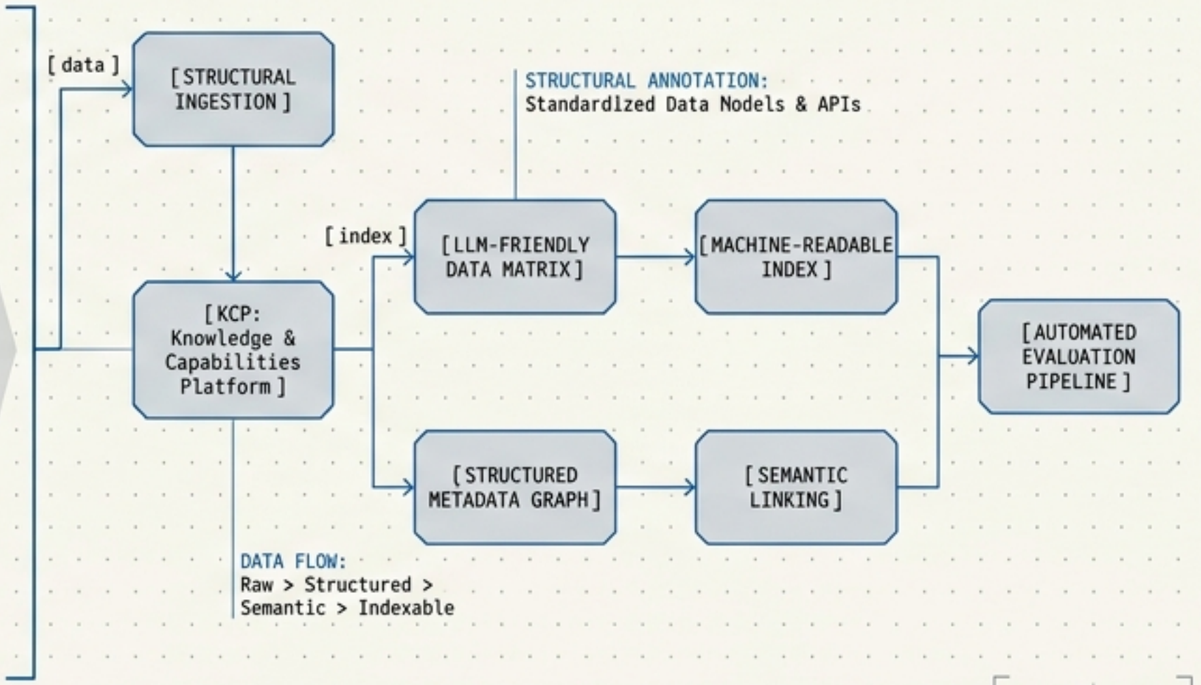
Machine-Readable Consulting

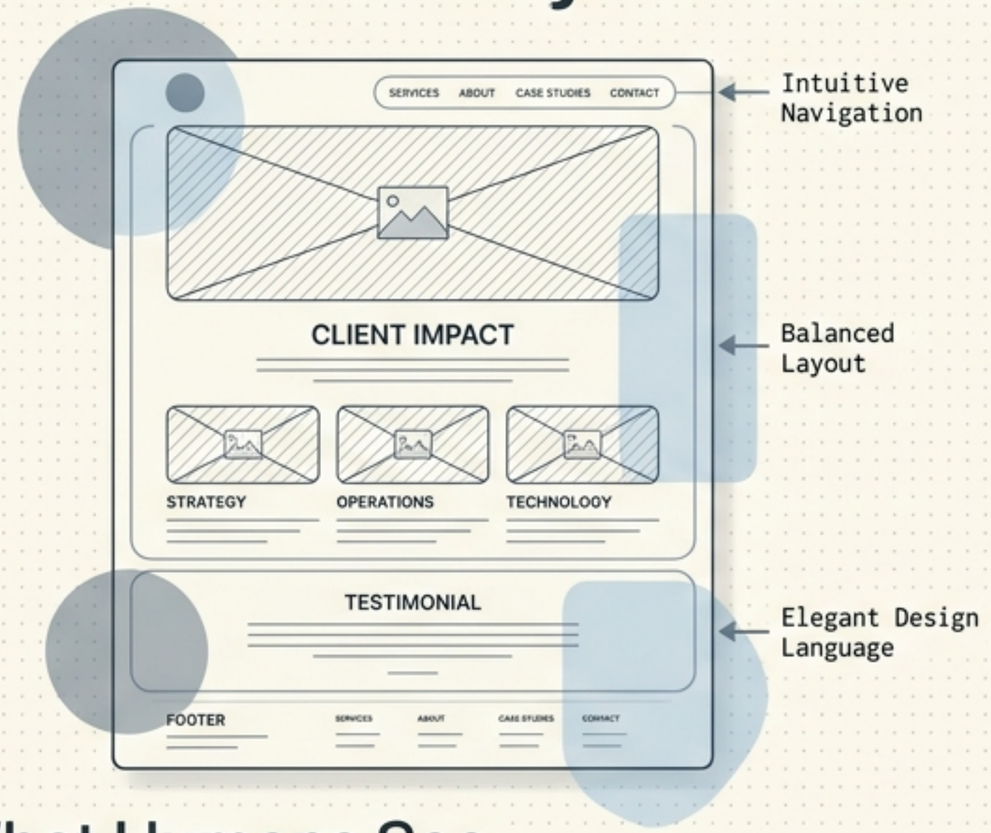
Implementing KCP and LLM-Friendly Structures

> ⚠ If your consulting firm isn't machine-readable, you are invisible to half the evaluation pipeline.

```
<h1>Unstructured Data Input</h1>
data point 1, text string, unorganized content
<div class="raw-input">
  { "name": "consulting firm A", "service": "strategy",
    "notes": "not standardized", ...
  }
  <ul><li>service description</li<li>no clear index</li></ul>
  </ul>
    file_path/to/unstructured/report_v3.docx
    [unclassified info]
    some random text here and there
    <span class="legacy-tag">old format</span>
    data={"key": "value", "nested": "chaos"}
</div>
```

