

Open AI Glossary

Version 0.7 / 2026-10-12 A shared reference for research and engineering: compare meanings, inspect their boundaries, and follow the evidence.

Read in the order Workflow, Harness, Model, LLM, Agent, Schema. The sequence is a reading path, not a claim that all six form a single subtype hierarchy.

External evidence and repository certification are separate. Each working definition remains a proposal until a documented repository consensus decision is recorded.

Source policy

Research applicable formal institutional definitions first, then peer-reviewed papers, then industry standards, first-party documentation, preprints and academic materials, and finally Repository internal proposals.

Verify the original text, edition, publication status and context. A corporate engineering blog belongs to tier 3. Hosting a paper at a university does not change its publication type.

Prefer relevance over an unrelated authority. Preserve attributed disagreements, record source gaps, and do not turn a single paper into a consensus statement.

Citations accompany definitions, boundaries and typed relations. An editorial inference is marked explicitly; a source citation identifies its basis, not an endorsement of our inference.

Repository certification

Yellow means Repository / Proposed / not certified. Green means Repository / Certified by consensus and requires an explicit unanimous repository decision, date, accessible decision link and evidence record. A merge, publication or authoritative external source does not establish repository certification.

Licensing

Original handbook prose: CC BY 4.0. Build scripts and site code: MIT. Third-party sources retain their own rights.

Reading index

- [Workflow](#)
- [Harness](#)
- [Model](#)
- [LLM](#)
- [Agent](#)
- [Schema](#)

Workflow

Process, description, execution An arrangement of activities and control rules for carrying out a process. Distinguish the process, its description, and the system that executes it. [19, 20] *Editorial interpretation.*

Definition comparison

WfMC terminology

Institutional; business automation WfMC describes partial or complete business-process automation, with work or information passed between participants under procedural rules. [20, 21]

Workflow Patterns

Peer-reviewed; control flow A process definition specifies activities and routing; an instance enacts that definition for a particular case. The paper evaluates static control-flow patterns. [19]

LLM engineering

First-party; narrower architecture Anthropic calls predefined code paths coordinating LLMs and tools workflows, and contrasts them with model-directed agents. This is an architectural convention within LLM applications. [1]

Scope

This entry uses Workflow broadly for research and engineering. WfMC places it in business-process automation; its narrower historical definition does not itself define every scientific or Python workflow. [20, 12] *Editorial interpretation.*

Includes

- Editorial example: a deterministic Python load -> validate -> transform -> save pipeline has activities, dependencies and a completion condition. An LLM is unnecessary for our broader working usage. [12] *Editorial interpretation.*
- Editorial example: a fixed retrieve -> summarize -> verify LLM pipeline is a workflow even when individual model outputs vary. Fixed control flow does not imply deterministic outputs. [1] *Editorial interpretation.*

Excludes and counterexamples

- A scheduler or execution engine is an implementation that runs work, not the process description itself. [20, 19] *Editorial interpretation.*
- A list of unrelated functions with no stated dependencies or process purpose fails our operational test. [12] *Editorial interpretation.*

Different usages and open boundaries

- Workflow may name a process arrangement, a stored definition, or a run in casual usage. State which object is meant before comparing implementations. [19] *Editorial interpretation*.
- Predefined paths are not a universal requirement: WfMC distinguishes production workflows from ad-hoc workflows whose procedural rules can change during execution. [21]
- The 2003 patterns cover static control flow, not all resource allocation, exceptions or transaction semantics. Do not treat their scope as a universal exclusion rule. [19]

Workflow and execution governance

Sequence, dependencies, branches and loops describe the process. Validation and retry can also be workflow steps; a harness may enforce shared execution policies around them. This overlap does not make Harness a Workflow subtype. See the paired comparison in Harness for the schema-extraction example. [19, 11, 12] *Editorial interpretation*.

Operational criteria

Repository test: identify a purpose, activities, dependencies or transition rules, and a completion or termination condition. Label the artifact as description, execution instance or executor. These are editorial criteria, not a quoted standard. [12] *Editorial interpretation*.

Concept relationships

Only is-a means a strict subtype. Uses and supports/orchestrates describe composition and control. Workflow uses LLM

An LLM workflow uses an LLM for one or more tasks; many workflows use no language model. [1] *Editorial interpretation*. Workflow uses Agent

A designed workflow may delegate an activity to an agent; this is composition, not a claim that every workflow is agentic. [12] *Editorial interpretation*. Harness supports/orchestrates Workflow

A harness realizes or coordinates workflows; runtime-to-process is not strict subtype nesting. [4, 2] *Editorial interpretation*. Schema supports/orchestrates Workflow

Supports a workflow at data-interpretation or validation boundaries; it supplies a description or contract, not the executing control flow. This is a supporting-artifact relation, not a subtype. [18, 12] *Editorial interpretation*.

Repository working definition

Repository / Proposed / not certified No consensus decision recorded. Use Workflow for an organized process with explicit coordination rules, independent of whether its tasks are manual, deterministic code or model calls. Keep the executor separate from the process it realizes. [12]

[Harness vs Workflow](#)

Sources and research gaps

Institutional scope gap: the historical WfMC definition concerns business automation. The broader Python example and our minimum operational test are explicit Repository proposals.

Retrieval limitation: indexed original excerpts verified the WfMC cover, glossary version and relevant definitions; direct retrieval of the two WfMC PDFs failed.

