LIGATE

The Receipt Layer for AI.

An open, permissionless attestation protocol,
and the flagship applications built on it.

Whitepaper v2.0

April 2026

Ligate Labs · ligate.io
Themisra · themisra.xyz
Kleidon · kleidon.xyz
Iris · ligate.io/iris

WORKING DRAFT // FOR DISCUSSION

Table of Contents

1. Abstract

2. Introduction

2.1 The Problem

2.2 The Opportunity

2.3 Our Approach

3. Ligate Chain: The Attestation Protocol

3.1 Architecture Overview

3.2 Sovereign SDK and Celestia

3.3 Protocol Primitives

3.3.1 Formal Type Signatures

3.4 Permissionless by Design

3.5 Trust Model

3.6 Non-Goals

3.7 Threat Model and Security Analysis

4. Flagship Applications

4.1 Themisra: Proof of Prompt

4.2 Kleidon: Web3 SaaS Suite

4.3 Iris: Attestation for AI Agents

4.4 Worked Example: EU AI Act Compliance

4.5 Beyond AI: General-Purpose Attestation

5. Tokenomics: $LGT

5.1 Distribution

5.2 Fee Structure

5.3 Value Accrual

5.4 Stake-backed Attestor Economics (v1)

6. Competitive Positioning

7. Technology Stack

8. Roadmap

9. Revenue Model

10. Risk Factors

11. Conclusion

About the founder

References

PROLOGUE

1. Abstract

Abstract

*This paper presents **Ligate Labs** and **Ligate Chain**: a sovereign blockchain built on the Sovereign SDK and Celestia, exposing a single primitive, the **attestation protocol**. The protocol lets anyone register a schema (the shape of an attestation), register an attestor set (the signers authorised to sign under it), and submit attestations that are stored on chain forever and verifiable by anyone. The chain stores only hashes and signatures, never plaintext. Three flagship applications ship alongside the protocol and validate it at launch: **Themisra** (Proof of Prompt for AI provenance), **Kleidon** (a full-stack Web3 SaaS suite), and **Iris** (an MCP server and relayer that brings attestation to autonomous AI agents). All three use the same permissionless primitive. Third parties can register their own schemas and attestor sets on the same protocol. $LGT captures fees across every attestation across every application, independent of which application wins. Ligate is positioned as infrastructure for the decade of regulated, verifiable AI.*

Keywords: attestation, provenance, sovereign rollup, Sovereign SDK, Celestia, zero-knowledge, AI agents, Model Context Protocol, regulated AI.

CONTEXT

2. Introduction

2.1 The Problem

Every AI interaction today is invisible. A ChatGPT response, a generated image, a model's training data, an autonomous agent placing a trade, a human reviewing a medical recommendation: none of these come with a receipt. There is no cryptographic record of what model was used, what prompt produced what output, whether a human signed off, whether an image is synthetic or not. The information simply does not exist.

At small scale this was tolerable. At the scale AI is reaching in 2026 it is not. Regulation has arrived (the EU AI Act is live, US disclosure mandates follow), creators demand attribution and royalty rights, and enterprises deploying autonomous agents in regulated industries need defensible audit trails. The gap between what AI produces and what can be proven is now a market problem, not a research curiosity.

2.2 The Opportunity

Every AI-generated artefact will eventually need a receipt. That receipt is a small data structure: who generated it, when, with which model, under whose authority, and cryptographically signed so it cannot be repudiated. Receipts compose: an enterprise audit trail is a sequence of receipts; a creator royalty claim is a receipt with a schema that enforces downstream payout; a regulatory filing is a receipt with a schema aligned to a specific statute.

Building this as a single closed product would re-fragment a market that benefits from openness. Anthropic, OpenAI, enterprises, regulators, creators, and the thousands of downstream applications that consume AI all have different attestation shapes. What is shared is the primitive: *a signed, timestamped, on-chain record that someone authorised claims something happened.* That primitive is what Ligate Chain supplies.

2.3 Our Approach

Ligate Chain is deliberately narrow. It exposes exactly three primitives (Schema, AttestorSet, Attestation), enforces write-once semantics on attestation pairs, and stops there. Identity, reputation, slashing, cryptographic inference verification, payments, and agent registries are explicitly out of scope for v0; they are planned as separate modules composed on top of the primitive, not baked into it.

Three flagship applications ship alongside the protocol to prove that the primitive is enough. **Themisra** registers the first canonical schema, themisra.proof-of-prompt/v1, for AI prompt and output attestation. **Kleidon** registers schemas for subscription events, asset mints, token deployments, and marketplace sales across a Web3 SaaS suite. **Iris** ships an open source Model Context Protocol server plus a USD-billed relayer so autonomous agents can attest their actions in ten lines of integration code. None of these are baked into the chain. A third party can ship a competing AI provenance app, a competing subscription NFT product, or a competing agent attestation service by registering their own schemas and attestor sets. Ligate wins by being the

default and shipping the best reference implementations, not by closing the gate.

