Standards Body · Foundational source, public edition · Released July 17, 2026
Canonical record: https://standardsbody.ai/library/foundational-source/taxonomy/

TAXONOMY.md

Standards Body Taxonomy

Project: Standards Body
Primary domain: standardsbody.ai
Core line: Foundations for Frontier AI
Document type: Canonical classification system, entity model, and controlled taxonomic architecture
Version: 1.0
Status: Approved foundational source
Document owner: Standards Body
Applies to: All canonical files, research programs, evaluation protocols, registries, standards proposals, assurance schemes, incident records, contributor systems, public reports, website content, data schemas, interoperability profiles, and future machine-readable infrastructure
Related canonical sources: `PROJECT_IDENTITY.md`, `TERMINOLOGY.md`, `FOUNDATIONS_APPENDIX.md`, `EVIDENCE_STANDARDS.md`, `RESEARCH_METHODOLOGY.md`, `EVALUATION_PHILOSOPHY.md`, `WEBSITE_SOURCE_OF_TRUTH.md`, `VERSION_HISTORY.md`
Review cycle: Annual review, with event-triggered revision after material changes in terminology, evaluation science, assurance practice, standards, law, interoperability, or project identity

---

Authority Note

This document defines the internal canonical taxonomy of Standards Body.

It does not:

- Establish legal classifications
- replace a statutory or regulatory taxonomy
- replace a standard issued by an authorized standards organization
- create certification or accreditation authority
- determine whether an AI system is safe, compliant, or approved
- create a universal ontology for all artificial intelligence
- require external institutions to adopt Standards Body classifications
- make a classification decision binding outside the scope of Standards Body work

Where law, regulation, contract, or an external recognized standard defines a term or classification differently, the external meaning should be preserved and mapped explicitly rather than silently replaced.

Document Purpose

This document establishes the canonical classification architecture of Standards Body.

It is the authoritative source for:

- The major classes of objects Standards Body studies
- The relationships among models, systems, actors, capabilities, risks, evaluations, evidence, safeguards, standards, and institutions
- The distinction between hierarchical classes and cross-cutting facets
- The identifiers used for records and taxonomic entities
- The rules for classifying one object into multiple valid categories
- The handling of ambiguity, uncertainty, overlap, and change
- The structure of future registries and machine-readable data
- The mapping of Standards Body concepts to external vocabularies
- The creation, revision, deprecation, and retirement of taxonomic classes
- The minimum metadata required for consequential records
- The cross-foundation entity model
- The consistency rules governing future standards, protocols, reports, and public claims

A terminology file defines what words mean.

A taxonomy defines how concepts are organized.

An ontology defines entities and the formal relationships among them.

This document performs all three functions at a bounded level:

1. It establishes taxonomic classes.
2. It establishes facets and identifiers.
3. It defines a practical entity-and-relationship model.
4. It provides a path toward future machine-readable ontologies without claiming that the present document is a complete formal ontology.

The taxonomy exists because frontier AI work frequently combines incompatible levels of analysis.

Examples include:

- A model is treated as though it were the full deployed system.
- A capability is treated as though it were a risk.
- A safeguard is classified as a guarantee.
- A review is classified as an audit.
- A certification is classified as regulatory approval.
- A national legal category is treated as a universal technical category.

- A benchmark result is treated as a permanent property of a model family.
- A developer, deployer, provider, evaluator, and standard setter are treated as one actor.
- A public benchmark, held-out task bank, protocol, and evidence package are treated as interchangeable.
- A company policy is classified as operational evidence.
- A high-stakes domain is confused with a high-stakes capability level.
- A result status is omitted, causing expired or superseded evidence to appear current.

The taxonomy prevents these errors by requiring every material object to be classified according to:

- What kind of object it is
- Which layer it belongs to
- Which version and status apply
- Which actors are related to it
- Which capabilities, risks, safeguards, evaluations, and decisions are relevant
- Which jurisdiction, deployment context, and evidence conditions apply
- Which relationships are established and which remain uncertain

The governing taxonomic rule is:

> **Classify the object at the correct level, preserve its context, and do not infer a stronger relationship than the evidence supports.**

Executive Summary

The Standards Body taxonomy is a multi-axis classification system for frontier AI evaluation and institutional infrastructure.

It is not a single tree.

Frontier AI systems are too complex to be represented accurately by one hierarchy.

A single object may be classified simultaneously by:

- Artifact type
- system layer
- lifecycle stage
- access condition
- deployment context
- capability domain
- capability level
- risk domain
- safeguard type
- evaluation method
- evidence level
- assurance function

- institutional role
- standards stage
- jurisdiction
- status
- confidentiality
- interoperability profile

For example, one evaluation record may concern:

- An open-weight model
- embedded in an agentic system
- using external tools
- assessed during pre-deployment
- in the cyber domain
- for autonomous vulnerability exploitation
- under a held-out dynamic protocol
- administered by a third-party evaluator
- independently reviewed
- classified as confidential
- supporting a high-stakes deployment decision
- with evidence level E3
- status current
- expiration in six months

A useful taxonomy must preserve all of those dimensions.

The Standards Body taxonomy therefore uses five structural mechanisms.

1. Classes

Classes answer:

> What kind of thing is this?

Examples:

- AI model
- AI system
- evaluation protocol
- capability
- safeguard
- incident
- standard
- evaluator organization

2. Subclasses

Subclasses answer:

> Which more specific kind is this?

Examples:

- Frontier model
- agentic system
- held-out evaluation
- preventive safeguard
- certification body
- cyber capability

3. Facets

Facets answer:

> Which independent characteristics apply?

Examples:

- Public or confidential
- pre-deployment or post-deployment
- first-party or third-party
- open-weight or closed-weight
- low, material, high, or critical consequence
- current, superseded, or withdrawn

4. Relationships

Relationships answer:

> How is this object connected to another object?

Examples:

- Model is component of system
- system is evaluated by protocol
- protocol contains task family
- result supports claim
- safeguard mitigates risk
- evaluator performs evaluation
- reviewer reviews evidence package
- standard defines requirement
- certificate attests conformity
- accreditation recognizes evaluator competence

5. Status and Version

Status and version answer:

> Which identified state is being described, and is it still valid?

Examples:

- Protocol version 2.1
- model release 2026-07
- certificate active
- evaluation result expired
- standard superseded
- recognition suspended

The taxonomy is organized into twenty primary domains.

1. Project and institutional identity
2. Actors and roles
3. AI artifacts and models
4. AI systems and system components
5. Lifecycle stages
6. Access and release conditions
7. Deployment contexts
8. Capabilities
9. Risks and harms
10. Safeguards and controls
11. Evaluation and testing
12. Evidence and claims
13. Review, audit, and assurance
14. Standards, requirements, and conformity
15. Governance and decision processes
16. Incidents, failures, and corrections
17. Incentives and recognition
18. Interoperability and international coordination
19. Research methods and outputs
20. Versioning, status, security, and records

The taxonomy also defines a common entity model.

Core entities include:

- Actor
- organization
- model
- system
- component
- deployment

- capability
- hazard
- risk
- safeguard
- protocol
- task
- evaluation run
- result
- evidence object
- claim
- review
- evaluator
- requirement
- standard
- certification
- accreditation
- incident
- decision
- jurisdiction
- registry record

Core relationships include:

- `is_a`
- `part_of`
- `derived_from`
- `version_of`
- `deployed_as`
- `evaluated_by`
- `administered_by`
- `reviewed_by`
- `supports`
- `challenges`
- `mitigates`
- `triggers`
- `governed_by`
- `conforms_to`
- `certified_against`
- `accredited_for`
- `recognized_by`
- `supersedes`
- `withdraws`
- `occurs_in`
- `applies_in`

The taxonomy is designed for human and machine use.

Human-readable use includes:

- Writing consistent reports
- selecting correct terminology
- classifying case studies
- creating standards
- avoiding overclaiming

Machine-readable use includes:

- Registries
- interoperable evaluation records
- protocol metadata
- evidence graphs
- incident databases
- evaluator directories
- standards crosswalks

The taxonomy does not require premature formalization.

Every class has a status.

A class may be:

- Preferred
- accepted
- context-specific
- provisional
- deprecated
- retired

Every consequential classification should also include confidence.

Classification confidence may be:

- Confirmed
- high
- moderate
- low
- disputed
- unknown

The taxonomy is intended to evolve.

New classes should be added only when:

- A meaningful distinction exists
- existing classes cannot represent it
- the distinction affects evidence, decisions, governance, or interoperability
- a definition and parent relationship can be stated

- implementation value exceeds complexity

The final rule is:

> **Use hierarchy for identity, facets for context, relationships for meaning, and versioning for time.**

---

1. Taxonomic Design Principles

1.1 Object Before Label

Identify the actual object before selecting its class.

A product name may refer to:

- A model family
- a checkpoint
- an API
- a hosted service
- an agentic application
- a deployment configuration

The taxonomy should classify the actual object, not only the marketed name.

1.2 Correct Level of Analysis

Do not classify:

- A model-level property as a system-level property
- a system-level safeguard as a model property
- an organizational control as technical model behavior
- a legal classification as a scientific fact

1.3 Polyhierarchy

An object may belong to more than one class where the classes describe genuinely different aspects.

Example:

A model may be both:

- General-purpose
- multimodal
- frontier
- open-weight

1.4 Faceted Classification

Independent dimensions should be represented as facets rather than forced into one hierarchy.

1.5 Explicit Relationships

Relationships should be named.

Avoid vague statements such as:

- Related to
- associated with
- relevant to

when a stronger relation is known.

1.6 No Unsupported Inference

If the evidence supports only correlation, do not classify the relation as causal.

If a safeguard reduces one risk, do not classify it as preventing all harms.

1.7 Scope Preservation

Every classification should preserve:

- Object
- version
- date
- context
- jurisdiction
- evidence
- status

1.8 Stable Core, Extensible Edges

The top-level taxonomy should remain stable.

New subclasses and local extensions may evolve.

1.9 Interoperability

Classes should support mapping to external vocabularies without erasing legitimate differences.

1.10 Public Legibility

Class names should be understandable to informed non-specialists.

1.11 Machine Readability

Identifiers and relationships should support future structured data.

1.12 No Prestige Classification

Labels such as *frontier*, *independent*, *accredited*, *official*, and *international* should not be assigned for reputational benefit.

1.13 Revision and Retirement

Obsolete classes should be deprecated or retired visibly.

1.14 Bounded Completeness

The taxonomy should cover the project domain sufficiently.

It should not attempt to classify every concept in computer science, law, ethics, or public policy.

---

2. Taxonomic Structure

2.1 Primary Class

A broad category of entity.

Example:

`EVAL.EVALUATION_OBJECT`

2.2 Subclass

A narrower class inheriting the relevant meaning of its parent.

Example:

`EVAL.HELD_OUT_EVALUATION`

2.3 Facet

A cross-cutting attribute that may apply to many classes.

Example:

`ACCESS.PUBLIC`

2.4 Property

A value associated with an entity.

Example:

- version
- date
- owner
- status
- jurisdiction

2.5 Relationship

A typed connection between two entities.

Example:

`SYSTEM evaluated_by PROTOCOL`

2.6 Record

A structured representation of an entity and its metadata.

2.7 Profile

A defined bundle of required classes, facets, properties, and relationships for a use case.

Example:

- Evaluation Result Profile
- Evaluator Organization Profile
- Interoperability Profile

2.8 Scheme

A governed set of classes and rules used for assessment, certification, recognition, or reporting.

2.9 Local Extension

A jurisdictional, domain-specific, linguistic, or institutional addition that preserves mapping to the common core.

2.10 Crosswalk

A structured mapping between this taxonomy and another taxonomy, standard, law, or data model.

---

3. Identifier Architecture

3.1 Taxonomy Class Identifier

Preferred pattern:

`DOMAIN.CLASS_NAME`

Examples:

- `AI.MODEL`
- `SYS.AGENTIC_SYSTEM`
- `CAP.CYBER`
- `RISK.SYSTEMIC`
- `SAFEGUARD.ACCESS_CONTROL`
- `EVAL.HELD_OUT`
- `EVIDENCE.DIRECT`
- `ASSURANCE.CERTIFICATION`
- `STATUS.SUPERSEDED`

3.2 Entity Record Identifier

Preferred pattern:

`SB-[ENTITY]-[YEAR]-[SEQUENCE]`

Examples:

- `SB-MODEL-2026-0001`
- `SB-PROTOCOL-2026-0012`
- `SB-RESULT-2026-0044`
- `SB-INCIDENT-2026-0007`

3.3 Version Identifier

Preferred semantic structure:

`MAJOR.MINOR.PATCH`

Example:

`2.1.0`

3.4 Relationship Identifier

Preferred lowercase verb phrase:

- `is_a`
- `part_of`
- `version_of`
- `derived_from`
- `evaluated_by`
- `supports_claim`

3.5 Human-Readable Label

Every identifier should have a plain-language label.

