levi nine

November 2025 | **Alex Bita**

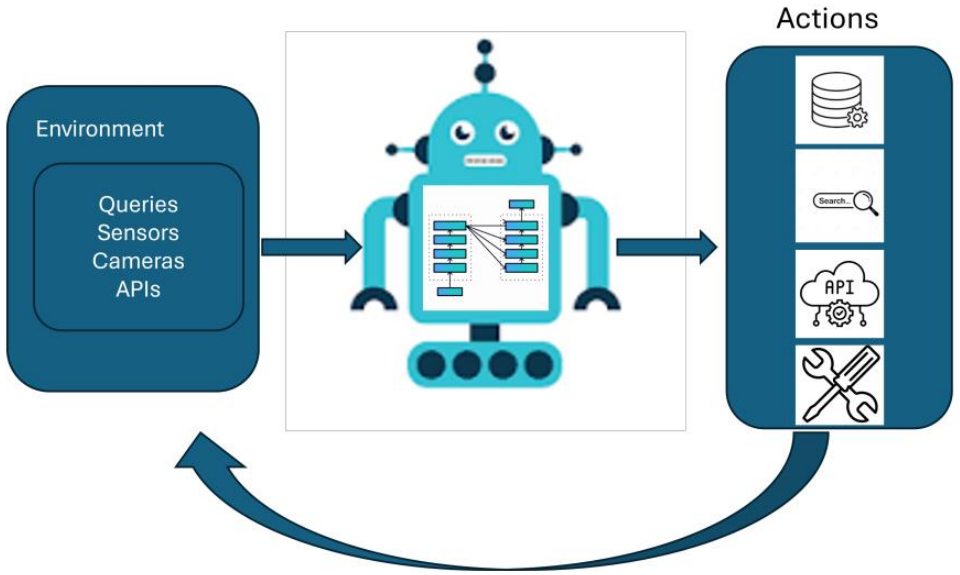
When Natural Laziness meets Artificial Intelligence, or How to Build Agentic Systems

Table of Content

Introduction (1) What is an **agent system**

- **Agent**
  - autonomous entity (powered by **LLM**)
  - **perceive** the environment
  - **act** towards specific goals
- **Agent system**
  - **multiple agents** cooperating/competing
  - **shared environment**
  - various architectures: single-agent, network, **supervisor**, hierarchical etc.

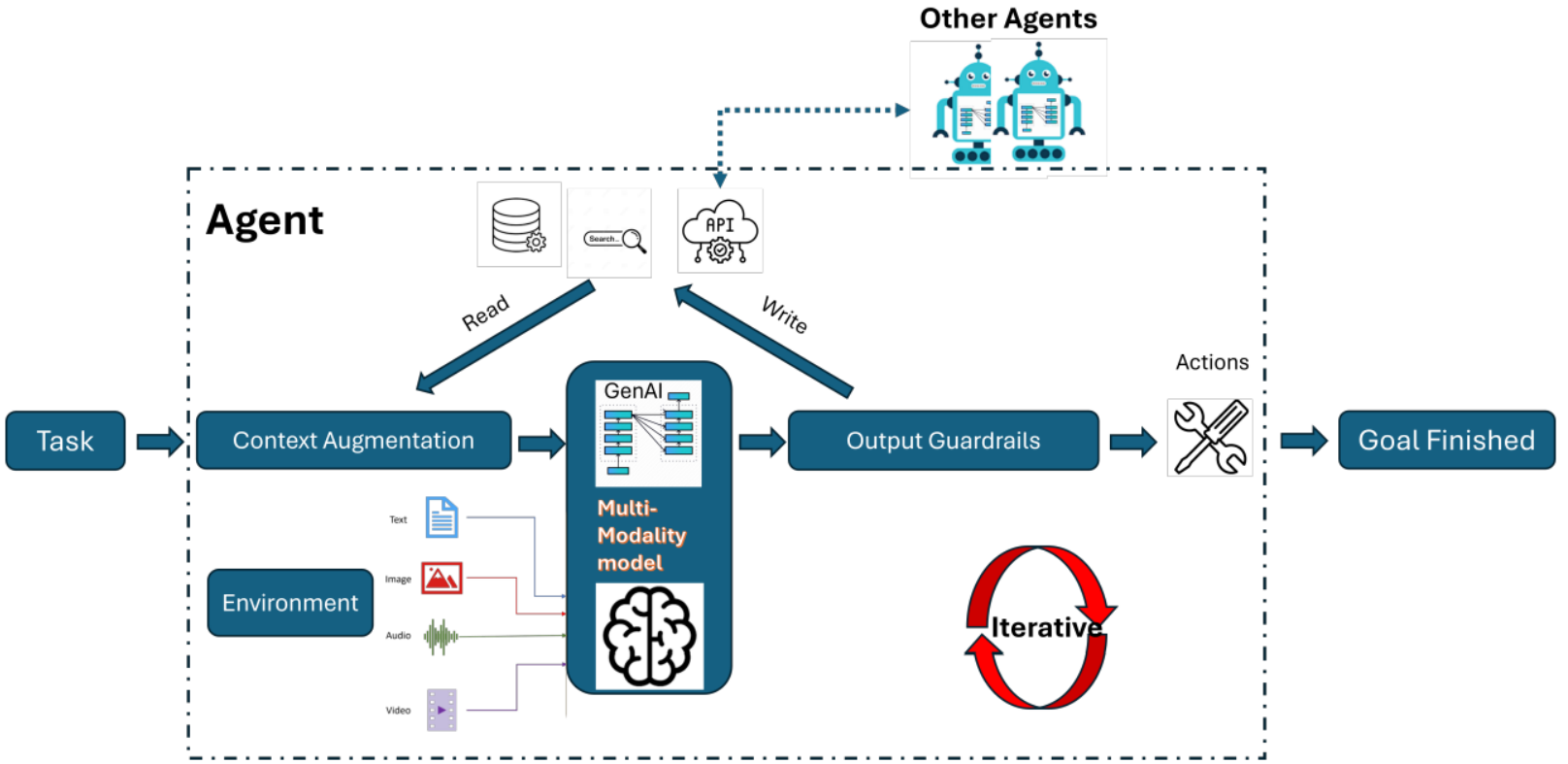
A system is more agentic the more autonomy an agent has over an application's control flow.



