

# Coding with AI



**[https://github.com/up1/  
course-ai-coding-2026](https://github.com/up1/course-ai-coding-2026)**



# Coding with AI



# Topics

Programming/Coding workflow  
Specification-Driven Development (SDD)  
Model-Driven Development (MDD)  
Coding with AI Agent workshop

Requirement

Planning

Coding

Validate and Testing



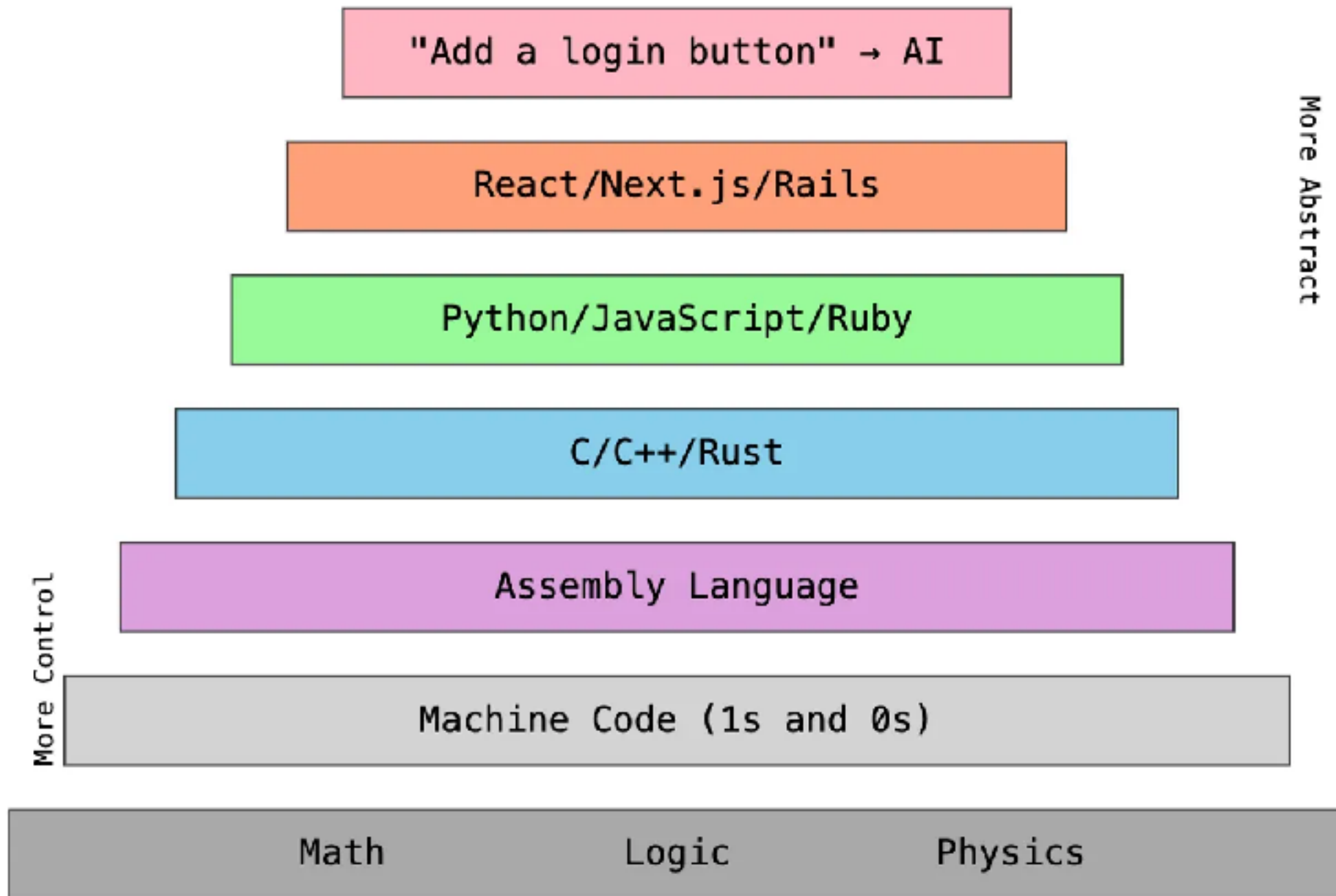
# History of Programming



Increase of abstraction



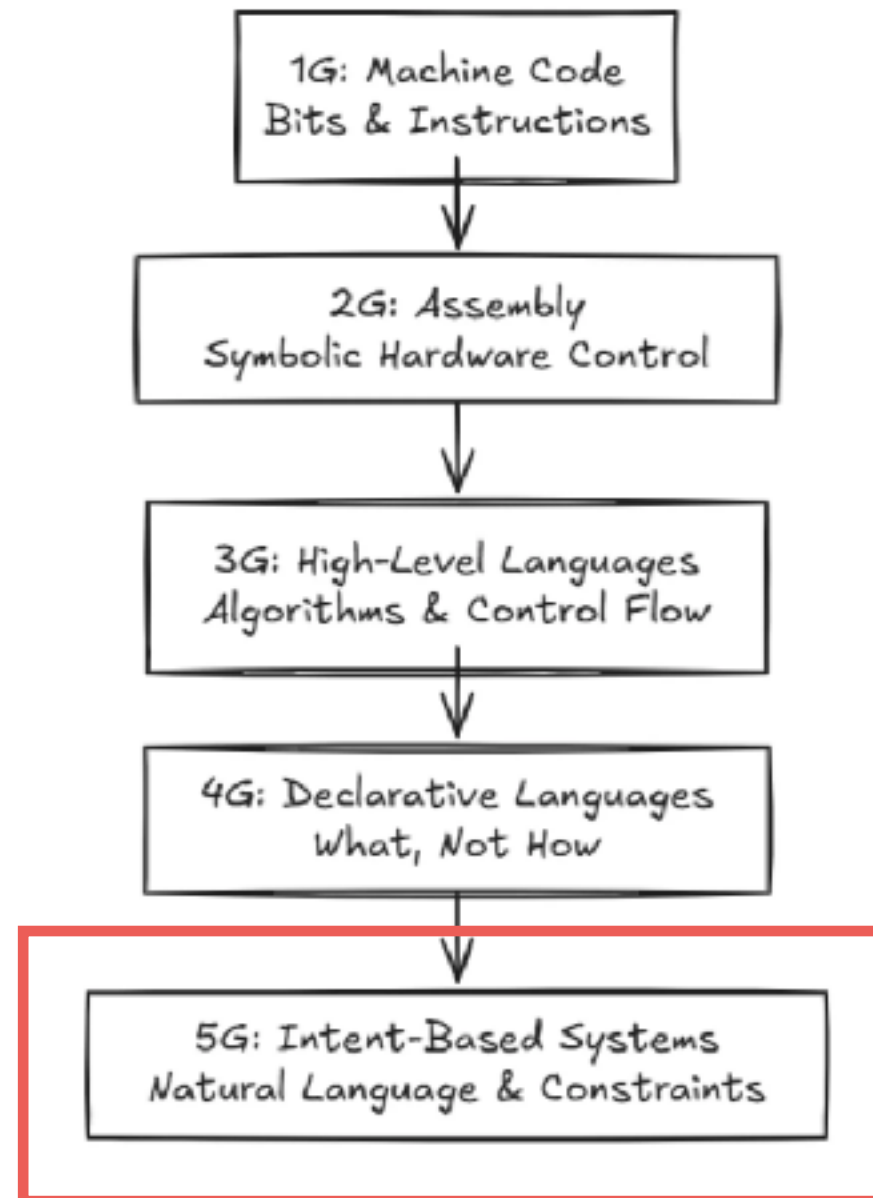
# The Modern Tower of Abstraction



<https://cline.bot/blog/from-assembly-to-ai-why-vibe-coding-is-just-another-chapter-in-our-abstraction-story>



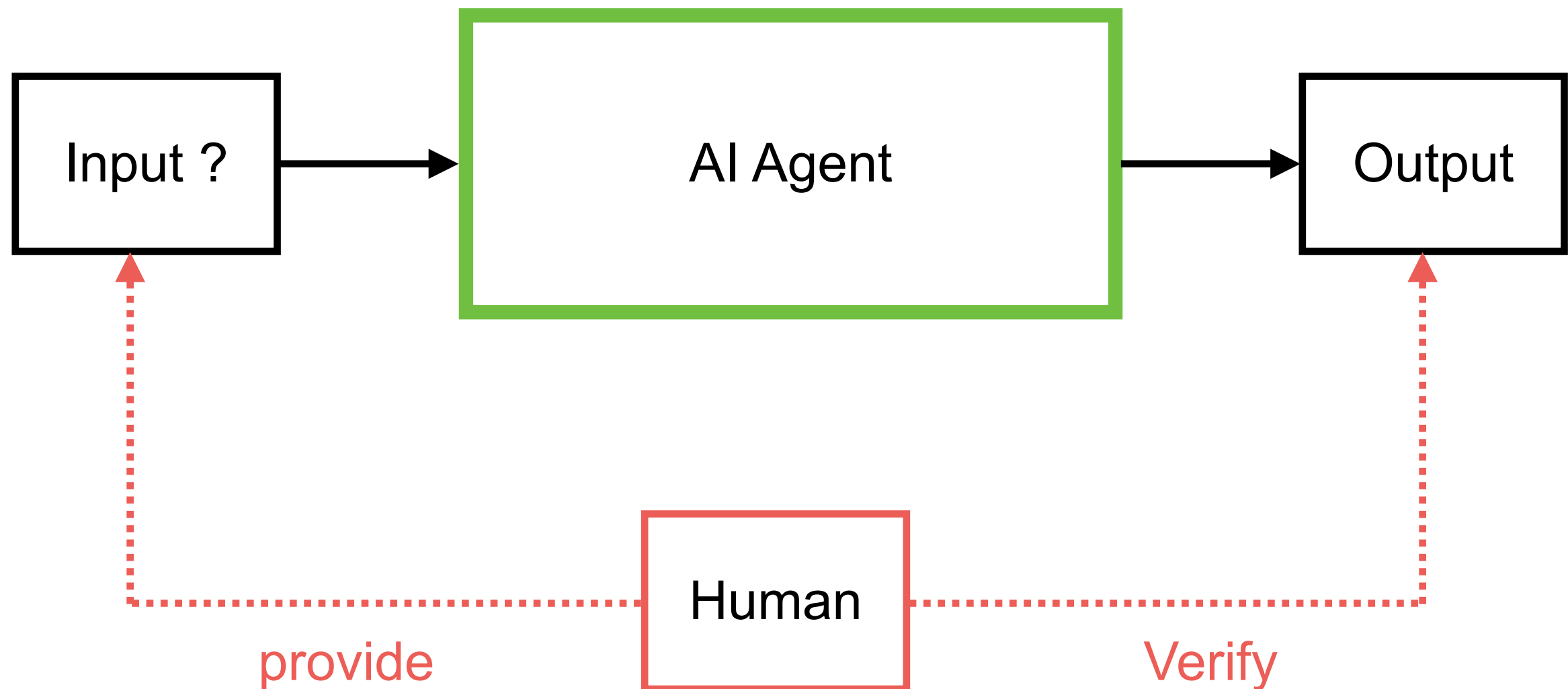
# Generation of Programming languages



<https://medium.com/@ajuatahcodingarena/generations-of-programming-languages-bed30d19ea8e>



# Coding with AI Agent





# AI Agent for Coding



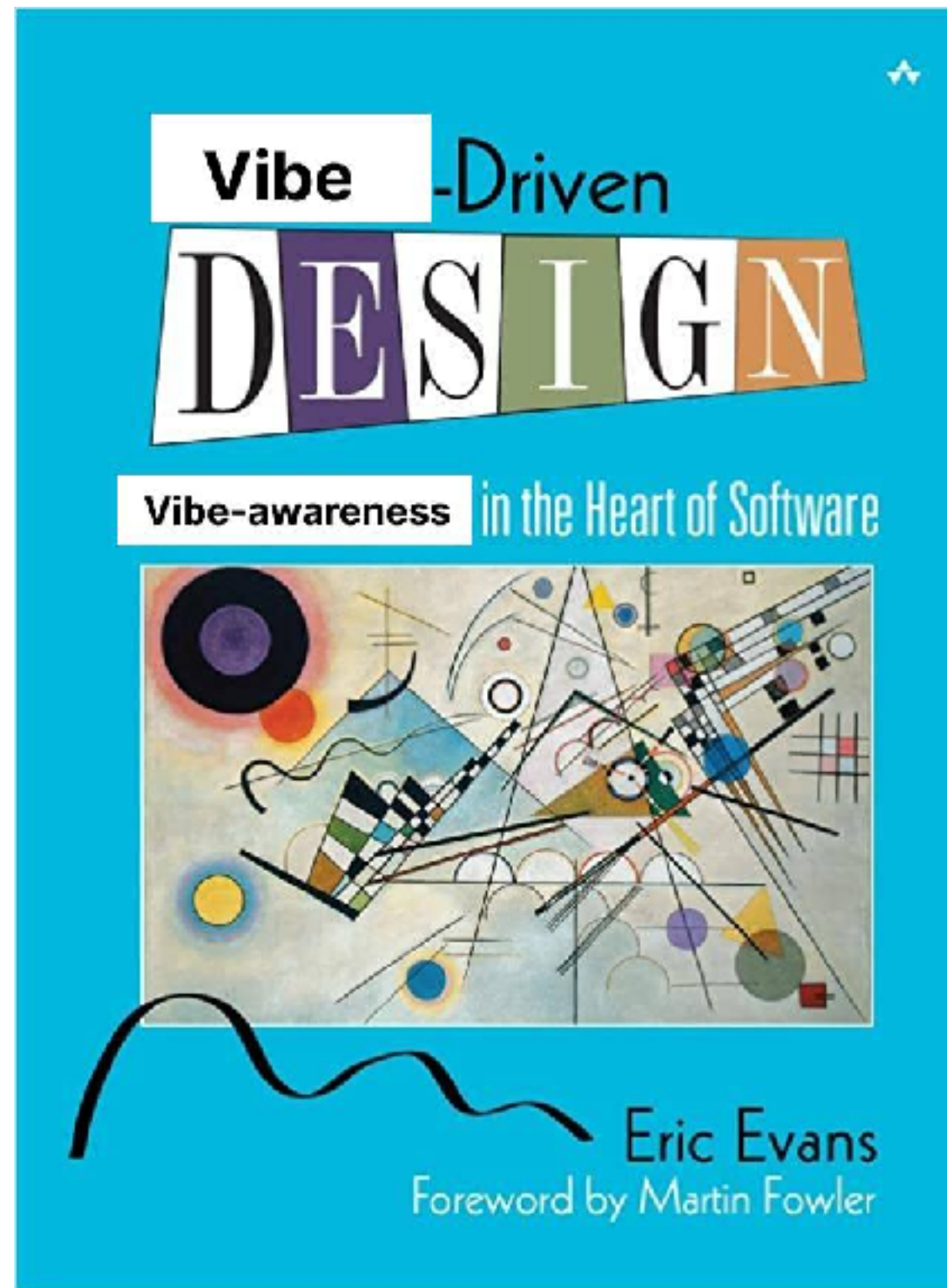
LLM

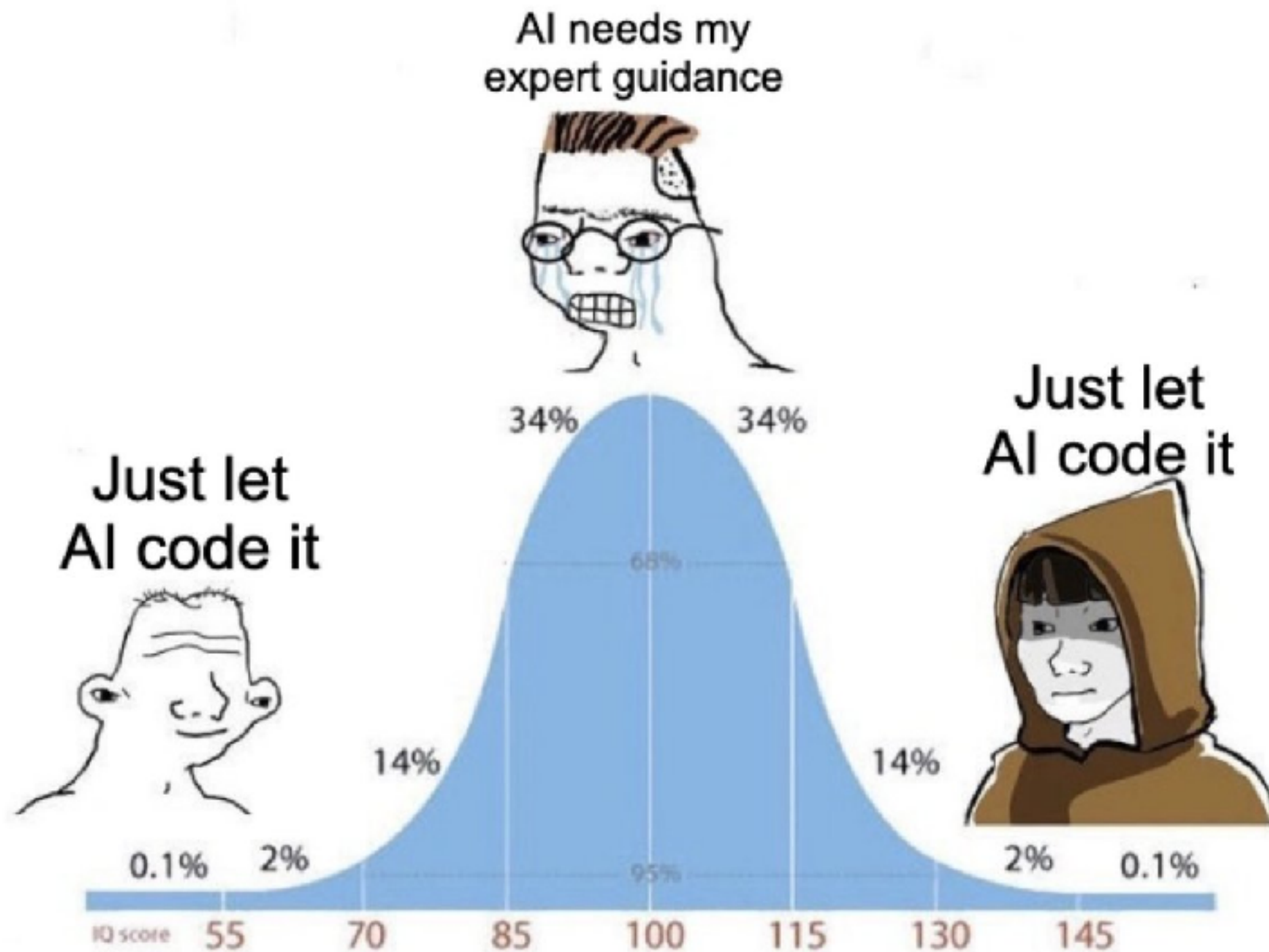
Code writing  
tools

Memory and  
Context

Other tools









Andrej Karpathy ✓

@karpathy



There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

6:17 AM · Feb 3, 2025 · 5.3M Views

<https://x.com/karpathy/status/1886192184808149383>





# Vibe Coding

**Software development practice** that uses artificial intelligence (AI) to generate functional code from **natural language prompts**, accelerating development, and making app building more accessible, especially for those with limited programming experience

