NVNM Chain

A sovereign layer 1 blockchain for verifiable agent attestation

Accountability infrastructure for agentic activity

Whitepaper

Albert Berdellans

Global Head of Artificial Intelligence at Inveniam

September 2026

NVNM CHAIN

Whitepaper

Accountability infrastructure for agentic activity

Albert Berdellans
Global Head of Artificial Intelligence at Inveniam

Version 1.1 · Revised September 2026

Originally published May 2026

Abstract

NVNM Chain is accountability infrastructure for agentic activity. Every AI agent has a beneficial owner, and the owner is the accountable party for what the agent does. NVNM Chain is the substrate that makes that accountability mechanically traceable: every agent operates from a unique persistent onchain wallet, and the inputs, processing, and outputs of consequential agent actions are recorded as cryptographic attestations that any authorized counterparty can independently verify. In single-party or trusted-counterparty contexts, agentic workflows can be verified internally and trust assumptions are tractable. In multi-party contexts where counterparties cannot rely on each other to adjudicate the truth of inputs, processing, or outputs, immutable attestation between parties is the only mechanism that lets each side directly verify what the others did without anyone serving as intermediary truth-arbiter.

The chain is a sovereign Layer 1 blockchain, EVM-compatible and purpose-built for this workload. It provides a native anchoring module exposed through a fixed-address Ethereum Virtual Machine (EVM) precompile, hierarchical role-based access control suited to institutional data governance, and a typed agent access surface implementing the Model Context Protocol (MCP). Execution runs on Reth, and consensus runs on Commonware Simplex, delivering deterministic finality in approximately 500 milliseconds. Gas is paid in stablecoins under a fee model enshrined at the protocol layer, giving institutional and agentic workloads predictable dollar-denominated transaction costs without requiring any participant to hold a volatile asset.

Financial markets are the first developed instantiation of the pattern. Inveniam.io, the decentralized data management platform for private market assets, is the canonical first integrated data layer at launch. The same primitives apply to multi-party clinical trials, supply chain attestation across non-cooperating parties, regulated AI compute, and any other multi-party agentic workflow where attestation must be verifiable by counterparties without a vendor as intermediary, and must outlive any specific vendor. After publication, this document is anchored to NVNM Chain. The anchoring record, including the document checksum, the registry, the transaction, and the block height, is published at nvnmchain.io. A reader can verify integrity by recomputing the document's checksum and confirming it matches the anchored record. Definitions of recurring terms appear in the Key Terms section near the end of the paper.

1. Executive Summary

Two trillion-dollar trends are converging in financial markets. AI agents are crossing from analytics into transaction execution, and the world's largest asset classes are being represented on programmable rails[1]. Each of these trends, taken alone, is reshaping how capital is allocated. Taken together, they create a requirement that does not yet have an answer: every meaningful action an agent takes must be defensible to a regulator, an auditor, or a counterparty who was not in the room when the action was taken. Without that defensibility, agentic capital markets cannot operate at institutional scale.

The same problem exists wherever multiple parties depend on agentic outputs and cannot rely on each other to adjudicate the truth. Multi-party clinical trials, supply chain attestation across non-cooperating parties, regulated AI compute, and inter-organizational regulatory reporting all share the same structure: handoffs are tractable in single-party or trusted-counterparty contexts. However, in multi-party non-cooperative contexts, immutable attestation between parties is the only

mechanism that lets each side directly verify the others' work without an arbiter sitting in the middle. Financial markets are the most developed instance. The pattern is general.

The current state of the world has the data, the agents, and the rails[2,3]. What the current state lacks is a verification layer that is independent of any individual data vendor, agent operator, or counterparty. NVNM Chain has been created to be that layer.

NVNM Chain is a sovereign Layer 1 blockchain purpose-built to record cryptographic proofs of agentic activity. Each meaningful action an agent takes can be summarized by three artifacts: a checksum of the source data the agent consulted, a checksum of the processing it applied, and a checksum of the state of the world it acted upon. NVNM Chain anchors these artifacts as records in permissioned registries, with role-based access controls suited to institutional data governance, facilitating verification queries by any party empowered to consume them. The underlying source data is never published. Only the cryptographic fingerprints traverse to the chain.

Three architectural choices distinguish the chain. First, anchoring is implemented as a native chain primitive, exposed through an EVM precompile at a fixed canonical address. The precompile is Turing-incomplete, so its execution cost is bounded and predictable, and anchoring never competes with general-purpose contracts for execution time. Second, the permission system is hierarchical and operates at three scopes: record, registry, and global. This admits institutional governance patterns that a flat owner-only model cannot. Third, the agent access surface is typed and authenticated, implementing the Model Context Protocol. The server validates inputs, prepares unsigned transactions, and never holds private keys. Signing happens externally, where institutional security architecture already terminates.

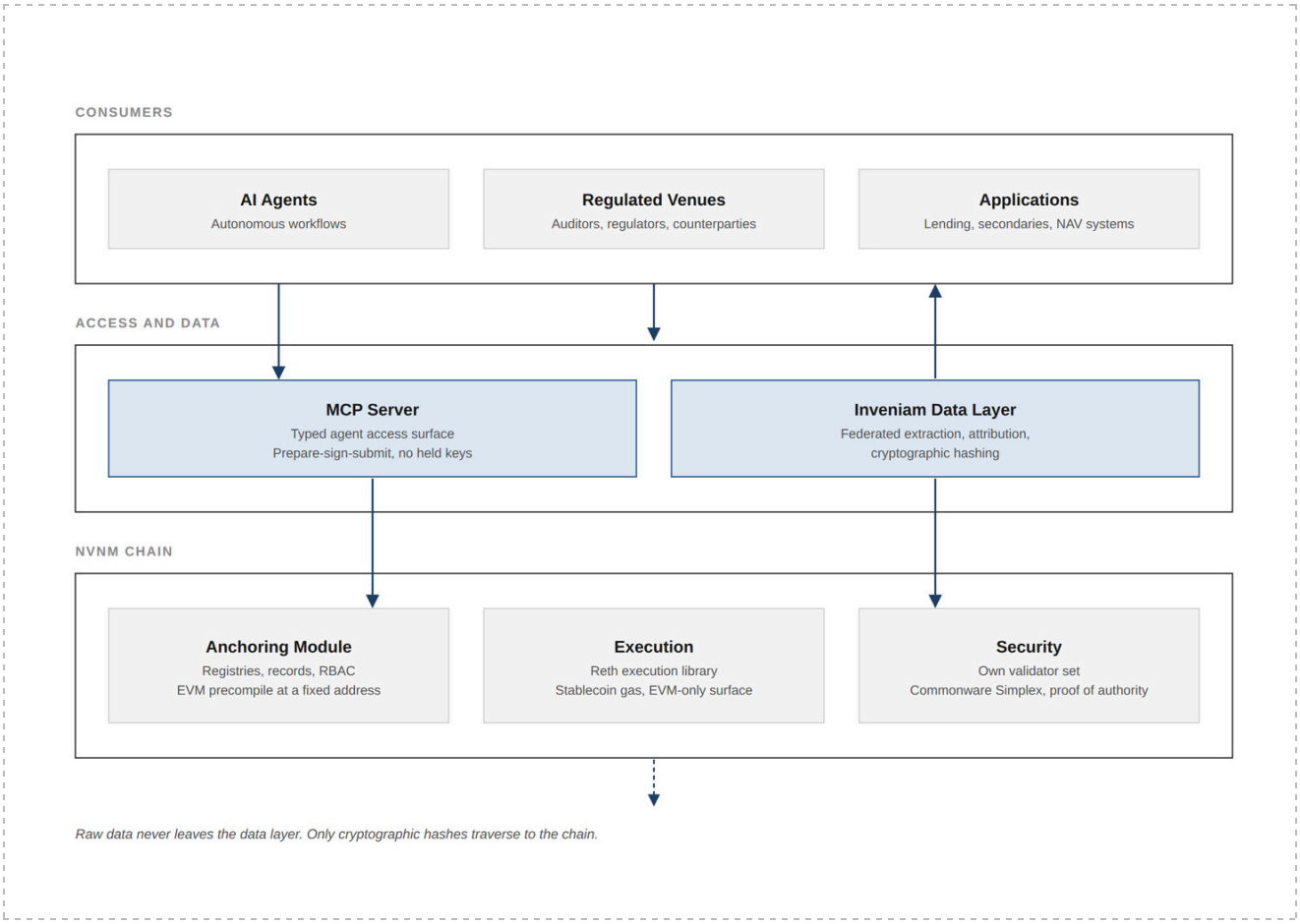


Figure 1. NVNM Chain in three layers. *Consumers interact with the chain through the typed access surface. Source data remains in the operator's environment; only cryptographic hashes traverse to the chain. Lending, secondaries, clinical trial cohort production, supply chain verification, and similar workflows are protocols that compose on top of the attestation substrate.*

The chain operates its own permissioned validator set under proof of authority at launch, with consensus on Commonware Simplex. Gas is paid in stablecoins under a fee model enshrined at the protocol layer, providing predictable transaction costs to institutional and agentic workloads that cannot tolerate volatile gas pricing. Governance and value-accrual mechanisms beyond the Foundation's launch authority are out of scope for this whitepaper and will be described in a separate companion document published alongside relevant feature announcements.

Why this matters now

Three independent forces are pushing toward the same requirement. Regulators of advanced AI compute are tightening reporting, recordkeeping, and audit obligations.[4] Institutional tokenization is reaching scale at which DeFi protocols depend on continuously refreshed, attributable price signals for assets whose underlying data has historically lived in PDFs.[5] And agents acting on behalf of operators are increasingly making decisions whose accountability needs to be machine-readable, not merely auditable in a quarterly review.[6] NVNM Chain answers all three with a single primitive: cryptographic anchors that outlast any specific vendor and are programmatically verifiable by any authorized consumer.

2. The Agentic Provenance Problem

> *Reliance upon the word of others plays a crucial role in the economy of knowledge, though the extent and depth of this reliance have gone largely unrecognized.*
>
> — C. A. J. Coady, Testimony: A Philosophical Study

A modern multi-party workflow is a chain of testimony. A fund administrator publishes a Net Asset Value (NAV). A bank underwriter relies on borrower-provided financials. A clinical trial sponsor submits aggregated cohort data to a regulator. An export-controlled chip operator declares its workload to a government inspector. A manufacturer declares country-of-origin to a customs inspector. In each case, downstream parties act on data they did not produce, processed by parties they cannot directly observe, describing conditions they cannot independently reconstruct. The system functions because participants build reputational capital over time, and because regulators, auditors, and binding contracts impose consequences for misrepresentation. The system does not function because the testimony itself is verifiable. It functions because the testifier is accountable.

AI agents operating in this environment change the second condition without changing the first. An agent that reads a NAV and adjusts a collateral parameter at machine speed leaves its operator legally accountable under existing principal-agent doctrine; the operator can be deposed, sued, and sanctioned. What the operator cannot easily do is demonstrate what the agent saw, what it computed, and what state the world was in at the moment it acted. The records that would settle these questions are kept, if they are kept at all, in systems the operator controls and has every incentive to curate after the fact. As agents act faster and across more organizational boundaries, the gap between what an operator is liable for and what an operator can evidentially defend widens.

In single-party or trusted-counterparty contexts, the gap is tractable. Agentic handoffs within one organization can be verified by internal logs and process audits. Where two counterparties have established trust, signed bilateral records can carry the weight. The gap becomes structural in multi-party contexts where counterparties cannot rely on each other to adjudicate the truth of inputs, processing, or outputs[7]. There is no neutral party available to play notary. Choosing one introduces a vendor as the trust anchor, and the choice itself is a negotiation that one party has to concede. Immutable attestation between parties is the only mechanism that lets each side directly verify what the others did, without anyone serving as intermediary truth-arbiter.

Origin, Process, and State

The word provenance is used loosely in industry literature. NVNM Chain uses it precisely. Provenance is the union of three properties:

- **Origin.** Where a data point came from. The source document, the page, the reporting party, the timestamp of issuance.
- **Process.** What transformed the source data into the form the agent or counterparty consumed. The extraction model, its version, its parameters, the validation steps applied.
- **State.** What snapshot of the dataset the consumer acted on. Not just the value, but its position in a versioned history.

A claim of provenance that does not address all three is incomplete. A signed source document with no record of how it was processed says nothing about how the agent read it. A processing trace with no anchor to a source artifact can be reconstructed to fit any conclusion. A snapshot with no history offers no protection against silent revision.

The three primitives compose. Each on its own is useful in narrow contexts. Together, they make a verifiable claim: this specific data came from this specific source, was transformed in this specific way, and was in this specific state when the action that depended on it was taken.

Previous infrastructure attempts

Several classes of infrastructure address adjacent problems. None of them, alone or in combination, addresses the problem stated above.

Public oracle networks

Public oracle networks deliver high-frequency price feeds and similar public data points to onchain consumers. They solve a real problem and they solve it well. The problem they solve is not the same problem. Document-level provenance for private market data, multi-party clinical trial data, supply chain attestations, or proprietary banking records is not a feed of public prices that can be aggregated from third-party sources. The data is structurally not public.

General-purpose blockchains

Any general-purpose chain can store a hash. Most major chains have. What general-purpose chains lack is the operations multi-party agentic contexts require: agent identity tied to specific wallets that travel with their action history, scope-of-authorization enforcement at the access surface, role-based access control over registries with multiple administrators, and typed access tooling that constrains agents to audit-shaped operations. These are not application-layer features that compose well from generic primitives. Building them per application produces a different security posture for every consumer of the chain. NVNM Chain provides them as native chain primitives, so an institution's compliance team reasons about access control once for the whole chain rather than re-auditing it per integration.

Centralized data warehouses

A centralized data warehouse can prove integrity of its records to itself. It cannot prove integrity to a counterparty without exposing the records, and it cannot prove integrity at all if the warehouse operator has been compromised or has rewritten history. The trust model collapses to trust in the warehouse operator, and that operator is precisely the party whose claims need verification.

