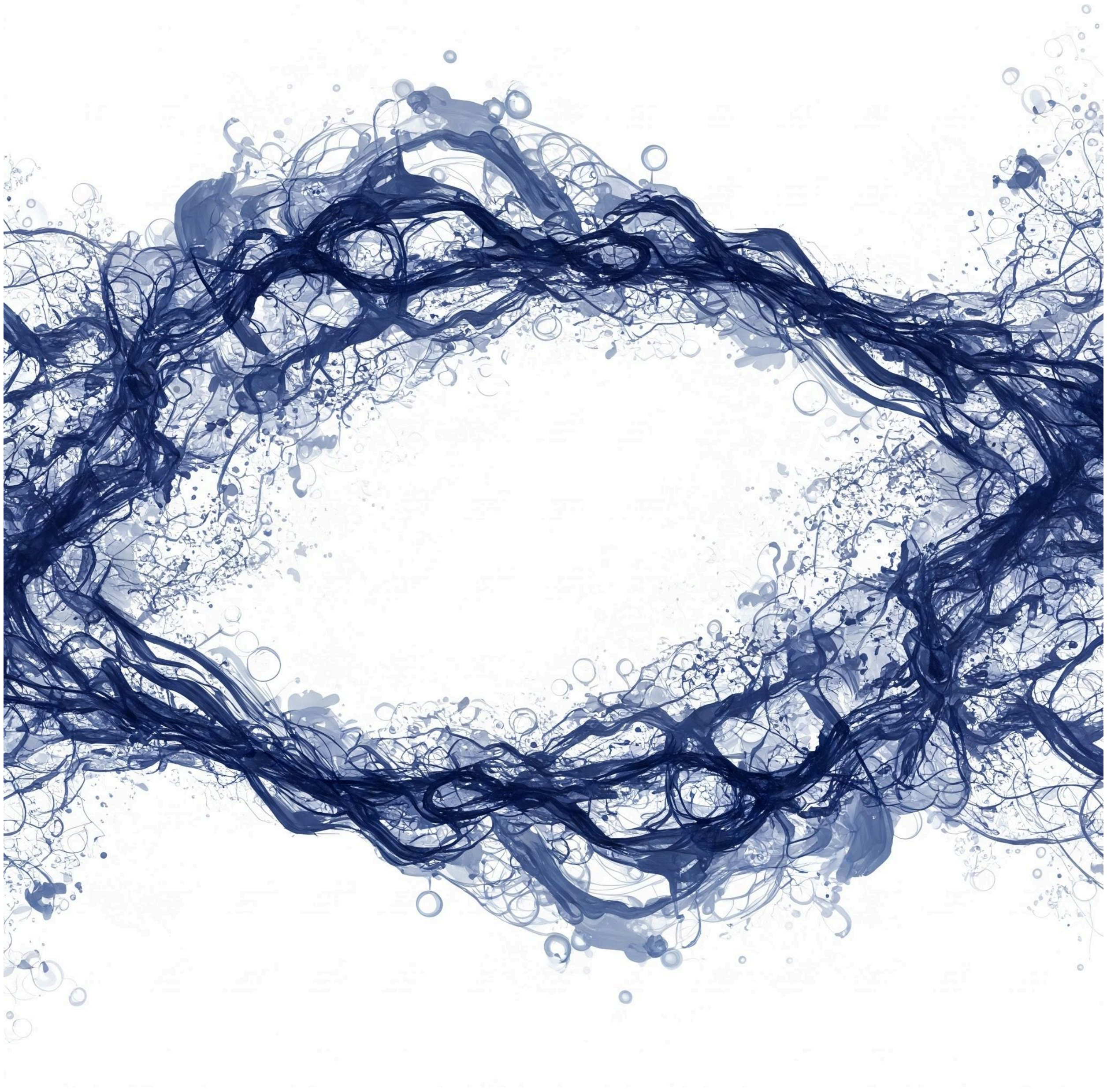
# Agentic Design Patterns

*A Hands-On Guide to Building Intelligent Systems*[1], *Antonio Gulli*

Table of Contents - total 424 pages = 1+2+1+1+4+9+103+61+34+114+74+5+4 11

**Pre Print**: *https://www.amazon.com/Agentic-Design-Patterns-Hands-Intelligent/dp/3032014018/*

[1] All my royalties will be donated to Save the Children



To my son, Bruno,

who at two years old, brought a new and brilliant light into my life. As I explore the systems that will define our tomorrow, it is the world you will inherit that is foremost in my thoughts.

To my sons, Leonardo and Lorenzo, and my daughter Aurora,

My heart is filled with pride for the women and men you have become and the wonderful world you are building.

This book is about how to build intelligent tools, but it is dedicated to the profound hope that your generation will guide them with wisdom and compassion. The future is incredibly bright, for you and for us all, if we learn to use these powerful technologies to serve humanity and help it progress.

With all my love.

# Acknowledgment

I would like to express my sincere gratitude to the many individuals and teams who made this book possible.

First and foremost, I thank Google for adhering to its mission, empowering Googlers, and respecting the opportunity to innovate.

