

# Nodalync: A Protocol for Fair Knowledge Economics

Gabriel Giangi  
gabegiangi@gmail.com

## Abstract

We propose a protocol for knowledge economics that ensures original contributors receive perpetual, proportional compensation from all downstream value creation. A researcher can publish valuable findings once and receive perpetual royalties as the ecosystem builds upon their work. A writer's insights compound in value as others synthesize and extend them. The protocol enables humans to benefit from knowledge compounding—earning from what they know, not just what they continuously produce. The protocol structures knowledge into four layers 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 routes 95% of transaction value to foundational contributors regardless of derivation depth. Unlike prior approaches that attempted to price and transfer data, the protocol monetizes query access—buyers gain the right to query a node, not ownership of transferable assets. This eliminates secondary markets that have historically enabled royalty bypass. The protocol implements Model Context Protocol (MCP) as the standard interface for AI agent consumption, creating immediate demand from agentic systems. The result is infrastructure where contributing valuable foundational knowledge once creates perpetual economic participation 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, and even when they can, enforcement mechanisms collapse under market pressure.

Current approaches 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. The result is a knowledge economy where humans must work perpetually, never able to benefit from the compounding value of their past contributions.

We propose a protocol that inverts this dynamic. By structuring knowledge into layers with cryptographic provenance and monetizing query access rather than data ownership, we ensure value flows backward through derivation chains to original contributors. Foundational contributors—those who provide 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. A domain expert's knowledge compounds in value as others synthesize and extend it. The protocol enables humans to earn from what they know, not just what they continuously produce—creating a path toward economic participation that does not require perpetual labor.

The protocol serves as a knowledge layer between humans and AI. Any agent can query personal knowledge bases through the Model Context Protocol (MCP), with every query triggering automatic compensation to all contributors in the provenance chain. This creates infrastructure for a fair knowledge economy—one that bridges the historical gap between research and commerce, enabling foundational contributors to participate economically in all derivative value their work enables.

## 2. Prior Work

The components of this protocol draw from established systems. Content-addressed storage, pioneered by Git and formalized by IPFS, provides cryptographic integrity guarantees through hash-based identification. Merkle trees enable efficient verification with logarithmic proof sizes. The Model Context Protocol, released by Anthropic and now stewarded by the Linux Foundation, provides a standard interface for AI systems to consume external resources.

Prior attempts at data marketplaces—Ocean Protocol, Streamr, Azure Data Marketplace—failed primarily on the pricing problem: data value varies dramatically by context, and sellers consistently could not determine appropriate prices. NFT royalty systems failed differently: royalties were never enforced on-chain but relied on marketplace cooperation, which collapsed under competitive pressure when platforms began offering zero-royalty trading to attract volume.

Academic citation systems demonstrate that attribution without compensation creates no economic incentive for foundational contribution. Publishers capture margins while authors receive prestige as a substitute for payment. This protocol proposes that attribution and compensation must be unified—provenance chains that simultaneously prove contribution and trigger payment.

Our contribution is not novel components but their integration into a coherent system that solves the enforcement problem through architectural choice: monetizing query access rather than transferable ownership eliminates the secondary markets where prior royalty systems failed.

## 3. Knowledge Layers

The protocol structures all knowledge into four 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	Entity Graph	Entities + RDF relations	Internal only, never sold
L3	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. Critically, buyers receive query access to source material—the data itself never leaves the owner's node. This preserves sovereignty while enabling monetization.

L2 is the synthesis layer used for internal organization. It represents entities and the RDF relations between them (subject-predicate-object triples), enabling structured queries across source material. L2 is never sold because it represents reorganization rather than new creation— preventing value extraction through mere restructuring.

L3 represents genuinely emergent insights—conclusions abstract enough to constitute new intellectual property. L3 can be sold. Purchased L3 can be imported as L0 by buyers, enabling knowledge to compound across ownership boundaries while preserving attribution chains. When Alice purchases Bob's L3 insight and imports it as her own L0, all of Bob's upstream provenance transfers forward—Alice's subsequent work inherits the full chain of foundational contributors.

## 4. 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—identical content produces identical hashes regardless of where or when it is created.

L0 is immutable once published. Updates are published as new L0 nodes, enabling versioned provenance. Provenance chains reference specific versions, preserving the historical record of what actually contributed to what. This immutability is essential: if sources could change after derivation, the integrity of downstream work would be compromised.

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 ordering and staleness detection

The root\_L0L1 array is the key structure for revenue distribution. Regardless of how many intermediate derivation steps occur (L2 synthesis, L3 insight generation), every node maintains direct reference to all foundational sources. An L3 insight derived from another L3 (imported as L0) inherits the original L3'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 L3's root\_L0L1 array, Alice's contribution is provable without requiring social trust or centralized verification. The provenance is in the data structure itself.

## 5. Transactions

The protocol supports two transaction types: query access and L3 purchase. Both operate on the principle that buyers gain access rights, not ownership of transferable assets.

### 5.1 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 but non-transferable. 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. Critically, because query rights cannot be resold, there is no secondary market where buyers could bypass the original node. Every query must route through the owner's node, triggering the revenue distribution mechanism.