<https://cloud.google.com/discover/what-is-vibe-coding>



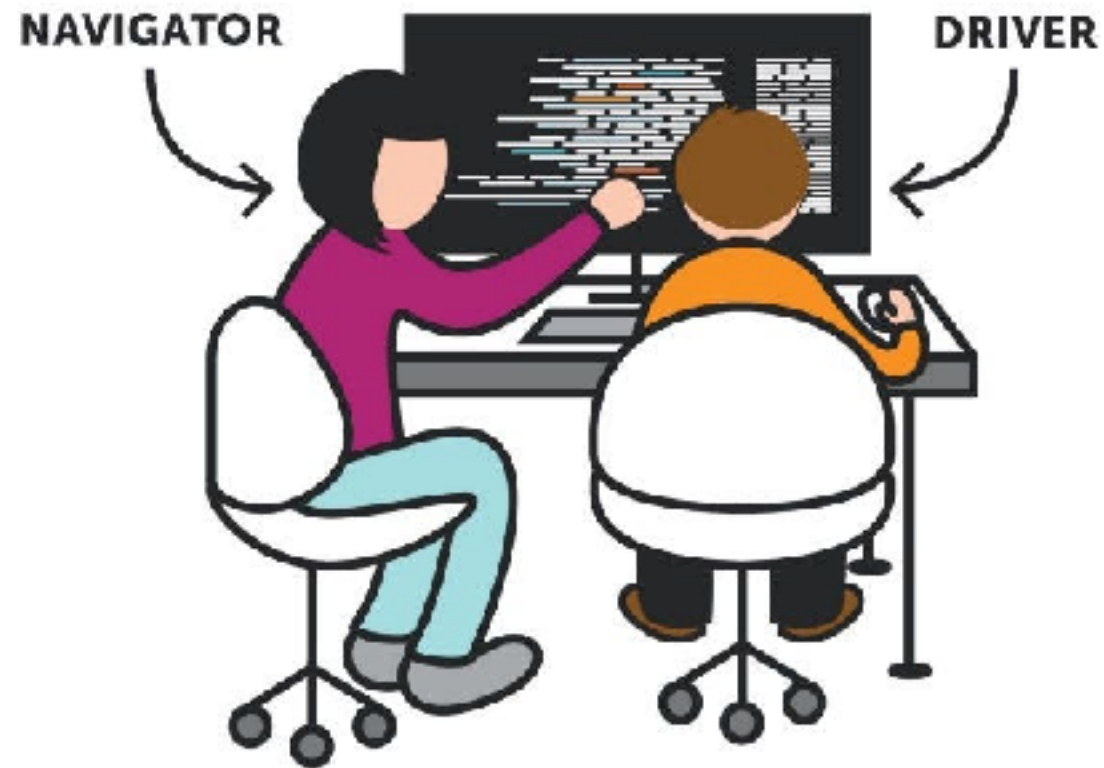
# Types of Vibe coding ?

Pure vibe coding

**Responsible** AI-assist development



## PAIR PROGRAMMING



Reviews, Tests and Understands  
with experience



# Workflows ?

Application level

Code level

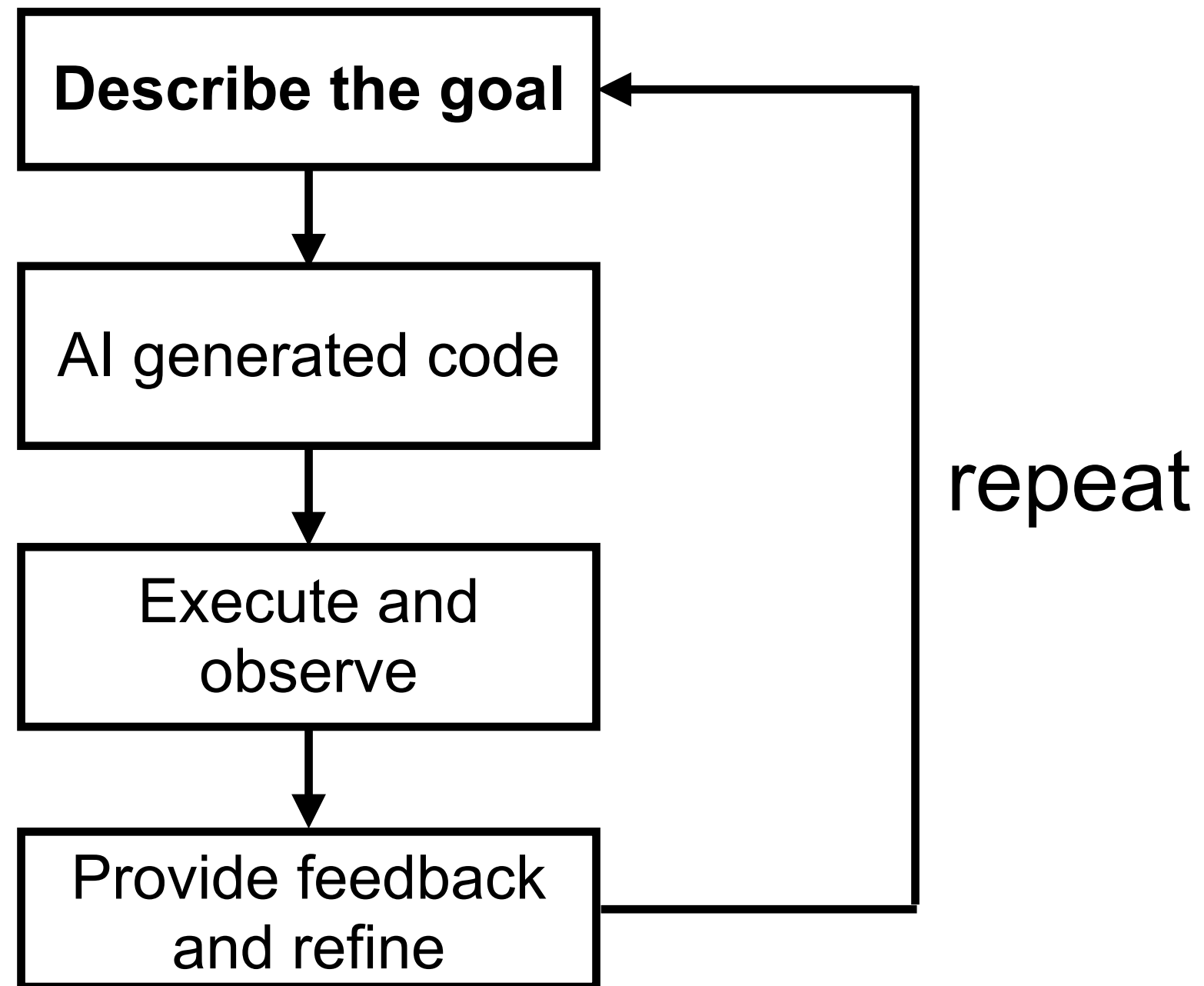
<https://cloud.google.com/discover/what-is-vibe-coding>





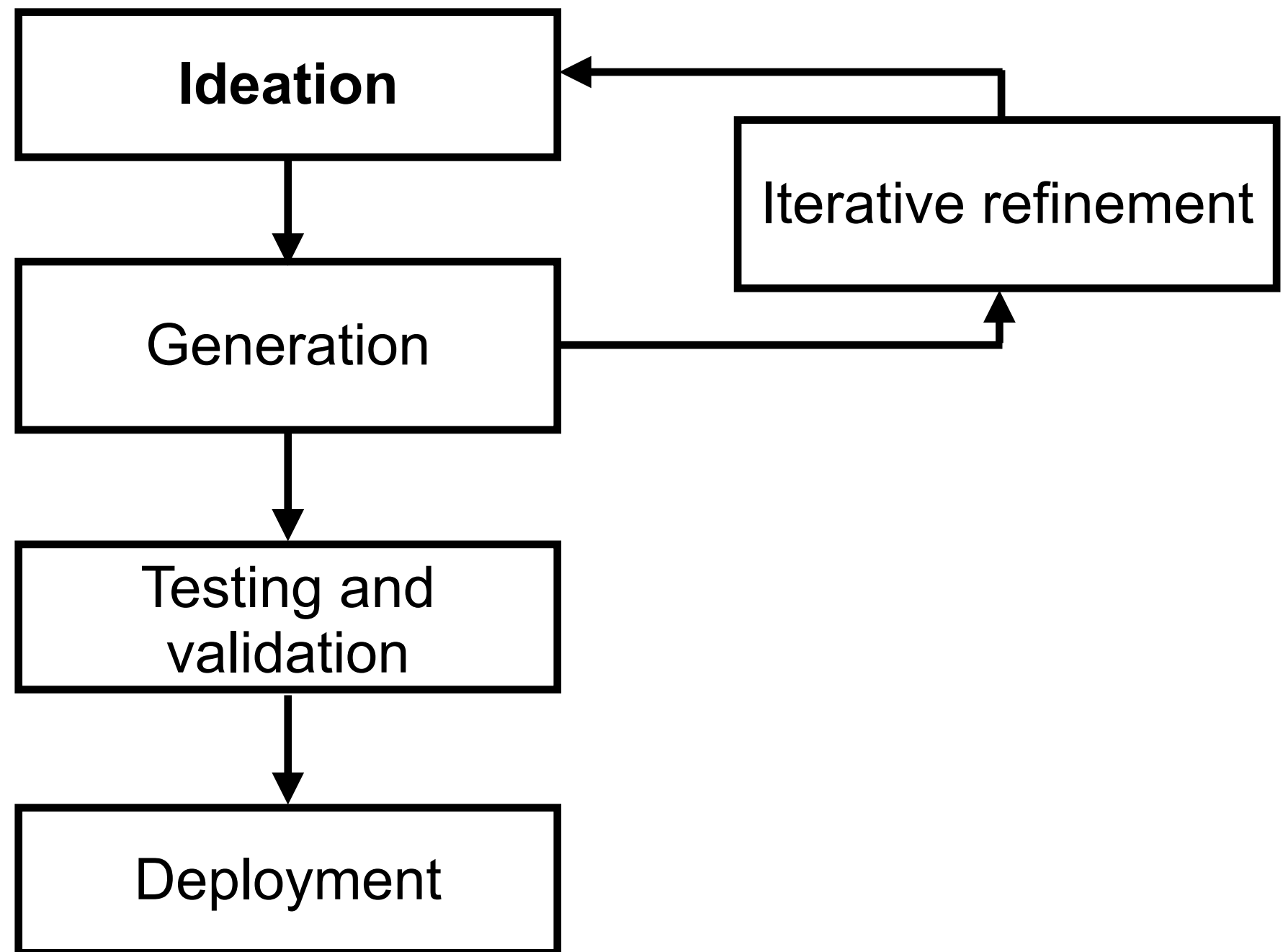
# Code level workflow

Conversational loop to create a specific code



# Application level workflow

Talking about high-level idea from concept to deployment



# AI Agent for Coding

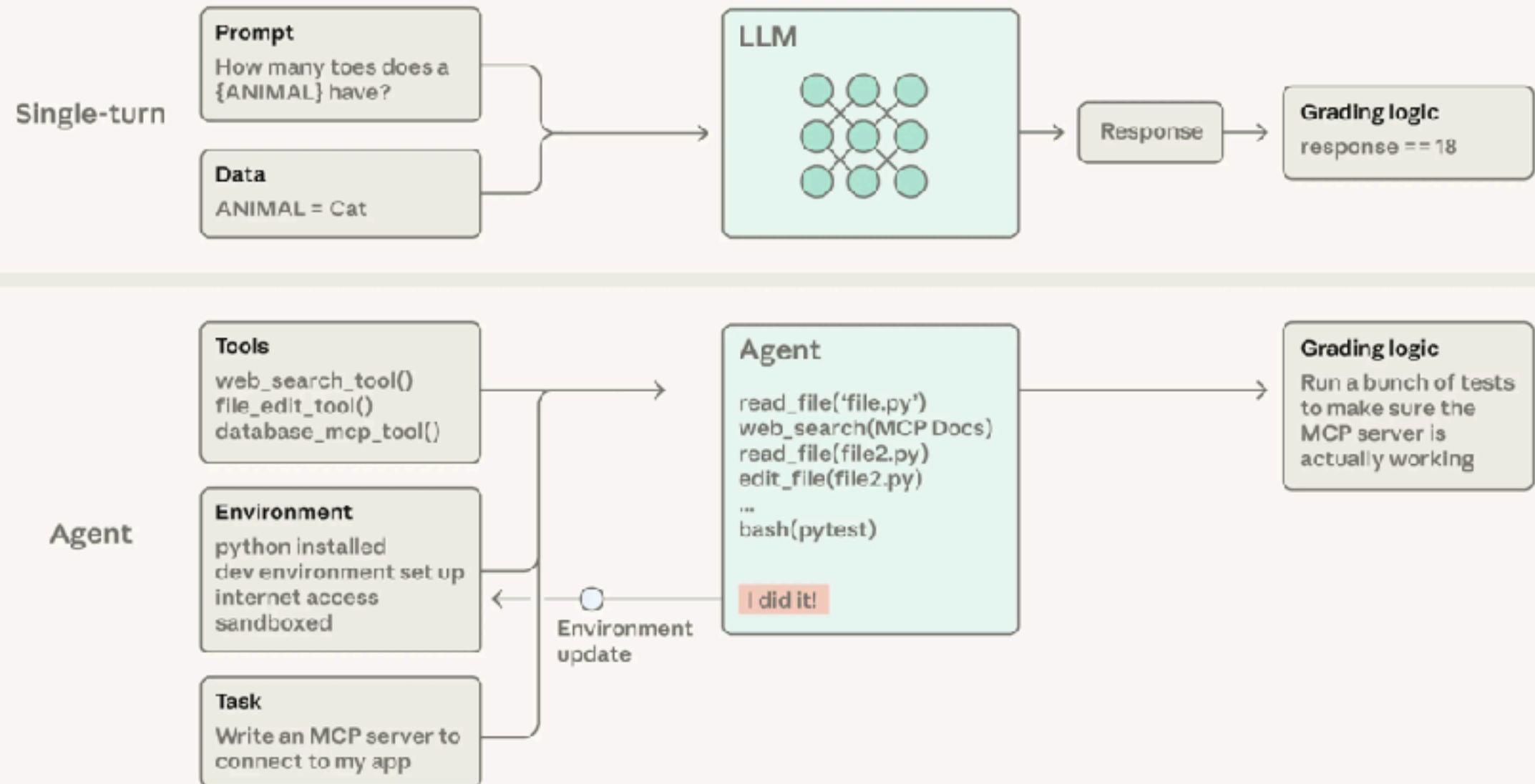


# AI Agent for Coding



# AI Agent for Coding

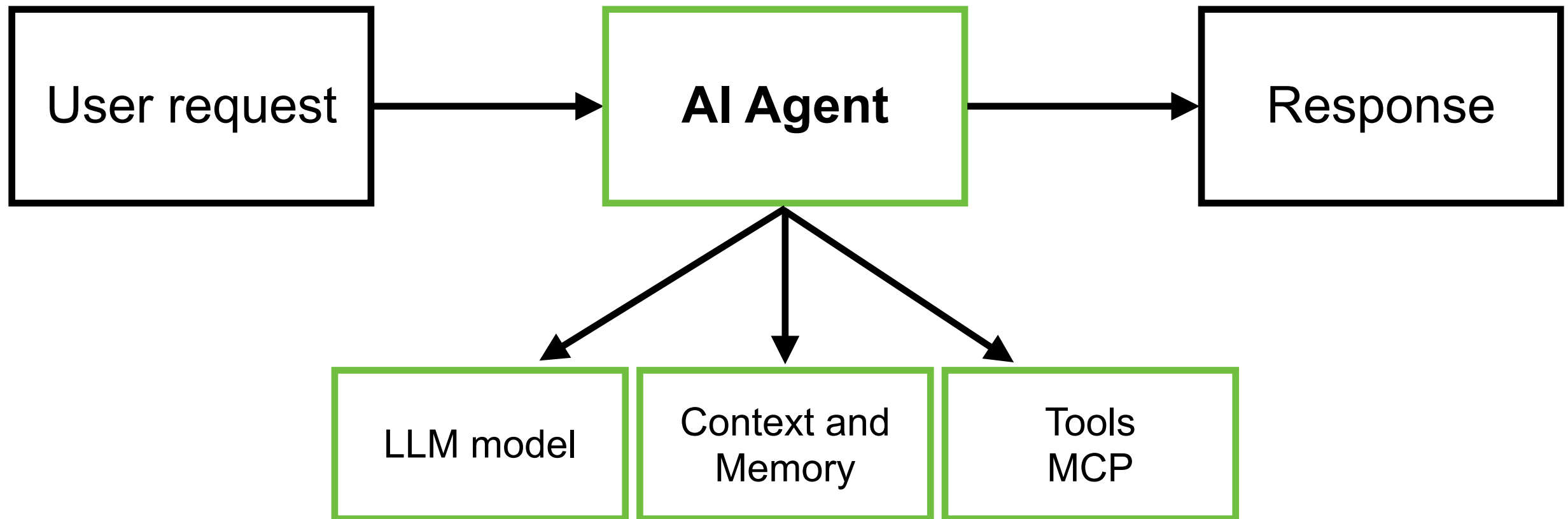
## Comparison: Single-Turn vs Agent Evaluations



<https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents>



# Simple AI Agent



How to manage context or knowledge for Agent ?



