Types of AI agents

The four an enterprise builds, the ten forms they take, and how to choose

IN BRIEF

01 An enterprise builds four types of AI agent: conversational agents, workflow agents, autonomous agents and multi-agent systems. They differ in what starts the agent and in who decides its next step. Between them they take ten forms, from an agent that only answers questions to a guardian agent that checks the work of other agents.

02 Ask what types of AI agent exist and most sources give the same five: simple reflex, model-based, goal-based, utility-based and learning agents. That list is accurate, and it comes from a textbook. It describes how an agent reasons. It does not tell a team what to build, what will set the agent off, or what it needs before it can be trusted with real work.

03 This paper sorts agents the way a builder meets them. For each of the ten forms it says what the form is, when to use it, when not to, what it needs to run safely, and what other people call it. It then covers three settings that apply to every form, the applications that put agents in front of people, six questions for choosing, and what every agent needs whatever its type.

04 The position throughout is simple. Start with the simplest form that does the job. Add a second agent only when one can no longer hold the work. Whatever the type, a named person owns the agent, and it runs on infrastructure the organisation controls.

Conversational agents
Answering agent, acting agent

Workflow agents
Scheduled agent, event-triggered agent, API agent

Autonomous agents
Autonomous agent

Multi-agent systems
Agent pipeline, manager agent, agent team, guardian agent

IN THIS PAPER

01 Why the usual list does not help you choose 2
02 What counts as an agent 2
03 Two questions that sort every agent 3
04 The four types and ten forms 4
05 Three settings that apply to every form 7
06 Agent applications: where people meet the forms 8
07 Sorting by the job 9
08 How to choose: six questions 10
09 What every agent needs, whatever its type 11
10 Where to start 11

PART 01

Why the usual list does not help you choose

5	6	4	5
types in the textbook list, sorted by how an agent reasons	architectures in one practitioner's list	quadrants in one analyst's taxonomy	levels of maturity in one vendor's model

The textbook five. IBM's guide lists five types: simple reflex agents, model-based reflex agents, goal-based agents, utility-based agents and learning agents [1]. The list comes from the standard university textbook on artificial intelligence [2]. Some guides add hierarchical agents and multi-agent systems and count seven or eight [3].

What it describes. The five are ways an agent decides: by fixed rules, by rules plus a memory of what it has seen, by working toward a goal, by weighing one outcome against another, or by learning from results. IBM notes that all five can be used together in one system [1]. An agent built on a language model usually is several of them at once: it keeps context, works toward an outcome and weighs options.

What it leaves out. Nothing in the five says whether a person or a system starts the agent, whether it may change anything, how many agents are involved, or who approves. Those are the first decisions a team makes, and they set the cost and the risk.

There is no agreed alternative. Google's own explainer says that definitions of agent types differ [4]. One practitioner lists six architectures [5]. One analyst uses four quadrants [6]. One vendor uses five levels of maturity [7]. The model makers and the cloud providers publish design patterns in place of types [8] [10] [11] [12].

This paper draws on all of them and keeps the words that recur: conversational, workflow, autonomous, multi-agent.

PART 02

What counts as an agent

The strict definitions agree on one point. An agent is a system in which the model decides how the work gets done: it chooses the steps and the tools. OpenAI's guide says that applications which use a model without letting it control the work, simple chatbots among them, are not agents [8]. Microsoft separates three things: ordinary automation for fixed steps, retrieval for answering from documents, and agents for work where the system must reason and choose its actions [9]. Google separates an agent, which acts toward a goal, from an assistant, which recommends and leaves the decision to the user [4].

By that line, an assistant that only answers questions from documents is the boundary case. We include it as the first form for two reasons: buyers ask for it by the name "agent", and most organisations start there. The honest description is that it is the simplest thing sold as an agent, and that the definitions above would call it an assistant.

The same sources make a second point that matters more. Microsoft's guidance says plainly that not every business problem needs an agent [9]. If the steps are fixed, ordinary automation is cheaper and more predictable.

PART 03

Two questions that sort every agent

What starts it?

A person asking, a clock, an event in a system, a call from another system, or a goal handed over.

Who decides the next step?

