



Elephant Network

Litepaper

The default network for verified property data

May 20, 2026

Contents

1	Executive Summary	3
2	The Problem: There Is No Default Place for Verified Property Data	3
3	The Wedge: Start with Address Data	4
4	The Solution: A Verified Property Data Network	5
5	How the Network Works	5
5.1	Technical Architecture	6
6	Data Model and Verified Evidence	7
7	Network Economics: Mahout and vMahout	8
8	Bootstrapping the Property Data Network	8
9	Ecosystem and Network Effects	9
10	Governance, Trust, and Data Quality	10
11	Roadmap: From Address Data to the Property Graph	10
11.1	Phase 1: Verified Address Layer	11
11.2	Phase 2: Property Identity Graph	11
11.3	Phase 3: High-Value Property Data Categories	11
12	Conclusion: The Default Network for Verified Property Data	12

Executive Summary

Elephant Network is building the default network for verified property data.

The initial wedge is U.S. address data. Addresses look simple, but they are the starting point for almost every property-data workflow: delivery, underwriting, servicing, title, identity verification, field inspection, fraud detection, tax administration, and real estate operations. Today, the same address information is repeatedly validated, standardized, and reconciled across enterprises, creating billions of dollars of duplicated cost and effort.

Elephant changes that model. The network turns fragmented public records into reusable, verified data elements. It starts with addresses, then expands into property identity and other high-value property data categories. Data is contributed, validated, refreshed, accessed, and monetized through a shared network rather than repeatedly reconstructed inside each enterprise.

The network uses two related tokens. **Mahout** is the network payment and settlement token. **vMahout** is a constrained-transfer, data-element contribution-right token that identifies who currently earns from, and governs, fresh verified data elements. Data consumers access the network through an open protocol, with API compatibility available for legacy systems. Enterprise users do not need to experience Elephant as a crypto workflow; they can pay in fiat or stablecoin while the network settles internally in Mahout.

Elephant's goal is not to build another private property data vendor. It is to create the missing coordination layer for verified property data: the place enterprises go to access data they can trust, and the network where active contributors are rewarded for keeping that data accurate, current, and reusable.

The Problem: There Is No Default Place for Verified Property Data

There is no Amazon for verified U.S. property data. There is no default place where enterprises go to access it, and no default network where contributors go to provide, validate, refresh, and monetize it. Elephant is building that place: a coordination layer where property data is **contributed, validated, refreshed, accessed, and monetized**.

Much of the source material is public information, but public records are not public infrastructure. County records, address records, assessor data, parcel data, and permit records may be available at the source, but they are fragmented across thousands of jurisdictions and systems. A record may be public, yet still difficult to use because it is not standardized, source-linked, continuously refreshed, or made available through a shared access layer.

Commercial data vendors stepped into that gap. They aggregate public records and resell them, often with opaque provenance, uneven freshness, and high costs for enterprise users. Elephant's aim is different: to build a public coordination layer that makes public property information usable as shared infrastructure.

Elephant is not charging for the fact that a public record exists. The network creates value by

collecting, standardizing, validating, refreshing, indexing, and serving source-linked data at scale, while rewarding the contributors and validators who keep the network accurate and fresh.

For the initial wedge—address data—the immediate problem is repeated validation. Logistics companies, online retailers, insurers, mortgage servicers, real estate operators, identity-verification providers, and field-service businesses all spend time and money validating, cleaning, and reconciling address data. Much of that work is duplicated across organizations. The address-validation burden likely represents billions of dollars in annual spend.

The problem is not that address data cannot be found. The problem is that there is no shared network where verified address data becomes reusable infrastructure.

The Wedge: Start with Address Data

Elephant begins with U.S. addresses because every property-data workflow starts with identity: what place is this, which parcel or structure does it refer to, and which source proves it?

The initial wedge is not a simple postal-address validation product. It is a canonical address layer that reconciles multiple address types—including mailing, property, parcel-linked, building / unit, emergency / 911, service, jurisdictional, and geocoded addresses—into verified, source-linked address objects.

In Elephant, an address object is not a single flat record. It is a small virtual graph of related elements. These may include MSA, FIPS code, county, location / shape, building placement, and other address-linked attributes that help connect a standardized address to government datasets, county records, property identity, and later property data categories.

USPS is the authoritative source for standardized, mail-deliverable addresses. But USPS is not the only official source relevant to property identity. Records from more than 3,100 counties—including assessor records, parcel data, local addressing authority records, GIS files, tax records, and emergency addressing systems—all matter because a mailing address is not always the same as a situs address, parcel-linked address, service address, or geocoded property location. More than 500 MLS systems add another non-public layer of property-location data.