> *Ligate is the protocol. Themisra, Kleidon, and Iris are the flagships.*

PROTOCOL

3. Ligate Chain: The Attestation Protocol

3.1 Architecture Overview

Ligate Chain is a sovereign rollup built with the **Sovereign SDK**, using **Celestia** for data availability. It runs as a permissioned sequencer at v0 and decentralises sequencing in later phases, following the same trajectory as Base, Arbitrum, and Optimism. Cross-chain interoperability is provided by **Hyperlane** for asset bridging (used by Kleidon products) and by **Relay** for shared liquidity. Wallet auth and onboarding are provided by **Privy** (email + wallet, free up to 10K monthly active users).

The attestation protocol is implemented as the attestation module in the rollup, following Sovereign SDK module conventions. Future modules ship across two phases: v1 introduces staking and disputes (stake-backed attestor sets and slashing), identity (DID-style lookups), and tokens plus nft (the primitives Kleidon products consume). v2 adds payments (metered billing) and agents (on-chain agent wallets, distinct from the v0.5 Iris service). The protocol stays narrow at v0 by design and earns each subsequent module on shipped traction.

3.2 Sovereign SDK and Celestia

Sovereign SDK was chosen over alternative rollup frameworks for three concrete reasons. First, it is ZK-native: attestation workloads benefit from provable properties (timestamp, attestor identity, optionally private payload proofs) without retrofitting them onto EVM semantics. Second, it is Rust-native with RISC Zero and SP1 integration, which gives us room to tune per-operation cost at the inference-level attestation volumes we expect, where EVM gas would be prohibitive. Third, it produces sovereign rollups rather than enshrined L2s, which means Ligate retains upgrade authority. For a protocol whose role is to govern provenance itself, depending on another chain's governance politics would be a category error.

Celestia handles data availability. Rollups only need DA, not execution, and Celestia offers the cheapest production DA today with a clear path to scale. DA remains swappable: the module boundary between settlement and DA is explicit in Sovereign SDK, and Ligate can add alternative DA layers without rewriting the application.

3.3 Protocol Primitives

The attestation module exposes three primitives, each backed by a StateMap.

Primitive	Key fields	Notes
Schema	id, owner, name, version, attestor_set, fee_routing_bps, fee_routing_addr	Permissionless registration. Immutable per id. Namespace collisions resolved off-chain via owner address and verified-tag metadata, same model as ERC-20 names.
AttestorSet	id, members (PubKey[]), threshold	Immutable once registered. Rotation happens by registering a new set and bumping the schema version.

Primitive	Key fields	Notes
Attestation	schema_id, payload_hash, submitter, timestamp, signatures	Write-once per (schema_id, payload_hash). Replay is structurally impossible. Chain stores hashes and signatures only, never plaintext.

Table 1: The three primitives exposed by the attestation module.

The signed payload is the canonical Borsh encoding of (schema_id, payload_hash, submitter, timestamp). Signatures do not cover the signatures field itself (that would be circular). Replay is prevented by the write-once invariant on (schema_id, payload_hash): the same pair cannot be submitted twice, so a captured signature cannot be replayed.

3.3.1 Formal Type Signatures

The three primitives are formally specified as the following Rust types in the attestation module. Identifiers use the Bech32 prefix ladder defined in constants.toml (lig1 account addresses, lpk1 public keys, lsc1 schema IDs, las1 attestor-set IDs, lph1 payload hashes; a lat1 wrapper for the compound attestation ID is planned pre-mainnet, tracked in chain repo issue #64).

```
struct Schema {
    id:                SchemaId,           // lsc1...
    owner:             Address,            // lig1...
    name:              String,
    version:           u32,
    attestor_set:      AttestorSetId,      // las1...
    fee_routing_bps:   u16,                // 0..=5000 (max 50% to schema owner)
    fee_routing_addr:  Address,
    payload_shape_hash: Hash,              // off-chain spec digest
}

struct AttestorSet {
    id:        AttestorSetId,              // las1...
    members:   Vec<PubKey>,                // lpk1... per signer
    threshold: u8,                         // m-of-n required for valid attestation
}

struct Attestation {
    schema_id:   SchemaId,                 // foreign key into Schema
    payload_hash: PayloadHash,             // lph1... = sha256(borsh(off-chain payload))
    submitter:   Address,
    timestamp:   u64,                      // unix seconds, set by submitter
    signatures:  Vec<AttestorSignature>,   // must satisfy AttestorSet.threshold
}

struct SignedAttestationPayload {
    schema_id:    SchemaId,
    payload_hash: PayloadHash,
    submitter:    Address,
    timestamp:    u64,
}

// Stored in the runtime's StateMap, keyed by (schema_id, payload_hash):
//   StateMap<AttestationId, Attestation>
// where AttestationId = (SchemaId, PayloadHash).
```

Listing 1: Rust type signatures for the attestation module's three primitives. Storage uses Sovereign SDK's StateMap; (schema_id, payload_hash) is the unique attestation key, enforcing the write-once invariant at the type level.

3.4 Permissionless by Design

Anyone can register an AttestorSet with any set of public keys. Quality is a social concern, not a protocol gate. Anyone can register a Schema with any name; two different owners registering the same name produce two different schema_id values because the owner address is hashed into the ID. Anyone can submit an Attestation under any schema, provided they supply valid signatures that meet the schema's AttestorSet threshold. The AttestorSet is the permission layer.

