

Universal Resource and Fabrication Network (URFN)

**also referred to as: Commons Compute and
Fabrication Architecture (CCFA)**

Working draft — constitutional architecture specification

1. Founding Statement

The system exists to make digital information and computation broadly available without charging users for possession, storage, retrieval, or ordinary virtual transformation. It does so by converting distributed idle storage, computation, and bandwidth into a verified shared resource fabric; by organizing uploaded material through content-sensitive hashing, segmentation, semantic classification, and manifold structure; and by financing the resulting digital commons through the sale of scarce physical realizations.

The architecture treats informational abundance and physical scarcity asymmetrically:

- Digital objects may be copied, analyzed, transformed, and related at negligible marginal user cost.

- Physical objects require material, energy, machinery, labour, safety controls, and delivery, and are therefore sold.
- Revenue from physical realization sustains the infrastructure that keeps virtual participation free.

Uploading an object does not create an entitlement to payment. Compensation is attached only to the *verified provision of infrastructure and services* — storage, computation, transmission, validation, maintenance, fabrication, and fulfillment. This distinction is what prevents "upload garbage for money" as an attack vector; spam and adversarial ingestion still carry real computational cost, but they are not directly rewarded.

Principle of Informational Abundance

Much of the rest of the architecture derives from a single implicit principle, worth naming directly:

Information should become easier to preserve, organize, and access as collective participation increases.

More formally: **informational abundance is the condition in which the marginal cost of providing an additional authorized user access to an object approaches zero, while preserving provenance, integrity, and availability.** This wording matters because it prevents three common confusions: abundance is not piracy (provenance and authorization are preserved), abundance is not infinite storage (it's marginal access cost, not the underlying resource cost, that approaches zero), and abundance is not the absence of governance (authorization still gates access).

What Is an Object?

The document uses "object" throughout without formally defining it. Provisionally:

```
Object = (I_payload, I_history, S)
```

where **S** is a *changing collection of semantic interpretations* rather than a fixed semantic identity — consistent with §3's claim that semantic location is derived, not canonical. This matters once third parties start building applications on top of the archive: they need to know whether they're depending on the payload, the history, or a particular (possibly stale) semantic view.

2. Economic Constitution: Three Resource Classes

1. **Digital information** — admitted and used without per-object charges to the user.
2. **Mesh resources** — storage, computation, bandwidth, verification, and maintenance, supplied by participants and compensated according to useful, *verified* service.
3. **Physical realization** — consumes matter, energy, machinery, labour, certification, and delivery, and is therefore purchased by the user.

"Free" describes the **user-facing contract**, not an assumption that resources are costless. The specification needs an explicit subsidy equation describing how idle-

resource contribution, platform revenue, manufacturing margins, and cooperative allocation together absorb the real cost of digital abundance.

3. The Object State Model

Most storage systems have only one or two states (exists/doesn't, public/private). This architecture requires a richer, **orthogonal** state vector rather than a pipeline:

$$O = (A, P, V, D, T, C)$$

Symbol	Dimension	Meaning
A	Acceptance	The network has received the object and assigned it an identity. Nothing else follows automatically.
P	Preservation	The network has committed resources to ensure the object survives node failure and time.
V	Availability	Authorized parties can retrieve it.
D	Discoverability	It appears in search, semantic neighborhoods,

Symbol	Dimension	Meaning
		recommendations, indexes.
T	Transformability	Derivative processes are permitted: OCR, transcription, embedding generation, training, style transfer, summarization, printing, etc.
C	Commercial eligibility	The object may participate in revenue-generating activity.

These are **orthogonal, not sequential**. A private family photo can be Accepted, Preserved, and Available to its owner while remaining non-Discoverable, non-Transformable, and non-Commercial. A public-domain book can be active on every dimension. A copyrighted song might be Preserved and Discoverable but never Commercial.

Open design question: each dimension likely needs internal subpermissions — e.g. Transformability probably needs to distinguish OCR/indexing consent from AI-training consent from physical-printing consent, since these carry very different implications for the uploader.

A Seventh Dimension: Governance

The six-dimensional vector describes *what* is permitted, but not *who can change what's permitted*. Questions like "can I permit indexing but not AI training?", "can I authorize

printing but not resale?", "can I delegate another party to grant commercial rights?", or "can a library preserve something it cannot itself distribute?" aren't quite Discoverability, Transformability, or Commercial eligibility questions — they're questions about **grant authority**. Without a dedicated dimension, permission management risks tangling itself inside **T** and **C**. Provisionally:

$$O = (A, P, V, D, T, C, G)$$

where **G** governs who may alter the other six dimensions, and under what delegation chain.

Handling duplicates and edge cases

If one user privately uploads an image and another later uploads the same image publicly, these become **two distinct object records**, even though a content-level identity relation exists between them. The public object becomes Discoverable; the private one remains private. The semantic layer can register the relationship without collapsing ownership, permissions, or provenance.

Three layers of identity

This requires separating what most systems collapse into one:

- **I_payload** — the actual bytes/media content.
- **I_semantic** — the object's location in embedding/manifold space.
- **I_history** — provenance, ownership, permissions, transformations, economic relationships.

