

Agent Skills: A Blueprint for Capable AI

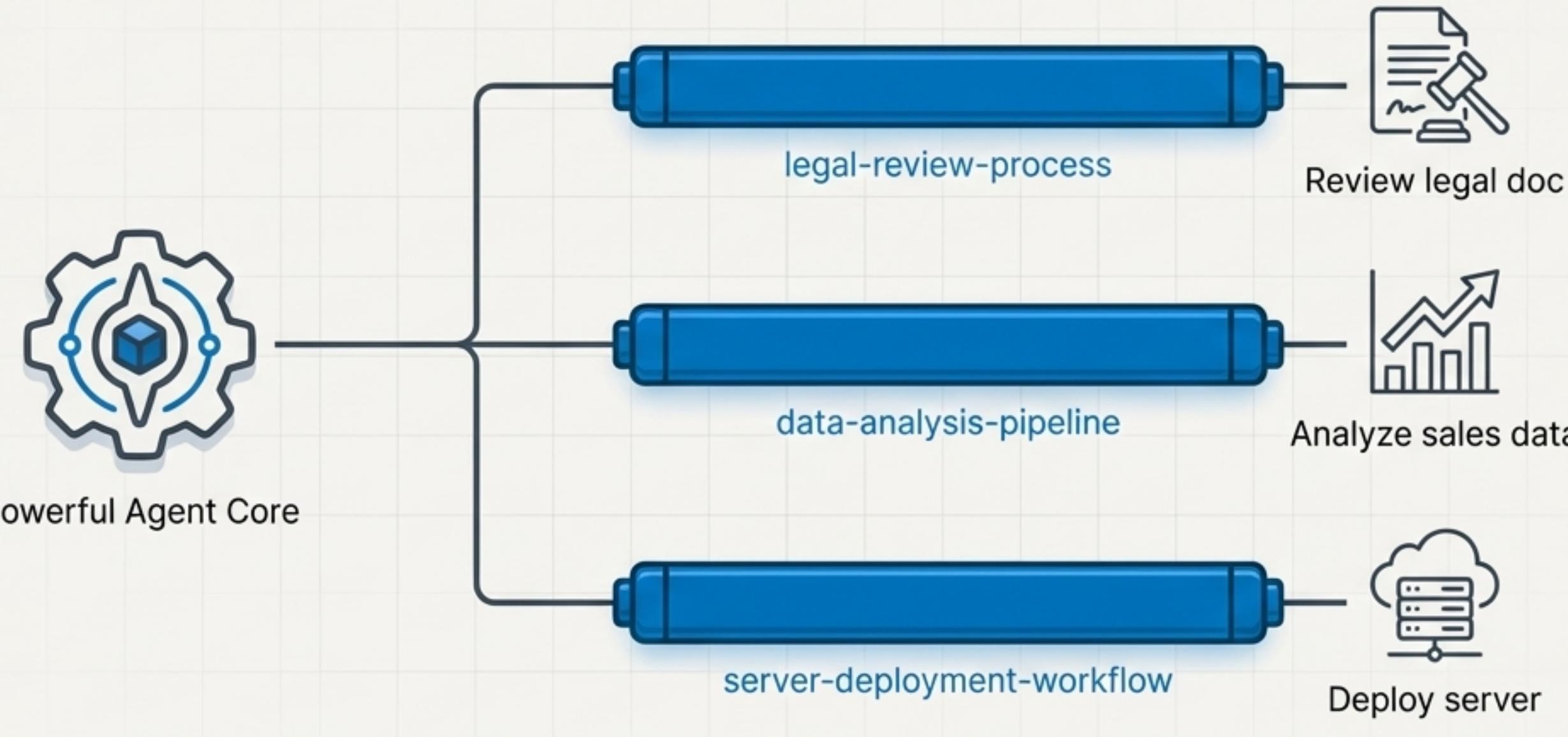
A simple, open format for giving agents new capabilities and expertise.

Today's agents are powerful, but context-starved.



Agents are increasingly capable, but they often don't have the context they need to do real work reliably. They lack access to procedural knowledge and the specific context of a company, team, or user.

Agent Skills bridge the gap between capability and context.



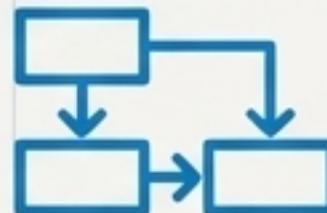
Skills solve this by giving agents access to procedural knowledge and company-, team-, and user-specific context they can load on demand. An agent with access to a set of skills can extend its capabilities based on the task it's working on.

Unlocking new applications, from expertise to interoperability.



Domain Expertise

Package specialized knowledge into reusable instructions, from legal review processes to data analysis pipelines.



Repeatable Workflows

Turn complex, multi-step tasks into consistent and auditable workflows.