Spam is mitigated economically, not by gating: registration requires a fee in $LGT (default 10 for AttestorSets, 100 for Schemas) that routes to the protocol treasury. Per-attestation fees (default 0.001 $LGT) are split between the schema's designated builder address and the treasury, with the split controlled by the schema owner at registration (capped at 50% to the builder).

3.5 Trust Model

The trust story is staged and honest. At v0, attestor sets are federated: whoever registers a schema operates its signers, and the correctness of attestations under that schema depends on those signers being trustworthy. This is enough to ship, enough for regulated pilots, and not a permanent state.

Phase	Trust model	Flagship
v0 (devnet)	Federated attestors: "signers you trust"	Themisra
v1 (next)	Federated plus slashing via the disputes module	Themisra + first Kleidon product
v2 (later)	Partner cryptographic verification (zkML / TEE proof hashes carried in attestation payloads)	Ecosystem apps

Table 2: Three-phase trust arc for Ligate Chain.

Ligate does not build its own zkML. Customers who need cryptographic inference verification in v2 bring their own proving system (Ora, Ezkl, Succinct, or a future partner) and register schemas whose payload format carries the external proof hash. Verification is done off-chain by the consuming application. This keeps the protocol narrow and lets the proving ecosystem evolve without forcing Ligate to pick a winner.

3.6 Non-Goals

The protocol deliberately does not provide any of the following in v0:

- **Identity.** DID-style lookups (address to handle, pubkey, metadata URI) arrive as the v1 identity module.
- **Stake-backed attestors and slashing.** Bonded capital, fee-share emissions, on-chain flagging, and attestor slashing arrive as the v1 staking and disputes modules.
- **Token and NFT primitives.** Native token issuance, transfers, NFT minting, and metadata standards (the primitives Kleidon products consume) arrive as the v1 tokens and nft

modules.

- **Cryptographic inference verification.** v2 schemas such as themisra.verified-inference/v1 carry external zkML or TEE proof hashes. Clients verify with the external proving system that generated them. Ligate stays neutral.
- **Payments, metered billing.** v2 payments module extends the standard bank module.
- **Agent registry.** On-chain agent wallets with allowlists and rate limits arrive as the v2 agents module, distinct from the v0.5 Iris product.

3.7 Threat Model and Security Analysis

We enumerate the realistic adversaries against the v0 attestation protocol and the structural defence each one runs into. v1 (stake-backed attestors with slashing, covered in Section 5.4) hardens this further; v0 already gives meaningful guarantees against the most common attacks.

Adversary	Goal	Defence
Replay attacker	Re-submit a captured signed payload to inflate counts or deceive a verifier.	Write-once invariant on (schema_id, payload_hash). The chain rejects duplicate keys; a captured signature cannot be re-anchored under the same schema. See Section 3.3.
Schema squatter	Register the canonical name of an upcoming application to extort or confuse.	schema_id is hashed from (owner, name, version). Two owners registering the same name produce two different IDs. Off-chain registries (verified tags, social signals) resolve display, same model as ERC-20 names. No protocol-level fix attempted because there is no canonical "true" owner of an arbitrary name.
Malicious schema owner	Set fee_routing_bps to skim from honest attestors.	fee_routing_bps is capped at 5000 (50%) by the protocol. Set at registration, immutable per schema version. Clients can inspect the routing before submitting; if unfavourable, a competing schema can be registered.
Compromised attestor (single signer)	Sign a fraudulent attestation under a federated AttestorSet.	AttestorSet enforces a threshold (m-of-n). A single compromised key cannot meet the threshold alone. v1 adds slashing: a successful dispute against the set burns bonded stake, making compromise economically costly even when the threshold is met.
Colluding attestor majority	Coordinate to sign fraudulent attestations under a federated set.	v0: not protocol-prevented; the set's social and reputational stake is the deterrent. v1 (Section 5.4): bonded stake is slashable on successful fraud proofs, with a bounty paid to the disputer. Economic asymmetry makes collusion increasingly expensive as stake grows.
Spam attacker	Flood the chain with worthless attestations to degrade availability or storage.	Per-attestation fee (default 0.001 $LGT) makes large-volume spam economically uninteresting. Schema and AttestorSet registration fees (100 and 10 $LGT) provide further friction at the schema level.
Privacy attacker	Recover prompt or output content from on-chain attestations.	The chain stores hash and signatures only. Plaintext payloads never touch the chain. Recovery requires inverting sha256, which is infeasible for non-trivial payloads. Schemas that need privacy can use additional commitments (e.g. Pedersen) before hashing, with the protocol unaffected.

Adversary	Goal	Defence
Censorship by sequencer	Refuse to include valid attestations from a particular submitter.	v0 runs a permissioned sequencer; censorship is possible but auditable (clients can detect missing attestations). Mitigation path matches Base/Arbitrum/Optimism: progressive sequencer decentralisation post-mainnet. The sovereign rollup architecture means we control this trajectory directly, no enshrined L2 dependency.

Table 2: Adversary models against the v0 attestation protocol and the structural defences in place. v1 hardens the attestor-set models with bonded stake and slashing.

APPLICATIONS

4. Flagship Applications

Three applications ship alongside the protocol. All three are first-class consumers of the attestation module, none is privileged by the protocol, and all three exist to prove that the primitive is sufficient.