Off-chain attestation services

Several established firms offer attestation services that produce signed records of document existence or integrity. The major audit firms among them have credible institutional longevity and have operated public key infrastructure reliably for decades. The issue is not durability. The issue is that any single attestation service introduces a vendor as the trust anchor between non-cooperating parties. In a two-party transaction this can work if both parties already trust the vendor. In a multi-party non-cooperative context, choosing a vendor is itself a negotiation, and accepting a vendor is itself a concession. Cryptographic anchoring on a public ledger removes the vendor from

the trust graph entirely. Counterparties can still use attestation services for the underlying signatures or audit work. They no longer need to agree on a single vendor as the durable record-keeper.

Architectural requirements

A verification layer fit for multi-party agentic workflows must satisfy four conditions. It must be independent of any individual operator, vendor, or counterparty. It must preserve data sovereignty: the underlying data must remain with its lawful custodian. It must be programmatically verifiable by machine consumers without human translation. And it must outlive the specific vendors that populate it today.

The cryptographic primitive that satisfies these conditions is more than thirty years old. Haber and Stornetta proposed linked hash chains for digital document timestamping in 199.[18] The contribution of NVNM Chain is to combine that primitive with the institutional governance, the agent access patterns, and the operational discipline required for multi-party agentic workflows in regulated and adversarial settings.

3. NVNM Chain: System Overview

NVNM Chain is organized in three layers: the chain itself, the data infrastructure that produces what gets anchored, and the access surface through which agents and applications interact with both.[9] The chain provides the attestation substrate. Specific applications including lending, secondaries, clinical trial cohort production, supply chain verification, regulated AI compute reporting, and similar workflows are protocols that compose on top of the substrate. The chain does not implement any of these workflows itself.

Functional definition

NVNM Chain is a sovereign Layer 1 blockchain optimized for cryptographic anchoring of agent-produced attestations. It is EVM-compatible, so standard Ethereum tooling works against it without modification. Execution runs on Reth as an execution library, and consensus runs on Commonware Simplex, delivering deterministic finality in approximately 500 milliseconds. Its native execution environment exposes a purpose-built anchoring module through a fixed-address EVM precompile. Gas is paid in stablecoins, giving institutional and agentic workloads predictable transaction costs denominated in US dollars rather than in volatile crypto-native tokens.

Scope and non-goals

NVNM Chain is a purpose-built attestation chain rather than a general-purpose smart contract platform. It is not a payments rail and it does not compete with payments-focused infrastructure. It is not a tokenization platform. The chain has a focused mandate: anchoring verifiable claims about data, processing, and decisions in multi-party agentic workflows. Other infrastructure tokenizes assets. Other infrastructure moves value. NVNM Chain provides the substrate that lets counterparties verify each other's work directly, without an intermediary adjudicating truth. Anything outside that mandate is intentionally absent from the design.

Three architectural pillars

The system is organized in three layers. Section 4 develops each in detail.

The Anchoring Module

The Anchoring Module is the chain's defining technical contribution. It implements registries, records, and a hierarchical role-based access control system as native chain primitives. It exposes a single interface: an EVM precompile at a fixed canonical address. Every caller, whether a Solidity contract or an external client, reads and writes the same anchoring state through that one surface.

The Data Layer

The Data Layer produces what gets anchored. At launch, the canonical first integrated data layer is Inveniam.io, the decentralized data management platform for private market assets. Inveniam.io connects to existing enterprise storage environments, structures unstructured documents into machine-readable form, attributes every extracted data point to a specific source, and produces the cryptographic hashes that are submitted for anchoring. The chain itself is platform-agnostic. Other data providers, in finance and in other domains, can integrate by producing valid attestations to registries they control.

The Agent Access Surface

The Agent Access Surface is the typed, authenticated, authorization-aware interface through which AI agents and developer tools interact with the chain. It is implemented as a Model Context Protocol (MCP) server hosted at mcp.nvnmchain.io. The server validates inputs against published schemas, enforces role-based access control, prepares unsigned transactions for state-changing operations, and never holds private keys. Signing happens externally, through whatever key management the operator already trusts.

Agent Identity

Each agent operating against the chain has a unique persistent onchain wallet, never shared with another agent. The agent's actions are forever tied to that wallet. A counterparty consuming an agent's outputs can query the wallet on chain and verify the agent's full history of actions independently. The lifecycle implications are direct: an operator can pre-evaluate a counterparty's agent based on its prior onchain history before integrating with it; revocation is the revocation of authorizations on a single wallet; and audit across multiple counterparties is a query against a known address. The registry mechanism that ties agent wallets to their operators and their declared scopes of authorization is described in Section 5.4 as Know Your Agent (KYA).

Layer Composition

A typical end-to-end flow proceeds in five steps. First, source data lives in an operator's existing storage. Second, the Data Layer extracts the relevant content, attributes it, and produces a cryptographic hash. Third, an agent or developer tool requests an anchoring operation through the MCP server. Fourth, the MCP server validates the request and prepares an unsigned transaction; an external signer signs it; the signed transaction is broadcast to the chain. Fifth, the chain executes the precompile call, persists the record, and returns the verification handle. Downstream consumers can independently query the chain to confirm that a specific record exists, that its checksum matches a document they hold, and that its metadata is consistent with their expectations.

Security model

NVNM Chain operates its own validator set . Block production, finality, and the economic security of the network are properties of this chain itself. The set is permissioned at launch and expands in stages. Section 4.1 develops the model in detail, including the consensus protocol, the design rationale, and the failure modes the design must contemplate.

Gas and fees

Gas is paid in stablecoins. Predictability of transaction cost matters more than nominal cost for institutional workloads, and denominating fees in a dollar-stable unit gives a treasurer and an agent the same basis for reasoning about cost. The fee model is enshrined at the protocol layer rather than implemented as an application-level abstraction, so no participant is required to acquire or hold a volatile native asset in order to transact. The mechanism is described in Section 4.6.

4. Technical Architecture

4.1 Consensus and Security Model

NVNM Chain runs Commonware Simplex , a Byzantine fault-tolerant consensus protocol providing deterministic finality . Once a block is finalized by the validator set it is final: there is no probabilistic settlement window, and a consumer that has observed finality faces no reorganization risk. Safety holds in the absence of more than one third Byzantine validators, the threshold established by Lamport, Shostak and Pease and refined for production use by Castro and Liskov . For an anchoring workload this is the property that matters most. A counterparty verifying a record needs a definite answer about whether that record exists, and it needs the answer within a bounded time.

Finality lands at approximately 500 milliseconds, measured on the development network. The institutional chains built for comparable workloads operate in the same range, and the design reasoning is the same across all of them: an institution integrating verification into a transaction workflow needs settlement that completes inside the window a human reviewer or an autonomous agent is willing to wait. Finality at this speed also keeps the anchoring path usable inside a synchronous request, so an agent can anchor an attestation and proceed rather than polling for confirmation.

The validator set is permissioned and operates under proof of authority at launch. Every operator is an identifiable party holding a position it acquired itself, which makes responsibility for the network attributable to named parties. The set expands in stages, and the conditions governing each expansion, together with the path to broader participation, are set out in the companion tokenomics whitepaper.

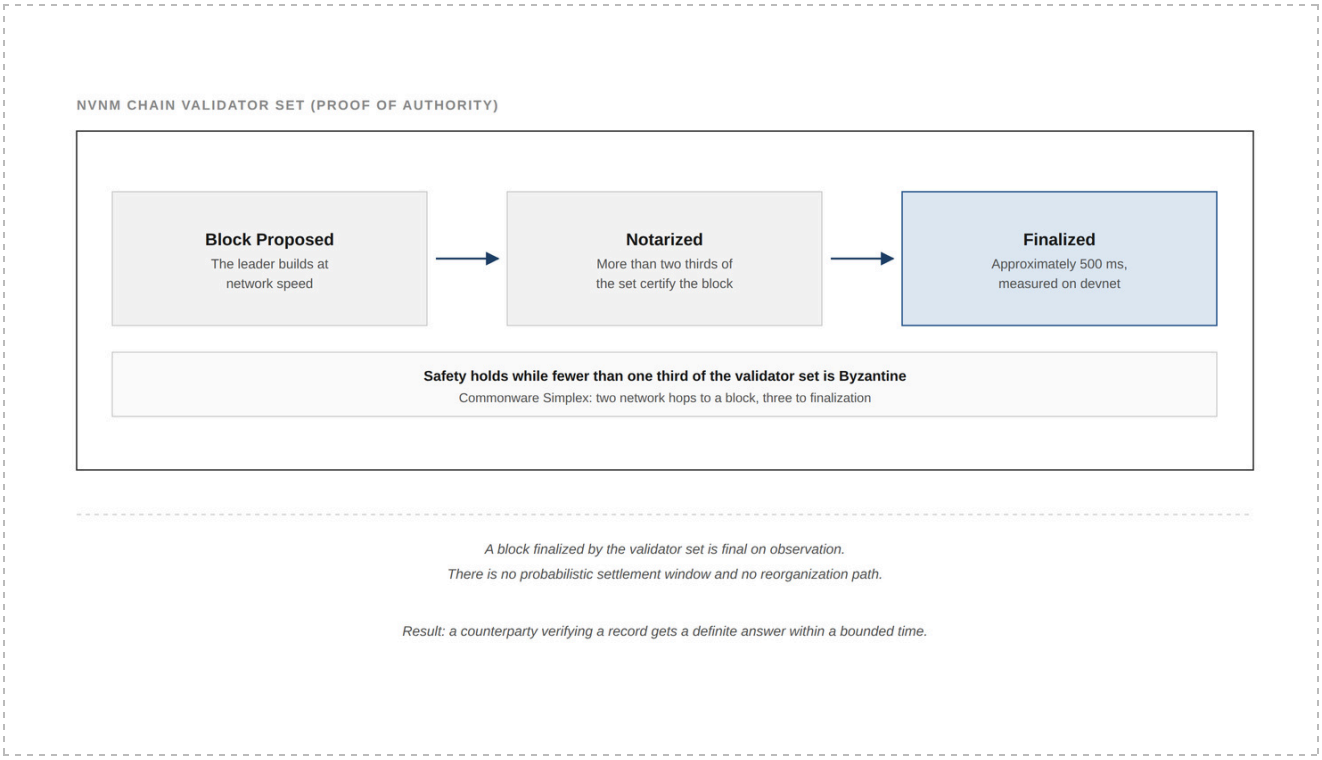


Figure 2. Consensus and finality. *A block finalized by the validator set is final on observation, with no settlement window and no reorganization path.*

Design rationale

Three architectures were evaluated against the requirements of an attestation chain.

- An Ethereum rollup architecture was rejected because the chain's workload is not compute-bound and the dependency on a separate settlement and data availability layer adds latency and cost without commensurate benefit. The throughput envelope of mainstream rollups is also less suited to high-volume anchoring than a purpose-built chain.
- A general-purpose Layer 1 with anchoring implemented in application-layer contracts was set aside because anchoring would compete with unrelated contract execution for block resources, and because the access-control semantics institutional governance requires would have to be rebuilt and re-audited for every integration.
- A permissioned-only network was set aside because it defeats the purpose of public verification. A counterparty who can independently verify an anchor against a public ledger is not in the same position as a counterparty who can verify it only against a private one. The institutional value of the chain depends on the public verifiability.

A sovereign Layer 1 gives NVNM Chain independent execution, a dedicated anchoring module, predictable gas economics, and an upgrade path the chain governs itself. It also places the full burden of operation on the chain. Validator selection, security auditing, dependency maintenance and operational hardening are NVNM Chain's own responsibilities. Failure modes are documented honestly: a validator set of this size concentrates liveness risk in a small number of operators, and network availability depends on their operational discipline; upstream defects in the execution or consensus libraries become NVNM Chain's maintenance burden; and the composition of the set at launch places responsibility with a small group of identifiable parties. Each is named here because a

candid description of failure modes is more useful to an institutional integrator than a list of nominal performance figures.

4.2 Execution Environment and the Fee Model

NVNM Chain executes on Reth, a modular Ethereum execution client used as a library rather than as a standalone node[15]. Reth gives the chain current Ethereum compatibility[17,18], so wallets, indexers, development frameworks, block explorers and contract tooling work against NVNM Chain the way they work against any EVM network. A team that has built tooling for Ethereum can target NVNM Chain without rewriting its stack.

The EVM is the chain's only execution surface. This is a deliberate simplification. Native chain capabilities, including the Anchoring Module described in Section 4.3, are exposed as Turing-incomplete EVM precompiles at fixed canonical addresses. Turing incompleteness is the operative constraint: a precompile with bounded execution carries a predictable cost, cannot enter an unbounded loop, and cannot be made to compete with general contract execution for block resources. An institution integrating with the chain reasons about one address space, one signing scheme and one access-control model rather than reconciling several.

The fee model is enshrined at the protocol layer. Transactions are paid in eligible stablecoins, and the protocol performs conversion and settlement in the background. A participant anchoring an attestation holds the stablecoin its treasury function already holds, submits the transaction, and pays a cost quoted in dollars. Section 4.6 describes the mechanism and the role of the chain's settlement currency.

4.3 The Anchoring Module

The anchoring module organizes data into registries containing records. A registry is a logical namespace owned and administered by a single party or a small group of parties. A record is a single attestation entry within a registry, identifying an artifact by its cryptographic checksum, recording a URI by which authorized parties can fetch the artifact, and carrying structured metadata about its lifecycle state[19,20,21].

Data model

A registry has the following fields: an internal identifier assigned at creation time, a human-readable name, a description, a creator, and a creation timestamp. The creator becomes the registry's administrator at creation. A registry conceptually corresponds to an organizational namespace. A bank's compliance attestations live in one registry. A fund administrator's NAV anchors live in another. The boundaries of a registry are the boundaries of an institutional governance domain.

A record has the following fields: a registry reference, an internal record identifier, a version index, the cryptographic checksum, a checksum algorithm identifier, a URI pointer, a status, an isLatest flag, a timestamp, and structured metadata. The default checksum algorithm is SHA-25625. The checksum is the primary identity of the record. A given checksum can be registered exactly once per registry. Adding the same checksum to the same registry again is a no-op.

Versioning

Records are append-only. Errors in mutable metadata such as descriptions, tags, or status fields are correctable by the registry administrator or an editor. Errors in the checksum or the checksum algorithm identifier are not correctable: a wrong checksum is a different record. Correcting a wrong checksum requires archiving the original record and creating a new one. This is a deliberate property. The checksum is the record's identity. Allowing it to change would defeat the integrity guarantee.