[20] Tier 1 / Institutional white paper. Context: Business-process automation; quotes the WfMC glossary. Verification: Cover and page 8 verified through indexed excerpts of the original PDF and the official publication catalogue. Direct PDF retrieval failed; no claim of full-document review. [21] Tier 1 / Institutional terminology specification. Context: Production and ad-hoc business workflow. Verification: Version and page 8 verified in indexed original excerpts and the official publication catalogue. Direct PDF retrieval failed. [11] Tier 1 / Institutional engineering description. Context: Healthcare interoperability and conformance testing, not a universal definition of every harness. Demonstrates a non-LLM execution and orchestration harness. Verification: Read the original NIST page, including the test-harness/workflow-management descriptions and creation/update dates. [19] Tier 2 / Peer-reviewed journal article. Context: Business workflow control-flow patterns; not an exhaustive theory of every process. Verification: Final published PDF read; peer-review status, journal and DOI checked against the TU/e publication record. [1] Tier 3 / First-party engineering blog. Context: Company-specific distinction between predefined LLM workflows and dynamic agents. Verification: Original text and publication record checked 2026-10-10. [12] Tier 4 / Repository internal proposal; not certified. Context: Internal proposals and editorial interpretations, not external consensus. Verification: Versions 1.1 and 1.2 add the Schema proposal and the flow/governance experimental example while preserving prior proposal sections. All remain unratified; no adoption decision claimed. Version 1.3 changes branding and certification scope only; it does not certify any proposal.

Revision record

Version 1.2 / Updated 2026-10-12 1.0 / 2026-10-10: Researched and rewritten with attributed definitions, typed relations and an unratified Repository proposal. 1.1 / 2026-10-11: Add the cited flow/governance comparison, overlap boundaries and illustrative experiment; Repository meaning remains proposed. 1.2 / 2026-10-12: Scope the working definition and certification labels to this repository under Open AI Glossary. Citations and proposed status remain unchanged.

Related terms

[Harness](#); [LLM](#); [Agent](#)

Harness

The execution layer around a model In agentic AI, a harness is software around a model that connects it to context and tools and controls execution. Sources differ on whether this means the control loop or the wider non-model runtime. [4, 2] *Editorial interpretation.*

Definition comparison

ASD advisory

Institutional; security and governance ASD uses harness for the non-LLM components of an agentic system, including connections to tools, data, memory and planning workflows, privileges and the control loop. [4]

SWE-agent architecture

Peer-reviewed; agent-computer interface SWE-agent supplies a task-oriented interface for a language-model agent to navigate repositories, edit files and execute tests. This informs the runtime boundary; the paper is not treated as a general definition of Harness. [8] *Editorial interpretation.*

Managed Agents architecture

First-party; product boundary Anthropic separates session, harness and sandbox. Its harness drives the loop and routes model-requested tool calls; the sandbox executes them. [2]

Conceptual proposal

Preprint; not established consensus Macedo proposes an operational boundary against SDKs, frameworks, IDE plugins, evaluation scaffolds and orchestrators. This is one author’s definition proposal. [14]

Scope

Focus here is an agent runtime harness. An evaluation harness instead runs tasks and measures results; a framework or SDK supplies construction abstractions and may include a runtime. Shared naming does not make these identical objects. [14] *Editorial interpretation.*

Includes

- Repository example: a running loop that assembles context, dispatches tools, records outcomes and applies stop or permission rules qualifies under our working definition. [12] *Editorial interpretation.*
- A harness can realize fixed workflows as well as dynamic agent loops. Describe its actual control responsibilities rather than inferring them from a product name. [4, 2] *Editorial interpretation.*

Excludes and counterexamples

- Model weights alone do not supply the surrounding runtime. [4] *Editorial interpretation.*
- An SDK import or workflow diagram alone does not demonstrate a running harness. An evaluation runner is labeled evaluation harness unless it also has the claimed agent-runtime responsibilities. [14] *Editorial interpretation.*

Different usages and open boundaries

- ASD’s broad organizational boundary and Anthropic’s separated runtime components answer different questions. Neither establishes that a harness is a kind of process. [4, 2] *Editorial interpretation.*
- A runtime may be delivered inside an SDK or framework; packaging and execution responsibility are separate axes. [14] *Editorial interpretation.*

Harness vs Workflow

Workflow describes how work proceeds; Harness describes how execution is governed. This is an editorial distinction between responsibilities, not an official two-term taxonomy. A workflow can run within a harness, and one harness can realize several workflows. [19, 11, 12] *Editorial interpretation.*

The unqualified word harness is not exclusive to language models: NIST describes a healthcare test harness that executes and orchestrates testing services. Keep that usage separate from ASD’s specifically agentic-AI scope. [11, 4] *Editorial interpretation.*

Editorial comparison; each cell cites its basis.

Dimension	Workflow	Harness
Main question	What activities run, with which dependencies and control flow? [19]	Which interfaces, permissions, execution policies and checks govern those activities? [4]
LLM required?	No. Steps can involve people, deterministic code, LLM calls or agents. [12]	No for the broader term. An agentic AI harness surrounds an LLM/agent; NIST’s testing usage has another scope. [11, 4]
Order and loops	Sequence, branching, concurrency and iteration describe the process; a workflow need not be a simple linear chain. [19]	May implement a loop or realize workflows; governance does not prescribe one universal step order. [11, 2]
Checks and recovery	Validation and retry branches may be steps in the process. [1, 12]	May enforce validation and retry budgets across calls, hold state and record failures. Exact capabilities must be specified. [4, 12]
Typical design concern	Missing dependencies, unsuitable branches or an incorrect task decomposition. [19]	Excessive permissions, unbounded execution, lost state or inadequate failure records. [4, 2]

Schema extraction: an illustrative experiment

For a paper and dataset, a proposed workflow is candidate discovery -> evidence retrieval -> attribute extraction -> deterministic schema assembly -> validation. Model calls and code can occupy different stages; this describes the task logic. [12] *Editorial interpretation*.

The proposed harness fixes the input snapshot and adapters, checks outputs against the declared JSON Schema dialect, and tracks field-to-evidence support separately. A structurally valid JSON object does not by itself prove that a field exists in the source or that an inferred attribute is true. [3, 12] *Editorial interpretation*.

Example policies include a three-attempt limit, explicit token/time budgets, checkpointing, timeout and truncation handling, and recording model/configuration versions, prompts, outputs and failure reasons. Allow unknown only where the output contract permits it. These are proposed experiment controls, not mandatory properties of every harness. [12] *Editorial interpretation*.

For this experiment, an orchestrator schedules model/replicate/device assignments and task dependencies; the harness supplies the configured runtime and controls. They can overlap: NIST's test harness itself includes orchestration. This is a responsibility split, not three mutually exclusive software boxes. [11, 12] *Editorial interpretation*.

