DataCamp

Building a Multi-Agent MCP Application: Promotion Tycoon

Mikiko Chandrasekhar
Staff AI/ML Developer Advocate
@MongoDB

Intro & Setup

Agenda

- Welcome & Setup
- What We're Building – **Promotion Tycoon**
- What We're Using – **Tech Stack**
- Why – **Explaining Technology Features**
- Code Walkthrough – **Plus Optimization Opportunities**
- Wrap-Up / Q&A

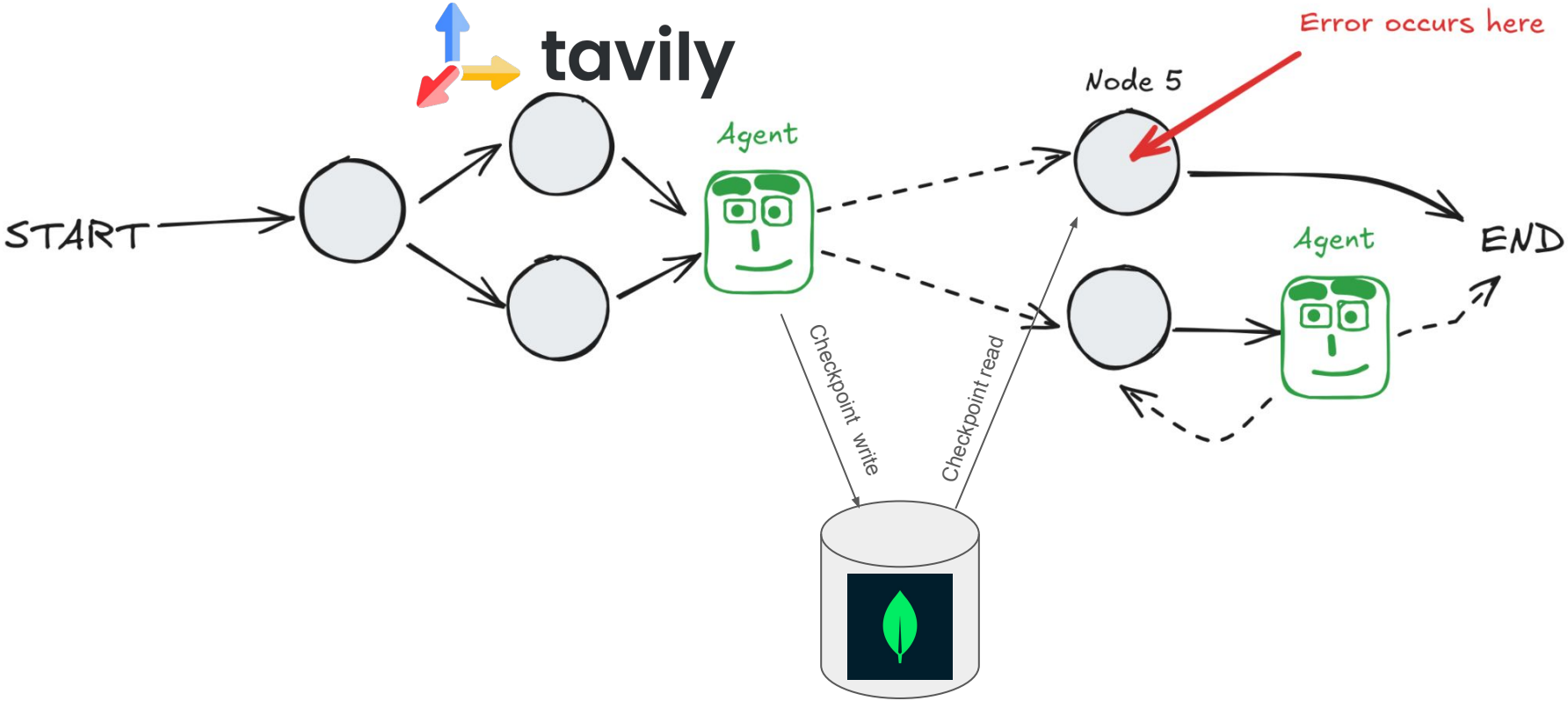
whoami

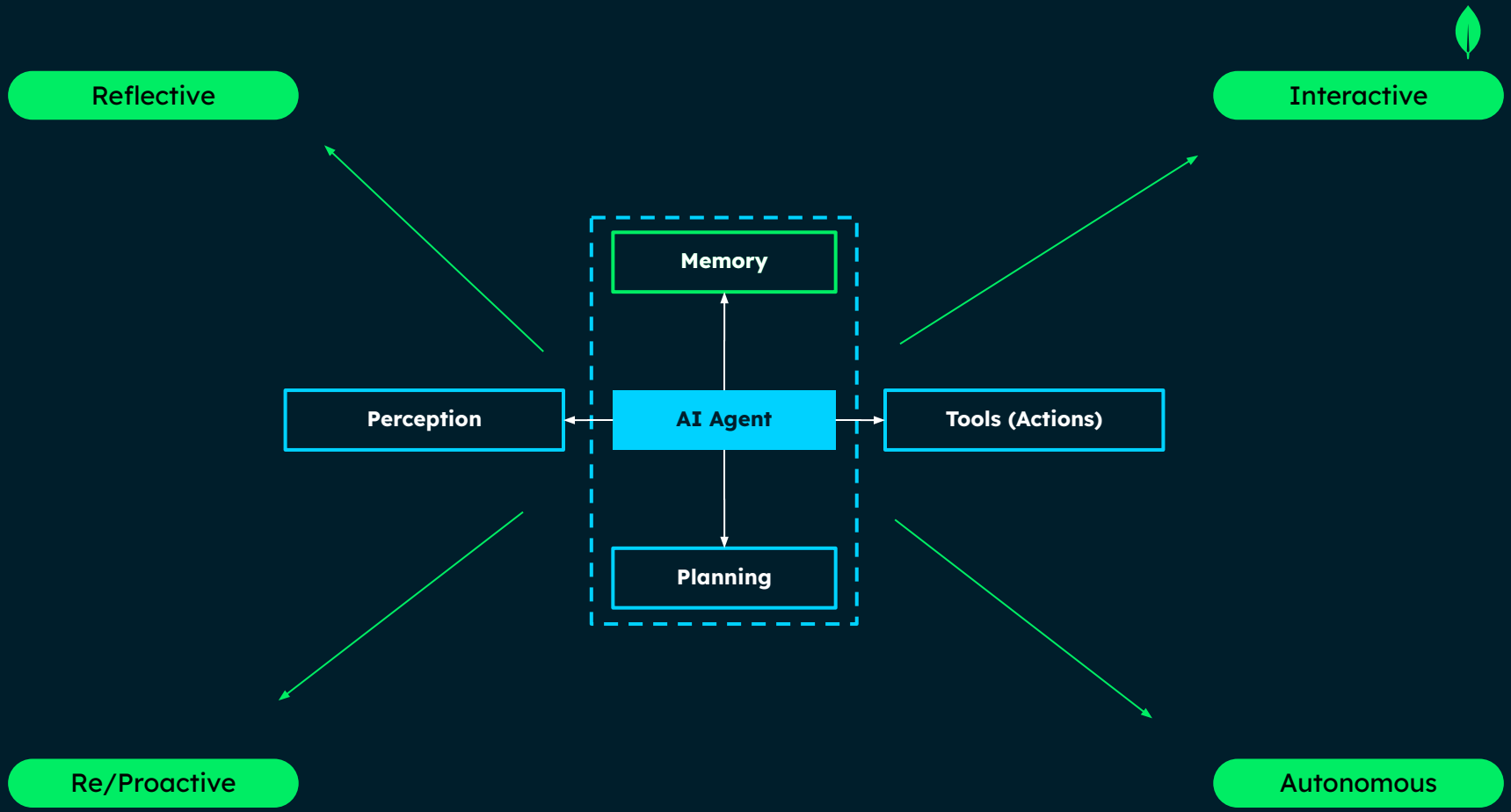
```
{
  "name": "Mikiko Bazely",
  "MongoDB":
    "positions": ["AI Developer Advocate"]
    "since": new Date("2025-04")

  "Pre-MongoDB":
    "positions": ["DevRel"]
    "companies": ["Fireworks", "Labelbox", “Mailchimp”]

  "likes": ["big arrays and I cannot lie"]
}
```

