THE DEVELOPER'S REFERENCE

What is AI?

Artificial Intelligence Explained

Types · Generative AI · Agentic AI · Frameworks
Platform Comparisons · Architecture · 2026 State of AI

7
AI Types

8
Leading Platforms

9
Agentic Frameworks

Curated by Angela Guilherme

Edition 2026 · Revised October 2026

Abstract. The term “artificial intelligence” now covers everything from a spam filter to autonomous software agents that plan and execute multi-step work. This handbook gives precise, developer-oriented definitions of the field and its principal types; compares the leading generative AI platforms as of October 2026; explains the mechanics of generative, predictive and agentic systems; surveys the agent frameworks and interoperability protocols in production use; and closes with a section on governance and a practical decision guide. This revision replaces the platform, model and framework information of the original 2026 edition, which had been overtaken by several release cycles.

Contents

1. Core Definition 1
2. Main Types of AI 2
3. Generative AI Platform Comparison 2
4. Generative and Predictive AI: Examples 3
5. Agentic AI: Multi-Step Reasoning 4
6. Agentic AI Frameworks (2026) 5
7. Framework Comparison 6
8. Governance, Safety and Regulation 6
9. Decision Guide 6
10. Conclusion 7
Glossary 7
References and Sources 7

1. Core Definition

Artificial intelligence (AI) is the design of computer systems that perform functions we associate with human cognition: understanding language, recognising images, making decisions, solving problems and learning from experience. The field spans seven decades, from the symbolic and rule-based expert systems of the 1980s, through statistical machine learning in the 2000s and deep learning in the 2010s, to the transformer-based large language models (LLMs) introduced by Vaswani et al. (2017) that dominate the present era.

Unlike traditional programming, in which developers write explicit rules for every scenario, AI systems *learn* patterns from data. Given enough examples, a model generalises those patterns to inputs it has not seen. This shift from hand-crafted rules to learned representations is what defines modern machine learning and deep learning.

In 2026 the term most often denotes systems built on neural networks trained on very large corpora. Three developments distinguish the current period from the one captured in earlier editions of this handbook. First, **reasoning models** now spend additional computation at inference time, producing extended internal chains of thought before answering. Second, **multimodal** models accept and produce text, images, audio, video and code within a single system. Third, **agentic** systems wrap models in loops of planning, tool use and observation, so that software can carry out multi-step tasks with limited supervision. Context windows of roughly one million tokens are now standard among frontier models [3].

2. Main Types of AI

AI is classified along several independent axes, and a given system belongs to one category on each. Classification by **capability** asks how general the system is; classification by **function** asks how it uses memory and state; classification by **technique** asks what the system produces or does. The six types highlighted in the original edition are retained below and placed on these axes (Table 1).

Table 1. *Principal types of AI by classification axis.*

Type	Axis	Definition	Examples (2026)
Narrow AI (weak AI)	Capability	Task-specific systems that excel in one defined domain. Almost all deployed AI is narrow, including today's LLMs, which are broad in language yet remain bounded systems.	Voice assistants, recommendation engines, spam filters, image classifiers, fraud models
General AI (AGI)	Capability	Hypothetical systems matching human performance across essentially all cognitive tasks. Not achieved; definitions and timelines remain contested.	None; an active research and policy debate
Reactive AI	Function	No memory; responds only to the current input using fixed rules or trained weights.	IBM Deep Blue; simple rule-based classifiers
Limited-memory AI	Function	Uses recent history or context to improve decisions. Most modern ML systems fall here.	Self-driving perception stacks; chatbots within a session
Generative AI	Technique	Models that create new content (text, images, audio, video, code) by learning the statistical structure of training data.	Transformer LLMs; diffusion and video models
Predictive AI	Technique	Models that forecast outcomes from historical patterns, often with gradient-boosted trees, time-series or regression models.	Credit scoring, demand planning, churn prediction
Agentic AI	Technique	Autonomous systems that decompose goals, act through tools and APIs, observe results and iterate with minimal human input.	Coding agents, research agents, browser and computer-use agents

Two further categories from classical taxonomies, *theory-of-mind* AI and *self-aware* AI, remain theoretical and are not discussed further. The distinction between generative and predictive AI is increasingly blurred in practice, since an LLM can be used to classify or forecast, while predictive models are often embedded inside generative workflows.