levi nine

Introduction (2) Perks of multi-agent systems

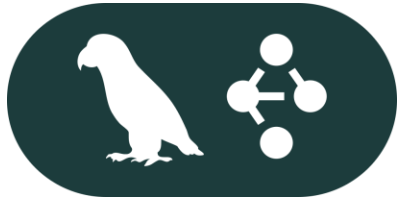
- **Modularity**
  - clarity
  - maintainability
  - testability
  - *divide and conquer* approach
- **Specialization**
  - separation of concerns
  - tooling access constraints
- **Versatility**
  - solution design

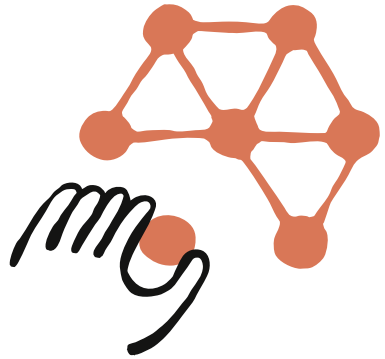


Introduction (3) Key capabilities in modern agent systems

1. **Reasoning & Planning ★**
    - anticipate, prioritize, adapt
2. **Tooling Access & Service Integrations ★**
    - interact with external tools and APIs (MCPs)
3. **Memory & Context Awareness ★**
    - retain contextual information, recall past interactions, optimize ongoing tasks
4. **Retrieval-Augmented Generation (RAG)**
    - retrieve external knowledge dynamically
5. **Instruction-Based Fine-Tuning & Prompt-Engineering**
    - define behavior, understand and execute nuanced directives

Introduction (4) Trending **frameworks**





- LangGraph
- Google Agent Development Kit (ADK)
- Claude Agents
- Amazon Bedrock Agent

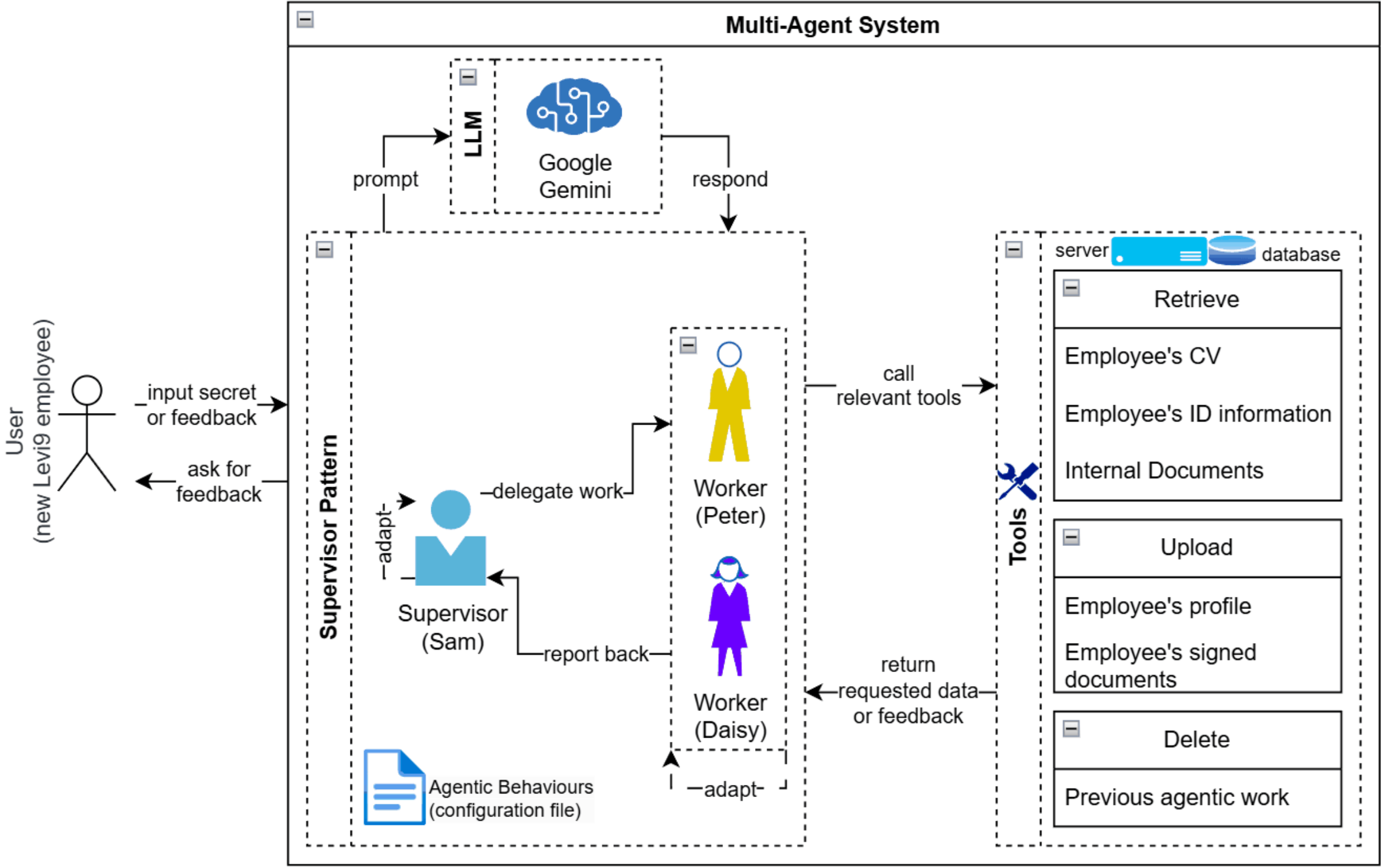
… and many others (too many!): IBM watsonx, Databricks AI, W&B Weave , Dapr Agents etc.