I am grateful to the Office of the CTO for giving me the opportunity to explore new areas, for adhering to its mission of "practical magic," and for its capacity to adapt to new emerging opportunities.

I would like to extend my heartfelt thanks to Will Grannis, our VP, for the trust he puts in people and for being a servant leader. To John Abel, my manager, for encouraging me to pursue my activities and for always providing great guidance with his British acumen.I extend my gratitude to Antoine Larmanjat for our work on LLMs in code, Hann Hann Wang for agent discussions, and Yingchao Huang for time series insights. Thanks to Ashwin Ram for leadership, Massy Mascaro for inspiring work, Jennifer Bennett for technical expertise, Brett Slatkin for engineering, and Eric Schen for stimulating discussions. The OCTO team, especially Scott Penberthy, deserves recognition. Finally, deep appreciation to Patricia Florissi for her inspiring vision of Agents' societal impact.

My appreciation also goes to Marco Argenti for the challenging and motivating vision of agents augmenting the human workforce. My thanks also go to Jim Lanzone and Jordi Ribas for pushing the bar on the relationship between the world of Search and the world of Agents.

I am also indebted to the Cloud AI teams, especially their leader Saurabh Tiwary, for driving the AI organization towards principled progress. Thank you to Salem Salem Haykal, the Area Technical Leader, for being an inspiring colleague. My thanks to Vladimir Vuskovic, co-founder of Google Agentspace, Kate (Katarzyna) Olszewska for our Agentic collaboration on Kaggle Game Arena, and Nate Keating for driving Kaggle with passion, a community that has given so much to AI. My thanks also to Kamelia Aryafa, leading applied AI and ML teams focused on Agentspace and Enterprise NotebookLM, and to Jahn Wooland, a true leader focused on delivering and a personal friend always there to provide advice.

A special thanks to Yingchao Huang for being a brilliant AI engineer with a great career in front of you, Hann Wang for challenging me to return to my interest in Agents after an initial interest in 1994, and to Lee Boonstra for your amazing work on prompt engineering.

My thanks also go to the 5 Days of GenAI team, including our VP Alison Wagonfeld for the trust put in the team, Anant Nawalgaria for always delivering, and Paige Bailey for her can-do attitude and leadership.

I am also deeply grateful to Mike Styer, Turan Bulmus, and Kanchana Patlolla for helping me ship three Agents at Google I/O 2025. Thank you for your immense work.

I want to express my sincere gratitude to Thomas Kurian for his unwavering leadership, passion, and trust in driving the Cloud and AI initiatives. I also deeply appreciate Emanuel Taropa, whose inspiring "can-do" attitude made him the most exceptional colleague I've encountered at Google, setting a truly profound example. Finally, thanks to Fiona Cicconi for our engaging discussions about Google.

I extend my gratitude to Demis Hassabis, Pushmeet Kohli, and the entire GDM team for their passionate efforts in developing Gemini, AlphaFold, AlphaGo, and AlphaGenome, among other projects, and for their contributions to advancing science for the benefit of society. A special thank you to Yossi Matias for his leadership of Google Research and for consistently offering invaluable advice. I have learned a great deal from you.

A special thanks to Patti Maes, who pioneered the concept of Software Agents in the 90s and remains focused on the question of how computer systems and digital devices might augment people and assist them with issues such as memory, learning, decision making, health, and wellbeing. Your vision back in '91 became a reality today.

I also want to extend my gratitude to Paul Drougas and all the Publisher team at Springer for making this book possible.

I am deeply indebted to the many talented people who helped bring this book to life. My heartfelt thanks go to Marco Fago for his immense contributions, from code and diagrams to reviewing the entire text. I'm also grateful to Mahtab Syed for his coding work and to Ankita Guha for her incredibly detailed feedback on so many chapters. The book was significantly improved by the insightful amendments from Priya Saxena, the careful reviews from Jae Lee, and the dedicated work of Mario da Roza in creating the NotebookLM version. I was fortunate to have a team of expert reviewers for the initial chapters, and I thank Dr. Amita Kapoor, Fatma Tarlaci, PhD, Dr. Alessandro Cornacchia, and Aditya Mandlekar for lending their expertise. My sincere appreciation also goes to Ashley Miller, A Amir John, and Palak Kamdar (Vasani) for their unique contributions. For their steadfast support and encouragement, a final, warm thank you is due to Rajat

Jain, Aldo Pahor, Gaurav Verma, Pavithra Sainath, Mariusz Koczwara, Abhijit Kumar, Armstrong Foundjem, Haiming Ran, Udita Patel, and Kaurnakar Kotha.

This project truly would not have been possible without you. All the credit goes to you, and all the mistakes are mine.

*All my royalties are donated to Save the Children.*

# Foreword

The field of artificial intelligence is at a fascinating inflection point. We are moving beyond building models that can simply process information to creating intelligent systems that can reason, plan, and act to achieve complex goals with ambiguous tasks. These "agentic" systems, as this book so aptly describes them, represent the next frontier in AI, and their development is a challenge that excites and inspires us at Google.