Overlap, reliability and reproducibility

The same extract -> validate -> retry code can play both roles. Describing its branch and loop structure is a workflow view; enforcing a shared attempt budget, validation contract and failure record is a harness view. The distinction follows responsibility, not Python syntax. [12] *Editorial interpretation*.

The harness metaphor suggests bounded, observable and recoverable execution. It does not establish that the model becomes deterministic or always correct, or that every harness actually enforces its advertised policies. An implementation must test those policies and state the limits of its guarantees. [4, 12] *Editorial interpretation*.

Recording inputs, configuration, model versions and seeds supports an auditable experiment setup. It is not a universal promise of identical outputs: OpenAI's archived 2023 example explicitly allows differing responses even when seed, request parameters and system fingerprint match. That provider-specific example is not current API guidance for every model. [16, 12] *Editorial interpretation*.

Operational criteria

Repository test: locate the running component responsible for model invocation, context/tool mediation, execution feedback and termination or privilege policy. State where session storage and tool execution live. If the software only constructs that component, label it a framework or SDK. [12] *Editorial interpretation*.

Concept relationships

Only is-a means a strict subtype. Uses and supports/orchestrates describe composition and control. Harness supports/orchestrates Workflow

A harness realizes or coordinates workflows; runtime-to-process is not strict subtype nesting. [4, 2] *Editorial interpretation.* Harness supports/orchestrates Agent

When it supplies the running loop, tools and execution policy for an agent. [4] *Editorial interpretation.* Harness uses LLM

In an LLM-based agent runtime; evaluation harnesses and other harness usages have different scopes. [4, 14] *Editorial interpretation.*

Repository working definition

Repository / Proposed / not certified No consensus decision recorded. Use agent harness for the model-surrounding execution layer that mediates context, tools, feedback and execution policy. For the proposed experiment, Workflow emphasizes flow and dependencies; Harness emphasizes controls and boundaries. Specify the policies and guarantees actually implemented. [12]

Sources and research gaps

Peer-reviewed definition gap: this review did not establish a directly applicable peer-reviewed consensus definition of agent harness. The conceptual paper is a June 2026 preprint, not promoted to tier 2.

The NIST testing description establishes an additional, non-LLM usage; it does not supply a universal Harness definition. The flow/governance table and experimental example are editorial synthesis and an unratified internal proposal. The OpenAI reproducibility recipe is explicitly archived.

[4] Tier 1 / Institutional security advisory. Context: Organizational security and governance of agentic AI; not a universal taxonomy. Verification: Original text and publication record checked 2026-10-10. [11] Tier 1 / Institutional engineering description. Context: Healthcare interoperability and conformance testing, not a universal definition of every harness. Demonstrates a non-LLM execution and orchestration harness. Verification: Read the original NIST page, including the test-harness/workflow-management descriptions and creation/update dates. [19] Tier 2 / Peer-reviewed journal article. Context: Business workflow control-flow patterns; not an exhaustive theory of every process. Verification: Final published PDF read; peer-review status, journal and DOI checked against the TU/e publication record. [8] Tier 2 / Peer-reviewed conference paper. Context: Software-engineering agent architecture; relevant runtime evidence, not a universal harness definition. Verification: Published NeurIPS proceedings PDF and conference record checked 2026-10-10. [2] Tier 3 / First-party engineering blog. Context: Architecture of a specific hosted agent product. Verification: Original text and publication record checked 2026-10-10. [14] Tier 3 / Preprint; peer review not established. Context: A proposed operational definition in coding-agent engineering; no consensus claim. Verification: Original text and publication record checked 2026-10-10. [1] Tier 3 / First-party engineering blog. Context: Company-specific distinction between predefined LLM workflows and dynamic agents. Verification: Original text and publication record checked 2026-10-10. [3] Tier 3 / First-party specification; IETF Internet-Draft, not an RFC. Context: The JSON Schema dialect and instance data model; assertions and annotations have different roles. Verification: Dated Core text and definitions checked. The document identifies itself as an Internet-Draft with intended Informational status; it is not relabeled a final IETF standard. [16] Tier 3 / Archived first-party engineering recipe. Context: Historical Chat Completions example for 2023 model versions; not a

current feature-support promise for all APIs or a statement about every LLM. Verification: Read the original recipe, its archived warning, date and explicit statement that matching seed, parameters and fingerprint does not guarantee identical outputs. [12] Tier 4 / Repository internal proposal; not certified. Context: Internal proposals and editorial interpretations, not external consensus. Verification: Versions 1.1 and 1.2 add the Schema proposal and the flow/governance experimental example while preserving prior proposal sections. All remain unratified; no adoption decision claimed. Version 1.3 changes branding and certification scope only; it does not certify any proposal.

Revision record

Version 1.2 / Updated 2026-10-12 1.0 / 2026-10-10: Researched and rewritten with attributed definitions, typed relations and an unratified Repository proposal. 1.1 / 2026-10-11: Add the cited flow/governance comparison, overlap boundaries and illustrative experiment; Repository meaning remains proposed. 1.2 / 2026-10-12: Scope the working definition and certification labels to this repository under Open AI Glossary. Citations and proposed status remain unchanged.