Demo (1) Scenario

As a **new joiner** at **Levi9**, I am required to go through some **on-boarding procedures** in order to (1) set up my profile and (2) acknowledge some legal commitments. I have received a special link from HR so I can get this done.

Conveniently, they told me that the whole process is **automated by AI agents** or something, not sure what that means, but it's supposed to make it *really quick*, so that's cool.

I just need to enter this secret code they gave me, wait a bit, and then confirm that everything looks good. **Sounds easy enough!**


Multi-Agent System
LLM
Google Gemini
prompt
respond
User
(new Levi9 employee)
input secret or feedback
ask for feedback
Supervisor Pattern
delegate work
adapt
Supervisor (Sam)
report back
Worker (Peter)
Worker (Daisy)
adapt
call relevant tools
return requested data or feedback
Agentic Behaviours (configuration file)
Tools
server
database
Retrieve
Employee's CV
Employee's ID information
Internal Documents
Upload
Employee's profile
Employee's signed documents
Delete
Previous agentic work

Demo (2) Insights

- The **supervising agent** (also called **supervisor** or **orchestrator**)
  - Manages **working agents** (**workers**)
  - **Controls the flow** and **delegates work** to workers
  - **Reasons** about the workers' **results**
  - Responds only to user input
  - **Adapt to changes in the environment**
- The **working agent** (**worker**)
  - Receives tasks from the **supervisor**
  - **Processes the taskwork**
  - **Reviews** the output against set **policies & guardrails**
  - Reports back to the supervisor
  - **Adapt to changes in the environment**
- The **tools:** APIs for **retrieving relevant data** (CV, ID, documents), **uploading output** and **deleting previous agentic work**

Demo (3) Using a **framework**

Let's explore together how we can utilize one of the more popular existing frameworks in order to **easily** develop agentic AI systems.

We'll use: **Google ADK**

Curious to learn more?

- Getting started with Agent Developer Kit
- ADK docs
- The Agent Factory

Word of advice

Keep this in mind when developing an agent system:

- **More steps** in your agentic workflow can lead to a **greater error rate** // Set up minimum bounded contexts with few discrete & verifiable operations, define manual review sections and configure escalation mechanisms
- An agent's **large context window** will create **huge costs** // Seek to optimize tokens usage right from the get-go. If possible, try to implement a stateless solution
- **Don't** expect agents to **interact properly with tools** by default // You must design proper feedback systems and adjust tools for agentic usage needs
- **Not everything needs** to be powered by **an agent** // Prioritize reliability over autonomy. Customers prefer consistency over unreliable magic

Conclusion

Agents serve well in **narrow**, **tool-integrated** environments with **defined goals**. The ability to make **context-aware**, **autonomous decisions** is **nice** but **challenging** to manage in **real-world scenarios**. **Remember,** build systems on **solid foundations** and use agents **responsibly.**

Resources

- Agentic Systems: A Guide to Transforming Industries with Vertical AI Agents
- AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges
- LLM-Powered AI Agent Systems and Their Applications in Industry
- Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey
- ... more papers

- What are AI Agents?
- Conceptual Guide: Multi Agent Architectures
- LangChain vs LangGraph: A Tale of Two Frameworks

- Why I'm Betting Against AI Agents in 2025 (Despite Building Them)

thanks ;p