Each record carries a single immutable checksum. An operator can express version relationships across records by including version metadata in the record's metadata field. The chain does not enforce version semantics. It preserves the records and their order, and downstream verifiers interpret version relationships from metadata according to the operator's conventions. This pattern supports artifacts that legitimately evolve, such as a fund's NAV reported at successive valuation dates, while preserving the integrity of every prior version.

Interfaces

The anchoring module is exposed through a single interface: an EVM precompile at the fixed canonical address 0x0000000000000000000000000000000000000A00. The precompile is implemented natively inside the chain binary. It does not consume gas at the rate of an equivalent Solidity implementation, and its gas cost is predictable across protocol upgrades. The functions exposed are addRegistry, addRecord, updateRecordStatus, records (query), registries (query), grantRole, and revokeRole. Method signatures and parameter types appear in Appendix B.

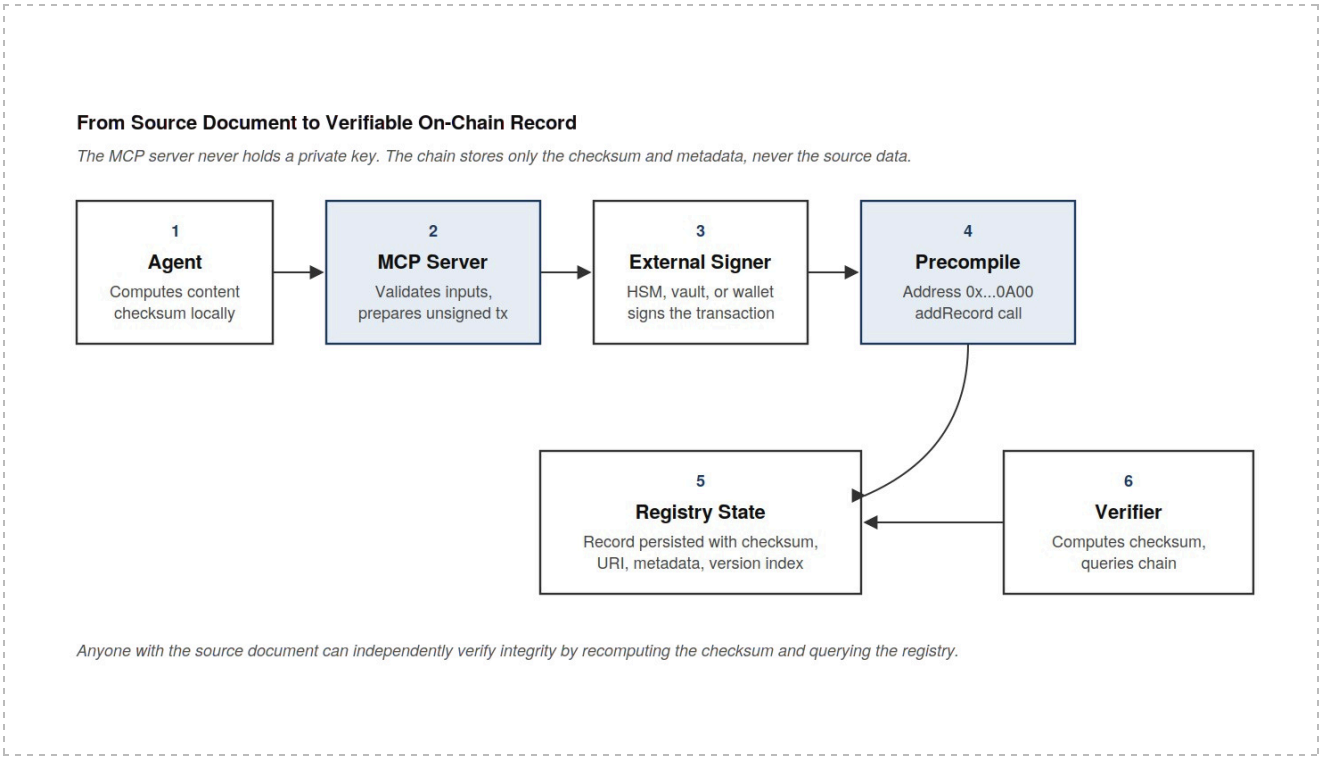


Figure 3. Anchoring data flow. *A write runs from the agent's locally computed checksum through validation, external signing, and the precompile into registry state. The anchoring module enables a verifier to recompute the checksum from its own copy of the document and to query that same registry state.*

Onchain and off-chain boundaries

The onchain footprint of an anchor is small and intentional. The chain stores the checksum, the checksum algorithm, the URI reference, the status, the timestamp, and the structured metadata. The chain does not store the underlying data. It does not store extracted content. It does not store any artifact that could disclose proprietary or sensitive information. The footprint is on the order of a few hundred bytes per anchor regardless of the size of the underlying data.

This property is load-bearing. It is what allows institutional operators with strict data sovereignty requirements to anchor verifiably without surrendering control of the underlying records. A bank can anchor proof of regulatory compliance attestations without exposing the customer-level data the attestations summarize. A clinical trial sponsor can anchor proof of cohort definitions without exposing patient records. A regulated AI infrastructure operator can anchor proof of compliance events without disclosing the proprietary workloads that produced them.

4.4 Hierarchical Role-Based Access Control

A flat owner-only access model is too coarse for institutional data governance. Most institutional registries have multiple operators, with different scopes of authority over different subsets of records, and the scopes change over time as personnel and partnerships evolve. The anchoring module supports this through a three-level role-based access control system.

Permission levels

Permissions can be granted at three scopes. A record-level grant attaches to a specific checksum within a specific registry. A registry-level grant applies across the entire registry. A global grant applies across all registries. A grant at any scope is sufficient for access. Absence of a grant at every applicable scope denies access. The protocol exposes grantRole and revokeRole; there is no separate deny primitive.

Roles

Two roles are defined at the protocol level. An admin can grant and revoke roles, archive records, and update mutable metadata. An editor can write records, archive records, and update mutable metadata, but cannot modify roles. Read access to record metadata is public for queries by checksum or by registry-and-record identifier. The data the metadata refers to is access-controlled separately, off chain, by whatever system holds it.

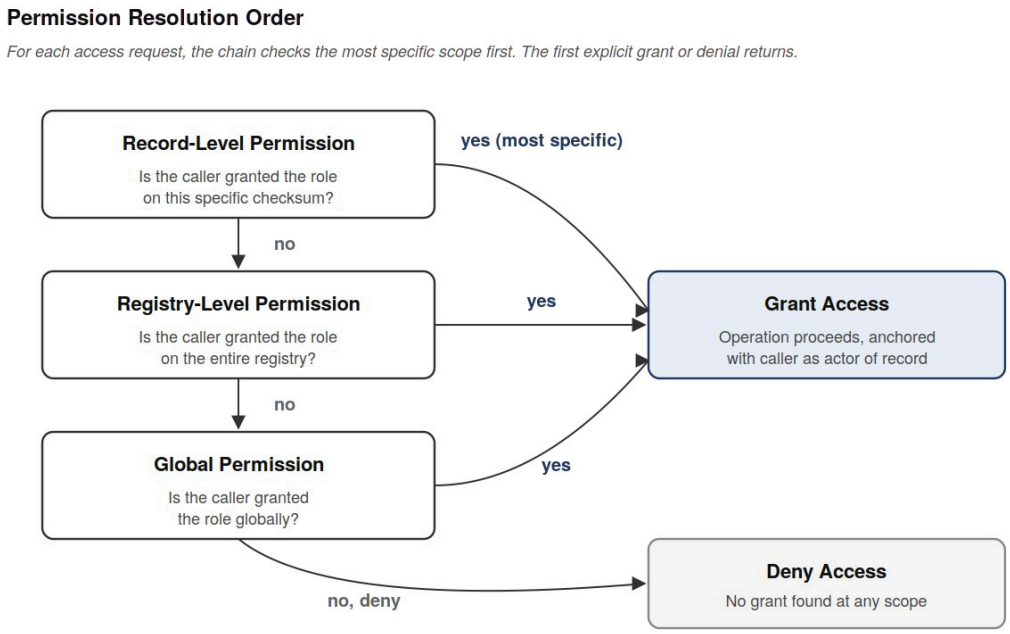


Figure 4. Permission resolution. *A grant at any scope is sufficient for access. Absence of a grant at every applicable scope denies access.*

Why this model

The model is deliberately small. Two roles, three scopes, one resolution rule. A more elaborate model would require operators to reason about subtle interactions during incident response. A simpler model would not capture the patterns that institutional governance actually uses. The chosen design admits the patterns that matter and rejects the rest.

4.5 The Proof Stack: Origin, Process, State

> *The whole obsession of inscriptions is to make the world stay where you have put it.*
>
> — Bruno Latour, on inscription as a means of holding shape across contexts

The proof stack is the conceptual framework that the anchoring module implements. It distinguishes three primitives. Each attests to a different property of the same underlying data. Each is supported by specific patents in the Inveniam portfolio[23,24]. The primitives compose: a complete attestation references all three.

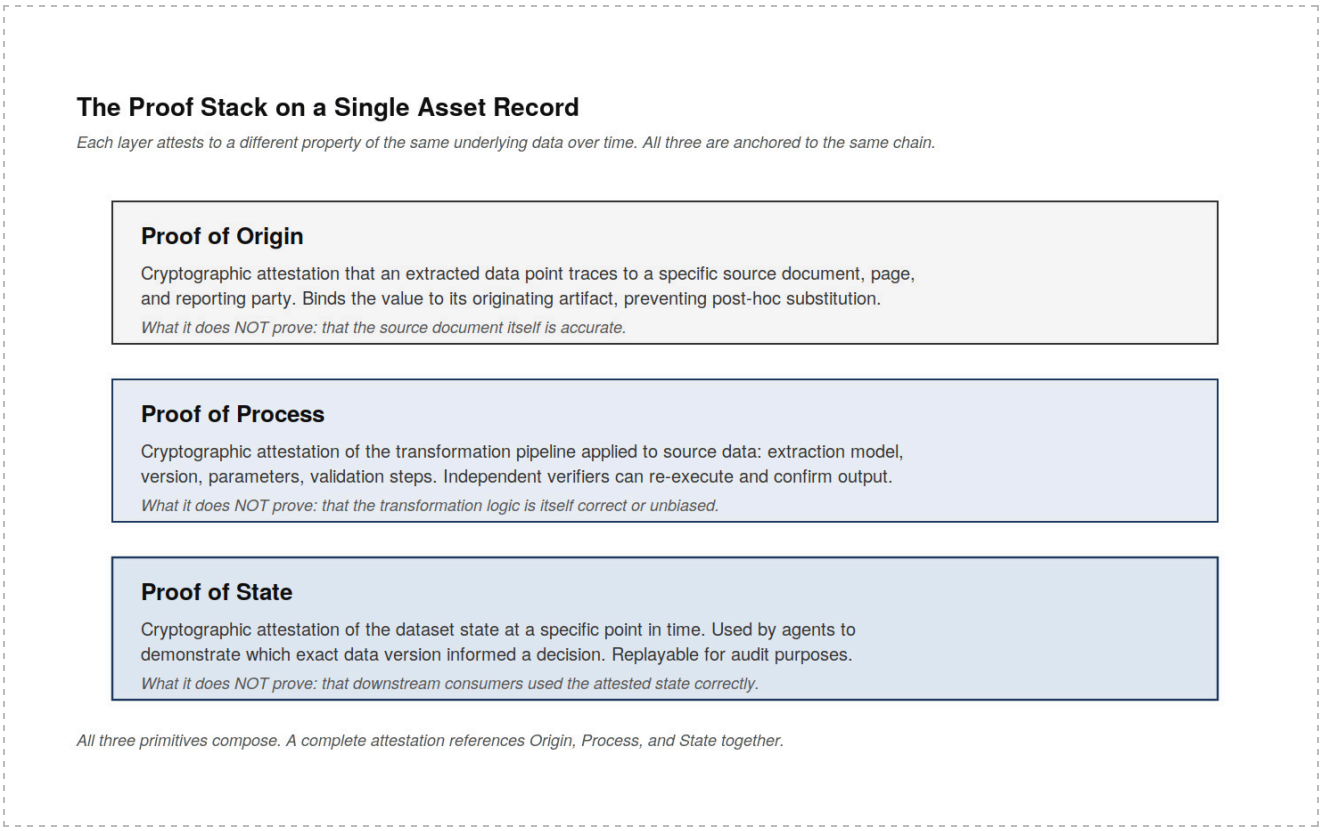


Figure 5. The three primitives of the proof stack. *Each attests to a different property; each has a corresponding limit on what it does not prove.*

Proof of Origin

Proof of Origin is a cryptographic attestation that an extracted data point traces to a specific source document, page, and reporting party. The onchain hash binds the extracted value to the originating artifact, preventing post-hoc tampering or substitution. A counterparty who possesses the source document can independently verify that the extracted value originated there by recomputing the hash and querying the registry[25]. Proof of Origin does not prove that the source document itself is accurate. It establishes the chain of custody, not the truth of the contents.

Proof of Process

Proof of Process is a cryptographic attestation of the transformation pipeline applied to source data. It records the extraction model, its version, its parameters, and the validation steps applied. Independent verifiers who hold the source data and the pipeline specification can re-execute the process and confirm that they obtain the same output[26]. Proof of Process does not prove that the transformation logic is itself correct or unbiased. It allows an auditor or counterparty to reproduce what was done, which is the precondition for evaluating whether what was done was right.

Proof of State

Proof of State is a cryptographic attestation of the dataset state at a specific point in time. It is what an agent uses to demonstrate which exact data version informed a decision. The state is replayable for audit purposes: a regulator who needs to evaluate why an agent acted as it did at 14:32 on a

particular trading day can recover the exact state of the inputs the agent saw at that moment[27,28]. Proof of State does not prove that downstream consumers used the attested state correctly. It establishes what the consumer had available, leaving the question of whether the consumer used it well to whatever audit or review process is relevant.