"Agentic Design Patterns: A Hands-On Guide to Building Intelligent Systems" arrives at the perfect moment to guide us on this journey. The book rightly points out that the power of large language models, the cognitive engines of these agents, must be harnessed with structure and thoughtful design. Just as design patterns revolutionized software engineering by providing a common language and reusable solutions to common problems, the agentic patterns in this book will be foundational for building robust, scalable, and reliable intelligent systems.

The metaphor of a "canvas" for building agentic systems is one that resonates deeply with our work on Google's Vertex AI platform. We strive to provide developers with the most powerful and flexible canvas on which to build the next generation of AI applications. This book provides the practical, hands-on guidance that will empower developers to use that canvas to its full potential. By exploring patterns from prompt chaining and tool use to agent-to-agent collaboration, self-correction, safety and guardrails, this book offers a comprehensive toolkit for any developer looking to build sophisticated AI agents.

The future of AI will be defined by the creativity and ingenuity of developers who can build these intelligent systems. "Agentic Design Patterns" is an indispensable resource that will help to unlock that creativity. It provides the essential knowledge and practical examples to not only understand the "what" and "why" of agentic systems, but also the "how."

I am thrilled to see this book in the hands of the developer community. The patterns and principles within these pages will undoubtedly accelerate the development of innovative and impactful AI applications that will shape our world for years to come.

Saurabh Tiwary

VP & General Manager, CloudAI @ Google

# A Thought Leader's Perspective: Power and Responsibility

Of all the technology cycles I've witnessed over the past four decades—from the birth of the personal computer and the web, to the revolutions in mobile and cloud—none has felt quite like this one. For years, the discourse around Artificial Intelligence was a familiar rhythm of hype and disillusionment, the so-called "AI summers" followed by long, cold winters. But this time, something is different. The conversation has palpably shifted. If the last eighteen months were about the engine—the breathtaking, almost vertical ascent of Large Language Models (LLMs)—the next era will be about the car we build around it. It will be about the frameworks that harness this raw power, transforming it from a generator of plausible text into a true agent of action.

I admit, I began as a skeptic. Plausibility, I've found, is often inversely proportional to one's own knowledge of a subject. Early models, for all their fluency, felt like they were operating with a kind of impostor syndrome, optimized for credibility over correctness. But then came the inflection point, a step-change brought about by a new class of "reasoning" models. Suddenly, we weren't just conversing with a statistical machine that predicted the next word in a sequence; we were getting a peek into a nascent form of cognition.

The first time I experimented with one of the new agentic coding tools, I felt that familiar spark of magic. I tasked it with a personal project I'd never found the time for: migrating a charity website from a simple web builder to a proper, modern CI/CD environment. For the next twenty minutes, it went to work, asking clarifying questions, requesting credentials, and providing status updates. It felt less like using a tool and more like collaborating with a junior developer. When it presented me with a fully deployable package, complete with impeccable documentation and unit tests, I was floored.

Of course, it wasn't perfect. It made mistakes. It got stuck. It required my supervision and, crucially, my judgment to steer it back on course. The experience drove home a lesson I've learned the hard way over a long career: you cannot afford to trust blindly. Yet, the process was fascinating. Peeking into its "chain of thought" was like watching a mind at work—messy, non-linear, full of starts, stops, and self-corrections, not unlike our own human reasoning. It wasn't a straight line; it was a random walk toward a solution. Here was the kernel of something new: not just an intelligence that could generate content, but one that could generate a *plan*.

This is the promise of agentic frameworks. It's the difference between a static subway map and a dynamic GPS that reroutes you in real-time. A classic rules-based automaton follows a fixed path; when it encounters an unexpected obstacle, it breaks. An AI agent, powered by a reasoning model, has the potential to observe, adapt, and find another way. It possesses a form of digital common sense that allows it to navigate the countless edge cases of reality. It represents a shift from simply telling a computer *what* to do, to explaining *why* we need something done and trusting it to figure out the *how*.

As exhilarating as this new frontier is, it brings a profound sense of responsibility, particularly from my vantage point as the CIO of a global financial institution. The stakes are immeasurably high. An agent that makes a mistake while creating a recipe for a "Chicken Salmon Fusion Pie" is a fun anecdote. An agent that makes a mistake while executing a trade, managing risk, or handling client data is a real problem. I've read the disclaimers and the cautionary tales: the web automation agent that, after failing a login, decided to email a member of parliament to complain about login walls. It's a darkly humorous reminder that we are dealing with a technology we don't fully understand.

This is where craft, culture, and a relentless focus on our principles become our essential guide. Our Engineering Tenets are not just words on a page; they are our compass. We must *Build with Purpose*, ensuring that every agent we design starts from a clear understanding of the client problem we are solving. We must *Look Around Corners*, anticipating failure modes and designing systems that are resilient by design. And above all, we must *Inspire Trust*, by being transparent about our methods and accountable for our outcomes.

In an agentic world, these tenets take on new urgency. The hard truth is that you cannot simply overlay these powerful new tools onto messy, inconsistent systems and expect good results. Messy systems plus agents are a recipe for disaster. An AI trained on "garbage" data doesn't just produce garbage-out; it produces plausible, confident garbage that can poison an entire process. Therefore, our first and most critical task is to prepare the ground. We must invest in clean data, consistent metadata, and well-defined APIs. We have to build the modern "interstate system" that allows these agents to operate safely and at high velocity. It is the hard, foundational work of building a programmable enterprise, an "enterprise as software," where our processes are as well-architected as our code.