New Capabilities

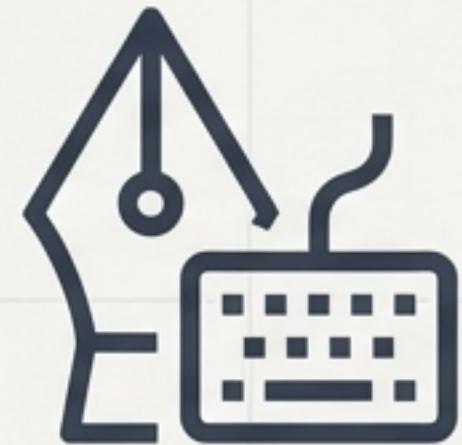
Give agents entirely new functions, like creating presentations, building MCP servers, or analyzing datasets.



Interoperability

Reuse the same skill across different skills-compatible agent products, creating a portable ecosystem.

A unified standard benefits the entire development ecosystem.



For Skill Authors

Build capabilities once and deploy them across multiple agent products.



For Agent Developers

Support for skills lets end users give agents new capabilities out of the box.



For Teams & Enterprises

Capture organizational knowledge in portable, version-controlled packages.

The Anatomy of a Skill: A self-contained directory.

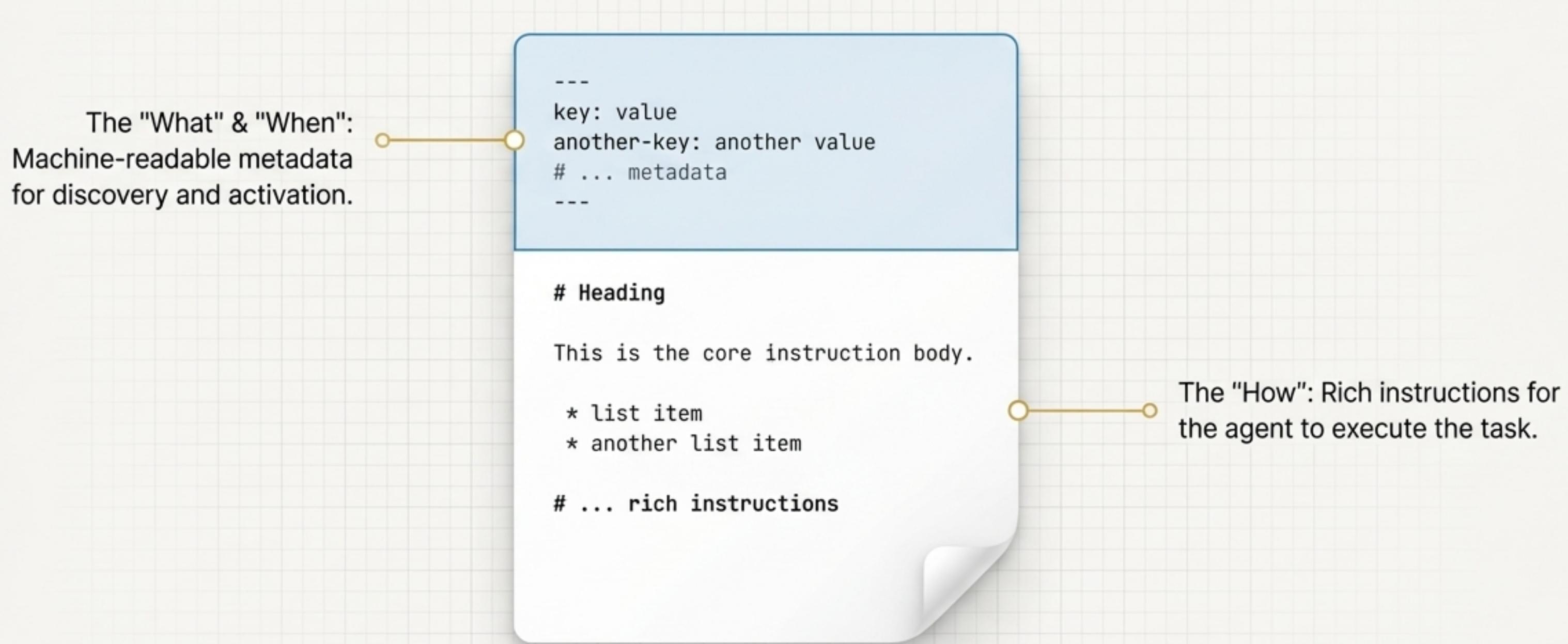
A skill is a directory containing, at minimum, a SKILL.md file.

You can optionally include additional directories to support your skill with scripts, reference materials, and assets.

```
skill-name/
  └── SKILL.md      # Required: Core instructions & metadata
  └── scripts/      # Optional: Executable code (Python, Bash, etc.)
  └── references/   # Optional: Detailed documentation & guides
    └── assets/      # Optional: Static resources (templates, data)
```

The core of every skill is the `SKILL.md` file.

The `SKILL.md` file combines structured metadata with free-form instructions.
It must contain YAML frontmatter followed by Markdown content.