This architectural choice solves the enforcement problem that defeated NFT royalties. NFT royalties failed because they were enforced by marketplaces, not by the protocol—when marketplaces competed by offering zero-royalty trading, creators lost. In this protocol, there is no marketplace to bypass. The node owner controls every interaction with their content.

## 5.2 L3 Purchase and Import

L3 insights can be purchased and imported as L0 by buyers. This enables knowledge compounding across ownership boundaries. The imported L3 becomes foundational source material for the buyer's own derivation chains.

L3 purchase grants the insight only, not automatic access to underlying sources. Buyers receive the L3 (importable as their L0), and foundational contributors receive their 95% share via provenance. If the buyer wants to query the underlying L0/L1 sources directly, they purchase that access separately. This creates clean separation: you are buying synthesis, not sources.

When L3 is imported as L0, the full provenance chain inherits forward:

```
buyer.new_L0.root_L0L1 = seller.L3.root_L0L1 ∪ {seller.L3.hash}
```

The original L3 creator joins the root contributor set. All upstream L0 sources remain tracked. Any L3 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.

## 6. Revenue Distribution

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

### 6.1 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 regardless of content type or derivation

distance.

## 6.2 Rationale for 95/5

This distribution inverts typical platform economics, where intermediaries capture 10-45% of value. The inversion is intentional: foundational knowledge is systematically undervalued in current markets. Researchers, domain experts, and original thinkers provide the substrate on which all synthesis depends, yet receive nothing from downstream value creation. The 95% allocation to foundational contributors corrects this market failure.

The 5% synthesis fee may appear to disincentivize synthesis, but this concern misunderstands the mechanism. Consider a concrete example: Bob creates an L3 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 + 2 roots):  $5 + (2/5 \times 95) = 43$  tokens

Alice (2 roots):  $2/5 \times 95 = 38$  tokens

Carol (1 root):  $1/5 \times 95 = 19$  tokens

Bob receives 43% of the transaction despite the 5% synthesis fee because he also contributed foundational material. The protocol incentivizes synthesizers to also be contributors. A pure synthesizer using entirely others' sources receives only the 5% floor—this is by design. The incentive structure rewards those who contribute original knowledge, not those who merely reorganize others' work.

The 5% synthesis fee is not the endgame for valuable synthesis. If an L3 is foundational enough that others build upon it (import as their L0), the original synthesizer becomes part of their root\_L0L1[] arrays. The protocol incentivizes creating insights worth building on, not just worth buying. First-order sale earns 5%; becoming foundational for others' work is where compounding happens. A synthesizer who creates genuinely valuable L3 insights will see those insights imported and built upon, earning from all downstream generations.

## 6.3 Compounding Returns

The mechanism creates exponential potential for foundational contributors. Consider Alice's L0 document over three generations of derivation:

Direct sales: 10 buyers  $\times$  per-sale royalty

Second-order: 10 buyers each create L3s, each sold 10 $\times$  = 100 royalties

Third-order: 100 L3s each enable 10 more = 1,000 royalties

Alice's single L0 contribution earns from all downstream generations. She need not create L3s herself to benefit from the ecosystem building on her work. Contributing valuable foundational knowledge once creates perpetual economic participation—enabling earlier exit from continuous production while maintaining income as others build on one's contributions.

## 6.4 Fairness Priorities

Fairness priorities are embedded in protocol design at three levels:

**Fair distribution** (highest priority): The 95/5 split inverts typical platform economics. Equal root weighting distributes value across all foundational contributors. The more sources an L3 builds upon, the more widely value distributes—rewarding comprehensive synthesis that draws from diverse foundations.

**Fair contribution:** No gatekeeping on L0 publication. No credentials required. No institutional approval necessary. The market determines value, not committees. Anyone can contribute foundational knowledge; quality is determined by whether others choose to build upon it.

**Fair access:** Access enables contribution. The protocol supports tiered pricing (commercial/academic/individual), a commons layer for explicitly open contributions, and contributor credits for those who publish L0. These mechanisms ensure that the protocol does not create a knowledge economy accessible only to the wealthy.

## 7. Agent Interface

The protocol implements Model Context Protocol (MCP) as the standard interface for AI agent consumption. MCP, originally developed by Anthropic and now stewarded by the Linux Foundation's Agentic AI Foundation, provides a standardized way for AI systems to access external resources. Any MCP-compatible system can query personal knowledge bases while automatically triggering compensation.

### 7.1 Query Mechanism

Agents submit natural language queries to knowledge nodes. The node processes the query against its L0-L3 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. This creates immediate demand from agentic AI systems—a consumer base that did not exist for prior data marketplace attempts.

### 7.2 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.

### 7.3 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.

## 8. 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 paradigm where users upload data to platforms that monetize it—instead, agents come to users.

Contributors choose their privacy level per contribution. The protocol supports both named and pseudonymous contributions. For contributors requiring stronger privacy guarantees, zero-knowledge proofs can demonstrate membership in a verified contributor set without revealing specific identity—proving "this hash was created by someone in the verified contributor pool" without revealing who.

