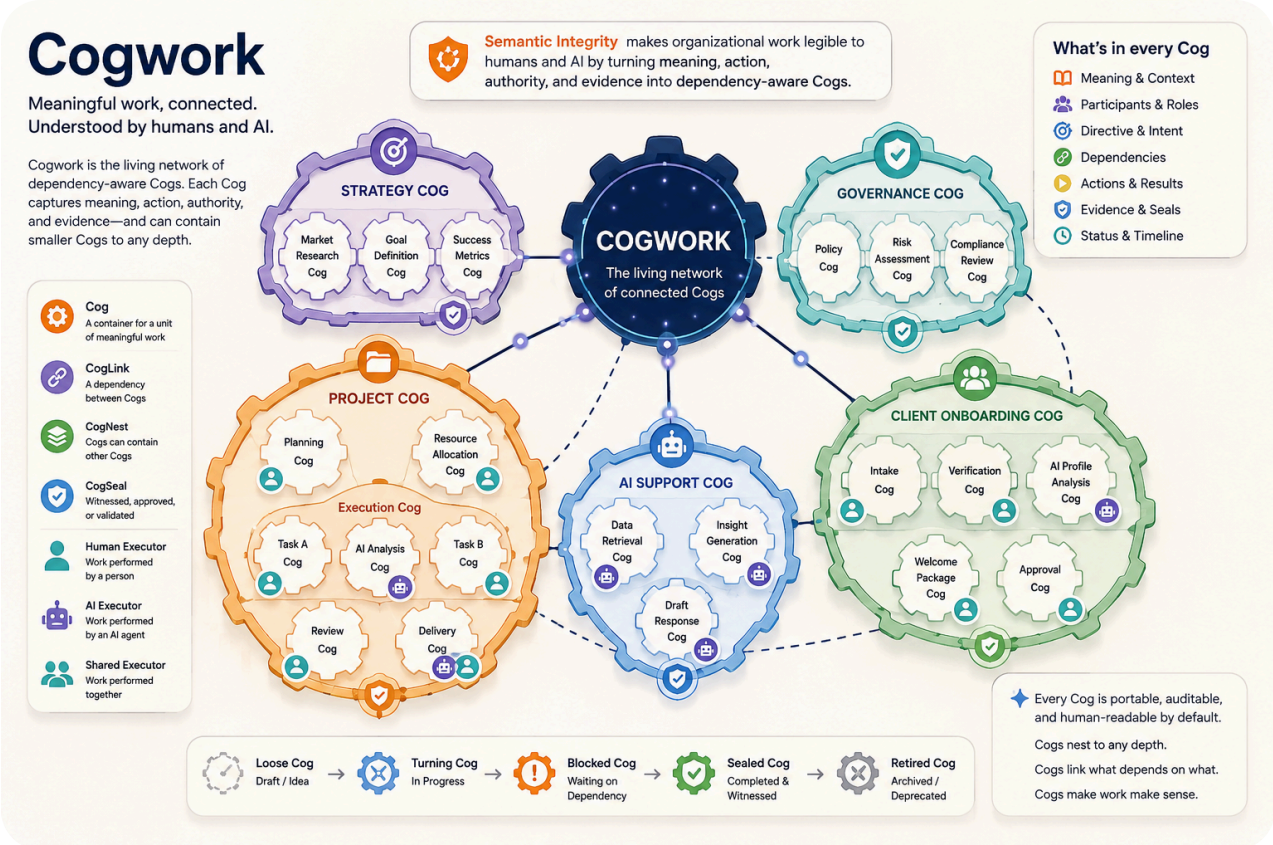
Semantic Integrity

Cog Primer v0.1
Print-ready HTML · May 2026

Cog Primer

A simple field guide to dependency-aware Cogs: the Lego-like infrastructure for making organizational work legible to humans and AI.

One container. Many bindings. Nested structure. Dependency-aware execution. Human-readable by default.



Cogwork is the living network of dependency-aware Cogs. Each Cog captures meaning, action, authority, and evidence—and can contain smaller Cogs to any depth.

POSITIONING

The core idea

Semantic Integrity makes organizational work legible to humans and AI by turning meaning, action, authority, and evidence into dependency-aware containers.