3. Generative AI Platform Comparison

Table 2 compares the principal generative AI platforms as of early October 2026. Release cadence is now measured in weeks, and published specifications differ between secondary sources; the figures below are therefore approximate and should be confirmed against vendor documentation before any procurement or architectural decision [3, 4]. Context windows are given in tokens (K = thousand, M = million).

Table 2. *Leading generative AI platforms, October 2026.*

Platform (developer)	Current model families	Context	Distinguishing strengths	Typical uses
ChatGPT / GPT (OpenAI)	GPT-6 Astra (flagship, staged rollout from September 2026); GPT-5.6 Sol, Terra, Luna (three price/capability tiers); GPT-5.5; open-weight gpt-oss	About 1M (GPT-5.6: 1.05M, 128K output)	Broad general ability, agentic coding, computer use, tool ecosystem (Responses API, Agents SDK), voice and image	General assistance, coding, customer support, document and data analysis
Claude (Anthropic)	Claude Fable 5.1, Opus 5.5, Sonnet 5.5, Haiku 5.5; the Mythos tier (Claude Mythos Preview) is restricted to selected organisations	Up to about 1M	Coding and agentic reliability, long-document work, instruction-following, writing quality, safety focus; native MCP support	Software engineering, legal and research documents, enterprise and safety-critical use
Gemini (Google DeepMind)	Gemini 3.1 Pro and Ultra; Gemini 3.x Flash tiers; Deep Think reasoning mode	About 1M (2M on some tiers)	Native multimodality (text, image, audio, video, code), Google Workspace and Cloud integration, strong price-to-performance	Video and audio analysis, Workspace automation, very long documents

Platform (developer)	Current model families	Context	Distinguishing strengths	Typical uses
Grok (xAI)	Grok 4.6 (flagship), Grok 4.3, Grok 4.20	500K (4.6); 1M (4.3); 2M (4.20)	Real-time access to X data, competitive reasoning at low API price, long-running agentic work	Current events, social-media analysis, research with live data
Llama (Meta)	Llama 4 Scout, Llama 4 Maverick (open weights)	Scout: 10M; Maverick: up to 1M	Open weights, self-hostable, fine-tunable, mixture-of-experts design; the longest published context window	On-premise and privacy-first deployment, domain fine-tuning
DeepSeek	DeepSeek V4 family (open weights)	About 1M	Low-cost, high-capability open models, strong reasoning and coding	Cost-sensitive inference, self-hosting, research
Mistral (Mistral AI)	Mistral Large, Medium and small open-weight models	Varies by model	Efficient models, multilingual strength, function calling, European data-sovereignty positioning	European deployments, multilingual applications, cost-optimised APIs
Perplexity	Answer engine routing across several third-party and in-house models	Not model-specific	Real-time web search with cited answers; accuracy and provenance over free generation	Research, cited question answering, competitive intelligence

Three structural observations follow from the table. First, the frontier has converged on seven-figure context windows, so model selection now turns more on reliability over long agentic runs and on pricing structure than on raw window size; some providers apply surcharges above a threshold (for example, requests above 272K input tokens on GPT-5.6, or above 200K on Grok 4.6) [3]. Second, long context does not guarantee uniform recall: models attend more reliably to the beginning and end of an input than to its middle. Third, open-weight models (Llama, DeepSeek, Mistral, gpt-oss) have narrowed the gap to proprietary systems on many benchmarks, which makes self-hosting a credible option where data control is paramount.

4. Generative and Predictive AI: Examples

4.1 Generative AI

Table 3. *Generative AI by output modality.*

Modality	Representative systems and typical products
Text	ChatGPT, Claude, Gemini, Grok and Llama-based assistants producing essays, summaries, email, analysis and answers from a prompt.
Images	Midjourney, Stable Diffusion, and image models built into the major assistants, creating photorealistic images, illustrations and concept art from text.
Video	OpenAI’s Sora, Google’s Veo and Runway’s Gen series, producing short clips from text or image prompts, increasingly with synchronised audio.
Audio and music	ElevenLabs for synthetic speech and voice cloning; Suno and Udio for full songs from genre and mood descriptions.
Code	GitHub Copilot, Cursor, Claude Code and Codex, which complete, refactor and now autonomously implement features across repositories.
3D and design	Meshy, Luma AI and similar tools generating 3D assets and scenes from text or images for games and AR/VR.

