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 pay-per-query model routes 95% of each transaction to foundational contributors regardless of derivation depth. Users add references to shared nodes freely; payment occurs only when content is actually queried—flowing through the entire provenance chain to compensate everyone who contributed. 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 a pay-per-query transaction model, we ensure value flows backward through derivation chains to original contributors every time knowledge is used. Foundational contributors—those who provide source material—receive proportional compensation automatically with each query. 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 with a pay-per-query model that ensures compensation flows to all contributors every time knowledge is used. There is no upfront purchase to bypass, no secondary market to circumvent—every query to every node triggers payment through the entire provenance chain.

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, queryable
L1	Mentions	Atomic facts with L0 pointers	Extracted, visible as preview
L2	Entity Graph	Entities + RDF relations	Internal only, never shared
L3	Insights	Emergent patterns and conclusions	Shareable, importable as L0

L0 represents raw source material—documents, transcripts, notes, research. L0 is immutable once published; updates are published as new versions (see Section 4.2). When shared, L0 content remains on the owner's node and is accessed only through paid queries.

L1 consists of atomic facts extracted from L0, each maintaining a pointer to its source. L1 serves as a preview layer: when browsing shared content, users see L1 mentions as a summary of what the L0 contains. This enables informed decisions about what to query without requiring payment to evaluate relevance.

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 shared 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 shared and queried like L0. When imported into another user's graph, L3 functions as their L0, enabling knowledge to compound across ownership boundaries while preserving attribution chains.

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.

4.1 Node Structure

Each node maintains:

hash: content-addressed identifier for this version

derived_from[]: hashes of content directly contributing to this node

root_L0L1[]: flattened array of all ultimate L0+L1 sources with weights

timestamp: creation time for ordering and staleness detection

previous_version: hash of prior version (null if original)

version_root: hash of first version in chain (stable identifier)

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 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 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.

4.2 Versioning

L0 is immutable once published. Updates are published as new nodes with new hashes. The previous_version field links to the prior version; the version_root field provides a stable identifier across all versions of the same content.

When Alice updates her L0:

```
new_L0.previous_version = old_L0.hash
```

```
new_L0.version_root = old_L0.version_root (or old_L0.hash if original)
```

Old versions remain accessible. Users who added references to v1 continue using v1; they can add references to v2 separately if desired. Provenance chains reference specific versions, preserving the historical record of what actually contributed to what. This ensures derivations remain valid even as sources evolve.

5. Transactions

The protocol operates on a pay-per-query model. Adding references is free; payment occurs when content is actually queried.

5.1 Reference and Query

Users discover shared content through network indexes that expose metadata: title, L1 mentions (as summary), hash, owner, visibility tier, and version information. This metadata is visible without payment, enabling informed decisions about relevance.

To use content, users add a reference (pointer) to their personal graph. Adding a reference is free—no content is transferred, only a hash is stored locally. The actual content remains on the owner's node.

When the user (or their agent) queries the reference, the protocol triggers a transaction:

1. Query request sent to content owner's node
2. Payment verified via handshake
3. Response delivered to requester
4. Revenue distributed through provenance chain

The query response can be cached and re-read locally without additional payment. The initial query is logged as "viewed," enabling local access to already-received content. Subsequent queries to the same node (for updated information or different query parameters) trigger new payments.

5.2 Derivation

To create an L3 that derives from external sources, the user must have queried (and paid for) each source at least once. This ensures foundational contributors are compensated before their work is incorporated into derivative content.

When L3 is created, the full provenance chain is computed:

```
new_L3.root_L0L1 = union of all source.root_L0L1 arrays
```

Every foundational source that contributed to any input is included. When this L3 is later queried by others, revenue flows to all contributors in the chain.

5.3 L3 Import

When a user queries an L3 and imports it as their own L0, the full provenance chain inherits forward:

```
imported_L0.root_L0L1 = original_L3.root_L0L1 ∪ {original_L3.hash}
```

The original L3 creator joins the root contributor set. All upstream sources remain tracked. Any subsequent L3 created using this imported knowledge will distribute revenue to all contributors in the extended chain.

6. Revenue Distribution

Every query triggers revenue distribution through the entire provenance chain.

6.1 Distribution Formula

For a query generating value V to a node with root contributor set R:

```
owner_share = 0.05 × V
```

```
root_pool = 0.95 × V
```

```
per_root_share = root_pool / |R|
```

The node owner 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. A single query distributes payment to every contributor who helped create that knowledge. When the same source appears multiple times in a provenance chain (through different derivation paths), it receives proportionally more: a source contributing twice receives twice the share.