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).

## 9. 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 L3 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.

### 9.1 Governance

The governance model remains under development. Design goals include: decentralization where possible, market-driven decision-making for most parameters, and protections ensuring broad participation rather than plutocracy. Options under consideration include one-node-one-vote, quadratic voting, and contribution-weighted governance. The final model will be determined through community input prior to mainnet launch.

## 10. Threat Model

We identify and address the primary attack vectors against the protocol.

### 10.1 Sybil Attacks

Without identity verification, actors could create multiple pseudonymous identities to claim foundational portions of knowledge, fragmenting legitimate contributions. The protocol is identity-agnostic by design—we do not require identity verification at the base layer.

Instead, economic incentives align behavior. Quality content earns; spam does not. The market determines which L0 sources are valuable through purchasing behavior. A fragmented identity strategy—creating many accounts with thin contributions—produces no advantage because revenue

distributes based on which sources are actually used, not how many sources exist.

Furthermore, reputation accrues to consistent identity. A single account with many high-quality L0 contributions becomes discoverable and trusted. Fragmenting across pseudonyms sacrifices this reputation benefit. Optional reputation layers can build on the base protocol for contexts requiring stronger identity guarantees.

## 10.2 Attribution Gaming

Actors might attempt to insert themselves into provenance chains through trivial first contributions or synthetic citation chains between controlled addresses. The protocol does not prevent this at the technical layer—but economic incentives make it unprofitable.

Revenue distributes only when content is actually queried. Creating thousands of unused L0 documents generates no income. The market determines value through actual consumption. Synthetic chains between controlled addresses simply redistribute funds within the attacker's own wallet.

## 10.3 Content Copying

Once query access is granted, a buyer could theoretically copy content and republish. This is a limitation of any system that provides access to information. However, several factors mitigate this risk: republished content lacks provenance linkage to the original, so subsequent derivations from the copy do not benefit from the original's reputation; buyers have perpetual access to updates from the original source, which a static copy cannot provide; and the audit trail documents the original access, providing evidence for legal recourse if needed.

## 10.4 Disputes

The protocol does not adjudicate disputes—it provides evidence. Provenance chains are cryptographic fact: a hash is either in root\_L0L1[] or it is not. For suspected plagiarism or parallel discovery:

Embedding similarity detection can flag potential copies at the application layer. Audit trails document access patterns, showing who accessed what and when. Two independent derivation chains arriving at similar insights is valuable data, not necessarily a conflict—it may indicate robust conclusions. Legal systems handle disputes; the protocol provides complete evidence for those systems to adjudicate.

## 10.5 External Plagiarism

The protocol cannot prevent unauthorized publication of others' work at the entry point. Someone could copy-paste an externally published paper as their own L0. However, the protocol makes such theft transparent and traceable:

Timestamps record when L0 was published in-system. Earnings are fully visible and auditable. Evidence for legal recourse is built-in, not forensic. Contributors are encouraged to establish external prior art (journal publication, arXiv, timestamps) as authoritative record of original creation. Alternatively, the protocol itself can serve as a proof-of-creation layer—publish to Nodalync first as timestamped record, then pursue traditional publication.

## 11. Limitations

The protocol does not solve all problems in knowledge economics. We acknowledge the following limitations.

**Pricing discovery.** The protocol does not determine what knowledge should cost. Sellers set prices; the market accepts or rejects them. This may result in inefficient pricing, particularly in early stages before market norms emerge. However, unlike prior data marketplaces that failed attempting to solve pricing algorithmically, we treat price discovery as a market function rather than a protocol function.

**Cold start.** The protocol's value increases with participation. Early adopters face a network with limited content and few buyers. We expect adoption to begin in specific domains where knowledge value is clear (research, technical documentation, domain expertise) before expanding to broader use cases.

**Regulatory uncertainty.** Immutable provenance chains may conflict with data protection regulations requiring deletion rights. Implementations must consider jurisdictional requirements. The separation of content (deletable at the node) from provenance hashes (persistent) provides partial mitigation, but legal analysis is required for specific deployments.

**Not all knowledge should be monetized.** The protocol creates an option for compensation, not a mandate. Commons-based knowledge sharing remains valuable and should continue. The protocol complements rather than replaces open knowledge systems—it provides a path for those who wish to receive compensation without requiring everyone to participate in economic exchange.

## 12. Conclusion

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

Foundational contributors are the substrate 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 L3 insights, need not compete with aggregators. The protocol routes value backward through derivation chains automatically—creating a path to economic participation that does not require perpetual labor.

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 without attribution, but transaction with fair compensation.

The alternative to this protocol is not the knowledge commons—it is the current reality where AI systems train on human knowledge with no mechanism for attribution or payment. The protocol offers a third path: knowledge that flows freely through derivation chains while ensuring that those who contribute to that flow receive proportional benefit.

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.
- [4] Merkle, R. (1988). A Digital Signature Based on a Conventional Encryption Function.
- [5] Douceur, J. (2002). The Sybil Attack. IPTPS.
- [6] World Wide Web Consortium. (2014). RDF 1.1 Concepts and Abstract Syntax.