# Types of Knowledge/Context ?

## Shared context

Static Knowledge

Prompt engineering  
Global Instructions  
AGENTS.md  
CLUDE.md

## Task context

Dynamic Knowledge

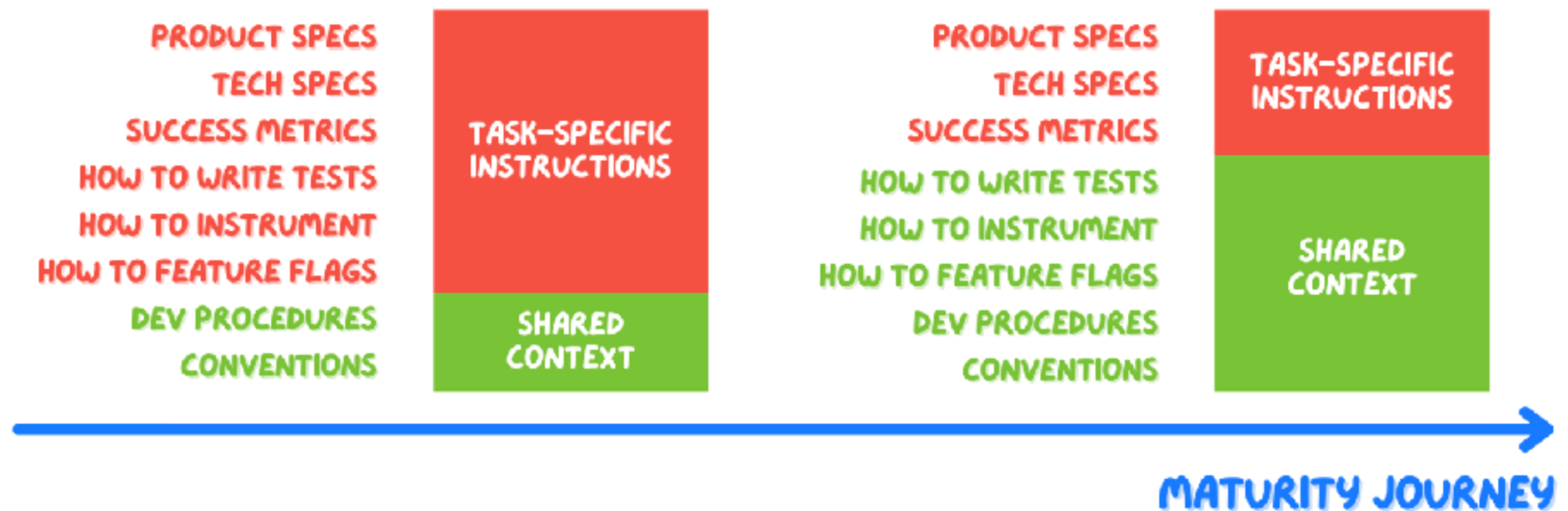
Memory  
Graph/Vector database  
MCP tools  
Specific Instructions  
Agent SKILLS

Sub-agent

**Context engineering ?**



# Types of Knowledge/Context ?



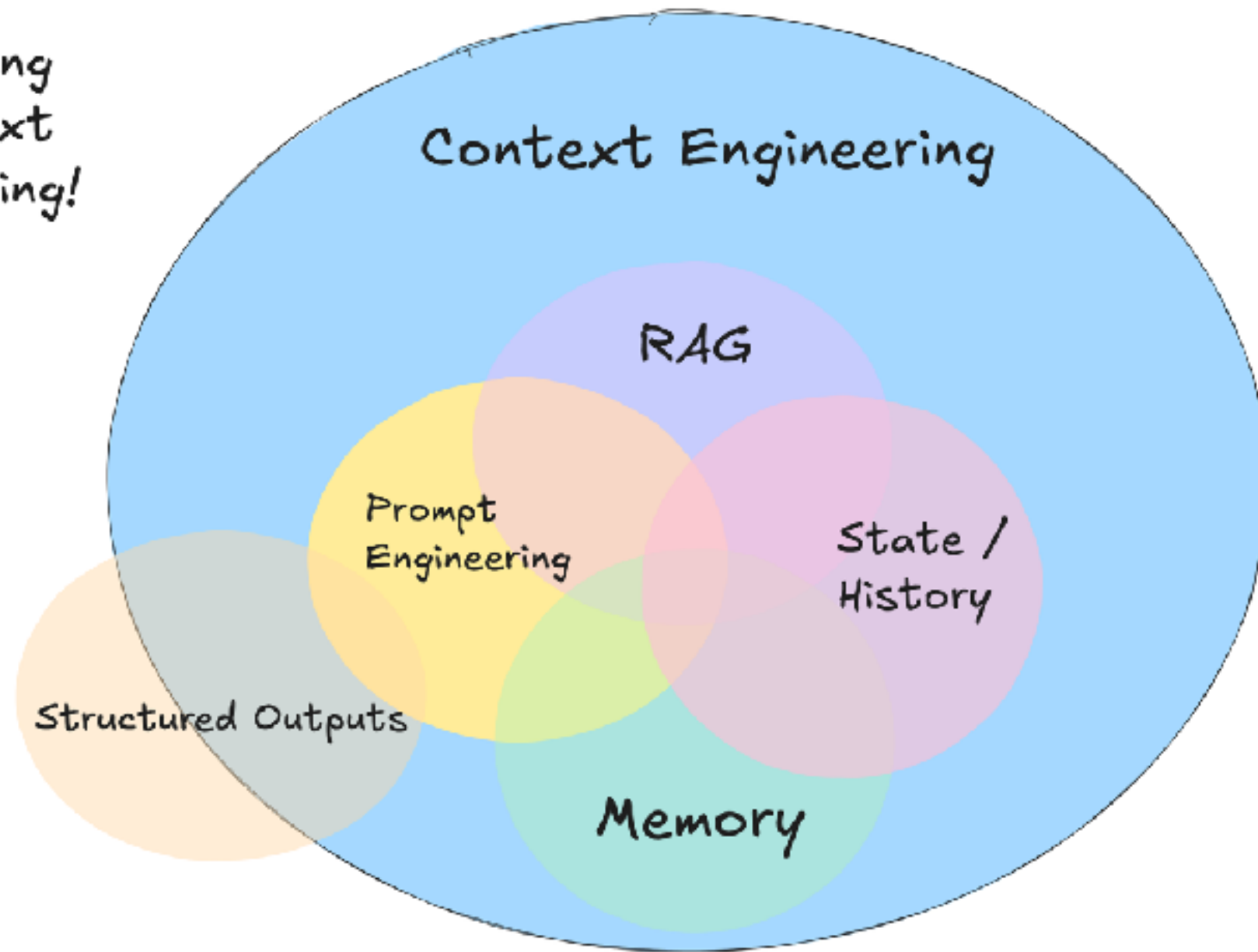
<https://refactoring.fm/p/context-capabilities-and-tech-hubs>





# Context Engineering

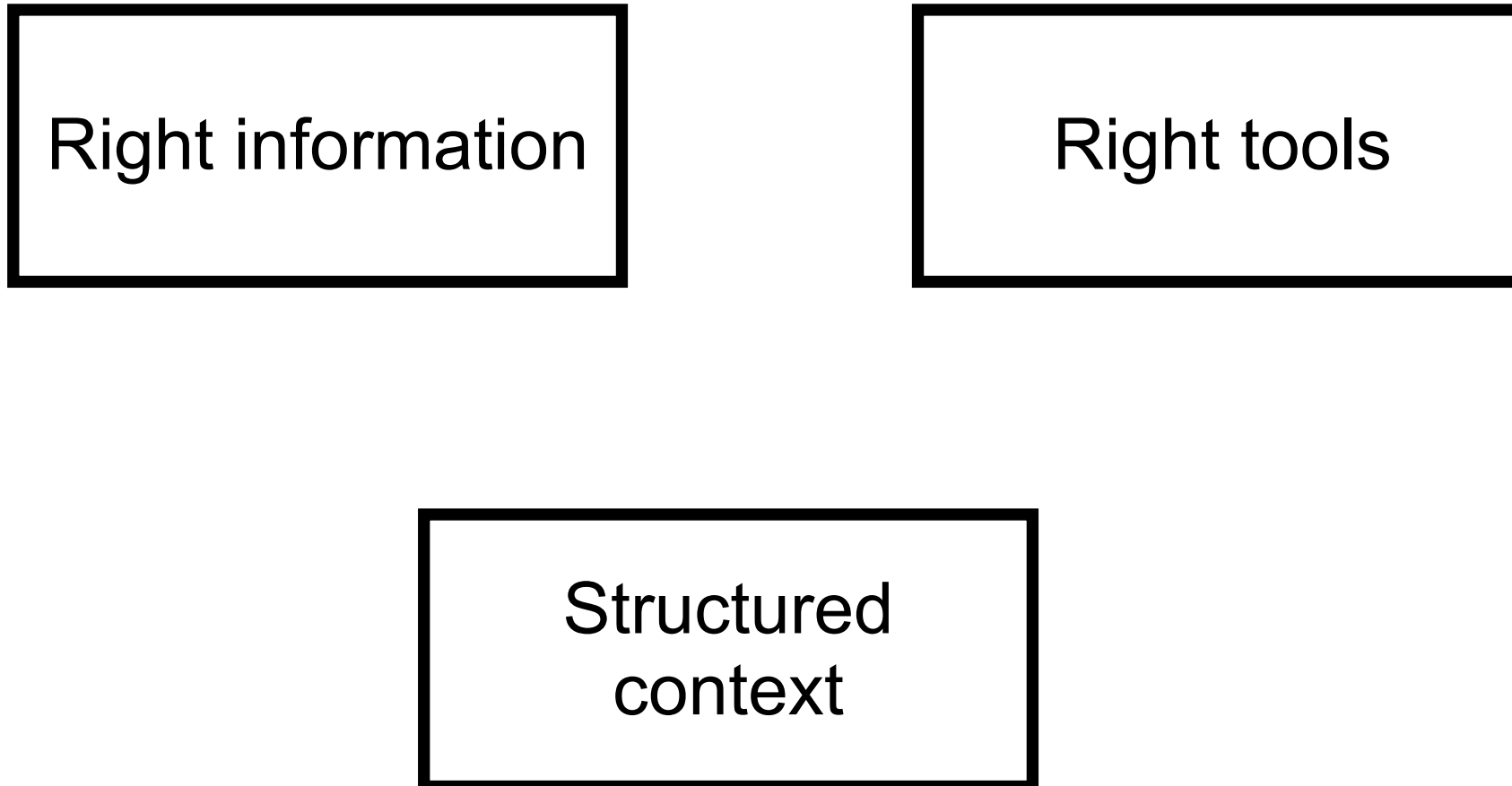
Everything  
is Context  
Engineering!



<https://www.promptingguide.ai/guides/context-engineering-guide>



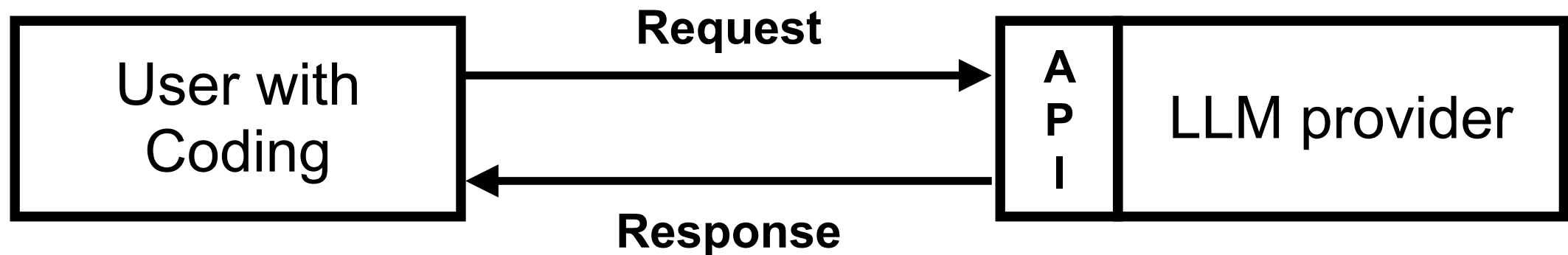
# Context Engineering



# Context window size ?

Amount of text a model can process and remember  
at once time

Measure with **token**



# Context window size of models

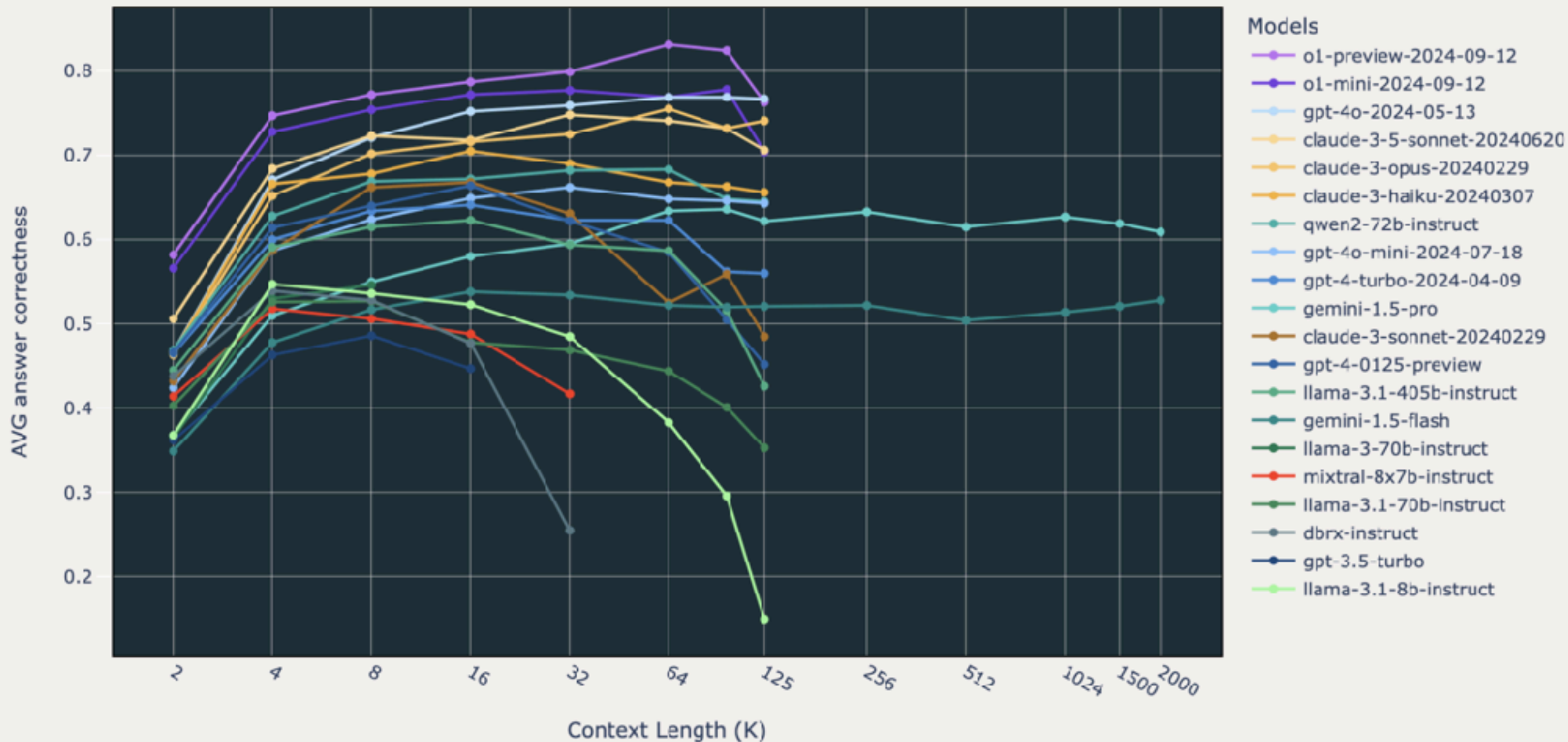
Different models may have different window size

Model name	Context window size
GPT 4.1	1M
Gemini 1.5, 2.5	1M
Llama 4	1M
Claude Sonnet 3.7	200,000
o4	200,000