3.6 Persistent Identifier

Once published, an identifier should not be reassigned to a different concept.

3.7 Deprecated Identifier

Deprecated identifiers should remain resolvable and point to the replacement.

---

4. Taxonomic Status

4.1 Preferred

Default Standards Body class.

4.2 Accepted

Permitted but not preferred.

4.3 Context-Specific

Valid only under a defined legal, technical, sectoral, or institutional context.

4.4 Provisional

Under active research and subject to material revision.

4.5 Deprecated

Still visible for transition but replaced.

4.6 Retired

No longer used for current classification.

4.7 External

Imported from another authoritative taxonomy and preserved with source context.

---

5. Classification Confidence

5.1 Confirmed

The class assignment is directly supported and unambiguous.

5.2 High

Strong evidence supports the assignment with minor uncertainty.

5.3 Moderate

The assignment is reasonable, but alternatives remain.

5.4 Low

Evidence is limited or ambiguous.

5.5 Disputed

Qualified parties materially disagree.

5.6 Unknown

No reliable classification can be made.

5.7 Confidence Rule

Do not convert low-confidence classifications into precise requirements without review.

---

6. Primary Domain Map

Domain code	Domain
ID	Project and institutional identity
ACTOR	Actors and roles
AI	AI artifacts and models
SYS	AI systems and components
LIFE	Lifecycle
ACCESS	Access and release
DEPLOY	Deployment context
CAP	Capability
RISK	Risk and harm
SAFEGUARD	Safeguard and control
EVAL	Evaluation and testing
EVIDENCE	Evidence and claims
REVIEW	Review
ASSURANCE	Audit, certification, accreditation, and assurance
STANDARD	Standards and requirements
GOV	Governance and decisions
INCIDENT	Incidents, failures, and corrections
INCENTIVE	Incentives and recognition
INTEROP	Interoperability and international coordination
RESEARCH	Research methods and outputs
STATUS	Versioning, status, and records
SECURITY	Information classification and research security
JURIS	Jurisdiction and legal scope

---

7. Project and Institutional Identity Taxonomy

7.1 `ID.PROJECT`

A bounded organized effort with a defined purpose, owner, scope, and outputs.

Subclasses

- `ID.RESEARCH_PROJECT`
- `ID.INSTITUTIONAL_DESIGN_PROJECT`

- ID.STANDARDS_PROJECT
- ID.PILOT_PROJECT
- ID.INFRASTRUCTURE_PROJECT

7.2 ID.ORGANIZATION

A legally, administratively, or functionally organized body.

Subclasses

- ID.PUBLIC_ORGANIZATION
- ID.PRIVATE_ORGANIZATION
- ID.NONPROFIT_ORGANIZATION
- ID.ACADEMIC_ORGANIZATION
- ID.MULTILATERAL_ORGANIZATION
- ID.CONSORTIUM
- ID.COMMUNITY_ORGANIZATION
- ID.INFORMAL_NETWORK

7.3 ID.INSTITUTION

An enduring system of authority, rules, roles, incentives, and practices.

An institution may be embodied in one organization or distributed across several organizations.

7.4 ID.STANDARDS_BODY

An organization or institution that develops and maintains standards through a defined process.

Classification warning: The proper name *Standards Body* refers to this project. It does not automatically classify the project as a formally recognized standards body.

7.5 ID.REGULATOR

A legally authorized public institution that makes, administers, or enforces regulatory requirements.

7.6 ID.ACCREDITATION_BODY

A body that performs accreditation.

7.7 ID.CERTIFICATION_BODY

A third-party body that performs certification under a defined scheme.

7.8 `ID.EVALUATION_ORGANIZATION`

An organization that designs, administers, scores, or interprets evaluations.

7.9 `ID.RESEARCH_INSTITUTION`

An institution whose primary function includes systematic knowledge creation.

7.10 `ID.SCHEME_OWNER`

An organization responsible for the rules and governance of an assurance, certification, recognition, or reporting scheme.

7.11 Institutional Facets

Classify institutions by:

Authority

- `ID.AUTHORITY.NONE`
- `ID.AUTHORITY.RESEARCH`
- `ID.AUTHORITY.CONTRACTUAL`
- `ID.AUTHORITY.PROFESSIONAL`
- `ID.AUTHORITY.STANDARDS_PROCESS`
- `ID.AUTHORITY.ACCREDITATION`
- `ID.AUTHORITY.REGULATORY`
- `ID.AUTHORITY.JUDICIAL`

Geographic Scope

- Local
- subnational
- national
- regional
- international
- global participation, without implied universal authority

Ownership

- Public
- private
- nonprofit
- member-owned
- hybrid
- distributed

Functional Stage

- Conceptual
- pilot
- operational
- recognized
- statutory
- retired

---

8. Actor and Role Taxonomy

8.1 `ACTOR.PERSON`

An individual human participant.

8.2 `ACTOR.ORGANIZATION`

An organization acting in a defined role.

8.3 `ACTOR.DEVELOPER`

An actor that designs, trains, fine-tunes, or materially modifies an AI model or system.

Subclasses

- `ACTOR.BASE_MODEL_DEVELOPER`
- `ACTOR.FINE_TUNER`
- `ACTOR.SYSTEM_DEVELOPER`
- `ACTOR.APPLICATION_DEVELOPER`
- `ACTOR.SCAFFOLD_DEVELOPER`
- `ACTOR.OPEN_COMMUNITY_DEVELOPER`

8.4 `ACTOR.PROVIDER`

An actor that makes an AI model or system available to others.

Subclasses

- `ACTOR.MODEL_PROVIDER`
- `ACTOR.API_PROVIDER`
- `ACTOR.PLATFORM_PROVIDER`
- `ACTOR.INFRASTRUCTURE_PROVIDER`
- `ACTOR.TOOL_PROVIDER`

8.5 ACTOR.DEPLOYER

An actor that places an AI system into operational use.

8.6 ACTOR.OPERATOR

An actor responsible for day-to-day control or use of a deployed system.

8.7 ACTOR.USER

A person or organization interacting with the system.

Subclasses

- General user
- professional user
- expert user
- administrator
- developer user
- malicious user
- researcher
- affected non-user

8.8 ACTOR.AFFECTED_PARTY

A person, group, institution, community, or system materially affected by an AI deployment or institutional decision.

8.9 ACTOR.EVALUATOR

An actor that performs evaluation activities.

Subclasses

- `ACTOR.FIRST_PARTY_EVALUATOR`
- `ACTOR.SECOND_PARTY_EVALUATOR`
- `ACTOR.THIRD_PARTY_EVALUATOR`
- `ACTOR.PUBLIC_EVALUATOR`
- `ACTOR.ACADEMIC_EVALUATOR`
- `ACTOR.COMMUNITY_EVALUATOR`
- `ACTOR.DOMAIN_EVALUATOR`

8.10 ACTOR.REVIEWER

An actor that reviews methods, evidence, findings, or institutional processes.

Subclasses

- Peer reviewer
- domain reviewer
- methodological reviewer
- statistical reviewer
- security reviewer
- ethics reviewer
- legal reviewer
- public-interest reviewer
- independent expert reviewer

8.11 `ACTOR.AUDITOR`

An actor qualified and assigned to conduct an audit against defined criteria.

8.12 `ACTOR.INSPECTOR`

An actor conducting inspection.

8.13 `ACTOR.CERTIFIER`

An actor or body responsible for a certification decision.

8.14 `ACTOR.ACCREDITOR`

An actor or body responsible for an accreditation decision.

8.15 `ACTOR.STANDARD_SETTER`

An actor participating in standards development.

Subclasses

- Committee member
- working-group member
- secretariat
- chair
- public commenter
- technical editor
- liaison

8.16 `ACTOR.DECISION_OWNER`

An actor with authority to make the decision the evidence informs.

8.17 `ACTOR.SPONSOR`

An actor that funds, commissions, or formally requests work.

8.18 `ACTOR.CUSTODIAN`

An actor responsible for protected evidence, tasks, records, or registries.

8.19 `ACTOR.REGISTRY_OPERATOR`

An actor maintaining a registry.

8.20 `ACTOR.WHISTLEBLOWER`

A person disclosing suspected wrongdoing, risk, failure, or concealment through a protected or public channel.

8.21 `ACTOR.THREAT_ACTOR`

A person, group, organization, or state capable of intentionally causing harm.

Threat-Actor Facets

- Skill level
- resources
- access
- persistence
- coordination
- intent
- legal authority
- geographic location

8.22 Role Relationship Rules

One organization may hold several roles.

Example:

A developer may also be:

- Provider
- deployer
- first-party evaluator
- sponsor

The record should preserve each role separately because conflicts and responsibilities differ by role.

---

9. AI Artifact and Model Taxonomy

9.1 `AI.ARTIFACT`

A technical object created, trained, configured, or used in an AI system.

9.2 `AI.MODEL`

A computational model producing outputs from inputs through learned parameters, structured logic, or both.

Model Scope Subclasses

- `AI.BASE_MODEL`
- `AI.FOUNDATION_MODEL`
- `AI.GENERAL_PURPOSE_MODEL`
- `AI.DOMAIN_SPECIFIC_MODEL`
- `AI.TASK_SPECIFIC_MODEL`
- `AI.FRONTIER_MODEL`
- `AI.EMBEDDED_MODEL`
- `AI.ENSEMBLE_MODEL`

Model Modality Subclasses

- `AI.TEXT_MODEL`
- `AI.IMAGE_MODEL`
- `AI.AUDIO_MODEL`
- `AI.VIDEO_MODEL`
- `AI.MULTIMODAL_MODEL`
- `AI.ROBOTICS_MODEL`
- `AI.SCIENTIFIC_MODEL`
- `AI.CODE_MODEL`

Model Function Subclasses

- Generative
- predictive
- classificatory
- ranking
- recommendation
- control
- planning

- representation
- detection
- optimization

9.3 AI.MODEL_FAMILY

A set of related models sharing lineage, architecture, training approach, or product identity.

9.4 AI.MODEL_VERSION

A distinct identified release, checkpoint, or deployed state.

9.5 AI.CHECKPOINT

A saved model-parameter state.

9.6 AI.WEIGHTS

The learned parameters of a model.

9.7 AI.ADAPTER

A separately identifiable learned component modifying model behavior.

Examples:

- Low-rank adapter
- domain adapter
- safety adapter

9.8 AI.TOKENIZER

A component mapping inputs into model-readable units.

9.9 AI.TRAINING_DATASET

A dataset used in pretraining, fine-tuning, alignment, or adaptation.

Data Facets

- Public
- licensed
- proprietary
- synthetic
- generated

- personal
- sensitive
- unknown provenance
- mixed

9.10 `AI.TRAINING_RUN`

A bounded training process producing or modifying a model.

9.11 `AI.POST_TRAINING_PROCESS`

A process after base training.

Subclasses

- Instruction tuning
- preference optimization
- reinforcement learning
- safety tuning
- domain adaptation
- distillation
- compression
- quantization

9.12 `AI.MODEL_CARD`

A documentation artifact describing a model.

9.13 `AI.SYSTEM_CARD`

A documentation artifact describing a system, evaluation, risk, or deployment.

9.14 Model Access Facets

- `ACCESS.WEIGHTS_PUBLIC`
- `ACCESS.WEIGHTS_CONTROLLED`
- `ACCESS.WEIGHTS_PRIVATE`
- `ACCESS.API_PUBLIC`
- `ACCESS.API_RESTRICTED`
- `ACCESS.INTERNAL_ONLY`

9.15 Model Scale Facets

Use only where relevant and defined:

- Small
- medium
- large
- frontier-scale
- unknown

The taxonomy should not infer capability directly from scale.

9.16 Frontier Classification

A model may be classified as frontier only with:

- Defined comparison basis
- relevant capability or scale dimension
- date
- evidence
- confidence

Frontier status is time-dependent.

---

10. AI System and Component Taxonomy

10.1 `SYS.AI_SYSTEM`

The complete operational arrangement through which one or more AI models are configured, accessed, integrated, monitored, and used.

10.2 `SYS.MODEL_COMPONENT`

A model acting as one component in a larger system.

10.3 `SYS.SYSTEM_PROMPT`

A high-priority instruction or contextual component.

10.4 `SYS.SCAFFOLD`

Software, prompts, memory, planning, tools, or control structure added around a model.