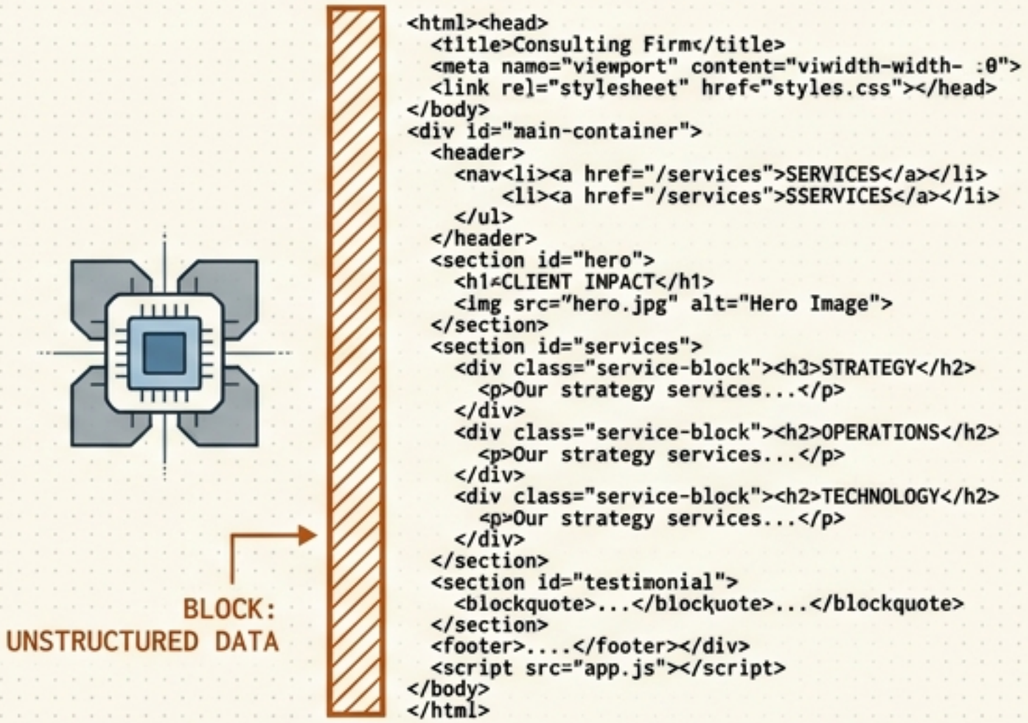


Evaluated by Machines, Designed for Humans



What Humans See

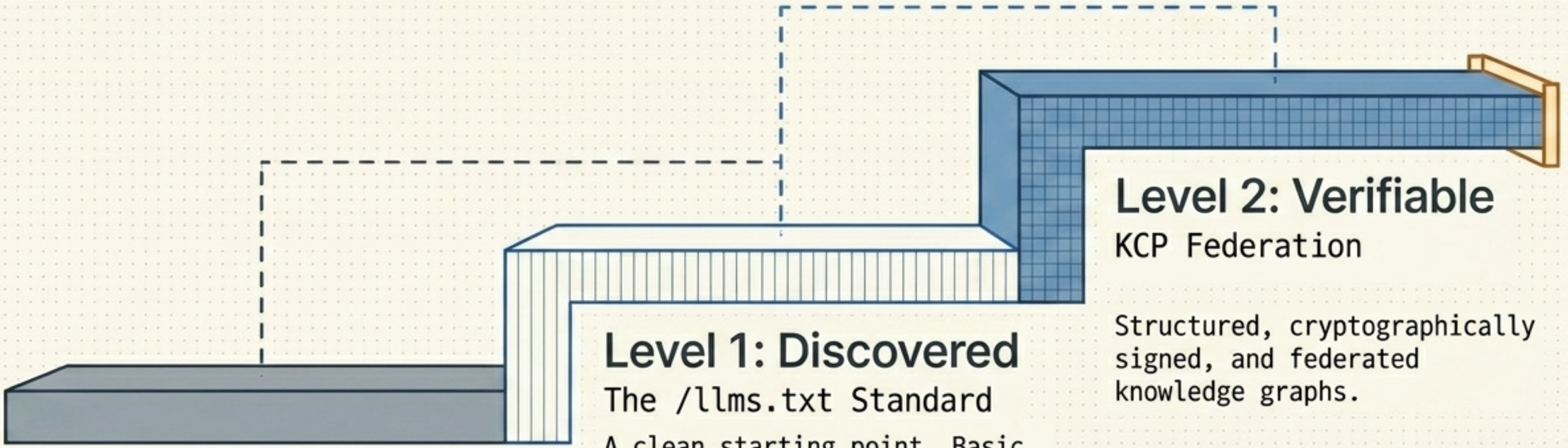
Human clients want elegant design, intuitive navigation, and persuasive copy.



What Agents See: Noise

AI agents—used by clients to evaluate services, methodologies, and pricing—just see a wall of HTML. If an agent has to reverse-engineer your navigation to find your offerings, you lose the deal before the first human conversation even starts.

The AI-Visibility Maturity Scale



Level 0: Legacy

/HTML Only

Completely invisible to agents.
Requires active reverse-engineering.

Level 1: Discovered

The /llms.txt Standard

A clean starting point. Basic, text-based orientation for language models.

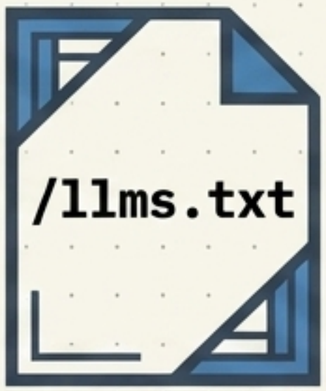
Level 2: Verifiable

KCP Federation

Structured, cryptographically signed, and federated knowledge graphs.

