2025	Monitored service level objective with alerts
Batching	About one transaction per batch	Many transactions per batch
Staking payouts	Pushed inside block production, one winning node per batch	Continuous accrual by time, paid by claim
Identity	Names only	Names, verified identity, ZK facets, agent identities
Name service TLD	.rev in early ACommerce material	.revo

4.2 Economic changes

Area	V1	V2
Emission total and curve	250M REVO, 40-era decay	Unchanged

Area	V1	V2
Emission clock	Per batch (assumed 40 s)	Per second, same curve, same total
Node and creator split	Published rules; code paid the technical-draft rule	Published rules, enforced exactly (section 9.3)
Node fee	20% default, 50% max	20% minimum, 50% maximum
Cornerstone Budget	Specified as a share of emissions	Implemented. 5% proposed, 20% hard cap in code.
Settlement fees	Treasury, nodes, and Cornerstone Budget (draft)	Node operators, pro rata by node weight

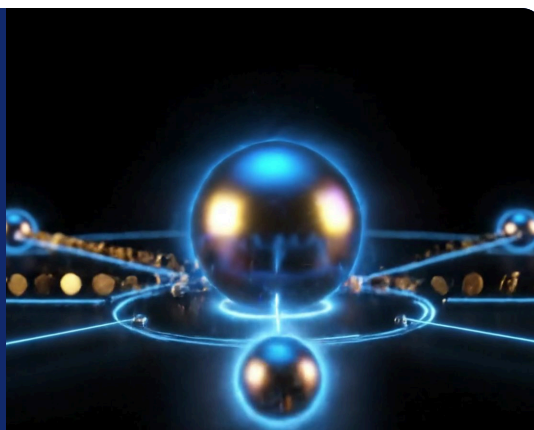
4.3 Reasons

- **ZKsync OS.** EraVM is superseded upstream. ZKsync OS is EVM equivalent, faster, and proven by Airbender. Standard tools work. A genesis also removes the dependency on Libertas code that exists in no repository.
- **Standards over custom VM features.** Wallets, bundlers, and auditors already understand ERC-4337, ERC-7579, and ERC-7562. Using them lowers integration cost and audit risk.
- **Emissions by time.** ZKsync OS produces blocks only when transactions exist. A per-batch clock lets traffic change token supply. A per-second clock does not.
- **Claims over pushes.** Pushing rewards to 250 nodes and 50,000 creators inside block production does not scale. Index-based claims cost the same at any size and keep the chain live if accounting fails.
- **Real proofs and real settlement.** A rollup without validity proofs and L1 settlement does not deliver the security it advertises. V2 does.

05

THE CHAIN

V2 architecture: the chain



Revolution V2 runs `zksync-os-server` (pinned at v0.23.0, protocol v31.0). It executes EVM bytecode, proves every batch with Airbender, and publishes its data to Ethereum in blobs.

5.1 Off chain decides, on chain verifies

Four constraints set what goes on chain. Throughput is not one of them.

Constraint	Why it matters	Response
Proving cost	Every on-chain transaction must be proven	Fewer, denser on-chain transactions
Data availability cost	Every byte posted to Ethereum blobs is paid for	Net and batch before posting
State growth	On-chain state is permanent and every node carries it	Store roots and hashes, not records
Privacy and regulation	On-chain data is public and cannot be erased	No PII, KYC data, or payment detail on chain. Ever.

Each function therefore splits between an on-chain part and an off-chain part:

Function	On chain	Off chain
Settlement	REVO and USDC transfers, batch settlement, staking, emissions, claims	Nothing
Authority	Nomen (name) registry, Sigillum (verified identity) roots, revocation lists, Potestas (permissions) levels	Credential issuance, verification workflows
Delegation	Mandatum (delegated authority) hashes, spend caps, expiry, revocation	Per-transaction policy evaluation, AP2 mandate handling
Compliance	Attestation hashes and issuer signatures	Provider data, screening, case management
Privacy	Z402 and facet proof verifiers	Proof generation on the client or a prover service
Sponsorship	Cornerstone Paymaster (thin verification)	Sponsorship policy service: eligibility, quotas, abuse controls

Function	On chain	Off chain
Access	Nothing	Node SDK, developer APIs, MCP servers, indexers

Every off-chain service is an operator, and an opaque operator is an intermediary. Each service must therefore meet four requirements: signed outputs, on-chain verification of those outputs, audit logs with the input hash and policy version, and replaceability through key rotation by governance.

5.2 Execution

Contracts compile with standard `solc`. Foundry, Hardhat, viem, ethers, and Otterscan work unchanged.

Item	ZKsync OS behavior	Consequence
Native account abstraction	Not supported	All V2 accounts use ERC-4337
EIP-7702	Planned upstream	Adopted when available (Phase 5)
Bootloader hooks	Not available to chains	Gas sponsorship runs through a paymaster
Block production	Only when transactions exist	Staking uses <code>block.timestamp</code> , never <code>block.number</code>
P-256 precompile	Present at <code>0x100</code>	Passkeys verify on chain at about 31,000 gas
CREATE2 deployer	Present at the canonical address	EntryPoint v0.8 sits at its canonical address
Fees	Maximum of EVM gas and (proving plus pubdata cost)	Local simulation underestimates; deploy with a gas multiplier

5.3 Proving

Every batch carries an Airbender validity proof. Fake provers run only on local devnets. Revolution rents GPUs. On testnet one host with a 24 GB class GPU and at least 192 GB of RAM runs both the FRI and SNARK roles. On mainnet two dedicated hosts run them, with marketplace GPUs for bursts.

5.4 Settlement and data availability

Batches are committed, proven, and executed on Ethereum. Separate operator keys sign each step and are held in a KMS. Revolution V2 is a rollup that publishes its data to Ethereum in blobs. Validium was considered and not chosen. It would add a data availability vendor, a verification bridge, and audit scope, and weaken recoverability. Data on Ethereum is the strongest assurance for RWA, custody, and ACommerce counterparties. V1 references to EigenDA are withdrawn.

5.5 Verifying nodes

A verifying node is an external node run by a staked, approved operator. It replays every block and co-signs each batch before commit, using ZKsync OS batch verification. A batch moves forward only with a threshold of signatures: **2 of 3 on testnet, 3 of 5 on mainnet**, with at least three mainnet signers

independent of Revolution and Amplify. This gives two properties. Committed data already exists on independent machines. The signatures add an execution check independent of the proof system.

The staking contract links economics to this role. Only the node's registered verifier key can send its heartbeat. A node that misses heartbeats beyond the liveness window can be jailed by anyone and stops earning.

5.6 Traffic handling

Planning shares by traffic type set how each reaches the chain:

Traffic	Planning share	Path to chain
Agent micro-payments	70%	Netted off chain; one settlement per batch, anchored by a Z402 proof
Direct payments and transfers	15%	Individually on chain
Identity and delegation operations	8%	Aggregated into registry root updates
Staking and reward claims	5%	Individually on chain
Other contract calls	2%	Individually on chain

Revocation always posts immediately. A revoked identity or mandate fails at the next transaction, not the next batch.

5.7 Performance and capacity

The V2 devnet ran 3,000 transactions at 1,629 tx/s and 10,000 at 819 tx/s with zero errors, on a 2 vCPU host with fake provers. These are floor numbers, not capacity claims. The load model projects that the layered design cuts on-chain transactions by about 76%:

Gross volume per day	Design	On-chain tx per day	Peak TPS	Planning cost per day
1M	All on chain	1,000,000	116	\$1,205
1M	Layered	235,144	27	\$614
10M	All on chain	10,000,000	1,157	\$8,155
10M	Layered	2,350,144	272	\$2,247

Cost inputs are placeholders until replaced with a prover quote, live blob fees, and a Virtus (V2 testnet) benchmark with real proofs.

5.8 Operations and cost

One active sequencer runs with a warm standby. A second consensus node is the standby batcher, promoted by a drilled runbook within 30 minutes. A scaled RPC tier of external nodes serves public traffic. State runs on local NVMe. Keys sit in a KMS or HSM, never in configuration or repositories.