Composition

The three primitives are designed to compose. A complete attestation of an agent's decision references all three: the Origin attestations of the inputs the agent consulted, the Process attestation of the transformations the agent applied, and the State attestation of the dataset snapshot the agent acted on. The chain treats this as a graph: each record can reference other records by checksum, allowing a downstream verifier to walk the chain of attestations in either direction. The graph is itself anchored. The graph itself can be verified.

4.6 Gas Economics

Gas on NVNM Chain is paid in stablecoins. The chain maintains a whitelist of eligible fee assets, and a transaction specifies which eligible asset it pays in. Conversion and settlement happen at the protocol layer, so a participant handles one asset and sees one dollar-denominated cost.

Why a stable gas asset

Institutional and agentic workloads cannot tolerate volatile gas pricing. A bank's compliance system that anchors thousands of attestations per day cannot have its operating cost vary by an order of magnitude with the price of a volatile token. An agent that needs to decide whether a transaction is economically rational must be able to reason about its cost in terms a treasurer would recognize. Denominating gas in a unit that is stable in fiat terms eliminates a class of failure modes that volatile gas would otherwise introduce.

The settlement currency

Behind the eligible-asset whitelist sits a settlement currency: a root asset the protocol uses internally to route and settle fees. Participants are not expected to hold it, acquire it, or interact with it directly. Its function is to give the fee layer a single unit of account, so that a transaction paid in one eligible stablecoin and a transaction paid in another settle consistently against the same reference. Keeping the whitelist deliberately narrow limits the chain's exposure to a depeg in any single eligible asset.

The settlement currency is a genesis predeploy occupying a fixed canonical address, which means the chain controls it from the first block rather than adopting it afterwards. The specific settlement asset, its issuer, and the bridge route that supplies it are subject to final selection and will be published in a genesis parameter document before mainnet.

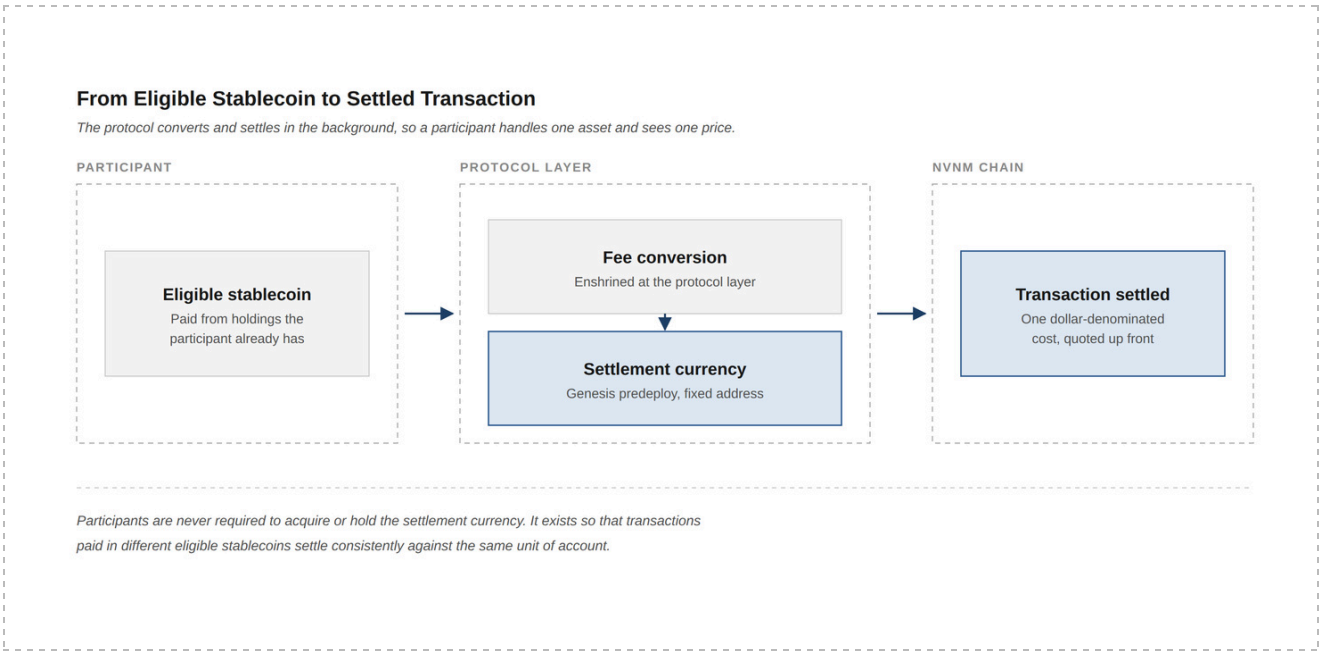


Figure 6. Fee flow. *A participant pays in an eligible stablecoin; the protocol converts and settles through the settlement currency; the participant sees a single dollar-denominated cost.*

Fee market

The chain implements a base fee floor and a base fee mechanism in the EIP-1559 family, denominated in the settlement currency. A minimum gas price floor prevents the base fee from collapsing to zero during periods of low demand, which protects the chain against trivial spam and gives the validator set a predictable revenue floor. The exact parameter values are tunable by Foundation governance during the launch period and will be tunable by community governance subsequently. Concrete base-fee and gas-price parameter values will be published in a separate genesis parameter document.

MEV stance

Maximal extractable value is negligible at the chain's current and projected transaction profile. Anchoring is sequential. The economic value of reordering anchoring transactions is essentially zero. The Foundation will revisit this stance as transaction patterns evolve. If applications emerge that introduce ordering-sensitive economics, the Foundation will publish a formal MEV policy and, if necessary, a mitigation mechanism.

4.7 Bridging and Cross-Chain Operations

NVNM Chain reaches other networks through general-purpose bridge infrastructure rather than through a protocol-native channel to any single chain. Eligible fee assets arrive on NVNM Chain through a bridge route that produces an asset conforming to the chain's fee-asset standard. Additional routes can be added by governance as the ecosystem requires, and each addition is a governance decision of this chain. As stated previously, the bridge provider and stablecoin-denominated fee-asset standard implementation are subject to final selection and will be published before mainnet genesis.

The user-facing experience of acquiring a fee asset is a bridge transfer followed by ordinary use of the chain. The Foundation's product roadmap commits to abstracting acquisition into a single user-facing operation and to publishing supported routes as they are added. The underlying protocol primitives remain unchanged. The improvement is in the orchestration layer that sits above them.

4.8 The MCP Server: Agent Access Surface

The MCP server is the typed, authenticated, authorization-aware interface through which AI agents and developer tools interact with NVNM Chain. It is implemented as a server in the Model Context Protocol family[29,30,31]. MCP is an open protocol developed for connecting AI assistants to external systems. The Foundation's decision to implement the agent access surface as an MCP server reflects a judgment about where agent infrastructure is converging[32].

What it does

The server exposes a typed tool surface. Each tool corresponds to a specific operation: addRecord, addRegistry, updateRecordStatus, queryRecord, queryRegistry, grantRole, revokeRole, and a small number of read-only utility tools that wrap common precompile queries. Each tool publishes a schema for its inputs and outputs. An agent that calls a tool through the server has its inputs validated against the schema before any chain interaction occurs. Outputs are returned in normalized JSON suitable for agent reasoning.

Why not a JSON-RPC proxy

A naive design would expose JSON-RPC directly to agents and let them call the chain themselves. This would shift validation responsibility to the agent. It would also expose the agent to the full surface area of the EVM, which is far larger than the surface area required to anchor records. The MCP server is deliberately narrower than JSON-RPC. It exposes only the operations relevant to the chain's mandate. Agents cannot accidentally interact with the chain in ways that fall outside the authorized scope.

Prepare-sign-submit

The most consequential security property of the MCP server is that it does not hold private keys for any state-changing operation. For any tool call that produces a chain-modifying transaction, the server prepares an unsigned transaction and returns it to the caller. The caller signs the transaction externally, using whatever key management the operator already trusts: a hardware security module, a Vault instance, a browser-based wallet, or a custodian-managed signing service[33]. The signed transaction is then broadcast either by the caller directly or by the server on the caller's behalf, but in either case the server has not held the key.

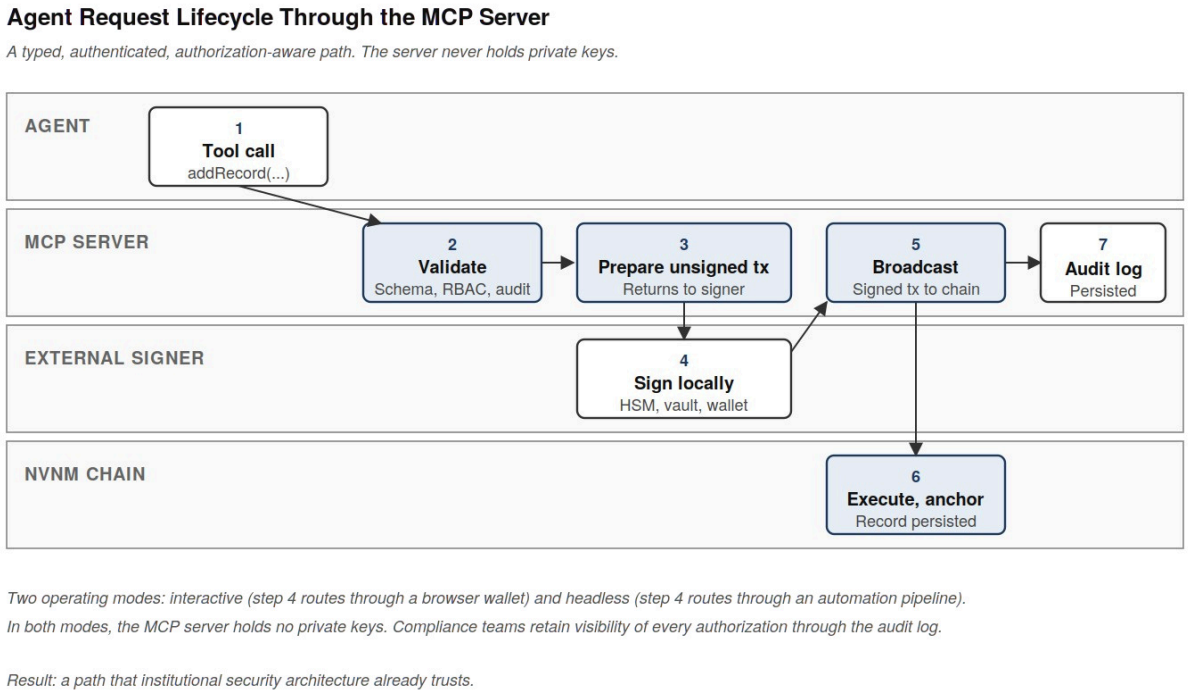


Figure 7. Agent request lifecycle. *The MCP server validates, prepares, broadcasts, and audit-logs. The signing step is performed externally to whatever security perimeter the operator already trusts.*

Operating modes

The server supports two operating modes that share the same backend. In interactive mode, the signer is a human operator using a browser wallet. The unsigned transaction is presented for human review before signing. This mode is appropriate for low-frequency, high-value operations such as registry creation or role assignments. In headless mode, the signer is an automated pipeline backed by a hardware security module or equivalent. The unsigned transaction is delivered to the pipeline through the operator's existing infrastructure. This mode is appropriate for high-frequency anchoring such as continuous compliance attestation. Both modes preserve the no-server-keys property.

Why this matters for institutional adoption

An institutional integrator evaluating NVNM Chain has to defend the integration to a compliance committee that did not write it. Three questions have to be answerable: what data could the agent touch, what actions could it take, and who authorized which action. The MCP server answers all three. The typed tool surface limits what data the agent can touch. The RBAC system limits what actions it can take. The audit log records who authorized which action, with what arguments, at what time. The compliance committee can read the answer in language they already use[34,35,36].

4.9 The Data Layer

NVNM Chain is platform-agnostic. Any party that produces valid attestations can anchor records to a registry it controls. The chain does not privilege a particular data feed at the protocol level. At launch, the canonical first integrated data layer is Inveniam.io, the decentralized data management platform for private market assets.

What Inveniam.io does

Inveniam.io is a federated data extraction and attribution platform. It connects to existing enterprise storage environments such as SharePoint, Amazon S3, secure file servers, and administrator portals. It does not require source data to be migrated. Source data remains in the operator's environment. Inveniam.io reads it in place, structures it through AI-driven extraction pipelines, attributes every extracted data point to a specific source document and page, and produces cryptographic hashes of the source artifacts (the "Golden Copies") and of the structured outputs.