Ultimately, this journey is not about replacing human ingenuity, but about augmenting it. It demands a new set of skills from all of us: the ability to explain a task with clarity, the wisdom to delegate, and the diligence to verify the quality of the output. It requires us to be humble, to acknowledge what we don't know, and to never stop learning. The pages that follow in this book offer a technical map for building these new frameworks. My hope is that you will use them not just to build what is possible, but to build what is right, what is robust, and what is responsible.

The world is asking every engineer to step up. I am confident we are ready for the challenge.

Enjoy the journey.

Marco Argenti, CIO, Goldman Sachs

# Preface

Welcome to "Agentic Design Patterns: A Hands-On Guide to Building Intelligent Systems." As we look across the landscape of modern artificial intelligence, we see a clear evolution from simple, reactive programs to sophisticated, autonomous entities capable of understanding context, making decisions, and interacting dynamically with their environment and other systems. These are the intelligent agents and the agentic systems they comprise.

The advent of powerful large language models (LLMs) has provided unprecedented capabilities for understanding and generating human-like content such as text and media, serving as the cognitive engine for many of these agents. However, orchestrating these capabilities into systems that can reliably achieve complex goals requires more than just a powerful model. It requires structure, design, and a thoughtful approach to how the agent perceives, plans, acts, and interacts.

Think of building intelligent systems as creating a complex work of art or engineering on a canvas. This canvas isn't a blank visual space, but rather the underlying infrastructure and frameworks that provide the environment and tools for your agents to exist and operate. It's the foundation upon which you'll build your intelligent application, managing state, communication, tool access, and the flow of logic.

Building effectively on this agentic canvas demands more than just throwing components together. It requires understanding proven techniques – **patterns** – that address common challenges in designing and implementing agent behavior. Just as architectural patterns guide the construction of a building, or design patterns structure software, agentic design patterns provide reusable solutions for the recurring problems you'll face when bringing intelligent agents to life on your chosen canvas.

## What are Agentic Systems?

At its core, an agentic system is a computational entity designed to perceive its environment (both digital and potentially physical), make informed decisions based on those perceptions and a set of predefined or learned goals, and execute actions to achieve those goals autonomously. Unlike traditional software, which follows rigid, step-by-step instructions, agents exhibit a degree of flexibility and initiative.

Imagine you need a system to manage customer inquiries. A traditional system might follow a fixed script. An agentic system, however, could perceive the nuances of a customer's query, access knowledge bases, interact with other internal systems (like order management), potentially ask clarifying questions, and proactively resolve the issue, perhaps even anticipating future needs. These agents operate on the canvas of your application's infrastructure, utilizing the services and data available to them.

Agentic systems are often characterized by features like **autonomy**, allowing them to act without constant human oversight; **proactiveness**, initiating actions towards their goals; and **reactiveness**, responding effectively to changes in their environment. They are fundamentally **goal-oriented**, constantly working towards objectives. A critical capability is **tool use**, enabling them to interact with external APIs, databases, or services – effectively reaching out beyond their immediate canvas. They possess **memory**, retain information across interactions, and can engage in **communication** with users, other systems, or even other agents operating on the same or connected canvases.

Effectively realizing these characteristics introduces significant complexity. How does the agent maintain state across multiple steps on its canvas? How does it decide *when* and *how* to use a tool? How is communication between different agents managed? How do you build resilience into the system to handle unexpected outcomes or errors?

## Why Patterns Matter in Agent Development

This complexity is precisely why agentic design patterns are indispensable. They are not rigid rules, but rather battle-tested templates or blueprints that offer proven approaches to standard design and implementation challenges in the agentic domain. By recognizing and applying these design patterns, you gain access to solutions that enhance the structure, maintainability, reliability, and efficiency of the agents you build on your canvas.

Using design patterns helps you avoid reinventing fundamental solutions for tasks like managing conversational flow, integrating external capabilities, or coordinating multiple agent actions. They provide a common language and structure that makes your agent's logic clearer and easier for others (and yourself in the future) to understand and maintain. Implementing patterns designed for error handling or state management directly contributes to building more robust and reliable systems. Leveraging these established approaches accelerates your development process, allowing you to focus on the unique aspects of your application rather than the foundational mechanics of agent behavior.

This book extracts 21 key design patterns that represent fundamental building blocks and techniques for constructing sophisticated agents on various technical canvases.

Understanding and applying these patterns will significantly elevate your ability to design and implement intelligent systems effectively.

## Overview of the Book and How to Use It

This book, "Agentic Design Patterns: A Hands-On Guide to Building Intelligent Systems," is crafted to be a practical and accessible resource. Its primary focus is on clearly explaining each agentic pattern and providing concrete, runnable code examples to demonstrate its implementation. Across 21 dedicated chapters, we will explore a diverse range of design patterns, from foundational concepts like structuring sequential operations (Prompt Chaining) and external interaction (Tool Use) to more advanced topics like collaborative work (Multi-Agent Collaboration) and self-improvement (Self-Correction).