Environment	Infrastructure per month	L1 settlement per month
Public testnet	\$960 to \$1,700	About \$0 (Sepolia)
Mainnet at launch	\$2,800 to \$5,450	\$350 to \$1,730 at a 1 hour batch cadence and 0.2 to 1 gwei

These are September 2026 planning figures from public list prices, not commitments. Batch cadence is the largest variable cost. Mainnet launches at a 1 hour cadence with size-based sealing and shortens it as fee revenue grows.

06

IDENTITY

Identity: Nomen (name), Sigillum (verified identity), and Agens (agent)



Revolution binds every agent to a verified human or legal entity and proves facts about that owner without revealing them. The identity stack covers three subjects: Know Your Customer for people, Know Your Business for entities, and Know Your Agent for agents tied to their owner.

6.1 Nomen (name)

The Revolution Name Service (RNS) issues Nomen (names): human-readable `.revo` names. A name resolves to one or more accounts and service records. The name is the durable namespace. The smart account is the execution principal. Public records never contain personal data.

Nomen (name) follows the structure of a Roman name:

Part	Revolution meaning	Example	Status
Nomen (name)	Root name a principal owns	<code>rob.revo</code>	V2 launch
Praenomen (agent name)	Label of an agent under a root	<code>shopping</code> in <code>shopping.rob.revo</code>	Phase 2
Cognomen (subname)	Entity, treasury, or project subname that is not an agent	<code>treasury.acme.revo</code>	Proposed, Phase 5
Agnomen (earned reputation)	Standing earned through conduct	A reputation threshold proven as a facet	Phase 3

6.2 Sigillum (verified identity) levels

A Sigillum (verified identity) is the verified identity behind a Nomen (name). RNS and approved issuers verify the holder off chain with identity data partners and KYC modules. The chain records commitments, never attributes.

Level	Meaning	Potestas (permissions) unlocked
L1 Basic	Email and device bound, uniqueness checked	Agent creation, low-value settlement
L2 Verified	Government identity checked against data partners	Age and jurisdiction facets, standard settlement

Level	Meaning	Potestas (permissions) unlocked
L3 Enhanced	KYC and AML screening completed	Accredited-investor facets, RWA-linked settlement
L3 Entity	Legal entity verified with an authorized signatory	Merchant agents, bonded reputation pools

Levels are *governance-set*. Additional issuers can join the Facet Registry issuer set. An issuer adapter verifies an external issuer's signature and revocation status off chain and maps its claims to facet leaves. The target for mainnet is direct issuance of circuit-friendly commitments, so no adapter trust is needed.

6.3 Agens (agent) identities

An Agens (agent) identity is a subname of a verified Nomen (name):

```
rob.revo      verified human (parent)
shopping.rob.revo Agens (agent), child
acme.revo     verified legal entity (parent)
sales.acme.revo merchant Agens (agent), child
```

Rules:

1. Only a parent with a verified Sigillum (verified identity) can create an Agens (agent) identity.
2. Agent identities are soulbound. There is no transfer function.
3. The parent can suspend or revoke a child, and its Mandatum (delegated authority), at any time. Revocation is permanent.
4. A revoked identity's history stays attached to the parent.
5. A parent that revokes a bonded merchant agent forfeits bond subject to open disputes.
6. If RNS suspends a parent for fraud, all of its children are suspended.
7. Agents cannot create agents.

Each record holds the agent ID, the parent ID, the agent's smart account, a rotatable controller key, a hash of the parent's policy, a facet root, a reputation pointer, a status, and a read-only mirror ID in the ERC-8004 registry with transfer disabled. The parent's policy is the standing Mandatum (delegated authority) (section 7.4).

6.4 Facets and zero-knowledge proofs

A facet is one provable attribute of a Sigillum (verified identity): a verified human stands behind this agent, the human is over 21, the residence is in an allowed set, the holder is accredited, the agent's reputation exceeds a threshold, the mandate covers this purchase.

The issuer commits each credential as a leaf in a Merkle tree. The root is the identity's facet root. To prove a facet, the holder's agent generates a zero-knowledge proof that a leaf under the root satisfies the predicate and is not in the issuer's revocation set. The Proof Verifier checks it. Each proof carries a nullifier bound to the settlement, so it cannot be replayed.

Initial facets (*governance-set*): `human.verified`, `entity.verified`, `age.over.18`, `age.over.21`, `jurisdiction.in`, `jurisdiction.not.in`, `investor.accredited`, `sanctions.clear`, `reputation.above`, `kya.certified`, `mandate.covers`.

Per identity the chain stores one facet root, one reputation pointer, and per-settlement nullifiers. Nothing else. Routine root updates aggregate on a short cadence (planning figure: every 10 minutes). Revocations and high-value changes post immediately.

6.5 Identity in AI assistants

The identity beta runs inside Claude and ChatGPT through MCP servers. An assistant can create a user's verified identity, agents, and smart account in conversation. Identity, facet proofs, the Intent Book, and Settlement are exposed as MCP tools and A2A agent cards.

6.6 Compliance for real-world assets

Revolution is the compliance and settlement layer for tokenized assets. It is not an issuer, a custodian, or a broker-dealer. Issuance and custody run through licensed partners.



Compliance travels with the asset: proofs, not documents.

Role	Who	What they do
Identity and credentials	Revolution Name Service (RNS)	Issues privacy-preserving, non-transferable proofs of KYC, accreditation, and jurisdiction
Custody	Fortuna Custody	Regulated custody for tokenized securities, treasuries, and income-distribution contracts
Issuance and secondary trading (EU)	Black Manta Capital Partners	Licensed MiFID II broker-dealer for compliant issuance and trading across the EU
Origination	COSIMO Digital	Originates, structures, and manages the real-world asset pipeline

REAL-WORLD ASSET PARTNERS



Compliance is enforced by facets, not by documents on chain. A transfer of a regulated instrument requires an `investor.accredited`, `jurisdiction.in`, or `sanctions.clear` proof from the holder, verified by the Proof Verifier at the point of transfer. L3 Enhanced and L3 Entity identities unlock RWA-linked settlement (section 6.2). Data on Ethereum in blobs and real validity proofs give issuers and custodians the settlement assurance they require (section 5.4).

07

ACCOUNTS

Accounts, Potestas (permissions), and Mandatum (delegated authority)



The account is the primary product primitive. A Revolution account signs in with a passkey, recovers without a seed phrase, proves eligibility privately, transacts without holding gas, stakes, and grants bounded authority to agents.

7.1 EntryPoint v0.8 and modular accounts

V2 uses the standard ERC-4337 EntryPoint v0.8 at [0x4337084D9E255Ff0702461CF8895CE9E3b5Ff108](#), the same address as on Ethereum. Existing bundlers, SDKs, and wallets recognize Revolution without custom configuration. Two self-hosted Pimlico Alto bundlers serve it.

Accounts are Biconomy Nexus accounts, pinned at an audited commit and built from source. Nexus implements ERC-7579, so validators, executors, hooks, and fallback handlers are separate modules.

Module type	Revolution use	Status
Validator	ECDSA today; passkey (P-256) next; multisig and institutional signer sets later	ECDSA built and tested. Passkey in Phase 2.
Executor	Session keys, agent permissions, staking automation, scheduled actions	Phase 2
Hook	Policy checks: spend limits, target and selector allowlists, time windows, credential predicates	Phase 2
Fallback handler	Compatibility and message signing	As needed

The default wallet allows no delegatecall through sponsored paths and no arbitrary third-party modules. A curated module list applies until a module registry and attestation process exist.

7.2 Account classes and Potestas (permissions)

The account class sets the baseline Potestas (permissions) of an account. The Sigillum (verified identity) level extends it.

Class	Typical user	Default controls
Personal	Consumer, creator, fan	Passkey plus recovery; sponsored cornerstone calls

Class	Typical user	Default controls
Professional	Operator, advisor	Passkey plus hardware key; sessions
Business	Company, merchant, issuer	Multisig or policy quorum; entity verification
Institutional	Fund, custodian	Hardware or MPC signers; multi-role approvals
Agent	AI or automation	No Potestas (permissions) of its own; acts only under a Mandatum (delegated authority)

7.3 Passkeys

The P-256 precompile on ZKsync OS verified passkey signatures on the devnet at about 31,000 gas. Passkeys are the default consumer sign-in. The passkey validator is integrated from an audited implementation.