The builder, in advance. Or an agent, while the work is under way.

Google's Agent Development Kit labels its own sample agents along the same lines: an interaction type, which is conversational, workflow or event-driven, and an agent type, which is single or multi-agent [13].

TYPE	FORM	WHAT STARTS IT	WHO DECIDES THE NEXT STEP
CONVERSATIONAL	**1. Answering agent**	A person's question	**The agent, within one answer. It changes nothing**
	2. Acting agent	A person's request	**The agent, choosing among the tools it was given**
WORKFLOW	**3. Scheduled agent**	A clock or a batch	**The builder sets the job. The agent handles each item**
	4. Event-triggered agent	Something happening in a system	**The builder sets the job. The agent handles each event**
	5. API agent	A call from another system	**The builder sets the job. The caller decides what follows**
AUTONOMOUS	**6. Autonomous agent**	A goal, from a person or an event	**The agent plans its own steps**
MULTI-AGENT	**7. Agent pipeline**	Any of the above	**The builder, in advance**
	8. Manager agent	Any of the above	**One agent, while the work is under way**
	9. Agent team	Any of the above	**Whichever agent holds the work**
	10. Guardian agent	Another agent's work	**The guardian, against rules you set**

The ten forms fall out of the two questions.

PART 04 · THE FOUR TYPES AND TEN FORMS

Conversational agents

A conversational agent is one a person talks to. It starts when someone asks, and it replies in the same conversation. The person is present throughout. It is the type most enterprises meet first [5].

FORM 01

Answering agent

It answers questions from knowledge you have approved and shows the source. It changes nothing in any system.

USE IT WHEN	people need answers from policies, manuals, contracts or product documents.
NOT WHEN	the person must then go and do the task by hand every time. That is the sign you need an acting agent.
IT NEEDS	a defined set of sources, the same reading rights as the person asking, an answer that cites what it relied on, and the ability to say that the documents are silent.
EXAMPLE	A staff assistant that answers a question on leave policy and links the paragraph it used.
ALSO CALLED	knowledge agent, retrieval agent, question-and-answer assistant.

FORM 02

Acting agent

It answers and also carries out the request through tools: the actions it has been given, such as looking up a record, making a booking or raising a ticket.

USE IT WHEN	the request ends in an action, and the steps vary with what the person says.
NOT WHEN	every request follows the same steps. A form and a fixed workflow will do it for less.
IT NEEDS	only the tools the job requires, the permissions of the person it acts for, confirmation before anything that cannot be undone, and a handover to a person when it is stuck [8].
EXAMPLE	A service agent that finds the order, explains the delay and books the replacement.
ALSO CALLED	task agent, service agent, assistant with tools.

Workflow agents

A workflow agent works with nobody asking. A clock, an event or another system starts it, and it does a defined job in the background. Google describes these as background agents: little or no human interaction, and generally driven by events [4].

FORM 03

Scheduled agent

It runs at set times, or over a batch of items, and produces the same kind of output each run.

USE IT WHEN	the work follows the organisation’s rhythm: a morning digest, a weekly report, a nightly check.
NOT WHEN	the work must happen within minutes of a change. Use an event-triggered agent.
IT NEEDS	a history of runs, an alert when a run fails or produces nothing, and a named person who reads the output.
EXAMPLE	Every morning at six, read what the regulators published overnight and write a digest of what changed and whom it affects.
ALSO CALLED	batch agent, recurring agent.

PART 04 · CONTINUED

FORM 04

Event-triggered agent

It starts when something happens in a system: an email arrives, a file lands, a record changes, an alert fires.

USE IT WHEN	speed matters and the trigger is clear.
NOT WHEN	the event carries too little to act on, or fires thousands of times an hour for work that could wait for a batch.
IT NEEDS	protection against acting twice on the same event, a limit on runs per hour, a queue for the cases it cannot finish, and a rule for which actions wait for a person.
EXAMPLE	When a claim form arrives, extract the details, check them against the policy and open a case with a recommendation.
ALSO CALLED	event-driven agent, ambient agent [14], background agent, triggered workflow [5].

FORM 05

API agent

Another application calls it, it does one job, and it returns a result in a fixed format. There is no conversation.