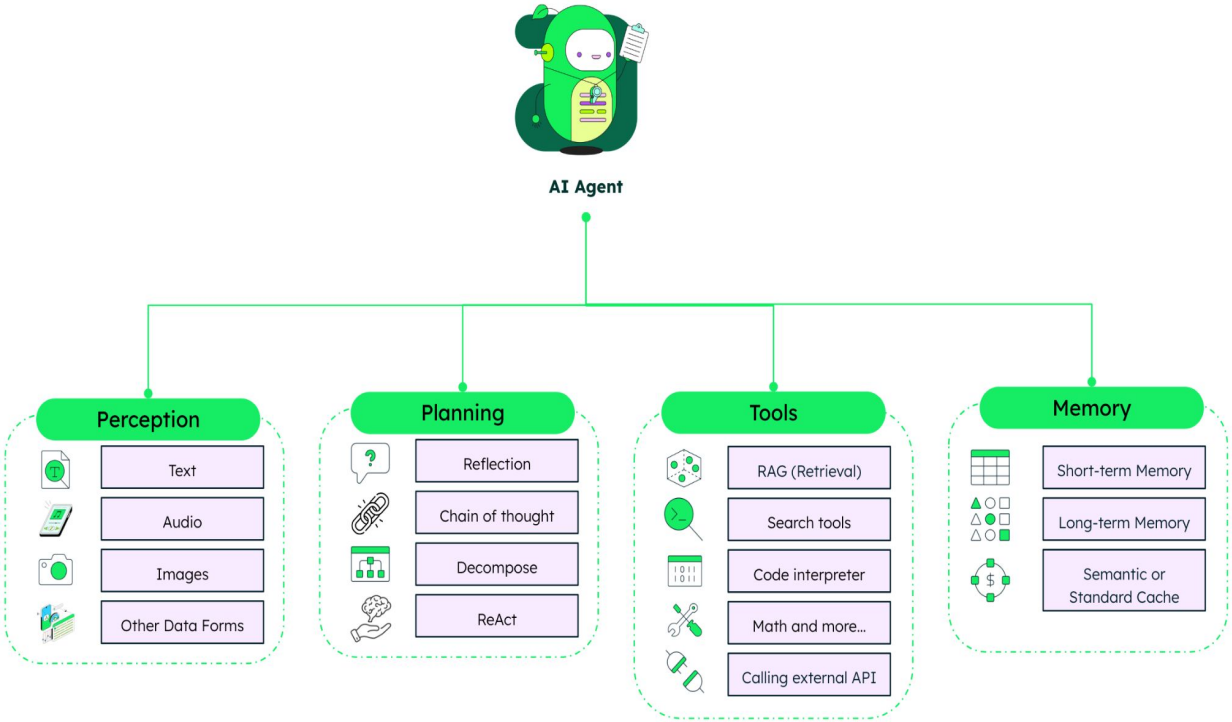
What We're Building: AI Agents

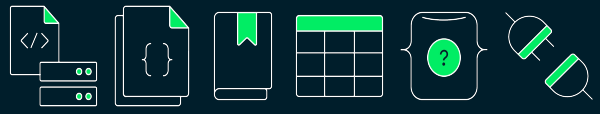

tavily
START
Agent
Node 5
Error occurs here
Agent
END
Checkpoint write
Checkpoint read


Reflective
Interactive
Memory
Perception
AI Agent
Tools (Actions)
Planning
Re/Proactive
Autonomous

Agentic AI

- AI agents that can operate **autonomously** and take actions based on goals and real-world interactions
    - Fully autonomous
    - Semi-autonomous: human-in-the-loop (HITL)
- Equipped with advanced tools, models, memory, and data storage.
    - **Memory** → Informed decision-making and continuity.
    - **Tools** and **models** → to decompose tasks into steps and execute them cohesively.





Data Sources: (Databases, Data Warehouse, Repositories, PDFs, Webpages, APIs)

Raw data

Data Preparation
(Preprocessing, splitting and chunking)

Chunks

Metadata

Embedding Generation
(Vector representation of data objects)

Embedding + Metadata

Operational and Vector Database

Retriever As A Tool

Tools

Large Language Model

Task Decomposition

Reasoning and Planning

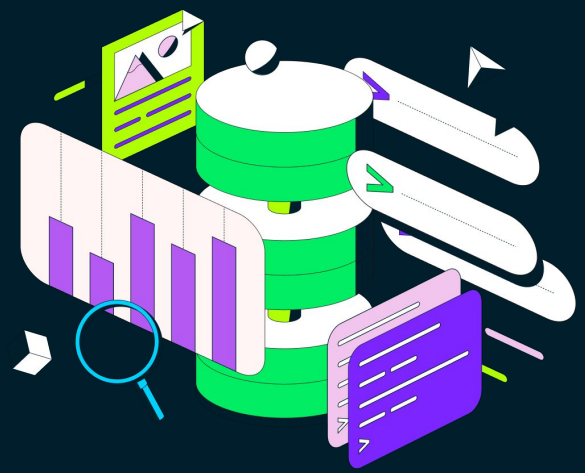
Iterative execution of operations, using tools and synthesizing responses based on previous outputs. Termination occurs upon meeting predefined criteria, including objective achievement or reaching set limits.