7.4 Mandatum (delegated authority)

A Mandatum (delegated authority) has two forms.

- **Standing Mandatum (delegated authority).** The parent's signed policy for one Agens (agent). It is stored off chain with its hash on chain. Minimum fields: spend cap per transaction, spend cap per period, allowed categories, optional counterparty allowlist, human-approval threshold, expiry, and a kill switch.
- **Transaction Mandatum (delegated authority).** The human's signed authorization for a specific purchase, in AP2 form (Intent, Cart, and Payment Mandates). The Mandate Anchor records its hash with the agent identity and expiry (section 8.4).

An Agens (agent) never holds the owner's key. It holds a session key through an executor module. A policy hook enforces the Mandatum (delegated authority) on every call: target, selector, asset, amount, rate, and expiry. A call outside it fails. The parent can revoke it at any time. The same caveats map to ERC-7710 delegations, so compatible wallets can issue a Mandatum (delegated authority) without Revolution tooling. On chain keeps hashes, caps, expiry, and revocation. Per-transaction policy evaluation runs off chain. For netted micro-payments, the batch is checked against the caps on chain.

7.5 Cornerstone services

Free to prepare. Paid to settle. A buyer will not pay to post an intent that may never fill. A fan should not pay gas to claim rewards. A user who bridges in without REVO should still be able to act. Revolution covers that gas.

ZKsync OS gives chains no bootloader hook, so the V1 gas waiver becomes a standard ERC-4337 paymaster run by the network: the **Cornerstone Paymaster**. It is a verifying paymaster:

1. An off-chain sponsorship policy service decides eligibility: identity level, quota, bridge-in status, sanctions screen, and mandate scope.
2. It signs a short-lived voucher bound to the sender, target, nonce, and gas ceiling.

3. The paymaster checks the voucher signature, its expiry, the target allowlist, and a per-period spend cap from its own storage. Nothing else.

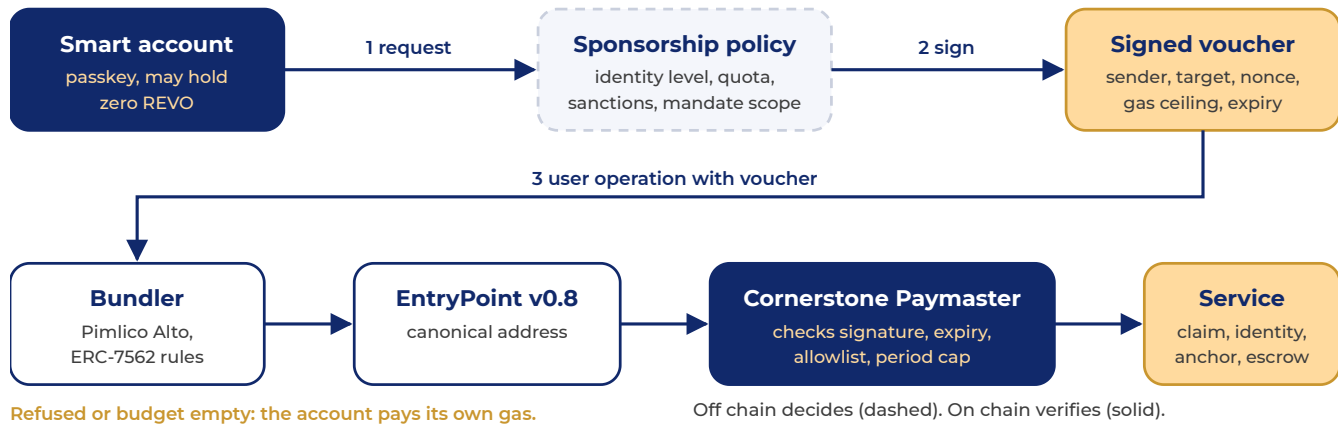


Figure 4. Cornerstone sponsorship. The policy service decides off chain; the paymaster verifies on chain.

This keeps validation inside ERC-7562 rules, so public bundlers accept the paymaster under load. The paymaster decodes the account's call data and refuses any target that is not a registered cornerstone contract and any delegatecall. If the service refuses or the budget runs out, the client resubmits and pays its own gas. Sponsorship fails open to user-paid gas. It never fails closed.

Cornerstone targets at launch: staking claims, the Agent Identity Registry, the Proof Verifier, the Mandate Anchor, the Intent Book, escrow creation, and bond and delegate calls on the Reputation Registry. Governance adds each as it ships.

On the devnet, a new account holding zero REVO sent sponsored single and batch calls through EntryPoint v0.8, with gas paid from the budget. Non-cornerstone targets, ineligible senders, over-allowance requests, and delegatecalls were each refused in tests. The built contract checks eligibility on chain. Moving eligibility into the voucher is the Phase 1 change.

7.6 The Cornerstone Budget

The Cornerstone Budget is a share of staking emissions. It is not new issuance. On every emission update the staking contract sets aside `cornerstoneBps` of the scheduled emission before the staker split. Anyone can call `releaseCornerstone()` to move it into the paymaster's EntryPoint deposit.

Parameter	Proposed
<code>cornerstoneBps</code>	500 (5%); hard maximum 2,000 (20%) in code
<code>periodLength</code>	1 day
<code>allowancePerPeriod</code>	Set from measured cost of a normal day's cornerstone calls
<code>maxCostPerOp</code>	Set from the most expensive cornerstone call plus margin
<code>maxPauseDuration</code>	3 days

At 5% the budget in era 1 is 0.1 REVO per second, about 777,600 REVO over the 90-day era.

08

ACOMMERCE

Payments and ACommerce



ACommerce is commerce conducted agent to agent. A personal agent holding a verified identity and a funded mandate negotiates, contracts, and settles with a merchant agent. The human sets policy. The agents execute. Revolution supplies the trust at the moment of settlement.

8.1 The stack

Layer	Components
4. Enforcement	Dispute contract, Reputation Registry, bonds
3. Discovery	Intent Book
2. Payments	Settlement contract, Mandate Anchor, Escrow, Splits, x402 facilitator, Z402
1. Identity	Agent Identity Registry, Facet Registry, Proof Verifier (section 6)
0. Chain and economics	ZKsync OS, ERC-4337 accounts, Cornerstone Paymaster, staking

The V1 draft placed several of these in privileged system contracts. On ZKsync OS they are ordinary contracts made free through the Cornerstone Paymaster. They are simpler to audit and upgrade, and the user sees the same result.

8.2 The Settlement contract

Every ACommerce transaction resolves to one Settlement instance: buyer agent, seller agent, mandate hash, required facet proofs, optional intent proof, offer hash, escrow, splits, delivery terms, and state. The diagram below shows its states.

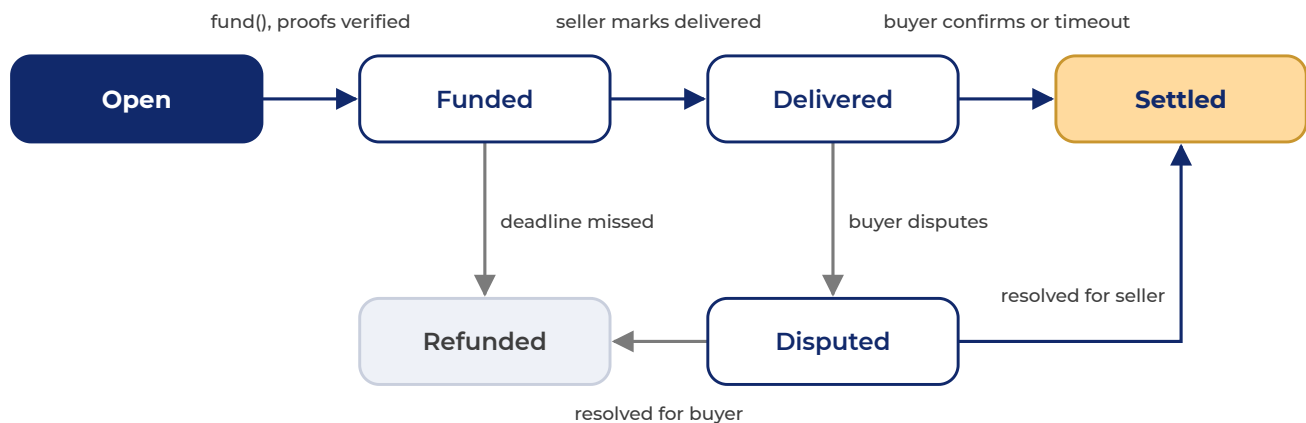


Figure 5. Settlement lifecycle. Funds release only after delivery, confirmation, or a resolved dispute.