4.2 Predictive AI

Table 4. *Predictive AI in practice.*

Application	Description
Fraud detection	Payment processors score each transaction against learned behavioural patterns and block anomalies in real time.
Sales and churn forecasting	CRM platforms estimate pipeline close probability, quarterly revenue and per-customer churn risk.
Medical risk and diagnostics	Imaging and risk models flag early-stage disease for clinician review; AlphaFold-style structure prediction supports drug discovery.

Application	Description
Demand planning	Retailers and logistics firms forecast inventory by product, location and season, reducing both stock-outs and overstock.
Recommendation	Netflix, Spotify and TikTok use collaborative filtering and sequence models to predict what each user will engage with next.

The predictive pipeline is conceptually simple: historical data (sales history, behaviour, sensor readings, transactions, market signals) is used to train a model, which then outputs a calibrated forecast such as a fraud probability, a revenue estimate or a demand figure. Its defining property, compared with generative AI, is that the output is a *prediction about a known quantity* rather than newly created content, so it can be evaluated directly against what later happens.

5. Agentic AI: Multi-Step Reasoning

Agentic systems implement multi-step reasoning through structured loops that turn a high-level goal into a sequence of observable actions. An LLM is placed at the centre of a control loop and given access to tools, memory and a stopping criterion; it is the loop, not the model alone, that makes the system agentic.

5.1 Core mechanisms

Chain-of-thought reasoning. The model produces explicit intermediate steps before a final answer or action, which markedly improves accuracy on multi-step problems (Wei et al., 2022). Reasoning models now internalise this behaviour and expose controllable “thinking” budgets.

Observation loops. After each action the agent receives feedback (tool output, API response, error) and revises its plan. This closed loop distinguishes agents from one-shot completions.

Task decomposition. A goal is split into sub-goals. A market-research report might decompose into identifying sources, retrieving and summarising each, outlining, drafting and verifying.

Memory. Short-term memory is the working context for the current task; long-term memory lives in vector databases or structured stores and preserves facts, prior actions and user preferences across sessions. Persistent files and project instructions (for example the AGENTS.md convention) are increasingly used as lightweight long-term memory for coding agents.

5.2 Reasoning frameworks

Table 5. *Principal agent reasoning patterns.*

Pattern	Mechanism	Suited to
ReAct (Yao et al., 2023)	Alternates Thought, Action (tool call) and Observation until the goal is met.	Dynamic environments, such as agents querying live inventory or search
Tree of Thoughts (Yao et al., 2023)	Explores several candidate reasoning paths as a tree, prunes dead ends and backtracks.	Planning under uncertainty: architecture decisions, routing under constraints
Plan-and-execute / Pre-Act (Rawat et al., 2025)	Produces a complete multi-step plan before acting, then executes and revises it.	Predictable workflows: report generation, CI/CD automation
Reflection (Shinn et al., 2023)	The agent critiques its own output or failures and retries with the lesson stored in memory.	Coding and tasks with verifiable outcomes

5.3 Architecture

A complete agentic architecture combines four layers. The **memory layer** holds short-term state and long-term knowledge. The **planning layer** decomposes the goal and sequences sub-goals. The **execution layer** invokes tools such as search, code interpreters, databases, browsers and external APIs. The **feedback loop** evaluates each result and decides whether to proceed, retry or escalate to a human. In **multi-agent systems**, specialised agents (for example one gathering data, one analysing it, one writing the report) collaborate under an orchestrator, each with its own tools and memory.

5.4 Interoperability protocols

Two open protocols now underpin most agent architectures. The **Model Context Protocol (MCP)**, introduced by Anthropic, standardises how agents connect to tools, data sources and applications. The **Agent2Agent (A2A) protocol**, originated by Google and partners, standardises how agents built on different frameworks discover and call one another. A useful summary is that MCP connects agents to tools, whereas A2A connects agents to agents. In December 2025 the Linux Foundation formed the Agentic AI Foundation, with Anthropic, OpenAI and Block as founding contributors and Google, Microsoft and Amazon among supporting members; it hosts MCP, Block's goose framework and OpenAI's AGENTS.md convention [1]. The practical consequence for developers is reduced lock-in: a tool exposed once as an MCP server can be consumed by most major frameworks and assistants.

5.5 Reliability and oversight

Because agents act, their errors compound over long runs and can have side effects. Production deployments therefore combine permission scoping, sandboxed execution, human approval for high-impact actions, tracing and logging, and task-level evaluation (for example computer-use benchmarks such as OSWorld). Security concerns specific to agents, notably prompt injection through tool outputs and over-broad tool permissions, should be treated as first-order design constraints.