USE IT WHEN	an existing system needs judgement at one step: classify this, summarise that, draft a decision note.
NOT WHEN	a fixed rule would give the same answer.
IT NEEDS	a fixed format for what goes in and what comes out, a check on who is calling, a time limit, and a defined answer for “cannot decide”.
EXAMPLE	The lending system sends an application and receives a risk summary with its reasons listed.
ALSO CALLED	headless agent, task agent. The caller can be another agent. The Agent2Agent protocol, started by Google and now hosted by the Linux Foundation, sets out how one agent finds and calls another built by a different team or company [15].

Autonomous agents

FORM 06

Autonomous agent

It is given a goal, not steps. It plans, acts, checks the result and adjusts, and it can keep working for minutes, hours or days [10] [16].

USE IT WHEN	the path cannot be written down in advance: research, investigation, diagnosis [5] [10].
NOT WHEN	the steps are known. Small errors compound over a long run [6], and a pipeline will be cheaper and more reliable.
IT NEEDS	a limit on steps, time and spend, a contained place to work, points at which it reports or asks, and a full record of what it did.
EXAMPLE	Given “find out why last night’s settlement run failed”, it reads the logs, tests its explanations and writes up the cause with the evidence.
ALSO CALLED	long-horizon agent, deep research agent.

PART 04 · CONTINUED

Multi-agent systems

A multi-agent system is several agents, each with its own instructions and tools, working on one piece of work. The forms differ in who decides the order. OpenAI advises starting with one agent and moving to several only when needed [8]. Microsoft's guidance is to start with several where the work crosses security boundaries or teams, and otherwise to test a single agent first [9]. Gartner expects 70 percent of AI applications to use multi-agent systems by 2028 [17].

FORM 07

Agent pipeline

The builder fixes the order in advance. Each agent does its part and passes its output to the next. Stages can also run side by side and be combined.

USE IT WHEN	the work has known stages that need different skills: extract, draft, check, assemble.
NOT WHEN	the right order depends on the case.
IT NEEDS	a check on the output between stages, a stop when a stage fails, and a record that follows one item through every stage.
EXAMPLE	A tender response: one agent extracts the requirements, a second drafts answers from past bids, a third checks compliance, a fourth assembles the document.
ALSO CALLED	sequential or parallel workflow [11] [12], prompt chaining [10].

FORM 08

Manager agent

One agent receives the work, decides while the work is under way which specialist handles each part, and assembles the result. The specialists never deal with the user.

USE IT WHEN	requests vary widely and you want one point of control and one voice.
NOT WHEN	the order is always the same. The manager then adds cost and one more thing that can fail.
IT NEEDS	a limit on how often it may delegate, a clear description of each specialist, and a record of which agent did what.
EXAMPLE	A network operations manager receives an alarm, hands diagnosis, customer impact and ticketing to three specialists, and writes one incident note.
ALSO CALLED	orchestrator, coordinator, supervisor, hierarchical agents [1] [8] [10] [11].

FORM 09

Agent team

Specialists hand work directly to one another, and whichever holds the work deals with the user. No single agent is in charge.

USE IT WHEN	a case moves between specialisms and each specialist should own its part.
NOT WHEN	you need one accountable point of control. This is the least mature form. One analyst rates it the highest risk of four patterns [6], and it is the hardest to trace.
IT NEEDS	rules on who may hand to whom, detection of work going round in circles, and one record across every handover.
EXAMPLE	An intake agent passes a customer to the billing agent, which passes a cancellation to the retention agent.
ALSO CALLED	handoff, swarm, group chat [8] [11] [12].

PART 04 · CONTINUED

FORM 10

Guardian agent

Its job is to watch, check or stop other agents. Gartner names three roles: reviewers that check output, monitors that track actions, and protectors that adjust or block an action as it happens [17].

USE IT WHEN	agents act on customers, money or regulated records, or when more agents are running than people can review.
NOT WHEN	it shares the instructions and the blind spots of the agent it checks. It then adds cost and no independence.
IT NEEDS	independence from what it checks, the authority to block and not only to report, and a record of its own.
EXAMPLE	Before a customer letter drafted by another agent is sent, the guardian checks it against the disclosure rules and holds the ones that fail.
ALSO CALLED	reviewer, critic, evaluator, monitor [10] [11]. Google ships a sample agent that monitors other agents in production [13]. Gartner expects guardian agents to take 10 to 15 percent of the agentic AI market by 2030 [17].