Related terms

[Workflow](#); [LLM](#); [Agent](#)

Model

Representation and inference component Model has several scopes: a representation of a subject; an executable representation; or, in AI, the inference component transforming inputs into outputs. State the scope before using the term. [10, 6] *Editorial interpretation.*

Definition comparison

General representation

Institutional; NASA modeling and simulation NASA defines a model as a description or representation of a system, entity, phenomenon or process. It separately defines conceptual and computational models. [10]

Computational representation

Institutional; NASA terminology A computational model is a usable implementation of a conceptual model, including mathematical, numerical, logical or qualitative representations. Executability alone does not specify what is represented. [10] *Editorial interpretation.*

AI inference component

Institutional; AI-system context The EU-U.S. terminology identifies a model as a core AI-system component that infers outputs from inputs. This context is narrower than every possible software function. [6] *Editorial interpretation.*

Scope

A model need not be learned. NASA's modeling-and-simulation sense includes conceptual representations, while the AI terminology concerns an inference component within a system. [10, 6] *Editorial interpretation.*

Includes

- Editorial examples: a conceptual climate description and its numerical simulation fit representation-oriented usage; a trained classifier fits learned inference usage. These are distinct cases under a broad term. [10, 12] *Editorial interpretation.*
- An LLM is one specialized learned model, rather than a synonym for all models. [6] *Editorial interpretation.*

Excludes and counterexamples

- "Any input -> output" is insufficient as a universal definition: it omits what the artifact represents and collapses ordinary utilities into model terminology. We retain it only as an internal proposal. [10, 12] *Editorial interpretation.*

- An application containing a model, orchestration and tools is a system; the model name should identify its inference component. [6, 12] *Editorial interpretation.*

Different usages and open boundaries

- In learned-model usage, identify the fitted parameters and inference rule; in conceptual-model usage, identify the referent and assumptions. Do not silently require training for all models. [10, 12] *Editorial interpretation.*
- A fixed formula may be a model when it represents a stated phenomenon. Being deterministic neither establishes nor prevents that role. [10] *Editorial interpretation.*

Operational criteria

Repository test: record the intended sense, represented subject or inference task, assumptions, inputs and outputs if executable, and whether parameters were learned. For learned models, distinguish the checkpoint from deployment and orchestration. [12] *Editorial interpretation.*

Concept relationships

Only is-a means a strict subtype. Uses and supports/orchestrates describe composition and control. LLM is-a Model

Within learned AI-model usage: an LLM is a specialized language model, hence a Model. [6] *Editorial interpretation.* Agent uses Model

When a computational model supplies inference or action selection; this is a component relation. [15, 12] *Editorial interpretation.*

Repository working definition

Repository / Proposed / not certified No consensus decision recorded. Default to a representation or inference component with an explicit subject or task. Use "learned model" when training matters. The broader "any input-to-output mapping" guide remains a pending alternative, not our universal definition. [12]

Sources and research gaps

Scope gap: NASA's general representation terminology applies to modeling and simulation; it does not settle every AI use. The EU-U.S. AI meaning and the internal mapping proposal are shown separately.

[10] Tier 1 / Institutional technical standard. Context: Models and simulations supporting NASA activities. Verification: Final B PDF checked. Portal historical Change Date is inconsistent with its B document date; this record follows the PDF cover, not that legacy field. [6] Tier 1 / Institutional terminology document. Context: AI-system terminology for transatlantic risk-management and standards cooperation. Verification: Original text and publication record checked 2026-10-10. [12] Tier 4 / Repository internal proposal; not certified. Context: Internal proposals and editorial

interpretations, not external consensus. Verification: Versions 1.1 and 1.2 add the Schema proposal and the flow/governance experimental example while preserving prior proposal sections. All remain unratified; no adoption decision claimed. Version 1.3 changes branding and certification scope only; it does not certify any proposal.

Revision record

Version 1.1 / Updated 2026-10-12 1.0 / 2026-10-10: Researched and rewritten with attributed definitions, typed relations and an unratified Repository proposal. 1.1 / 2026-10-12: Scope the working definition and certification labels to this repository under Open AI Glossary. Citations and proposed status remain unchanged.

Related terms

[LLM](#); [Agent](#)

LLM

Large language model A large language model is a deep learning language model trained on very large textual datasets. It is a kind of Model; its complete application or agent runtime is a separate system. [6, 12] *Editorial interpretation.*

Definition comparison

EU-U.S. terminology

Institutional; language models The second edition emphasizes deep learning and extremely large text datasets, and includes models that generate text or form representations useful for downstream tasks. [6]

GPT-3 study

Peer-reviewed; a specific experimental scale Brown and colleagues study an autoregressive model with 175 billion parameters alongside smaller models. Its training corpus mixes web sources, books and Wikipedia; that recipe is an example, not a definition. [17]

Scope

This page concerns learned language models, including text generation and representation uses. The word "large" needs a stated scale and comparison context. [6, 17] *Editorial interpretation.*

Includes

- A trained text-generation checkpoint is a model artifact; adding retrieval, tools and execution control creates a larger application around it. [1, 12] *Editorial interpretation.*
- Representation-producing language models are within the cited institutional usage; generation is not its only permitted behavior. [6]

Excludes and counterexamples

- A corpus alone is training material, not a trained model. A tool-connected assistant is not identical to the language model inside it. [12] *Editorial interpretation.*
- No universal parameter cutoff is established by the sources used here. Report parameters, training tokens and the relevant comparison rather than inventing a threshold. [17, 12] *Editorial interpretation.*

Different usages and open boundaries

- The institutional definition foregrounds textual training. For a multimodal product, specify its language-model component and additional modalities instead of assuming the same definition covers the whole system. [6, 12] *Editorial interpretation.*

- Training a model and conditioning an inference call are different operations: the GPT-3 few-shot experiments add examples in context without gradient updates. [17]

Operational criteria

Repository test: identify the learned model artifact, language task and objective, available training-material evidence, and reported scale. Record unknown proprietary details as unknown. Separately identify any tokenizer, serving system, tools and controller. [12] *Editorial interpretation.*

Concept relationships

Only is-a means a strict subtype. Uses and supports/orchestrates describe composition and control. Workflow uses LLM

An LLM workflow uses an LLM for one or more tasks; many workflows use no language model. [1] *Editorial interpretation.* Harness uses LLM

In an LLM-based agent runtime; evaluation harnesses and other harness usages have different scopes. [4, 14] *Editorial interpretation.* LLM is-a Model

Within learned AI-model usage: an LLM is a specialized language model, hence a Model. [6] *Editorial interpretation.* Agent uses LLM

For LLM-based agents only; general agents need not use a language model. [15, 1] *Editorial interpretation.*

Repository working definition

Repository / Proposed / not certified No consensus decision recorded. Use LLM for a learned language-model component at a documented large scale. Avoid reducing the term to corpus size alone or using it as the name for a tool-using system. No minimum parameter count is ratified by the repository. [12]

Sources and research gaps

Scale gap: the institutional definition supplies qualitative scale, not a universal threshold; a concrete GPT-3 size cannot settle the category for all architectures.