`fund()` is the gate. It reverts unless every required facet has a verified proof, the mandate hash is anchored, the intent proof verifies where required, and escrow covers the price. Settlement is the step that pays the network fee.

8.3 Netting for micro-payments

Agent micro-payments net off chain. One settlement transaction carries a batch of payments, and a Z402 proof anchors the batch. Direct payments settle individually. Per-payment mandate checks run off chain. The batch is checked against the Mandatum (delegated authority) caps on chain. The planning ratio of 100 payments per settlement is validated in the ACommerce proof of concept.

8.4 The Mandate Anchor and proof of intent

The Mandate Anchor records `keccak256(mandate)` with the agent identity and expiry. Anchoring is a free cornerstone call. An auditor with the mandate can verify it. Anyone else learns only that an anchor exists.

Proof of purchase intent is a zero-knowledge proof over the mandate that the settlement's category, counterparty class, and amount fall inside its bounds. The merchant learns the purchase is authorized. It does not learn the budget or the urgency.

8.5 Escrow and splits

Escrow is created free when a buyer posts an intent or accepts an offer. Funds lock in USDC or REVO. Three modes: Full (release on confirm or timeout), Milestone (tranches on delivery marks), and Stream (metered release up to a cap, for subscriptions, utilities, and API usage).

Splits are fixed at funding and paid in the settlement transaction: merchant, affiliate (the creator or expert agent whose recommendation was accepted), network fee, and any withholding the buyer's jurisdiction facet requires. Affiliate payouts encoded in the settlement are a core business model for creators.

8.6 Payments on every rail

Stablecoin path (x402). A merchant's HTTP 402 response names Revolution as the network and USDC as the asset. Revolution publishes an open-source x402 facilitator. The Revolution extension adds a `facets` field and a `requireIntentProof` flag, and accepts a `.revo` name in `payTo`.

```
{
  "scheme": "exact",
  "network": "revolution",
  "asset": "USDC",
  "amount": "4999",
  "payTo": "sales.acme.revo",
  "facets": ["human.verified", "age.over.18", "jurisdiction.in:US,CA,GB"],
  "requireIntentProof": true
}
```

Z402: private proof of payment. Standard x402 accepts a transaction hash as proof. That exposes the buyer's wallet and history to the merchant. Z402 replaces the hash with a zero-knowledge proof that a settlement exists, matches the stated terms, and has not been presented before. The merchant learns those three facts and nothing else. One verification can combine payment, age, jurisdiction, and mandate checks. Z402 is proposed as an extension inside the x402 ecosystem, not a rival protocol. The proof is free. The fee sits on settlement.

Card path. The Settlement contract records identity, proofs, and mandate hash as in the stablecoin path. The card processor authorizes with the agent's scoped credential. The authorization reference is written to the contract. On-chain splits settle from a merchant-funded balance. Card money never touches the chain.

On-ramp and off-ramp. A licensed provider lets a user buy USDC and REVO by card or bank transfer directly into a smart account, and sell back to fiat. The provider is chosen after legal review. USDC launches as bridged USDC through the ZK Stack shared bridge. Native USDC requires Circle support for the chain.

8.7 The Intent Book

The buyer publishes. Merchants respond. A buyer agent posts an intent: category, a hash of the specification, a commitment to the price ceiling, a zero-knowledge proof that escrow covers the hidden ceiling, required seller facets, delivery constraints, expiry, and a responder filter. Merchant agents that pass the filter respond with signed offers. Expert agents can attach recommendations and are paid as affiliates if their recommended offer is accepted. Accepting an offer creates the Settlement contract.

Posting, offering, recommending, and accepting are free cornerstone calls. Only active agent identities with verified parents can post. Every intent needs a funds proof. Posters who let intents expire at a high rate are throttled.

8.8 Reputation, bonds, and disputes

Merchant and expert agents bond REVO before they can respond to intents. Bonds use the Creator Pool delegation model: third parties can co-stake into a merchant's reputation pool, share its fee revenue, and share its slashing risk.

The Reputation Registry records settlement history, dispute history, and bond per agent identity. A published, deterministic function produces a score, exposed only as a `reputation.above` facet. The threshold is proven. The number is not revealed. Reputation is the Agnomen (earned reputation) of an identity: earned through conduct, never bought. Because identities are soulbound and records follow the verified parent, a merchant cannot buy a good record or discard a bad one.

Disputes require a refundable REVO deposit. A governance-appointed adjudicator set handles them at launch, with the option to route to external arbitration. Slashed bond pays buyer restitution first, then the adjudicator fee, then the treasury.

Slashing is programmatic once a dispute is adjudicated:

Event	Consequence
Non-delivery	Refund to buyer from bond, plus a <i>governance-set</i> penalty (<code>slashPenaltyBps</code> , 1,000 bps proposed)
Misdescription	Partial refund from bond under the adjudicator's ruling
Withdrawn offer	Fixed penalty
Fraudulent facet	Full bond forfeited; identity suspended

Once Z402 produces a private payment history bound to persistent identities, the registry can score agents for underwriting: lower bonds, higher caps, deferred settlement. The score covers agents and merchant entities only, never the human, and is exposed only as threshold proofs.

8.9 Interoperability

Protocol	Connection
x402	Revolution facilitator, <code>facets</code> extension, Z402
AP2	Canonical mandate format; hashes anchored; intent proofs over AP2 fields
UCP catalogs	Merchant agents read UCP manifests to price offers
TAP and Web Bot Auth	Agent controller key signs HTTP requests
Card network agentic tokens	Card path authorization reference
Machine Payments Protocol	Stream mode maps session cap and rate to escrow cap and rate
ERC-8004	Read-only mirror of every agent identity, transfer disabled
ERC-8183	Adapter over the Settlement state machine

Protocol	Connection
ERC-7710	Agent policy as a scoped delegation
MCP and A2A	Identity, proofs, Intent Book, and Settlement as tools and agent cards
Know-Your-Agent frameworks	Parent binding, <code>kya.certified</code> facet import, monitoring feed

Revolution does not issue cards, run a catalog, custody fiat, or replace a merchant's processor.

8.10 Fees

Operation	Fee
Identity, facet proof, mandate anchor, intent post, offer, escrow creation, staking claim	None (cornerstone)
Settlement	<code>settlementFeeBps</code> on settled value in USDC or REVO. Proposed 50 bps, 20% discount when paid in REVO.
Stream tick	Flat fee per release
Dispute filing	Refundable REVO deposit

09

STAKING

Programmable staking and the creator economy



V2 keeps the V1 staking economics and enforces the published split exactly. Nodes, creators, and fans share a fixed emission of 250 million REVO over about ten years. What changes is the machinery: rewards accrue by time and are paid by claim.