6. Agentic AI Frameworks (2026)

The framework landscape has consolidated since the original edition. AutoGen and Semantic Kernel were merged into the **Microsoft Agent Framework**, which reached version 1.0 general availability on 3 April 2026 [2]. Model vendors now ship their own agent SDKs, and MCP support is near-universal. Four orchestration styles have stabilised: graph-based, role-based, handoff-based and hierarchical [5]. GitHub star counts, which the original edition reported, change quickly and are omitted here in favour of release and positioning information.

LangChain and LangGraph (LangChain Inc.). LangChain provides a large integration ecosystem; LangGraph, now at version 1.x, is its graph-based runtime for stateful, cyclical, long-running agents, with durable execution, checkpointing, human-in-the-loop interrupts and short- and long-term memory. It is widely regarded as the leading choice for production control, RAG systems and complex multi-agent coordination, and is provider-agnostic.

CrewAI. A role-based multi-agent framework in which each agent has a role, goal and backstory, supporting sequential and hierarchical processes and delegation. Now at version 1.x with native MCP tools and A2A task delegation, it is the fastest route to role-specialised prototypes such as content, research and development crews.

Microsoft Agent Framework. The unified successor to AutoGen and Semantic Kernel (1.0 GA, April 2026) for Python and .NET. It offers workflows with checkpointing, sessions, built-in OpenTelemetry, MCP clients and A2A support, and connects to Azure, OpenAI, Anthropic, Gemini, Bedrock and Ollama. It is the default for Microsoft and Azure-centred enterprises. AutoGen and Semantic Kernel should be regarded as legacy for new projects.

OpenAI Agents SDK. A lightweight, low-abstraction SDK built around agents, handoffs, guardrails, sessions and built-in tracing, with MCP tool support and compatibility with many non-OpenAI models. Suited to GPT-centred deployments and simple delegation chains.

Claude Agent SDK (Anthropic). The SDK that exposes the agent loop, tools, subagents and context management used by Claude Code. It has the deepest MCP integration of the lab SDKs and suits coding, file-system and computer-use agents built on Claude.

Google Agent Development Kit (ADK). An open-source (Apache 2.0) toolkit for hierarchical agent compositions with built-in evaluation, available in Python, TypeScript, Java, Go and Kotlin, with MCP and A2A support. Best for Gemini-first and Google Cloud teams.

LlamaIndex. Specialised in retrieval-augmented generation and agentic retrieval over structured and unstructured data, now through its Agent Workflows. Strong vector-store integrations and query planning make it the usual choice for knowledge-base agents over large document collections.

Other notable frameworks. Pydantic AI offers type-safe agent definitions for Python; **Strands Agents** (AWS) and **smolagents** (Hugging Face) are lightweight code-first options; **Mastra** targets TypeScript; **Haystack**

(deepset) provides pipeline-first multimodal document processing; **Agno** offers lightweight multi-agent systems with built-in memory. **AutoGPT**, once the most-starred agent project, remains useful for open-ended experimentation but is less structured than the frameworks above for production pipelines.

7. Framework Comparison

Table 6. *Agent frameworks compared (October 2026).*

Framework	Languages	Orchestration style	Best use case	Stack fit
LangGraph	Python, JS/TS	Graph, stateful	Production workflows needing control, durability and RAG	Any stack; cloud-agnostic
CrewAI	Python	Role-based crews	Role-specialised teams and rapid prototypes	Python; works with LangChain tools
Microsoft Agent Framework	Python, .NET	Graph workflows	Enterprise agents with checkpointing and observability	Azure and .NET first; multi-provider
OpenAI Agents SDK	Python, TypeScript	Handoffs	Lightweight GPT-centred agents and delegation	OpenAI first; many models supported
Claude Agent SDK	Python, TypeScript	Agent loop with subagents	Coding, file and computer-use agents	Claude first; deep MCP
Google ADK	Python, TS, Java, Go, Kotlin	Hierarchical	Gemini-first and GCP-native multi-agent systems	Google Cloud; A2A and MCP
LlamaIndex	Python, TypeScript	Workflow-based	Agents over large document and data collections	Any LLM; all major vector stores
Pydantic AI	Python	Typed agents	Type-safe, composable Python agents	Python; provider-agnostic
Haystack	Python	Pipelines	Multimodal document pipelines at scale	Cloud-agnostic; production NLP