The book is organized chapter by chapter, with each chapter delving into a single agentic pattern. Within each chapter, you will find:

- A detailed **Pattern Overview** providing a clear explanation of the pattern and its role in agentic design.
- A section on **Practical Applications & Use Cases** illustrating real-world scenarios where the pattern is invaluable and the benefits it brings.
- A **Hands-On Code Example** offering practical, runnable code that demonstrates the pattern's implementation using prominent agent development frameworks. This is where you'll see how to apply the pattern within the context of a technical canvas.
- **Key Takeaways** summarizing the most crucial points for quick review.
- **References** for further exploration, providing resources for deeper learning on the pattern and related concepts.

While the chapters are ordered to build concepts progressively, feel free to use the book as a reference, jumping to chapters that address specific challenges you face in your own agent development projects. The appendices provide a comprehensive look at advanced prompting techniques, principles for applying AI agents in real-world environments, and an overview of essential agentic frameworks. To complement this, practical online-only tutorials are included, offering step-by-step guidance on building agents with specific platforms like AgentSpace and for the command-line interface. The emphasis throughout is on practical application; we strongly encourage you to run the code examples, experiment with them, and adapt them to build your own intelligent systems on your chosen canvas.

A great question I hear is, 'With AI changing so fast, why write a book that could be quickly outdated?' My motivation was actually the opposite. It's precisely because things are moving so quickly that we need to step back and identify the underlying principles that are solidifying. Patterns like RAG, Reflection, Routing, Memory and the others I discuss, are becoming fundamental building blocks. This book is an invitation to reflect on these core ideas, which provide the foundation we need to build upon. Humans need these reflection moments on foundation patterns

## Introduction to the Frameworks Used

To provide a tangible "canvas" for our code examples (see also Appendix), we will primarily utilize three prominent agent development frameworks. **LangChain**, along with its stateful extension **LangGraph**, provides a flexible way to chain together language models and other components, offering a robust canvas for building complex sequences and graphs of operations. **Crew AI** provides a structured framework specifically designed for orchestrating multiple AI agents, roles, and tasks, acting as a canvas particularly well-suited for collaborative agent systems. The **Google Agent Developer Kit (Google ADK)** offers tools and components for building, evaluating, and deploying agents, providing another valuable canvas, often integrated with Google's AI infrastructure.

These frameworks represent different facets of the agent development canvas, each with its strengths. By showing examples across these tools, you will gain a broader understanding of how the patterns can be applied regardless of the specific technical environment you choose for your agentic systems. The examples are designed to clearly illustrate the pattern's core logic and its implementation on the framework's canvas, focusing on clarity and practicality.

By the end of this book, you will not only understand the fundamental concepts behind 21 essential agentic patterns but also possess the practical knowledge and code examples to apply them effectively, enabling you to build more intelligent, capable, and autonomous systems on your chosen development canvas. Let's begin this hands-on journey!

# What makes an AI system an Agent?

In simple terms, an **AI agent** is a system designed to perceive its environment and take actions to achieve a specific goal. It's an evolution from a standard Large Language Model (LLM), enhanced with the abilities to plan, use tools, and interact with its surroundings. Think of an Agentic AI as a smart assistant that learns on the job. It follows a simple, five-step loop to get things done (see Fig.1):

1. **Get the Mission:** You give it a goal, like "organize my schedule."
2. **Scan the Scene:** It gathers all the necessary information—reading emails, checking calendars, and accessing contacts—to understand what's happening.
3. **Think It Through:** It devises a plan of action by considering the optimal approach to achieve the goal.
4. **Take Action:** It executes the plan by sending invitations, scheduling meetings, and updating your calendar.
5. **Learn and Get Better:** It observes successful outcomes and adapts accordingly. For example, if a meeting is rescheduled, the system learns from this event to enhance its future performance.