Provide an objective as a prompt

Application User

Agentic System

AI STACK

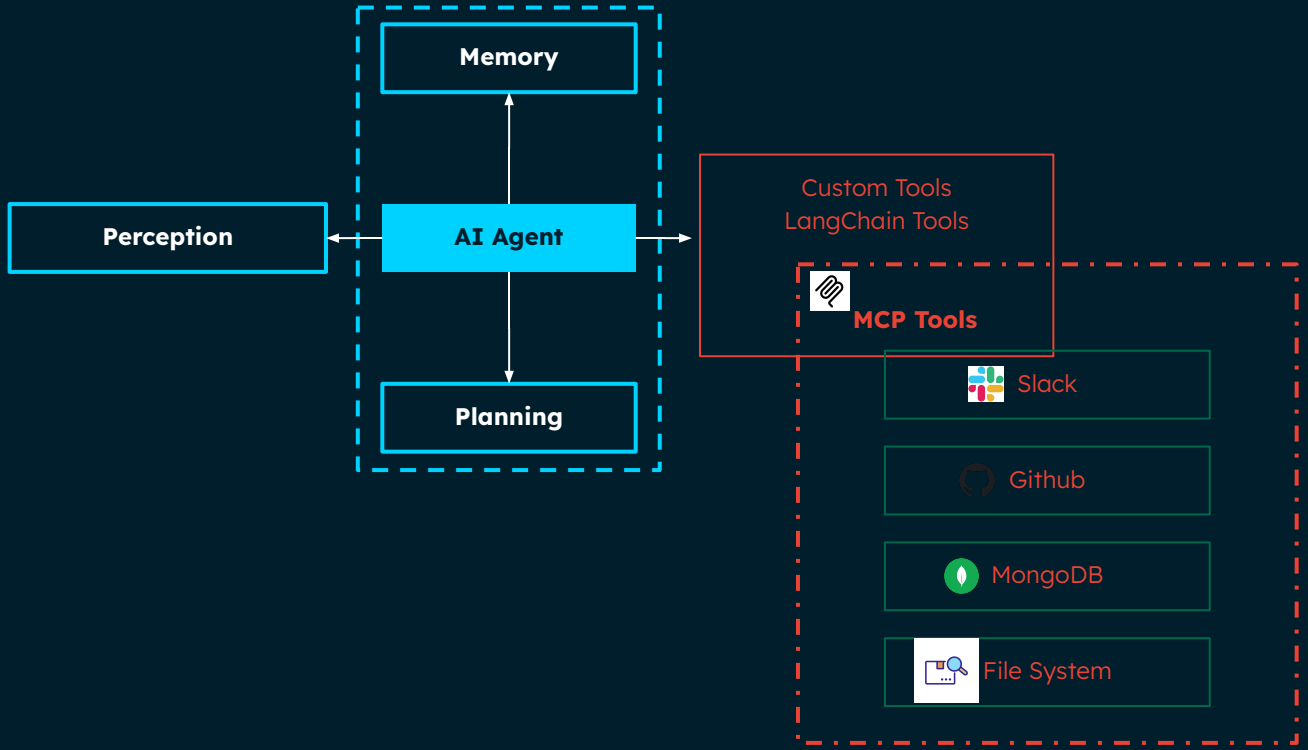


Low Level View

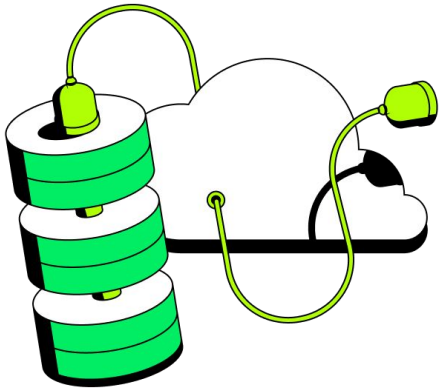
- Programming Language
- Model Provider
- LLM Orchestrators and Frameworks
- **Operational and Vector Database**
- Monitoring and Observability
- Deployment

Read: https://mdb.link/understanding_ai_stack

MCP


Memory
Perception
AI Agent
Planning
Custom Tools
LangChain Tools
MCP Tools
Slack
Github
MongoDB
File System

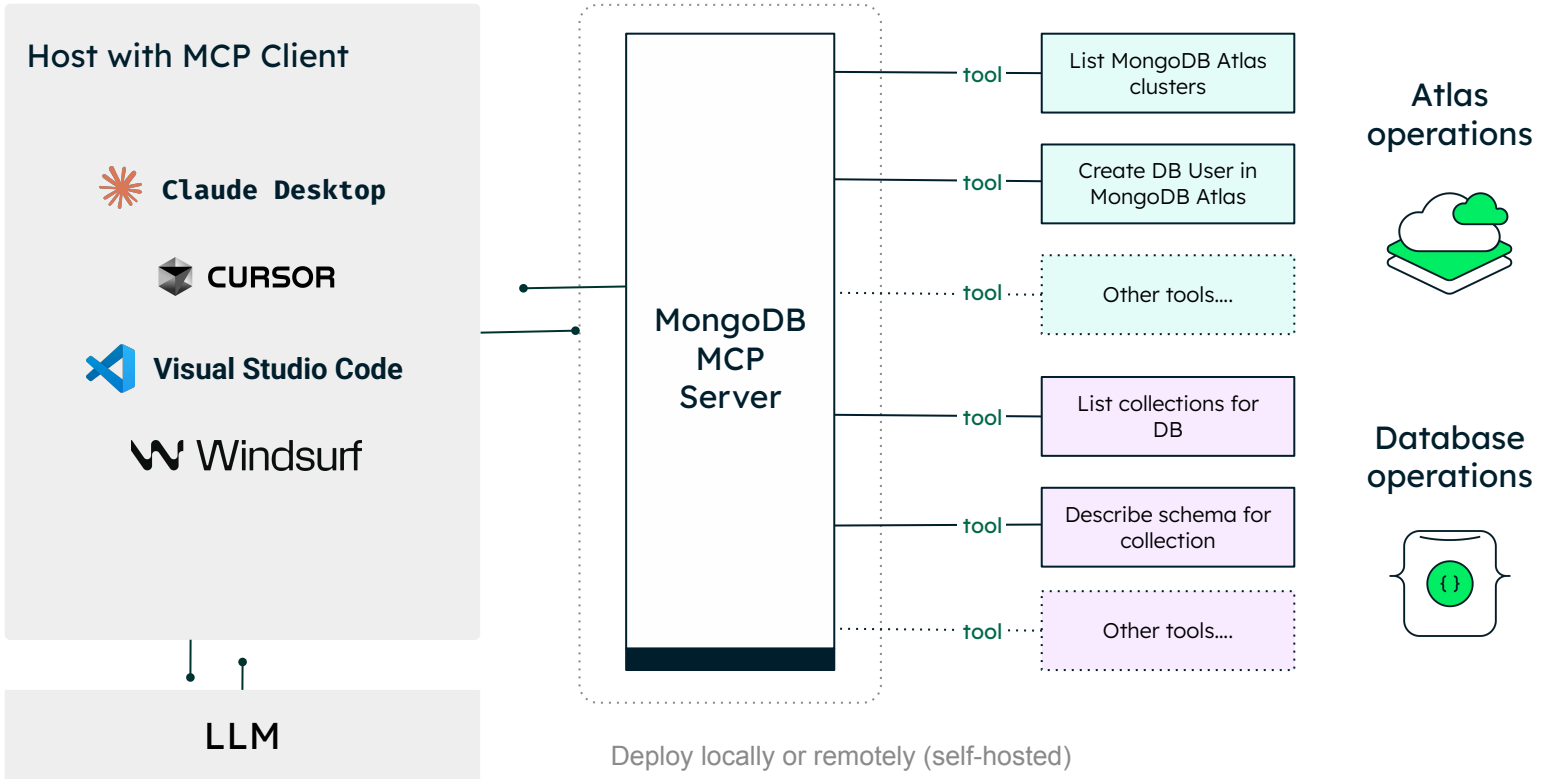
What is the MongoDB MCP Server?