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. When queried for 100 tokens:

```
Bob (owner + 2 roots): 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 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 querying. First-order queries earn 5%; becoming foundational for others' work is where compounding happens.

6.3 Compounding Returns

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

```
Direct queries: 10 users query Alice's L0
```

```
Second-order: 10 L3s built on Alice's L0, each queried 10× = 100 payments
```

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

Alice's single L0 contribution earns from all downstream queries. 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, provides a standardized way for AI systems to access external resources. Any MCP-compatible system can query knowledge nodes 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[]: hashes of nodes accessed

response.provenance: full derivation chain

response.cost: tokens paid for this query
```

The response includes everything needed for the agent (or its operator) to verify sources and confirm payment. Provenance is embedded in the response, not stored externally. Applications may compute additional metrics (such as confidence scores) from the provenance data based on derivation depth, source accessibility, and domain-specific requirements.

7.2 Payment Handling

The protocol handles the handshake: payment verification triggers response delivery, and revenue distributes through the provenance chain. Application-level concerns—budget controls, cost previews, spending limits, auto-approve settings—are outside protocol scope. Implementations may offer cost estimates before query execution, user-defined budgets for agent sessions, or approval workflows for high-value queries.

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[]: nodes 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 and Visibility

The protocol is local-first. All data remains on the owner's node. No centralized storage, no uploads to external platforms. Queries deliver responses; content itself never transfers permanently. This inverts the current paradigm where users upload data to platforms—instead, agents come to users.

8.1 Visibility Tiers

Content owners choose visibility per node:

Tier	Discoverable	Addable by Others	Queryable
Private	No	No	No (personal use only)
Unlisted	No	Yes (if hash known)	Yes (pay-per-query)
Shared	Yes	Yes	Yes (pay-per-query)

Private nodes exist only for personal use—internal organization, drafts, sensitive material. They cannot be discovered, referenced, or queried by others.

Unlisted nodes are queryable but not discoverable. Owners share hashes directly with specific users or groups. This enables selective sharing: grant access to collaborators without public exposure.

Shared nodes are fully public—discoverable through network indexes, addable by anyone, queryable with standard pay-per-query economics.

8.2 Private Sources in Provenance

A shared L3 may derive from private L0 sources. In this case:

The private source's hash appears in root_L0L1[]—its existence is visible. The private source's content remains inaccessible—others cannot query it. The private source's owner still receives their share of revenue when the L3 is queried. Others see "private source" in provenance—they know it exists but cannot access it.

This enables selective disclosure: publish valuable insights while keeping underlying research private. Consumers trust the synthesis or they don't—provenance shows that sources exist even if content is not verifiable.

8.3 Identity Privacy

Contributors choose their identity level per contribution. The protocol supports named contributions (full identity attached), pseudonymous contributions (wallet address only), and ZK-verified contributions (prove membership in a verified contributor set without revealing specific identity). Provenance hashes are public and enable verification; the identity behind those hashes is configurable.

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 shared content without revealing the content itself.

Settlement uses smart contracts for payment verification and distribution. When a query executes, the contract verifies payment and distributes revenue according to the provenance chain. Minimal data goes on-chain: 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. 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 sources are valuable through query volume. A fragmented identity strategy—creating many accounts with thin contributions—produces no advantage because revenue distributes based on which sources are actually queried, not how many sources exist.

Furthermore, reputation accrues to consistent identity. A single account with many high-quality 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 contributions or synthetic 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 queried. Creating thousands of unused nodes generates no income. The market determines value through actual queries. Synthetic chains between controlled addresses simply redistribute funds within the attacker's own wallet.

10.3 Content Copying

After querying content, a user could theoretically republish it as their own. This is a limitation of any system providing information access. However, several factors mitigate this risk: republished content lacks provenance linkage to the original; the original has earlier timestamps providing evidence of priority; copied content cannot benefit from the original's reputation or query history; and audit trails document the original query, providing evidence for legal recourse.

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 queried 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 external work at the entry point. Someone could publish an externally-created paper as their own L0. However, the protocol makes such theft transparent and traceable:

Timestamps record when content 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 queries should cost. Owners 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 users. 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, implementing pay-per-query transactions, and distributing revenue through complete derivation chains, 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.