[6] Tier 1 / Institutional terminology document. Context: AI-system terminology for transatlantic risk-management and standards cooperation. Verification: Original text and publication record checked 2026-10-10. [17] Tier 2 / Peer-reviewed conference paper. Context: GPT-3 experiments and corpus; a concrete example, not a universal LLM definition. Verification: Original text and publication record checked 2026-10-10. [1] Tier 3 / First-party engineering blog. Context: Company-specific distinction between predefined LLM workflows and dynamic agents. Verification: Original text and publication record checked 2026-10-10. [12] Tier 4 / Repository internal proposal; not certified. Context: Internal proposals and editorial interpretations, not external consensus. Verification: Versions 1.1 and 1.2 add the Schema proposal and the flow/governance experimental example while preserving prior proposal sections. All remain unratified; no adoption decision

claimed. Version 1.3 changes branding and certification scope only; it does not certify any proposal.

Revision record

Version 1.1 / Updated 2026-10-12 1.0 / 2026-10-10: Researched and rewritten with attributed definitions, typed relations and an unratified Repository proposal. 1.1 / 2026-10-12: Scope the working definition and certification labels to this repository under Open AI Glossary. Citations and proposed status remain unchanged.

Related terms

[Harness](#); [Model](#); [Agent](#)

Agent

Goal-directed action with feedback In general computing, an agent is a situated system with some control over its actions, responsiveness to its environment and goal-directed behavior. LLM-based agents are one implementation, not the whole category. [9, 12] *Editorial interpretation.*

Definition comparison

Classical weak agency

Peer-reviewed; general computing Wooldridge and Jennings characterize a weak notion through autonomy, social ability, reactivity and pro-activeness. This influential characterization is attributed, not presented as the only universal test. [9]

Institutional project scoping

NIST draft; identity and authorization The NCCoE concept paper describes autonomous decisions and actions under limited human supervision toward goals. It is a draft project-scoping document, not a final general-agent standard. [7]

LLM engineering

First-party; architectural distinction Anthropic distinguishes predefined workflows from agents whose LLM dynamically directs the process and tool use. Its usage is narrower than general agent theory. [1]

Scope

Begin with general software or hardware agency. For LLM engineering, explicitly identify the model, action interface and feedback loop; access to a tool alone does not settle agency. [9, 1] *Editorial interpretation.*

Includes

- Editorial example: a goal-directed robot controller can be an agent without an LLM. A software agent can act through commands in a virtual environment. [9, 12] *Editorial interpretation.*
- Bounded autonomy can coexist with approval gates. Record which decisions the system controls and which require a person. [7, 12] *Editorial interpretation.*

Excludes and counterexamples

- A bare text-completion call does not, by itself, show goal-directed interaction with an environment under our test. [12] *Editorial interpretation.*
- A fixed tool pipeline is not automatically an agent in the dynamic-control engineering sense, even if it contains an LLM and an API. [1] *Editorial interpretation.*

Different usages and open boundaries

- Planning, long-term memory and self-learning are not universal prerequisites in our working definition. Capability claims should be tested separately from the name Agent. [9, 12] *Editorial interpretation*.
- We retain autonomy, goal direction, environment feedback and action as Repository criteria. Social ability is explicit in the classical weak notion, but is not a mandatory gate in our narrower operational proposal. [9, 12] *Editorial interpretation*.

LLM-based agents

An LLM-based agent places a language model within an action-and-observation system. ReAct interleaves reasoning with actions that obtain external information, allowing later reasoning to respond to outcomes. It is one architecture, not a requirement that every agent expose a chain of thought. [15] *Editorial interpretation*.

Distinguish choosing an action, granting permission and executing it. A model’s proposed tool call does not itself execute a tool; the surrounding runtime provides that capability. [2, 12] *Editorial interpretation*.

Operational criteria

Repository test: identify a goal, observations, an action interface, the decisions controlled by the system, and how outcomes affect later actions. For an LLM-based agent, trace a model-selected action, its execution and returned feedback. State stopping and approval conditions. [12] *Editorial interpretation*.

Concept relationships

Only is-a means a strict subtype. Uses and supports/orchestrates describe composition and control. Workflow uses Agent

A designed workflow may delegate an activity to an agent; this is composition, not a claim that every workflow is agentic. [12] *Editorial interpretation*. Harness supports/orchestrates Agent

When it supplies the running loop, tools and execution policy for an agent. [4] *Editorial interpretation*. Agent uses LLM

For LLM-based agents only; general agents need not use a language model. [15, 1] *Editorial interpretation*. Agent uses Model

When a computational model supplies inference or action selection; this is a component relation. [15, 12] *Editorial interpretation*.

Repository working definition

Repository / Proposed / not certified No consensus decision recorded. Use Agent for a system with bounded autonomy, a goal, environment feedback and an ability to act. Use LLM-based agent when

an LLM participates in selecting actions. The agent uses a model; it is not classified as an LLM subtype. [12]

Sources and research gaps

Institutional definition gap: the NIST material checked is a draft identity/authorization concept paper. No final, general-agent institutional definition was established in this review; the classical peer-reviewed account remains central.

[9] Tier 2 / Peer-reviewed journal article; author-hosted text. Context: General software/hardware agency, predating LLM engineering. Verification: Author HTML checked; its 1994 conversion timestamp is not the publication year. Journal issue and DOI verified at Cambridge. [15] Tier 2 / Peer-reviewed conference paper; author manuscript. Context: An LLM reasoning/action architecture; not a definition of every possible agent. Verification: ICLR publication header and arXiv v3 camera-ready record checked 2026-10-10. [2] Tier 3 / First-party engineering blog. Context: Architecture of a specific hosted agent product. Verification: Original text and publication record checked 2026-10-10. [7] Tier 3 / Institutional draft concept paper. Context: Identity and authorization project scoping. Official draft, not a final general-agent definition. Verification: Original text and publication record checked 2026-10-10. [1] Tier 3 / First-party engineering blog. Context: Company-specific distinction between predefined LLM workflows and dynamic agents. Verification: Original text and publication record checked 2026-10-10. [12] Tier 4 / Repository internal proposal; not certified. Context: Internal proposals and editorial interpretations, not external consensus. Verification: Versions 1.1 and 1.2 add the Schema proposal and the flow/governance experimental example while preserving prior proposal sections. All remain unratified; no adoption decision claimed. Version 1.3 changes branding and certification scope only; it does not certify any proposal.