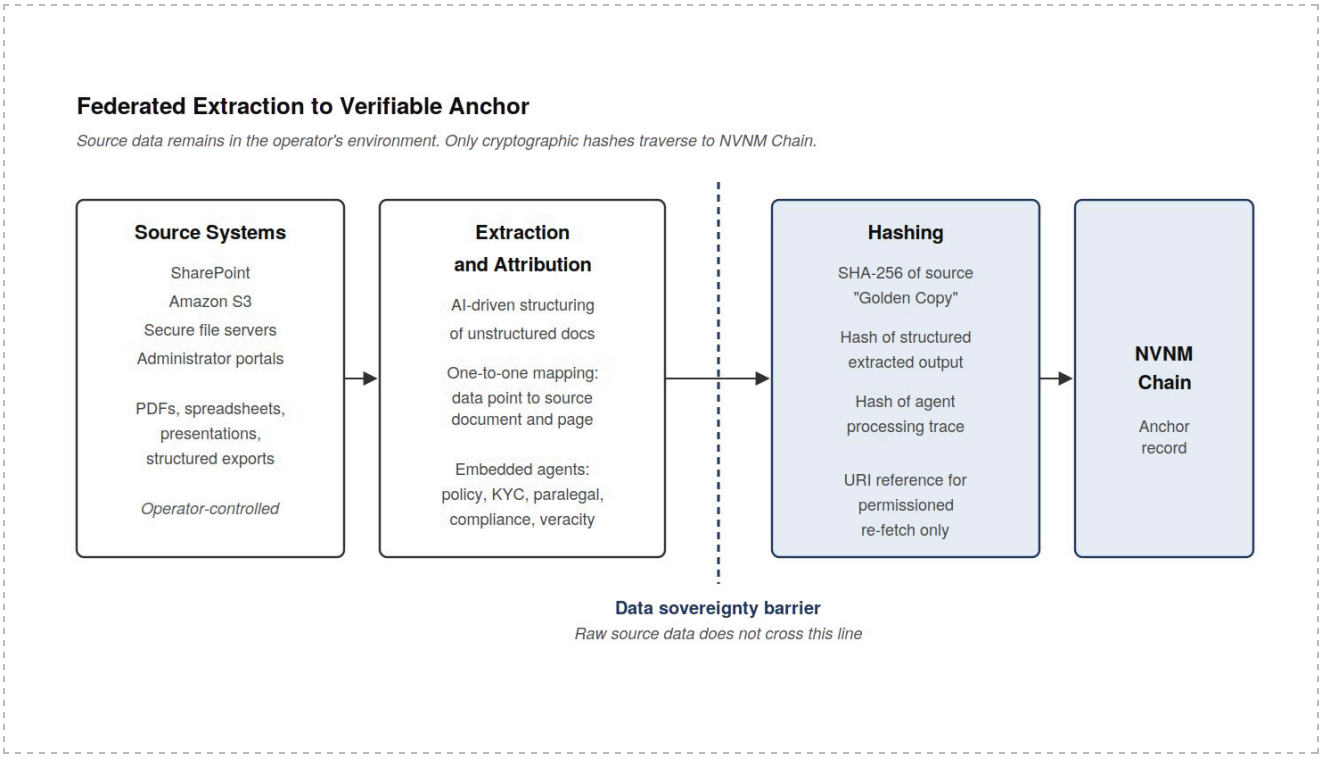


Figure 8. Federated extraction to verifiable anchor. *The data sovereignty barrier is preserved: source data does not cross the line.*

Asset class coverage

The platform is integrated across the major private market asset classes. Private equity coverage includes partnership agreements, capital account statements, and valuation memoranda. Private credit and Collateralized Loan Obligation (CLO) coverage includes loan tapes, collateral schedules, trustee reports, waterfall statements, and compliance certificates. Real estate coverage includes rent rolls, operating statements, appraisal reports, and debt service schedules. Each asset class is supported by document types specific to its conventions. Each extracted data point retains a one-to-one attribution chain back to its originating document, page, and reporting counterparty.

Embedded agents

Inveniam.io exposes structured data to a set of embedded agents that act on it for purposes useful to operators. Examples include a Policy Alignment Agent that evaluates document content against internal governance policies and external regulatory requirements; a KYC and AML Agent that streamlines onboarding and monitoring; a Paralegal Agent that extracts and structures key legal terms from fund documents and loan agreements; a Compliance Agent that monitors regulatory obligations across multiple jurisdictions; a Market Intelligence Agent that contextualizes asset-level data against market conditions; and an Asset Veracity Agent that produces a confidence signal reflecting the completeness and consistency of inputs to reported valuations. These are illustrative examples; the agent set will continue to expand. They demonstrate the substrate end-to-end in financial services. Equivalent domain-specific data layers can integrate against the same chain primitives in other verticals.

Open data layer

Inveniam.io is the canonical first integrated data layer. The chain is not dependent on it. Any data provider that produces valid Proof of Origin attestations to a registry it controls can anchor records on equal terms. The chain's long-term role is as an open verification layer. Inveniam.io's long-term role is as the most fully integrated of many possible data layers[37].

5. Use Case Patterns

The patterns below describe what the chain does for the operator and the counterparties. Specific real-world instantiations appear as examples within each pattern.

The use cases that follow are weighted toward financial markets because that is where the deepest existing data layer (Inveniam.io) and the most developed institutional integration paths sit at launch. The same pattern applies to other multi-party agentic workflows where parties need to verify each other's work without an intermediary adjudicating truth. Multi-party clinical trials with sponsors, contract research organizations, sites, and regulators acting on data they did not produce; supply chain attestation across manufacturers, customs, distributors, and regulators who do not share systems and have non-aligned incentives; and regulated AI compute reporting between operators, exporters, and government inspectors are direct instantiations. The chain does not change shape across these domains. Each domain brings its own data layer.

5.1 Regulatory Compliance Anchoring

Any regulated activity that requires continuous, tamper-evident audit trails accessible to inspectors without exposing operational details fits this category. Export controls on advanced AI infrastructure are a representative example. Operators of advanced AI compute face increasingly demanding reporting, monitoring, recordkeeping, and audit obligations, including periodic chip accounting, multi-year record retention, on-site inspection rights, and audit cooperation requirements[38,39]. Cryptographic anchoring is a plausible technical architecture for satisfying those obligations. Current public rules do not generally mandate continuous cryptographic verification, and the regulatory landscape for AI export controls is itself in flux[40].

The compliance posture this implies is unlike anything that legacy enterprise logging was designed for. It must be always-on, tamper-proof, independently auditable, and capable of supplying evidence

on demand without exposing the underlying operational data. NVNM Chain provides the substrate for that posture. Audit metadata is captured by the operator's instrumentation, batched into JSON bundles, summarized as Merkle roots, and anchored on chain through a single transaction per batch. A batch can summarize thousands of underlying events. Inspectors who request evidence for a specific time window receive the corresponding records together with proofs that link those records to the onchain anchor. Verification proceeds against the anchor. The operational data is exposed only to the extent the inspection requires[41].

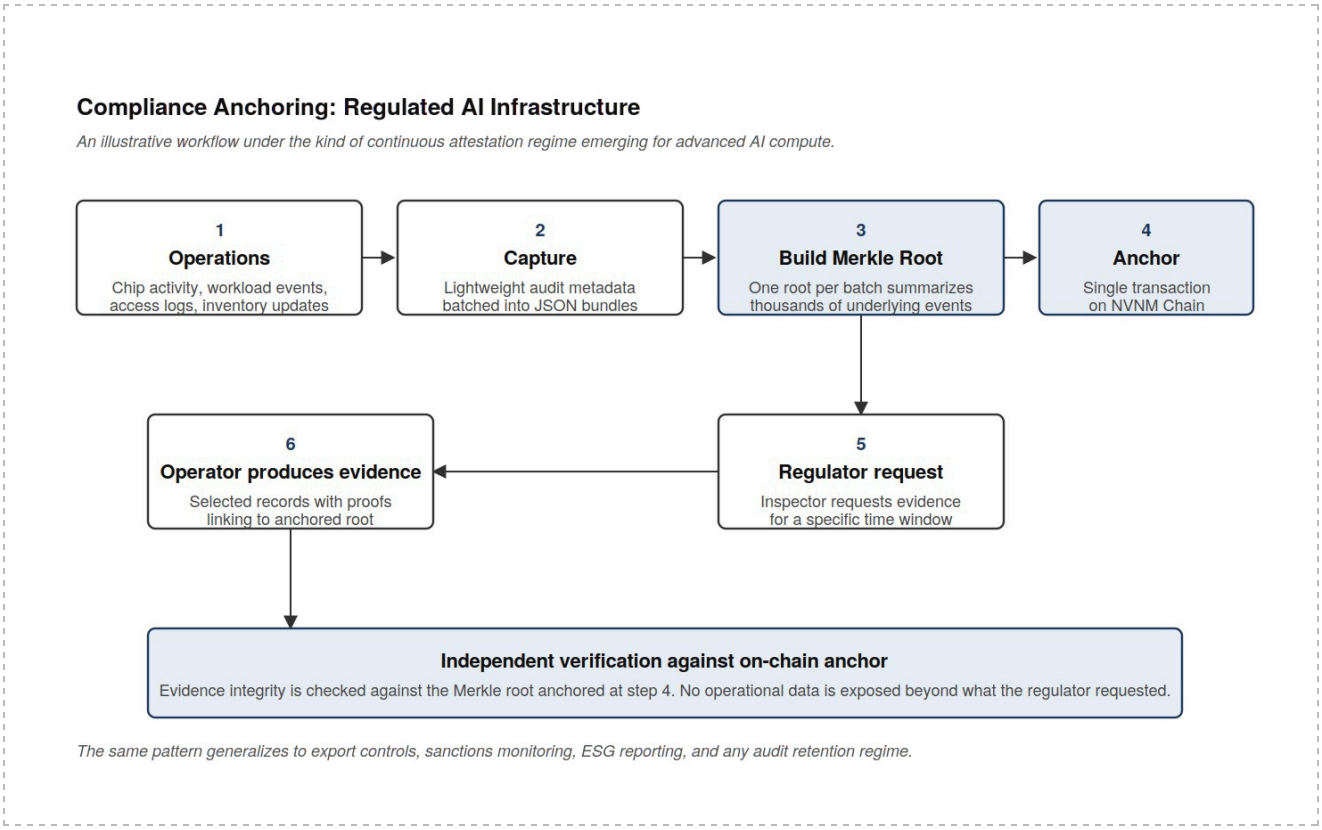


Figure 9. Compliance anchoring workflow. *The same pattern generalizes to export controls, sanctions monitoring, ESG reporting, and any audit retention regime that requires tamper-evident records accessible to inspectors.*

The pattern generalizes. Export controls, sanctions screening, ESG reporting, professional services audit retention, banking compliance, and any other regime that demands tamper-evident records accessible to a regulator on request can use the same primitives. The chain does not dictate what gets anchored. It guarantees that what is anchored cannot be quietly altered, and that downstream consumers can verify what they are looking at without depending on the operator who produced it.

5.2 Tokenized Real-World Asset Verification

Continuous attribution of asset valuations and underlying data to source documents lets DeFi protocols and institutional consumers safely consume tokenized real-world assets. Tokenization without continuous, attributable data is a wrapper around an illiquid claim[42,43]. The wrapper is convenient. It does not address the underlying trust gap. NVNM Chain fills that gap by anchoring proof of the data inputs that drive valuations and the processing that produces them.

Several specific applications follow from the pattern. Lending protocols collateralized by tokenized private credit positions need continuous proof that the inputs to collateral monitoring are fresh and attributable. Secondary markets for tokenized fund interests need an independent indicative mark, anchored to verifiable inputs, to clear transactions at fair prices. Perpetual swaps platforms referencing private market underlyings need a price signal whose provenance can be traced through the chain to the source data. Onchain fund structures with NAV-based subscription and redemption need automated valuation that does not depend on a manually-certified administrator snapshot. Vaults that expand into RWA collateral need dynamic monitoring whose evidence is independently verifiable.

The market itself is moving quickly. Daily trading volume on RWA perpetuals platforms has reached material scale[44]. The total value of tokenized real-world assets exceeds three hundred billion dollars when stablecoins are included; the represented asset value across all tokenized RWA categories sits in the mid-hundreds of billions, with the distributed (non-stablecoin) tokenized RWA value an order of magnitude smaller[45]. The bottleneck on further growth is not market demand. The bottleneck is verifiable data infrastructure that institutional risk officers and regulators can sign off on.

5.3 Agentic Workflows in Regulated Contexts

AI agents operating in regulated or institutionally sensitive contexts must produce decisions that are defensible after the fact. Several sub-patterns appear in the workflows the chain serves at launch[46,47,48].

Compliance monitoring agents

A compliance agent continuously evaluates covenant compliance, regulatory thresholds, concentration limits, and other rule-based conditions. Each evaluation can be anchored: an attestation of the rule the agent applied, an attestation of the inputs it consulted, and an attestation of the conclusion it reached. The audit trail is machine-readable and complete. A compliance committee that needs to review the agent's record over the prior quarter does so by querying the chain.

Secondary trading agents

A secondary trading agent executing private credit or fund interest secondaries needs full proof of pricing inputs at decision time. The agent anchors a Proof of State of the dataset it consulted, a Proof of Process of the pricing model it applied, and a Proof of Origin of the inputs that fed the model. A counterparty disputing the trade can replay the exact decision against the exact inputs. A regulator examining best execution can do the same.

Valuation agents

A valuation agent producing indicative NAVs for tokenized fund interests anchors the methodology, the inputs, and the result. Downstream consumers verify the mark by checking that the inputs match what they expect, that the methodology is the one the fund's offering documents prescribe, and that the result was produced from those inputs through that methodology. The valuation becomes auditable in a way that quarterly administrator-certified marks have never been.

Cross-jurisdictional regulatory agents

A cross-jurisdictional agent monitoring obligations across multiple authorities (such as the SEC, FCA, BaFin, VARA, and ADGM) anchors evidence of monitoring activity per jurisdiction. A regulator inspecting the operator can confirm that monitoring was performed, when it was performed, and what conclusions were reached. The chain does not displace the underlying compliance obligations. It provides verifiable evidence that the obligations were honored.

Multi-party clinical trial data

Multi-party clinical trials present a structurally similar problem. Sponsors, investigators, contract research organizations, and regulators all need to act on data that originates with subjects whose privacy is non-negotiable. The pattern of cryptographic anchoring with permissioned access maps to this domain directly[49,50,51]. Recent peer-reviewed research has explored the combination of secure multi-party computation with blockchain-anchored notarization for the production of trial cohorts[52,53]. NVNM Chain is the substrate on which such architectures can be deployed without the overhead of standing up a chain per trial.

5.4 Agent Identity and Accountability

Each AI agent operating against NVNM Chain has a unique persistent onchain wallet, never shared with another agent. The agent's actions are forever tied to that wallet. This convention is what NVNM Chain calls Know Your Agent (KYA).

The convention is small but the consequences are direct. A counterparty consuming an agent's outputs can query the agent's wallet on chain and verify the agent's full history of actions independently of the operator. An operator can pre-evaluate a third-party agent before integrating with it, by inspecting that agent's prior onchain behaviour. Revocation reduces to revocation of authorisations on a single wallet rather than a model-wide or vendor-wide policy change. Audit across multiple counterparties reduces to a query against a known address[54,55]. The wallet, not the underlying model or vendor, is the unit of identity for accountability purposes.

KYA at launch is the wallet convention plus optional registration of the operating organisation and declared scope of authorisation. It is not a token-staking gate, and it does not require participation in any incentive programme. Future enhancements will layer reputation primitives on top of the wallet convention, but the wallet convention itself is sufficient to give counterparties what they need: a stable, persistent, queryable identity for every agent that produces consequential output on the chain.