# Long context LLM is better !!

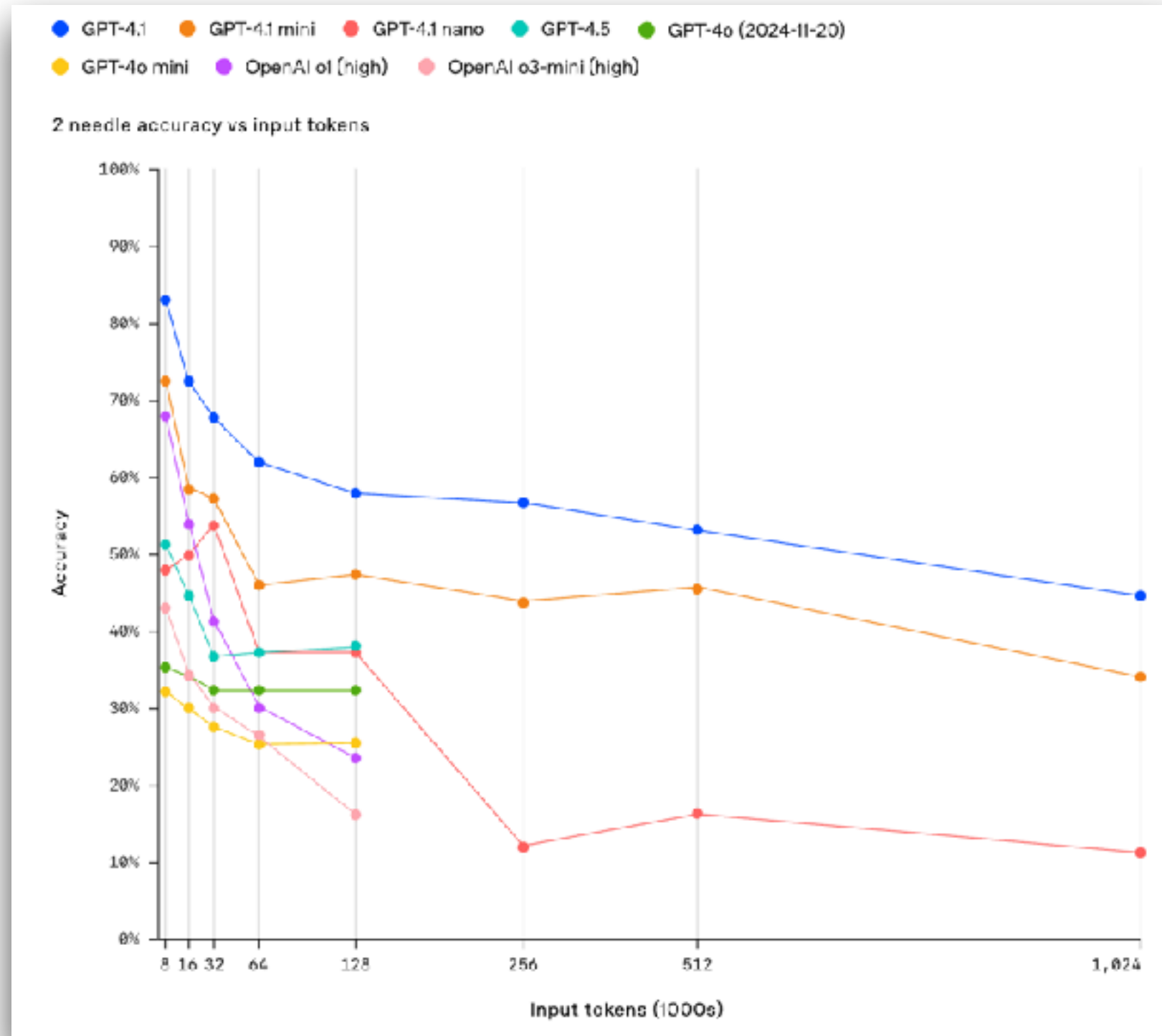
Long Context RAG Performance of LLMs



<https://arxiv.org/abs/2411.03538>



# Long context LLM is better !!



<https://openai.com/index/gpt-4-1/>



# **Better context == Better result**

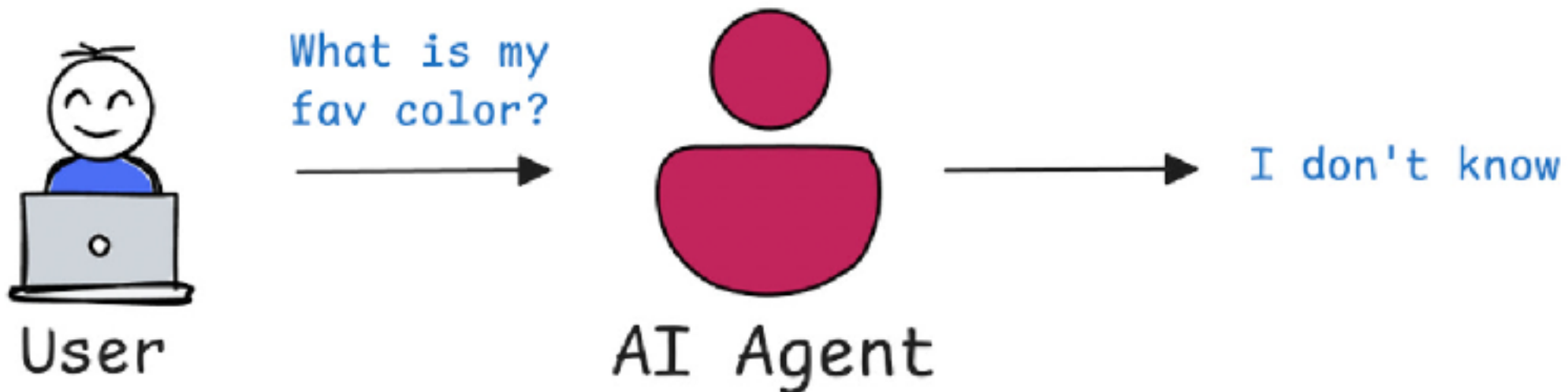
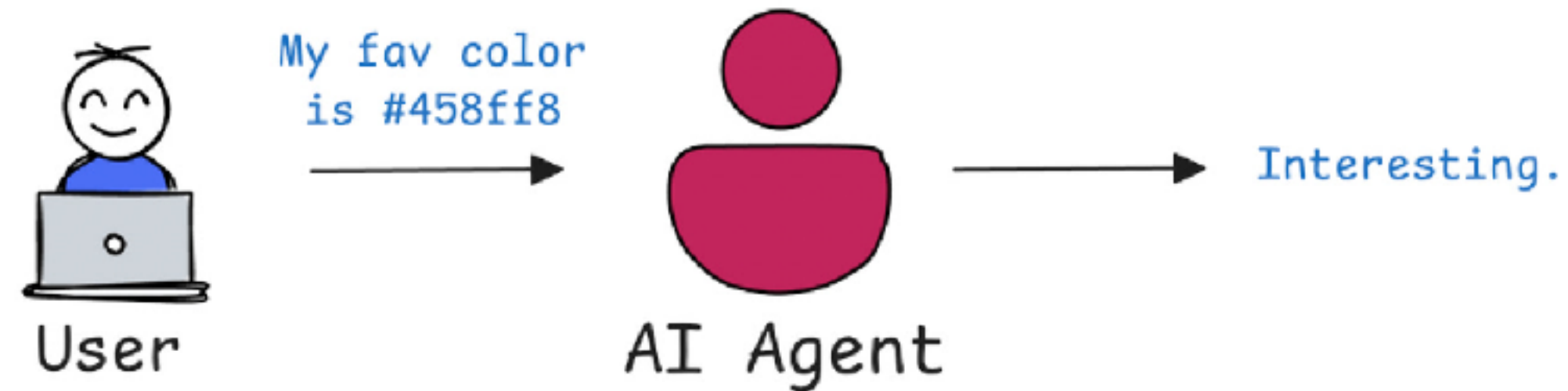


# Agent Memory Memory Bank





# Interaction without memory

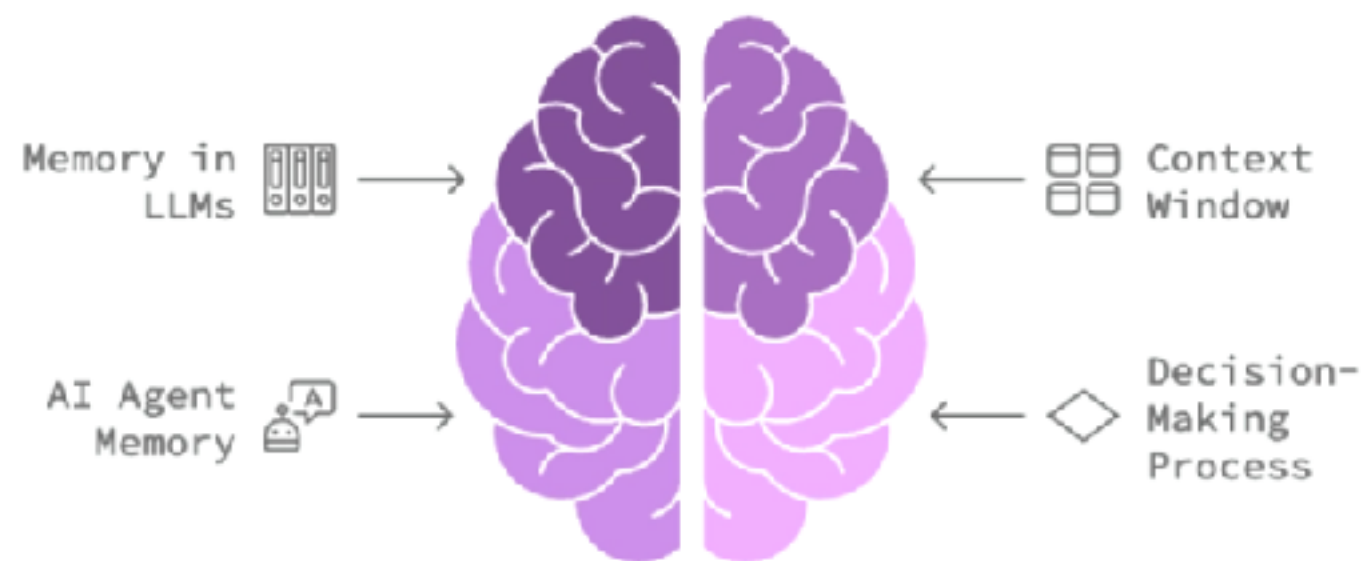


# Why AI Agent memory important ?

Enable personalized and context-aware interactions

Improve decision-making and adaptability

Enhance LLM's performance and response



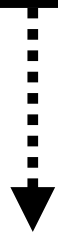
<https://www.falkordb.com/blog/ai-agents-memory-systems/>



# Types of Memory

To make intelligent decisions  
Agents combine different types of memory

Short-term



Current goals and steps  
File system (markdown, JSON)  
In-memory database

Episodic



Logs of past interactions  
Success and failure

Semantic

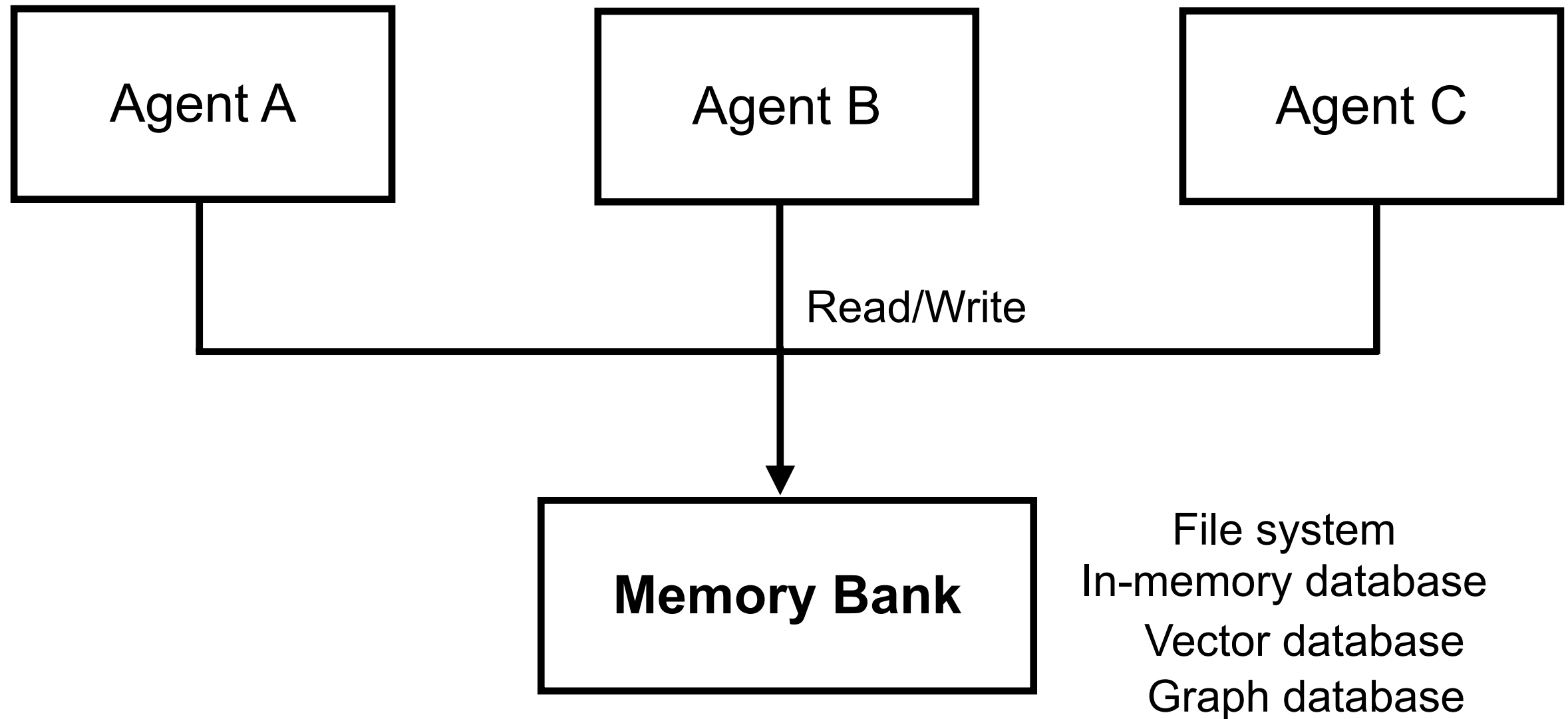


Long-term knowledge  
Fact and strategies  
RAG  
Vector and graph database

<https://bhavishyapandit9.substack.com/p/how-memory-works-in-agentic-ai-a>

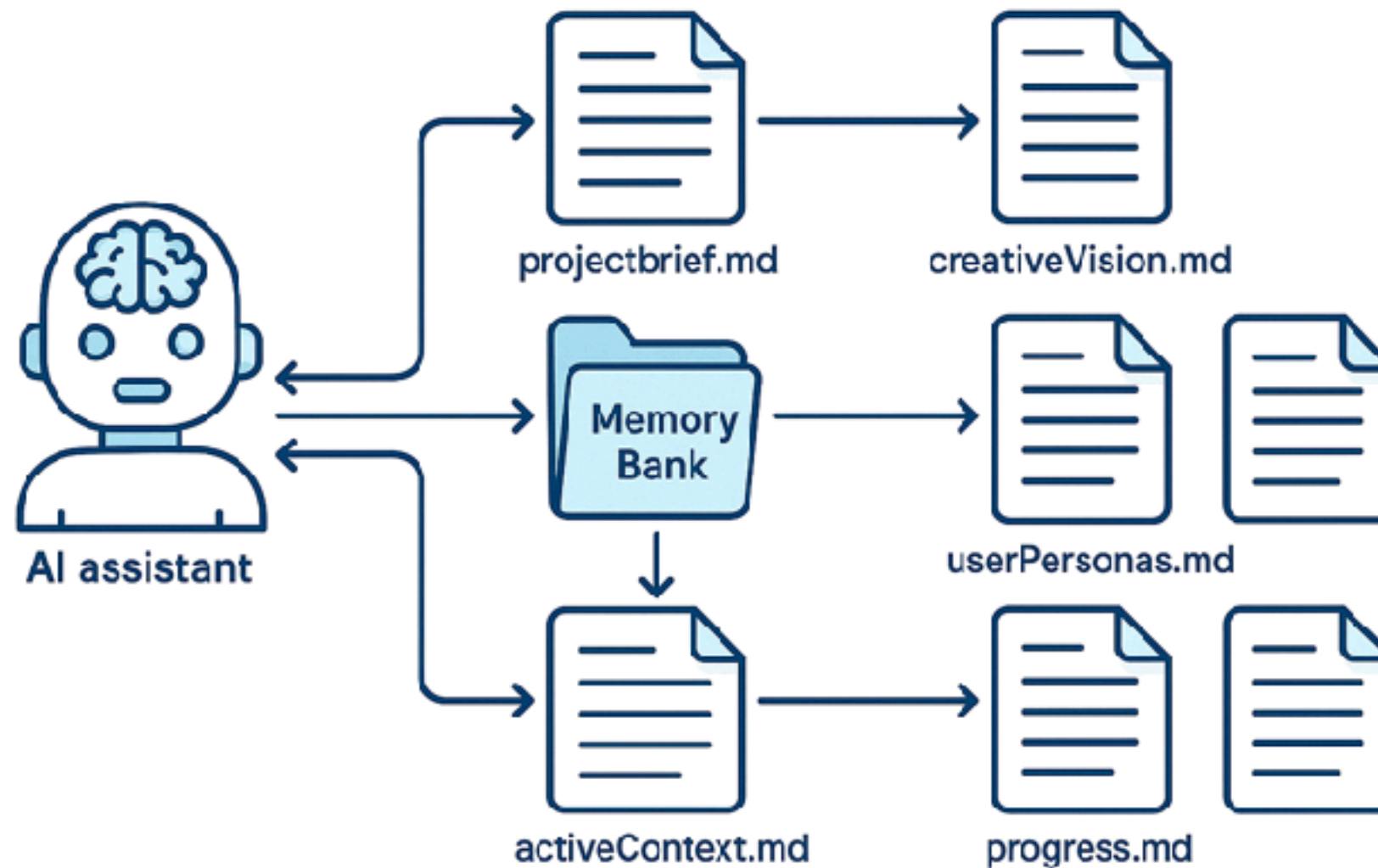


# Shared memory between Agent



# Cline Memory Bank

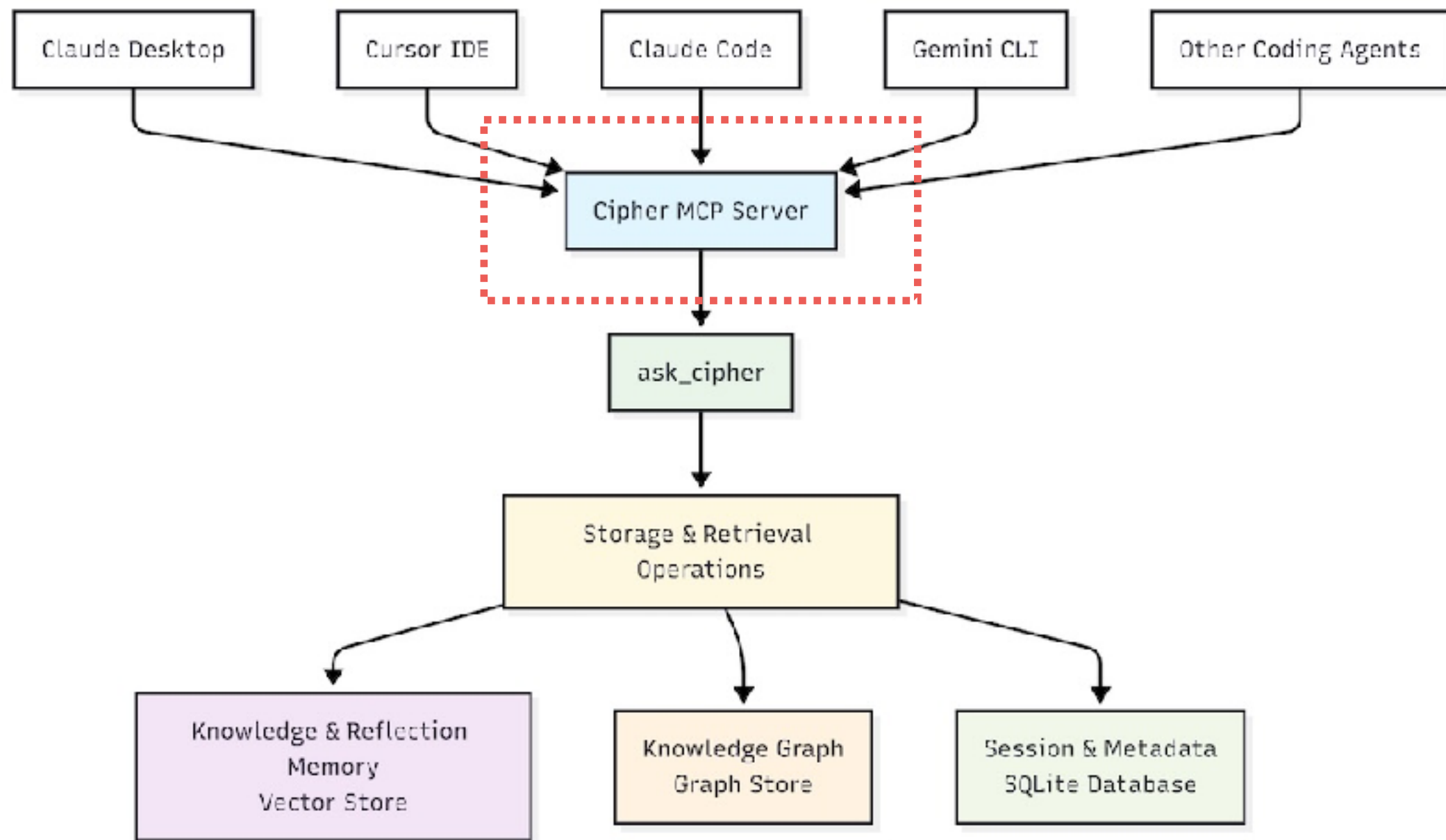
## Working with file system



<https://docs.cline.bot/prompting/cline-memory-bank>



# Memory Bank via MCP



<https://www.somkiat.cc/share-memory-for-ai-ide/>



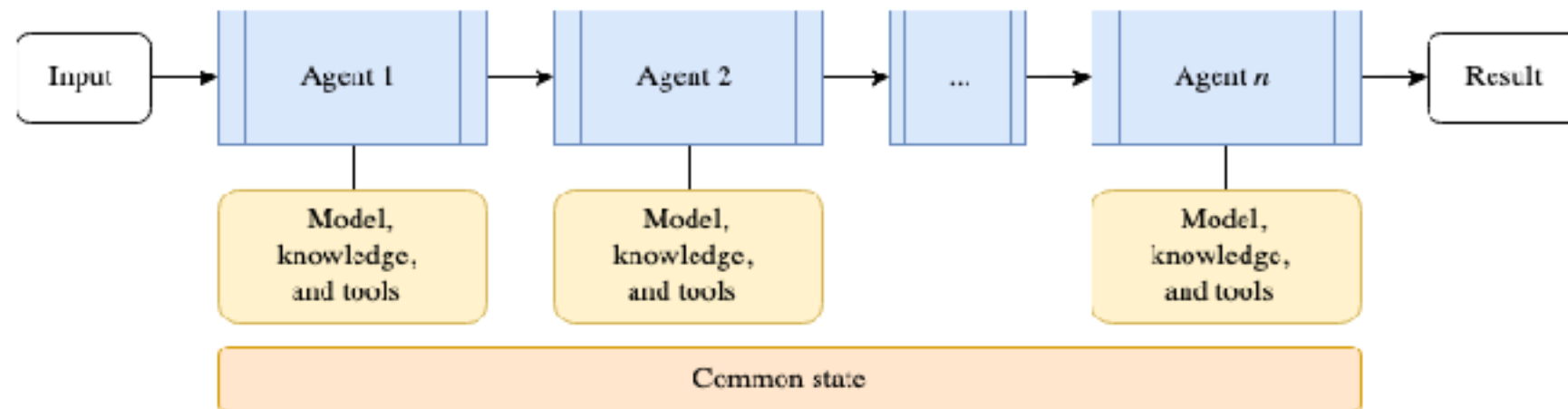
# Sub-agent and Multi-agent !!