The Low Bar: Deploying /llms.txt

10 Minutes



The Spec

The llmstxt.org standard is refreshingly simple. A single Markdown file describing your site, what you offer, where to find things, and necessary context.

The Investment

5 minutes of writing.
5 minutes of deployment.

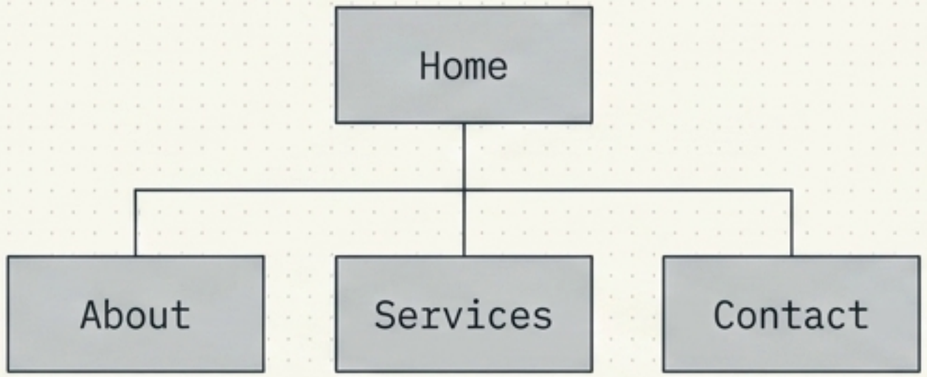
The ROI

Any agent querying your site gets a clean, explicit starting point rather than guessing your structure.

There is no excuse not to have this. But it is only the beginning.

