Nodalync: A Protocol for Fair Knowledge Economics

Gabriel Giangi
gabegiangi@gmail.com

Abstract. A purely protocol-based system for knowledge economics would allow original contributors to receive perpetual compensation without requiring continuous production. We propose a solution using a five-layer knowledge model where source material (L0) forms an immutable foundation from which all derivative value flows. Cryptographic provenance chains link every insight back to its roots. A revenue distribution mechanism ensures that 95% of all transaction value flows to L0 contributors proportionally, regardless of derivation depth. The protocol creates a knowledge layer between humans and AI systems, where agents query personal knowledge bases through a standard interface while automatically compensating all upstream contributors. L0 owners benefit from all downstream activity without needing to participate in synthesis. The result is infrastructure where contributing valuable foundational knowledge once creates a perpetual economic stake in all derivative work.

1. Introduction

The digital economy has systematically failed knowledge creators. Researchers publish findings that become foundational to entire industries, receiving citations but not compensation. Writers produce content that trains AI models worth billions, with no mechanism for attribution or payment. The problem is architectural: existing systems cannot track how knowledge compounds through chains of derivation.

Current approaches to knowledge monetization require continuous production. Creators must constantly generate new content to maintain income. This model favors aggregators who consolidate others' work over original contributors who establish foundations. When insight A enables insight B which enables insight C, creator A receives nothing from C's value despite providing the foundation.

We propose a protocol that inverts this dynamic. By structuring knowledge into layers with cryptographic provenance, we ensure that value flows backward through derivation chains to original contributors. L0 owners—those who contribute foundational source material—receive proportional compensation from all downstream transactions automatically. A researcher can publish valuable findings once and receive perpetual royalties as the ecosystem builds upon their work, without needing to continuously create.

The protocol serves as a knowledge layer between humans and AI. Any agent can query personal knowledge bases through a standard interface (MCP), with every query triggering automatic compensation to all contributors in the provenance chain. This creates infrastructure for a fair knowledge economy where foundational contributions are valued in perpetuity.

2. Knowledge Layers

The protocol structures all knowledge into five distinct layers with specific properties:

Layer	Name	Contents	Properties
L0	Raw Inputs	Documents, transcripts, notes	Immutable, publishable, purchasable
L1	Mentions	Atomic facts with L0 pointers	Extracted, publishable, purchasable
L2	Entities	Synthesized profiles	Internal only, never sold
L3	Relations	Connections between entities	Internal only, never sold
L4	Insights	Emergent patterns and conclusions	Sellable, importable as L0

L0 and L1 constitute the source layers. These represent original contributions that can be published and purchased. Buyers receive query access to source material—the data itself never leaves the owner's node. This preserves sovereignty while enabling monetization.

L2 and L3 are synthesis layers used for internal organization. They represent structured reorganization of source material rather than new creation, and are never sold. This prevents value extraction through mere reorganization.

L4 represents genuinely emergent insights—conclusions abstract enough to constitute new intellectual property. L4 can be sold. Critically, purchased L4 can be imported as L0 by buyers, enabling knowledge to compound across ownership boundaries while preserving attribution chains.

3. Provenance

Every node in the system stores its complete derivation history through content-addressed hashing. When content is created or modified, a hash is computed over its contents. This hash serves as a unique identifier enabling trustless verification.

Each node maintains:

```
input_hashes[]: hashes of all content contributing to this node
derived_from[]: edges to source nodes enabling graph traversal
root_L0L1[]: flattened array of all ultimate L0+L1 sources
timestamp: creation time for staleness detection
```

The root_L0L1 array is the key structure for revenue distribution. Regardless of how many intermediate derivation steps occur, every node maintains a direct reference to all foundational sources. An L4 insight derived from another L4 (imported as L0) inherits the original L4's root_L0L1 array, extending rather than replacing the provenance chain.

This creates cryptographic proof of contribution. If Alice's L0 document hash appears in Bob's L4's root_L0L1 array, Alice's contribution is provable without requiring social trust or centralized verification. The provenance is in the data structure itself.

4. Transactions

The protocol supports two transaction types: query access and L4 purchase.

4.1 Query Access

Buyers purchase query rights to L0+L1 content. The source material never leaves the owner's node. When a query is executed, results return with provenance hashes enabling attribution tracking. This model preserves data sovereignty while enabling monetization.

Query access is perpetual. A one-time purchase grants unlimited future queries, including access to updates the owner makes to their L0 content. This aligns incentives: owners benefit from maintaining and improving their sources.

4.2 L4 Purchase and Import

L4 insights can be purchased and imported as L0 by buyers. This enables knowledge compounding across ownership boundaries. The imported L4 becomes foundational source material for the buyer's own derivation chains.

When L4 is imported as L0, the full provenance chain inherits forward:

```
buyer.new_L0.root_L0L1 = seller.L4.root_L0L1 ∪ {seller.L4.hash}
```

The original L4 creator joins the root contributor set. All upstream L0 sources remain tracked. Any L4 the buyer creates using this imported knowledge will distribute revenue to all contributors in the chain, including the original seller and everyone they built upon.