4.1 Themisra: Proof of Prompt

Themisra (themisra.xyz) is the flagship AI provenance application. Named after Themis, the Greek Titaness of divine law, it proves who prompted what, when, and with which model. Architecturally Themisra is one app among many that could be built on Ligate Chain. It registers the canonical schema themisra.proof-of-prompt/v1, operates the first attestor quorum (federated at v0, Ligate Labs plus two to four recruited partner organisations), and ships reference SDKs (@ligate/themisra on npm, themisra-pop on crates.io).

The Themisra client flow is deliberately simple:

```
themisra.attest(model, prompt, output):
  payload      = borsh((model, sha256(prompt), sha256(output),
                        timestamp, client))
  payload_hash = sha256(payload)
  signatures   = fetch from Themisra attestor quorum (HTTP)
  chain.SubmitAttestation { THEMISRA_SCHEMA_ID, payload_hash,
                            signatures }
  return receipt_id = (THEMISRA_SCHEMA_ID, payload_hash)


themisra.verify(receipt_id):
  return chain.get_attestation(receipt_id)
```

A user of the Themisra app chats normally with a chosen model; every message produces a payload, the attestor quorum signs the payload hash, and the signed hash is submitted to Ligate Chain. The user receives a receipt. The chain never sees prompt or output plaintext; only hashes.

v0 Themisra ships exactly this: native chat with Anthropic models, an external proof API that accepts prompt and output pairs from ChatGPT / Claude / Gemini / elsewhere, SDKs in TypeScript and Rust, a receipt verification page, and Privy auth. The ambitious feature surface (prompt marketplace, AI Battles, prompt NFTs, zkML private prompts) is the post-v0 roadmap, gated on demand and funding, not promised at launch.

4.2 Kleidon: Web3 SaaS Suite

Kleidon (kleidon.xyz) is a full-stack multi-chain Web3 SaaS suite, "we bind your business to the chain." It gives game studios, SaaS companies, and content creators a no-code operator dashboard and SDKs covering four product lines, all of which register their own schemas on the Ligate

attestation module and optionally bridge to EVM and Solana chains via Hyperlane.

Product	Schema	Capability
Passify	kleidon.subscription-event/v1	Subscription NFTs, transferable access passes, embeddable widget
SkinsVault	kleidon.asset-mint/v1, kleidon.asset-evolution/v1	Gaming asset manager with Unity and Unreal SDKs and dynamic metadata
TokenForge	kleidon.token-deploy/v1	No-code token deployment (native Ligate, ERC-20, SPL) with anti-cheat controls
MintMarket	kleidon.marketplace-sale/v1	White-label NFT marketplace with primary and secondary sales, gasless mint

Table 3: Kleidon product lines and their attestation schemas.

Operators choose where to deploy at setup. Native Ligate deployment is the cheapest and fastest and is the only path with full feature parity (native Themisra integration for AI-driven features, $LGT gas). Hyperlane bridging exposes Kleidon products to Ethereum, Base, Polygon, and Solana customers who want to stay in a specific ecosystem. In all deployment modes, on-chain events produce Kleidon-schema attestations on Ligate Chain, which means operators have a single, uniform source of verifiable truth for every subscription, mint, deployment, and sale, regardless of deployment target.

No direct competitor covers the full stack. Unlock Protocol covers subscriptions. Thirdweb serves developers. Crossmint handles minting. Kleidon covers all four, with one SDK and one dashboard, on an attestation-native chain.

4.3 Iris: Attestation for AI Agents

Iris (ligate.io/iris) is the Ligate MCP server and relayer. It targets autonomous AI agents: Claude Desktop, Cursor, Claude Code, Devin, Replit Agents, LangGraph, the OpenAI Agents SDK, and any custom MCP-compatible agent. In roughly ten lines of setup, an agent framework gains cryptographic attestation of every action it takes.

The problem Iris solves is straightforward. AI agents are increasingly autonomous: they execute code, place trades, write content, handle customer interactions, and make regulated decisions. Today, no one can verify what an agent actually did, only what it claims it did. Regulated industries cannot deploy agents without audit trails. Agent-to-agent trust is broken: if agent A passes work to agent B, agent B has no way to verify A's claims. Attestation solves this, but agents do not hold wallets, do not hold $LGT, and do not sign transactions. They need a standardised interface.

The Model Context Protocol, an open standard Anthropic introduced, is how modern agents call external tools. Iris ships an MCP server implementing four tools: record (post an attestation), verify (check an incoming attestation), query (look up historical records by agent, schema, or time), and schemas (list available attestation shapes). The server is open source, MIT licensed.

The relayer is the monetised half. Agents cannot hold $LGT, so Iris runs a sponsored service that receives MCP calls, signs and submits attestations on behalf of the agent's parent organisation,

bills the organisation monthly in USD or USDC, and covers the $LGT fee internally. This is the Alchemy, Thirdweb, Pimlico "sponsored gas" pattern.

Tier	Price	Attestations / month	Audience
Free	$0	1,000	Developers, hobbyists, testing
Pro	$49/mo	100,000	Individual devs, small teams, indie agent apps
Enterprise	Custom	Unlimited + SLA	Regulated industries, agent platforms

Table 4: Iris pricing tiers. Revenue is margin on USD billing over $LGT costs.