Subclasses

- Prompt scaffold
- planning scaffold
- coding scaffold
- retrieval scaffold
- multi-agent scaffold
- verification scaffold
- workflow scaffold

10.5 SYS.AGENT

A system or configuration selecting and executing actions over multiple steps toward an objective.

10.6 SYS.AGENTIC_SYSTEM

A system exhibiting planning, action, observation, adaptation, or persistence across multiple steps.

Agentic Facets

- Single-agent
- multi-agent
- human-supervised
- autonomous
- episodic
- persistent
- tool-using
- environment-embedded

10.7 SYS.MULTI_AGENT_SYSTEM

A system containing multiple interacting agents.

10.8 SYS.RETRIEVAL_COMPONENT

A component retrieving external information.

10.9 SYS.MEMORY_COMPONENT

A component preserving information across turns, sessions, tasks, or users.

Memory Facets

- Session memory
- persistent memory
- user-specific memory

- shared memory
- vector memory
- symbolic memory

10.10 SYS.TOOL

An external software, API, instrument, service, or environment accessible to the system.

Tool Subclasses

- Search
- code execution
- file system
- communication
- database
- financial
- cyber
- laboratory
- robotic
- productivity
- administrative

10.11 SYS.ORCHESTRATOR

A component coordinating models, tools, agents, or workflows.

10.12 SYS.INTERFACE

The channel through which users or systems interact.

Interface Subclasses

- Chat
- API
- embedded application
- autonomous process
- voice
- robotics
- batch
- developer tool

10.13 SYS.MONITORING_COMPONENT

A component observing system behavior, access, or outputs.

10.14 SYS.SAFETY_COMPONENT

A system component implementing a safeguard.

10.15 SYS.HUMAN_COMPONENT

A human role embedded in system operation.

Examples:

- Approver
- supervisor
- operator
- reviewer
- escalation authority

10.16 SYS.DEPLOYMENT_CONFIGURATION

A versioned operational configuration.

10.17 SYS.SYSTEM_MANIFEST

A structured record of system components and versions.

10.18 System Relationship Rules

- A model `part_of` a system.
- A system `uses` a tool.
- A system `implements` a safeguard.
- A system `deployed_as` a deployment.
- A system `version_of` a prior system state.
- A system `derived_from` a development lineage.

10.19 System Identity Minimum

A consequential system record should include:

- Model identifier
- system version
- prompts or prompt status
- tools
- retrieval
- memory
- scaffolds
- safeguards
- access tier

- deployment context
- date

---

11. Lifecycle Taxonomy

11.1 `LIFE.CONCEPT`

Idea or early concept stage.

11.2 `LIFE.RESEARCH`

Research and feasibility stage.

11.3 `LIFE.DEVELOPMENT`

Model or system development.

11.4 `LIFE.TRAINING`

Base or subsequent training.

11.5 `LIFE.POST_TRAINING`

Post-training modification.

11.6 `LIFE.INTERNAL_TESTING`

Internal testing before external use.

11.7 `LIFE.EXTERNAL_EVALUATION`

Evaluation by an outside or independent actor.

11.8 `LIFE.PRE_DEPLOYMENT`

Final assessment and governance before operational release.

11.9 `LIFE.PILOT_DEPLOYMENT`

Bounded operational use.

11.10 `LIFE.GENERAL_DEPLOYMENT`

Broad operational use.

11.11 `LIFE.MONITORING`

Ongoing operational observation.

11.12 `LIFE.UPDATE`

Material model or system change.

11.13 `LIFE.SUSPENSION`

Temporary halt to use or access.

11.14 `LIFE.ROLLBACK`

Reversion to a prior version.

11.15 `LIFE.WITHDRAWAL`

Formal removal from use or availability.

11.16 `LIFE.RETIREMENT`

Planned end of active life.

11.17 `LIFE.ARCHIVE`

Historical preservation.

11.18 Lifecycle Relationship Rule

Every evaluation result should identify the lifecycle stage at which it was produced.

A pre-deployment result should not be treated as post-deployment evidence without justification.

---

12. Access and Release Taxonomy

12.1 `ACCESS.PUBLIC`

Available to the general public.

12.2 `ACCESS.OPEN`

Available under defined open-use, open-source, or open-weight conditions.

12.3 `ACCESS.REGISTERED`

Available after registration.

12.4 `ACCESS.VERIFIED`

Available to identity-verified users.

12.5 `ACCESS.TIERED`

Different capabilities are available at different access levels.

12.6 `ACCESS.CONTROLLED`

Available under contractual, institutional, security, or research controls.

12.7 `ACCESS.RESTRICTED`

Available only to a narrow authorized group.

12.8 `ACCESS.INTERNAL`

Available only within the responsible organization.

12.9 `ACCESS.EMBARGOED`

Temporarily unavailable pending a release date or condition.

12.10 `ACCESS.REVOKED`

Previously available access has been withdrawn.

12.11 Release Object Facets

Classify what is released:

- Weights
- source code
- API
- training data
- technical report
- evaluation tasks
- system prompt
- research artifacts
- deployment service

12.12 Permission Facets

- Read
- query
- fine-tune
- modify
- redistribute
- commercial use
- tool use
- autonomous use
- high-volume use

12.13 Access Risk Facets

- Identity requirement
- rate limit
- use restriction
- monitoring
- geographic restriction
- domain restriction
- capability restriction
- revocation support

---

13. Deployment Context Taxonomy

13.1 `DEPLOY.INTERNAL`

Use within the developing or deploying organization.

13.2 DEPLOY.CONSUMER

Use by general consumers.

13.3 DEPLOY.ENTERPRISE

Use in organizational operations.

13.4 DEPLOY.PROFESSIONAL

Use by qualified professionals.

13.5 DEPLOY.PUBLIC_SECTOR

Use by government or public bodies.

13.6 DEPLOY.CRITICAL_INFRASTRUCTURE

Use in systems whose disruption may create severe societal consequence.

13.7 DEPLOY.RESEARCH

Use for scientific or technical research.

13.8 DEPLOY.EDUCATION

Use in educational contexts.

13.9 DEPLOY.HEALTH

Use in health or medical contexts.

13.10 DEPLOY.FINANCE

Use in financial systems or decisions.

13.11 DEPLOY.CYBERSECURITY

Use in defensive or offensive cybersecurity contexts.

13.12 DEPLOY.LEGAL

Use in legal services, adjudication support, or legal administration.

13.13 DEPLOY.LABORATORY

Use with scientific instruments, biological systems, chemical systems, or physical experiments.

13.14 DEPLOY.ROBOTICS

Use in embodied systems.

13.15 DEPLOY.COMMUNICATION

Use for media, persuasion, or public communication.

13.16 Deployment Scale Facets

- Individual
- team
- organization
- sector
- population
- cross-border
- infrastructure-wide

13.17 Decision Authority Facets

- Advisory
- recommendation
- human-approved action
- delegated action
- autonomous action
- binding decision support

13.18 Temporal Facets

- Episodic
- continuous
- real-time
- delayed
- persistent

13.19 Human Oversight Facets

- Human in the loop
- human on the loop
- human in command

- no active human oversight
- unknown

---

14. Capability Taxonomy

14.1 `CAP.CAPABILITY`

The ability of a model or system to perform a defined task or class of tasks under specified conditions.

14.2 Capability Object Facets

- Model capability
- system capability
- human-AI team capability
- organizational capability
- effective deployed capability
- latent capability
- demonstrated capability

14.3 General Capability Domains

`CAP.REASONING`

- Logical reasoning
- mathematical reasoning
- causal reasoning
- spatial reasoning
- strategic reasoning
- scientific reasoning

`CAP.KNOWLEDGE`

- Factual recall
- domain knowledge
- procedural knowledge
- situational knowledge

`CAP.LANGUAGE`

- Comprehension
- generation
- translation
- summarization
- dialogue

- rhetorical adaptation

CAP.CODE

- Code generation
- debugging
- architecture
- repository navigation
- software maintenance
- exploit development

CAP.PLANNING

- Goal decomposition
- sequencing
- scheduling
- resource allocation
- contingency planning

CAP.TOOL_USE

- Search
- code execution
- database use
- API use
- laboratory tool use
- robotic control

CAP.LEARNING

- In-context learning
- adaptation
- memory-based improvement
- self-correction
- skill acquisition

CAP.AUTONOMY

- Independent action selection
- persistence
- recovery
- long-horizon execution
- delegation
- self-monitoring

CAP.SOCIAL

- Persuasion
- negotiation

- coordination
- deception
- emotion recognition
- role adaptation

CAP.CREATIVE

- Ideation
- design
- artistic production
- scientific hypothesis generation
- invention

14.4 High-Stakes Capability Domains

CAP.CYBER

Subclasses:

- Reconnaissance
- vulnerability discovery
- exploit development
- credential attack
- lateral movement
- persistence
- evasion
- autonomous cyber operations
- defensive remediation

CAP.BIOLOGICAL

Subclasses:

- Biological knowledge
- protocol design
- experimental troubleshooting
- pathogen-related reasoning
- synthesis planning
- laboratory automation
- safety analysis

CAP.CHEMICAL

Subclasses:

- Chemical synthesis planning
- hazardous-material reasoning
- laboratory optimization

- safety and containment analysis

CAP.AUTONOMOUS_REPLICATION

Subclasses:

- Resource acquisition
- infrastructure deployment
- service replication
- persistence
- identity creation
- coordination

CAP.AI_RESEARCH_AND_DEVELOPMENT

Subclasses:

- Algorithm design
- experiment design
- code implementation
- training optimization
- evaluation automation
- model improvement research

CAP.PERSUASION_AND_MANIPULATION

Subclasses:

- Targeted persuasion
- mass persuasion
- personalized influence
- deception
- coercive interaction
- social engineering

CAP.CRITICAL_INFRASTRUCTURE

Subclasses:

- Control-system analysis
- operational planning
- failure exploitation
- resilience support

CAP.FINANCIAL_AND_ECONOMIC

Subclasses:

- Trading
- fraud

- market manipulation
- financial planning
- economic forecasting
- automated enterprise operation

`CAP.SCIENTIFIC_AND_ENGINEERING`

Subclasses:

- Hypothesis generation
- simulation
- experimental design
- engineering design
- materials discovery
- laboratory operation

14.5 Capability Dimensions

Every capability classification may include:

- Accuracy
- reliability
- speed
- cost
- autonomy
- generalization
- transfer
- stealth
- scalability
- resource need
- tool dependence
- human uplift
- recoverability
- task horizon

14.6 Capability Level

A capability level should be protocol-defined.

Generic levels may include:

- Not demonstrated
- basic
- intermediate
- advanced
- expert-comparable
- beyond reference group

- critical threshold

These labels should not be used without task, reference, and evidence context.

14.7 Capability Relationship Rules

- Capability `demonstrated_by` result.
- Capability `enabled_by` system component.
- Capability `contributes_to` risk.
- Capability `constrained_by` safeguard.
- Capability `compared_with` human baseline.
- Capability `requires` resource or access.

---

15. Risk and Harm Taxonomy

15.1 `RISK.HAZARD`

A source or condition with potential to cause harm.

15.2 `RISK.THREAT`

A potential cause of an unwanted incident.

15.3 `RISK.VULNERABILITY`

A weakness that may be exploited or activated.

15.4 `RISK.EXPOSURE`

The degree to which an actor, system, asset, or population is subject to a hazard.

15.5 `RISK.EVENT`

An occurrence connecting hazard, vulnerability, and consequence.

15.6 `RISK.CONSEQUENCE`

The outcome or effect of an event.

15.7 `RISK.RISK`

The combination of likelihood and consequence under a defined context.

15.8 Risk Origin

- Intentional misuse
- accidental misuse
- system failure
- organizational failure
- emergent interaction
- security compromise
- market failure
- governance failure
- systemic propagation

15.9 Risk Domain

RISK.PHYSICAL_HARM

RISK.PSYCHOLOGICAL_HARM

RISK.ECONOMIC_HARM

RISK.CIVIL_RIGHTS_HARM

RISK.PRIVACY_HARM

RISK.SECURITY_HARM

RISK.INFORMATION_HARM

RISK.ENVIRONMENTAL_HARM

RISK.INSTITUTIONAL_HARM

RISK.DEMOCRATIC_HARM

RISK.SYSTEMIC_HARM

RISK.CATASTROPHIC_HARM

RISK.EXISTENTIAL_HARM

15.10 Risk Actor Model

- Unskilled user
- skilled individual
- organized criminal group
- insider
- corporation
- state actor
- automated system

- multiple interacting actors

15.11 Risk Temporal Horizon

- Immediate
- near-term
- medium-term
- long-term
- persistent
- intergenerational

15.12 Risk Scale

- Individual
- household
- organization
- community
- sector
- national
- international
- global

15.13 Risk Reversibility

- Reversible
- partly reversible
- difficult to reverse
- irreversible
- unknown

15.14 Risk Uncertainty

- Known and measured
- known but weakly measured
- plausible
- speculative
- unknown

15.15 Systemic Risk Subclasses

- Cascading technical failure
- correlated decision failure
- market concentration
- institutional dependency

- information ecosystem degradation
- infrastructure coupling
- cross-border propagation

15.16 Risk Relationship Rules

- Hazard `may_cause` harm.
- Vulnerability `enables` threat pathway.
- Safeguard `mitigates` risk.
- Capability `increases_or_decreases` risk under context.
- Incident `realizes` risk.
- Evidence `supports` risk claim.
- Decision `accepts`, `reduces`, `transfers`, or `avoids` risk.