The frontmatter defines the skill's identity and constraints.

```
---
```

```
# REQUIRED
name: pdf-processing
description: Extracts text and tables...
```

- Unique ID, max 64 chars, matches dir name
- What it does & when to use it, max 1024 chars

```
# OPTIONAL
license: Apache-2.0
compatibility: Requires git, docker, jq
metadata:
  author: example-org
  version: "1.0"
  allowed-tools: Bash/git:*) Bash/jq:*)
```

- License name or reference to a file
- Specific environment or system requirements
- Arbitrary key-value pairs for custom info
- [Experimental] Pre-approved tools

```
---
```

A great description tells the agent **what** the skill does and **when** to use it.

The `description` field is critical for discovery. It should include specific keywords that help agents identify relevant tasks.

POOR 

1 | `description`: Helps with PDFs.

GOOD 

1 | `description`: Extracts text and tables from PDF files, fills PDF forms, and merges multiple PDFs. Use when working with PDF documents or when the user mentions PDFs, forms, or document extraction.

2

The body contains the procedural knowledge for the agent.

After the frontmatter, the Markdown body contains the skill instructions. While there are no format restrictions, structured instructions are most effective.

Recommended Sections

- Step-by-step instructions
- Examples of inputs and outputs
- Common edge cases and how to handle them

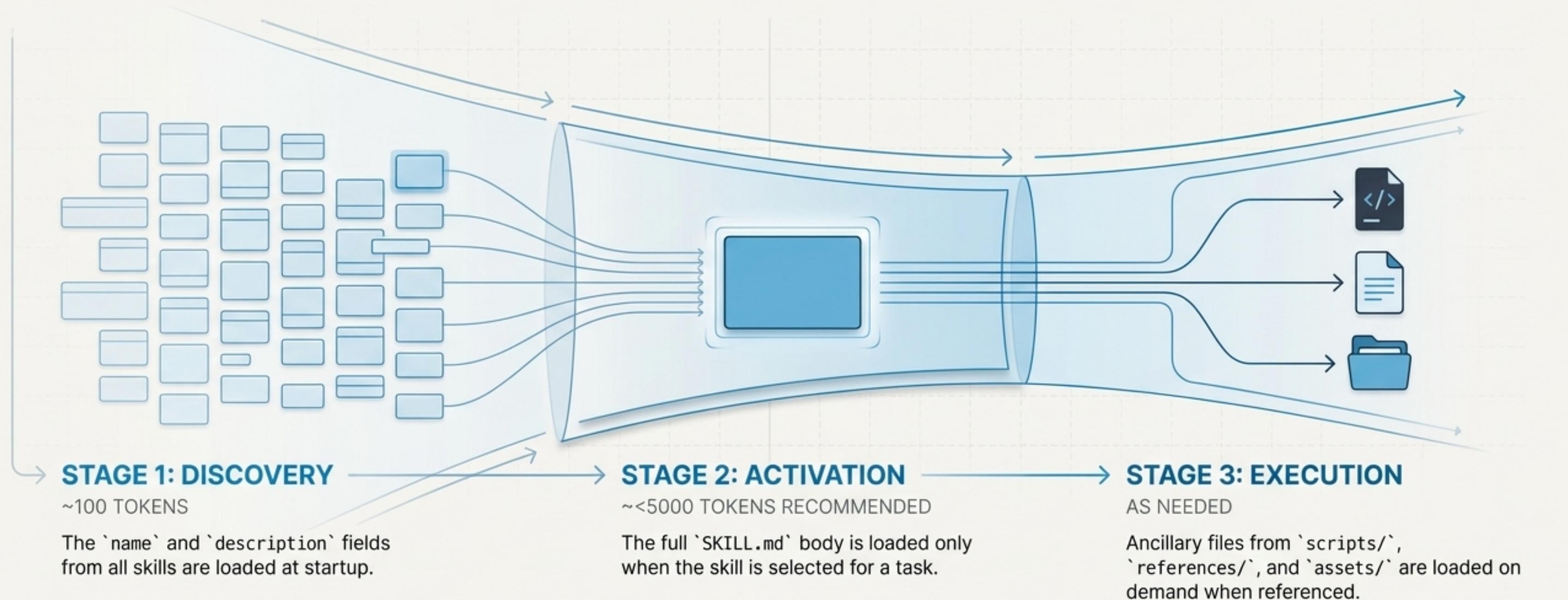


Pro Tip

The entire SKILL.md is loaded when a skill is activated. For lengthy instructions, move detailed material into the references/ directory and link to it.

Designed for efficiency with progressive disclosure.

Skills are structured to minimize context usage. Information is loaded in stages, only as it's needed by the agent.



An open standard, built for community and growth.

- Originally developed by Anthropic and released as an open standard.
- Adopted by a growing number of agent products and leading AI development tools.
- The standard is **open to contributions** from the broader ecosystem.



Start building and using skills today.



1. Explore Example Skills

Browse a library of curated skills on GitHub to see best practices and get inspired.

[Browse Examples](#)

2. Integrate Skills Support

Add skill compatibility to your agent or tool by following our integration guide.

[Read Integration Docs](#)

3. Author Your First Skill

Consult the complete format specification and start capturing knowledge in a portable, reusable format.

[View Specification](#)