> ## A Cog is a small container of meaningful work.
>
> It shows what the work means, what it depends on, who can act, what happened, and how we know.

Protocol people can call the base object a **Semantic Integrity Container**. Everyone else can call it a **Cog**. A Cog can describe a task, decision, role, policy, model action, approval, meeting, workflow, evidence record, or whole organizational process.

WHY COGS EXIST

Organizations already contain intelligence.

Semantic Integrity does not start from the assumption that AI should replace the organization. The organization already has people, judgment, context, authority, memory, habits, roles, records, and feedback loops. The problem is that much of this intelligence is scattered across documents, chats, meetings, tickets, dashboards, inboxes, spreadsheets, and tacit memory.

Cogs make that intelligence legible. They do not flatten work into text. They preserve the relationships around work: meaning, action, authority, evidence, dependencies, witnesses, boundaries, and state.

> The goal is not more documentation. The goal is work that remains understandable across people, AI systems, workflows, audits, and time.

THE BASE UNIT

Every Cog answers six questions.

Meaning

What is this? Why does it exist? What context, scope, and definitions make it understandable?

Action

What should happen, what is happening, what happened, and what output is expected?

Authority

Who or what may act? Which role, identity, directive, permission, or approval applies?

Evidence

How do we know? What records, witness events, signatures, seals, or audit trail support it?

Dependency

What must exist, finish, authorize, inform, supply, or validate before this can proceed?

Nesting

What larger Cog contains this? What smaller Cogs does this contain?

VOCABULARY

The Cog vocabulary

Semantic Integrity uses one simple base object plus a small set of helper terms.

TERM	PLAIN MEANING
Cog	The basic unit of meaningful work. A human-readable, machine-readable container for meaning, action, authority, evidence, state, and dependencies.
CogLink	A dependency or relationship between Cogs. A CogLink can require, block, inform, validate, authorize, supply, follow, conflict with, or supersede another Cog.
CogNest	A parent-child structure. Cogs can contain smaller Cogs to any depth.
CogSeal	A witness, approval, validation, signature, or evidence marker. A sealed Cog has been completed and witnessed in a defined way.
CogTrail	The audit and lineage history: what changed, when, by whom or what, and under which authority.
CogMap	The searchable registry and graph view of Cogs, CogLinks, CogNests, CogSeals, and CogTrails.
CogKit	A reusable template or profile for common work patterns, such as approvals, AI actions, meetings, onboarding, policies, reviews, or escalations.
Cogwork	The living dependency graph of organizational meaning, action, authority, and evidence.

BINDINGS

One container. Many bindings.

A Cog stays simple because it does not need to become a different object type every time it does a different job. Instead, a Cog receives optional bindings.

Context Binding

Defines the interpretive frame, assumptions, scope, and relevant background.

Ontology Binding

Defines terms, meaning anchors, invariants, and local definitions.

Task Binding

Defines work to be performed, acceptance criteria, owner, timing, and expected output.

Directive Binding

Defines instructions, constraints, priorities, prohibitions, and source authority.

Role Binding

Defines capacity, responsibility, permissions, and what the role cannot do.

Identity Binding

Binds a specific person, AI agent, system, credential, or execution identity to a role or action.

Airlock Binding

Defines controlled inputs, outputs, entry conditions, exit conditions, and escalation behavior.

Witness Binding

Defines evidence, review, approval, validation, confidence, disputes, and seals.

DEPENDENCY-AWARE EXECUTION

Work becomes a legible graph.

A hierarchy shows where work lives. A dependency graph shows what work needs. Semantic Integrity keeps both visible.

Container tree

Company Cog → Department Cog → Workflow Cog → Task Cog → Model Action Cog

Dependency graph

Contract signed → Identity verified → Legal approval → AI draft generated → Human review → CogSeal

Define

Create the Cog

Link

Declare dependencies

Check

Readiness state

Execute

Human or AI action

Seal

Witness and record

A Cog may be **loose**, **waiting**, **ready**, **turning**, **blocked**, **completed**, **sealed**, **disputed**, or **retired**. These states make blockers and readiness visible without requiring people to reconstruct the whole story from memory.