PART 05

Three settings that apply to every form

These are often listed as types. They are settings, because any of the ten forms can have them.

How much a person approves

One published scale gives the person five possible roles: operator, collaborator, consultant, approver and observer [18]. In practice an agent brings a person in three ways: it notifies, it asks a question, or it presents an action for review [14]. OpenAI names two triggers for bringing a person in: repeated failure, and an action that is sensitive, irreversible or high in value [8]. Reading is free. A change that can be undone is recorded. A change that cannot be undone waits for a person.

The channel

Chat, voice, a panel inside another application, email. The channel changes how a person reaches a conversational agent. It does not change the type: a voice agent is an answering or acting agent that you speak to.

How it acts

Through an interface built for software, or through the screen as a person would, by clicking and typing. Screen use reaches systems that have no interface for software [19]. It is slower and less predictable, so keep it for systems with no other way in.

Set approval by action, not by agent.

PART 06

Agent applications: where people meet the forms

Most people in an organisation never meet an agent directly. They meet an application: a screen with a queue of cases, a draft to read, a button to approve, and a register of what was issued. The agents sit behind it.

An agent application is not an eleventh form. It is how the forms are delivered, and it is where two of the three settings live: the approval step is a button in the application, and the application is the channel. The two sorting questions have no single answer for an application, because inside it different things start the agents.

Take a regulatory change desk in a bank's compliance team.

WHAT THE OFFICER SEES	WHAT IS BEHIND IT
New publications arrive as cases	**A sweep of the regulators' sites. Put on a timetable, it is a scheduled agent**
"Draft the gap analysis"	**An API agent. The desk sends the publication and the matching policy clauses, and gets back a table**
"Ask the agent" on a case	**An answering agent**
Approve, send back, issue	**The approval setting, exercised by a named officer**
The register	**The record of who decided what, and on which draft**

Two things follow. First, a form describes how an agent is used, not how many you build: in this desk one agent both drafts on the desk's call and answers the officer's questions. Second, the order of work is fixed by the application, not by an agent. The desk is a pipeline whose stages are an agent, a person and a record, which keeps it predictable. When the order has to vary with the case, the application calls a manager agent instead.

Build the application last. Prove each form on its own, then put the screen around it.

PART 07

Sorting by the job

Most lists a buyer meets are sorted by the job, not by how the agent works. Google groups uses into six: customer, employee, creative, data, code and security agents [4]. That view is the right one for finding where to start. It does not tell you what to build.

THE JOB	THE FORMS USUALLY USED
Staff questions on policy, HR and IT	Answering agent first, acting agent for requests
Customer service	Acting agent. An agent team where cases move between specialisms
Regulatory and market monitoring	Scheduled agent for the daily read. A pipeline to assess each change
Intake of claims, invoices and applications	Event-triggered agent. An API agent for a single decision
Research and investigation	Autonomous agent
Operations and incidents	An event-triggered start, with a manager agent to coordinate
Anything sent to a customer or a regulator	A guardian agent before it leaves, and a person who approves

The two views fit together.

PART 08

How to choose: six questions

01

Does a person ask each time?

If yes, it is conversational. If the answer ends the matter, an answering agent. If something must then be done, an acting agent.

02

If nobody asks, what starts the work?

A clock: scheduled. Something happening: event-triggered. Another system: API.

03

Can you write the steps down in advance?

If you can, stay with a workflow agent or a pipeline. If you cannot, an autonomous agent, with limits.

04

Is the work too wide for one agent?

The signs are instructions that pull against each other, more tools than one agent chooses between reliably, or parts that belong to different teams or security boundaries [8] [9].

05

Who should decide the order?

You, in advance: a pipeline. One agent: a manager. The specialists themselves: a team.

06

What happens if it is wrong?

If the answer involves a customer, money or a regulated record, add an approval step and a guardian.