Elephant's first sources are **USPS and county records**. The network's first task is to normalize and validate address data against those sources, connect it to available property identity evidence, and make the result accessible through the network.

This wedge is commercially meaningful because address validation is repeated constantly across high-volume workflows. Elephant can reduce adoption friction by letting applications replace existing address-validation API calls with an Elephant Network call returning equivalent address information at an estimated cost reduction of approximately **80%**.

Elephant has already mined and structured **10.5 million property identity records in Florida**, demonstrating the practical foundation for this approach. Addresses are the first layer of property identity.

The Solution: A Verified Property Data Network

Elephant is an indexed, secure, immutable, source-linked **verified property data network**. It starts by turning fragmented public records into reusable verified property data, beginning with addresses and expanding across the property graph. Over time, the network becomes more powerful because anyone can contribute property data, validators can check it, users can consume it, and the history of contributions reveals which participants are reliable.

Elephant is both a data access layer and a coordination network. Data consumers come to access verified property data. Contributors come to provide and refresh data. Miners collect or submit data. Validators confirm correctness, freshness, and source match. The network's purpose is to make property data reusable across the ecosystem rather than repeatedly reconstructed inside each enterprise.

The compounding logic is central. More contributed data creates more coverage. More usage creates more incentives. More validators and users create more scrutiny. Contributor history creates a reliability signal through validation outcomes and dispute history. As the network matures, the property graph becomes not just larger, but more trustworthy.

This is why Elephant should not be understood as another property data vendor. A vendor sells a dataset. Elephant coordinates a market for verified data elements that can be contributed, validated, refreshed, accessed, and monetized by many participants over time.

How the Network Works

Elephant operates as a network for collecting, standardizing, validating, refreshing, and serving property data. A data element begins as a source-linked submitted value, is mapped into Elephant's property model, validated against accepted evidence, and becomes available to data consumers with its source, timestamp, status, and contributor attribution.

At a high level, the network lifecycle works as follows:

- 1. Collect** — contributors and miners gather data from USPS, county records, and later additional sources.
- 2. Standardize** — raw records are mapped into canonical property objects and data elements.
- 3. Validate** — validators use a form of **zero-knowledge proof (zkProof)** to confirm source match, data transformation correctness, freshness, and identity binding.
- 4. Refresh** — county data elements must be refreshed at a required heartbeat, initially weekly. A miner's possession of the vMahout corresponding to a data element depends on maintaining that heartbeat. If the heartbeat is not maintained, another miner can take over and provide that data.

5. **Resolve conflicts** — when two miners submit competing fresh versions of the same data element within the heartbeat window, the current vMahout holder is given precedence, subject to validation and dispute rules.
6. **Serve** — data consumers access verified data through the open protocol, with API compatibility, bulk delivery, and property-level lookup available where needed.
7. **Reward** — Mahout and vMahout connect usage, contribution, freshness, and governance.

Participant terminology is important. **Contributors** is the broad term for parties that add or refresh data, including scale contributors and individual contributors. **Miners** are the subset of contributors who collect, submit, or refresh data. **Validators** validate correctness, freshness, source match, transformation, and identity binding. **Data consumers** access and use verified data from the network.

Technical Architecture

Elephant uses an agentic mining layer—software agents that collect and refresh source data—along with a validation layer, persistence layer, property graph, and access layer. Miners collect data from USPS, county records, and later additional sources, using source-specific mappings to convert fragmented records into Elephant’s data model. County data elements are refreshed on a required heartbeat, initially weekly.

Validators use zkProofs to validate source match, data transformation correctness, freshness, and identity binding. This allows the network to verify not only that a submitted value came from the claimed source, but that it was transformed correctly, refreshed within the required heartbeat, and bound to the correct property identity.

Elephant does not store every raw data payload directly on-chain. For each minted property, the network stores the root hash—the uppermost hash—of a Merkle tree on-chain. The underlying tree corresponds to the data elements related to that property, while the supporting records, evidence, and payloads are stored off-chain in IPFS. The root hash becomes the property’s unique on-chain identifier and anchors the off-chain evidence to a tamper-evident network record.

Verified data elements are organized into the property graph and made available through the open protocol, with API compatibility, bulk delivery, and property-level lookup available where needed. For the address wedge, a simple code snippet allows applications to replace existing address-validation API calls with an Elephant Network call.

Blockchain is the system of record for provenance, rights, settlement, and governance. The core function is to make the network’s claims, rights, and economic coordination inspectable and tamper-evident.

Additional protocol specifications will define the precise on-chain event model, dispute procedures, and staking / slashing mechanics.