5.5 Human Oversight Patterns

Structured human review can be integrated into agent workflows where the stakes of an agent's decision are too high for full automation to be acceptable[56]. The chain supports several variants of this pattern, none of them mandatory. Operators can integrate human review into any workflow where they judge it appropriate. The chain provides primitives that make the review record auditable.

A simple variant is human-in-the-loop approval: an agent prepares an action, a designated human reviewer approves or rejects it, and the chain anchors both the agent's preparation and the reviewer's decision. A more elaborate variant uses structured reviewer accountability, where review quality is

measurable and consequential. Operators can parameterize this in different ways depending on the stakes of the workflow. The chain anchors whatever parameterization the operator adopts, so that the auditable record reflects the rules the operator was actually running. Appendix E sets out one illustrative parameterization that operators are free to use, modify, or substitute.

6. Regulatory Framework

> *When government steps aside, it is not as if nothing takes its place.*
>
> — Lawrence Lessig, Code and Other Laws of Cyberspace

NVNM Chain is built for institutional and regulated use cases. Compliance is a feature, not a constraint. Infrastructure designed with regulated participation in mind serves more of the market than infrastructure designed to evade it.

Foundation governance

The NVNM Chain Foundation will hold early administrative authority over the chain. Its initial responsibilities include parameter administration during the launch period, validator-set administration, and protocol upgrade management under a documented process. The Foundation's mandate, governance structure, and relationship to onchain operating entities and Inveniam will be specified in a forthcoming governance document, to be published in advance of any community-governance transition. The chain itself does not depend on Foundation discretion for record integrity. Records are persisted on chain under the rules in Section 4 and remain verifiable independent of any Foundation action.

Anchoring and data exposure

The chain anchors only cryptographic checksums and reference metadata. The underlying data does not traverse to the chain. This property is regulatorily relevant because it satisfies the institutional and statutory constraints around proprietary data, personally identifiable information, and trade secrets. An institution that cannot place customer-level data on a public ledger can still place a verifiable proof of compliance attestations on a public ledger. The proof is independently auditable. The data remains where it lawfully belongs.

Multi-jurisdictional positioning

European Union

The Markets in Crypto-Assets Regulation provides the operative framework for crypto-asset issuers and crypto-asset service providers across the European Union[58]. The chain itself is infrastructure rather than an issued asset, and is not directly subject to MiCA. Service providers operating on the chain in the European Union will be subject to MiCA where applicable. The Foundation does not assert that the chain is MiCA-aligned in any sense beyond infrastructure that serves regulated entities operating within MiCA.

United Arab Emirates

The Virtual Assets Regulatory Authority in Dubai operates a comprehensive rulebook framework for virtual asset activities[59]. ADGM in Abu Dhabi operates a parallel framework for digital assets. Distribution and service activities by service providers operating in the UAE are subject to the appropriate authority's rulebook.

United States

The Foundation is monitoring guidance from the US Securities and Exchange Commission, the Commodity Futures Trading Commission, the Bureau of Industry and Security, and the Office of Foreign Assets Control as it applies to the chain's use cases. The Foundation will publish a US-specific accessibility framework before engaging in any activity that requires it. Until that framework is published, US persons should consult with counsel before participating in chain-based services. The chain itself is global infrastructure. Specific service providers operating on the chain may impose geographic restrictions appropriate to their jurisdictional posture.

7. Comparative Landscape

NVNM Chain is part of a larger ecosystem of infrastructure that has been built by others over many years. Each of the systems described below has solved a real problem and continues to do so. NVNM Chain solves a different problem.

Public oracle networks

Networks such as Chainlink and Pyth deliver high-frequency public data feeds, including price feeds for crypto and traditional assets, to onchain consumers. They have built economic and operational discipline around the problem of getting public data onto chains in a timely and reliable way. The problem they solve and the problem NVNM Chain solves overlap at the edges and diverge at the center. Public oracles publish public data. NVNM Chain anchors proofs of private data. A future protocol might compose the two: a public oracle network could anchor proofs of its feed integrity to NVNM Chain. The Foundation considers such compositions complementary rather than competitive.

General-purpose RWA chains

Several chains have been built specifically to host tokenized real-world assets. Their focus is on issuance, transfer, and the lifecycle of the tokens themselves. The chain itself is a vehicle for representing claims. NVNM Chain's focus is upstream of that. Tokens that represent claims to private market assets need data to support their valuation and their compliance. Where the token lives is a separate question from where the supporting data is verified. The two layers can coexist.

Decentralized data networks

Networks such as The Graph and Ocean provide indexing, marketplace, or computation services for data that is itself onchain or made available through specific interfaces. Their model is data-as-product. NVNM Chain's model is proof-as-product. The chain does not take possession of data. It records cryptographic claims about data that remains where its lawful custodian holds it.

Document anchoring services and standards

A range of services and open standards address parts of the document anchoring problem. W3C Verifiable Credentials and Decentralized Identifiers provide an interoperable model for tamper-evident credentials and identifiers. HL7 FHIR provides a structured exchange standard for healthcare data. The Coalition for Content Provenance and Authenticity provides production-grade content provenance for digital media[60,61,62,63]. NVNM Chain is positioned to interoperate with each of these. A Verifiable Credential can carry a reference to an NVNM Chain anchor as its proof of issuance. A C2PA manifest can be anchored on NVNM Chain for verifiability beyond the manifest signer's key infrastructure. The chain extends, rather than displaces, the work that has produced these standards.

Multi-agent orchestration frameworks

Frameworks for orchestrating multiple AI agents (such as LangGraph, CrewAI, and AutoGen) handle agent-to-agent task handoff, state passing, and tool invocation within a coordinated runtime. They are essential infrastructure for building agentic systems within a single organization, where the agents share an operating environment and the handoff records can be trusted because the operator controls them. They are not designed for the case where the agents belong to different operators with non-aligned incentives, and where the handoff record itself is what one party would need to challenge. Orchestration frameworks make agent collaboration tractable inside a trust boundary. NVNM Chain provides the substrate that lets agent activity be verified across trust boundaries. The two layers compose: an orchestration framework can use NVNM Chain to anchor the records that need to outlive the framework's own runtime and that need to be verifiable by parties outside it.

Generic blockchain timestamping

Several services use general-purpose chains to publish document hashes for time-stamping purposes. Such services demonstrate the value of public ledger anchoring as a primitive. They do not provide the role-based access semantics, the agent integration patterns, or the institutional integration paths that the use cases described in Section 5 require. NVNM Chain is purpose-built for those requirements. Generic timestamping services remain useful for simpler use cases.

8. Roadmap and Vision

Specific calendar dates are not committed here. Phases progress as the underlying ecosystem matures. The Foundation will publish quarterly updates as the ecosystem evolves. The phases below describe the rollout of the chain's capabilities and use cases; expansion of the validator set follows its own revenue-gated schedule, set out in the NVNM Tokenomics Whitepaper.

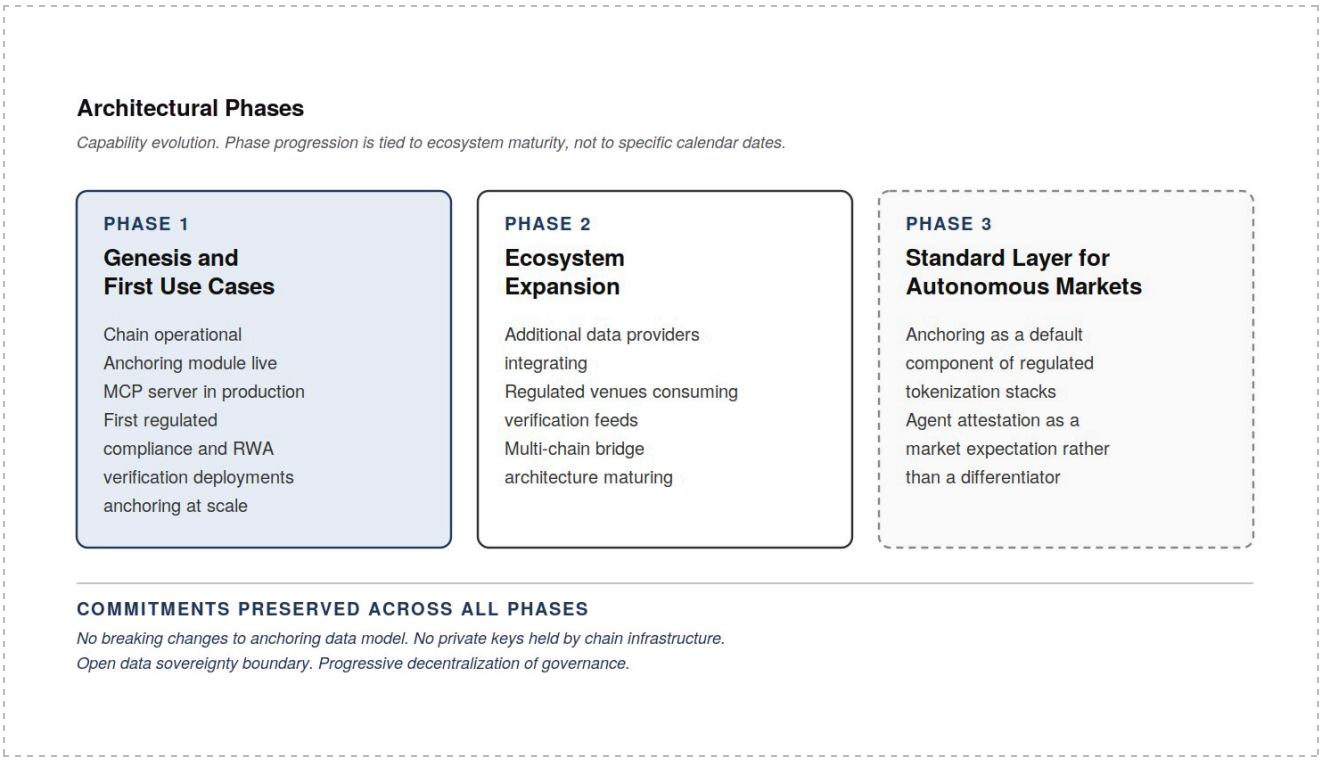


Figure 10. Architectural phases. *Phase progression is tied to ecosystem maturity rather than to specific calendar dates. Architectural commitments are preserved across all phases.*

Phase 1: Genesis and first use cases

In Phase 1, the chain is operational. The anchoring module is live. The MCP server is in production. The first regulated compliance and RWA verification deployments are anchoring at scale. The Foundation operates the chain under administrative authority, with parameter changes and upgrades managed through a documented process. NVNM Chain's own validator set secures the chain under proof of authority. Distribution of the chain's capabilities to consumers is through direct integration with the operators who have committed to using the chain at launch.

Phase 2: Ecosystem expansion

In Phase 2, additional data providers integrate as data feeds. Additional regulated venues consume verification feeds from the chain. The cross-chain bridge architecture matures, allowing assets and proofs to move between NVNM Chain and other ecosystems with which integration is useful. Governance progressively decentralizes as the ecosystem develops; the form and pace of decentralization will be specified in a separate governance document. The Foundation's administrative authority narrows over the course of this phase.

Phase 3: Standard layer for agentic markets

In Phase 3, anchoring becomes a default component of regulated tokenization stacks. Agent attestation becomes a market expectation rather than a competitive differentiator. NVNM Chain serves as infrastructure for the agentic capital markets layer that the financial industry has been moving toward for several years.

Invariants across phases

- No breaking changes to the anchoring data model. Records and registries created at Phase 1 remain verifiable at Phase 3 and beyond.
- No private keys held by chain infrastructure components. The MCP server's prepare-sign-submit pattern is preserved.
- Foundation Treasury and operational addresses are publicly identified. The operating posture is transparent by default.
- Open data sovereignty boundary. Source data does not traverse the chain in any phase, under any circumstance.
- Progressive decentralization of governance. Foundation administrative authority narrows over time.

Upstream dependencies

The chain's execution and consensus layers are built on externally maintained open-source libraries: Reth for execution[15] and the Commonware consensus library for Simplex.[12] Both are permissively licensed and actively developed, and improvements made upstream reach NVNM Chain through scheduled dependency upgrades rather than through protocol redesign. Commonware reported a pipelined Simplex implementation sustaining 200 blocks per second with 300 millisecond finality in a 50-validator global deployment.[16] The maintenance obligation this creates runs in both directions and is named among the failure modes in Section 4.1: upstream defects become the chain's burden, and the Foundation tracks upstream releases as a standing operational responsibility.

Governance and value-accrual token

A governance and value-accrual token, $NVNM, launches separately from the chain. Its design and economics, including supply schedule, distribution, governance structure, staking mechanics and value accrual, are described in the NVNM Tokenomics Whitepaper. The chain's utility is independent of any token issuance and operates at launch with stablecoin-denominated economics under Foundation administrative authority.

Areas of active research

Several areas are under active research and will inform later phases. Privacy-preserving proofs (such as zk-SNARK constructions for cases where even the existence of an anchor is sensitive) are being evaluated against the chain's use cases. Agent reputation primitives, beyond the identity registration provided by KYA, are being designed in cooperation with institutional partners. Adversarial robustness against model-theoretic and model-extraction attacks on the embedded agents is being studied in line with NIST guidance.[64] Native cross-chain anchoring for assets that exist on multiple ledgers is on the research agenda.

9. Conclusion

The financial system has always run on testimony. A bank lends because a borrower's financials show what they show. An auditor signs because a sample of records reveals what it reveals. A regulator approves because a filing represents what it represents. The same is true of every multi-party domain in which counterparties have to act on records they did not produce: clinical trials, supply chains, regulated AI compute, inter-organizational reporting. The system works when

consequences for misrepresentation are credible and when the participants accumulate reasons to trust each other over time. It does not extend cleanly to AI agents acting at machine speed across organizational boundaries, where there is no time to accumulate reputational capital and no neutral party available to adjudicate when reputations diverge. NVNM Chain provides the substrate on which agentic participation in regulated multi-party workflows can be built. Its primitives are old. Cryptographic timestamping is more than thirty years old. Append-only Merkle logs are more than fifteen years old. Byzantine fault-tolerant consensus is more than forty years old. The contribution of the chain is to compose these primitives, with the institutional governance, the agent access patterns, and the operational discipline that the use cases of the next decade require. Nick Szabo observed in 2001 that "the invocation or assumption in a security protocol design of a trusted third party constitutes the introduction of a security hole into that design."[65] The remark is more than two decades old and remains accurate. NVNM Chain does not claim to eliminate trust. It claims to relocate trust away from third parties whose continued operation is uncertain and toward cryptographic constructions whose security properties are well understood. That relocation is what allows long-dated regulated records to be verifiable for longer than any specific vendor.