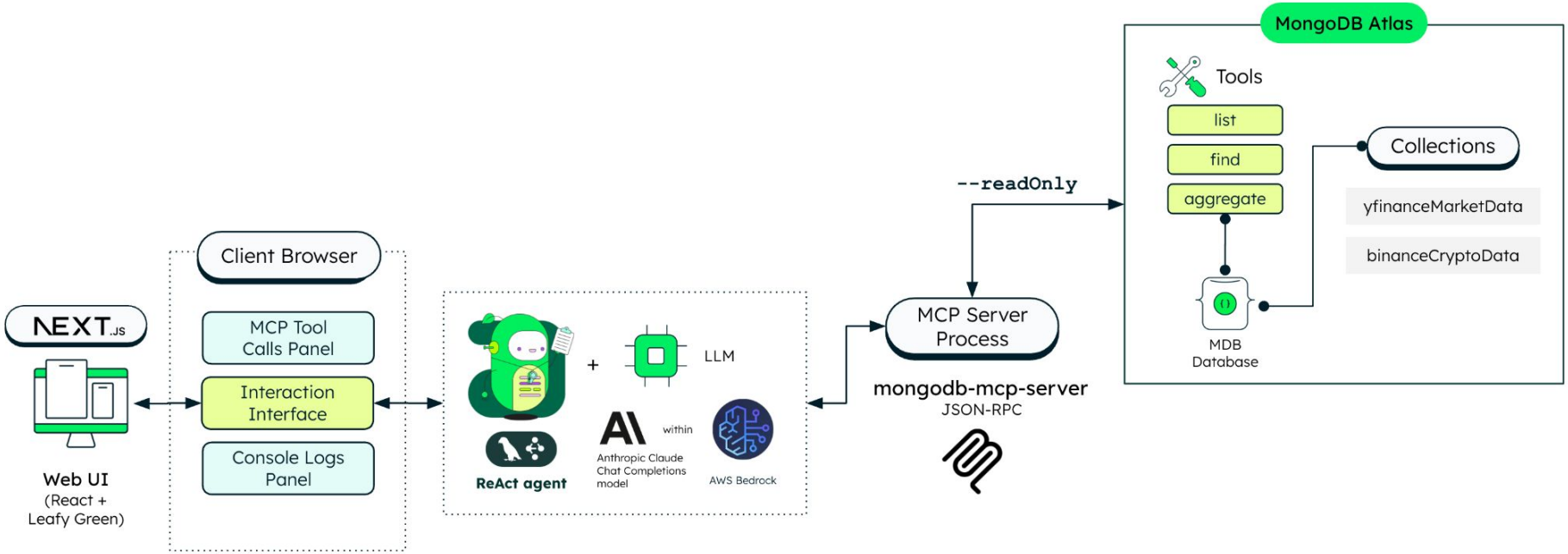
The **MongoDB MCP Server connects MongoDB deployments** to **AI-powered developer tools** like Windsurf, Cursor, GitHub Copilot, Claude Code or any MCP-enabled agentic system.

It brings valuable context from MongoDB databases into developers' environments, enabling **data exploration, database operations, and better, context-aware code generation** via natural language.

MongoDB MCP Server



MongoDB MCP Server



MongoDB MCP Server: Use Cases

AI-assisted software development
(agentic coding)

The MongoDB MCP Server lets developers explore and manage databases in their environment while enabling coding agents to query data, iterate faster, and generate context-aware code for more reliable feature development with AI.

Primary use case

Power agentic systems

Integrate the MongoDB MCP Server into agentic systems to let them explore and query MongoDB deployments seamlessly. It works as a plug-and-play plugin for any MCP-enabled agent.

