

Who and what does what in an AI system.

A high-level guide to how AI assistants, agents, runtimes, tools and services fit together.

Who starts the work? Who picks the next step? Who allows the action? The model proposes; the runtime runs things and applies the checks built into it; tools, APIs and services do the work.

01 Start

Work starts with a person or an event. Either can kick off an assistant or an agent. Policy sets what is allowed.

02 Decide and coordinate

An assistant is a user-facing role; an agent works toward a goal, with varying autonomy. One runtime can do both: it calls the model, applies the checks built into it, looks at results, retries, and stops at its limits.

03 Reason

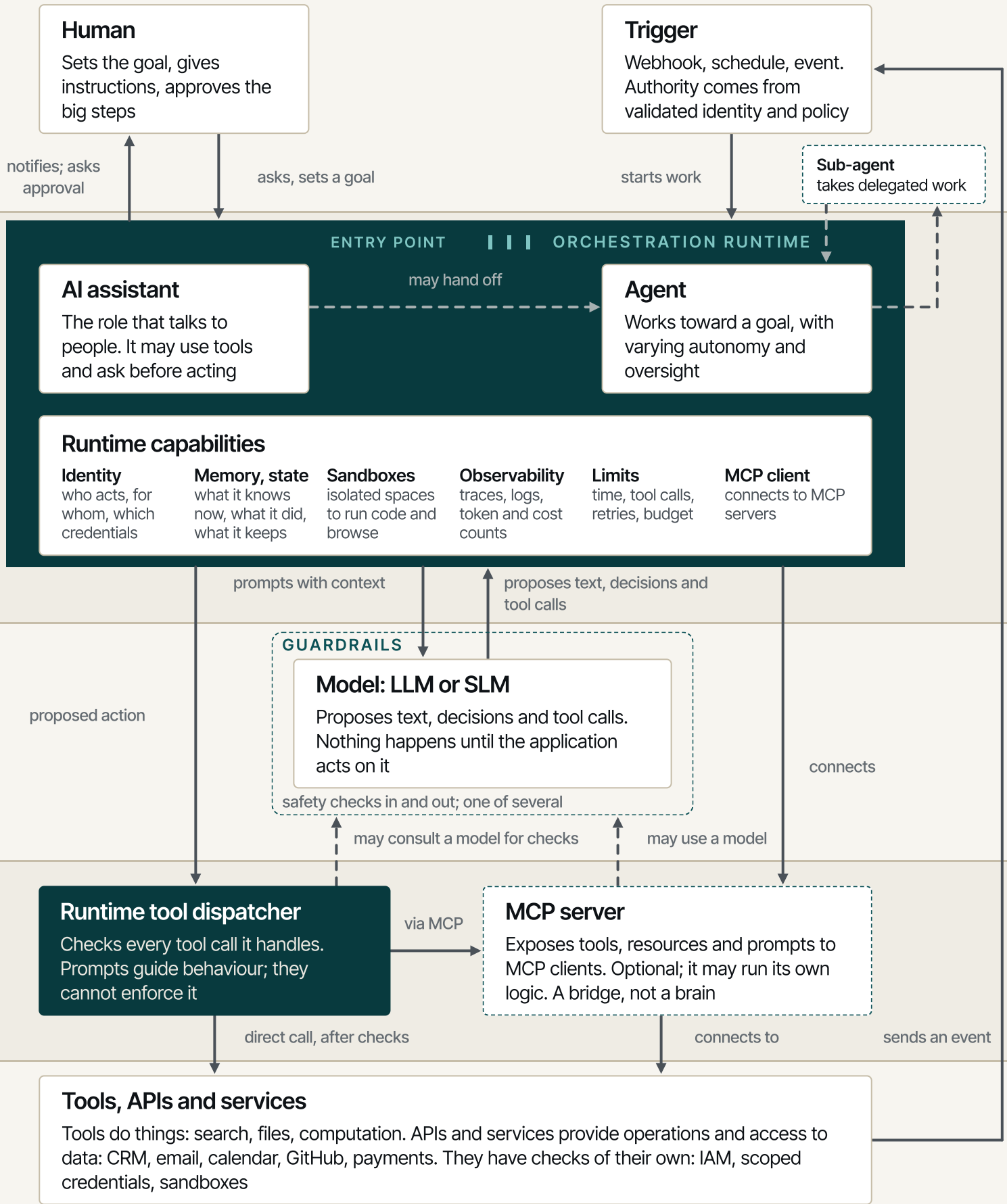
The model may make a plan or follow one. It proposes the next action; the runtime carries it out, after whatever checks are set up.

04 Permit and connect

The dispatcher runs the configured checks. MCP and direct calls are both fine; tools and services add their own.

05 Do

Tools do things. Services provide operations and data.



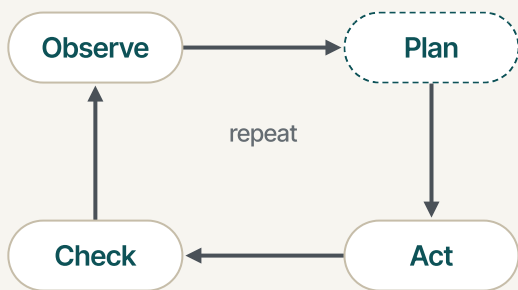
Not two boxes: six dials

Different settings, not different systems

Autonomy	supervised	independent
Initiative	asked	unprompted
Workflow	one reply	many steps
Authority	read-only	changes state
Persistence	one turn	long-running
Oversight	every step	checkpoints

The agent loop

A common pattern, not a requirement



Planning can be implicit; some agents act only once. Limits live in the runtime.

Example execution patterns

Examples, not a scale. Agency is a spectrum.

- Plain automation**
code runs code: service > service
- Model-backed assistant**
a model answers; code runs
- Tool-using assistant**
the model proposes; code runs the tools
- Code-directed workflow**
fixed logic sets the steps; the model fills them in
- Model-directed agent**
the model picks each next action, within limits
- Multi-agent system**
several agents cooperate; a design choice, not a step up