Iris is margin-positive by design. A thousand Pro subscribers is $49K MRR; ten enterprise customers at five thousand per month is $50K MRR. This is recurring USD revenue, independent of the $LGT price, and it strengthens the Series A pitch without relying on token speculation.

Iris is not a new chain module. It is a developer-facing service that uses the existing attestation module. The chain-level agent registry (on-chain agent wallets with allowlists) is a separate v2 concern.

4.4 Worked Example: EU AI Act Compliance

Article 50 of the EU AI Act, in force from 2026, requires providers of generative AI to mark machine-generated content in a machine-readable, detectable way. Voluntary disclosures and statistical watermarks fail under adversarial conditions. Ligate's attestation primitive maps cleanly onto this enforcement requirement. We walk through the concrete chain-level flow as a worked example of how a regulator (distinct from any of the three flagship apps above) would adopt the protocol.

The EU AI Office, or any designated authority, registers an AttestorSet whose members are public keys of national AI authorities (BfAI, CNIL, DPC, Garante, AEPD, AP, UODO, IMY, and so on). The set carries an m-of-n threshold (e.g. 5-of-9) so that no single agency can sign attestations alone. Registration costs 10 $LGT and produces an AttestorSetId with the las1 prefix.

The Office then registers a Schema named eu.ai-content/v1 bound to this AttestorSet. The off-chain payload shape covers the fields a regulator needs: model_id (e.g. openai/gpt-5), model_weights_hash, content_hash (sha256 of the generated output), content_type, generated_at, consent_disclosed, prompt_provided, and provider_country. Schema registration costs 100 $LGT and produces a SchemaId with the lsc1 prefix. fee_routing_bps can route up to 50% of attestation fees back to the Office, providing a built-in funding mechanism for the regulator.

AI providers (OpenAI, Anthropic, Mistral) sign and submit attestations under this schema at content-generation time. They do not need to hold $LGT or operate a wallet themselves: a relayer (the same pattern Iris uses for AI agents) sponsors the gas and bills the provider monthly in USD. Per-attestation cost is 0.001 $LGT. The chain stores the payload hash and the threshold-meeting signatures only, never the prompt or the output content. Privacy by design.

Three downstream query patterns then follow:

- **Citizen verification.** A user hashes a suspicious image and queries the chain. If an attestation exists under eu.ai-content/v1 with that content hash, the verifier returns model, timestamp, signing authorities, and consent flag. Absence of an attestation is itself a useful signal: the content was either human-made or generated by a non-compliant provider.
- **Regulator audit.** The Office queries all attestations submitted under a given (schema, attestor_set, model_id, time_window) tuple and compares the count and consent rate against a provider's self-reported compliance filing. Discrepancies become enforcement leads. Cryptographically anchored ground truth, not provider-trusted logs.
- **Court forensics.** A defamation case turns on whether a specific image was AI-generated on a specific date. The court hashes the image, queries the chain, and gets back a co-signed attestation with timestamp and signing authorities. Cryptographic evidence from independent national authorities, not "we trust the provider's logs."

Several design properties make this practical at AI-inference scale that EAS-on-Ethereum or naive on-chain logging cannot match. Per-attestation fees of 0.001 $LGT make billion-attestation-per-year volumes economically viable. Hash-only storage satisfies GDPR's data-minimisation mandate. Permissionless schema registration means the EU does not need anyone's permission to register their schema, and China, the US, India can each register their own without coordinating with the EU. The protocol is designed for a market of compliance regimes, not a single top-down standard.

4.5 Beyond AI: General-Purpose Attestation

The three flagship apps and the EU compliance worked example all sit in the AI domain because that is where the most acute regulatory and audit pressure exists in 2026. The protocol itself is content-agnostic. The attestation module does not know or care whether a payload hash points to a model output, a deed transfer, or a temperature reading. Anyone can register a schema for anything; the AttestorSet decides who is authorised to sign under it; the chain stores the hash and signatures. The full v0 stack supports these use cases today, with no protocol changes required.

A non-exhaustive sketch of schemas third parties could register on Ligate Chain today, by category:

Domain	Schema	What the attestation proves
Tokenised RWA	rwa.deed-transfer/v1	Title transfer for a tokenised real-asset representation. Attestor set: licensed title companies and notaries. Use case: on-chain proof a real-world asset was lawfully transferred at a given time, anchored to off-chain registry filings.
Smart-contract audits	audit.contract-review/v1	An audit firm has reviewed a specific contract bytecode hash and issued a finding. Attestor set: the audit firm itself, or a federated quorum of audit firms. Use case: counterparties verify a contract has been audited without trusting an off-chain badge.
Document signing	notary.document-sig/v1	A document with a given content hash was signed by a set of parties on a given date. Attestor set: licensed notaries or e-signature providers. Use case: tamper-evident signature proofs without storing the document on-chain.