8. Governance, Safety and Regulation

Capability growth has been matched by formal safety and regulatory frameworks. Leading developers publish frontier-safety policies that tie model release to capability evaluations; OpenAI, for example, reported rating its GPT-6 Astra model at its highest tier for cybersecurity risk and began rollout with vetted enterprise partners, and Anthropic restricts its most capable Mythos-tier model to a small group of trusted organisations. In the European Union, the AI Act applies in phases: obligations for general-purpose AI models began in August 2025, and the main obligations for high-risk systems were scheduled for August 2026, although proposals to postpone parts of that schedule have been under discussion; readers should verify the current status before relying on any date. For agentic systems, traceability is a practical requirement, and standardised protocols such as MCP make tool calls and context exchanges easier to log and audit.

9. Decision Guide

Table 7. *Choosing a framework by requirement.*

If you need…	Choose
Complex graph-based workflows, durable state or RAG pipelines	LangGraph / LangChain
Role-based specialist agent teams, quickly	CrewAI
A .NET or Azure-native enterprise agent platform	Microsoft Agent Framework
A lightweight GPT-centred agent with handoffs	OpenAI Agents SDK
Coding or computer-use agents on Claude, with deep MCP integration	Claude Agent SDK
Multimodal agents on Google Cloud with Gemini	Google ADK
Reasoning over large document sets	LlamaIndex
Type-safe Python agents	Pydantic AI
Multimodal document processing at scale	Haystack

For model selection, a pragmatic rule is to prototype with the strongest model available, then move each step of the workflow to the cheapest model that passes your own evaluation set. Open-weight models are appropriate where data residency or cost dominates; frontier proprietary models remain preferable for the hardest reasoning and long-horizon agentic work. Because both models and frameworks are changing monthly, build thin abstraction layers, use MCP for tool integration, and keep an evaluation suite so that migration is a measured decision rather than a rewrite.

10. Conclusion

This handbook has defined AI, set out its principal types, compared the leading generative platforms of October 2026, explained generative, predictive and agentic mechanics, surveyed the agent frameworks and protocols now in production use, outlined the governance context, and offered a decision guide. The central trend is a movement from models that answer to systems that act, which makes protocols, evaluation and oversight as important as model quality.

Glossary

Context window. The maximum number of tokens a model can consider in one request.

Diffusion model. A generative model that produces images or video by iteratively removing noise.

MCP. Model Context Protocol; an open standard connecting AI applications to tools and data.

A2A. Agent2Agent protocol; an open standard for communication between agents.

Mixture of experts. An architecture that activates only a subset of parameters for each token.

RAG. Retrieval-augmented generation; grounding model output in documents retrieved at query time.

Token. The unit of text (roughly a word fragment) that models read and generate.

References and Sources

[1] Linux Foundation (9 December 2025). Announcement of the Agentic AI Foundation, with MCP, goose and AGENTS.md as founding projects. Reported by Redmond Magazine and IT Brief, 2025–2026.

[2] Microsoft (3 April 2026). Microsoft Agent Framework 1.0 general availability, unifying AutoGen and Semantic Kernel. As summarised in agent-framework comparisons by LangChain and Morph, 2026.

[3] Upsolve (September 2026). *Context window sizes: GPT-5.6, Claude, Gemini and Grok.* Also Sigma Browser (2026) and Luke Stahl, *AI Models Guide 2026*. Secondary sources; figures vary and should be verified with vendors.

[4] Anthropic model line-up and access notes per Anthropic product documentation, October 2026. Other vendor model names per public release coverage, August to October 2026.

[5] vdf.ai (October 2026), *Best AI Agent Frameworks (2026)*; Uvik (2026), *Agentic AI Frameworks 2026: Production Comparison*.

Vaswani, A. et al. (2017). Attention is all you need. *NeurIPS*.

Wei, J. et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*.

Yao, S. et al. (2023). ReAct: Synergizing reasoning and acting in language models. *ICLR*.

Yao, S. et al. (2023). Tree of Thoughts: Deliberate problem solving with large language models. *NeurIPS*.

Shinn, N. et al. (2023). Reflexion: Language agents with verbal reinforcement learning. *NeurIPS*.

Rawat, M. et al. (2025). Pre-Act: Multi-step planning and reasoning improves acting in LLM agents. *arXiv*.