I_semantic is explicitly not canonical identity. The manifold is a *learned, derived* neighborhood representing relationships among files — not "the file" itself. If semantic location were treated as identity, every model improvement would force a migration. Treating it as a derived index instead means the archive can be **re-indexed without being re-identified** — likely the single most important invariant in the whole system, since it decouples archive integrity from model quality over time.

Open design questions (unresolved):

- Who can trigger state transitions, and under what authority (e.g., can commercial eligibility be granted retroactively by a rights holder other than the original uploader)?
- Does re-classification into a new semantic neighborhood require re-consent for Discoverability?
- How are conflicting claims over I_history resolved when two objects share I_payload (e.g. one uploader claims public domain, another claims copyright over identical bytes)?
- Is the state vector itself versioned as part of I_history, or is provenance restricted to payload transformation lineage only?

4. Storage & the Semantic Layer

Uploads are not treated as isolated blobs. The network discovers common structure across uploads via semantic embeddings, locality-sensitive hashing, clustering, and

manifold learning — meaning the archive becomes more organized, not just larger, as it grows.

Suggested hierarchy:

```
Raw files → Content-addressed objects →  
Semantic embeddings →  
Clusters → Concepts → Navigable knowledge  
structures → Services
```

Important constitutional caution: "storage gets cheaper the more people upload" is an empirical claim, not a guaranteed one — it depends on corpus redundancy and semantic density, which adversarial or genuinely novel uploads won't necessarily provide. The constitutional wording should instead read:

"The architecture seeks to exploit redundancy, similarity, shared structure, and semantic organization to reduce the marginal infrastructure cost of preservation and retrieval."

This is a design objective, measurable but not guaranteed — the document should not promise a trend as if it were an invariant.

Payload vs. history under similarity: hash equality identifies byte identity; perceptual hashing identifies close media similarity; semantic manifolds identify conceptual relationship. None of these alone determines redundancy — two perceptually identical files may carry different provenance, encoding history, annotations, resolution, or ownership claims, and the architecture must preserve that distinction rather than deduplicating it away.

Existence, Preservation, and Remembering

Preservation is currently treated as largely a storage concern, but a preserved object nobody can find, classify, relate, or understand is archivally present yet functionally absent. Three states worth distinguishing:

Existence $\neq$ Preservation $\neq$ Remembering

An object can exist without being preserved (accepted but not committed to durable storage). It can be preserved without being remembered (durable but unindexed, unfindable, unrelated to anything). It can be remembered through derivatives even after the original is gone (a lost book surviving only through citations and summaries). This justifies treating indexing, clustering, and curation as first-class compensated services rather than conveniences layered on top of storage.

Storage, Replication, and Repair Are Not the Same

- **Storage** preserves objects.
- **Replication** survives failures.
- **Repair** restores continuity *after* failures.

Two archives can hold identical bytes while differing dramatically in repairability. Draft invariant: **preservation without repair is insufficient for long-term continuity**. This gives a theoretical basis for compensating maintenance and verification work, not just raw storage.

Objects vs. Views

Following from §3's payload/semantic-identity split: the payload is stable, but search results, recommendations, clusters, categories, and generated collections are all **views** — $\text{View}(M, t)$, a function of a model M at a time t , not part of the object itself. This is the same idea as Invariant 7 (re-indexable without re-identification), stated at the level of user-facing outputs rather than internal architecture: the archive stays stable while interpretations of it change as models improve.

Semantic Capital

Contributors are currently paid for storage, bandwidth, compute, indexing, and fabrication — but the network's value also grows from *relationships discovered between objects* (a dictionary, a corpus, and a translation model together are worth more than the sum of their parts).

Provisional definition:

Semantic capital is the increase in navigability, discoverability, interpretability, and utility generated by relationships among objects, rather than by the objects individually.

This gives a theoretical justification for funding indexing and manifold construction as value-generating activity in their own right, not overhead.

5. Optimization Target

The system does **not** primarily optimize for storage utilization, contributor income, popularity, or manufacturing

throughput. It optimizes for:

Collective informational utility produced per unit of scarce physical resource consumed.

An upload is valuable to the system when it strengthens semantic coverage, improves categorization, supplies a missing variant, reinforces provenance, enables better compression/indexing, or expands what the system can later retrieve, transform, or physically realize.

6. Contributor Economy

Payment attaches to **verified service**, not to uploaded content:

- **Storage providers** — paid for preservation.
- **Bandwidth providers** — paid for availability.
- **Compute providers** — paid for transformations and indexing.
- **Index builders** — paid for semantic organization.
- **Fabricators** — paid for physical realization.

Yet the system depends heavily on metadata quality, provenance verification, and dispute resolution — none of which is pure computation. A missing role:

curation/verification providers, whose job is maintaining trustworthy history rather than moving bytes or cycles. This becomes especially important once the archive holds books, photographs, cultural artifacts, research datasets,

and historical documents where provenance disputes are inevitable.