5. Revenue Distribution

When a transaction generates revenue, distribution follows a fixed mechanism designed to ensure value flows to foundational contributors:

5.1 Distribution Formula

For a transaction of value V with root contributor set R:

```
seller_share = 0.05 × V
root_pool = 0.95 × V
per_root_share = root_pool / |R|
```

The seller retains 5% as synthesis incentive. The remaining 95% splits equally among all L0+L1 roots in the provenance chain. All roots are weighted equally (1:1) regardless of content type or derivation distance.

5.2 Example

Bob creates an L4 insight using 2 of Alice's L0 documents, 1 of Carol's L0 documents, and 2 of his own L0 documents. Bob sells for 100 tokens:

```
Bob (seller): 5 + (2/5 × 95) = 43 tokens
Alice (2 roots): 2/5 × 95 = 38 tokens
Carol (1 root): 1/5 × 95 = 19 tokens
```

Bob receives 43% despite creating the L4, because the protocol values foundational contributions. If Bob had used entirely others' sources, he would receive only the 5% floor.

5.3 Compounding Returns

The mechanism creates exponential potential for foundational contributors. Consider Alice's L0 document:

```
Direct sales: 10 buyers × per-sale royalty
Second-order: 10 buyers each create L4s, each sold 10× = 100 indirect
royalties
Third-order: 100 L4s each enable 10 more = 1000 indirect royalties
```

Alice's single L0 contribution earns from all downstream generations. She need not create L4s herself to benefit from the ecosystem building on her work. Contributing valuable foundational knowledge once creates perpetual economic participation.

6. Agent Interface

The protocol implements Model Context Protocol (MCP) as the standard interface for AI agent consumption. Any MCP-compatible system can query personal knowledge bases while automatically triggering compensation.

6.1 Query Mechanism

Agents submit natural language queries to knowledge nodes. The node processes the query against its L0-L4 layers and returns structured responses with provenance metadata:

```
response.content: answer to query
response.sources[]: L0+L1 hashes accessed
response.confidence: derivation certainty
response.cost: tokens owed for this query
```

The response includes everything needed for the agent (or its operator) to verify sources and process payment. Provenance is embedded in the response, not stored externally.

6.2 Access Control

Node owners define permissions at layer, entity, or document granularity. Permissions specify which agents can access which content and at what price. Access can be revoked at any time; subsequent queries are denied.

6.3 Audit Trail

Every query is logged locally:

```
log.timestamp: when accessed
log.agent_id: who accessed
log.query: what was asked
log.sources_accessed[]: L0+L1 used in response
log.revenue_distributed{}: payment breakdown
```

The audit trail provides complete transparency into how knowledge is consumed by AI systems—transparency impossible with current web scraping approaches.

7. Privacy

The protocol is local-first. All data remains on the owner's node. No centralized storage, no uploads to external platforms. Buyers and agents receive query access, not data downloads.

This inverts the current data paradigm. Instead of users uploading data to platforms that monetize it, agents come to users. The protocol enforces data sovereignty architecturally rather than legally—your data literally cannot be accessed without your node's participation.

Provenance hashes are public and enable verification, but the underlying content they reference remains private until query access is granted. This separates attribution (public) from content (private).

8. Network

Nodes operate independently, storing their own knowledge graphs and serving their own queries. Discovery occurs through a decentralized index where nodes publish metadata about available L0+L1 content and L4 products without revealing the content itself.

Settlement uses smart contracts for access rights and payments. When a buyer purchases query access, the contract records the grant. When queries execute and generate revenue, the contract distributes payments according to the provenance chain. Minimal data goes on-chain: access grants, payment flows, and attestations. Content and queries remain off-chain.

This hybrid architecture—off-chain content, on-chain economics—preserves privacy while enabling trustless compensation.

9. Conclusion

The Nodalync protocol creates infrastructure for fair knowledge economics. By structuring knowledge into layers with cryptographic provenance and automatic revenue distribution, the protocol ensures that foundational contributors receive perpetual, proportional compensation from all downstream value creation.

L0 owners are the foundation of this economy. A researcher, writer, or domain expert can contribute valuable source material once and benefit as the ecosystem builds upon their work. They need not continuously produce, need not create sophisticated L4 insights, need not compete with aggregators. The protocol routes value backward through derivation chains automatically.

For AI systems, the protocol provides a standard interface for consuming human knowledge while respecting attribution and compensation. Every query triggers payment to all contributors in the provenance chain. This creates sustainable infrastructure for AI-human knowledge exchange—not extraction, but transaction.

The protocol is local-first, preserving individual sovereignty. It is AI-native, designed for agentic consumption. It is cryptographically verifiable, requiring no social trust. And it is fair by design, with mechanisms that resist value capture by intermediaries.

We propose this as the knowledge layer between humans and AI: infrastructure where contributing valuable knowledge creates perpetual economic participation in all derivative work.

. References

[1] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

[2] Anthropic. (2024). Model Context Protocol Specification.

[3] Benet, J. (2014). IPFS - Content Addressed, Versioned, P2P File System.