OpenAI, Anthropic and Microsoft give the same advice: begin with the simplest design and add complexity only when the simple one falls short [8] [9] [10].

Then take the simplest form the answers allow.

PART 09

What every agent needs, whatever its type

01	**An owner**	A named person answers for what the agent does.
02	**An identity and the least access**	The agent has its own identity. It reads what its user could read, or what its job requires, and no more.
03	**One gate for every action**	Each tool call is checked against policy, which can allow it, record it, hold it for a person, or block it.
04	**A record**	What started the run, what the agent read, which tools it used, what it produced, and who approved.
05	**Limits**	Steps, time and spend, set before the first run.
06	**A stop**	One agent, one tool, or all of them, at once, by you.
07	**A known place**	Where the model runs, where the data and the records are kept, and who else can reach them. For a regulated organisation this is the first question, not the last.

PART 10

Where to start

Start with an answering agent on your own documents. No system is changed, so it can run on the first day, and it shows whether your knowledge is in a state an agent can use.

Then take one process and one form: an acting agent for a single type of request, or an event-triggered agent for a single type of document. Run it beside the manual process and measure it against that.

This paper is the first of a series. A build tutorial and a short video follow for each of the ten forms: the same form, built step by step on the Katonic platform.

Add a second agent when the first can no longer hold the work, and not before.

About Katonic AI

Katonic AI builds the platform that organisations use to build, run and govern AI agents on their own infrastructure: in their own data centre, in a sovereign cloud, or with no connection to the internet at all. The platform is in production with governments, banks, telecom operators and utilities. Katonic is certified to ISO 27001.

KATONIC.AI

SOURCES

Read on 21 November 2026.

01 IBM, “Types of AI agents”. ibm.com/think/topics/ai-agent-types

02 Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, chapter 2, “Intelligent Agents”. aima.cs.berkeley.edu

03 Wrike, “Different types of AI agents”, 26 August 2025 (seven types); Red Hat, “Understanding AI agent types: a guide to categorizing complexity”, 3 September 2025 (eight types). wrike.com, redhat.com

04 Google Cloud, “What are AI agents? Definition, examples, and types”. cloud.google.com/discover/what-are-ai-agents

05 Bernard Aceituno, “The 6 kinds of AI agent architectures”, CIO, 20 July 2026. cio.com

06 Ian Thomas, “AI agents in the enterprise (part 2): a simple taxonomy”, diginomica, 5 June 2025. diginomica.com

07 Salesforce, “Agentic Maturity Model”, 10 April 2025. salesforce.com/news/stories/agentic-maturity-model

08 OpenAI, “A practical guide to building agents”. openai.com/business/guides-and-resources

09 Microsoft, Cloud Adoption Framework, “AI agent adoption strategy”, 9 November 2026. learn.microsoft.com/azure/cloud-adoption-framework/ai-agents

10 Anthropic, “Building effective agents”, December 2024. anthropic.com/engineering/building-effective-agents

11 Google Cloud Architecture Center, “Choose a design pattern for your agentic AI system”. docs.cloud.google.com/architecture

12 Microsoft, Azure Architecture Center, “AI agent orchestration patterns”, 12 February 2026. learn.microsoft.com/azure/architecture

13 Google, Agent Development Kit documentation and sample agents. adk.dev, github.com/google/adk-samples

14 Harrison Chase, “Introducing ambient agents”, LangChain, 14 January 2025. langchain.com/blog

15 IBM, “What is A2A protocol (Agent2Agent)?”. ibm.com/think/topics/agent2agent-protocol

16 George Saines, “Long-horizon agents”, Salesforce, 16 September 2026. salesforce.com/blog/long-horizon-agents

17 Gartner press release, “Gartner Predicts that Guardian Agents will Capture 10-15% of the Agentic AI Market by 2030”, 11 June 2025. gartner.com/en/newsroom

18 Kevin Feng, David McDonald and Amy Zhang, “Levels of Autonomy for AI Agents”, Knight First Amendment Institute, 28 July 2025. knightcolumbia.org

19 Microsoft Copilot Studio blog, “Computer-using agents now deliver more secure UI automation”. microsoft.com/en-us/microsoft-copilot/blog