**Reduce size of knowledge !!**



# Monolith to Multi-agent

Specialization  
Scalability  
Maintainability  
Optimization

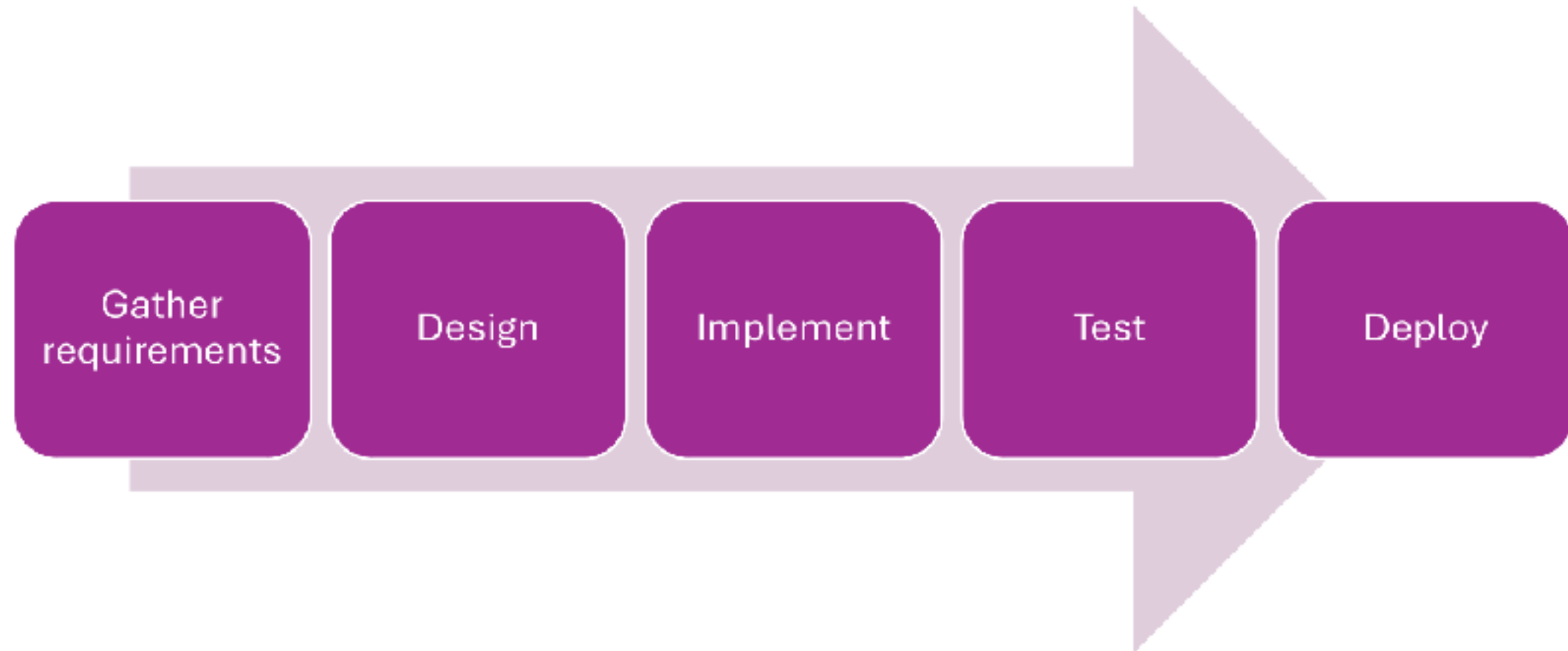


<https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/ai-agent-design-patterns>





# Design your Agents ?



<https://devblogs.microsoft.com/dotnet/introducing-microsoft-agent-framework-preview/>



# Design your Agents ?

Each step may contain subtasks  
Different specialists have different steps  
**Progress isn't always linear**

Bugs found during testing may send back to  
implementation



# Design your Agents ?

Full Stack Agent !!

Frontend

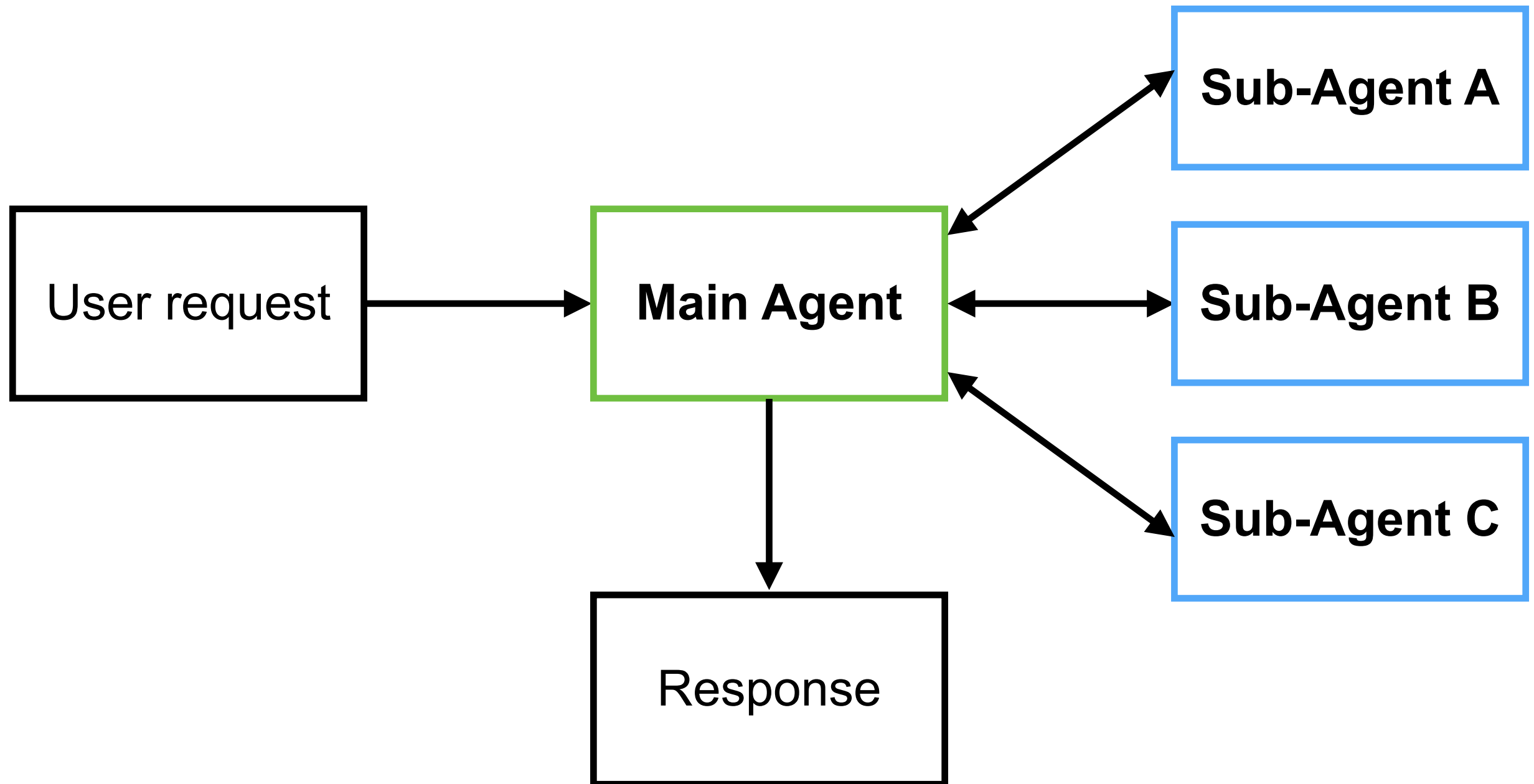
Backend

Database



# Multi AI Agents !!

Sub-agent with **centralized** orchestration

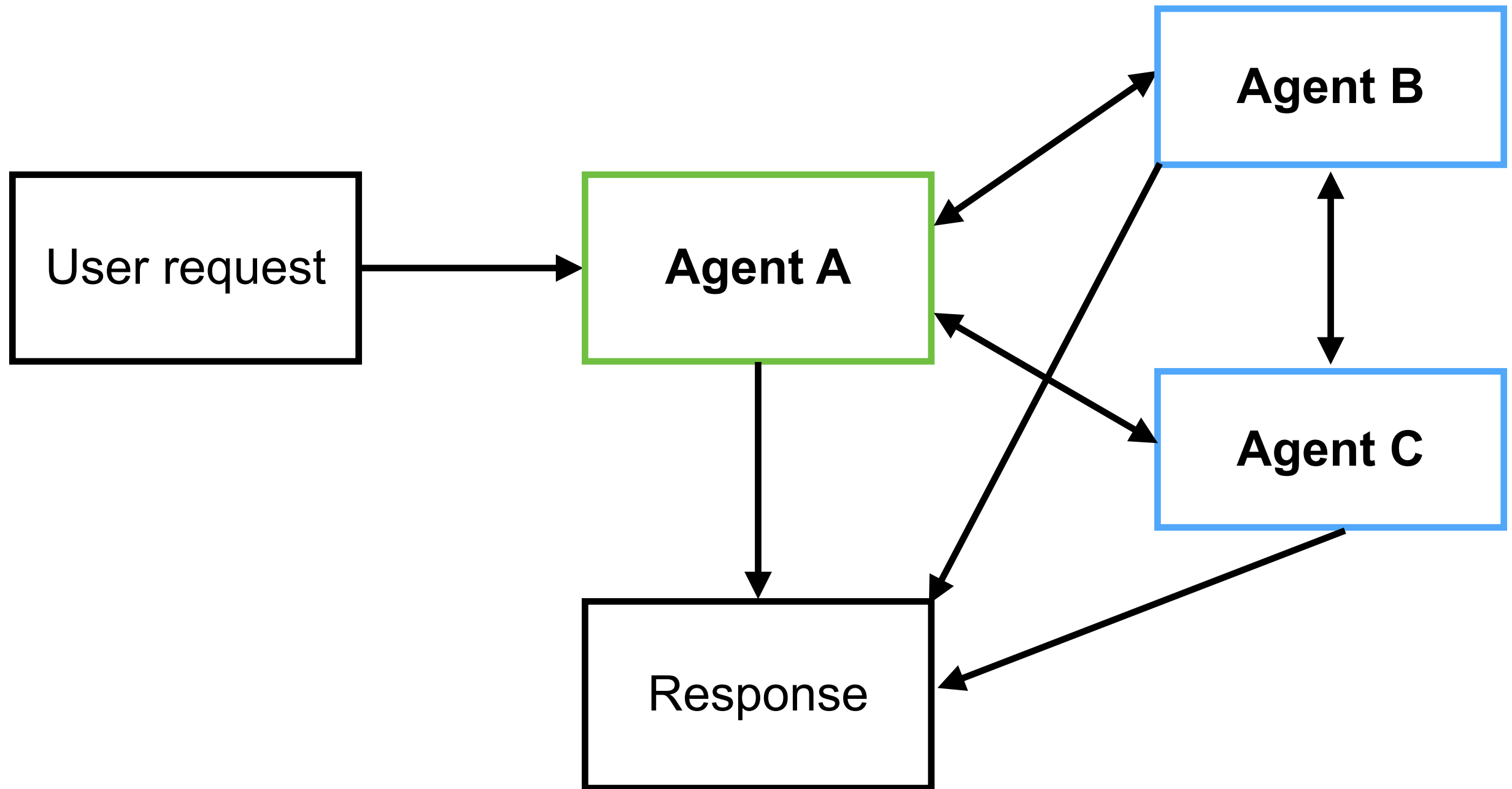


<https://www.blog.langchain.com/choosing-the-right-multi-agent-architecture/>



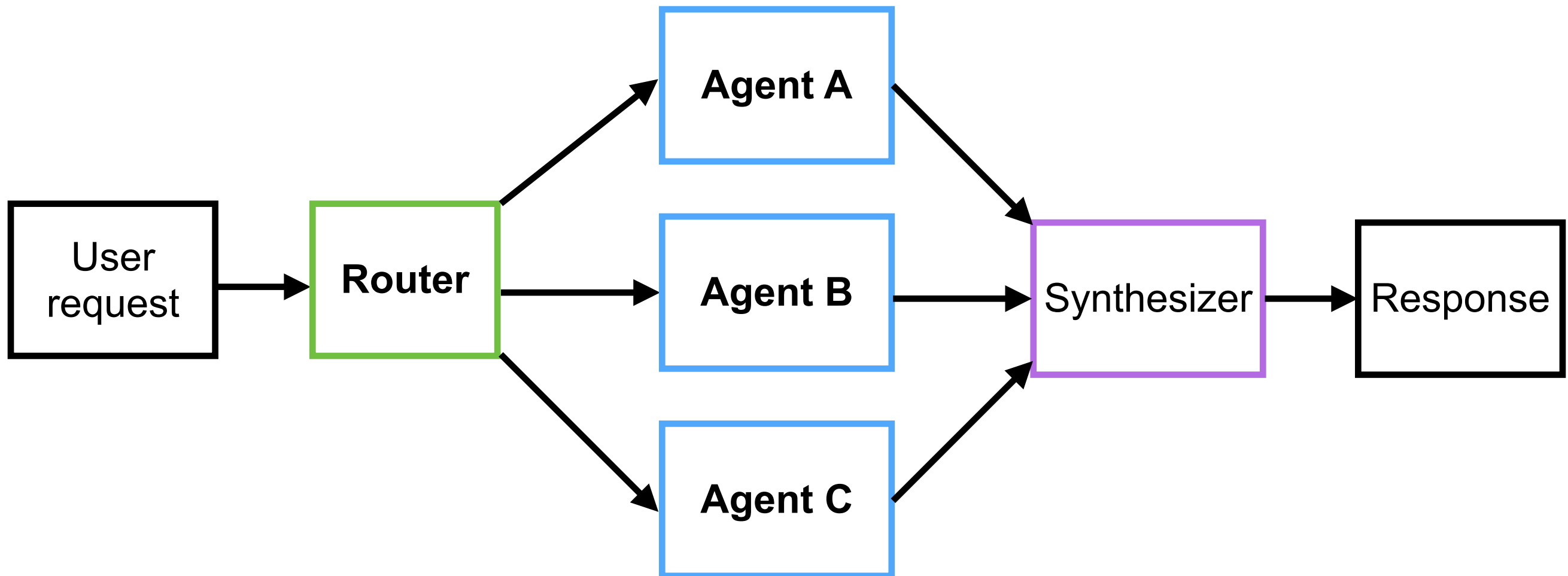
# Multi AI Agents !!

**State-driven transitions** based-on conversation context



# Multi AI Agents !!

Router and parallel run



# Prompt vs Agent Skills ?



Command

<https://agentskills.io/>



# Agent Skills

Moving from a paradigm of  
“prompt” to “programming by instruction”

