Enterprise Data Lifecycle™ Specification

Overview

Enterprise information often survives far longer than the technologies, applications, organizations, and business processes that created it. During that time, its value, use, relationships, risks, regulatory obligations, security requirements, and preservation needs can change substantially. Traditional lifecycle approaches often respond primarily to age, storage location, access frequency, or infrastructure cost. The **Enterprise Data Lifecycle™ (EDL)** takes a different approach: information is governed according to what it is, what it means, its current business and governance context, and what the organization requires from it.

The mission of the Enterprise Data Lifecycle is to enable organizations to govern enterprise information as a strategic business asset throughout its existence through consistent, policy-driven, and technology-independent decision-making. The Specification establishes the canonical terminology, models, architectural responsibilities, behaviors, and conformance requirements necessary to achieve that objective. It deliberately does not prescribe particular products, vendors, storage technologies, programming languages, deployment models, or implementation techniques.

Information Lifecycle Is Not Storage Lifecycle

A foundational principle of EDL is that the **business state of information is different from its storage state**. Information does not become unimportant merely because it becomes old, nor does moving information between SSD, disk, object storage, optical media, cloud, tape, or another technology inherently change its business lifecycle.

EDL therefore defines canonical lifecycle stages that describe the evolving **business state** of information, while a broader **operational state** captures the governance-relevant conditions surrounding it. Multiple information objects can occupy the same lifecycle stage while having very different retention, protection, security, availability, integrity, risk, value, relationship, and other requirements.

The **Enterprise Data State Vector (EDSV)** provides a canonical representation of that governed state. Rather than making lifecycle decisions from age or storage characteristics alone, policy and governance capabilities can evaluate a richer representation of the information and its current context.

Persistent Information Identity and Context

At the center of the architecture is the **Enterprise Information Object (EIO)**, which provides persistent governed identity for a logical information asset. That identity allows metadata, assertions, representations, relationships, Events, lifecycle conditions, Governance Decisions, authorizations, Authorized Actions, Outcomes, verification information, evidence, and other

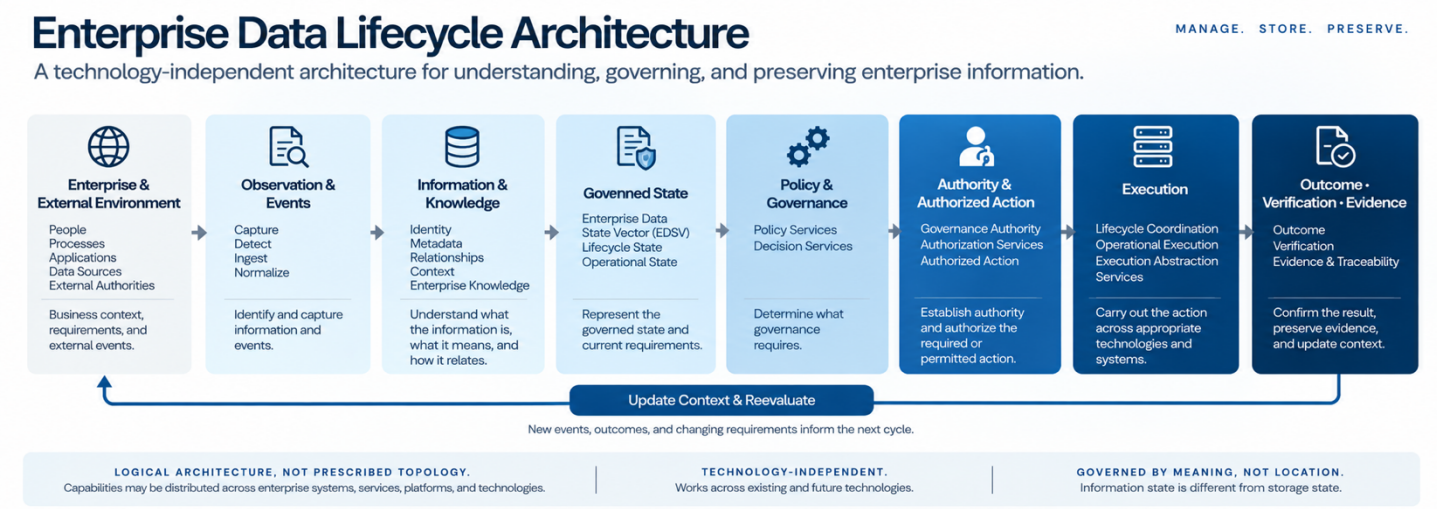
relevant context to remain associated with information over time even when the underlying technology changes.

EDL extends this understanding through canonical metadata, identity, relationships, events, assertions, provenance, and **Enterprise Knowledge**. Together, these constructs allow information to be understood not simply as a file, database row, object, or storage asset, but in the context of the enterprise: what it represents, where it came from, what it relates to, what has happened to it, and what governance requirements apply to it.

Policy-Driven Governance

The Enterprise Data Lifecycle separates **understanding information, evaluating policy, making governance decisions, establishing authority, authorizing action, executing work, and verifying results**. These responsibilities may exist within the same software product or across many enterprise systems, but their architectural meanings remain distinct.

Conceptually, the architecture connects:



This relationship is not a mandatory software workflow or processing sequence. Implementations may enter the architecture at different points, use only the capabilities relevant to a particular activity, distribute capabilities among different systems, or revisit them as information changes. What EDL preserves is the meaning and traceable relationship among the architectural constructs.