Domain	Schema	What the attestation proves
Supply-chain provenance	supplychain.cold-chain-event/v1	An IoT sensor recorded that a shipment stayed within a temperature range for a given duration. Attestor set: the logistics operator and an independent verifier. Use case: pharmaceutical and food cold-chain compliance audits.
KYC and identity	kyc.individual-verified/v1	A KYC provider has verified an individual against a given regulatory regime, returning a hash of the verification record. Attestor set: licensed KYC providers. Use case: counterparties accept KYC without re-running it, while the provider still controls the underlying data.
Healthcare consent	health.ehr-access-grant/v1	A patient has granted access to specific EHR records for a specific purpose with a specific expiry. Attestor set: hospitals and consent-management platforms. Use case: HIPAA-aligned consent receipts that providers can present in audits without exposing record content.
Carbon and ESG	esg.carbon-credit-issuance/v1	A carbon credit was issued against a verified offset project. Attestor set: registries plus independent verification bodies. Use case: addresses the credit-double-counting problem with public, replay-resistant proof of issuance.

Table 4: Illustrative non-AI schemas. None require protocol changes. Each is an independent application of the same three v0 primitives.

Ligate Labs is not building these. The flagship apps (Themisra, Kleidon, Iris) focus on the AI wedge because it is the largest immediate market with the clearest regulatory tailwind, and because shipping three integrated apps proves the primitive faster than spreading thin across seven domains. The point of listing non-AI schemas here is to make the platform thesis explicit. The token does not depend on Ligate Labs being right about which schemas grow fastest. Every attestation in every schema, by every operator, pays the same per-attestation fee. This is the Ethereum and Uniswap pattern: the protocol survives any individual application.

TOKENOMICS

5. Tokenomics: $LGT

Ligate Network is powered by a single utility token, $LGT. A unified token eliminates liquidity fragmentation across the protocol and its flagship apps, and aligns incentives: every attestation anywhere in the ecosystem flows value to the same token, regardless of which application produced it.

5.1 Distribution

Total supply is fixed at **1,000,000,000 $LGT** with no inflationary mechanism beyond the initial distribution schedule.

Allocation	Percentage	Tokens	Vesting
Node Rewards	30%	300,000,000	Linear over 10 years
Team and Development	20%	200,000,000	4-year vest, 1-year cliff
Community and Airdrops	15%	150,000,000	Event-based distribution
Treasury (DAO)	15%	150,000,000	Governed by token holders
Early Investors	10%	100,000,000	2-year vest, 6-month cliff
Ecosystem Rewards	10%	100,000,000	Builder grants, creator incentives, attestor onboarding

Table 5: $LGT distribution.

5.2 Fee Structure

Three classes of fee exist at the protocol level, all denominated in $LGT.

Event	Default fee	Routing
RegisterAttestorSet	10 $LGT	100% treasury
RegisterSchema	100 $LGT	100% treasury
SubmitAttestation	0.001 $LGT	Split per schema: 0 to 50% to the schema's fee_routing_addr, remainder to treasury

Table 6: Protocol-level fees. All values are governance-adjustable.

Schema owners elect a routing percentage at registration, up to the builder cap (50% by default). Changing the split requires bumping the schema version, which means users of a schema implicitly opt in by migrating; bait-and-switch on existing users is structurally impossible.

Separately, sequencer operators and full-node operators share transaction fees and block rewards from the 300M Node Rewards allocation, on a decreasing emission curve intended to incentivise early participation and taper as fee volume grows.

5.3 Value Accrual

Because the protocol is permissionless, any third-party schema pays the same fees that Themisra, Kleidon, and Iris pay. Every attestation in the ecosystem, across every flagship or third-party application, accrues fee revenue to the same token. The token is not a speculative claim on a single product; it is the settlement asset for a category. If Themisra underperforms but Kleidon wins, $LGT value accrues. If a third-party schema outgrows all three flagships, $LGT value still accrues. This is the Ethereum and Uniswap analogy: the protocol survives any individual application.

5.4 Stake-backed Attestor Economics (v1)

v0 ships with federated attestor sets: trust is anchored in the social and reputational stake of whoever operates the set. This is sufficient for shipping a protocol and for regulator-led pilots, but it is not sufficient for adversarial settings where a private attestor's economic incentive to cheat is non-trivial. v1 introduces the staking and disputes modules to bond real capital against attestor sets and slash it on successful fraud proofs.

The model is intentionally close to EigenLayer-style restaking, scoped to attestation security. Anyone (including the attestor's own operators, third-party stakers, or downstream users of the schema) can stake $LGT against any registered AttestorSet. Stakers earn a configurable share (default 30%) of attestation fees from any schema bound to that set. If a successful dispute is filed against an attestation, the staked pool is slashed by a governance-set fraction (default 50% of the bonded amount), with a bounty of 20% of the slashed amount paid to the challenger.

Concretely, if an AttestorSet has bonded stake S (sum of all staker positions), and a malicious signature is committed to chain, an honest party can submit a fraud proof. On successful adjudication, the protocol burns 0.5 * S, pays 0.2 * (0.5 * S) = 0.1 * S to the disputer as bounty, and the attestation is flagged invalidated. Stakers backing the set absorb the loss in proportion to their share. The attestor set itself is not rotated automatically (that is a v2 concern); instead, stakers are expected to unbond and re-stake against more trustworthy sets, which is the market mechanism that prices attestor reliability.