9.1 Roles

Role	Does	Earns
Verifying node	Stakes at least the minimum, runs a verifying node, sends heartbeats, sets a node fee	100% of rewards on its own stake, its node fee on delegated rewards, and settlement fees
Creator	Stakes its own REVO into a Creator Pool delegated to one node it chooses, sets a creator fee	100% of rewards on its own stake, plus its creator fee on fans' rewards
Fan	Stakes REVO into a creator's pool	A pro-rata share of the fans' portion, net of the creator fee

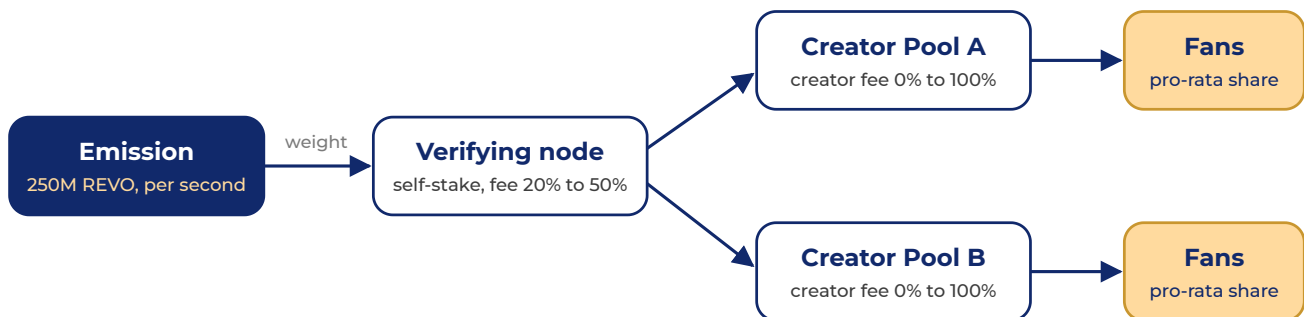


Figure 6. Two-level delegation. Emission flows to nodes by weight, then to Creator Pools, then to fans.

Topology is capped at two levels: fan to creator, creator to node. Each creator delegates to one node and selects that node itself. An address holds one role. Self-staking and creator-to-creator delegation are refused.

9.2 Node lifecycle

- 1. Register.** `registerNode(verifier, feeBps)` with at least `minNodeStake`. The stake locks. The node is Pending.
- 2. Approve.** Governance approves the node once it runs a verifying node and its verifier key joins the accepted batch signers. Entry is approved, not open. Operators buy their stake or receive a REVO grant from Revolution.
- 3. Heartbeat.** The node agent confirms the local node is synced and calls `heartbeat(node)` from the verifier key.
- 4. Jail.** If no heartbeat arrives within `livenessWindow`, anyone can jail the node. A jailed node earns nothing. Its creators can redelegate.
- 5. Exit.** `beginNodeExit()` starts unbonding. Stake is withdrawable after `nodeUnbonding`.

Caps: 250 node registrations and 200 creators per node. When a node is full, a larger creator can join by evicting the node's smallest creator. Evicted creators and their fans keep their principal and rewards and can redelegate.

9.3 The split

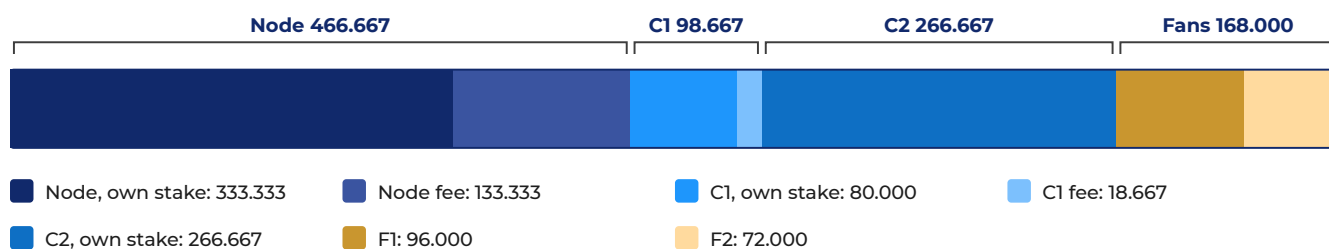
For node n with self-stake S , delegated stake D , and weight $W = S + D$:

1. Node n earns a share R of each emission in proportion to W over all active nodes.
2. The node keeps $R \times S / W$ on its own stake, with no fee.
3. On delegated rewards $R \times D / W$, the node takes its fee c (20% minimum, 50% maximum, 20% default).
4. The rest, $R \times D / W \times (1 - c)$, goes to pools in proportion to each pool's stake $d = s + f$ (creator stake s , fan stake f).
5. From its pool share P , the creator keeps $P \times s / d$ with no fee.
6. On the fans' portion $P \times f / d$, the creator takes its fee k (0% to 100%).
7. Fans share the rest in proportion to their stake.
8. A node with no delegation keeps all of R . A pool with no fans pays all of P to the creator.
9. Rounding remainders go to the node and the creator. Index dust stays reserved in the contract. Nothing is created.

Worked example (checked by a unit test). One node: self-stake 100, fee 20%. Creator C1: own stake 30, fee 10%, fans F1 40 and F2 30. Creator C2: own stake 100, fee 0%, no fans. $W = 300$. $R = 1,000$ REVO.

Step	REVO
Node self-stake rewards: $1,000 \times 100/300$	333.333
Delegated rewards: $1,000 \times 200/300$	666.667
Node fee: 20% of 666.667	133.333
Node total	466.667

Step	REVO
To pools: 533.333, split 100:100	266.667 each
C1 own stake: $266.667 \times 30/100$	80.000
C1 fee: 10% of 186.667	18.667
C1 total	98.667
C2 total (no fans)	266.667
F1: $168 \times 40/70$	96.000
F2: $168 \times 30/70$	72.000
Sum	1,000.000



One emission of 1,000 REVO under the published split. The node keeps all rewards on its own stake and takes its 20% fee only on delegated rewards. Creators keep their own-stake rewards and take their fee only from fans.

Figure 7. The worked example in section 9.3 as a single bar. Every REVO is accounted for.

9.4 Accounting

Rewards accrue through three indexes: per unit of node weight, per unit of delegation, and per fan share. Every settle step costs the same whether a node has 2 creators or 200. Nothing is pushed during block production. Four invariants are checked by stateful fuzzing on every build: the contract is solvent, everything claimable is backed, total paid never exceeds total emitted plus fees, and the Cornerstone Budget is conserved. Owed rewards can never be withdrawn by the owner.

9.5 Emissions by time

The curve and total are unchanged: 40 eras of 90 days, 250,000,000 REVO in total. Era 1 emits 2 REVO per second, about 15.55M REVO over the era. Each later era's per-second rate equals the Libertas (V1 network) per-batch value divided by 40, exactly. The emission table is generated from the live Libertas contract.

A per-second clock fixes the rate to the curve. A per-batch clock lets traffic and operator settings change token supply. That is how Libertas emitted about 20.8M REVO against about 53.0M intended. The V2 emission start point is an open decision (section 15).

9.6 Settlement fees to nodes

ACommerce settlement fees enter the staking contract through `notifyFees()`. They are paid 100% to node operators, pro rata by node weight, through a separate fee index. They do not pass through the creator and fan split. If no node is active, fees join the reward pot. This decision is flagged for legal review.

9.7 Claims

Claims are pull-based and free, because they are cornerstone calls. Three drivers use the same contract:

Driver	How
Smart contract	Any account calls <code>claim</code> , <code>claimCreator</code> , or <code>claimNode</code> , or the <code>...For</code> variants that pay the owner
Timer	The rewards keeper settles, releases the Cornerstone Budget, jails stale nodes, runs auto-claims, and compounds for fans who opted in
Agent or app	A REST API exposes emission, nodes, pools, fan positions, and claim, compound, and settle endpoints; an agent with a session key can call the contract directly

9.8 Parameters

Parameter	V1 published	V2
Minimum node stake	100,000 REVO	<i>Governance-set</i> ; 100,000 REVO proposed for mainnet
Node fee	20% default, 50% max	20% min, 50% max, enforced
Creator fee	0% to 100%	0% to 100%
Max nodes	250	250
Max creators per node	200	200
Fan unstake cooldown	10 minutes	10 minutes proposed
Liveness window	n/a	1 hour proposed
Node unbonding	n/a	7 days proposed

9.9 Creator Pools in V2

Every V1 use case in section 3.2 runs on V2 with three additions. Fans stake from a passkey account without holding gas. Creators earn affiliate splits when their recommendations close a sale (section 8.5). A Creator Pool explorer shows pools, fees, and rewards in public, so fans choose with full information.

10

TOKEN

Token economics



REVO has a fixed supply of 1,000,000,000. V2 changes none of the V1 allocation. It adds uses for the token: the Cornerstone Budget, merchant bonds, settlement fees, and dispute deposits. ACommerce adds no new emission.