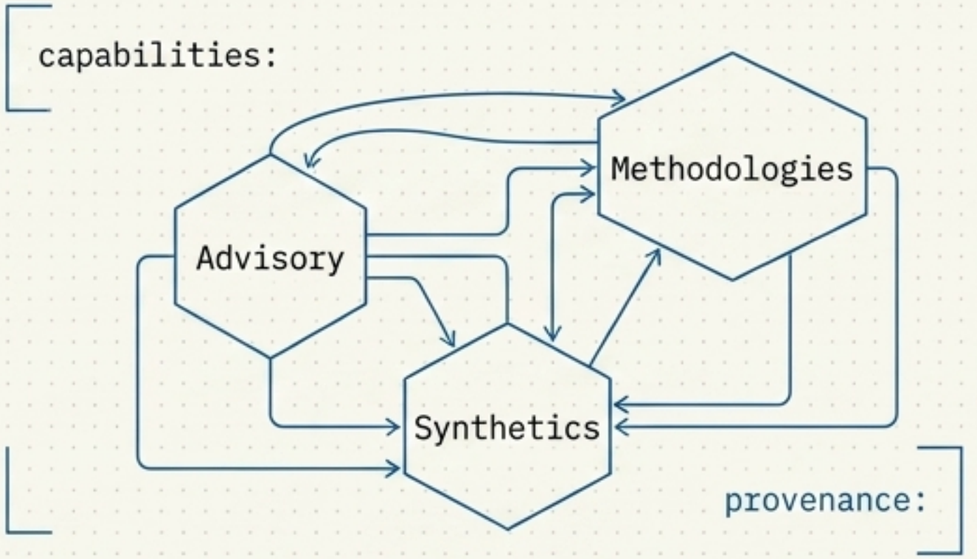
Moving to Structured Logic with KCP

Traditional Site Map



Maps URLs

Knowledge Map



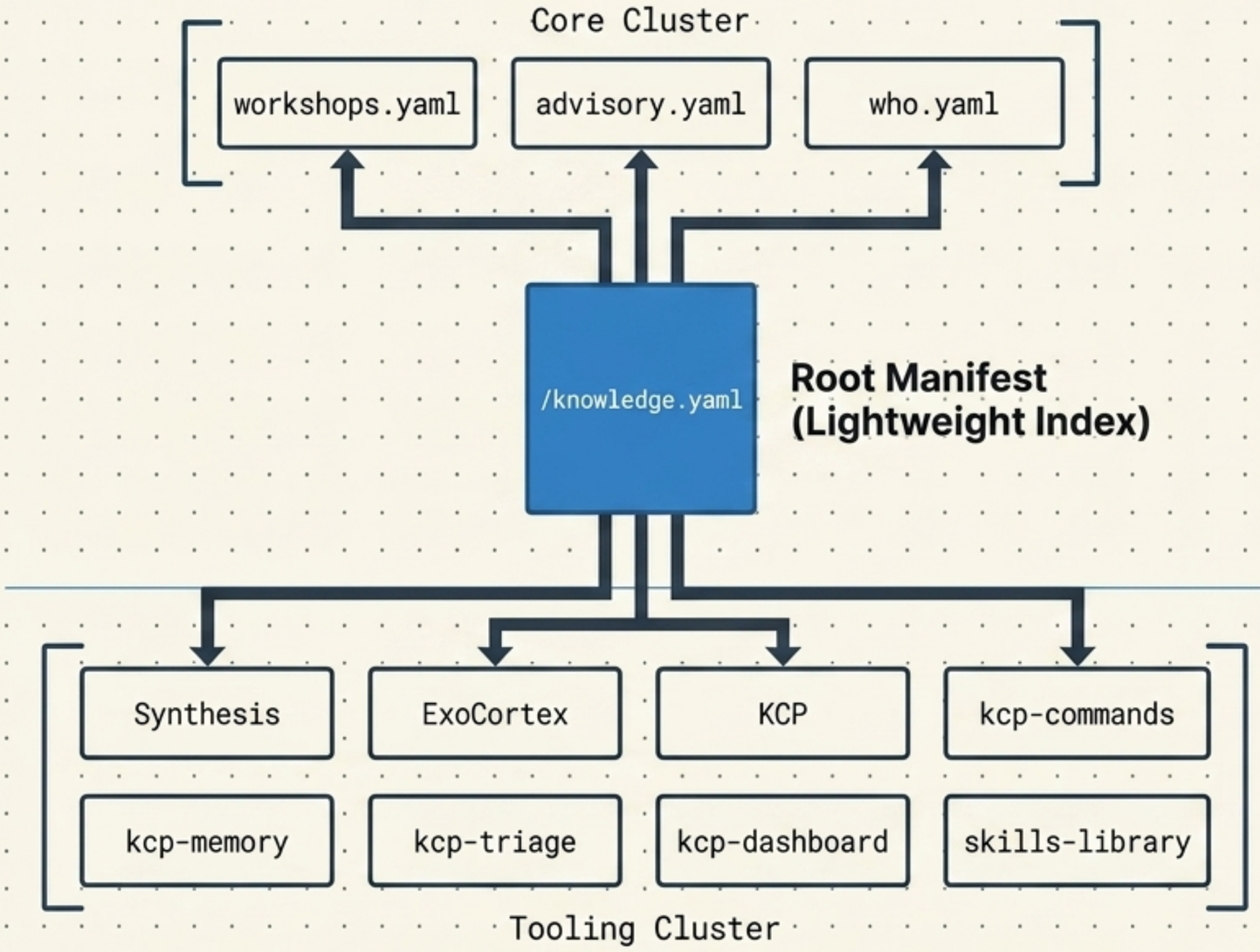
Maps Capabilities

Knowledge Context Protocol (KCP)

A YAML-based protocol for publishing structured knowledge as a machine-readable graph.

- Maps concepts and capabilities, not just web pages.
- Bakes in structural authority and data provenance.
- Transforms unstructured marketing text into actionable intelligence.