HUMANS + AI

Cogs can be executed by people, AI, or both.

Semantic Integrity treats humans, AI agents, and systems as possible participants in work, but it does not confuse execution with authority. A model may draft, summarize, classify, check, route, or propose. A human may review, decide, approve, reject, witness, or escalate. A Cog makes those distinctions visible.

Human executor

Work performed by a person under a defined role, identity, authority, and witness expectation.

AI executor

Work performed by a model or agent inside explicit context, constraints, permissions, and data boundaries.

Shared executor

Work performed through collaboration, such as AI drafting and a human reviewing, approving, or sealing.

PRIVATE SEMANTIC RUNTIME

Private by design when the work requires it.

Regulated organizations often cannot send sensitive client, operational, legal, health, financial, or personnel data to uncontrolled cloud inference. Semantic Integrity can run through a **Private Semantic Runtime**: a local, sovereign, or client-controlled AI layer that keeps sensitive data inside approved boundaries while still making work legible to humans and AI.

Continuity Office defines the boundary. Semantic Integrity builds the runtime.

Continuity Office establishes the governance standard: what must remain legible, sovereign, reversible, auditable, and consent-aligned. Semantic Integrity implements those requirements through Cogs, CogLinks, CogSeals, CogTrails, airlocked workflows, and private or sovereign AI runtimes.

Local or sovereign inference

Model execution can happen on-prem, in a private VPC, in a sovereign AI cloud, or in another approved client-controlled environment.

Airlocked workflows

Inputs, outputs, context, retrieval, permissions, and escalation conditions are bounded before AI acts.

Auditable AI action

Each AI-mediated action can be captured as a Cog with authority, dependency, evidence, and witness state.

EXAMPLE

A regulated client onboarding Cog

A client onboarding process may contain multiple nested Cogs and cross-cutting dependencies.

```
cog_version: "0.1"

cog:
  id: "cog_client_onboarding"
  title: "Client Onboarding"
  profile: "workflow"
  status: "waiting"

frame:
  intent: "Safely onboard a regulated client while preserving data boundaries."
  scope:
    includes: ["identity verification", "risk review", "welcome package"]
    excludes: ["unapproved data export", "unwitnessed AI decision"]

bindings:
  context: { business_unit: "Client Success" }
  role: { owner: "Onboarding Lead" }
  directive: { rule: "Sensitive data remains inside approved runtime." }
  airlock: { entry_conditions: ["contract_signed", "identity_verified"] }
  witness: { required_seal: "human_review" }

links:
  requires: ["cog_contract_signed", "cog_identity_verified"]
  authorizes: ["cog_ai_profile_analysis"]
  validates: ["cog_risk_review"]

runtime:
  mode: "private_semantic_runtime"
  inference_boundary: "client-controlled"
  cloud_exposure: "not permitted"

record:
  expected_outputs: ["approved profile", "welcome package", "sealed onboarding
record"]
  actual_outputs: []

audit:
  seals: []
  trail: []
```

PRACTICAL USES

Where Cogs help first

AI readiness

Map sensitive workflows before introducing AI execution, retrieval, summarization, or routing.

Governance modernization

Make authority, approval, evidence, and escalation paths visible and testable.

Client onboarding

Track identity, contract, risk, welcome, approval, and evidence dependencies in one legible structure.

Policy and compliance

Bind policies to roles, tasks, evidence, exceptions, and witness requirements.

Knowledge continuity

Preserve why decisions happened, what context mattered, and what outputs were trusted.

Local AI deployment

Install or configure private AI runtimes only after the semantic boundaries and governance requirements are clear.

CLOSING

The promise

Cogs make work understandable. CogLinks show what depends on what. CogNests preserve structure. CogSeals establish witness and evidence. CogTrails preserve lineage. CogMaps make the living system navigable.

Semantic Integrity turns scattered work into visible Cogwork.

The point is not “more AI.” The point is preserving semantic integrity where private work actually happens.

Semantic Integrity Cog Primer v0.1. This document is a explanatory artifact for discussion. Bottom line: “Semantic Integrity makes organizational work legible to humans and AI by turning meaning, action, authority, and evidence into dependency-aware containers.”