---

16. Safeguard and Control Taxonomy

16.1 `SAFEGUARD.SAFEGUARD`

A technical, procedural, organizational, contractual, or institutional measure intended to reduce risk.

16.2 Safeguard Function

`SAFEGUARD.PREVENTIVE`

Prevents or reduces the probability of an event.

`SAFEGUARD.DETECTIVE`

Detects an event, anomaly, or failure.

`SAFEGUARD.CORRECTIVE`

Contains, remediates, or recovers.

`SAFEGUARD.COMPENSATING`

Provides an alternative where a primary control is unavailable or insufficient.

`SAFEGUARD.DETERRENT`

Changes incentives to discourage harmful behavior.

16.3 Safeguard Layer

Model-Level

- Training intervention
- refusal behavior
- policy conditioning
- adversarial robustness
- capability limitation

System-Level

- Tool restriction
- monitoring
- rate limit
- authentication
- sandbox
- output filter
- approval workflow

Deployment-Level

- User eligibility
- domain restriction
- geographic restriction
- usage policy
- logging
- escalation

Organizational

- Governance
- personnel controls
- incident response
- training
- access review
- separation of duties

Contractual

- Terms of use
- liability
- audit right
- data-use restriction
- disclosure obligation

Institutional

- Standard

- certification
- accreditation
- licensing
- regulatory requirement
- registry
- mutual recognition

16.4 Safeguard Mechanism

- Access control
- identity verification
- least privilege
- rate limiting
- monitoring
- anomaly detection
- human approval
- tool isolation
- secure enclave
- redaction
- refusal
- content moderation
- logging
- rollback
- deployment suspension
- incident reporting
- evaluation requirement
- independent review

16.5 Safeguard Effectiveness Facets

- Coverage
- efficacy
- robustness
- bypass resistance
- adaptive resistance
- latency
- reliability
- cost
- usability
- scalability
- residual risk
- monitored status

16.6 Safeguard Relationship Rules

- Safeguard `implemented_in` system.
- Safeguard `mitigates` risk.
- Safeguard `tested_by` evaluation.
- Safeguard `required_by` standard.
- Safeguard `fails_in` incident.
- Safeguard `superseded_by` improved control.

---

17. Evaluation and Testing Taxonomy

17.1 `EVAL.EVALUATION`

A structured process for producing and interpreting evidence about an object or claim.

17.2 Evaluation Object

- Model
- system
- safeguard
- organization
- process
- protocol
- evaluator
- standard
- incident response
- deployment

17.3 Evaluation Purpose

- Capability measurement
- safeguard assessment
- risk assessment
- compliance assessment
- conformity assessment
- research
- procurement
- deployment decision
- monitoring
- incident investigation
- accreditation support

17.4 Evaluation Timing

- Development
- pre-deployment
- deployment
- post-deployment
- continuous
- incident-triggered
- scheduled reassessment

17.5 Evaluation Party

- First-party
- second-party
- third-party
- regulator
- academic
- public-interest
- community

17.6 Evaluation Openness

- Public
- open task
- partially protected
- held-out
- confidential
- restricted
- federated

17.7 `EVAL.TEST`

A defined procedure used to observe or measure one or more characteristics.

17.8 `EVAL.BENCHMARK`

A standardized task set, procedure, and metric used for comparison.

Benchmark Subclasses

- Public benchmark
- private benchmark
- live benchmark
- dynamic benchmark
- domain benchmark

- agent benchmark
- human-uplift benchmark

17.9 EVAL.PROTOCOL

The complete versioned specification governing evaluation.

17.10 EVAL.TASK

A defined activity assigned to a model or system.

17.11 EVAL.TASK_FAMILY

A group of related tasks.

17.12 EVAL.SCENARIO

A structured contextual situation for evaluation.

17.13 EVAL.ENVIRONMENT

The setting in which evaluation occurs.

17.14 EVAL.HARNESS

Software and infrastructure administering the evaluation.

17.15 EVAL.RUN

One execution of an evaluation or part of it.

17.16 EVAL.TRIAL

One attempt at a task or item.

17.17 EVAL.RESULT

A scored or interpreted output of an evaluation.

17.18 Evaluation Method

- Static evaluation
- dynamic evaluation

- held-out evaluation
- adversarial evaluation
- red-team evaluation
- simulation
- sandbox evaluation
- real-world evaluation
- human baseline
- human-uplift study
- observational monitoring
- formal verification
- stress test
- penetration test
- tabletop exercise

17.19 Elicitation Facet

- Zero-shot
- few-shot
- prompt optimized
- tool augmented
- scaffolded
- fine-tuned
- human assisted
- best effort
- standardized
- practical deployment

17.20 Scoring Facet

- Exact
- rule-based
- environment-based
- human rubric
- model judge
- hybrid
- expert adjudication
- probabilistic
- ordinal

17.21 Integrity Facet

- Unexposed
- exposure uncertain
- partially exposed
- contaminated

- compromised
- retired

17.22 Evaluation Relationship Rules

- Protocol `contains` task family.
- Task family `samples` task universe.
- Evaluation `uses` protocol.
- Run `evaluates` system.
- Result `generated_by` run.
- Result `supports` claim.
- Reviewer `reviews` result.
- Result `expires_on` date or trigger.

---

18. Evidence and Claim Taxonomy

18.1 `EVIDENCE.CLAIM`

A proposition asserted to be true, supported, justified, or sufficiently reliable for a defined purpose.

Claim Type

- Descriptive
- comparative
- causal
- predictive
- capability
- absence
- safeguard
- safety
- security
- reliability
- organizational
- legal
- standards
- compliance
- certification
- accreditation
- policy
- historical
- consensus
- reputational

18.2 `EVIDENCE.EVIDENCE_OBJECT`

An information object relevant to supporting or challenging a claim.

18.3 Evidence Directness

- `EVIDENCE.DIRECT`
- `EVIDENCE.INDIRECT`
- `EVIDENCE.CIRCUMSTANTIAL`
- `EVIDENCE.TESTIMONIAL`
- `EVIDENCE.DERIVED`

18.4 Evidence Source Type

- Primary
- secondary
- tertiary
- generated
- anonymous
- personal communication
- official record
- technical artifact
- expert judgment
- operational record
- incident record

18.5 Evidence Form

- Document
- data
- code
- log
- model output
- image
- video
- audio
- physical artifact
- testimony
- observation
- statistical estimate
- certificate
- registry record
- legal text
- standard
- review finding

18.6 Evidence Level

Use the canonical evidence levels from `EVIDENCE_STANDARDS.md`.

- `EVIDENCE.LEVEL.E0_UNSUPPORTED`
- `EVIDENCE.LEVEL.E1_PRELIMINARY`
- `EVIDENCE.LEVEL.E2_SUPPORTED`
- `EVIDENCE.LEVEL.E3_SUBSTANTIATED`
- `EVIDENCE.LEVEL.E4_DECISION_GRADE`

18.7 Confidence

- Very low
- low
- moderate
- high
- very high

18.8 Evidence Quality Facets

- Relevance
- directness
- construct validity
- internal validity
- external validity
- reliability
- provenance
- authenticity
- completeness
- independence
- recency
- representativeness
- reproducibility
- replicability
- security
- uncertainty
- decision relevance
- conflict exposure
- gaming susceptibility

18.9 Evidence Status

- Current
- preliminary
- disputed

- corrected
- superseded
- withdrawn
- expired
- inaccessible
- compromised

18.10 Claim Status

- Proposed
- under review
- supported
- substantiated
- decision-grade
- partially substantiated
- not substantiated
- inconclusive
- disputed
- corrected
- withdrawn
- retired

18.11 Evidence Direction

- Supports
- challenges
- narrows
- contextualizes
- supersedes
- invalidates
- does not resolve

18.12 Evidence Relationship Rules

- Evidence object `supports` claim.
- Evidence object `challenges` claim.
- Claim `concerns` entity.
- Claim `bounded_by` scope.
- Claim `used_for` decision.
- Claim `has_evidence_level` evidence level.
- Claim `has_confidence` confidence.
- Evidence object `derived_from` source.
- Evidence object `reviewed_by` reviewer.

---

19. Review Taxonomy

19.1 `REVIEW.REVIEW`

A structured examination of evidence, methods, reasoning, process, or claims.

19.2 Review Relationship

- Internal
- external
- independent
- peer
- second-party
- public
- regulatory

19.3 Review Subject

- Protocol
- task bank
- model
- system
- evaluation result
- evidence package
- standard
- organization
- safeguard
- incident
- decision

19.4 Review Discipline

- Technical
- methodological
- statistical
- domain
- security
- ethics
- legal
- governance
- public interest
- economic
- interoperability

19.5 Review Depth

- Screening
- limited review
- full review
- deep technical review
- continuous review
- re-review
- meta-review

19.6 `REVIEW.MANDATE`

A documented review question, scope, authority, access, methods, outputs, constraints, and decision relationship.

19.7 `REVIEW.FINDING`

A supported conclusion produced through review.

19.8 `REVIEW.DISSENT`

A reasoned disagreement with a primary or majority conclusion.

19.9 `REVIEW.MINORITY_REPORT`

A formal documented dissent.

19.10 `REVIEW.FACTUAL_CORRECTION`

Review focused on factual or configuration accuracy.

19.11 Review Access Facet

- Public-only
- document access
- raw evidence access
- system access
- weight access
- personnel access
- restricted access
- insufficient access

19.12 Review Independence Facet

- Organizational
- financial
- methodological
- informational
- operational
- publication
- intellectual
- political
- security

19.13 Review Outcome

- Approve
- approve with conditions
- revise
- defer
- narrow claim
- restrict publication
- reject
- withdraw
- inconclusive

---

20. Audit, Assurance, Certification, and Accreditation Taxonomy

20.1 `ASSURANCE.ASSURANCE`

Evidence-supported confidence that a claim or requirement is sufficiently reliable for a defined purpose.

20.2 Assurance Level

- Informal assurance
- limited assurance
- reasonable assurance
- continuous assurance
- decision-specific assurance

20.3 `ASSURANCE.AUDIT`

A systematic, independent, documented process for obtaining and evaluating evidence against defined criteria.

Audit Subject

- Organization
- management system
- process
- control
- data
- evaluation
- security
- compliance
- supplier
- model-development process

20.4 `ASSURANCE.INSPECTION`

Examination of a product, process, service, system, installation, or design for conformity with requirements.

20.5 `ASSURANCE.VALIDATION`

Confirmation that a method, model, process, or requirement is suitable for intended use.

20.6 `ASSURANCE.VERIFICATION`

Confirmation that specified requirements have been fulfilled.

20.7 `ASSURANCE.ATTESTATION`

Issue of a statement, after review and decision, that specified requirements have been demonstrated.

20.8 `ASSURANCE.CONFORMITY_ASSESSMENT`

Demonstration that specified requirements are fulfilled.

20.9 Conformity-Assessment Party

- First-party
- second-party
- third-party

20.10 ASSURANCE.CERTIFICATION

Third-party attestation under a defined certification scheme.

Certification Object

- Product
- process
- service
- management system
- person
- system configuration
- evaluation process

20.11 ASSURANCE.CERTIFICATE

A formal record of certification.

20.12 ASSURANCE.CERTIFICATION_SCHEME

The rules, procedures, requirements, assessment, surveillance, claims, and governance for certification.

20.13 ASSURANCE.ACCREDITATION

Independent recognition that a conformity-assessment body is competent and impartial for specified activities.

20.14 ASSURANCE.ACCREDITATION_SCOPE

The exact activities, methods, domains, systems, locations, and limits of recognition.

20.15 ASSURANCE.RECOGNITION

Acceptance of evidence, competence, process, status, or decision for a defined purpose.