Architectural Blueprint: KCP Federation



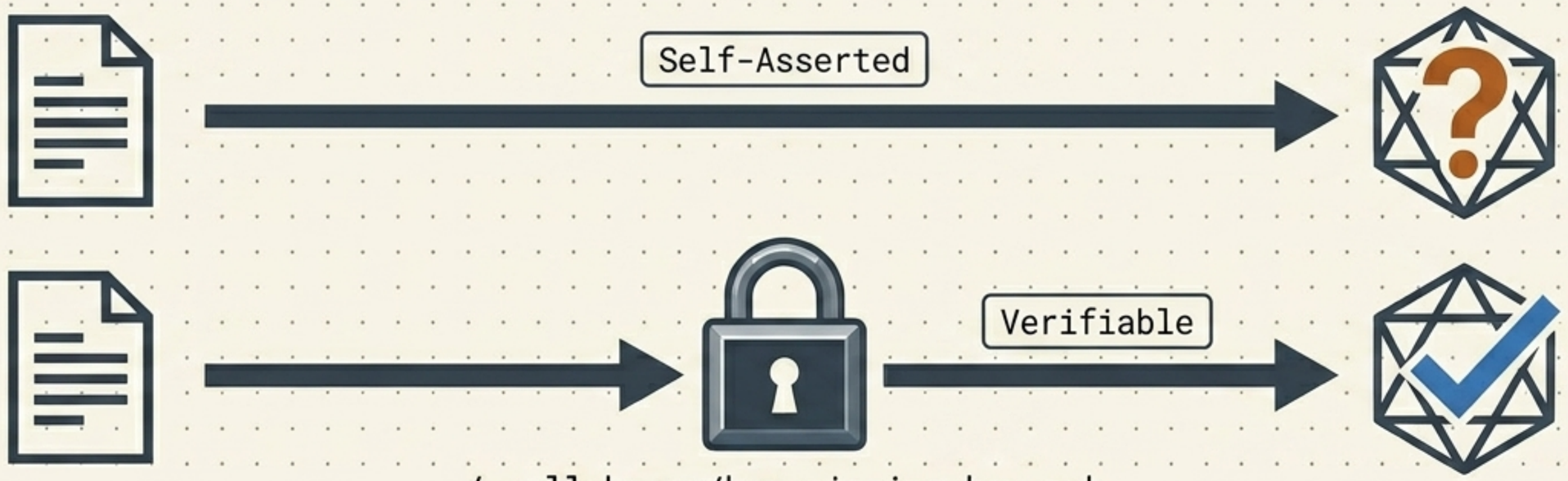
Anti-Maintenance Hell

A single flat manifest breaks at scale. The lightweight root index delegates to self-contained sub-manifests. Federation keeps each domain of knowledge owned and updated by the experts who actually understand it.

Protocol Diagnostics: Indexing vs. Mapping

	/llms.txt	KCP Federation
Format	Standard Markdown	Structured YAML Graph
Primary Purpose	Basic Agent Navigation & Orientation	Granular Capability & Context Mapping
Architecture Scale	Single, monolithic file	Root index with modular, federated sub-manifests
Provenance	Unverified, self-asserted text	Cryptographically signed and verifiable trust

Establishing Verifiable Trust
The Provenance Pathway



Core Principle

In an AI-driven web, canonical truth is self-asserted unless you sign it. Signing makes your expertise verifiable.