Revision record

Version 1.1 / Updated 2026-10-12 1.0 / 2026-10-10: Researched and rewritten with attributed definitions, typed relations and an unratified Repository proposal. 1.1 / 2026-10-12: Scope the working definition and certification labels to this repository under Open AI Glossary. Citations and proposed status remain unchanged.

Related terms

[Workflow](#); [Harness](#); [Model](#); [LLM](#)

Schema

Data schema A schema describes how data is organized and what its fields mean, with structural or value constraints where specified. State the data model and dialect: the exact obligations depend on the schema language. [5, 18, 3] *Editorial interpretation.*

Definition comparison

Institutional terminology

CNSSI 4009-2022, reproduced by NIST CSRC; information-security context CNSSI describes an object through its organization, field syntax, meanings and associated metadata. This is broader than a list of column names. [5]

XML validation

XSD 1.1; W3C Recommendation XSD separates abstract schema components from their XML schema documents; a validator applies a conforming schema to an XML instance. [18]

JSON description

Draft 2020-12 Core; first-party Internet-Draft JSON Schema describes an instance using an object or boolean schema. Assertions constrain instances; annotations supply information for applications. [3]

Data-derived summary

Goldman and Widom, VLDB 1997; semistructured databases DataGuides provide structural summaries generated from a database, serving as dynamic schemas for browsing and querying. Their scope is observed structure in that database. [13]

Scope

Here, schema means a data-description artifact used for interpretation or validation. Psychological schemata and prediction models are outside this entry. [5, 12] *Editorial interpretation.*

Includes

- Named fields, types, permitted structure and explicit constraints; meanings or metadata may also be documented. A schema need not specify every aspect of domain interpretation. [5, 3] *Editorial interpretation.*

Excludes and counterexamples

- A particular record is a data instance, not the rules applied to it. For example, {"temperature": 23.5} supplies a value; a schema separately describes the property and its constraints. [3] *Editorial interpretation.*
- A validator is software that applies a schema. The schema itself is neither that execution nor an entire workflow. [18, 12] *Editorial interpretation.*

Different usages and open boundaries

- Schema may mean declared requirements or a summary inferred from existing data. A DataGuide follows the database; a declared contract can reject data that fails its requirements. Label which direction is intended. [13, 18] *Editorial interpretation.*
- The schema/instance distinction is a role distinction: a JSON schema is itself an instance when checked against a meta-schema. JSON syntax alone does not identify that role. [3]

Relation to schema extraction

Schema extraction is the process; a recovered or inferred schema is its output. DataGuides illustrate inference from instances. Recovering an explicitly declared schema is a different evidence task from generalizing observed records. [13, 12] *Editorial interpretation.*

For Repository work, retain source-to-field evidence and label observed, declared and inferred constraints. An absent field in a sample does not by itself establish that the field is forbidden; a structural summary alone does not prove complete intended semantics. [13, 12] *Editorial interpretation.*

Operational criteria

Identify the described population, schema version/dialect, fields and constraints; distinguish declared rules from observed patterns and unknowns. Show how an example instance is interpreted or checked without treating that example as evidence for every possible record. [12] *Editorial interpretation.*

Concept relationships

Only is-a means a strict subtype. Uses and supports/orchestrates describe composition and control. Schema supports/orchestrates Workflow

Supports a workflow at data-interpretation or validation boundaries; it supplies a description or contract, not the executing control flow. This is a supporting-artifact relation, not a subtype. [18, 12] *Editorial interpretation.*

Repository working definition

Repository / Proposed / not certified No consensus decision recorded. Proposed Repository meaning: an explicit, versioned description of data structure, field meanings and stated constraints. Record

whether it is declared, recovered from a specification, or inferred from observed data, together with evidence and unresolved coverage. [12] *Editorial interpretation.*

Sources and research gaps

The applicable institutional definition was verified through the NIST reproduction; the linked original CNSSI document was not retrievable in this review.

The peer-reviewed DataGuides example supports one semistructured-data approach. This review does not establish a universal definition of schema extraction across paper, database and metadata workflows.

[5] Tier 1 / Institutional terminology; source-attributed glossary reproduction. Context: Object organization, field syntax, semantics and metadata in information-security terminology. Verification: NIST reproduction and its CNSSI attribution checked. The linked CNSS original could not be retrieved; no claim of direct review or a universal NIST-preferred definition. [13] Tier 2 / Peer-reviewed conference paper. Context: Data-derived structural summaries for the Lore semistructured database; one method, not a universal definition of extraction. Verification: Conference PDF abstract/introduction and proceedings publication record checked. The search engine crawl date is not the publication date. [18] Tier 3 / W3C Recommendation. Context: Schemas and schema-validity assessment for XML; not a definition of all research metadata. Verification: Dated Recommendation and cited sections checked, rather than an earlier Working Draft. [3] Tier 3 / First-party specification; IETF Internet-Draft, not an RFC. Context: The JSON Schema dialect and instance data model; assertions and annotations have different roles. Verification: Dated Core text and definitions checked. The document identifies itself as an Internet-Draft with intended Informational status; it is not relabeled a final IETF standard. [12] Tier 4 / Repository internal proposal; not certified. Context: Internal proposals and editorial interpretations, not external consensus. Verification: Versions 1.1 and 1.2 add the Schema proposal and the flow/governance experimental example while preserving prior proposal sections. All remain unratified; no adoption decision claimed. Version 1.3 changes branding and certification scope only; it does not certify any proposal.