10.1 Supply and allocation

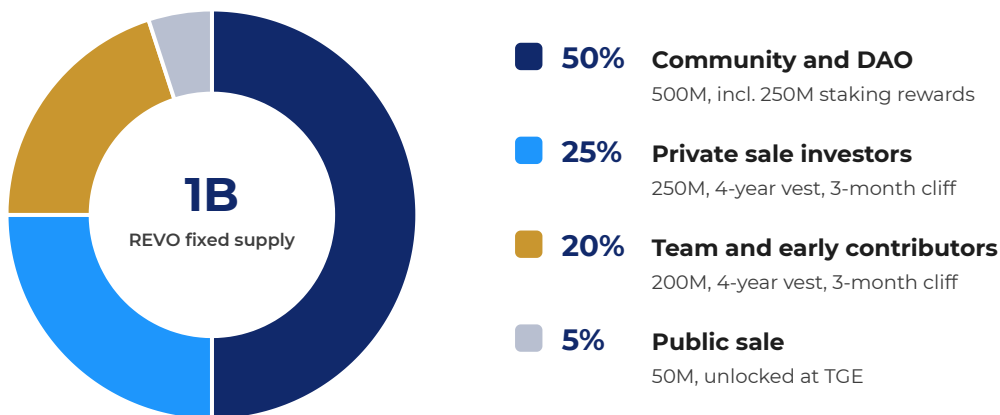


Figure 8. REVO allocation. Fixed supply of 1,000,000,000 REVO; unchanged from V1.

Allocation	Share	REVO	Terms
Community and DAO	50%	500,000,000	Airdrops, grants, and 250M staking rewards over about ten years
Private sale investors	25%	250,000,000	4-year linear vesting, 3-month cliff
Team and early contributors	20%	200,000,000	4-year linear vesting, 3-month cliff
Public sale	5%	50,000,000	Unlocked at TGE

Circulating supply at TGE is 30% (300,000,000 REVO).

10.2 Token creation and the base token

REVO is created once, as an ERC-20 on Ethereum with a fixed supply. On Revolution it is the chain's base token: it pays gas and is bridged from Ethereum through the ZK Stack shared bridge. The 250M staking reward pot is funded on L2 from the Community and DAO allocation in tranches. Vesting contracts sit on Ethereum.

The V1 Sepolia REVO token is a test token. Virtus (V2 testnet) uses a new test REVO. The mainnet token is created alongside the mainnet ecosystem contracts. Its owner, mint recipients, and parameters are approved in governance before deployment.

10.3 Uses of REVO

Use	Mechanism
Gas	Base token of the chain
Staking	Nodes, creators, and fans (section 9)
Verifying-node bond	Minimum node stake
Cornerstone Budget	A share of emissions pays for free cornerstone calls
Merchant and expert bonds	Staked into reputation pools; slashable
Settlement fees	Payable in REVO at a discount; paid to node operators
Dispute deposits	Refundable lock
Governance	Voting under section 13.3

10.4 Value flows

- Emission (250M over about ten years) splits into the Cornerstone Budget (5% proposed) and the staker split across nodes, creators, and fans.
- The Cornerstone Budget funds the paymaster deposit, which pays gas for free cornerstone calls.
- Settlement fees flow to node operators by node weight.
- Slashed bonds pay buyer restitution, then the adjudicator fee, then the treasury.

As emissions decay, settlement fees become the main reward for operators. V1 described a transition to a fee-sustained network. V2 makes the fee source concrete: agentic commerce settlement.

10.5 Proposed parameters

All *governance-set*. Values are proposals for testnet.

Parameter	Proposed
<code>cornerstoneBps</code>	500
<code>settlementFeeBps</code>	50

Parameter	Proposed
revoFeeDiscountBps	2,000 of the fee
minMerchantBond	Per-category table
bondToVolumeRatio	500 bps of trailing 30-day settled value
slashPenaltyBps	1,000 bps of settlement value on non-delivery
delegatorFeeShareBps	2,000 bps of a merchant's fee revenue to its bond delegators
minNodeStake	100,000 REVO
Node fee range	2,000 to 5,000 bps

11

BUILDERS

Developer platform



Every outside party reaches Revolution through one generated surface. Contract ABIs, one OpenAPI 3.1 specification, and one GraphQL schema are the single source of truth. Everything else is generated from them. Writes stay on chain as user operations. No Revolution service holds user keys.

11.1 Products

Product	What it is	Who uses it	Phase
@revolution/sdk	TypeScript SDK that works with viem; generated Python and Go clients	App, wallet, and agent developers	1 to 3
Developer APIs	REST v1, a GraphQL indexer, and webhooks behind one gateway at api.revolutionnetwork.io	All integrators	1 to 3
revo-node	CLI and SDK module to install, register, and run a verifying node	Approved node operators	1
Bridge Portal	Front end over the ZK Stack canonical bridge, with deposit gas sponsorship and auto-claimed withdrawals	Users moving REVO, ETH, and USDC	1
Block explorer	Blocks, batches, L1 settlement status, verified contracts, user operations, staking activity	Everyone	1
Creator Pool Explorer	Public view of pools, nodes, leaderboards, and a reward simulator	Fans, creators, operators	1
Revolution Developers	Documentation portal at developers.revolutionnetwork.io	All builders	1
RNS Console	Owner control room for Nomen (name), Sigillum (verified identity), agents, and Mandatum (delegated authority)	Identity owners	2
revo-mcp	Remote MCP server for Claude and ChatGPT	People acting through AI assistants	2
@revolution/merchant-sdk	Turns any HTTP service into a Z402 merchant	Merchants	3

11.2 SDK vocabulary

SDK namespaces use the Latin terms, each with a plain-English alias:

Namespace	Alias
<code>revo.nomen</code>	<code>revo.names</code>
<code>revo.sigillum</code>	<code>revo.identity</code>
<code>revo.potestas</code>	<code>revo.permissions</code>
<code>revo.mandatum</code>	<code>revo.delegation</code>
<code>revo.agens</code>	<code>revo.agents</code>

Method names stay in plain English, for example `Mandatum.revoke()`.

11.3 Gateway rules

- One API key per project covers every surface. Public RPC needs no key and has a low rate limit.
- Rate limits are tiered by compute units. Tracing and keeper triggers sit on a gated tier.
- The indexer pages by block range, because ZKsync OS produces blocks only when transactions exist.
- Sponsored gas uses the verifying paymaster through ERC-7677 paymaster web service calls.
- Security-critical reads stay on chain views.
- Any reward-rate estimate stays gated until legal sign-off.

11.4 Running a node

`revo-node` runs pinned `zksync-os-server` as an external node with the batch verification client, a node agent, and a metrics exporter. Admission has three steps: business verification through RNS, which issues a business Sigillum (verified identity); an on-chain allowlist; and a purchased or granted REVO stake. Creators choose their node from a public list. There is no platform default and no promoted slot.

11.5 AI assistants

`revo-mcp` lets a person act on Revolution from Claude or ChatGPT. It is an OAuth 2.1 resource server with RNS as the authorization server. Its tools cover the full flow: check identity status, start verification, claim a name, create an agent, set a Mandatum (delegated authority), list, pause, or revoke agents, find a merchant, check out, and fetch a receipt. Owner-signed actions go to a passkey approval page. An agent pays with a session key bounded by its on-chain Mandatum (delegated authority).

The documentation portal is built for agents too. It publishes `llms.txt`, a Markdown twin of every page, a docs MCP server, and "Open in Claude" and "Open in ChatGPT" on every page.

11.6 Merchants

The merchant SDK adds Z402 to any HTTP service through x402 with scheme `z402`. It provides `z402Required`, `verifyPayment`, and `confirmOrder`, with adapters for Express, Hono, and Next.js. Z402 internals stay unpublished until counsel clears them.

12

MIGRATION

Migration from Libertas (V1 network)



V2 starts from a new genesis. The state that matters is carried across by a published snapshot. Libertas (V1 network) stopped settling to Ethereum in December 2025, so the snapshot is the authoritative record of V1 holdings.