Agent loads specialized prompts and knowledge on-demand

Specialized  
task

Reduce  
repetition

Compose  
capabilities

Efficient  
loading

Interoperability

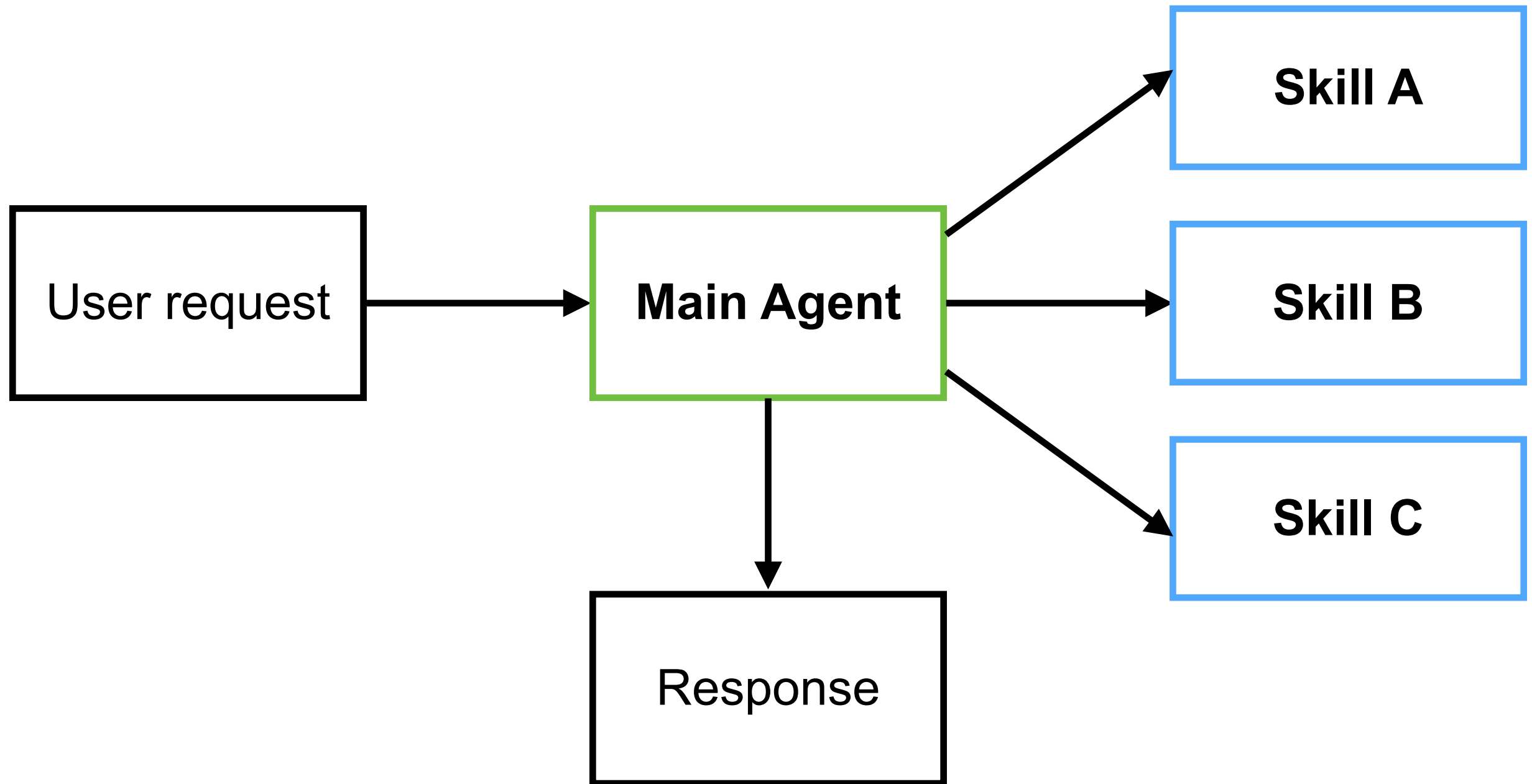
<https://agentskills.io/>





# Agent with Skills

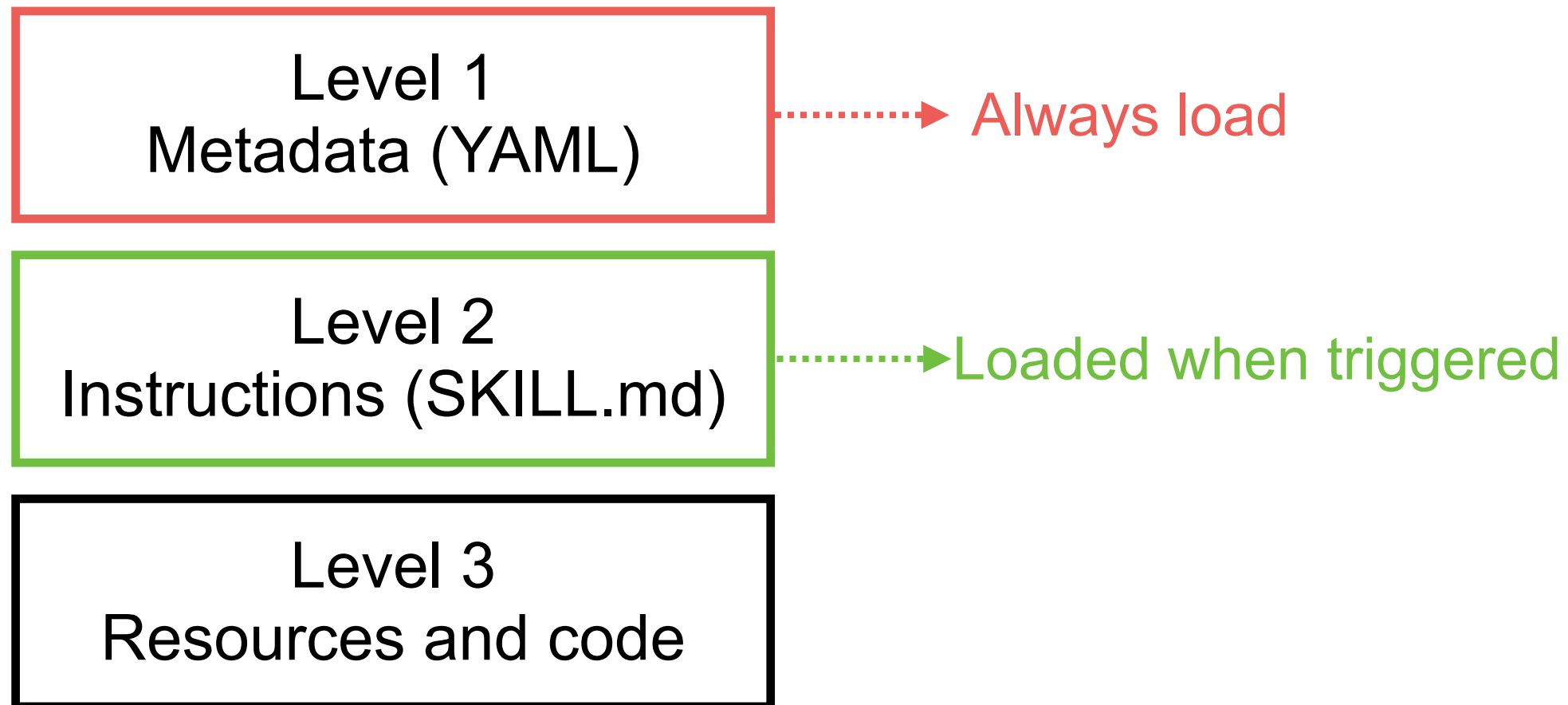
Load skills on-demand from context



<https://www.blog.langchain.com/choosing-the-right-multi-agent-architecture/>



# Structure of Agent Skills



<https://agentskills.io/>



# Prompt vs Skill

Features	Prompt	Agent Skills
Persistence	Ephemeral (chat-specific)	Persistent (Useable cross conversation)
Activation	Manual by user	Automatic (Triggered by context)
Structure	Unstructured text/ instruction	Structured markdown File system-based
Context load	Full instruction	Load only when needed
Token cost	Higher (Repeated usage)	Lower (Load on-demand)
Use case	One-time, creative quick query	Repeatable workflows Specialized tasks

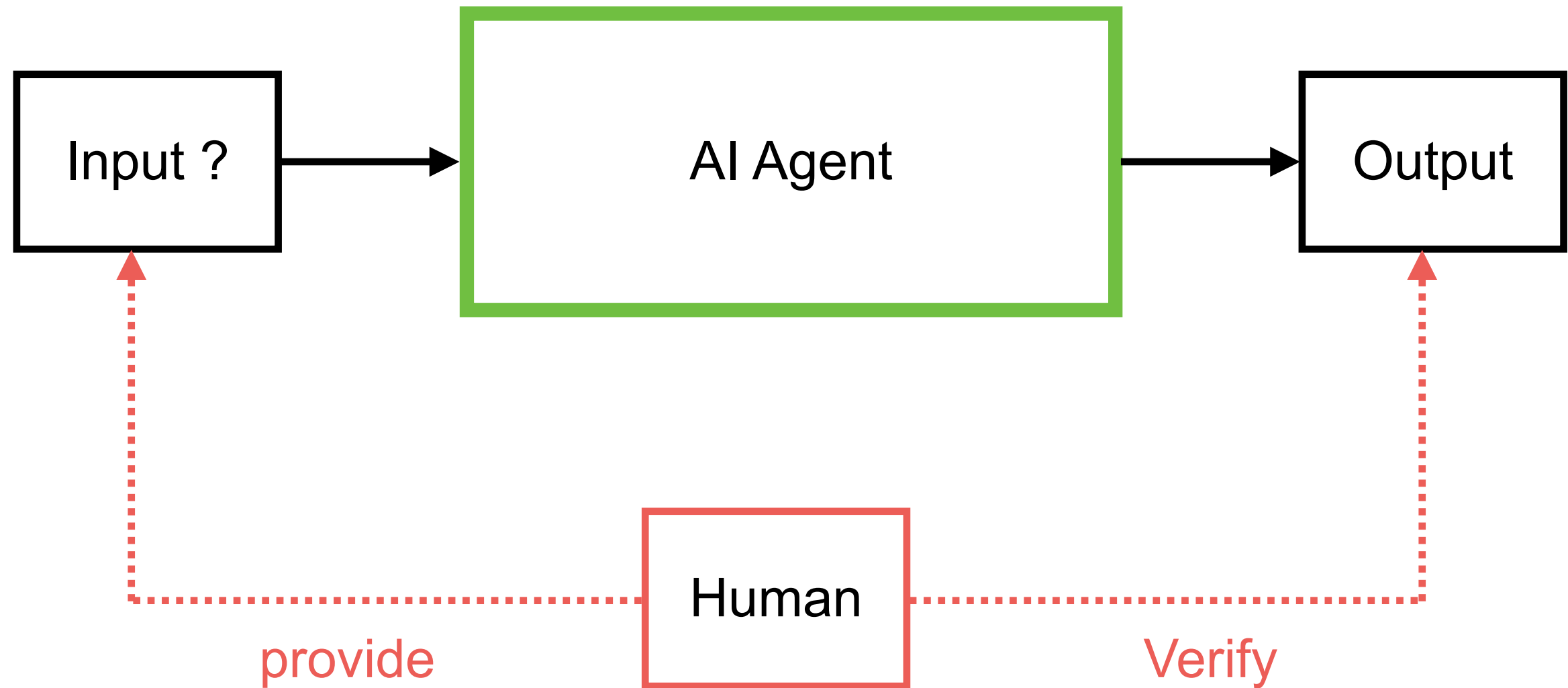
<https://agentskills.io/>



# How to provide inputs ?



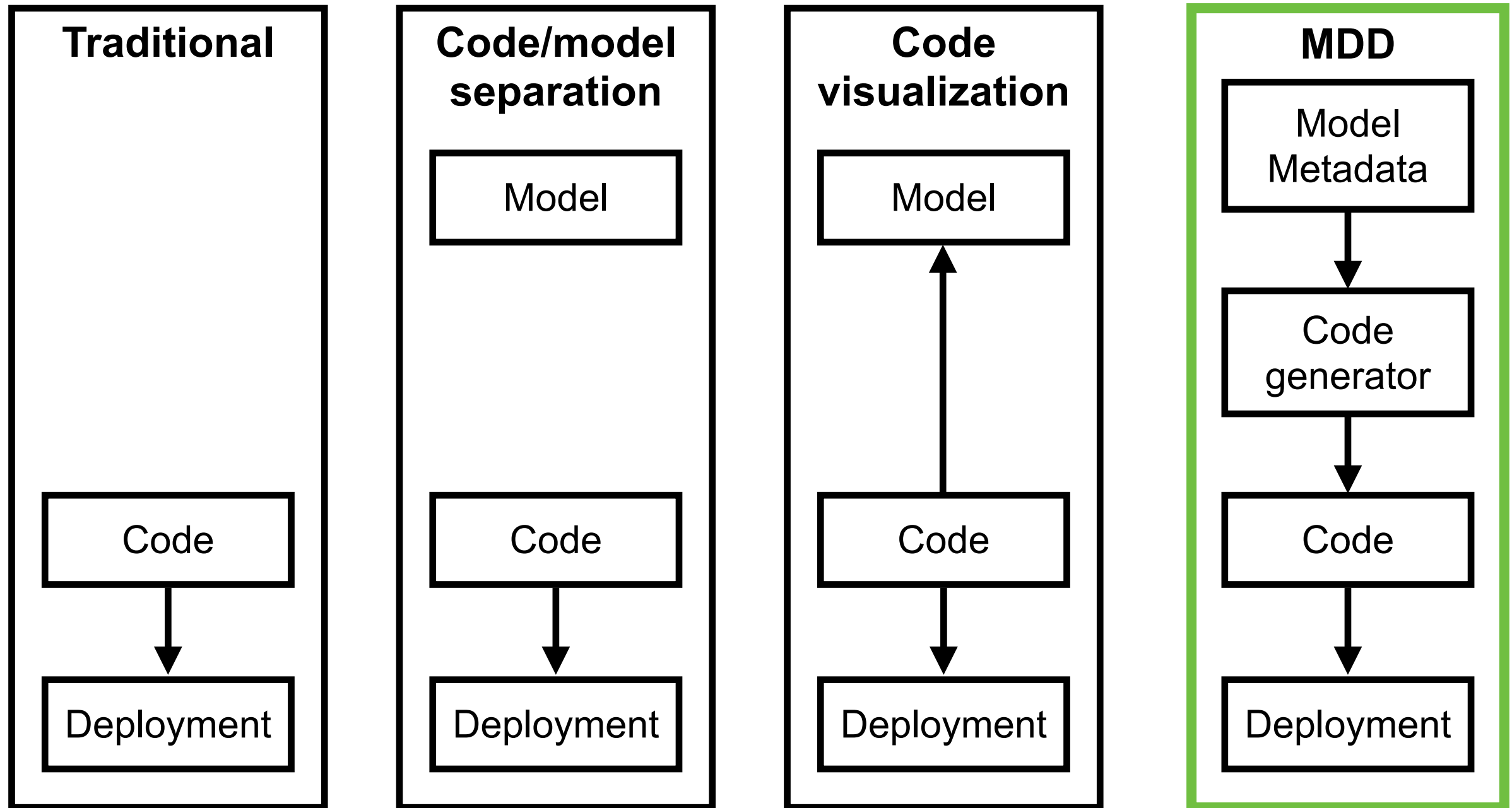
# How to provide inputs ?



# Model-Driven Development (MDD)



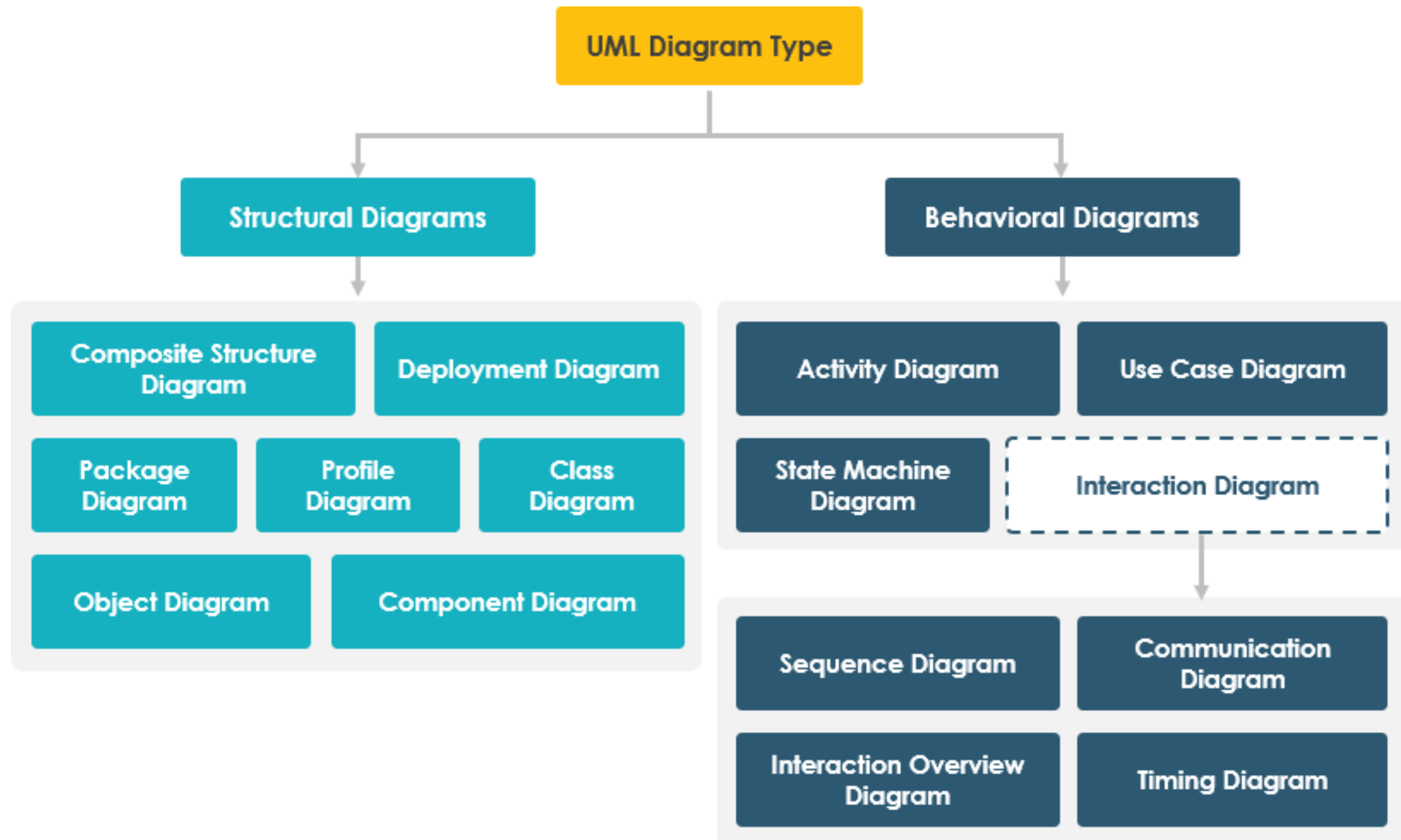
# Traditional coding to MDD



<https://arxiv.org/abs/2410.18489>



# UML Diagrams !!

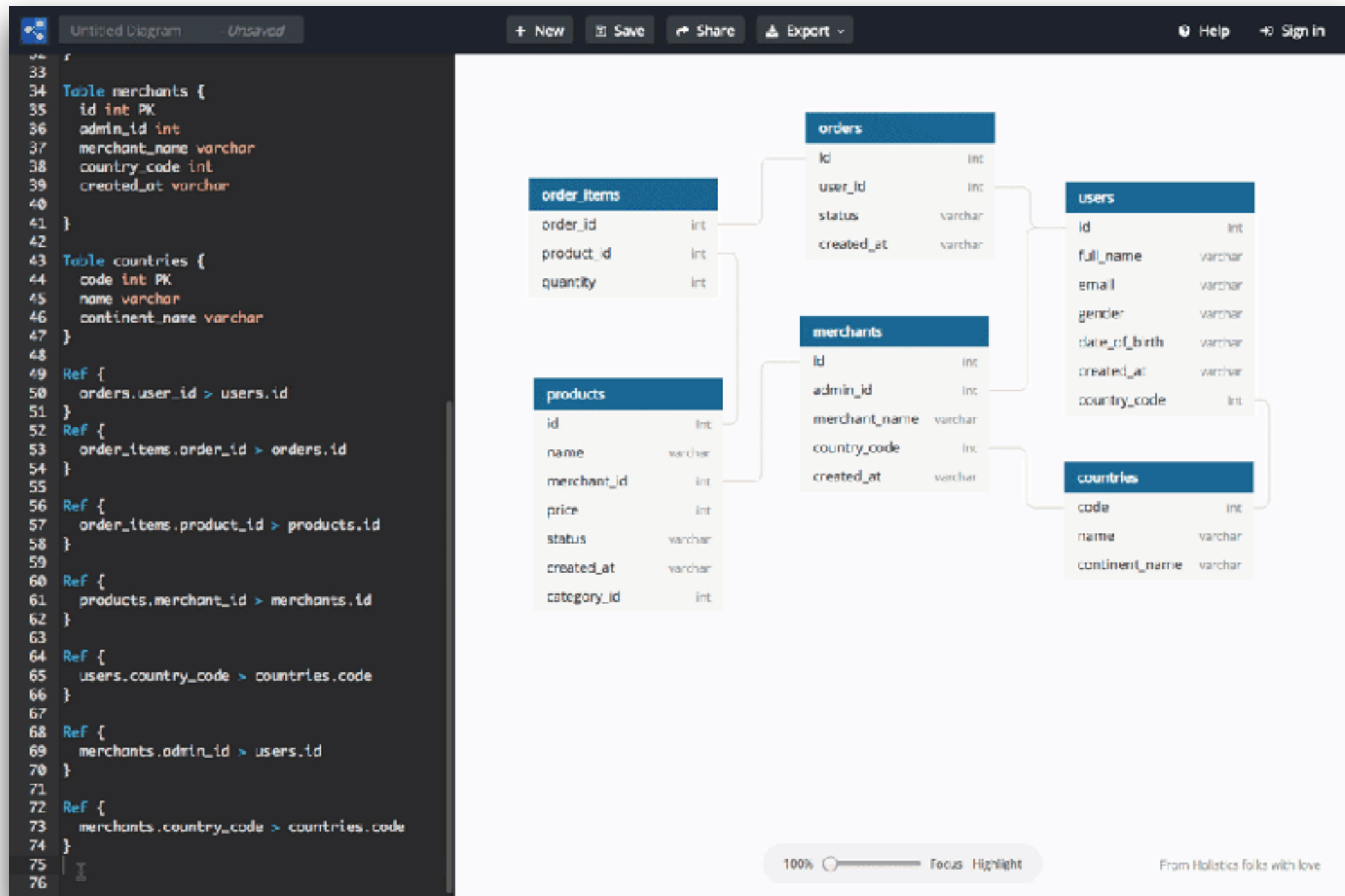


<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/behavior-vs-structural-diagram/>





# ER Diagrams for Database !!



<https://dbml.dbdiagram.io/home/>



# Mermaid Diagram

Diagram as a Code (AI-friendly)

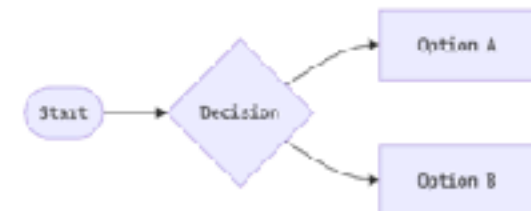
## Mermaid Diagramming and charting tool

JavaScript based diagramming and charting tool that renders Markdown-inspired text definitions to create and modify diagrams dynamically.