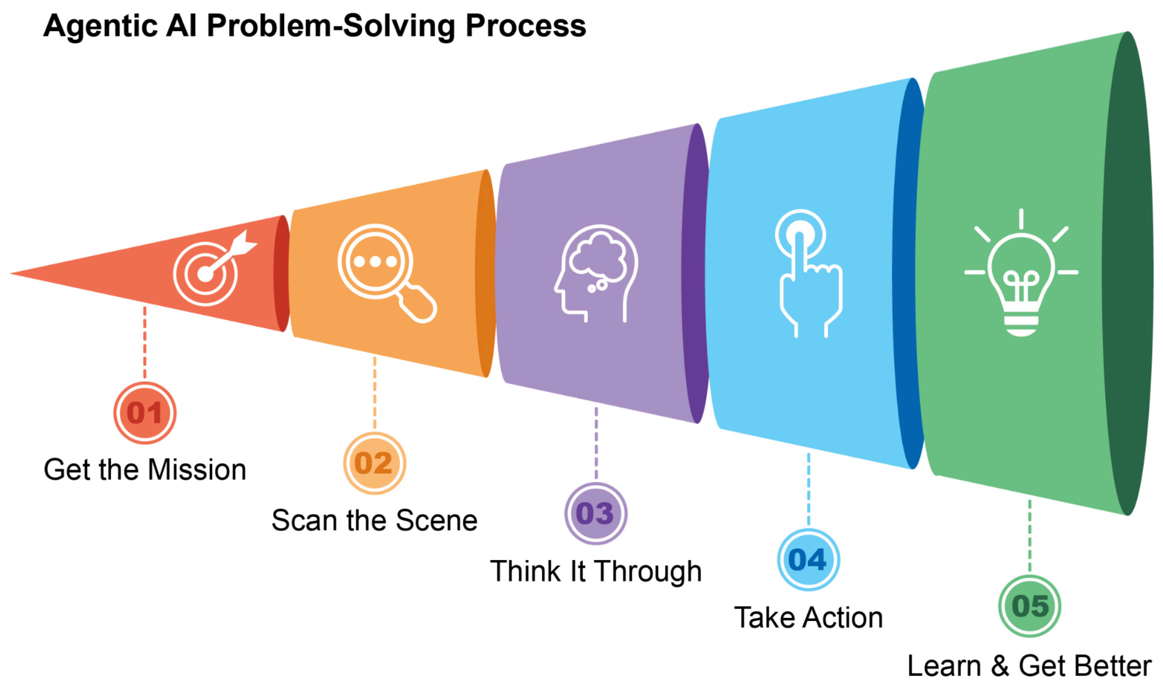


Fig.1: Agentic AI functions as an intelligent assistant, continuously learning through experience. It operates via a straightforward five-step loop to accomplish tasks.

Agents are becoming increasingly popular at a stunning pace. According to recent studies, a majority of large IT companies are actively using these agents, and a fifth of them just started within the past year. The financial markets are also taking notice. By the end of 2024, AI agent startups had raised more than $2 billion, and the market was valued at $5.2 billion. It's expected to explode to nearly $200 billion in value by 2034. In short, all signs point to AI agents playing a massive role in our future economy.

In just two years, the AI paradigm has shifted dramatically, moving from simple automation to sophisticated, autonomous systems (see Fig. 2). Initially, workflows relied on basic prompts and triggers to process data with LLMs. This evolved with Retrieval-Augmented Generation (RAG), which enhanced reliability by grounding models on factual information. We then saw the development of individual AI Agents capable of using various tools.  Today, we are entering the era of Agentic AI, where a team of specialized agents works in concert to achieve complex goals, marking a significant leap in AI's collaborative power.