This document is itself an instance of the proof primitives it describes. After publication it is anchored to NVNM Chain as a record in the nvnm-whitepaper registry, of which it is the first entry. The anchoring record is published at nvnmchain.io. A reader who holds a copy of this document can verify integrity by recomputing the document's SHA-256 checksum and confirming it matches the anchored record. The chain begins with a record of its own design intent. Successive versions of this document, recording successive refinements of the design, will join the same registry. Each will reference the prior. The history will be auditable for as long as the chain runs, and the chain is built to run.

Appendix A: Glossary

Terms defined here are used consistently throughout the document. Where a term has both a colloquial meaning and a technical meaning in the chain's context, the technical meaning governs.

Anchoring. The act of recording a cryptographic checksum of an off-chain artifact onto NVNM Chain, together with metadata sufficient to identify the artifact and its lifecycle state. The artifact itself is not anchored; only its fingerprint and reference metadata are.

Checksum. A cryptographic hash of a piece of data, produced by an algorithm such as SHA-256. A checksum is small (32 bytes for SHA-256), deterministic, and infeasible to forge. Two pieces of data that produce the same checksum are computationally indistinguishable.

Commonware Simplex. The Byzantine fault-tolerant consensus protocol used by NVNM Chain. Simplex provides deterministic finality in approximately 500 milliseconds: once a block is finalized by the validator set it is final, with no probabilistic settlement window. Safety holds in the absence of more than one third Byzantine validators.

EVM precompile. A function exposed to the Ethereum Virtual Machine at a fixed canonical address, implemented in native chain code rather than in EVM bytecode. Precompiles execute deterministically and at gas costs predictable across protocol upgrades.

Editor. An anchoring module role that grants the holder the ability to write records, archive records, and update mutable record metadata, but not to grant or revoke roles.

Federated extraction. A pattern of data processing in which the processor connects to existing storage systems and reads data in place, without requiring data migration to a new system. Source data remains under the lawful custodian's control.

Foundation. The NVNM Chain Foundation, the legal entity that holds early administrative authority over the chain. Being incorporated under the Abu Dhabi Global Market Digital Asset Regulatory Framework.

KYA (Know Your Agent). The convention by which each AI agent operating against NVNM Chain has a unique persistent onchain wallet, never shared with another agent. The agent's actions are forever tied to that wallet, and counterparties verify what an agent has done by querying its wallet on chain. KYA at launch is the wallet convention plus optional registration of the operating organization and declared scope of authorization.

MCP (Model Context Protocol). An open protocol for connecting AI assistants and developer tools to external systems through typed, authenticated tool surfaces. NVNM Chain operates an MCP server at mcp.nvnmchain.io as its agent access surface.

Proof of authority. The consensus governance model under which NVNM Chain launches. The validator set is permissioned, each operator is an identifiable party, and positions are acquired rather than delegated.

NAV (Net Asset Value). The accounting value of a fund or portfolio at a given point in time. In private markets, NAV is reported episodically on cycles measured in months or quarters; on chain, NAV-based mechanics require continuously refreshed and verifiable inputs.

Provenance. The verifiable chain of custody connecting a piece of data to its source, its processing history, and its state at a given time. Composed of Proof of Origin, Proof of Process, and Proof of State. Proof of Origin. A cryptographic attestation that an extracted data point traces to a specific source document, page, and reporting party.

Proof of Process. A cryptographic attestation of the transformation pipeline applied to source data, including the extraction model, version, parameters, and validation steps.

Proof of State. A cryptographic attestation of the dataset state at a specific point in time, used to demonstrate which exact data version informed a decision.

Record. An attestation entry within a registry. Each record carries a checksum, a checksum algorithm identifier, a URI reference, structured metadata, a status, and a version index.

Registry. A permissioned namespace on NVNM Chain that contains records. Each registry has its own administrators, editors, and access controls. A registry corresponds conceptually to an institutional governance domain.

RBAC (Role-Based Access Control). A pattern of access management in which permissions are granted to roles, and roles are granted to identities. NVNM Chain implements RBAC at three scopes: record, registry, and global.

SHA-256. The specific instance of the SHA-2 cryptographic hash family that produces a 256-bit output. NVNM Chain treats SHA-256 as the default checksum algorithm. Other algorithms can be specified per record through the checksum algorithm identifier field.

Reth. The modular Ethereum execution client used as NVNM Chain's execution library. It provides current EVM compatibility, so standard Ethereum tooling works against the chain without modification.

Settlement currency. The root asset the protocol uses internally to route and settle transaction fees. Participants pay in eligible stablecoins; the settlement currency gives the fee layer a single unit of account. It is a genesis predeploy at a fixed canonical address.

Appendix B: Anchoring Module Function Reference

The anchoring module is exposed through a single interface: an EVM precompile at the fixed canonical address 0x0000000000000000000000000000000000000A00. The signatures below describe the operations conceptually. Concrete Solidity ABI signatures and parameter types are published in the engineering specification and are versioned independently of this paper.

B.1 Data structures

Record

Represents a single entry in a registry, including content hash, metadata, and versioning status. Fields: registry, uri, checksum, checksumAlgo, metadata, timestamp, status, recordId, index, isLatest. The checksum and checksumAlgo are immutable once written. The timestamp, recordId, index, and isLatest are set by the chain.

Registry

Defines a logical container for a category of records. Fields: id, name, description, creator, createdAt.

PageRequest

Standard pagination input for handling large result sets in queries. Fields: key, offset, limit, countTotal, reverse.

PageResponse

Pagination output indicating the cursor for the next set of results and the total count. Fields: nextKey, total.

B.2 State-changing functions

addRegistry

Creates a new registry. The caller becomes the registry administrator at creation time.

addRegistry(name, description, metadata) -> uint64 registryId

Inputs: name (string), description (string), metadata (string). Returns the registry's unique identifier. addRecord

Adds a new record to an existing registry. The caller must hold an editor or admin role at the registry or global scope. The record is passed as a struct; caller-supplied fields are registry, uri, checksum, checksumAlgo, metadata, and status. Chain-set fields are timestamp, recordId, index, and isLatest. addRecord(Record record)

updateRecordStatus

Updates the status of a specific record version. The checksum and checksum algorithm cannot be modified by this call. Common transitions include archiving a record that has been superseded. updateRecordStatus(registryId, recordId, index, status)

Inputs: registryId, recordId, index, status (string).

B.3 Read functions

Read access is public. Empty filter values are ignored, allowing a single function to serve point lookups and ranged queries.

records

Searches records by registry, checksum, recordId, or index, with pagination support.

records(registry, checksum, recordId, index, pagination) -> (Record[],PageResponse)

registries

Lists or finds registries by identifier or name, with pagination support.

registries(registryId, name, pagination) -> (Registry[], PageResponse)

B.4 Permission functions

Two roles are grantable: admin and editor. Public read access does not require a grant. Scope is determined by the parameters: an empty checksum and a registry identifier produce a registry-scope grant; a specific checksum produces a record-scope grant. Global-scope roles are administered through governance and module parameters rather than these functions.

grantRole

grantRole(registryId, checksum, account, role)

revokeRole

revokeRole(registryId, checksum, account, role)

B.5 Record status

Status is a string field on each record. Operators may define their own status conventions. Common values include Active for a current record, Archived for a record that has been intentionally retired, Superseded for a record replaced by a later version sharing the same recordId, and Voided for a record withdrawn due to a write error. The chain does not enforce status semantics; downstream verifiers interpret the value according to the operator's convention.

Appendix C: MCP Server Tool Reference

The MCP server at mcp.nvnmchain.io exposes a typed tool surface. Each tool publishes a schema for its inputs and outputs. The surface is expected to grow and evolve over time as integrations and use cases expand. The list below is illustrative rather than exhaustive, summarizing categories of tools available at launch. Full schemas, parameter types, return types, authentication requirements, and the complete current tool list are published in the server documentation and versioned independently.

Generic EVM read tools

- eth_call. Executes an EVM call against a target contract and returns the raw output. Subject to RBAC restrictions configured at the server.
- eth_getBalance. Returns the fee-asset balance of an address.
- eth_getCode. Returns the deployed bytecode at an address.
- eth_getTransactionByHash. Returns the details of a transaction by its hash.
- eth_getTransactionReceipt. Returns the receipt of a confirmed transaction. Anchoring read tools
- readRecord. Returns a record by registry and record identifier.
- readRecordByChecksum. Returns a record by registry and checksum.
- readRegistry. Returns a registry by identifier or by name.
- listRecords. Returns a paginated list of records within a registry.
- listRegistries. Returns a paginated list of registries.
- readRoles. Returns the role grants on a record, registry, or globally. Anchoring write tools (prepare-sign-submit pattern)
- prepareAddRegistry. Builds an unsigned transaction to create a new registry. The signer signs externally; the signed transaction is broadcast separately.
- prepareAddRecord. Builds an unsigned transaction to add a record to a registry.
- prepareUpdateRecordStatus. Builds an unsigned transaction to update a record's status.
- prepareGrantRole. Builds an unsigned transaction to grant a role at a specified scope.
- prepareRevokeRole. Builds an unsigned transaction to revoke a role at a specified scope. Operational tools
- broadcastSignedTransaction. Accepts a signed transaction and broadcasts it to the chain. The server validates the format but does not modify the signature.
- getServerInfo. Returns server version, supported protocol version, and operational status.
- getAuditLog. Returns the audit log for the calling principal, paginated. Records every tool call with timestamp, arguments, and outcome.

Appendix D: References

References are listed in the order they appear in the document. Footnote N in the body corresponds to reference [N] below. A source cited in more than one passage appears at each citation site. The Relevant Patents summary that closes this appendix groups patents by family and indicates which sections of the document each family supports.

D.1 Numbered References

[1] Boston Consulting Group and Ripple. "Approaching the Tokenization Tipping Point." 2025. [2] RWA.xyz. "Market Overview." Live analytics dashboard. Accessed April 30, 2026. rwa.xyz. [3] Bank for International Settlements. "Artificial Intelligence and the Economy: Implications for Central Banks." BIS Annual Economic Report 2024, Chapter III. 2024.

[4] 15 C.F.R. Parts 730–774, Export Administration Regulations. Electronic Code of Federal Regulations, current as of May 2026. Controls on advanced computing integrated circuits and AI-related items are distributed across Categories 3, 4, and 5 of the Commerce Control List.

[5] McKinsey & Company. "From Ripples to Waves: The Transformational Power of Tokenizing Assets." 2024. [6] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. January 2023. doi.org/10.6028/NIST.AI.100-1.

[7] European Central Bank. "The Rise of Artificial Intelligence: Benefits and Risks for Financial Stability." Financial Stability Review. May 2024.

[8] Haber, Stuart, and W. Scott Stornetta. "How to Time-Stamp a Digital Document." Journal of Cryptology 3, no. 2 (1991): 99–111.

[9] Cerf, Vinton G., and Robert E. Kahn. "A Protocol for Packet Network Intercommunication." IEEE Transactions on Communications COM-22, no. 5 (May 1974): 637–648.

[10] Author's note. NVNM Chain is a sovereign Layer 1. Execution runs on Reth as an execution library; consensus runs on Commonware Simplex.

[11] Chan, Benjamin Y., and Rafael Pass. "Simplex Consensus: A Simple and Fast Consensus Protocol." IACR Cryptology ePrint Archive, Report 2023/463. 2023. eprint.iacr.org/2023/463.

[12] Commonware. commonware-consensus library, simplex module: Byzantine fault-tolerant agreement in partially synchronous networks, with block production at two network hops and finalization at three. Rust, dual-licensed MIT and Apache-2.0. docs.rs/commonware-consensus; commonware.xyz.

[13] Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem." ACM Transactions on Programming Languages and Systems 4, no. 3 (July 1982): 382–401.

[14] Castro, Miguel, and Barbara Liskov. "Practical Byzantine Fault Tolerance." Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation. 1999.

[15] Paradigm. "reth: Modular, contributor-friendly and blazing-fast implementation of the Ethereum protocol, in Rust." github.com/paradigmxyz/reth. Documentation at reth.rs.

[16] Chou, Brendan. "Simplex, Pipelined." Commonware. August 12, 2026. commonware.xyz/blogs/simplex-pipelined.html.

[17] Buterin, Vitalik. "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform." 2014.

[18] Wood, Gavin. "Ethereum: A Secure Decentralised Generalised Transaction Ledger." Yellow Paper. 2014, with subsequent revisions.

[19] Merkle, Ralph C. "A Digital Signature Based on a Conventional Encryption Function." Advances in Cryptology (CRYPTO '87), Lecture Notes in Computer Science 293 (1988): 369–378.

[20] Laurie, B., A. Langley, and E. Kasper. "Certificate Transparency." RFC 6962, Internet Engineering Task Force. June 2013.

[21] Relevant Inveniam patent families for anchoring methodology and registry-based ledger structures include the Document Validation family (U.S. Patent Nos. 10,419,225; 11,044,100; 12,192,371), the Auditing of Electronic Documents family (U.S. Patent Nos. 10,817,873; 11,580,534), Blockchain Data Structures (U.S. Patent No. 12,231,566), and Private Ledger Processing (U.S. Patent No. 12,137,179).

[22] National Institute of Standards and Technology. FIPS 180-4, Secure Hash Standard (SHS). 2015. [23] Latour, Bruno. "Visualisation and Cognition: Drawing Things Together." In Knowledge and Society: Studies in the Sociology of Culture Past and Present 6 (1986): 1–40.