The uploaded object is not the thing being rewarded — it is the thing *around which services are organized*.

7. Physical Realization Layer

Physical goods are **not an add-on** — they are the terminal points of the pipeline:

```
Information → Organization → Transformation →  
Physical Realization
```

Each physical object should retain a resolvable link to its digital source, transformation history, production specification, manufacturer, and edition. A printed book is then a "**signed physical realization**" of a **versioned digital object**, not merely merchandise.

Physical Realization as Continuity, Not Just Commerce

The pipeline reads more precisely as:

```
Information → Knowledge → Artifact →  
Continuity
```

A book, watch, engraved plaque, educational kit, or printed photograph is harder to erase than a database entry. Physical realization is therefore not merely a revenue

mechanism — it's a **continuity mechanism**: a way of making some subset of the archive resistant to the kinds of failure (node churn, model obsolescence, platform collapse) the rest of the architecture is built to survive.

Example product lines

- Watches with eleven-hour clocks that time-sync with the [Haplopraxis](#) universe — one example of a physical product line under this layer; Haplopraxis is an independent project with its own design document, and URFN does not fund or depend on it beyond sharing this one artifact. A digital variant (displaying in-game year rather than date) and a companion color-changing orb (status/attack indicator) are also planned; see the Haplopraxis design document for the reasoning behind each.
 - Books embossed in Standard Galactic and Simplified Cursive Galactic scripts (also Latin and parallel-script editions).
 - Photo/album prints, posters, flash cards, educational kits, collectible editions.
 - Appliances and replacement parts (longer-term — carries meaningfully more product-liability/regulatory weight than books or watches, and is probably not a Phase I–III target).
 - Community activities and competitions built around the ecosystem: paper recycling, yogurt making, model-factory-making competitions.
-

8. Architectural Domains

Identity & trust · Storage & replication · Compute scheduling
· Contribution verification · Accounting & payments ·
Content addressing & metadata · Permissions, licensing &
provenance · Discovery & retrieval · Physical production ·
Governance & dispute resolution · Security & abuse
resistance.

9. Layered Architecture

Layer	Contents
0 – Resource Fabric	Storage, compute, bandwidth, energy, fabrication resources
1 – Verification	Proofs of storage, proofs of computation, reputation, auditing, dispute resolution
2 – Information Objects	Content-addressed files, metadata, version history, licensing, provenance
3 – Knowledge Services	Search, indexing, recommendation, translation, transcription, summarization, AI services
4 – Physical Realization	Books, electronics, clothing, tools, appliances, artwork, replacement parts
5 – Economic Coordination	Payments, rewards, royalties, revenue sharing, marketplace functions

10. Phased Rollout

1. **Archive** — uploads, content addressing, deduplication, semantic indexing, distributed storage.
 2. **Intelligence layer** — embeddings, clustering, classification, recommendation, search, transcription, translation, knowledge extraction.
 3. **Resource mesh** — idle storage/compute become tradable, verified, compensated resources.
 4. **Creator economy** — books, artwork, music collections, educational materials generated directly from archived content.
 5. **Physical realization** — full manufacturing/fulfillment marketplace.
-

11. Founding Invariants (draft)

1. Information objects are globally addressable.
2. Storage and computation are contributed voluntarily.
3. Contributors are compensated only for verifiable service — never for upload volume alone.
4. Physical production consumes scarce resources and therefore carries cost.
5. Provenance and licensing travel with information objects.
6. No single node is required for network operation; the network remains functional despite arbitrary node churn.

7. **The archive must be re-indexable without being re-identified** — semantic location is a derived index, never canonical identity.
 8. Object permission states (A, P, V, D, T, C) are orthogonal — no single decision about an object automatically grants all of them.
-

12. Unresolved / Needs Further Work

- Formal dispute-resolution process for conflicting `L_history` claims over shared `L_payload`.
- Granularity of Transformability subpermissions (indexing vs. training vs. physical output), and how the new Governance (`G`) dimension interacts with them.
- Proof-of-storage / proof-of-compute mechanism — adapt existing schemes (e.g. proof-of-replication) or design new ones.
- Explicit subsidy equation for what actually absorbs the cost behind "free" digital access.
- Legal/regulatory posture for physical fulfillment (product liability, shipping, customs) — likely needs its own trust model, separable from the P2P layers.
- Governance model for who can alter invariants once the network is live.
- **What is the unit of value?** Conventional networks implicitly value bytes, requests, CPU-hours, or dollars. This architecture is drifting toward coverage, preservation, continuity, and semantic capital instead — none of which are ordinary accounting quantities. At

some point the system likely needs a concept like "reachability volume" or "coverage contribution" so payment and optimization are tied to what the constitution actually values, rather than to whatever happens to be easiest to meter.

- **Candidate future chapter — Memory, Repair, and Continuity:** sits between the Object State Model (§3) and the Semantic Layer (§4), and would formally connect the existence/preservation/remembering distinction, the storage/replication/repair distinction, and the objects-vs-views distinction into one coherent section rather than leaving them scattered across §3–4 as they currently are.