12.1 Steps

1. Freeze Libertas at a published block.
2. Run the snapshot tool at that block. It exports REVO balances, staked positions, NodeContract nodes, deployed contracts, token holders, and each fan's unclaimed rewards in every Creator Pool.
3. Publish the snapshot and its hash.
4. Credit balances on Virtus (V2 testnet), then on mainnet, by airdrop.
5. Recreate stakes and pools on V2, subject to the decisions below.
6. Re-register names under `.revo`.
7. Run V1 and V2 in parallel for a defined window, with user notices.

A preliminary snapshot at Libertas block 530,866 is in progress. It stays preliminary until Libertas is frozen.

12.2 What carries over

Item	Treatment
REVO balances	Credited by airdrop
Staked positions and Creator Pools	Recreated on V2 if in migration scope
Unclaimed fan rewards	Snapshotted per pool and credited on V2
RNS names	Re-registered under <code>.revo</code>
Applications	Redeployed with standard <code>solc</code> ; Amped.Bio is made compatible with V2
NDAU holdings	Conversion to REVO is complete

12.3 Decisions before the freeze

- **Scope.** Balances only, or also stakes, pools, and names.

- **Split rule.** Libertas paid rewards under the Amplify technical-draft split. V2 enforces the published rule. Holders receive a notice explaining the change.
- **Emission start point.** Resume by amount emitted, resume by calendar, or restart at era 1. This decides how much of the 250M remains at V2 genesis.

13

TRUST

Security, compliance, and governance



Every contract receives an independent audit before mainnet. Circuits are audited before activation. No personal data goes on chain. Every privileged contract is owned by a timelock under a multisig from day one.

13.1 Threat model

Threat	Mitigation
Sybil farming of free calls	Signed sponsorship vouchers tied to identity level; per-identity allowance; cost cap per operation; fallback to paid gas
Paymaster drain	Call data decoded; every target must be registered; delegatecall refused
Bundler rejection of the paymaster	Validation reads only the voucher and the paymaster's own storage (ERC-7562)
Compromised agent key	Mandatum (delegated authority) caps, parent kill switch, key rotation, human-approval threshold
Compromised parent key	Smart account recovery; RNS suspension of the parent and all children
Forged or replayed facet proof	Verification against facet and revocation roots; audited, versioned circuits; nullifiers
Reputation purchase or whitewashing	Soulbound identities; records follow the parent; bonds forfeited on revocation under dispute
Staking insolvency	Explicit reward pot; allocation capped by the pot; four invariants fuzzed on every build
Offline node earning rewards	Heartbeat from a dedicated verifier key; permissionless jailing
Verifying nodes offline	Signature threshold below signer count; alerts on signature wait time
Operator key compromise	Separate commit, prove, and execute keys in a KMS
Opaque off-chain operator	Signed outputs, on-chain verification, audit logs, replaceable keys (section 5.1)
Module risk in accounts	Curated modules; audited account base; no delegatecall in sponsored paths
Prompt injection via offers	Offers are structured records with hashed terms; free text excluded

13.2 Compliance posture

This section maps the design to rules now forming. It is not legal advice.

Expectation	Component
Identify the party behind an agent	Every Agens (agent) traces to a verified Sigillum (verified identity)
Prove authorization of each transaction	Mandatum (delegated authority): Mandate Anchor, proof of intent, policy caps
Verify age or jurisdiction without collecting documents	Age and jurisdiction facets
Screen counterparties	<code>sanctions.clear</code> facet
Gate regulated instruments	<code>investor.accredited</code> facet
Avoid personalized-pricing exposure	Price ceiling commitment; the merchant never sees willingness to pay
Log agent actions	Mandate Anchor and settlement record
Minimize personal data on chain	Commitments, attestation hashes, and issuer signatures only
Retain records for regulators	Compliance and sponsorship decisions logged off chain with policy versions
Provide a dispute path	Escrow, bonds, adjudication

Items that require counsel sign-off before mainnet:

1. Settlement fees paid to node operators: characterization of fee income, licensing, and tax by jurisdiction.
2. Fan staking and rewards: securities and consumer-protection treatment. No reward rate is advertised without review.
3. Agent credit score: confirm it stays a commercial reputation product outside consumer reporting rules.
4. Z402 intellectual property: prior art search and provisional filing before public description.
5. On-ramp and off-ramp: provider licensing, KYC responsibility, and merchant of record.
6. Verification data: data protection obligations of RNS and other issuers; confirm the on-chain record stays outside personal-data scope.

13.3 Governance

Governance follows the V1 path of progressive decentralization:

1. **Governance Association.** Makes decisions on upgrades, ecosystem development, and parameters, with advisory councils and signaling votes.
2. **Limited-Role DAO.** Token holders vote on proposals. The Association executes.
3. **Full DAO.** Token holders control upgrades and parameters on chain.

V2 controls from day one:

- Every privileged contract (staking, paymaster, identity registries, settlement) is owned by a timelock controlled by a multisig.
- A guardian can pause with bounded powers. A paymaster pause expires on its own. A staking pause stops new stake while exits stay open.
- Owed rewards cannot be withdrawn by the owner.
- Parameter changes are on-chain events visible in the explorer.
- Chain upgrades follow ZK Stack upgrade procedures through the timelock.
- Off-chain operator keys rotate through governance.

14

ROADMAP

Roadmap



Revolution ships in six phases. A phase is done when its exit test passes, not on a date. Phase 1 is under way: the foundation is built and tested in the V2 repository, and Virtus (V2 testnet) is being stood up.