Try Editor

Get started

```
1 flowchart LR
2   A(["Start"])
3   A --> B["Decision"]
4   B --> C["Option A"]
5   B --> D["Option B"]
```



<https://mermaid.js.org/>



# Swagger or OpenAPI

Standardized specification for describe REST APIs

The screenshot displays the Swagger.io homepage. On the left, a hero section titled 'The Future of AI Relies on API Quality' features a green highlight on 'API Quality'. Below the title, text states: 'Swagger enables design, governance, and testing across the full AI-enabled API lifecycle, ensuring quality at every step.' and 'Build APIs ready for humans, LLMs, agents, and continuous innovation.' There are two buttons: 'Find your tool' and 'Read the docs →'. At the bottom left, it says 'TRUSTED BY'. On the right, a preview of an API specification for 'pet' is shown, listing several endpoints with their methods and descriptions.

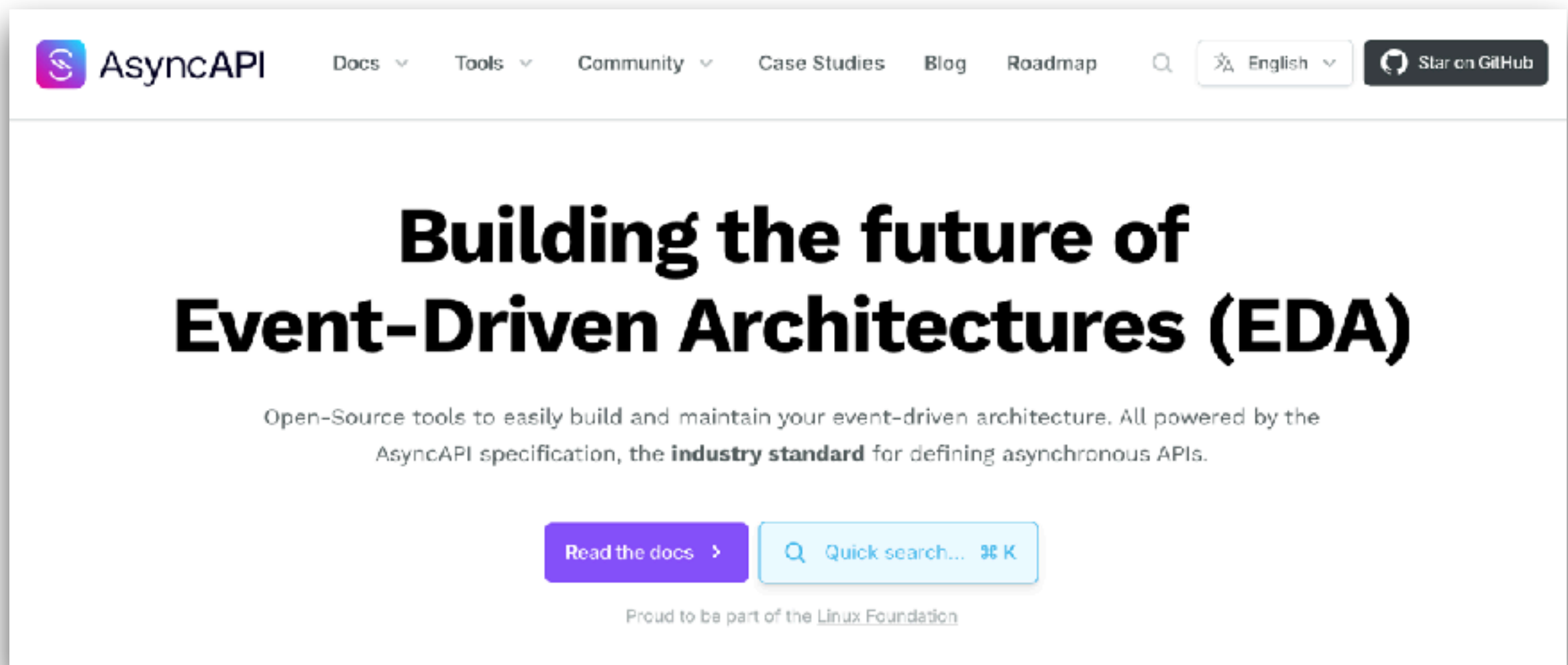
Method	Endpoint	Description
POST	/pet/{petId}/uploadImage	uploads an image
POST	/pet	Add a new pet to the store
PUT	/pet	Update an existing pet
GET	/pet/findByStatus	Finds Pets by status
GET	/pet/{petId}	Find pet by ID
POST	/pet/{petId}	Updates a pet in the store with form data
DELETE	/pet/{petId}	Deletes a pet

<https://swagger.io/>



# AsyncAPI

Standardized specification for describe asynchronous and event-driven APIs (Kafka, AMQP, WebSocket)



<https://www.asyncapi.com/>



# JSONSchema

Declarative language used to annotate and validate the structure, constraints and data type of JSON documents

Data validation

Documentation

Automated  
testing

<https://json-schema.org/>



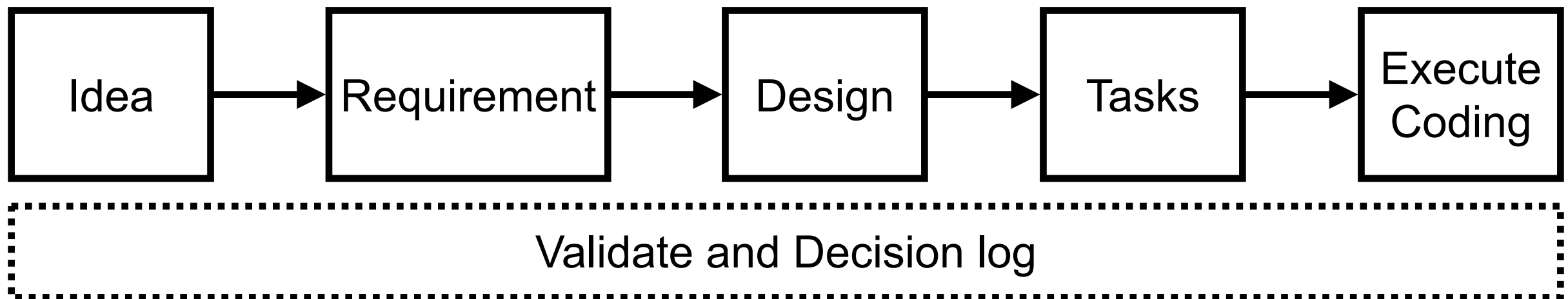
# Specification-Driven Development (SDD)



# SDD ?

## Specification or Documentation-First

Modern software development approach where detailed, structure specifications are created before writing code



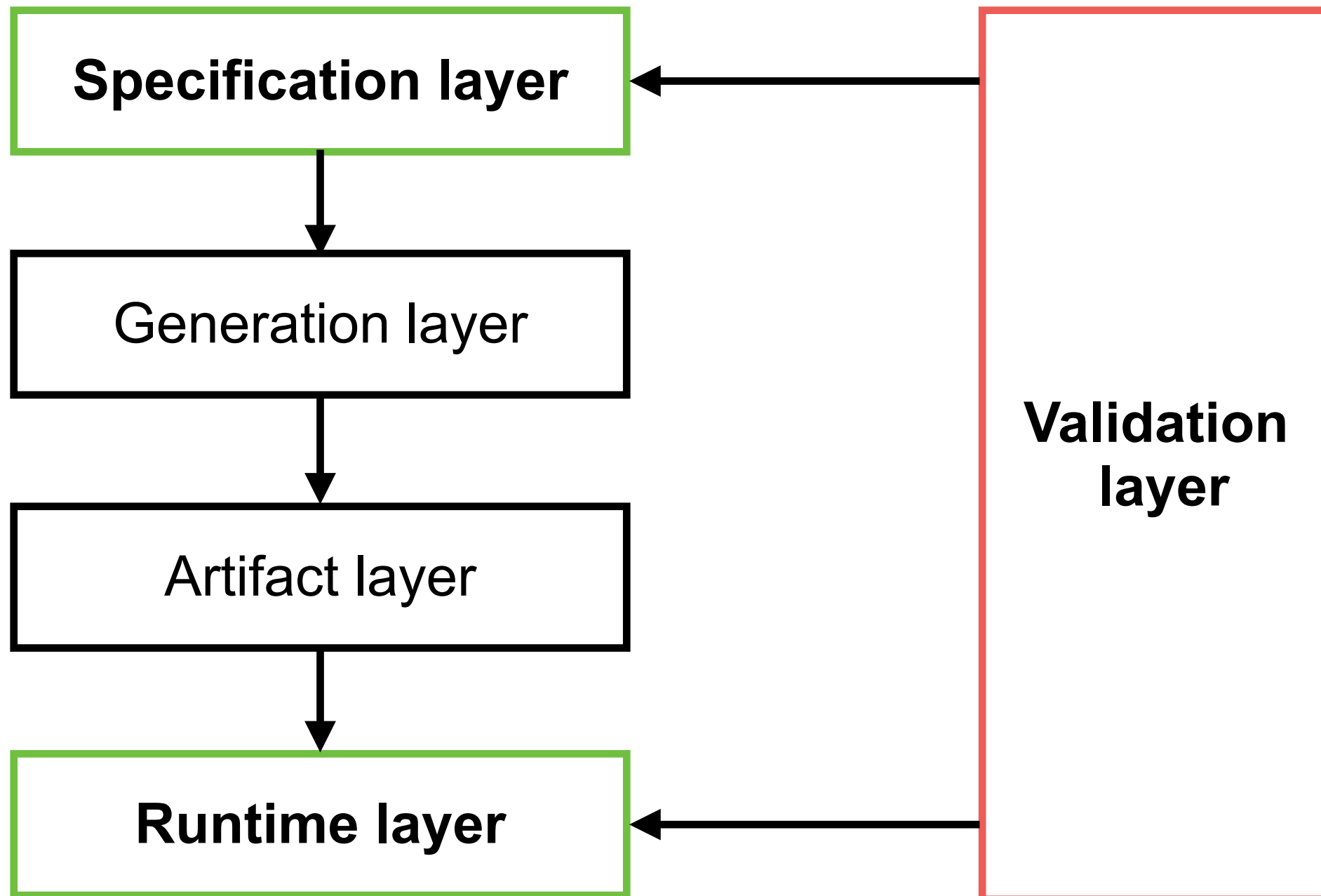
# Types of Specification ?

User requirements  
Design documents  
Interface contracts (UI, API)  
Data schemas  
Validation rules  
Data/business flows  
Security policy and constraints  
Resources and performance constraints





# SSD Architecture



<https://www.infoq.com/articles/spec-driven-development>



**MDD + SDD == Better result**



# When to use SDD ?

No one-size-fit-all !!

**Low complexity**

**High complexity**

<https://medium.com/google-cloud/benefits-and-challenges-of-spec-driven-development-and-how-antigravity-is-changing-the-game-3343a6942330>



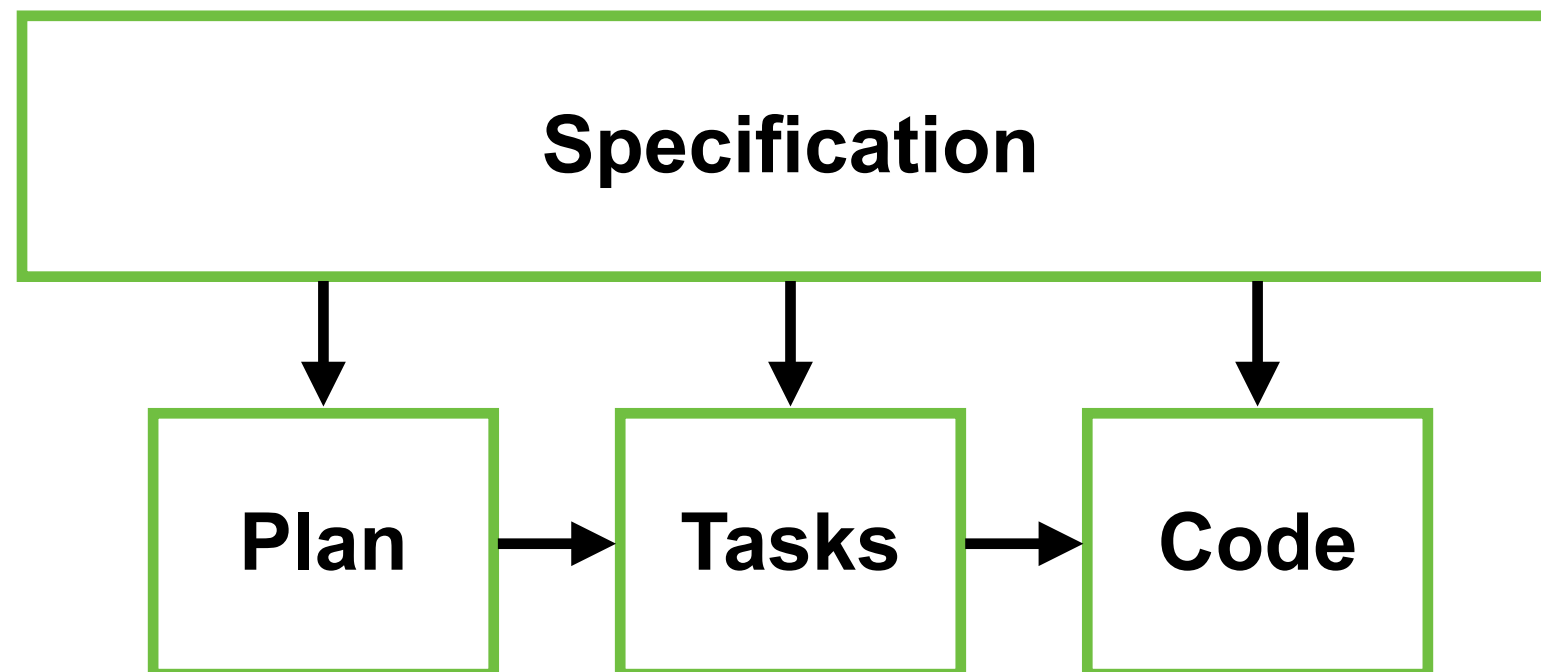
# SDD Tools ?

GitHub Spec-kit

Open Spec

Kiro.dev

Tessl



# Try by yourself !!





**Frontend**  
Web application

**Backend**  
RESTful API +  
Database

Write specification

Coding

Run and Test !!



**Frontend**  
Web application

**Backend**  
RESTful API +  
Database

Vercel React Skill

Write spec or feature

Prompt or command

Working with MCP





**Frontend**  
Web application

Vercel React Skill

Write spec or feature

Prompt or command

Working with MCP

**Backend**  
RESTful API +  
Database

Global instruction

Write spec or feature

Prompt or command

Working with MCP

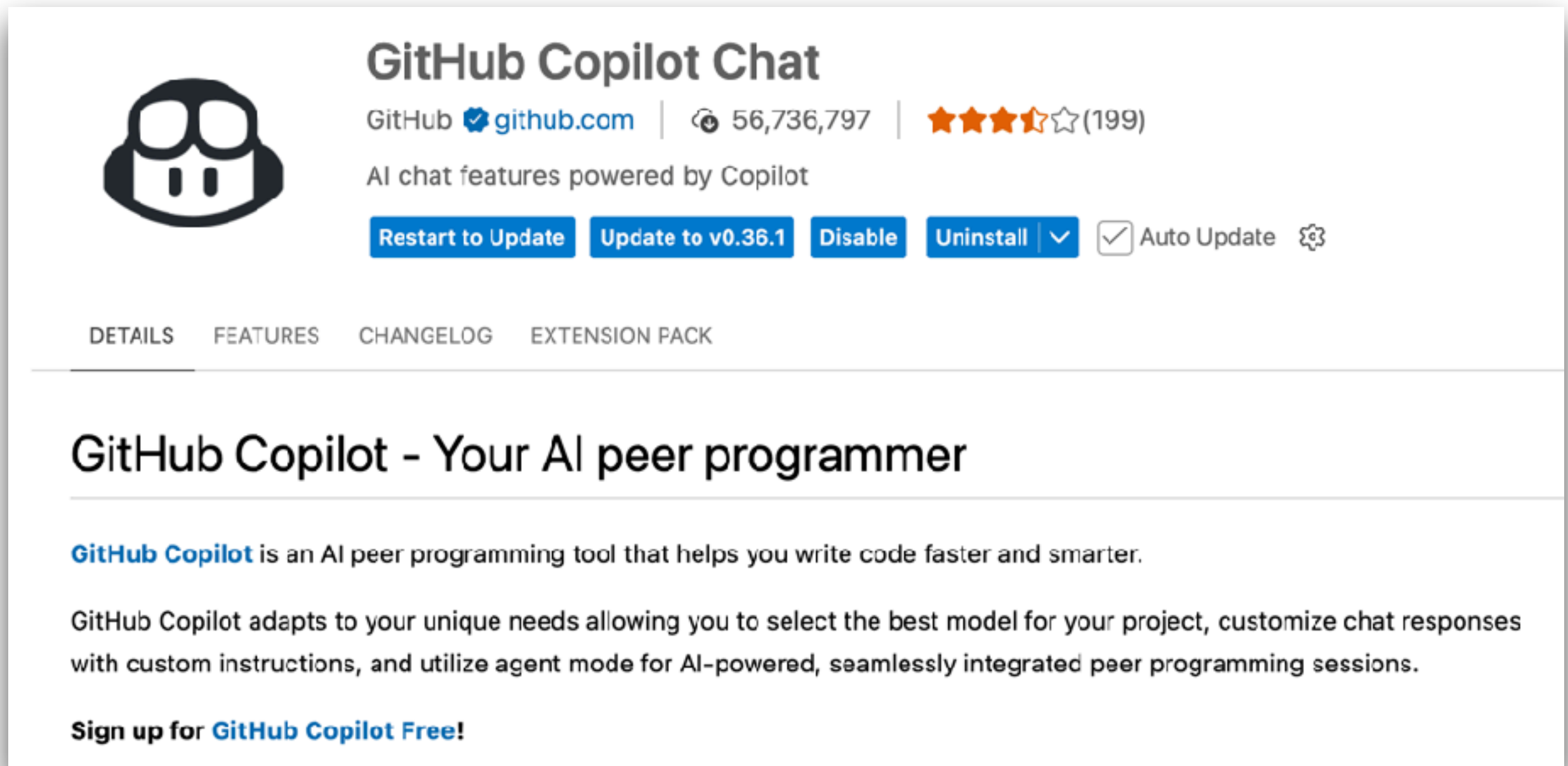


# Workshop with VS Code + GitHub Copilot Chat




<https://code.visualstudio.com/docs/copilot/chat/getting-started-chat>





# Install extension in VS Code



The image shows the GitHub Copilot Chat extension page in the Visual Studio Code marketplace. At the top, there's a header with the extension's name 'GitHub Copilot Chat', its publisher 'GitHub' with a verified badge and 'github.com', a download count of '56,736,797', and a star rating of '4.5 (199)'. Below this, a description states 'AI chat features powered by Copilot'. A row of buttons includes 'Restart to Update', 'Update to v0.36.1', 'Disable', 'Uninstall', and a dropdown menu. To the right of these buttons is a checked 'Auto Update' checkbox and a settings gear icon. A navigation bar below the buttons contains links for 'DETAILS', 'FEATURES', 'CHANGELOG', and 'EXTENSION PACK'. The main content area has a sub-header 'GitHub Copilot - Your AI peer programmer', followed by a paragraph describing it as an AI peer programming tool. Another paragraph explains how it adapts to user needs. At the bottom, there's a call to action to 'Sign up for GitHub Copilot Free!'.