20.16 Recognition Type

- Evidence recognition
- competence recognition
- certificate recognition
- process recognition
- legal recognition
- unilateral recognition
- mutual recognition

- conditional recognition

20.17 Assurance Status

- Applicant
- under assessment
- active
- conditional
- suspended
- expired
- withdrawn
- reduced scope
- superseded

20.18 Assurance Relationship Rules

- Auditor `audits` subject.
- Audit `uses_criteria` requirement set.
- Certification body `certifies` object.
- Certificate `attests_conformity_to` scheme.
- Accreditation body `accredits` conformity-assessment body.
- Accreditation `limited_to` scope.
- Recognition authority `recognizes` evidence or competence.
- Surveillance `maintains_or_changes` status.

---

21. Standards and Requirements Taxonomy

21.1 `STANDARD.DOCUMENT`

A governed document providing rules, requirements, guidelines, characteristics, or common practices.

21.2 Standards Stage

- Research concept
- draft
- committee draft
- public review draft
- proposed standard
- approved standard
- active standard
- amended
- superseded

- withdrawn
- retired

21.3 Standards Type

`STANDARD.TECHNICAL`

Defines technical methods, interfaces, measurements, or performance.

`STANDARD.PROCESS`

Defines procedures, records, and workflows.

`STANDARD.MANAGEMENT_SYSTEM`

Defines organizational management requirements.

`STANDARD.PERFORMANCE`

Defines required outcomes.

`STANDARD.INTERFACE`

Defines interactions and data exchange.

`STANDARD.REPORTING`

Defines disclosure and reporting.

`STANDARD.TERMINOLOGY`

Defines terms and classifications.

`STANDARD.TEST_METHOD`

Defines testing or evaluation procedures.

21.4 `STANDARD.SPECIFICATION`

A detailed technical or procedural description.

21.5 `STANDARD.GUIDANCE`

Advisory material explaining interpretation or implementation.

21.6 `STANDARD.FRAMEWORK`

A structured set of concepts, functions, outcomes, or practices.

21.7 STANDARD.CODE_OF_CONDUCT

A set of expected behaviors or voluntary commitments.

21.8 STANDARD.CODE_OF_PRACTICE

Operational guidance describing accepted implementation practices.

21.9 STANDARD.REQUIREMENT

A condition that must be fulfilled within a defined context.

Requirement Force

- Advisory
- voluntary
- contractual
- procurement
- professional
- scheme-mandatory
- legally mandatory
- regulatory

Requirement Form

- Performance-based
- prescriptive
- process-based
- risk-based
- capability-based
- deployment-based
- outcome-based
- reporting-based

Requirement Level

- Baseline
- minimum
- progressive
- tiered
- advanced
- emergency

21.10 STANDARD.CONFORMITY_CRITERION

A criterion used to determine whether a requirement is fulfilled.

21.11 `STANDARD.SAFE_HARBOR`

A provision providing defined protection when specified conditions are met.

21.12 `STANDARD.PRESUMPTION_OF_CONFORMITY`

A presumption that following a recognized standard supports fulfillment of a requirement.

21.13 `STANDARD.SUNSET`

A provision causing expiration unless renewed.

21.14 `STANDARD.REVIEW_CLAUSE`

A requirement for reassessment.

21.15 Standards Relationship Rules

- Standard `defines` requirement.
- Requirement `applies_to` object.
- Protocol `tests` requirement.
- Evidence `supports_conformity_with` requirement.
- Certificate `attests_conformity_to` scheme.
- Law `incorporates_by_reference` standard.
- Standard `supersedes` prior standard.

---

22. Governance and Decision Taxonomy

22.1 `GOV.GOVERNANCE_SYSTEM`

A system of authority, roles, decisions, accountability, oversight, and control.

22.2 Governance Body

- Governing body
- board
- secretariat
- committee
- technical committee
- advisory group
- working group
- appeals body
- ethics body

- security body
- public-interest council

22.3 Governance Function

- Strategy
- oversight
- standard setting
- protocol approval
- task custody
- evaluation administration
- assurance
- incident response
- appeals
- registry
- transparency
- enforcement

22.4 `GOV.DECISION`

A formal determination by an authorized actor.

Decision Type

- Technical
- methodological
- deployment
- access
- risk acceptance
- safeguard
- publication
- certification
- accreditation
- recognition
- standards progression
- legal
- funding
- correction
- retirement

22.5 Decision Outcome

- Approve
- approve with conditions
- pilot

- defer
- reject
- suspend
- withdraw
- recognize
- conditionally recognize
- decline recognition
- retire

22.6 GOV.DECISION_RIGHT

Authority to make a defined decision.

22.7 GOV.MANDATE

A documented assignment of purpose, authority, responsibility, scope, and limits.

22.8 GOV.ACCOUNTABILITY

Obligation to explain, justify, and accept responsibility.

22.9 GOV.OVERSIGHT

Supervisory or independent observation and review.

22.10 GOV.APPEAL

A formal request for review of a decision.

22.11 GOV.COMPLAINT

A documented expression of dissatisfaction or allegation.

22.12 GOV.RECUSAL

Withdrawal because of conflict or disqualification.

22.13 GOV.CONSENSUS

Broad agreement after addressing substantial objections.

22.14 `GOV.DISSENT`

A reasoned unresolved disagreement.

22.15 Governance Risk

- Capture
- conflict
- concentration
- opacity
- inconsistent decision
- authority inflation
- participation failure
- accountability failure
- emergency-power abuse

22.16 Governance Relationship Rules

- Body `has_decision_right` decision type.
- Actor `responsible_for` function.
- Decision `based_on` evidence package.
- Decision `subject_to` appeal.
- Conflict `requires` mitigation or recusal.
- Governance system `oversees` process.

---

23. Incident, Failure, and Correction Taxonomy

23.1 `INCIDENT.INCIDENT`

An event or condition that caused, could have caused, or revealed material harm, failure, compromise, misuse, or loss of control.

23.2 Incident Type

- Safety incident
- security incident
- misuse incident
- privacy incident
- evaluation incident
- research incident
- governance incident
- standards incident
- assurance incident

- interoperability incident
- near miss

23.3 Incident Role of AI

- Initiated
- enabled
- accelerated
- amplified
- failed to prevent
- detected
- mitigated
- incidental
- unknown

23.4 Incident Severity

- Minor
- limited
- material
- serious
- critical
- catastrophic

23.5 Incident Status

- Reported
- triaged
- under investigation
- confirmed
- disputed
- contained
- remediated
- closed
- reopened
- archived

23.6 `INCIDENT.FAILURE`

Inability of a system, process, safeguard, evaluator, or institution to fulfill an intended function or requirement.

23.7 Failure Layer

- Model
- system
- data
- tool
- human
- organizational
- governance
- evaluator
- standard
- incentive
- interoperability

23.8 Failure Mode

- Incorrect output
- refusal failure
- safeguard bypass
- hallucination
- loss of control
- unauthorized action
- monitoring failure
- access-control failure
- evidence contamination
- scoring failure
- conflict failure
- process nonconformity
- correction failure
- recognition drift

23.9 `INCIDENT.ROOT_CAUSE`

An underlying cause whose correction would materially reduce recurrence.

23.10 `INCIDENT.CONTRIBUTING_FACTOR`

A condition increasing likelihood or severity.

23.11 `INCIDENT.CORRECTIVE_ACTION`

Action addressing a detected failure.

23.12 `INCIDENT.PREVENTIVE_ACTION`

Action reducing the likelihood of a potential failure.

23.13 `INCIDENT.CORRECTION`

A visible change addressing error in a record, claim, method, or result.

23.14 Correction Type

- Editorial
- factual
- methodological
- statistical
- legal-status
- security
- evidence-level
- confidence
- withdrawal
- supersession

23.15 Incident Relationship Rules

- Incident `involves` system.
- Incident `realizes` risk.
- Failure `contributes_to` incident.
- Root cause `explains` failure.
- Corrective action `addresses` root cause.
- Incident `triggers` protocol or standards review.
- Correction `updates` claim or record.

---

24. Incentive and Recognition Taxonomy

24.1 `INCENTIVE.INCENTIVE`

A condition changing expected benefit, cost, status, opportunity, or consequence.

24.2 Incentive Mechanism

- Financial reward
- financial penalty
- access
- procurement preference

- insurance
- liability
- professional credit
- publication
- recognition
- prestige
- grant
- prize
- bounty
- enforcement
- membership
- data access

24.3 Incentive Target

- Developer
- deployer
- evaluator
- reviewer
- researcher
- maintainer
- whistleblower
- standard setter
- purchaser
- public institution

24.4 Desired Behavior

- Evaluation
- disclosure
- correction
- maintenance
- replication
- safeguard adoption
- standards adoption
- incident reporting
- public-goods contribution
- capacity building

24.5 Unintended Incentive Effect

- Goodhart effect
- gaming
- metric fixation
- prestige capture

- client capture
- crowding out
- adverse selection
- moral hazard
- free riding
- incumbent entrenchment
- underreporting
- overclassification

24.6 `INCENTIVE.RECOGNITION`

Formal or informal acknowledgment of contribution, competence, or achievement.

Recognition Type

- Contribution credit
- professional recognition
- public award
- evaluator recognition
- corrective credit
- institutional recognition
- mutual recognition

24.7 `INCENTIVE.PRESTIGE`

Durable esteem or status granted by a community or institution.

24.8 Recognition Status

- Nominated
- reviewed
- granted
- conditional
- suspended
- withdrawn
- expired

24.9 Incentive Relationship Rules

- Incentive `targets` actor.
- Incentive `rewards` behavior.
- Incentive `may_create` gaming risk.
- Recognition `based_on` evidence.
- Prestige `does_not_imply` competence or authority.

---

25. Interoperability and International Coordination Taxonomy

25.1 `INTEROP.INTEROPERABILITY`

The ability of distinct systems, protocols, organizations, or jurisdictions to exchange, interpret, and use information or evidence.

25.2 Interoperability Layer

- Semantic
- syntactic
- technical
- measurement
- protocol
- assurance
- institutional
- legal
- security
- operational

25.3 `INTEROP.CROSSWALK`

A structured mapping among classes, requirements, controls, protocols, or standards.

25.4 `INTEROP.BRIDGE_STUDY`

An empirical or analytical study connecting results across protocols, versions, languages, or systems.

25.5 `INTEROP.COMMON_CORE`

Shared elements preserved across implementations.

25.6 `INTEROP.LOCAL_EXTENSION`

A documented jurisdictional, sectoral, linguistic, or institutional addition.

25.7 `INTEROP.EQUIVALENCE`

A determination that different methods or requirements achieve sufficiently comparable outcomes for a stated purpose.

25.8 `INTEROP.COMPARABILITY`

The degree to which results can be meaningfully compared.

25.9 `INTEROP.MUTUAL_RECOGNITION`

Reciprocal acceptance of defined results, status, or competence.

25.10 `INTEROP.FEDERATED_REGISTRY`

A registry model in which multiple authorities maintain interoperable records.

25.11 `INTEROP.TRANSLATION_VALIDATION`

Assessment of whether a translation preserves intended meaning and construct.

25.12 `INTEROP.CAPACITY_BUILDING`

Development of local expertise, infrastructure, governance, and participation.

25.13 Interoperability Status

- Compatible
- partially compatible
- conditionally recognized
- not comparable
- conflicting
- superseded
- mapping incomplete

25.14 Interoperability Relationship Rules

- Crosswalk `maps` class or requirement.
- Bridge study `supports` comparability.
- Recognition `accepts_for_purpose` external result.
- Local extension `extends` common core.
- Translation `version_of` source vocabulary.
- Registry `exchanges_with` registry.

---

26. Research Taxonomy

26.1 RESEARCH.PROJECT

A structured activity intended to generate, test, synthesize, or apply knowledge.

26.2 Research Purpose

- Descriptive
- exploratory
- confirmatory
- evaluative
- causal
- predictive
- comparative
- synthesis
- institutional design
- metaresearch

26.3 Research Method

- Experiment
- observational study
- simulation
- literature review
- systematic review
- scoping review
- rapid review
- narrative review
- living review
- case study
- interview
- survey
- focus group
- Delphi
- structured expert judgment
- forecast
- replication
- reproducibility study
- methodological audit

26.4 Research Object

- Model

- system
- capability
- safeguard
- evaluator
- institution
- standard
- incident
- policy
- legal regime
- research method

26.5 Research Stage

- Question
- protocol
- registered
- collecting
- analyzing
- under review
- published
- monitoring
- corrected
- withdrawn
- retired

26.6 Research Consequence

- Minimal
- limited
- material
- high
- critical

26.7 Research Sensitivity

- Public
- controlled
- confidential
- restricted
- highly restricted

26.8 Research Output

- Note
- brief

- working paper
- white paper
- protocol
- dataset
- code
- task bank
- review
- standard proposal
- case study
- pilot report
- replication report
- registry record

26.9 Research Relationship Rules

- Project `investigates` question.
- Protocol `governs` project.
- Artifact `produced_by` project.
- Evidence `generated_by` project.
- Review `evaluates` project.
- Publication `reports` result.
- Correction `updates` publication.