[24] Coalition for Content Provenance and Authenticity. C2PA Technical Specification, version 2.4. 2026. [25] Relevant Inveniam patent families for Proof of Origin include U.S. Patent Nos. 10,419,225; 11,044,100; 10,411,897; 10,693,652; 11,296,889; 11,170,366. Patents 10,419,225 and 11,044,100 are also relevant to Proof of State. [26] Relevant Inveniam patent families for Proof of Process include U.S. Patent Nos. 11,044,097; 11,580,535; 11,587,074; 10,685,399; 11,443,370; 11,443,371; 11,468,510; 11,348,098; 11,687,916; 12,597,066. Several of these are also relevant to Proof of State.

[27] Relevant Inveniam patent families for Proof of State include U.S. Patent Nos. 10,419,225; 11,044,100; 10,817,873; 11,580,534; 10,685,399; 11,443,370; 11,443,371; 11,468,510.

[28] Author's note. This document is itself an instance of the proof primitives it describes. Its SHA-256 checksum, the registry to which it is anchored, the anchoring transaction hash, and the block height at which the anchor is recorded are listed in the front matter of every published version. A reader can verify integrity by recomputing the document checksum and querying the registry on NVNM Chain.

[29] Anthropic. "Introducing the Model Context Protocol." November 2024.

[30] Model Context Protocol. "Specification." Version 2025-11-25. Current as of April 2026. [31] Model Context Protocol. "Transports." Specification companion document. Current as of April 2026. [32] Anthropic. "Donating the Model Context Protocol and Establishing the Agentic AI Foundation." 2025. [33] National Institute of Standards and Technology. FIPS 186-5, Digital Signature Standard (DSS). 2023. [34] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. January 2023. doi.org/10.6028/NIST.AI.100-1.

[35] Booth, H., B. Fisher, R. Galluzzo, and J. Roberts. "Accelerating the Adoption of Software and AI Agent Identity and Authorization Concept Paper." NCCoE / NIST. Draft, February 5, 2026.

[36] Relevant Inveniam patent families for decisional architectures and AI verification include U.S. Patent Nos. 11,348,098; 11,687,916 (Decisional Architectures); 10,270,599 (Data Reproducibility Using Blockchains); 12,192,371 (AI in Federated Learning).

[37] Relevant Inveniam patent families for federated data structures and device-usage recordation include U.S. Patent No. 12,597,066 (Federated Data Room Server) and the Recordation of Device Usage family (U.S. Patent Nos. 11,044,097; 11,580,535; 11,587,074).

[38] 15 C.F.R. Parts 730–774, Export Administration Regulations. Electronic Code of Federal Regulations, current as of May 2026. Controls on advanced computing integrated circuits and AI-related items are distributed across Categories 3, 4, and 5 of the Commerce Control List.

[39] 15 C.F.R. Part 748, Supplement No. 10, Data Center Validated End User authorization. Electronic Code of Federal Regulations, current as of May 2026.

[40] Author's note. The Bureau of Industry and Security rescinded the January 2025 "Framework for Artificial Intelligence Diffusion" on May 13, 2025, before its compliance date. Pre-existing EAR controls on advanced computing items remain in force. BIS issued a final rule effective January 15, 2026 revising license review policy for certain advanced computing semiconductors. Further BIS rulemaking is expected. Specific obligations cited above derive from EAR provisions in force as of May 2026, not from the rescinded interim final rule.

[41] Author's note. The compliance pattern described here is illustrative. Operators of regulated AI compute and similar audit-bound activities should consult counsel and the controlling regulator for specific reporting, retention, and inspection obligations.

[42] McKinsey & Company. "From Ripples to Waves: The Transformational Power of Tokenizing Assets." 2024. [43] Boston Consulting Group and Ripple. "Approaching the Tokenization Tipping Point." 2025. [44] DefiLlama. RWA Perpetuals Dashboard. Accessed April 30, 2026. defillama.com.

[45] RWA.xyz. "Market Overview." Live analytics dashboard. Accessed April 30, 2026. rwa.xyz. [46] Crisanto, Juan Carlos, Cris Benson Leuterio, Jermy Prenio, and Jeffery Yong. "Regulating AI in the Financial Sector: Recent Developments and Main Challenges." FSI Insights on Policy Implementation No. 63, Bank for International Settlements. 2024.

[47] Barr, Michael S. "AI, Fintechs, and Banks." Speech, Board of Governors of the Federal Reserve System. April 4, 2025.

[48] Financial Conduct Authority. "AI Update." 2024.

[49] U.S. Food and Drug Administration. Considerations for the Use of Artificial Intelligence to Support Regulatory Decision-Making for Drug and Biological Products. Draft Guidance for Industry. January 2025. [50] European Medicines Agency. Reflection Paper on the Use of Artificial Intelligence in the Medicinal Product Lifecycle. 2024.

[51] International Council for Harmonisation. Guideline for Good Clinical Practice E6(R³). Step 4 Final. January 6, 2025.

[52] Ferreira, Bruno, Rafael Borges, et al. "Can Secure Multi-Party Computation Be Used to Create Clinical Trial Cohorts Based on Blockchain-Notarized Private Patient Data?" Procedia Computer Science 256 (2025): 996–1002. doi.org/10.1016/j.procs.2025.02.205.

[53] Borges, Rafael, et al. "Using Secure Multi-Party Computation to Create Clinical Trial Cohorts." Journal of Cybersecurity and Privacy 6, no. 1 (2026), article 2. doi.org/10.3390/jcp6010002.

[54] Financial Action Task Force. Updated Guidance for a Risk-Based Approach to Virtual Assets and Virtual Asset Service Providers. October 2021. Targeted Update on Implementation of the FATF Standards on Virtual Assets and VASPs. 2025.

[55] World Wide Web Consortium. "Decentralized Identifiers (DIDs) v1.0." W3C Recommendation. 2022. [56] National Institute of Standards and Technology. "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile." NIST AI 600-1. July 2024.

[57] Abu Dhabi Global Market, Financial Services Regulatory Authority. "Digital Assets." Official ADGM page, current as of 2026. Guidance: Regulation of Virtual Asset Activities in ADGM, version VER06.020125. 2025. [58] Regulation (EU) 2023/1114 of the European Parliament and of the Council of 31 May 2023 on Markets in Crypto-Assets (MiCA). Official Journal of the European Union.

[59] Virtual Assets Regulatory Authority (Dubai). Rulebooks and Virtual Asset Issuance Rulebook. Version dated May 19, 2025.

[60] World Wide Web Consortium. "Verifiable Credentials Data Model v2.0." W3C Recommendation. 2024. [61] Coalition for Content Provenance and Authenticity. C2PA Technical Specification, version 2.4. 2026. [62] World Wide Web Consortium. "Decentralized Identifiers (DIDs) v1.0." W3C Recommendation. 2022. [63] HL7 International. FHIR Specification, Release 5 (R5). First published 2023.

[64] National Institute of Standards and Technology. Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations. NIST AI 100-2e2025. 2025.

[65] Szabo, Nick. "Trusted Third Parties Are Security Holes." 2001.

D.2 Relevant Patents

The following granted U.S. patents are referenced throughout the document at the sections where the corresponding technology is described. Some patents are relevant to more than one of the proof primitives in Section 4.5; these are noted in the corresponding entry. The full Inveniam portfolio comprises additional issued patents and pending applications beyond those cited in this whitepaper.

U.S. Patent No. 10,270,599. Data Reproducibility Using Blockchains. Cited at Section 4.8 (decisional architectures and AI verification).

U.S. Patent No. 10,411,897. Secret Sharing via Blockchains. Cited at Section 4.5 (Proof of Origin). U.S. Patent No. 10,419,225. Validating Documents via Blockchain. Cited at Section 4.5 (Proof of Origin and Proof of State) and at Section 4.3 (anchoring methodology).

U.S. Patent No. 10,685,399. Due Diligence in Electronic Documents. Cited at Section 4.5 (Proof of Process and Proof of State).

U.S. Patent No. 10,693,652. Secret Sharing via Blockchain Distribution. Cited at Section 4.5 (Proof of Origin). U.S. Patent No. 10,817,873. Auditing of Electronic Documents. Cited at Section 4.5 (Proof of State) and at Section 4.3 (anchoring methodology).

U.S. Patent No. 11,044,097. Blockchain Recordation of Device Usage. Cited at Section 4.5 (Proof of Process) and at Section 4.9 (data layer).

U.S. Patent No. 11,044,100. Validating Documents. Cited at Section 4.5 (Proof of Origin and Proof of State) and at Section 4.3.

U.S. Patent No. 11,170,366. Private Blockchain Services. Cited at Section 4.5 (Proof of Origin). U.S. Patent No. 11,296,889. Secret Sharing via Blockchains. Cited at Section 4.5 (Proof of Origin). U.S.

Patent No. 11,348,098. Decisional Architectures in Blockchain Environments. Cited at Section 4.5 (Proof of Process) and Section 4.8.

U.S. Patent No. 11,443,370. Due Diligence in Electronic Documents. Cited at Section 4.5 (Proof of Process and Proof of State).

U.S. Patent No. 11,443,371. Due Diligence in Electronic Documents. Cited at Section 4.5 (Proof of Process and Proof of State).

U.S. Patent No. 11,468,510. Due Diligence in Electronic Documents. Cited at Section 4.5 (Proof of Process and Proof of State).

U.S. Patent No. 11,580,534. Auditing of Electronic Documents. Cited at Section 4.5 (Proof of State) and Section 4.3. U.S. Patent No. 11,580,535. Recordation of Device Usage to Public/Private Blockchains. Cited at Section 4.5 (Proof of Process) and Section 4.9.

U.S. Patent No. 11,587,074. Recordation of Device Usage to Blockchains. Cited at Section 4.5 (Proof of Process) and Section 4.9.

U.S. Patent No. 11,687,916. Decisional Architectures in Blockchain Environments. Cited at Section 4.5 (Proof of Process) and Section 4.8.

U.S. Patent No. 12,137,179. Private Ledger Processing. Cited at Section 4.3 (anchoring methodology). U.S. Patent No. 12,192,371. AI in Federated Learning. Cited at Section 4.8 (verification of agent processing). U.S. Patent No. 12,231,566. Blockchain Data Structures. Cited at Section 4.3 (anchoring methodology). U.S. Patent No. 12,597,066 B2. Federated Data Room Server and Method for Use in Blockchain Environments. Cited at Section 4.5 (Proof of Process) and Section 4.9 (data layer).

Appendix E: Illustrative Reviewer Accountability Schedule

Section 5.5 describes an oversight pattern in which structured reviewer accountability is layered onto agent workflows where review quality is measurable and consequential. The schedule below is one parameterization operators can adopt. It is illustrative. Operators can choose other parameterizations, different review queues, or no staked-reviewer model at all. The chain anchors whatever the operator commits to, so that the operator's commitments and the reviewers' actions can be independently checked.

Infraction Tier	Penalty	Consequence
Tier 1: Poor quality or unresponsive	5%	Thirty-day review suspension
Tier 2: Repeated negligence	15%	Ninety-day suspension; escalated review by the operator
Tier 3: Malicious or collusive behavior	50%	Permanent removal from the reviewer set

Table 1. Illustrative reviewer accountability schedule.

Acknowledgments

The work in this paper would not exist without Inveniam, which contributed more than fifteen years of accumulated research and twenty-two granted U.S. patents in document validation, decisional architectures, and proof-of-process to the design.

Paul Snow built the original Factom protocol that established the data anchoring primitives this chain extends. The early Factom team, including Brian Deery, Mahesh Paolini-Subramanya, and Jason Nadeau, did the foundational work on hashing, validation, and replicated state on which much of the technical architecture in Section 4 rests. Their contributions are visible in nearly every patent family cited in Appendix D.

G42 has been an exceptional partner. The author thanks G42's leadership for forward-thinking and visionary support of the broader effort to build credible AI infrastructure in regulated jurisdictions. John Neiman, Dr. Fiacc Larkin, and Dr. Robin Mills each contributed materially to the thinking that led to this paper. Their willingness to engage seriously with the technical and regulatory questions made it possible to design for the use cases that actually matter.

Professor Alexander Lipton and Dr. Edward Jung at ADIA Labs have been generous with their time and intellectual rigor. The framing of attestation as proof of existence, distinct from proof of truth, owes a great deal to their input.

Bruce Fenton and the attendees of the Satoshi Roundtable created a space where ideas can be tested honestly on the path to maturity. Many of the design decisions in this paper were sharpened in conversations there. The author is grateful for the candor and the company.

Cryptographic timestamping was set down in 1991 by Stuart Haber and W. Scott Stornetta. Byzantine fault-tolerant consensus was set down in 1982 by Lamport, Shostak, and Pease, then refined for production use by Castro and Liskov in 1999. The Reth execution client and the Commonware consensus library were built by teams working in the open, and the chain's execution and consensus layers build directly on that work. The Model Context Protocol was contributed by

Anthropic and donated to an open foundation. NIST, the W3C, HL7, and the C2PA have produced the standards the chain depends on. The Foundation acknowledges its debt to all of this work.

Key Terms

Recurring vocabulary at a glance. Appendix A contains the full glossary.

Anchoring. The act of recording a cryptographic checksum of an off-chain artifact onto NVNM Chain, together with metadata sufficient to identify the artifact and its lifecycle state.

Registry. A permissioned namespace on NVNM Chain that contains records. Each registry has its own administrators, editors, and access controls.

Record. An attestation entry within a registry. Each record carries a checksum, a checksum algorithm identifier, a URI reference, structured metadata, and a status.

Provenance. The verifiable chain of custody connecting a piece of data to its source, its processing history, and its state at a given time. Composed of Proof of Origin, Proof of Process, and Proof of State. MCP server. The Model Context Protocol server hosted at mcp.nvnmchain.io, providing typed, authenticated access to NVNM Chain for AI agents and developer tools.

KYA. Know Your Agent. The convention by which each AI agent operating against NVNM Chain has a unique persistent onchain wallet, never shared with another agent. The agent's actions are forever tied to that wallet, and counterparties verify what an agent has done by querying its wallet onchain. Settlement currency. The root asset the protocol uses internally to route and settle transaction fees. Participants pay in eligible stablecoins and are not required to hold the settlement currency.

Inveniam.io. The decentralized data management platform for private market assets. At launch, the canonical first integrated data layer into NVNM Chain. Inveniam.io performs federated extraction and attribution, then submits cryptographic proofs to the chain. Inveniam (without the .io) refers to the company that operates the platform.