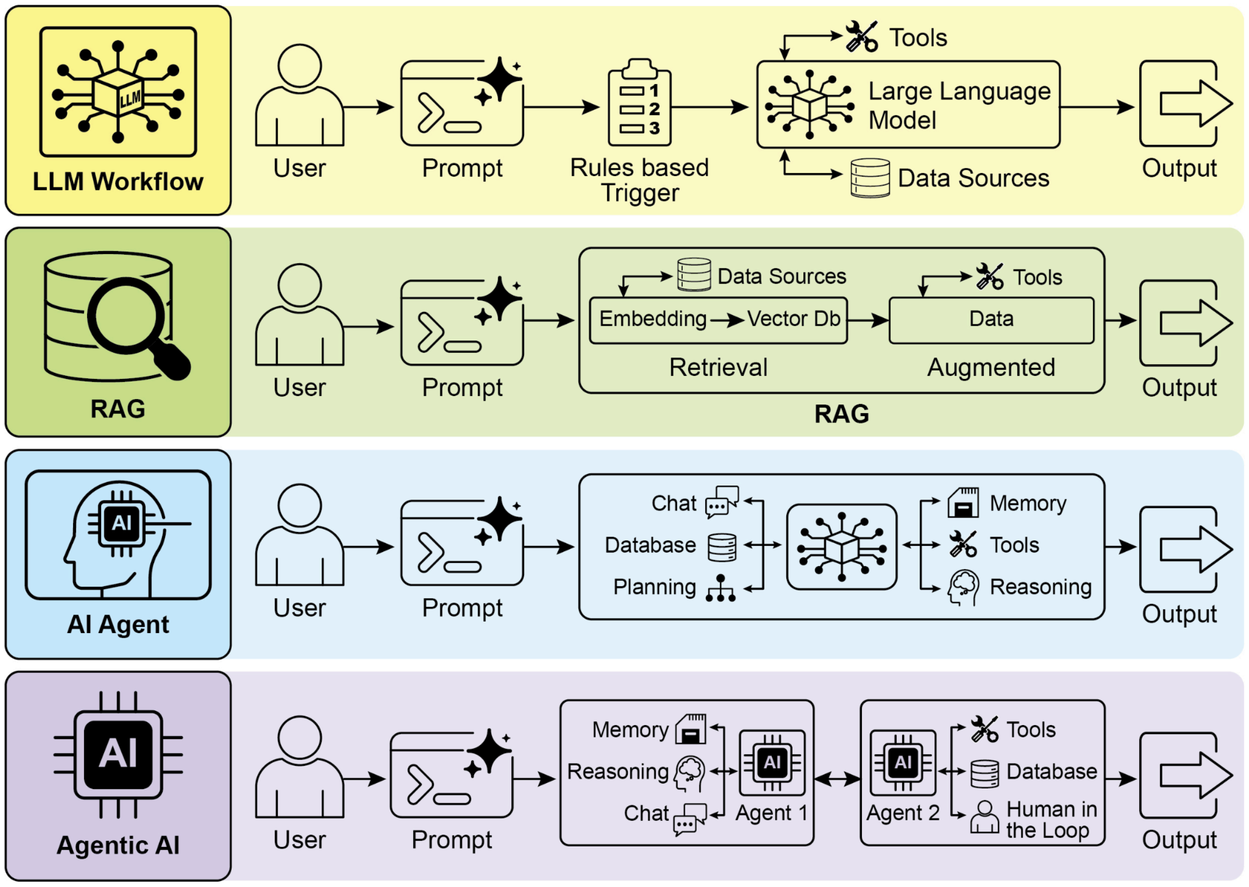


Fig 2.: Transitioning from LLMs to RAG, then to Agentic RAG, and finally to Agentic AI.

The intent of this book is to discuss the design patterns of how  specialized agents can work in concert and collaborate to achieve  complex goals, and you will see one paradigm of collaboration and interaction in each chapter.

Before doing that, let's examine examples that span the range of agent complexity (see Fig. 3).

## Level 0: The Core Reasoning Engine

While an LLM is not an agent in itself, it can serve as the reasoning core of a basic agentic system. In a 'Level 0' configuration, the LLM operates without tools, memory, or environment interaction, responding solely based on its pretrained knowledge. Its strength lies in leveraging its extensive training data to explain established concepts. The trade-off for this powerful internal reasoning is a complete lack of current-event awareness. For instance, it would be unable to name the 2025 Oscar winner for "Best Picture" if that information is outside its pre-trained knowledge.

## Level 1: The Connected Problem-Solver

At this level, the LLM becomes a functional agent by connecting to and utilizing external tools. Its problem-solving is no longer limited to its pre-trained knowledge. Instead, it can execute a sequence of actions to gather and process information from sources like the internet (via search) or databases (via Retrieval Augmented Generation, or RAG). For detailed information, refer to Chapter 14.

For instance, to find new TV shows, the agent recognizes the need for current information, uses a search tool to find it, and then synthesizes the results. Crucially, it can also use specialized tools for higher accuracy, such as calling a financial API to get the live stock price for AAPL. This ability to interact with the outside world across multiple steps is the core capability of a Level 1 agent.

## Level 2: The Strategic Problem-Solver

At this level, an agent's capabilities expand significantly, encompassing strategic planning, proactive assistance, and self-improvement, with prompt engineering and context engineering as core enabling skills.

First, the agent moves beyond single-tool use to tackle complex, multi-part problems through strategic problem-solving. As it executes a sequence of actions, it actively performs context engineering: the strategic process of selecting, packaging, and managing the most relevant information for each step. For example, to find a coffee shop between two locations, it first uses a mapping tool. It then engineers this output, curating a short, focused context—perhaps just a list of street names—to feed into a local search tool, preventing cognitive overload and ensuring the second step is efficient and accurate. To achieve maximum accuracy from an AI, it must be given a short, focused, and powerful context. Context engineering is the discipline that accomplishes this by strategically selecting, packaging, and managing the most critical information from all available sources. It effectively curates the model's limited attention to prevent overload and ensure high-quality, efficient performance on any given task. For detailed information, refer to the Appendix A.

This level leads to proactive and continuous operation. A travel assistant linked to your email demonstrates this by engineering the context from a verbose flight confirmation email; it selects only the key details (flight numbers, dates, locations) to package for subsequent tool calls to your calendar and a weather API.

In specialized fields like software engineering, the agent manages an entire workflow by applying this discipline. When assigned a bug report, it reads the report and accesses the codebase, then strategically engineers these large sources of information into a potent, focused context that allows it to efficiently write, test, and submit the correct code patch.

Finally, the agent achieves self-improvement by refining its own context engineering processes. When it asks for feedback on how a prompt could have been improved, it is learning how to better curate its initial inputs. This allows it to automatically improve how it packages information for future tasks, creating a powerful, automated feedback loop that increases its accuracy and efficiency over time. For detailed information, refer to Chapter 17.