Secondary use case

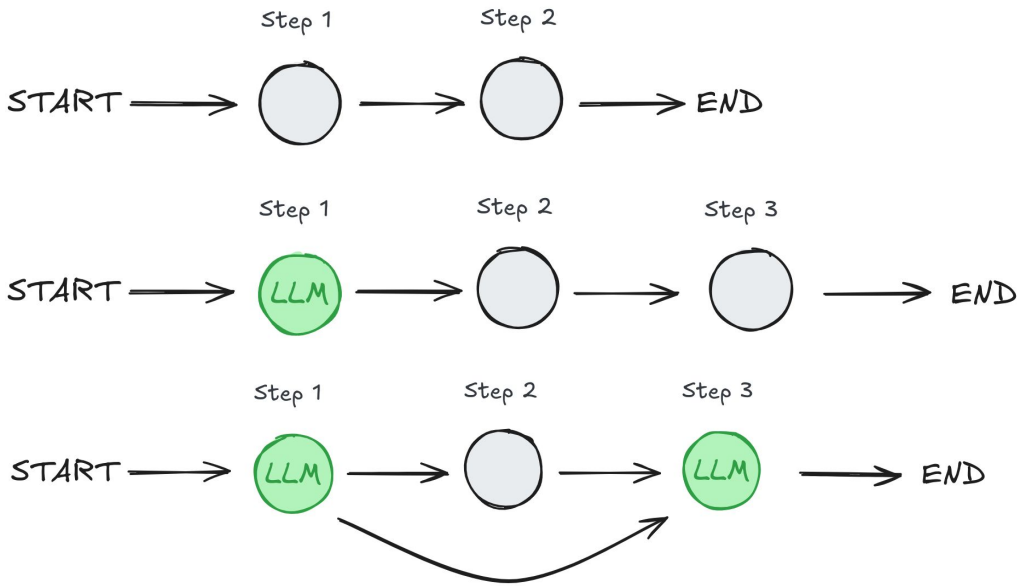
The Tech Stack

Orchestration: Langgraph

Building AI Systems

LangChain

Design Paradigm #1: “chains” aka a predetermined control flow

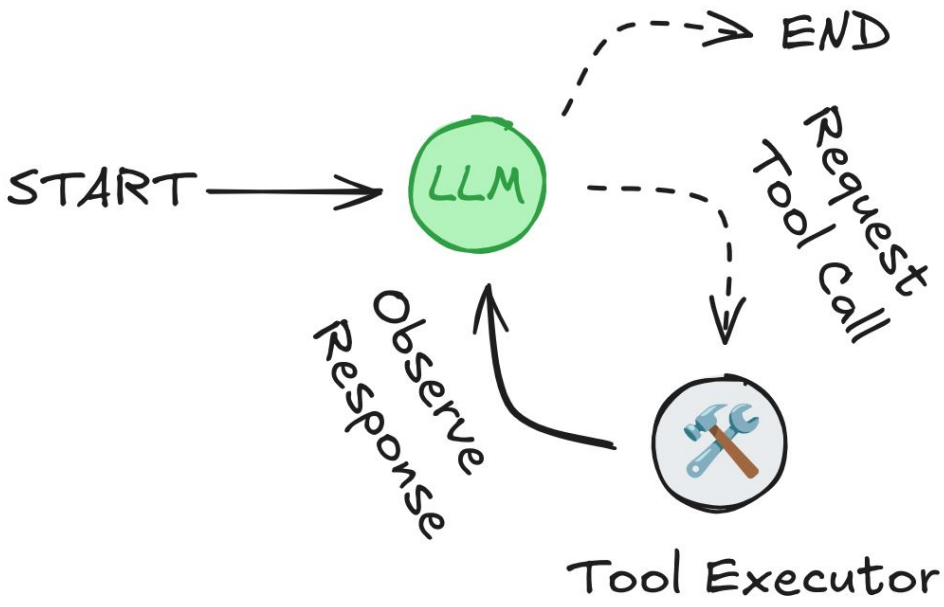


Reference: https://en.wikipedia.org/wiki/Control_flow

Building AI Systems

LangChain

Design Paradigm #2: an LLM running in a loop aka ReAct Agent...

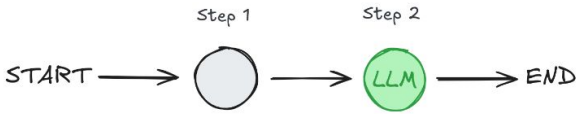


Building AI Systems

LangChain

Can we have both?

A chain with predetermined flow

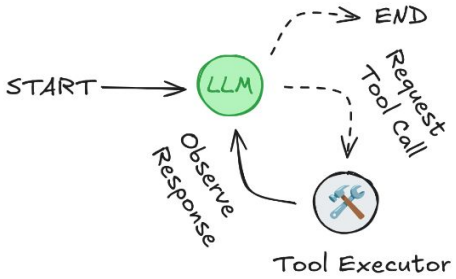


Not Flexible
More reliable

?

Flexible
Reliable

The Famous React Agent



Flexible
Less reliable

What is LangGraph?

LangChain

LangGraph applications balance agent control with agency

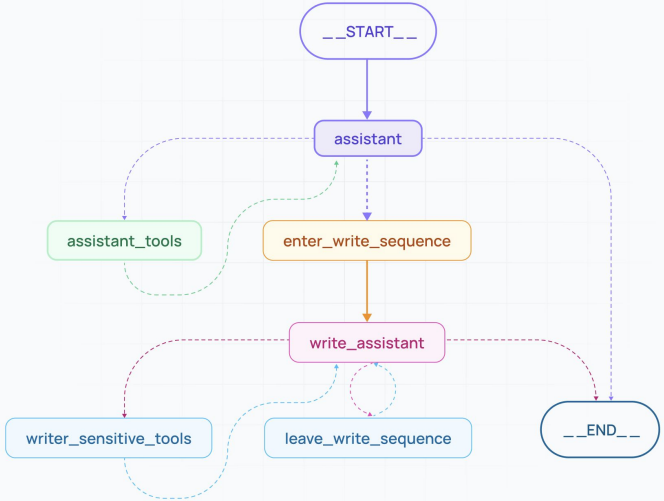
Its core pillars support:

1. **Controllability**: to define both explicit and implicit workflows
2. **Streaming**: to expose any event (or token) as it occurs
3. **Persistence**: to enable memory, human-agent interactions, multi-agent interactions & fault tolerance
4. **Human-in-the-loop:** to facilitate human guidance

LangGraph also:

Works with or without LangChain

Integrates with LangSmith **(Observability & Evaluations)**

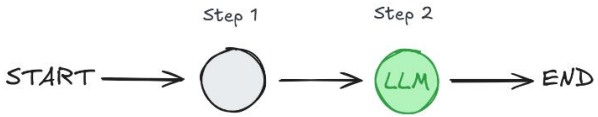


Building AI Systems

LangChain

LangGraph allows for developer + LLM-defined control flows

A chain with predetermined flow

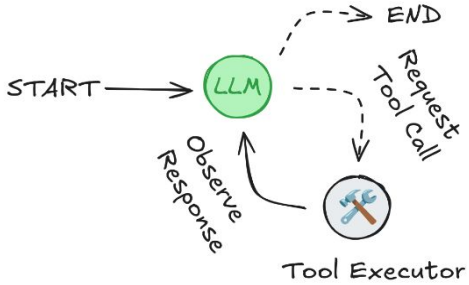


Not Flexible
More reliable

LangGraph

Flexible
Reliable

The Famous React Agent

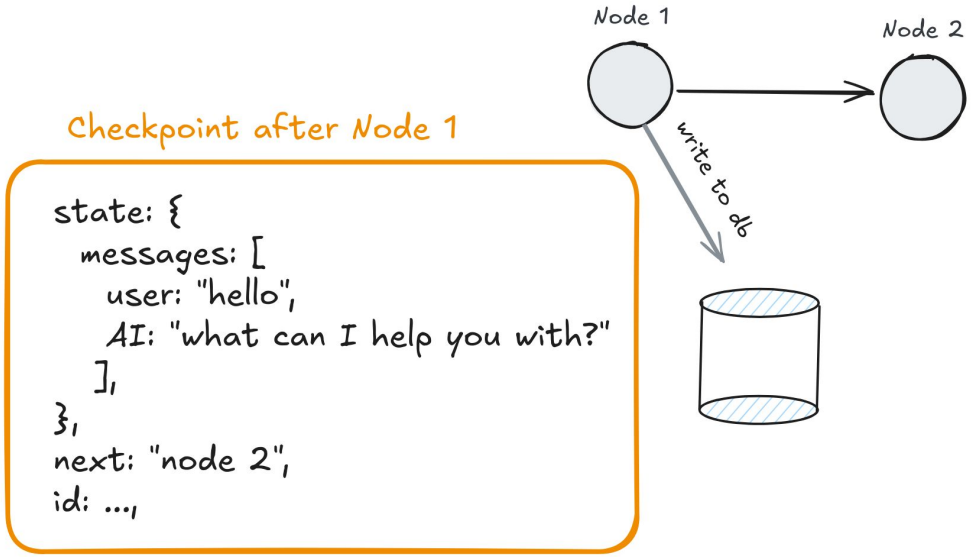


Flexible
Less reliable

Persistence

LangChain

Built-in persistence

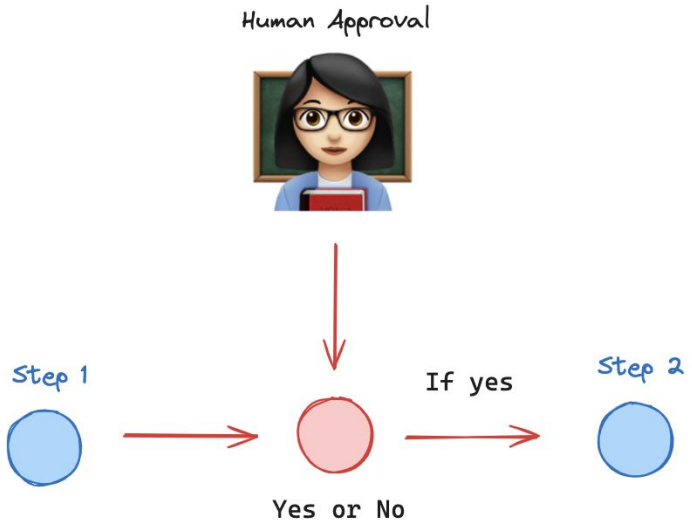


Reference: https://langchain-ai.github.io/langgraph/how-tos/persistence/

Human-in-the-loop

LangChain

Supports different human-in-the-loop interactions: approval, editing, input



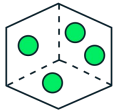
Reference: https://langchain-ai.github.io/langgraph/how-tos/human_in_the_loop/breakpoints/

Persistence: MongoDB

MongoDB®

Application Database

MongoDB Atlas, MongoDB Charts, Triggers, Stream Processing



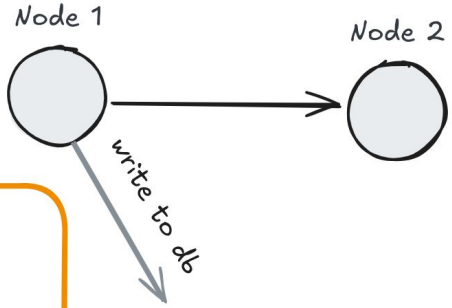
Vector Store + Search

Atlas Vector Search, Vector Database for RAG applications

Memory Store

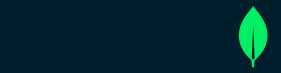
Memory Provider for AI Agents and Agentic Systems

Built-in Persistence



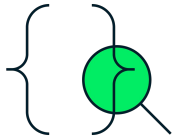
Checkpoint after Node 1

```
state: {
  messages: [
    user: "hello",
    AI: "what can I help you with?"
   ],
},
next: "node 2",
id: ...,
```



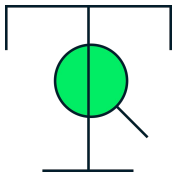
There are different ways to retrieve data

Searching for data with MongoDB



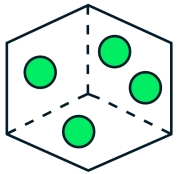
.find()

Basic query capabilities; great for retrieving a single document, or known criterias.



Full-text search

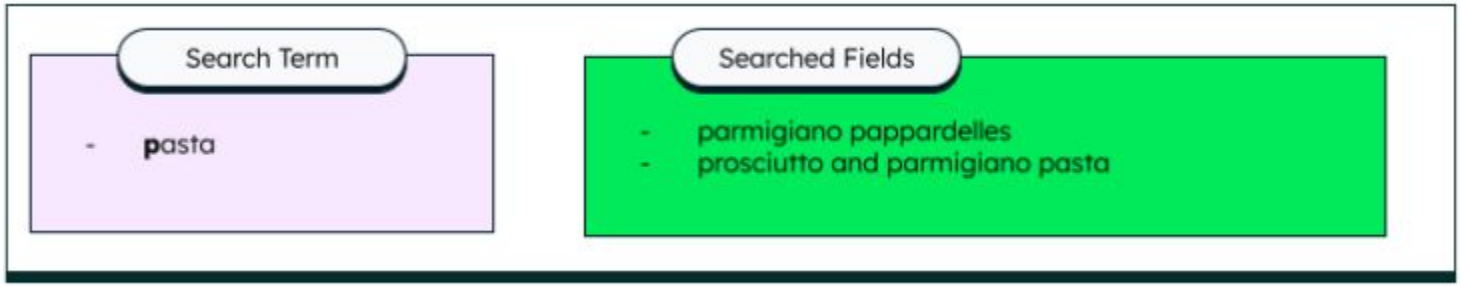
Great to find words that aren’t grouped; easily add fuzzy matching or autocomplete



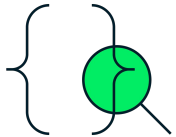
Vector search

Search based on meaning; great to retrieve related content or to search for similar topics

```
db.products.find( { name: { $regex: /pasta$/ } } )
```

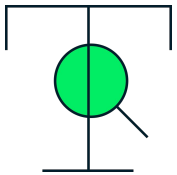


Searching for data with MongoDB



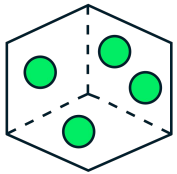
.find()

Basic query capabilities; great for retrieving a single document, or known criterias.



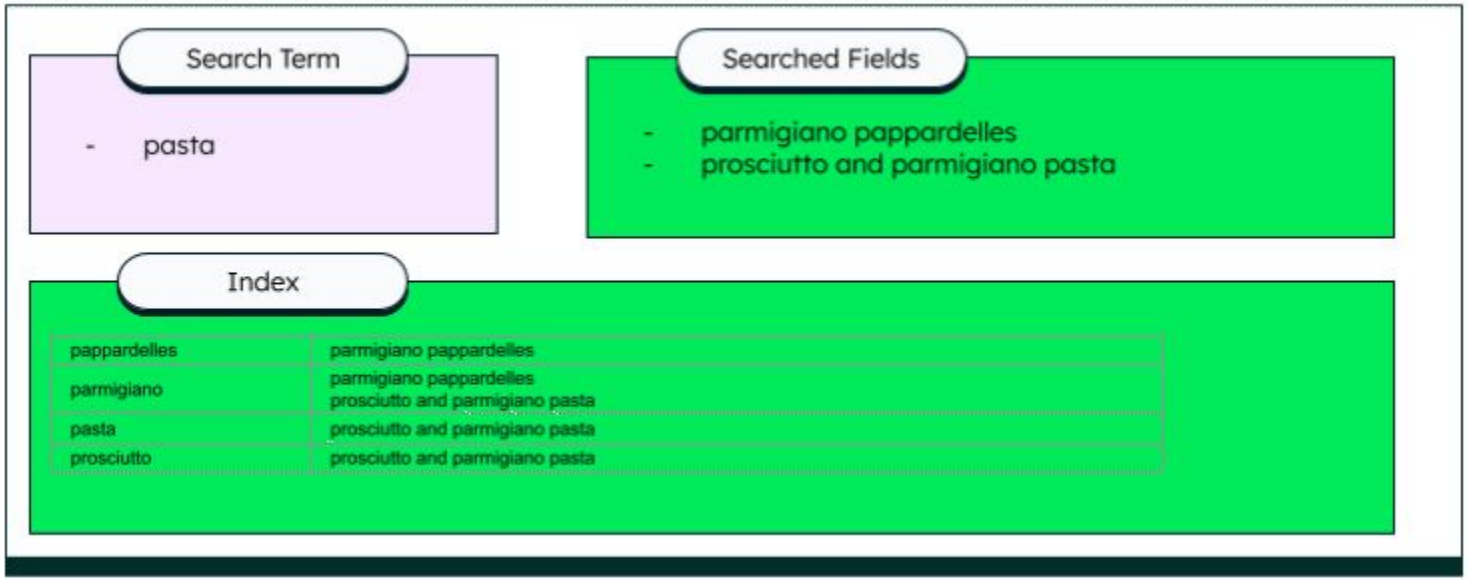
Full-text search

Great to find words that aren’t grouped; easily add fuzzy matching or autocomplete