Data Model and Verified Evidence

Elephant’s data model distinguishes between **current verified state** and **historical submitted values**. A **verified data element** is not only a value; it is a source-linked claim with metadata showing where it came from, when it was collected, who contributed it, how it was validated, and what its current network status is.

The network’s value comes from making verified data elements reusable, inspectable, and economically coordinated.

Core objects in the initial data model include:

- **Property** — the real-world asset or location being described.
- **Address** — the canonical address object connected to mailing, property, parcel, building, service, emergency, jurisdictional, and geocoded evidence.
- **Parcel** — the land parcel, tax parcel, or legal parcel associated with the property, where available in county data.
- **Building / Structure** — the physical structure associated with the property or parcel, where available in county data.
- **Source** — the origin of a data element, such as USPS or county records.
- **Submitted Value** — a value collected from a source and submitted to the network before validation.
- **Verified Data Element** — a specific source-linked property data value that has passed network validation.
- **Contributor / Miner** — the party that submitted or refreshed the data.
- **Validation Record** — the proof or validation result attached to the data.
- **vMahout Record** — the contribution-right associated with the current fresh data element.

A submitted value becomes a verified data element only after it passes validation. Each verified data element carries metadata for source, timestamp, validation, refresh status, contributor attribution, vMahout holder, dispute status, and zkProof reference.

Network data states identify where a submitted value or verified data element sits in the lifecycle: pending validation, verified, stale, disputed, superseded, not found, or in conflict.

Elephant does not merely store “the answer.” It preserves the relationship between the value, source, contributor, validation, economic rights, and status over time.

Network Economics: Mahout and vMahout

Elephant uses two related tokens with distinct roles.

Mahout is the network payment and settlement token. Data consumers pay for access to verified data, and Mahout is used internally to settle network activity and compensate participants.

vMahout is a constrained-transfer, data-element contribution-right token tied to current, fresh, verified data elements. It identifies who currently holds the right to receive gas or fees when a data element is consumed and who has governance weight as an active contributor.

The core distinction is simple:

Mahout pays for data. vMahout proves who currently earns from and governs the data.

Mahout supports access, settlement, rewards, and network usage. vMahout supports freshness, contribution rights, and active governance. For county data, maintaining the required heartbeat is what preserves the miner's vMahout position for the relevant data element. If the heartbeat lapses, another miner can refresh the data and take over the vMahout.

Usage fees flow to the vMahout holders associated with the verified data elements being consumed. These vMahout holders are the miners currently maintaining those fresh data elements. Validators are paid separately by miners at the time of validation request, rather than receiving gas directly from the end consumer of the data.

Enterprise users do not need to experience Elephant as a crypto workflow. Buyers can pay in fiat or stablecoin through familiar access and contracting models, while the network settles internally in Mahout and rewards active contributors through the Mahout / vMahout model. Elephant uses Mahout for staking and slashing, to incentivize data quality.

Bootstrapping the Property Data Network

Elephant's challenge is not only technical. It is social and economic: the network must move from "no one is here" to "everyone involved in property data comes here reflexively."

The path begins with a narrow wedge: verified U.S. address data. Address data is universal, high-volume, and foundational to every other property-data workflow. By solving this narrow problem first, Elephant creates an initial reason for enterprise buyers to use the network before the full property graph exists.

Elephant seeds the first supply using existing mined and structured data and early network operations. The 10.5 million Florida property identity records demonstrate that this is not just a concept. They provide a foundation for the first layer of verified property identity.

Elephant reduces adoption friction through two paths. First, it targets legacy high-volume users that already pay for address validation and property data. Second, it targets new commerce and

application development, where Elephant can become the default address layer through simple code snippets, SDKs, and integrations with AI coding assistants or AI coding agents.

Many applications already call external address-validation services. For legacy adoption, a simple code snippet can replace existing address-validation API calls with an Elephant Network call. This creates a low-friction bridge from today's API-based market to Elephant's protocol-native data network.

The initial economic case is direct: Elephant expects to provide address data at a significantly lower cost than existing suppliers, with an estimated cost reduction of approximately **80%**. This cost advantage creates the first reason for users to switch, while the broader network creates the reason to stay and expand usage.

The Amazon analogy is useful here, but only as a market-formation pattern. Amazon first became the default place to buy books, then became the default place for sellers to reach buyers. Elephant follows a similar network logic for property data. First, it becomes the place enterprises go to access verified data. Then, as usage creates rewards, it becomes the place contributors, miners, and validators go to provide, validate, refresh, and monetize verified data.

In the first phase, the adoption loop is simple: lower-cost address access attracts data consumers; real usage creates rewards; rewards attract contributors and validators; and better coverage and freshness make the address layer more useful. Early adoption should be measured by real usage: number of unique callers and number of addresses processed and paid through the network.