Revision record

Version 1.1 / Updated 2026-10-12 1.0 / 2026-10-10: Initial cited Schema entry; separates data instances, validation and extraction. Repository meaning is an unratified editorial proposal. 1.1 / 2026-10-12: Scope the working definition and certification labels to this repository under Open AI Glossary. Citations and proposed status remain unchanged.

Related terms

[Model](#); [Workflow](#)

Repository terminology guide

Version 1.3 / 2026-10-12 **Repository** / **Proposed** / **not certified** No consensus decision recorded.

Workflow and Harness

Initial proposal: Workflow is a general arrangement of work; deterministic Python code as well as LLM-based work can participate in a workflow.

Initial proposal: a harness is a workflow designed for an LLM or agent. Editorial revision for discussion: distinguish the execution layer from the process it supports. This edition uses supports/orchestrates rather than declaring a Harness is-a Workflow relationship.

Model and LLM

Initial proposal: any input-to-output mapping may be called a model. This remains an intentionally broad, unratified interpretation, not a universal external definition.

Working revision: specify whether model means a representation, an executable representation, or a learned inference component. An LLM is a learned language model; its training, scale and scope should be described rather than reduced to an arbitrary parameter threshold.

Agent

Initial proposal: an agent builds on an LLM and external tools, with its distinguishing attributes still to be decided.

Working revision: reserve the general term for systems with bounded autonomy, goal-directed behavior, environmental feedback and an ability to act. Describe an LLM-based agent as one particular implementation using an LLM. Planning, persistent memory and self-learning are optional capabilities, not automatic universal requirements.

Editorial and adoption policy

Check external evidence in the fixed order: applicable institutional definitions, peer-reviewed papers, industry standards or first-party documents or preprints or academic materials, then internal proposals. Preserve uncertainty and differing scopes.

Use cited statements, typed relations and a quiet reading layout. Strict nesting is reserved for supported is-a relations. All proposals in this guide are unratified; only a documented consensus decision may authorize green adoption tags.

Schema: an unratified editorial extension

Proposed extension: use schema for an explicit, versioned description of data structure, field meanings and stated constraints. Keep the data instance, the description and the validator separate; the same document can occupy different roles under a meta-schema.

Schema extraction is a process that recovers declarations or infers a description from evidence. Record the input population, source-to-field evidence, dialect/version, observed versus declared or inferred constraints, and unresolved coverage. A sample is not proof of every permitted record or intended semantic rule.

Workflow and Harness: flow and governance proposal

Proposed distinction: Workflow describes how work proceeds, including sequence, dependencies, branches and loops. Harness describes how execution is governed through interfaces, policies, validation, state, recovery and observability. These are overlapping responsibilities, not mutually exclusive categories or a universal subtype taxonomy.

Illustrative schema-extraction workflow: candidate discovery, evidence retrieval, attribute extraction, deterministic schema assembly and validation. This is a proposed teaching example, not a report of an implemented or evaluated Repository system.

Proposed experimental harness policies: standardize input snapshots and model adapters, bound attempts and resource consumption, permit an explicit unknown value only where the declared schema allows it, distinguish evidence support from structural validation, checkpoint progress and record inputs, model/configuration versions, prompts, outputs and failure reasons.

For this example, orchestration assigns models, replicates, devices and dependencies. An experimental harness supplies the configured execution and recording environment. These roles may live in the same implementation. Recording experimental conditions helps audit and repeat a setup; it does not establish that every output will be identical or correct.

Revision record

1.0 / 2026-10-10: Initial internal proposal, summarized for public citation. Editorial revisions are proposals; no consensus decision recorded. 1.1 / 2026-10-10: Add the Schema editorial proposal while preserving every 1.0 proposal section. No consensus decision recorded. 1.2 / 2026-10-11: Add the flow/governance comparison and illustrative schema-extraction experiment. Preserve prior proposal sections; no consensus decision recorded. 1.3 / 2026-10-12: Rename the project Open AI Glossary and scope certification to this repository. All working meanings remain proposed; no new consensus decision recorded.

Build provenance

Source content SHA-256: ad3e8205c5b6afd415421e877f04207ddeb883548c6ffc6eb222f4fc56c72af5.
Source commit: f8502528110058957d5c40e2f6470d91d1d56676. Build run: <https://github.com/williamQ96/plale-lab-glossary/actions/runs/38326869019>.

References

- [1] Anthropic. Building effective agents. First-party engineering blog, 2024. 19 December 2024; live page includes subsequent tooling updates. What are agents?; Building blocks, workflows, and agents. Accessed 2026-10-10. <https://www.anthropic.com/engineering/building-effective-agents>.
- [2] Anthropic. Scaling Managed Agents: Decoupling the brain from the hands. First-party engineering blog, 2026. 8 April 2026. Session, harness and sandbox architecture. Accessed 2026-10-10. <https://www.anthropic.com/engineering/managed-agents>.
- [3] Austin Wright, Henry Andrews, Ben Hutton and Greg Dennis. JSON Schema: A Media Type for Describing JSON Documents. First-party specification; IETF Internet-Draft, not an RFC, 2022. Draft 2020-12 Core, draft-bhutton-json-schema-01, 16 June 2022. Sections 4.2, 4.3, 4.3.1 and 4.3.4: instance, schema, keywords and meta-schema. Accessed 2026-10-10. <https://json-schema.org/draft/2020-12/json-schema-core>.
- [4] Australian Signals Directorate. Agentic AI harnesses. Institutional security advisory, 2026. Published and reviewed 11 September 2026. What is a harness; Model vs harness. Accessed 2026-10-10. <https://www.cyber.gov.au/advisory/agentic-ai-harnesses>.
- [5] Committee on National Security Systems; NIST CSRC glossary reproduction. CNSSI 4009-2022: schema (as reproduced by NIST CSRC). Institutional terminology; source-attributed glossary reproduction, 2022. CNSSI 4009-2022, NIST term record accessed 10 October 2026. NIST CSRC schema entry, attributed definition and source link. Accessed 2026-10-10. <https://csrc.nist.gov/glossary/term/schema>.
- [6] EU-U.S. Trade and Technology Council, Working Group 1. EU-U.S. Terminology and Taxonomy for Artificial Intelligence, Second Edition. Institutional terminology document, 2024. Second edition, prepared for the sixth TTC ministerial in 2024. Printed page 12, Model and Large language model; page 8, Machine Learning. Accessed 2026-10-10. <https://www.nist.gov/document/eu-us-terminology-and-taxonomy-artificial-intelligence-second-edition>.
- [7] Harold Booth, Bill Fisher, Ryan Galluzzo and Joshua Roberts; NIST NCCoE. Accelerating the Adoption of Software and AI Agent Identity and Authorization. Institutional draft concept paper, 2026. DRAFT, February 2026; public comment 5 February-2 April 2026. Challenge Overview, printed page 5, lines 54-58. Accessed 2026-10-10. <https://www.nccoe.nist.gov/sites/default/files/2026-02/accelerating-the-adoption-of-software-and-ai-agent-identity-and-authorization-concept-paper.pdf>.
- [8] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan and Ofir Press. SWE-agent: Agent-Computer Interfaces Enable Automated Soft-