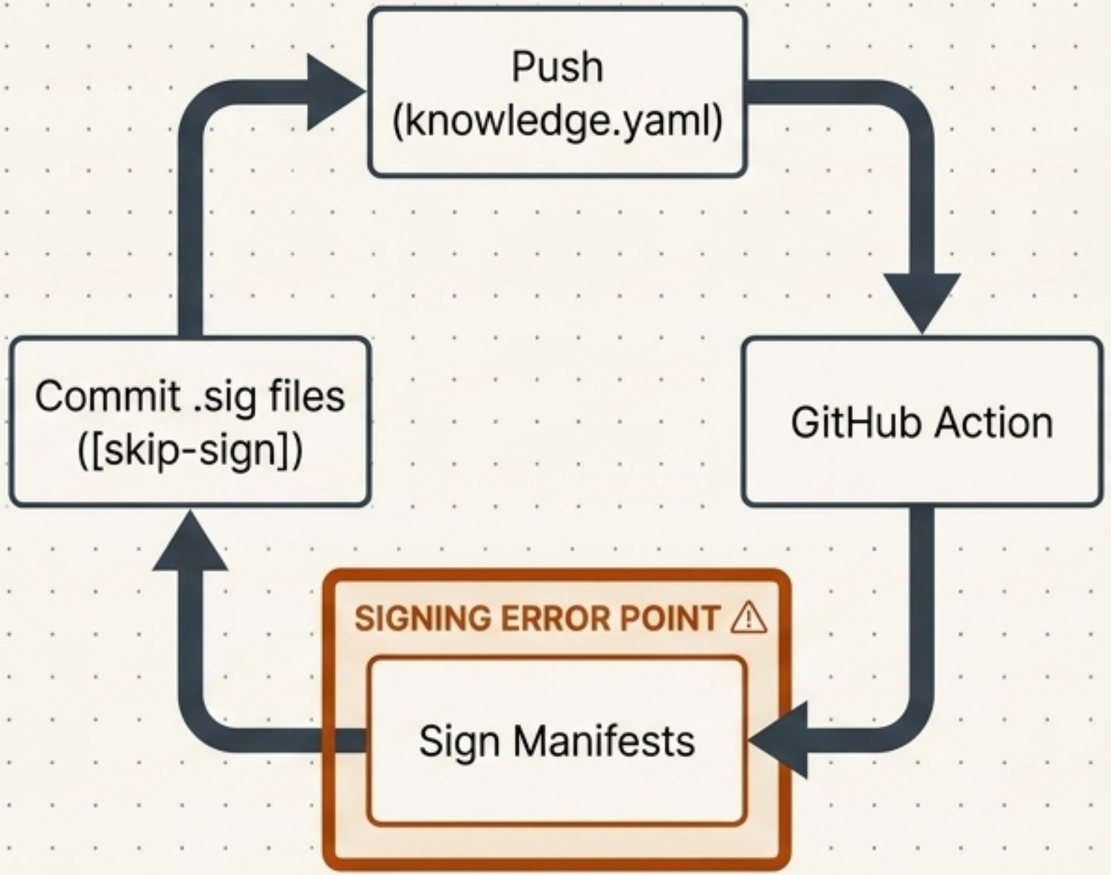
Implementation

Each sub-manifest is independently signed. The root manifest trusts the federation structure; the signing anchors the provenance.

The Future-Proof Advantage

A partner can contribute a sub-manifest, sign it with their own key, and explicitly anchor their specific provenance within your root federation.

The Automation Loop & The Cryptographic Gotcha



Crucial: Use `[skip-sign]` in the commit message to prevent an infinite GitHub Actions trigger loop.

⚠ The Ed25519 `-rawin` Gotcha

Error:
`operation not supported for this keytype`

The Reality: Ed25519 signs raw bytes directly; it does not use an external digest algorithm.

The Fix: Using `openssl pkeyutl` without the `-rawin` flag attempts a separate hashing step that Ed25519 rejects. You must force raw input.

The Machine-Readable Mandate

1.

Deploy /llms.txt Today.

Execute the 10-minute quick win to orient evaluating agents immediately.

2.

Map Capabilities via KCP.

Build a federated YAML structure (/knowledge.yaml) to avoid single-file maintenance hell.

3.

Automate Cryptographic Trust.

Implement per-manifest signing to transform self-asserted claims into verifiable canonical truth.

Do not let your firm's expertise get lost in translation between your server and their agent.