Ecosystem and Network Effects

Elephant's long-term value comes from more than lower-cost address validation. The network creates a compounding ecosystem around verified property data.

First, Elephant is designed for a world where AI agents can shop for data. As AI agents begin to perform research, underwriting, valuation, compliance, logistics, and application workflows, they will need reliable sources of structured, verified data. Elephant gives those agents a protocol-level way to discover, access, and pay for property data they can trust.

Second, Elephant creates an application platform for builders. Once verified property data becomes open, source-linked, and accessible through a shared protocol, developers can build new property applications without recreating the underlying data layer. This opens the door to new property search, listing, underwriting, servicing, local intelligence, field-service, and analytics applications that depend on property identity and property context.

Third, Elephant creates a new path for existing data suppliers. Many suppliers already have customers but spend substantial resources gathering, cleaning, and refreshing data. Elephant allows those suppliers to shift from maintaining duplicative collection pipelines to consuming verified network data, while still serving their customers through existing relationships, interfaces, and specialized products.

The access model matters. Elephant is not designed to make APIs the primary long-term interface.

The mainstream access layer is an open protocol for Web3-native data consumers. APIs can be bolted on for legacy users that need compatibility with existing systems. This allows Elephant to serve today's enterprises while building toward a protocol-native market where data consumers, contributors, validators, developers, and AI agents coordinate directly.

As the network matures, the effects extend beyond address validation. Verified property data becomes a shared substrate for applications, AI agents, data suppliers, and enterprise workflows. Builders create new products on top of the property graph, existing suppliers reduce duplicative data collection, and AI agents gain a reliable market for structured property data. Over time, Elephant becomes not just a source of property data, but the coordination layer for a property-data ecosystem.

Governance, Trust, and Data Quality

Elephant's trust model relies on source-linked evidence, zkProof validation, heartbeat refresh, dispute history, contributor accountability, staking / slashing, and active-contributor governance.

Every verified data element carries metadata showing where it came from, when it was collected, who contributed it, how it was validated, and what its current status is. The goal is to make property data inspectable rather than opaque.

Governance is designed to align authority with current useful contribution. vMahout gives active contributors governance weight because it represents current contribution rights tied to fresh, verified data elements. This is different from governance based only on passive token ownership. Contributor reliability develops over time through validation outcomes and dispute history.

In early governance, the Elephant founding team acts as the initial governance steward while the network matures. Over time, governance is intended to shift toward active contributors whose participation is reflected through vMahout.

Data quality is maintained through required refresh heartbeats, zkProof-based validation, dispute mechanisms, and the ability for another miner to replace a stale contributor when the heartbeat is not maintained. Mahout-based staking and slashing provide additional economic alignment.

This creates a dynamic system: the network not only stores data, but continuously tests and improves the reliability of the data layer.

Additional governance specifications will define voting scope, dispute procedures, and the transition from founding-team stewardship to active-contributor governance.

Roadmap: From Address Data to the Property Graph

Elephant's roadmap begins with verified U.S. address data and expands into a broader property graph. The first phase establishes the canonical address layer and the access mechanism that lets users replace existing address-validation API calls. Later phases extend the network from address

identity into property identity and then into high-value property data categories.

The roadmap is sequence-based, not date-based.

Phase 1: Verified Address Layer

- USPS and county-record sources
- canonical address objects
- replacement code snippet for existing address-validation API calls
- initial enterprise usage
- unique callers and paid addresses processed as key usage metrics

Phase 2: Property Identity Graph

- parcel association
- building / structure association, where available
- source-linked property identity records
- expansion from address validation to property identity

Phase 3: High-Value Property Data Categories

Examples include:

- permits
- ownership
- liens
- roof / structure details
- occupancy
- other property data categories added as network coverage and demand expand

This roadmap stays focused on the property opportunity. The property graph alone is large enough to justify a disciplined build sequence.

Conclusion: The Default Network for Verified Property Data

Elephant starts with addresses because every property-data workflow begins with identity. From there, the network expands into property identity and high-value property data categories.

The ambition is larger within property itself: to become the default network for verified property data. The place enterprises go to access data they can trust, and the place contributors go to provide, validate, refresh, and monetize it.

Property is one of the world's most important asset classes, yet its data infrastructure remains fragmented. Public records exist, but they are not coordinated as public infrastructure. Private vendors aggregate and resell slices of the data, but the market still lacks a shared network where verified property data becomes reusable, inspectable, and economically sustainable.

Elephant is building the shared layer that lets this market coordinate: the **default network for verified property data**.