---

27. Status, Version, Security, and Record Taxonomy

27.1 `STATUS.VERSION`

An identified state of an object.

27.2 Version Change

- Major
- minor
- patch
- emergency
- editorial

27.3 Lifecycle Status

- Draft
- proposed
- approved
- active

- current
- conditional
- expired
- suspended
- corrected
- superseded
- withdrawn
- deprecated
- retired
- archived

27.4 `STATUS.RECORD`

A structured representation of an entity.

27.5 Record Type

- Model record
- system manifest
- protocol record
- evaluation record
- evidence record
- claim record
- reviewer record
- evaluator record
- certificate record
- accreditation record
- incident record
- standard record
- recognition record
- correction record

27.6 Record Integrity

- Signed
- hashed
- witnessed
- version controlled
- mutable
- immutable
- unverifiable

27.7 `SECURITY.CLASSIFICATION`

A category governing access, handling, and disclosure.

Classes

- Public
- controlled
- confidential
- restricted
- highly restricted

27.8 Security Handling

- View
- edit
- export
- redistribute
- summarize
- cite
- destroy
- archive

27.9 Security Status

- Active
- under review
- compromised
- declassified
- released
- destroyed
- archived

27.10 Status Relationship Rules

- Version `supersedes` version.
- Record `represents` entity.
- Status `applies_to` record.
- Correction `amends` record.
- Withdrawal `invalidates_current_use_of` record.
- Archive `preserves` retired record.

---

28. Jurisdiction and Legal Scope Taxonomy

28.1 `JURIS.JURISDICTION`

A legal, geographic, organizational, or institutional domain of authority.

28.2 Jurisdiction Level

- Local
- subnational
- national
- supranational
- treaty-based
- international institutional
- contractual
- organizational

28.3 Legal Instrument

- Constitution
- statute
- regulation
- order
- judicial decision
- treaty
- guidance
- code
- contract
- procurement rule
- license

28.4 Legal Status

- Proposed
- enacted
- effective
- transitional
- stayed
- amended
- repealed
- expired
- superseded

28.5 Applicability

- Developer
- provider
- deployer
- user
- evaluator
- importer

- distributor
- public body
- specific sector
- specific system category

28.6 Legal Relationship Rules

- Requirement `applies_in` jurisdiction.
- Authority `issues` instrument.
- Standard `incorporated_into` legal instrument.
- Decision `binding_in` jurisdiction.
- Crosswalk `does_not_establish` legal equivalence without recognition.

---

29. Core Relationship Vocabulary

The following relationship types are canonical.

29.1 Identity Relationships

- `is_a`
- `instance_of`
- `version_of`
- `same_as`
- `not_same_as`

29.2 Composition Relationships

- `part_of`
- `contains`
- `uses`
- `implements`
- `depends_on`

29.3 Lineage Relationships

- `derived_from`
- `trained_from`
- `fine_tuned_from`
- `forked_from`
- `supersedes`

29.4 Evaluation Relationships

- evaluated_by
- administered_by
- tested_with
- scored_by
- reviewed_by
- replicated_by

29.5 Evidence Relationships

- supports
- challenges
- narrows
- contextualizes
- invalidates
- derived_from_source

29.6 Risk Relationships

- creates_hazard
- exposes
- enables
- may_cause
- mitigates
- detects
- contains
- realizes

29.7 Standards and Assurance Relationships

- defines_requirement
- conforms_to
- certified_against
- accredited_for
- recognized_by
- incorporated_by_reference

29.8 Governance Relationships

- governed_by
- owned_by
- authorized_by
- decided_by

- `appealable_to`
- `overseen_by`

29.9 Temporal Relationships

- `precedes`
- `follows`
- `effective_from`
- `expires_on`
- `superseded_by`
- `withdrawn_on`

29.10 Relationship Confidence

Every inferred relationship may include:

- Confirmed
- high
- moderate
- low
- disputed
- unknown

---

30. Classification Rules

30.1 Model Versus System Rule

Classify a model separately from the system using it.

30.2 Role Separation Rule

Classify every actor role separately.

30.3 Capability-Risk Rule

Do not classify a capability as a risk.

Link the capability to a risk through context.

30.4 Safeguard Rule

Classify a safeguard by function, layer, and mechanism.

Do not classify it as proof of safety.

30.5 Evaluation-Audit Rule

Do not classify an evaluation as an audit unless defined criteria and audit process exist.

30.6 Certification-Accreditation Rule

Certification applies to conformity under a scheme.

Accreditation applies to competence of a conformity-assessment body within scope.

30.7 External-Independent Rule

External is a location or organizational relationship.

Independent is a multidimensional condition.

30.8 Legal-Technical Rule

Preserve legal and technical categories separately.

30.9 Current-Status Rule

Every consequential record should include current status.

30.10 Multi-Label Rule

Use multiple facets when one label would hide important dimensions.

30.11 Uncertain Classification Rule

Use confidence and alternatives.

Do not force a false category.

30.12 Local-Extension Rule

Local extensions should retain mapping to the common core.

30.13 Public-Claim Rule

Public classification should not imply authority beyond the evidence and project status.

---

31. Core Entity Profiles

A profile defines the minimum classes, facets, properties, and relationships required for a recurring record type.

31.1 Model Profile

Required fields:

- Model record identifier
- public or internal name
- developer
- model family
- exact version or checkpoint
- release date
- modality
- function
- general-purpose or domain-specific status
- weight-access status
- training or lineage information where available
- applicable licenses
- current status
- related system records
- evidence and source records
- classification confidence

Optional fields:

- Parameter or scale information
- training-compute information
- training-data summary
- post-training methods
- known limitations
- frontier classification basis
- security restrictions

31.2 System Profile

Required fields:

- System identifier
- system version
- responsible developer or deployer
- model components
- system prompts or prompt-disclosure status
- tools

- retrieval
- memory
- scaffolds
- safeguards
- interfaces
- access tier
- deployment context
- lifecycle stage
- current status
- related evaluation records

31.3 Deployment Profile

Required fields:

- Deployment identifier
- system identifier
- deployer
- operator
- user class
- sector
- geographic scope
- scale
- decision authority
- human oversight
- connected infrastructure
- monitoring
- safeguards
- start date
- current status
- jurisdiction

31.4 Capability Profile

Required fields:

- Capability identifier
- domain
- task or task family
- evaluated object
- conditions
- tools
- elicitation
- reliability
- autonomy
- resource requirements

- human reference
- demonstrated level
- evidence level
- confidence
- limitations
- result status

31.5 Risk Profile

Required fields:

- Risk identifier
- hazard
- threat actor or source
- vulnerability
- exposure
- consequence
- likelihood or plausibility
- scale
- time horizon
- reversibility
- safeguards
- residual risk
- uncertainty
- decision owner
- evidence level
- review date

31.6 Safeguard Profile

Required fields:

- Safeguard identifier
- function
- layer
- mechanism
- risk addressed
- system or organization implementing it
- coverage
- threat model
- evaluation method
- effectiveness evidence
- bypass evidence
- residual risk
- owner
- monitoring

- status
- review date

31.7 Evaluation Protocol Profile

Required fields:

- Protocol identifier
- title
- version
- owner
- purpose
- claim or decision
- evaluated object
- construct
- task universe
- task families
- administration
- elicitation
- environment
- scoring
- uncertainty
- integrity classification
- change control
- expiration
- security
- review status
- current status

31.8 Evaluation Result Profile

Required fields:

- Result identifier
- protocol identifier and version
- system identifier and version
- evaluator
- date
- lifecycle stage
- task sample
- elicitation conditions
- environment
- score or finding
- uncertainty
- integrity status
- evidence level

- confidence
- reviewer
- limitations
- expiration
- status

31.9 Evidence Profile

Required fields:

- Evidence identifier
- source
- source type
- date
- claim relationship
- form
- directness
- quality profile
- provenance
- security classification
- review status
- evidence level
- current status

31.10 Independent Review Profile

Required fields:

- Review identifier
- mandate
- reviewed object
- reviewers
- qualifications
- independence profile
- conflicts
- access
- method
- findings
- dissent
- right of reply
- publication status
- date
- current status

31.11 Evaluator Organization Profile

Required fields:

- Organization identifier
- legal identity
- governance
- ownership
- services
- competence scopes
- personnel
- quality system
- security
- methods
- conflicts
- client concentration
- complaints
- proficiency
- accreditation or recognition status
- current status

31.12 Standard Profile

Required fields:

- Standard identifier
- issuing body
- title
- version
- type
- scope
- requirements
- applicable objects
- development stage
- consensus process
- assurance relationship
- jurisdictional use
- effective date
- review date
- status
- superseded version

31.13 Certificate Profile

Required fields:

- Certificate identifier
- certification body
- scheme
- certified object
- requirements
- scope
- decision date
- effective date
- expiration
- surveillance
- status
- registry location
- exclusions

31.14 Accreditation Profile

Required fields:

- Accreditation identifier
- accreditation body
- accredited body
- criteria
- scope
- locations
- methods
- effective date
- reassessment
- status
- suspension or withdrawal record

31.15 Incident Profile

Required fields:

- Incident identifier
- date
- reporting source
- system
- deployment
- incident type
- AI role
- affected parties

- actual and potential harm
- severity
- evidence
- confidence
- failure modes
- root causes
- response
- corrective action
- status
- disclosure level

31.16 Recognition Profile

Required fields:

- Recognition identifier
- recognizing authority
- object recognized
- recognition type
- purpose
- evidence
- scope
- conditions
- effective date
- expiration
- review
- status

---

32. Machine-Readable Representation

32.1 Design Goal

The taxonomy should support structured data without forcing the Markdown document to function as executable software.

A future machine-readable release may use:

- JSON
- JSON Schema
- JSON-LD
- RDF
- SKOS
- OWL
- YAML
- CSV for controlled lists

32.2 Human and Machine Labels

Every class should have:

- Stable identifier
- preferred label
- definition
- status
- parent
- related classes
- effective version
- change history

32.3 Example Class Record

```
yaml id: CAP.CYBER.AUTONOMOUS_OPERATIONS preferred_label: Autonomous cyber operations capability definition: > The capability of an AI system to plan and execute multiple cyber-operation steps with limited direct human instruction under defined conditions. parent: CAP.CYBER status: preferred facets: - capability_object - autonomy_level - reliability - tool_access - evidence_level related: - CAP.PLANNING - CAP.TOOL_USE - RISK.SECURITY_HARM - EVAL.AGENT_EVALUATION version: 1.0.0
```

32.4 Example Evaluation Result Record

```
yaml id: SB-RESULT-2026-0044 type: EVAL.RESULT evaluated_object: SB-SYSTEM-2026-0017 protocol: SB-PROTOCOL-2026-0008 protocol_version: 1.2.0 capability: - CAP.CYBER.AUTONOMOUS_OPERATIONS evaluation_party: ACTOR.THIRD_PARTY_EVALUATOR integrity: EVAL.INTEGRITY.UNEXPOSED evidence_level: EVIDENCE.LEVEL.E3_SUBSTANTIATED confidence: high status: STATUS.CURRENT security: SECURITY.CONFIDENTIAL expires_on: 2027-01-15
```

32.5 JSON-LD Relationship Example

```
json { "@id": "SB-RESULT-2026-0044", "@type": "EVAL.RESULT", "evaluatedBy": {"@id": "SB-PROTOCOL-2026-0008"}, "concernsSystem": {"@id": "SB-SYSTEM-2026-0017"}, "supportsClaim": {"@id": "SB-CLAIM-2026-0031"}, "reviewedBy": {"@id": "SB-REVIEW-2026-0009"}, "status": "STATUS.CURRENT" }
```

32.6 Schema Validation

Machine-readable records should validate:

- Required fields
- identifier format

- allowed classes
- version
- date format
- relationship targets
- status
- security classification

32.7 Provenance

Structured records should preserve:

- Creator
- creation date
- source
- modification
- reviewer
- approval
- supersession

32.8 Digital Signatures

High-consequence records may be digitally signed.

32.9 No Schema-Meaning Substitution

A technically valid JSON record can still be conceptually wrong.