The cryptoeconomic claim: a malicious signer with private benefit B from cheating faces expected loss L = P(detection) * 0.5 * S, where P(detection) is the probability that the fraud proof succeeds. For honest behaviour to dominate rationally, S and P(detection) must scale such that L > B at the margin. The bounty (0.1 * S) sets the disputer's economic incentive to actually file proofs, which raises P(detection) endogenously. As staked capital grows with usage, the size of malicious behaviour the system can resist scales linearly with S.

Two limits worth naming. First, this protects against signing fraud (an attestor asserting a payload that the off-chain payload shape does not actually entail), not against the more general problem of "the schema's payload shape is wrong" (a schema-design failure that the protocol cannot catch, only a competing schema or off-chain audit can). Second, the cryptoeconomic guarantee is meaningful only when S is large relative to B; v1 bootstrap stake will be modest and the guarantee correspondingly modest. The trajectory is the same as Ethereum's: stake grows with usage; security grows with stake.

LANDSCAPE

6. Competitive Positioning

Ligate operates at an underserved intersection: verifiable records for AI interactions, as infrastructure, not as a single product. Adjacent projects solve neighbouring problems but none is competing for the attestation substrate.

Project	Focus	Why it is adjacent, not competing
Bittensor (TAO)	Decentralised model inference and training	Compute network, not an attestation primitive. Cannot be used to verify who prompted what.
Ora, Ritual, Sertn	Verifiable inference (zkML and TEE execution)	Prove that a specific computation happened. Complementary to Ligate; v2 schemas can carry their proof hashes.
Gensyn, Akash	Decentralised compute markets	Sell compute, do not record what was computed or under whose authority.
Story Protocol	IP licensing and attribution on chain	Focused on traditional IP rights. Different schema space, different user base.
EAS (Ethereum Attestation Service)	Generic attestations on Ethereum	Closest competitor architecturally. Runs on Ethereum gas, which makes per-inference volume infeasible. Ligate fills the high-volume, AI-focused niche.
PromptLayer, Langfuse	Web2 prompt logging and analytics	Centralised logs, no cryptographic proof, no creator or operator royalty model.

Table 7: Competitive landscape. Ligate is closest to EAS architecturally and complementary to every verifiable-inference project.

The honest positioning: EAS proved attestation as a category on Ethereum. The gap Ligate fills is an attestation substrate with the throughput, fee model, and privacy posture required for AI-scale workloads, with first-class flagship apps that prove the primitive is enough.

STACK

7. Technology Stack

Layer	Technology
Rollup framework	Sovereign SDK (Rust)
Data availability	Celestia
ZK proving (v2 schema payloads)	SP1, RISC Zero, partner-supplied
Cross-chain bridging	Hyperlane
Shared liquidity	Relay
Auth / onboarding	Privy (email + wallet)
Monitoring	Grafana
Indexing	Native Sovereign SDK indexer
API docs	Swagger (auto-generated)
Client SDKs	TypeScript (npm), Rust (crates.io), Unity C#, Unreal C++
Smart contracts (EVM targets)	Foundry, Solidity, ERC-1155, ERC-20
Smart contracts (Solana)	Anchor, SPL tokens
Native token	$LGT

Table 8: Technology stack across protocol and flagship applications.

Launch infrastructure is deliberately lean: one sequencer node, two to three full nodes, and Celestia DA fees. Operating cost at devnet launch is projected at $110 to $250 per month. Sequencer decentralisation follows the same pattern as Base, Arbitrum, and Optimism: centralised at launch, progressively decentralised as fee revenue and attestor network mature.

TIMELINE

8. Roadmap

The roadmap is organised by protocol version and uses relative months from the start of the implementation phase. Absolute calendar dates are deliberately not committed here; devnet is targeted for mid-2026, with realistic slip tolerance into late 2026 depending on SDK integration work.

Phase	Window	Protocol	Flagship milestones
v0 devnet	Months 1 to 6	attestation module live on devnet; permissionless schema and attestor-set registration; federated trust	Themisra native chat + receipt; first Kleidon product schema (Passify); Iris MCP MVP with record and verify
v0.5 Iris + relayer	Months 4 to 8	Stable client SDKs (TS, Rust); paginated query RPC; observability	Iris relayer with Stripe and USDC billing; first enterprise pilot
v1	Months 8 to 14	identity and disputes modules; slashing-backed attestors; governance-controlled fee adjustment	Themisra creator economy (marketplace, royalties); two more Kleidon products; Iris at 100+ paying seats
v2	Months 14 to 24	payments and agents modules; proof-carrying schemas (zkML / TEE hashes)	Themisra confidential prompts; multi-chain Kleidon deployments; Iris ecosystem expansion

Table 9: Protocol and flagship milestones by phase.

> *Ship the primitive. Prove it with Themisra. Monetise it with Iris. Scale it with Kleidon.*

REVENUE

9. Revenue Model

Ligate has four distinct revenue surfaces, designed so that not all of them depend on the $LGT price or on token speculation.

