



A Constitution for Knowledge Systems

Principles embodied by CoreQuarry

Preamble

Knowledge systems should not be defined by the technologies of their time, but by the principles that remain valid as technologies evolve.

Representations change. Storage models change. Query languages change. Artificial intelligence will change. Yet the need to preserve, discover, relate, and understand knowledge endures.

This Constitution sets out the architectural principles for knowledge systems intended to remain useful across changing technologies, domains, and generations.

Article I. Knowledge is Derived

Information is not knowledge.

Knowledge arises through interpretation, communication, context, provenance, and inquiry.

A knowledge system therefore preserves the conditions from which knowledge may be derived rather than claiming to contain knowledge itself.

Article II. Artifacts are Primary

The fundamental objects of a knowledge system are artifacts.

Artifacts may be documents, records, fragments, observations, messages, datasets, images, software, or other representations of human or computational activity.

No particular representation shall be privileged.

Article III. Essence is Independent of Representation

Representation is not essence.

XML, JSON, MIME, RDF, relational tables, binary formats, or future representations are expressions of underlying abstractions rather than the abstractions themselves.

A knowledge system should preserve structural and semantic continuity independently of presentation syntax.

A Constitution for Knowledge Systems

Article IV. Inquiry Determines Retrieval

Storage boundaries shall not determine the boundaries of inquiry.

The appropriate unit of retrieval is determined by the question being asked.

An inquiry may retrieve a fragment, a document, a record, a collection, a structural element, or a composed result drawn from multiple sources.

Article V. Authority Shall Be Respected

Not all authoritative knowledge belongs within a repository.

Knowledge systems shall distinguish between preserved artifacts and maintained operational state.

Authoritative operational systems should remain authoritative while participating in inquiry when appropriate.

Article VI. Structure Precedes Implementation

Architectural abstractions should precede implementation choices.

Data structures, storage engines, query languages, indexing methods, embedding models, and retrieval algorithms are realizations of architectural principles rather than their definition.

No implementation shall define the architecture.

Article VII. Semantic Continuity Shall Be Preserved

Names, schemas, and representations evolve.

Meaning should remain discoverable despite those changes.

Knowledge systems should preserve semantic continuity through explicit mappings and structural abstractions rather than requiring uniform representations.

Article VIII. Generalization Before Specialization

Architectures evolve through discovering broader abstractions.

Capabilities that reveal general principles belong within the architecture.

Requirements that express local policy belong at explicit extension points.

A Constitution for Knowledge Systems

The architecture shall grow through abstraction rather than the accumulation of exceptions.

Article IX. Inquiry May Be Composed

Answers need not originate from a single repository.

Inquiry may compose preserved artifacts, derived knowledge, delegated capabilities, and authoritative external systems into a coherent and defensible result.

Composition is a property of inquiry rather than storage.

Article X. Provenance Enables Trust

Every contribution to an inquiry should be attributable.

The origin of artifacts, derived results, delegated evaluations, and transformations should remain visible.

Trust depends not upon certainty but upon defensibility.

Article XI. Engineering Accepts Uncertainty

Uncertainty is an inherent property of knowledge.

The objective of engineering is not to eliminate uncertainty, but to design systems in which uncertainty is visible, decisions are defensible, and failures remain proportionate to their consequences.

Article XII. Architectures Should Outlive Technologies

Technologies evolve.

Architectural principles endure.

A successful knowledge system is measured not by the popularity of its implementations, but by the continued relevance of the abstractions upon which they are built.