Schema validation does not replace expert classification.

---

33. External Mapping and Crosswalk Rules

33.1 Mapping Types

A class may map to an external class as:

- Exact match
- close match
- broader match
- narrower match
- related match
- no valid match
- disputed match

33.2 Mapping Record

Record:

- Standards Body class
- external source
- external identifier
- version
- mapping type
- rationale
- jurisdiction or domain
- reviewer
- date
- limitations

33.3 No Silent Equivalence

Different labels should not be treated as exact equivalents without review.

33.4 Legal Mapping

Legal classifications should retain:

- Jurisdiction
- authority
- effective date
- legal consequences

33.5 Standards Mapping

Standards crosswalks should identify:

- Requirement meaning
- object
- evidence
- assurance
- exceptions
- version

33.6 Translation Mapping

Translated classes should record:

- Source term
- target term
- literal meaning

- intended concept
- non-equivalence
- domain validation

33.7 External Vocabulary Preservation

When importing an external term:

- Preserve its exact source meaning
- identify differences
- avoid redefining it silently
- classify it as external or mapped

33.8 Interoperability Principle

Mapping should enable understanding.

It should not manufacture consensus or legal effect.

---

34. Domain Extension Framework

34.1 Purpose

Some capability, risk, safeguard, and evaluation domains require specialized extensions.

Examples:

- Cybersecurity
- biology
- chemical systems
- finance
- critical infrastructure
- healthcare
- robotics
- public administration

34.2 Extension Requirements

A domain extension should define:

- Scope
- classes
- parent classes
- domain-specific relationships
- evidence needs

- security needs
- legal context
- subject-matter reviewers
- version
- mappings

34.3 Extension Identifier

Preferred pattern:

`DOMAIN-SPECIFIC-PREFIX.CLASS`

Example:

`CYBER.TASK.EXPLOIT_CHAIN`

34.4 Common-Core Requirement

Every extension should map material classes to the Standards Body common core.

34.5 Domain Authority

Domain experts should review extensions.

34.6 Security

A public taxonomy may omit or generalize sensitive subclasses when detailed classification would increase harm.

34.7 Extension Status

Use:

- Provisional
- pilot
- approved
- deprecated
- retired

---

35. Taxonomy Governance

35.1 Taxonomy Owner

Standards Body owns and maintains the canonical taxonomy.

35.2 Stewardship Function

A future taxonomy steward or committee should:

- Maintain class definitions
- review proposals
- resolve overlaps
- manage identifiers
- approve mappings
- maintain machine-readable releases
- coordinate translations
- publish changes
- audit use

35.3 Proposal Requirements

A proposed class should include:

- Identifier
- preferred label
- definition
- purpose
- parent
- facets
- relationships
- examples
- nonexamples
- source
- status
- affected records
- mapping implications

35.4 Approval Criteria

Approve a new class only when:

- The distinction is meaningful
- existing classes are insufficient
- the class can be applied consistently
- it improves evidence, governance, or interoperability
- complexity remains proportionate
- qualified reviewers agree or dissent is documented

35.5 Duplicate Control

Before adding a class:

- Search existing taxonomy
- search terminology
- inspect external mappings
- examine whether a facet is more appropriate

35.6 Change Types

Editorial Change

No intended meaning change.

Clarifying Change

Improves definition without changing classification.

Substantive Change

Changes meaning, parent, relationships, or use.

Deprecation

Replaces a class.

Retirement

Ends active use.

35.7 Change Impact Review

Review affected:

- Canonical files
- registries
- protocols
- standards
- public claims
- data schemas
- crosswalks
- translations

35.8 Appeals

A contributor may appeal:

- Class rejection

- classification decision
- mapping
- deprecation
- public use

35.9 Emergency Correction

Correct immediately when a class:

- Creates a safety risk
- implies false authority
- causes systematic misclassification
- conflicts with a binding legal definition
- creates major interoperability failure

35.10 Public Change Log

Publish material class changes and replacements.

---

36. Taxonomy Quality Tests

36.1 Identity Test

Does the class describe a type of object rather than an opinion about it?

36.2 Parent Test

Is the parent relationship valid?

36.3 Distinction Test

Can qualified users distinguish the class from related classes?

36.4 Completeness Test

Does the taxonomy cover the material classes needed for the use case?

36.5 Exclusivity Test

Where classes are intended to be exclusive, can one object be assigned consistently?

36.6 Polyhierarchy Test

Where overlap is legitimate, does the structure permit multiple classifications?

36.7 Facet Test

Should the proposed class be an independent facet instead?

36.8 Evidence Test

Can a classification be supported by identifiable evidence?

36.9 Authority Test

Does the label imply unsupported legal or institutional status?

36.10 Temporal Test

Can the classification change over time without losing history?

36.11 Interoperability Test

Can the class be mapped to external systems?

36.12 Translation Test

Can the concept be translated without unacceptable loss?

36.13 Security Test

Could publishing the class or record increase harm?

36.14 Usability Test

Can researchers, evaluators, policymakers, and technical systems apply it?

36.15 Anti-Gaming Test

Could actors manipulate the classification to gain status or avoid obligations?

---

37. Common Classification Errors

37.1 Product-Name Collapse

Error:

Treating one marketed name as a stable model or system identity.

Correction:

Create separate records for family, model version, system, and deployment.

37.2 Frontier as Permanent Status

Error:

Treating frontier status as timeless.

Correction:

Record basis, date, dimension, and confidence.

37.3 Open-Source Overstatement

Error:

Classifying an open-weight model as fully open-source without component analysis.

Correction:

Classify weights, code, data, license, and documentation separately.

37.4 Capability-Risk Collapse

Error:

Classifying cyber capability as cyber risk.

Correction:

Link capability to risk through actor, access, vulnerability, safeguard, and consequence.

37.5 Evaluation-Assurance Collapse

Error:

Classifying a benchmark result as certification.

Correction:

Preserve evaluation, review, scheme, and certification as separate entities.

37.6 External-Independent Collapse

Error:

Classifying every outside reviewer as independent.

Correction:

Use the independence profile.

37.7 Accreditation Overreach

Error:

Classifying an accredited organization as competent for all AI evaluation.

Correction:

Record exact accreditation scope.

37.8 Status Omission

Error:

Displaying a withdrawn or expired record without status.

Correction:

Require current status and history.

37.9 Jurisdiction Omission

Error:

Presenting a legal category as universal.

Correction:

Attach jurisdiction and effective date.

37.10 Single-Score Classification

Error:

Classifying a system as safe or unsafe from one metric.

Correction:

Use capability, risk, safeguard, evidence, and context profiles.

37.11 Overclassification

Error:

Creating new classes for every minor variation.

Correction:

Use properties and facets.

37.12 Underclassification

Error:

Using broad classes that erase decision-relevant differences.

Correction:

Add a subclass when the distinction changes evidence, governance, or action.

---

38. Worked Classification Examples

38.1 Hosted General-Purpose Assistant

Possible classification:

- `AI.GENERAL_PURPOSE_MODEL`
- `SYS.AI_SYSTEM`
- `SYS.TOOL_USING`
- `ACCESS.API_PUBLIC`
- `DEPLOY.CONSUMER`
- `LIFE.GENERAL_DEPLOYMENT`
- `SAFEGUARD.ACCESS_CONTROL`
- `SAFEGUARD.MONITORING`

Additional records required:

- Exact model version
- system prompt status
- tools
- safeguards
- deployment geography
- evaluation history

38.2 Open-Weight Research Model

Possible classification:

- `AI.MODEL`
- `AI.OPEN_WEIGHT`
- `ACCESS.WEIGHTS_PUBLIC`
- `DEPLOY.RESEARCH`
- `LIFE.RELEASE`

Do not infer:

- Full open-source status
- low risk
- reproducibility
- unrestricted legal use

38.3 Held-Out Cyber Evaluation

Possible classification:

- `EVAL.EVALUATION`
- `EVAL.HELD_OUT`
- `EVAL.AGENT_EVALUATION`
- `CAP.CYBER`
- `SECURITY.RESTRICTED`
- `ACTOR.THIRD_PARTY_EVALUATOR`
- `REVIEW.INDEPENDENT`

Required relationships:

- Evaluates system
- uses protocol
- samples task universe
- produces result
- supports capability claim
- expires on date

38.4 Independent Review of Developer Evidence

Possible classification:

- `REVIEW.INDEPENDENT_EXPERT_REVIEW`
- subject: evidence package
- access: confidential evidence
- disciplines: technical and methodological
- outcome: approve with conditions
- dissent: present

Do not classify as:

- Audit
- certification
- regulatory approval

unless those additional processes exist.

38.5 Evaluator Certification

A certification of an evaluator's management system may be classified as:

- `ASSURANCE.CERTIFICATION`
- certified object: management system
- scheme: named
- body: named
- scope: named

It should not be classified automatically as:

- Accreditation for AI evaluation
- competence in every capability domain
- certification of evaluation results

38.6 Post-Deployment Incident

Possible classification:

- `INCIDENT.AI_INCIDENT`
- `INCIDENT.SECURITY_INCIDENT`
- lifecycle: post-deployment
- AI role: enabled or amplified
- severity: material
- status: under investigation
- evidence confidence: moderate

The record should preserve uncertainty about causation.

38.7 Proposed Procurement Requirement

Possible classification:

- `STANDARD.REQUIREMENT`
- force: procurement
- form: performance-based
- stage: proposed
- object: deployed high-stakes system
- evidence level: E2 or E3

- jurisdiction: purchasing institution

Do not classify as legally mandatory unless law creates that effect.

---

39. Taxonomy Implementation Pathway

Phase 1: Canonical Adoption

Use this taxonomy in new canonical files and registries.

Phase 2: Existing-File Audit

Review existing foundation and institutional files for:

- Misclassified objects
- inconsistent role labels
- status omissions
- model-system confusion
- audit-certification-accreditation confusion

Phase 3: Core Registry Schema

Create schemas for:

- Models
- systems
- protocols
- results
- evaluators
- incidents
- standards
- recognition

Phase 4: Identifier Service

Establish controlled identifier assignment.

Phase 5: Machine-Readable Release

Publish the common core in a structured format.

Phase 6: Domain Extensions

Develop controlled extensions for:

- Cyber
- biological
- critical infrastructure
- persuasion
- AI research and development

Phase 7: External Crosswalks

Map relevant external vocabularies and standards.

Phase 8: Classification Pilot

Classify a representative set of:

- Frontier systems
- evaluations
- evaluator organizations
- standards
- incidents

Measure inter-rater consistency and ambiguity.

Phase 9: Public Registry Integration

Use taxonomy identifiers in public records.

Phase 10: Continuous Governance

Review classes, mappings, and use annually and after major incidents or standards changes.

---

40. Taxonomy Scorecard

Dimension	Core question
Object identity	Is the actual object identified?
Correct level	Is classification at model, system, deployment, organization, or legal level correct?
Class validity	Does the class accurately represent the object?
Parent validity	Is the hierarchical relationship defensible?
Facet use	Are independent characteristics represented as facets?

Dimension	Core question
Relationship precision	Are entity relationships named accurately?
Evidence	Is the classification supported?
Confidence	Is uncertainty in classification visible?
Version	Is the relevant version identified?
Status	Is current, expired, superseded, or withdrawn status shown?
Jurisdiction	Is legal or institutional scope preserved?
Security	Is disclosure classification appropriate?
Role separation	Are developer, provider, deployer, evaluator, and authority roles separated?
Capability-risk separation	Are ability and risk kept distinct?
Evaluation-assurance separation	Are test, review, audit, certification, and accreditation distinct?
Interoperability	Can the class map to external vocabularies?
Machine readability	Can the record be represented structurally?
Human readability	Can informed users understand the class?
Temporal stability	Can changes occur without losing history?
Anti-gaming	Does the classification resist prestige or compliance gaming?
Extensibility	Can domains extend the taxonomy without breaking the core?
Governance	Can classes be reviewed, appealed, corrected, and retired?

40.1 Critical Failures

The following normally invalidate a consequential classification:

- Unidentified object
- model-system collapse
- unsupported legal status
- unsupported accreditation or certification claim
- missing version for a changing object
- missing jurisdiction for a legal category
- capability classified as risk without context
- expired record shown as current
- relationship asserted without evidence
- public classification exposing restricted information
- identifier reused for a different concept

40.2 No Universal Taxonomy Score

Do not average all dimensions into one number.

Critical errors should remain visible.