Vector search

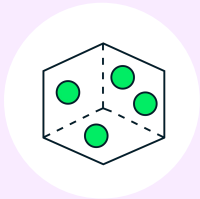
Search based on meaning; great to retrieve related content or to search for similar topics


Search Term
- pasta
Searched Fields
- parmigiano pappardelles
- prosciutto and parmigiano pasta
Index
pappardelles
parmigiano pappardelles
parmigiano
parmigiano pappardelles
prosciutto and parmigiano pasta
pasta
prosciutto and parmigiano pasta
prosciutto
prosciutto and parmigiano pasta

MongoDB Search Capabilities

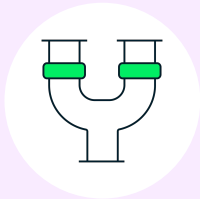
MongoDB Atlas Search

Offers full-text search with features like stemmed words, synonyms, relevance scoring, and fuzzy matching.



Atlas Vector Search

Combines operational and vector data in a unified, fully managed platform, eliminating the need for separate databases



MongoDB Query Language

MQL enables hybrid search by combining both full-text search and vector search within a single query.

📚 Concept: Memory, A Component of Agents

Agent Memory

What is AI Agent Memory?

Agent Memory is **persistent memory management** system that transforms stateless AI Agent into an intelligent entity capable of **learning, adapting**, and maintaining **continuity** across interactions.

Memory

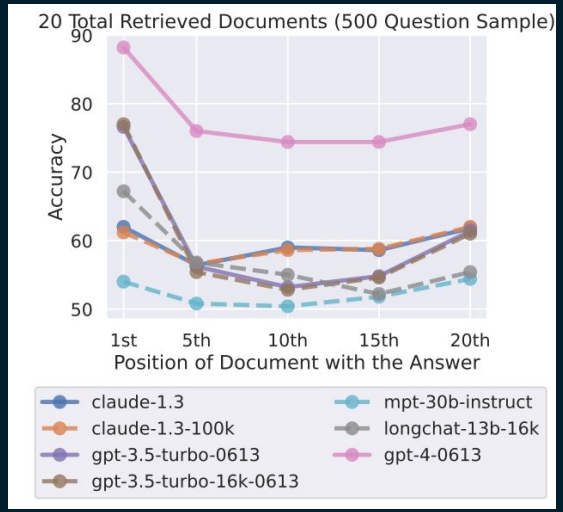
Enables agents to remember, reason, and learn from past interactions

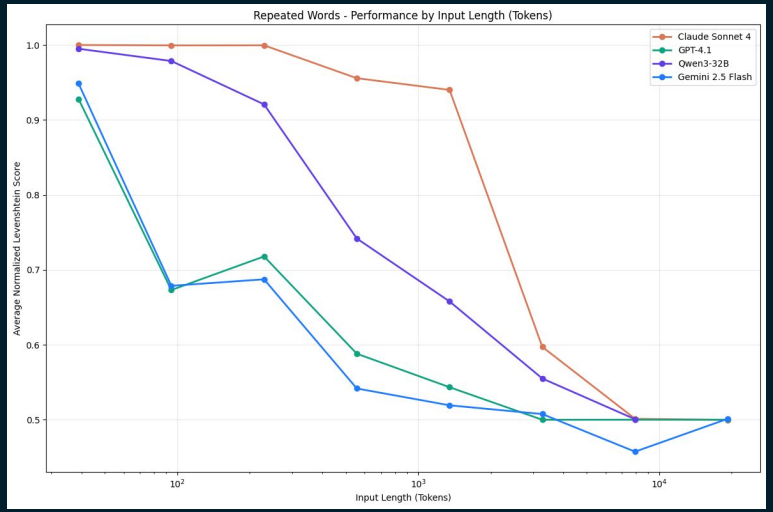
Memory

Short-term (Insights from a specific conversation)

Long-term (Insights from multiple conversations over a period of time)

Diminished performance as tokens increase





Source: Stanford, Lost in the Middle

Source: Chroma, Context Rot

MongoDB as a data foundation simplifies memory for AI agents

Modes of memory

Conversational, assistant, workflow

Tooling to enable business logic

LangGraph

MCP

AutoGen

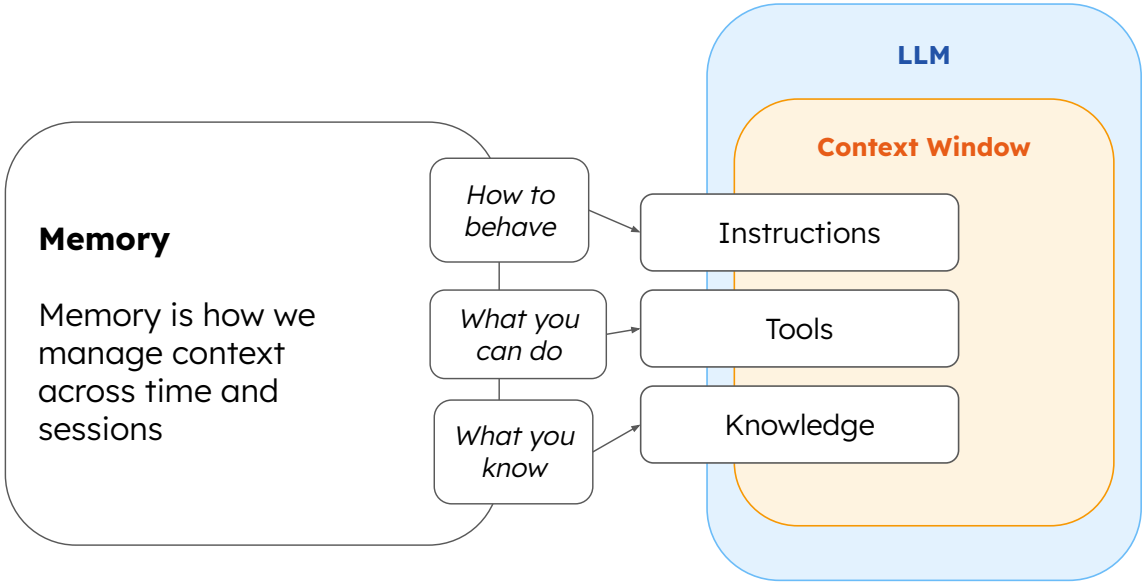
And more...