**GitHub Copilot Chat**  
GitHub  [github.com](https://github.com) |  56,736,797 |  (199)  
AI chat features powered by Copilot

[Restart to Update](#) [Update to v0.36.1](#) [Disable](#) [Uninstall](#)  ☒ Auto Update 

[DETAILS](#) [FEATURES](#) [CHANGELOG](#) [EXTENSION PACK](#)

## GitHub Copilot - Your AI peer programmer

**GitHub Copilot** is an AI peer programming tool that helps you write code faster and smarter.

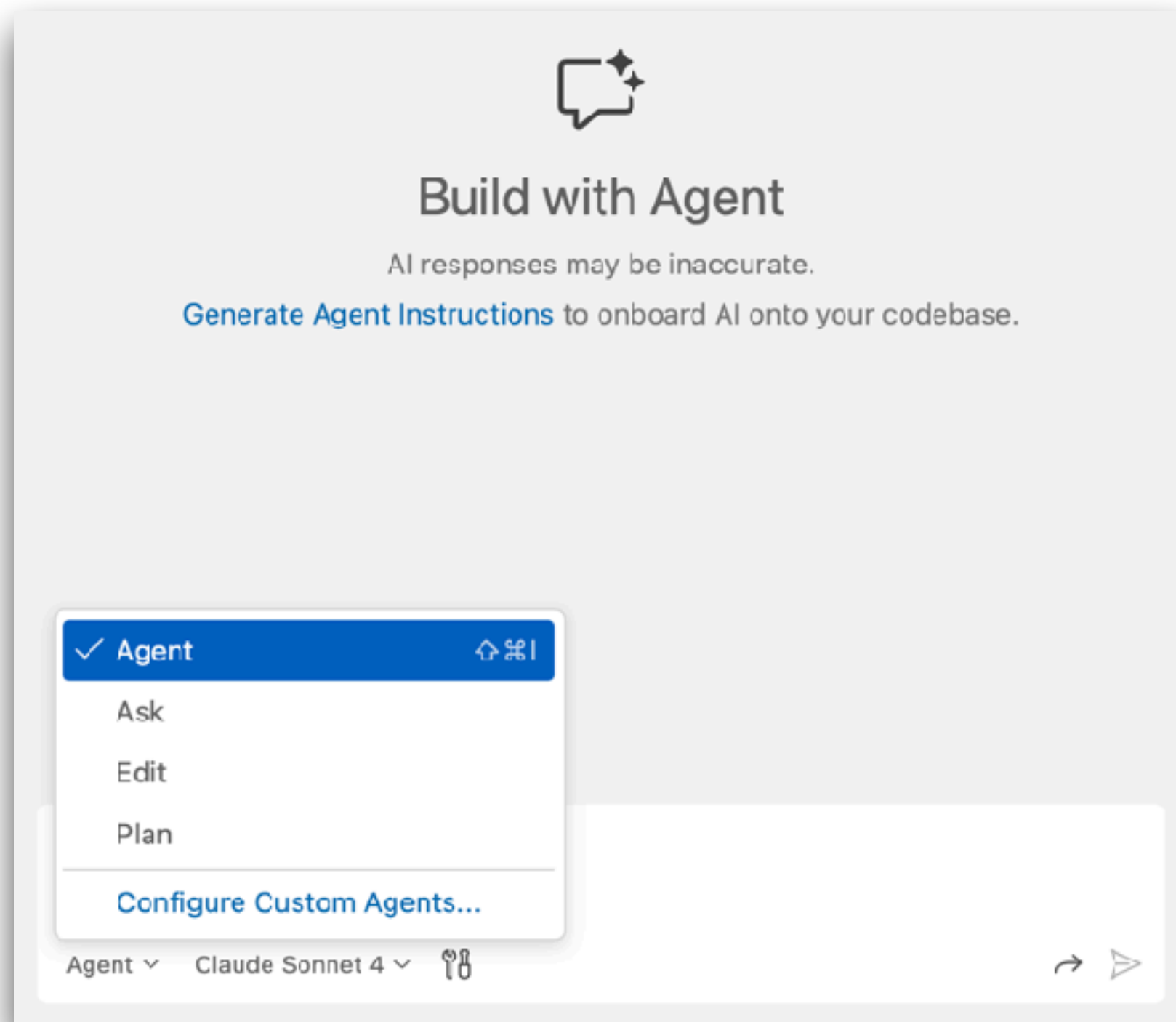
GitHub Copilot adapts to your unique needs allowing you to select the best model for your project, customize chat responses with custom instructions, and utilize agent mode for AI-powered, seamlessly integrated peer programming sessions.

Sign up for [GitHub Copilot Free!](#)

<https://marketplace.visualstudio.com/items?itemName=GitHub.copilot-chat>



# Hello Copilot Chat in VS Code



# Copilot Chat Modes

**Agent**

Ask

Edit

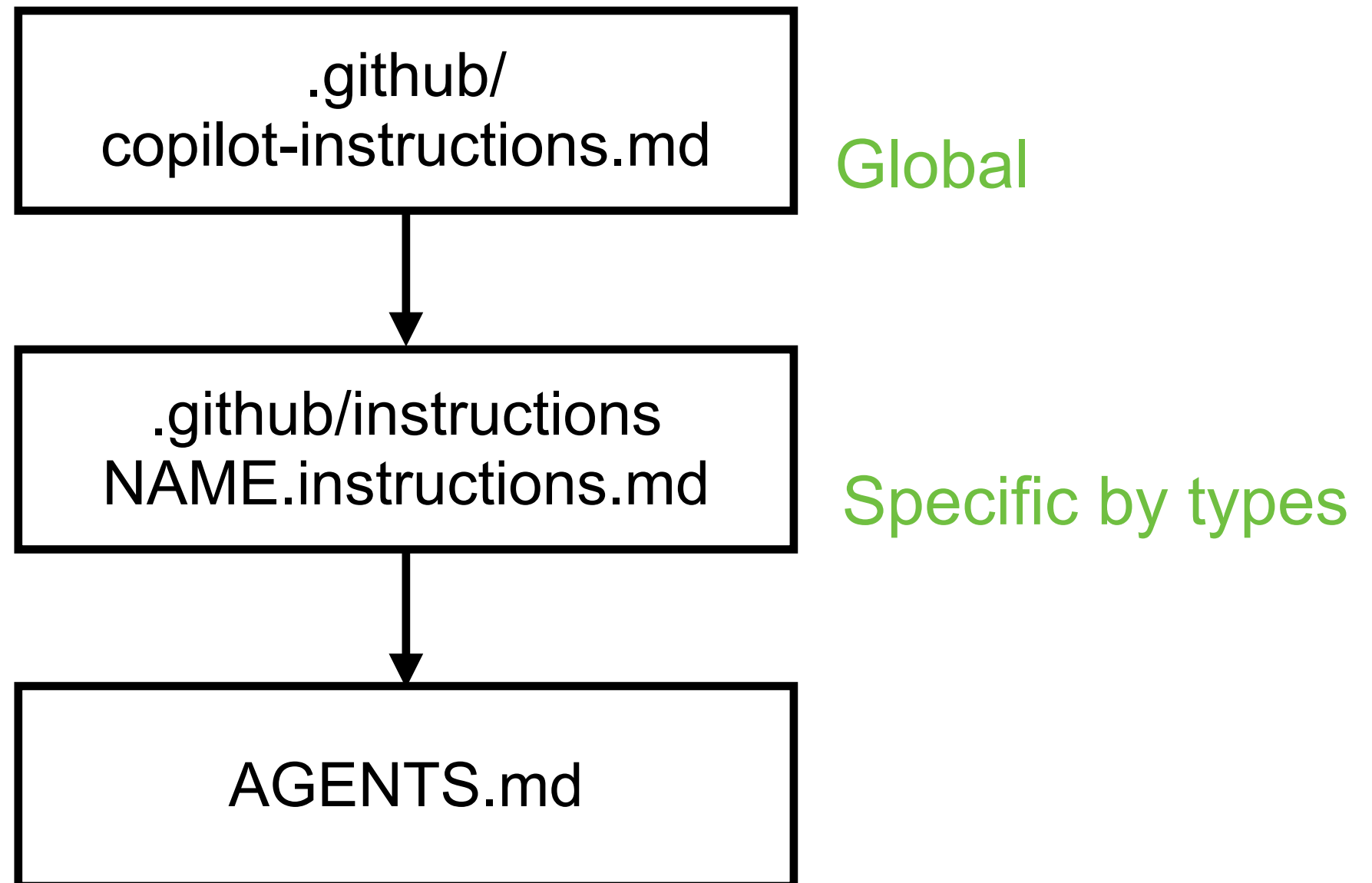
Plan

**Custom  
Agent**

<https://code.visualstudio.com/docs/copilot/customization/custom-agents>



# Custom instructions !!



Disable by default in VS Code

<https://docs.github.com/en/copilot/how-tos/configure-custom-instructions/add-repository-instructions>



# Customization !!

Agent

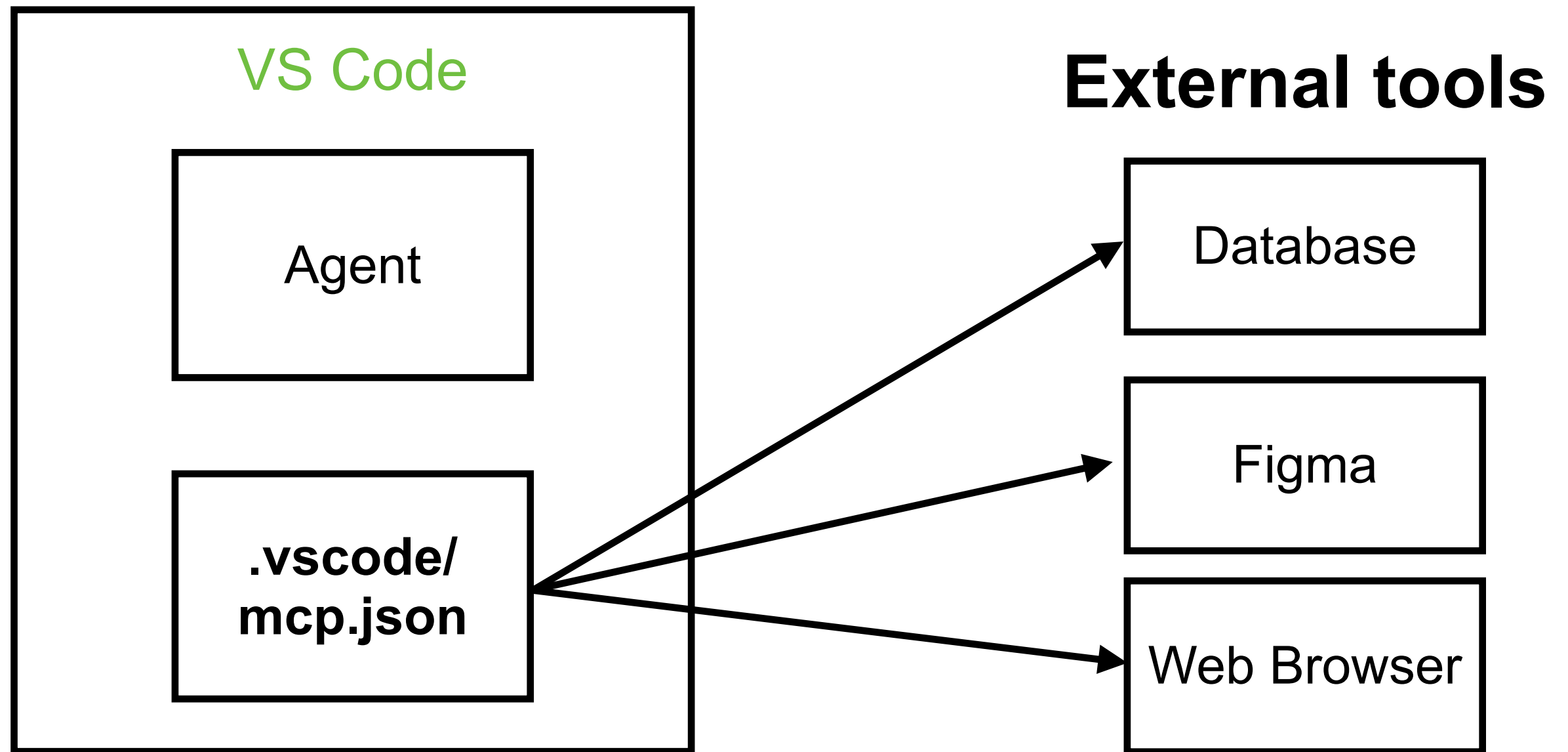
Instruction

Prompts

Skills



# Working with MCP



<https://code.visualstudio.com/docs/copilot/customization/custom-agents>

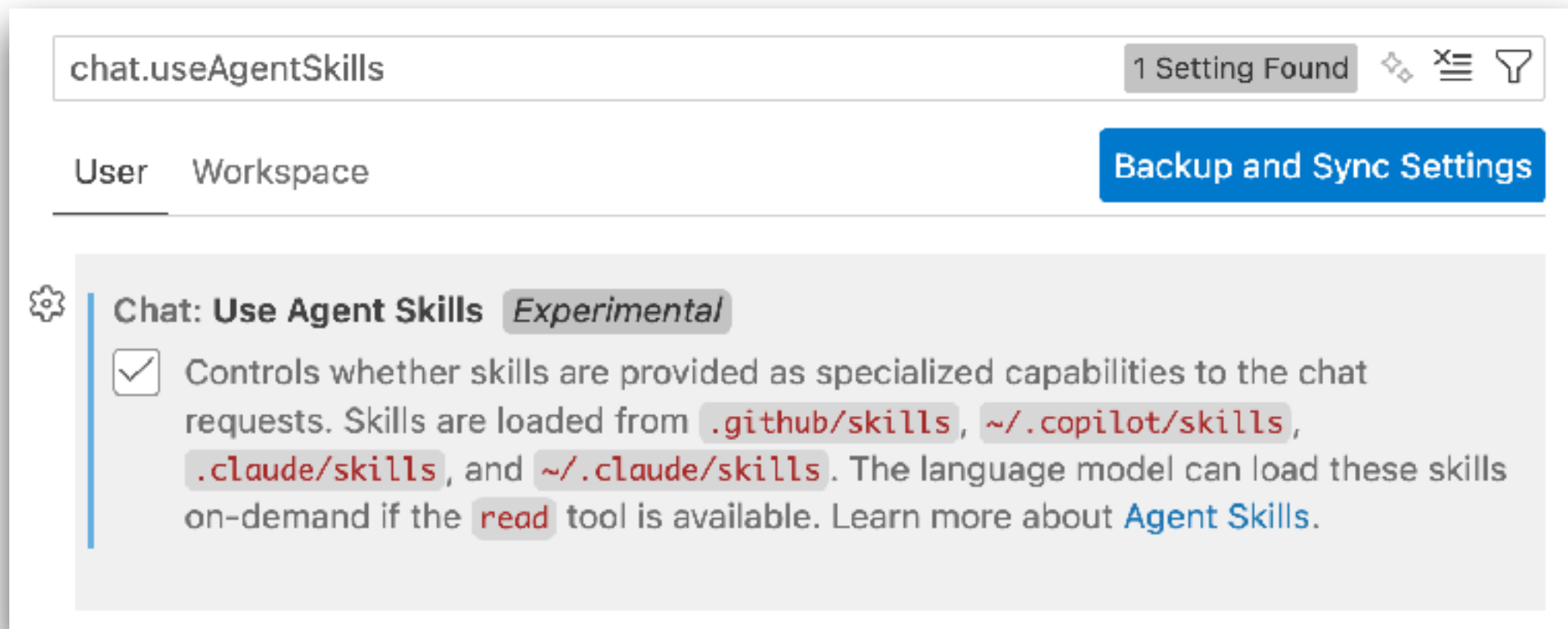




# Enable Agent Skills in VS Code

Setting => “**chat.useAgentSkills**”

Create skills in “.github/skills/name/SKILL.md”



<https://code.visualstudio.com/docs/copilot/customization/agent-skills>



# Workshop with Claude Code

<https://claude.com/product/claude-code>





**npm install -g @anthropic-ai/claude-code**



# Workshop with VS Code + Cline

<https://cline.bot/>



# Install extension in VS Code



## Cline

Cline  [cline.bot](#) |  2,902,099 |  (264)

Autonomous coding agent right in your IDE, capable of creating/editing files, running commands,

[Disable](#) [Uninstall](#)  ☒ Auto Update 

[DETAILS](#) [FEATURES](#) [CHANGELOG](#)

Meet Cline, an AI assistant that can use your **CLI** and **Editor**.

Thanks to [Claude Sonnet's agentic coding capabilities](#), Cline can handle complex software development tasks step-by-step. With tools that let him create & edit files, explore large projects, use the browser, and execute terminal commands (after you grant permission), he can assist you in ways that go beyond code completion or tech support. Cline can even use the Model Context Protocol (MCP) to create new tools and extend his own capabilities. While autonomous AI scripts traditionally run in sandboxed environments, this extension provides a human-in-the-loop GUI to approve every file change and terminal command, providing a safe and accessible way to explore the potential of agentic AI.

<https://marketplace.visualstudio.com/items?itemName=saoudrizwan.claude-dev>



# Q/A