- **$LGT attestation fee volume**. Every attestation, from every schema, across every application, pays a fee in $LGT. Value accrues to the token as volume grows. This is the long-term category bet.
- **Iris subscription revenue**. Recurring USD or USDC revenue from agent operators: free tier, $49/month Pro tier, custom Enterprise. Margin-positive over on-chain costs. Independent of $LGT price.
- **Themisra product subscriptions**. B2B plans for teams, SaaS, and creators who need managed provenance tooling, compliance dashboards, and integrations.
- **Kleidon SaaS fees**. Operator SaaS subscription, 2.5% primary sales fee, 1.0% secondary sales fee across Passify, SkinsVault, TokenForge, and MintMarket.

The Iris revenue line is the least token-dependent and the fastest to turn on, which makes it the most important for pre-Series-A traction. Themisra and Kleidon subscriptions scale with their respective product adoption. $LGT fee volume is the upside bet that pays off if the protocol becomes the default attestation substrate for AI.

RISKS

10. Risk Factors

- **Attestor recruitment.** v0 federated trust requires three to five independent attestor organisations for credible federation on the Themisra schema. If we cannot recruit them by mid-H2 2026, the trust story weakens. Mitigation: active design-partner pipeline, funded attestor onboarding grants from the Ecosystem Rewards allocation.
- **Sovereign SDK maturity.** The SDK is pre-mainnet itself. Integration bugs, breaking changes, and missing primitives may slip the devnet date. Mitigation: pinned SDK revision, close relationship with Sovereign Labs, willingness to contribute fixes upstream.
- **Regulatory ambiguity.** The EU AI Act is live but implementation guidance is still evolving; US disclosure mandates are not yet law. A regulatory pivot could reshape which schemas matter. Mitigation: the protocol is schema-agnostic; regulatory change widens demand, it does not invalidate the primitive.
- **EAS on Ethereum.** EAS could extend to AI-focused schemas and compete directly. Mitigation: Ethereum gas economics make per-inference volume infeasible; Ligate's differentiator is throughput and AI-native tooling, not the primitive itself.
- **Iris competitive fork.** Iris is open source and could be forked to a competing chain. Mitigation: $LGT network effects, relayer reliability, first-mover positioning, ecosystem integrations (Anthropic, Cursor, LangGraph partnerships).
- **MCP specification evolution.** Anthropic evolves MCP; Iris must track upstream changes. Mitigation: active participation in MCP working groups, fast response cadence to spec changes.
- **Concentration risk.** At v0, Ligate Labs operates the first attestor quorum for Themisra and the Iris relayer. A centralisation critique is fair and we do not hide it. Mitigation: the trust model is explicit about federation at v0, slashing at v1, and proof-carrying schemas at v2; the roadmap is the answer.

CLOSING

11. Conclusion

AI has outgrown the era in which its outputs were trusted on assertion. Creators need attribution, enterprises need audit, and regulators have started to legislate. The missing primitive is the receipt: a small, signed, timestamped record that something happened under someone's authority, verifiable by anyone later. That primitive is narrow enough to be a protocol, not a product, and broad enough to apply to every AI interaction, every autonomous agent action, every subscription event, and every marketplace sale.

Ligate Chain supplies exactly that primitive and nothing more. Themisra proves it works for AI provenance; Kleidon proves it scales to SaaS and gaming; Iris proves it reaches the entire autonomous-agent economy without any of them holding a wallet. Third parties can do the same thing, for any schema that makes sense in their domain. Value accrues to $LGT regardless of which application wins.

We are building infrastructure for the decade in which every AI-generated artefact carries a receipt. The protocol is the bet. The flagships are the proof.

> ***The protocol is the bet. The flagships are the proof.***

About the founder

Ligate Labs is led by **Stefan Stefanovic**, founder and CEO. Previously built a sovereign rollup at Fragment Chain (Rust sequencer, Solidity batching contracts, L2 proof generation). Senior engineer at Request Network and IOHK. Founder of Blockbyte since 2022. Eight years across protocol and product. Currently the single point of leverage on chain architecture, brand, marketing surface (ligate.io, docs.ligate.io), investor narrative, and design-partner conversations.

Hiring in parallel with the round, in priority order: protocol engineer (Sovereign SDK, Rust), head of growth (crypto-native and AI-native distribution), and one to two design-partner studios or SaaS teams committed by devnet. Active conversations with Sovereign Labs and Celestia ecosystem contacts. Post-pre-seed: ZK researcher, devrel, product designer.

References

1. Sovereign Labs. Sovereign SDK documentation. github.com/Sovereign-Labs/sovereign-sdk
2. Celestia. Modular data availability. celestia.org
3. Hyperlane. Permissionless interoperability. hyperlane.xyz
4. Relay. Cross-chain liquidity infrastructure. relay.link
5. Privy. Embedded wallet infrastructure. privy.io

6. Succinct Labs. SP1 zkVM. github.com/succinctlabs/sp1

7. RISC Zero. General-purpose zkVM. github.com/risc0/risc0

8. Anthropic. Model Context Protocol specification. modelcontextprotocol.io

9. Ethereum Attestation Service. Attestation schema registry on Ethereum. attest.sh

10. European Union. AI Act, Regulation (EU) 2024/1689.

11. Ligate Chain. Attestation protocol v0 specification. github.com/ligate-io/ligate-chain/blob/main/docs/protocol/attestation-v0.md