This separation is particularly important at the boundary between **decision and action**. A Governance Decision establishes what governance requires; it does not automatically authorize operational execution. Governance Authority and authorization determine whether and under what authority work may proceed. Authorized Actions then express the permitted or required work, while operational technologies determine how that work is actually performed.

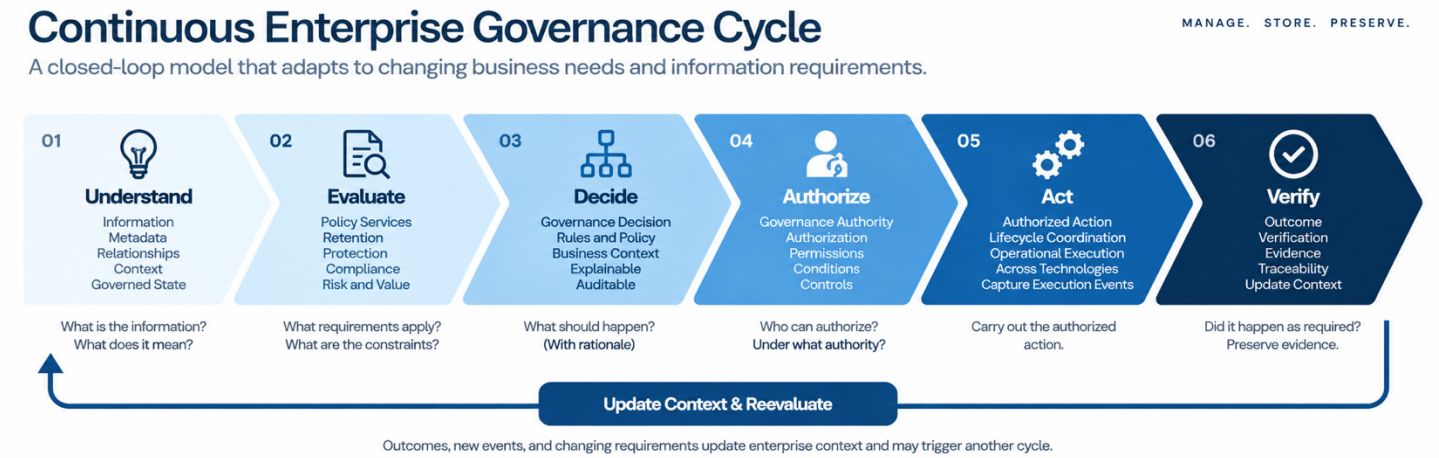
Governance Is a Continuous Cycle

EDL does not treat execution as the end of governance.

Operational activity produces new information. An action may succeed, fail, partially complete, create a new representation, change a relationship, alter availability, move information, preserve it, restrict it, or dispose of it. The architecture therefore establishes **Outcome, Verification, Evidence, and Traceability** as first-class governance responsibilities.

Outcomes establish what materially occurred. Verification determines whether the required result was actually achieved. Evidence preserves the information necessary to substantiate and reconstruct the activity. Those results return to enterprise context and may change Enterprise Knowledge, the EDSV, lifecycle state, policy applicability, or other governance conditions.

The result is a continuous governance cycle:



Governance therefore responds to changes in information and its requirements rather than relying on assumptions established at an earlier point in time.

Technology-Independent Enterprise Architecture

The Specification brings the lifecycle, governance, information, services, policy-execution, integration, and reference architectures together into a coherent logical model. The Reference Architecture defines **logical responsibilities and canonical behavior**, not a required physical topology. Capabilities may be consolidated, distributed, federated, supplied by existing enterprise systems, or provided by external services.

This allows EDL to operate across existing applications, repositories, databases, cloud services, identity and security systems, analytics and AI platforms, infrastructure, storage technologies, and external authorities without requiring those systems to be replaced or consolidated into a single platform.

The architectural objective is therefore not to standardize **where information is stored** or **which technology manages it**. It is to establish consistent meaning, governance, responsibility, authority, and traceability as information moves across technologies and through time.

Conformance, Adoption, and Evolution

The Specification also defines how implementations can make meaningful **conformance claims**. Conformance is evaluated against a declared scope because an EDL implementation does not need to contain every architectural capability within a single product or platform. Required capabilities may be supplied through identified enterprise, distributed, external, or federated dependencies. What matters is that the applicable architectural responsibilities, canonical semantics, boundaries, interactions, and behaviors are preserved within the claimed scope.

This distinction also supports incremental adoption. Organizations can begin with a defined information population, governance problem, business domain, or set of capabilities and expand over time. Adoption is measured by improvements in information understanding, governance consistency, policy effectiveness, traceability, execution, verified Outcomes, and other business and governance objectives, not simply by the amount of technology deployed.

Ultimately, the Enterprise Data Lifecycle establishes an enduring architectural principle:

Enterprise information is the persistent asset. Technologies are transient participants in its lifecycle.

The objective is not to preserve a particular technology architecture. It is to preserve the enterprise's ability to **identify, understand, govern, protect, preserve, act upon, verify, and ultimately dispose of information correctly** as technologies, organizations, requirements, and business conditions change.

Information persists. Its governance must persist with it.