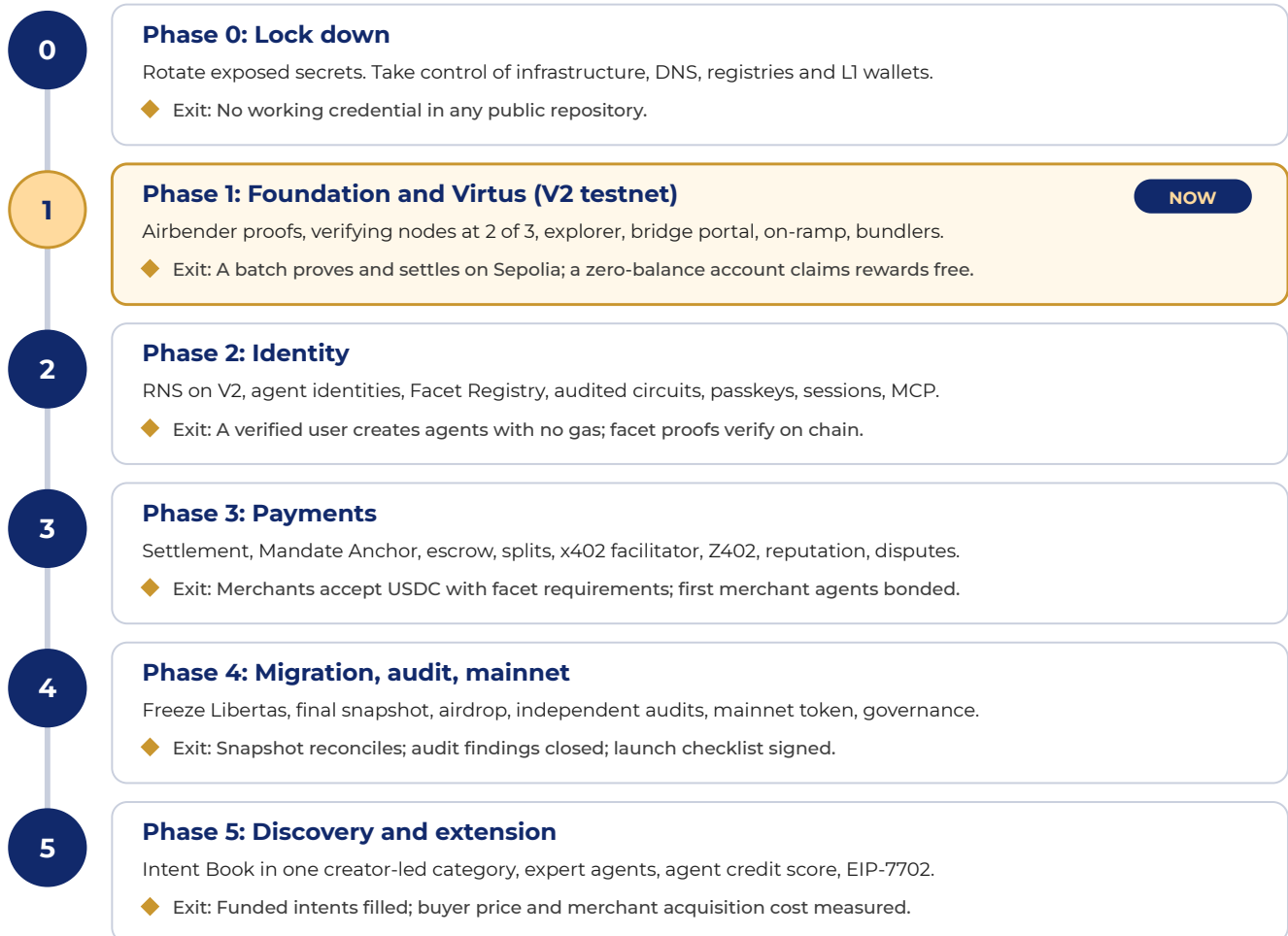


Figure 9. V2 roadmap. A phase is done when its exit test passes, not on a date.

Identity ships before payments, and payments before discovery. Identity must sit at the point of transaction before money moves through it.

14.1 Phase 1 status

Built and tested in the V2 repository: a ZKsync OS devnet from one script, staking V2 with time-based emissions, a keeper, an API, and a node agent; EntryPoint v0.8 at its canonical address; ERC-7579 Nexus accounts; the Cornerstone Paymaster and Budget; settlement fee routing; and the Libertas snapshot tool. 51 contract tests pass, including four stateful invariants.

Workstream	Phase 1 deliverable
L1 ecosystem	ZK Stack ecosystem contracts on Sepolia, new chain ID, operator keys in a KMS
Token	Test REVO on Sepolia as the base token; test USDC bridged; mainnet REVO and vesting contracts prepared for governance approval
Prover	Airbender FRI and SNARK on a rented GPU host; fake provers off
Verifying nodes	Batch verification at 2 of 3 signers; node heartbeats live
Explorer	Public ZKsync OS compatible explorer with batches, settlement status, user operations, and staking
Bridge portal	REVO and USDC deposits and withdrawals with deposit gas sponsorship
On-ramp and off-ramp	Licensed provider for card and bank purchase into a smart account, chosen after legal review
Bundler	Two self-hosted Pimlico Alto instances
Sponsorship	Verifying paymaster with signed vouchers from the policy service

15

REFERENCE

Open decisions, glossary, and disclosures



15.1 Open decisions before mainnet

Read this list before relying on any number in this document.

#	Decision	Why it matters	Proposed path
1	Emission start point	Sets how much of the 250M remains at V2 genesis	Choose resume by amount, resume by calendar, or restart at era 1
2	Migration scope	Sets snapshot processing and user notices	Decide before the Libertas freeze
3	Split rule notice	Libertas paid the technical-draft split; V2 pays the published rule	Confirm with the Association; publish a notice
4	Slashing of node stake	Jailing stops rewards but does not slash	Specify slashing tied to batch verification
5	Settlement fees to operators only	Balance between roles as emissions decay	Launch as decided; revisit with fee data
6	Cornerstone Budget size	Too low stalls free calls; too high dilutes stakers	Measure on Virtus (V2 testnet)
7	L2 gas fee destination	Gas fees go to the operator address, not stakers	Decide whether to forward to notifyFees
8	Node cap counts exited nodes	Churn could exhaust 250 slots	Reuse slots on exit or size for churn
9	Paymaster validation on ZKsync OS	The voucher design depends on it	Confirm account abstraction and ERC-7562 tracing behavior
10	Load model inputs	Batch cadence, gas price, and capacity figures differ across planning documents	Replace with a Virtus benchmark under real proofs
11	Native USDC	Requires Circle support for the chain	Launch with bridged USDC, labeled clearly
12	On-ramp provider	Licensing and KYC responsibility	Legal review, then select
13	Recovery model for verified parents	The hardest consumer question in the design	Specify before Phase 2 exit

#	Decision	Why it matters	Proposed path
14	Credential format for circuits	Issuers must emit Merkle commitments with revocation roots	Agree with each issuer
15	Netting ratio	The 100-to-1 planning ratio drives cost and load	Validate in the ACommerce proof of concept
16	Account build and audit coverage	Nexus is compiled at different optimizer settings than the audited deployments	Confirm coverage with the auditor
17	Competition	Kite, Tempo, Arc, and card-network registries are live and funded	Interoperate; be the identity and settlement layer those rails call

15.2 Glossary

Term	Meaning
ACommerce	Commerce conducted agent to agent under a human's mandate
Agens	Agent. An agent identity under a verified parent, acting under a Mandatum (delegated authority)
Agnomen	Earned reputation. Standing earned through conduct
Airbender	The ZKsync OS prover
Batch verification	ZKsync OS feature in which external nodes sign each batch before commit
Cognomen	Subname. An entity, treasury, or project name under a root (proposed)
Cornerstone service	A call that is free to the caller, paid by the Cornerstone Budget through the paymaster
Cornerstone Budget	The share of staking emissions set aside to pay for cornerstone calls
Creator Pool	A creator's stake plus its fans' stake, delegated to one node
Facet	One provable attribute of an identity, verified with a zero-knowledge proof
Intent Book	On-chain book of funded buyer intent
Libertas	V1 network. The EraVM testnet
Mandatum	Delegated authority. Bounded, revocable authority a principal delegates to an Agens (agent)
Nomen	Name. A <code>.revo</code> name issued by RNS
Potestas	Permissions. What a principal may do, set by its Sigillum (verified identity) level and account class
Praenomen	Agent name. The label of an Agens (agent) under a Nomen (name)
RNS	Revolution Name Service
Sigillum	Verified identity. The identity behind a Nomen (name), with its level and facets
Soft consensus	The V1 stake-weighted priority counter that picked one reward winner per batch; replaced in V2 by continuous accrual

Term	Meaning
Verifying node	A staked, approved operator that runs an external node, co-signs batches, and sends heartbeats
Virtus	V2 testnet. The short-lived V2 public testnet
Z402	Private proof-of-payment extension to x402
ZKsync OS	The EVM-equivalent execution environment of the current ZK Stack

15.3 Disclosures

This whitepaper describes software under development. It is not an offer to sell, or a solicitation to buy, any token or security in any jurisdiction. Nothing in it is investment, legal, or tax advice. Parameters marked *governance-set* are proposals and may change. Cost and performance figures are planning estimates, not commitments. Reward amounts depend on network participation and are not promised. Features described as specified or planned do not yet exist on mainnet. Participation in staking carries risk, including loss of value of REVO.

15.4 References

- Revolution Network Whitepaper (2025)
- REVO Chain Staking Technical Whitepaper, draft v1
- ACommerce on Revolution, overview and Addendum A (September 2026)
- Revolution V2 Vocabulary Guide (September 2026)
- Xiong et al., ERC-8004 reputation Sybil study, arXiv (June 2026)
- ZKsync OS server: github.com/matter-labs/zksync-os-server (v0.23.0)
- ERC-4337, ERC-7562, ERC-7579, ERC-7677, ERC-7710, EIP-7702: eips.ethereum.org
- EntryPoint v0.8: github.com/eth-infinitism/account-abstraction tag [v0.8.0](#)
- Biconomy Nexus: github.com/bcnmy/nexus

R REVOLUTION

Return network value to the people who create it.

ECOSYSTEM



revolutionnetwork.io

X @revochain · Telegram the_revolution_network

This whitepaper describes software under development. It is not an offer to sell, or a solicitation to buy, any token or security in any jurisdiction, and it is not investment, legal or tax advice. REVO is a utility token used for gas, staking, bonds, settlement fees, deposits and governance. Parameters marked governance-set are proposals. Cost and performance figures are planning estimates. Reward amounts are not promised.