- ware Engineering. Peer-reviewed conference paper, 2024. Advances in Neural Information Processing Systems 37, NeurIPS 2024; DOI 10.52202/079017-1601. Abstract and agent-computer interface architecture. Accessed 2026-10-10. https://proceedings.neurips.cc/paper_files/paper/2024/file/5a7c947568c1b1328ccc5230172e1e7c-Paper-Conference.pdf.
- [9] Michael Wooldridge and Nicholas R. Jennings. Intelligent Agents: Theory and Practice. Peer-reviewed journal article; author-hosted text, 1995. The Knowledge Engineering Review 10(2), June 1995, 115-152; DOI 10.1017/S0269888900008122. What is an agent? A Weak Notion of Agency; stronger notions and other attributes. Accessed 2026-10-10. <https://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/ker95/ker95-html.html>.
 - [10] NASA. Standard for Models and Simulations. Institutional technical standard, 2024. NASA-STD-7009B, 5 March 2024. Section 3.2, Model on printed page 15; Computational Model and Conceptual Model on page 12. Accessed 2026-10-10. <https://standards.nasa.gov/sites/default/files/standards/NASA/B/1/NASA-STD-7009B-Final-3-5-2024.pdf>.
 - [11] NIST Information Technology Laboratory. High-Level Requirements: Health IT Test Infrastructure. Institutional engineering description, 2016. Created July 14, 2016; updated April 8, 2026. High-Level Requirements; Test Infrastructure System Components, Test Harness Component. Accessed 2026-10-11. <https://www.nist.gov/itl/high-level-requirements>.
 - [12] Repository internal proposal. Repository terminology guide. Repository internal proposal; not certified, 2026. Repository proposal 1.3 / not certified. Term-specific proposal sections and editorial policy. Accessed 2026-10-12. [Internal proposal reproduced in this edition](#).
 - [13] Roy Goldman and Jennifer Widom. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. Peer-reviewed conference paper, 1997. Proceedings of the 23rd VLDB Conference, 1997, pp. 436-445. Abstract and section 1, printed p. 436; section 2, DataGuide definitions. Accessed 2026-10-10. <https://www.vldb.org/conf/1997/P436.PDF>.
 - [14] Sanderson Oliveira de Macedo. What makes a harness a harness: necessary and sufficient conditions for an agent harness. Preprint; peer review not established, 2026. arXiv:2606.10106v1, 8 June 2026. Abstract; conceptual boundaries and inclusion/exclusion test. Accessed 2026-10-10. <https://arxiv.org/abs/2606.10106v1>.
 - [15] Shunyu Yao and others. ReAct: Synergizing Reasoning and Acting in Language Models. Peer-reviewed conference paper; author manuscript, 2023. ICLR 2023 camera-ready manuscript; arXiv:2210.03629v3, 10 March 2023. Abstract and section 2; ICLR 2023 publication header. Accessed 2026-10-10. <https://arxiv.org/pdf/2210.03629v3>.
 - [16] Shyamal Anadkat / OpenAI. How to make your completions outputs consistent with the new seed parameter. Archived first-party engineering recipe, 2023. November 6, 2023; archived recipe. Implementing consistent outputs; seed and system_fingerprint. Accessed 2026-10-11. https://developers.openai.com/cookbook/examples/reproducible_outputs_with_the_seed_parameter.
 - [17] Tom B. Brown and others. Language Models are Few-Shot Learners. Peer-reviewed conference paper, 2020. Advances in Neural Information Processing Systems 33, 2020. Abstract; sections 2.1-2.2, model sizes and training datasets. Accessed 2026-10-10. <https://papers.nips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf>.

- [18] W3C; Shudi Gao, C. M. Sperberg-McQueen and Henry S. Thompson (editors). W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures. W3C Recommendation, 2012. W3C Recommendation, 5 April 2012. Sections 2.2, 2.4, 2.5 and 2.7: components, validators, validity and XML representations. Accessed 2026-10-10. <https://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/>.
- [19] Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, Bartek Kiepuszewski and Alistair P. Barros. Workflow Patterns. Peer-reviewed journal article, 2003. Distributed and Parallel Databases 14(1), 5-51; DOI 10.1023/A:1022883727209. Section 2, printed page 9; static-control-flow scope on page 8. Accessed 2026-10-10. <https://pure.tue.nl/ws/files/2053121/613310.pdf>.
- [20] Workflow Management Coalition. Workflow and Internet: Catalysts for Radical Change. Institutional white paper, 1998. June 1998. Page 8, Workflow Assists Business Processes. Accessed 2026-10-10. https://wfmc.org/wp-content/uploads/2022/09/Workflow_Internet_catalysts_for_change.pdf.
- [21] Workflow Management Coalition. Terminology & Glossary. Institutional terminology specification, 1999. WfMC-TC-1011, Issue 3.0, February 1999. Page 8, Workflow; process definition and enactment usage. Accessed 2026-10-10. https://wfmc.org/wp-content/uploads/2022/09/TC-1011_term_glossary_v3.pdf.