41. Classification Record Template

Record ID:
Record type:
Preferred label:
Version:
Status:
Classification date:
Classifier:
Reviewer:

Object Identity

Primary Class

Subclasses

Facets

Relationships

Jurisdiction

Security Classification

Supporting Evidence

Classification Confidence

- Confirmed
- high
- moderate
- low
- disputed
- unknown

Alternative Classifications Considered

Limitations

Review Date

Change History

---

42. Taxonomy Class Proposal Template

Proposed identifier:
Preferred label:
Proposer:
Date:

Definition

Purpose

Parent Class

Subclasses

Applicable Facets

Required Properties

Relationships

Examples

Nonexamples

Existing Classes Considered

External Mappings

Evidence and Sources

Security Considerations

Implementation Impact

Proposed Status

- Provisional
- preferred
- accepted
- context-specific

Reviewers

Decision

---

43. Taxonomy Change Request Template

Change request ID:
Affected class:
Current version:
Proposer:
Date:

Change Type

- Editorial
- clarification
- substantive
- deprecation
- retirement
- emergency correction

Current Definition or Structure

Proposed Definition or Structure

Rationale

Evidence

Affected Records

Affected Canonical Files

Interoperability Impact

Translation Impact

Transition Plan

Effective Version

Review and Decision

---

44. Crosswalk Record Template

Crosswalk ID:
Standards Body taxonomy version:
External source:
External version:
Jurisdiction or domain:
Reviewer:
Date:

Standards Body class	External class	Mapping type	Rationale	Limitations	Status

Mapping types:

- Exact match
- close match
- broader match
- narrower match

- related match
- no valid match
- disputed

Recognition Effect

State explicitly whether the crosswalk has:

- Informational effect only
- technical recognition effect
- assurance effect
- contractual effect
- legal effect

Do not imply legal effect unless authorized.

---

45. Registry Record Template

Registry:
Record ID:
Entity type:
Entity identifier:
Preferred label:
Owner:
Version:
Status:
Effective date:
Expiration or review date:

Classification

Scope

Relationships

Evidence

Confidence

Security and Access

Recognition

Corrections

Superseded Records

Digital Signature or Integrity Record

---

46. Taxonomic Decision Rules

46.1 New-Class Decision

Create a new class when the distinction:

- Changes evidence requirements
- changes governance
- changes risk or safeguards
- changes standards applicability
- changes interoperability
- cannot be represented adequately through an existing facet or property

46.2 Facet Decision

Create a facet when the characteristic:

- Applies across several primary classes
- does not change the entity's essential identity
- can vary independently

- supports filtering or comparison

46.3 Relationship Decision

Create or use a relationship when meaning depends on the connection among entities.

46.4 Provisional Classification Decision

Use provisional status when:

- Evidence is incomplete
- external definitions are changing
- class boundaries are not stable
- implementation has not been tested

46.5 Deprecation Decision

Deprecate a class when:

- A clearer replacement exists
- the term creates systematic ambiguity
- external use has shifted
- the class combines incompatible concepts

46.6 Retirement Decision

Retire a class when it no longer serves current work and transition is complete.

46.7 Public Disclosure Decision

Do not publish detailed classification when it would reveal:

- Active held-out content
- exploitable vulnerabilities
- protected personal data
- restricted capability information
- security architecture whose disclosure increases risk

Publish the highest safe parent class and a reason for restriction where appropriate.

46.8 Legal Classification Decision

Seek qualified legal review when a classification affects:

- Regulatory status
- compliance

- legal obligation
- liability
- protected legal category
- cross-border applicability

---

47. Canonical Standards Body Taxonomy Positions

Standards Body adopts the following working positions.

1. A taxonomy is operational infrastructure, not decorative organization.
2. The object should be identified before the label is selected.
3. Models, systems, deployments, organizations, and jurisdictions are distinct levels of analysis.
4. A marketed product name is not a sufficient technical identifier.
5. Model family, model version, system version, and deployment should have separate records.
6. A system may inherit model capability evidence only through documented analysis.
7. A capability is an ability under specified conditions.
8. A capability is not itself a risk.
9. Risk classification requires context, actor, access, exposure, vulnerability, safeguard, likelihood, and consequence.
10. A safeguard should be classified by function, layer, mechanism, scope, and effectiveness evidence.
11. A safeguard does not constitute a guarantee.
12. A test is not the same as an evaluation.
13. A benchmark is not the same as a complete protocol.
14. A review is not automatically an audit.
15. An external reviewer is not automatically independent.
16. An audit is criteria-based, systematic, independent, and documented.
17. Certification is third-party attestation under a defined scheme.
18. Accreditation recognizes conformity-assessment competence within scope.
19. A certificate does not establish properties outside its scheme and scope.
20. Legal approval, certification, accreditation, recognition, and endorsement are distinct.

21. Every consequential classification should include version and status.
22. Frontier status is time-dependent and dimension-dependent.
23. Open-weight, open-source, open-data, and public-access classifications should remain separate.
24. Public, controlled, confidential, restricted, and highly restricted are distinct access classes.
25. Evaluation integrity status should travel with a result.
26. A contaminated or compromised result should not remain classified as current without review.
27. Evidence level and confidence are distinct facets.
28. Evidence may support, challenge, narrow, contextualize, or fail to resolve a claim.
29. A negative evaluation result should not automatically classify a capability as absent.
30. Actor roles should be represented separately even when one organization performs several roles.
31. Sponsor, developer, provider, deployer, operator, evaluator, reviewer, and authority are distinct roles.
32. Institutional authority should be classified by source and scope.
33. A research project should not be classified as a regulator, certification body, or accreditation body without actual mandate and function.
34. Standards may be technical, process-based, performance-based, management-system, interface, reporting, terminology, or test-method standards.
35. A standard may be voluntary while requirements adopted through contract or law may be binding.
36. Standards stage and legal force should remain separate.
37. An incident may have several classifications simultaneously.
38. AI involvement in an incident should be classified by causal or contributory role rather than assumed.
39. Failure, root cause, contributing factor, correction, and preventive action are distinct.
40. Incentive mechanisms should be classified together with intended and unintended effects.
41. Prestige and recognition do not establish competence or authority.
42. Interoperability does not require identical classification systems.
43. A crosswalk does not establish equivalence automatically.

44. Legal equivalence requires recognition by the relevant authority.
45. Local extensions should preserve mapping to a common core.
46. Noncomparability is a valid taxonomic result.
47. Uncertain classifications should remain uncertain rather than forced.
48. Machine-readable validity does not guarantee conceptual validity.
49. Taxonomic classes should be corrected, deprecated, and retired visibly.
50. Standards Body should evaluate the consistency, usability, and real-world effects of its own taxonomy.

---

48. Relationship to Other Canonical Files

`PROJECT_IDENTITY.md`

Defines the identity, present role, authority boundaries, audiences, and public positioning of Standards Body.

The taxonomy must not classify the project in a way that exceeds those boundaries.

`TERMINOLOGY.md`

Defines the preferred terms and meanings.

This taxonomy organizes those terms into classes, facets, properties, and relationships.

`FOUNDATIONS_APPENDIX.md`

Defines the integrated eight-foundation system.

This taxonomy provides the entity architecture required to represent that system.

`EVIDENCE_STANDARDS.md`

Defines evidence levels, source quality, claims, confidence, and correction.

This taxonomy provides corresponding evidence and claim classes.

`RESEARCH_METHODOLOGY.md`

Defines how research is planned, conducted, reviewed, published, and corrected.

This taxonomy classifies research projects, methods, artifacts, review levels, and outputs.

EVALUATION_PHILOSOPHY.md

Will define the deeper conceptual approach to constructs, validity, capability, risk, thresholds, and interpretation.

This taxonomy provides the formal categories used by that philosophy.

INSTITUTION_DESIGN.md

Will define the institutional system.

This taxonomy provides classes for organizations, roles, authority, schemes, registries, and governance bodies.

GOVERNANCE_FRAMEWORK.md

Will define decision rights, committees, oversight, recusal, appeals, and accountability.

This taxonomy provides the corresponding governance entities and relationships.

STANDARDS_DEVELOPMENT_PROCESS.md

Will define the standards lifecycle.

This taxonomy provides standards types, stages, requirement forms, and statuses.

EVALUATOR_ACCREDITATION_FRAMEWORK.md

Will define evaluator competence and recognition.

This taxonomy distinguishes evaluators, audit, certification, accreditation, scopes, and statuses.

TRANSPARENCY_FRAMEWORK.md

Will define public, controlled, confidential, restricted, and highly restricted information handling.

WEBSITE_SOURCE_OF_TRUTH.md

Will apply approved classifications and public descriptions to website content.

SOURCES.md

Will maintain source records linked to evidence and classification decisions.

`FAILURE_DATABASE.md`

Will use the incident, failure, root-cause, and correction taxonomy.

`VERSION_HISTORY.md`

Will preserve the version and status history of taxonomy classes and classified records.

---

49. Taxonomy Maturity Model

Level 0: Uncontrolled Vocabulary

Characteristics:

- Inconsistent labels
- no stable identifiers
- model and system conflation
- no status records
- no change control

Level 1: Controlled Terms

Characteristics:

- Preferred vocabulary
- top-level classes
- basic definitions
- human-readable guidance

Level 2: Faceted Classification

Characteristics:

- Multiple classification axes
- version and status
- confidence
- actor-role separation
- evidence and security facets

Level 3: Relationship and Registry Model

Characteristics:

- Typed relationships

- entity profiles
- identifiers
- registry records
- cross-file consistency
- domain extensions

Level 4: Machine-Readable and Interoperable

Characteristics:

- Structured schema
- external mappings
- validation
- persistent identifiers
- federated exchange
- translation support

Level 5: Adaptive Taxonomic Infrastructure

Characteristics:

- Continuous audit
- classification-quality metrics
- automated status propagation
- public change history
- domain-governance network
- measured interoperability
- retirement of obsolete concepts

Maturity Rule

Taxonomic maturity depends on use and governance, not document length.

A complete vocabulary that is not applied consistently remains low maturity.

---

50. Taxonomy Audit Program

50.1 Audit Purpose

Determine whether Standards Body records and publications use the taxonomy correctly.

50.2 Audit Sample

Sample:

- Foundation papers
- website pages
- evaluation results
- system profiles
- incident records
- standards proposals
- evaluator records
- public claims

50.3 Audit Questions

- Is the object correctly identified?
- Is the level of analysis correct?
- Are roles separated?
- Are capability and risk distinct?
- Are evaluation and assurance distinct?
- Are version and status present?
- Is jurisdiction present?
- Is confidence visible?
- Are relationships supported?
- Are deprecated classes still used?
- Is security classification appropriate?

50.4 Error Categories

- Critical
- material
- minor
- editorial

50.5 Corrective Action

A taxonomic error may require:

- Record correction
- public correction
- schema update
- contributor guidance
- class revision
- crosswalk revision
- affected-file review

50.6 Audit Metrics

Possible metrics:

- Correct-classification rate
- inter-rater agreement
- missing-version rate
- missing-status rate
- role-conflation rate
- deprecated-class rate
- unresolved ambiguity
- correction time
- mapping coverage

---

51. Final Taxonomy Position

A frontier AI standards project cannot build credible evaluation, assurance, or governance on top of unstable categories.

The system fails when:

- A model name stands in for a full system
- a system result stands in for every deployment
- a capability stands in for risk
- a safeguard stands in for safety
- a review stands in for audit
- certification stands in for approval
- accreditation stands in for universal competence
- a national rule stands in for a global category
- an expired result stands in for current evidence
- a famous institution stands in for a defined role
- a score stands in for a multidimensional profile

Taxonomy is the discipline that prevents these substitutions.

It tells Standards Body what kind of object is being discussed.

It preserves the object's version, scope, context, status, and jurisdiction.

It makes relationships explicit.

It supports structured evidence.

It allows one institution to understand another without pretending that all systems are identical.

It allows the project to evolve without silently rewriting its history.

The defining taxonomic rule of Standards Body is:

Name the object, classify the level, attach the context, state the relationship, and preserve the version.

Revision Record

Version 1.0

Date: July 16, 2026

Change type: Complete foundational edition

Summary: Establishes the canonical Standards Body taxonomy. Defines design principles, class and facet architecture, identifiers, classification status and confidence, twenty-three primary domains, actors, AI artifacts, systems, lifecycle, access, deployment, capabilities, risks, safeguards, evaluations, evidence, review, assurance, standards, governance, incidents, incentives, interoperability, research, status, security, jurisdiction, relationship vocabulary, classification rules, entity profiles, machine-readable representation, external mappings, domain extensions, governance, quality tests, common errors, worked examples, implementation, scorecard, operational templates, canonical positions, maturity, audit, and cross-file interfaces.

Status: Approved foundational source.