MongoDB.

Multi-Cloud Scale, Resilience, Performance, & Security
supporting short-term and long-term memory, and state persistence on multimodal data

Memory Engineering and Context Engineering

Memory engineering and context engineering work hand-in-hand, where memory engineering builds the persistent, intelligent storage systems that context engineering then leverages to dynamically curate the most relevant information for each AI decision.



Tools: Tavily

Designing Multi-Agent Workflows & Systems

The Agentic Spectrum

Level 1

Minimal Agent

LLM equipped with an instructional or system prompt that continuously operates in a loop until the desired outcome is achieved.

Level 2

Controlled Flows

Systems where LLMs perform tasks within a fixed sequence of steps, maintaining structured decision-making with limited autonomy.

Level 3

Routing and Specialized Workflows

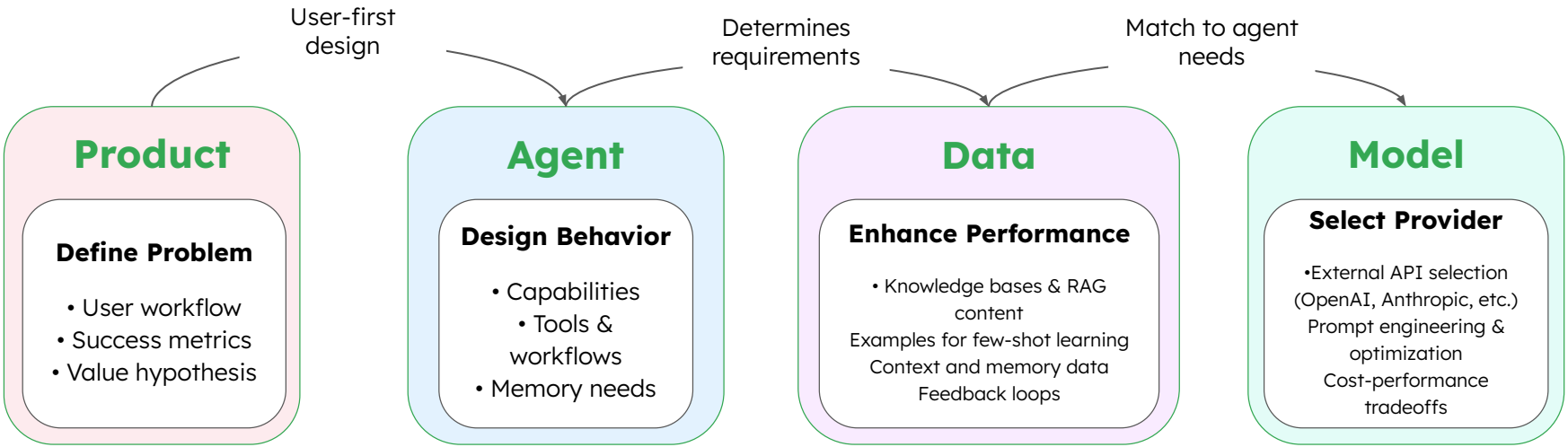
LLMs categorize and route inputs to specialized workflows, demonstrating more nuanced decision-making capabilities.

Level 4

Autonomous Agents

LLMs dynamically determine step sequences, use tools independently, and adapt their approach to complete complex, open-ended tasks with minimal human intervention.

Agentic Flow: Foundation Model Era



New considerations of the PRODUCT → AGENT → DATA → MODEL development paradigm

- Agent layer orchestrates everything
- Tools & workflows before model selection
- Data enhances, doesn't enable

Agentic AI Canvas Framework: From Idea to Production

Moving through the phases from identifying use case to prototyping, shipping, and then operationalizing agentic products.

POC Canvas (1-2 weeks)

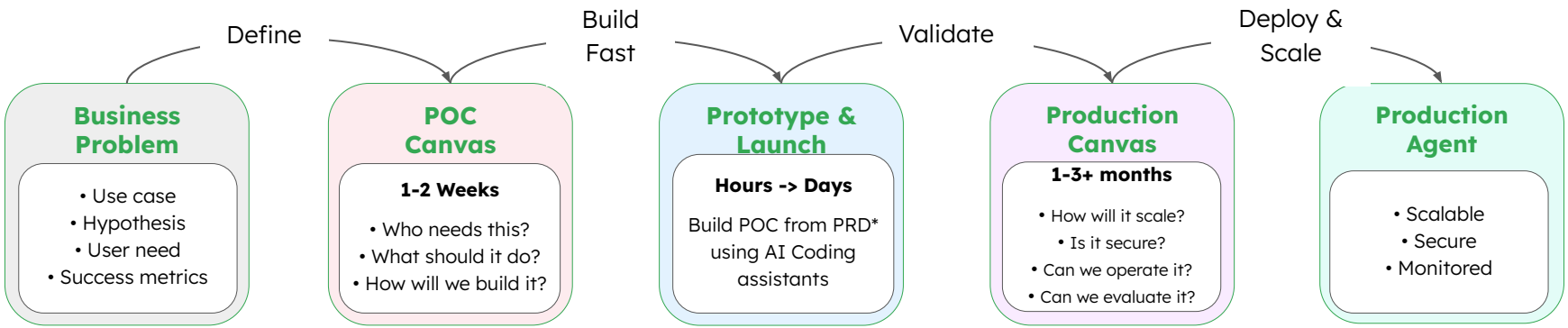
Answer 3 critical questions: Who, what, how. Outputs a clear PRD* ready for rapid prototyping.

Vibe-Coding Phase

Use AI assistants (Cursor, Claude, Copilot) to rapidly build POC from PRD*. Go from structured plan to working prototype in hours/days, not weeks.

Production Canvas (1-3 months)

Answer 3 scaling questions: Scale, security, monitoring. Proper engineering replaces vibe-coding for production systems.



PRD = Product Requirements Document

Agent POC: Canvas V1

The canvas helps teams systematically work through all aspects of an agentic AI project while avoiding redundancy and ensuring nothing critical is missed.

Product Validation	Agent Design	Data Requirements	External Model Integration
1. PRODUCT VISION & USER PROBLEM	**3. AGENT CAPABILITIES & WORKFLOW**	**5. KNOWLEDGE REQUIREMENTS & SOURCES**	**7. PROVIDER SELECTION & PROMPT ENGINEERING**
What user problem are we solving and why it matters	What the agent needs to do to solve the user problem	What information the agent needs to function effectively	Which external models and how to optimize prompts
2. USER VALIDATION & INTERACTION	**4. AGENT INTERACTION & MEMORY**	**6. DATA COLLECTION & ENHANCEMENT STRATEGY**	**8. API INTEGRATION & VALIDATION**
How users will actually engage with the agent	How the agent will communicate and remember	How to gather initial data and improve over time	How to connect to providers and validate performance

Check out our GenAI Examples repo!

mdb.link/genai-examples