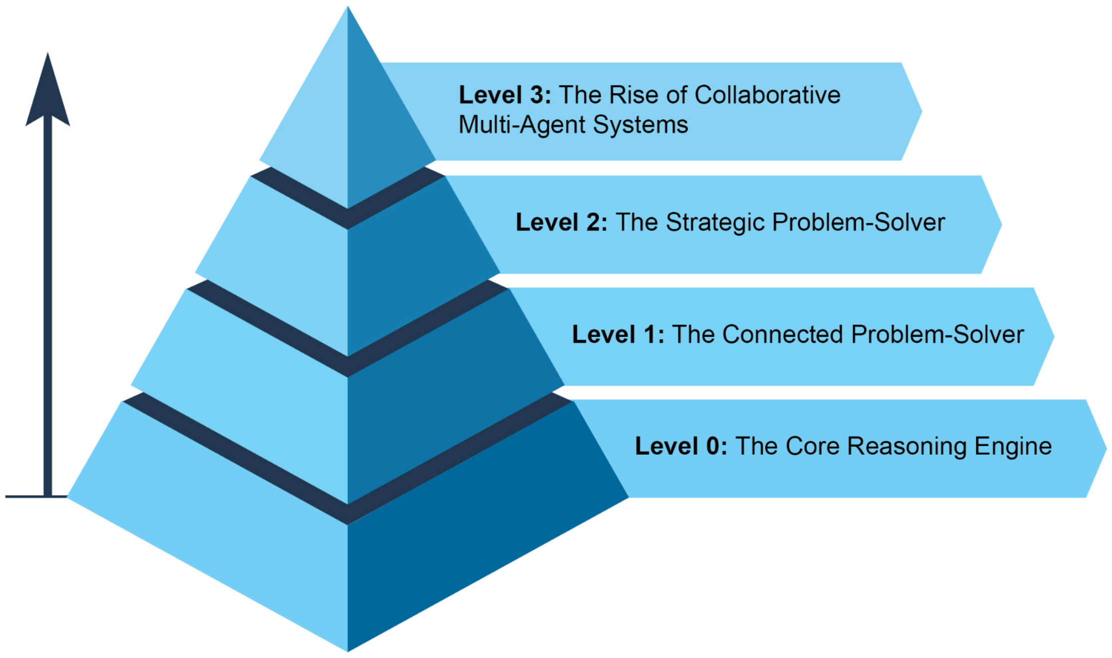

**Level 3:** The Rise of Collaborative Multi-Agent Systems
**Level 2:** The Strategic Problem-Solver
**Level 1:** The Connected Problem-Solver
**Level 0:** The Core Reasoning Engine

Fig. 3: Various instances demonstrating the spectrum of agent complexity.

## Level 3: The Rise of Collaborative Multi-Agent Systems

At Level 3, we see a significant paradigm shift in AI development, moving away from the pursuit of a single, all-powerful super-agent and towards the rise of sophisticated, collaborative multi-agent systems. In essence, this approach recognizes that complex challenges are often best solved not by a single generalist, but by a team of specialists working in concert. This model directly mirrors the structure of a human organization, where different departments are assigned specific roles and collaborate to tackle multi-faceted objectives. The collective strength of such a system lies in this division of labor and the synergy created through coordinated effort. For detailed information, refer to Chapter 7.

To bring this concept to life, consider the intricate workflow of launching a new product. Rather than one agent attempting to handle every aspect, a "Project Manager" agent could serve as the central coordinator. This manager would orchestrate the entire process by delegating tasks to other specialized agents: a "Market Research" agent to gather consumer data, a "Product Design" agent to develop concepts, and a "Marketing" agent to craft promotional materials. The key to their success would be the seamless communication and information sharing between them, ensuring all individual efforts align to achieve the collective goal.

While this vision of autonomous, team-based automation is already being developed, it's important to acknowledge the current hurdles. The effectiveness of  such multi-agent systems is presently constrained by the reasoning limitations of LLMs they are using. Furthermore, their ability to genuinely learn from one another and improve as a cohesive unit is still in its early stages. Overcoming these technological bottlenecks is the critical next step, and doing so will unlock the profound promise of this level: the ability to automate entire business workflows from start to finish.

## The Future of Agents: Top 5 Hypotheses

AI agent development is progressing at an unprecedented pace across domains such as software automation, scientific research, and customer service among others. While current systems are impressive, they are just the beginning. The next wave of innovation will likely focus on making agents more reliable, collaborative, and deeply integrated into our lives. Here are five leading hypotheses for what's next (see Fig. 4).

### Hypothesis 1: The Emergence of the Generalist Agent

The first hypothesis is that AI agents will evolve from narrow specialists into true generalists capable of managing complex, ambiguous, and long-term goals with high reliability. For instance, you could give an agent a simple prompt like, "Plan my company's offsite retreat for 30 people in Lisbon next quarter." The agent would then manage the entire project for weeks, handling everything from budget approvals and flight negotiations to venue selection and creating a detailed itinerary from employee feedback, all while providing regular updates. Achieving this level of autonomy will require fundamental breakthroughs in AI reasoning, memory, and near-perfect reliability. An alternative, yet not mutually exclusive, approach is the rise of Small Language Models (SLMs). This "Lego-like" concept involves composing systems from small, specialized expert agents rather than scaling up a single monolithic model. This method promises systems that are cheaper, faster to debug, and easier to deploy. Ultimately, the development of large generalist models and the composition of smaller specialized ones are both plausible paths forward, and they could even complement each other.

### Hypothesis 2: Deep Personalization and Proactive Goal Discovery

The second hypothesis posits that agents will become deeply personalised and proactive partners. We are witnessing the emergence of a new class of agent: the proactive partner. By learning from your unique patterns and goals, these systems are beginning to shift from just following orders to anticipating your needs. AI systems operate as agents when they move beyond simply responding to chats or instructions. They initiate and execute tasks on behalf of the user, actively collaborating in the process.  This moves beyond simple task execution into the realm of proactive goal discovery.

For instance, if you're exploring sustainable energy, the agent might identify your latent goal and proactively support it by suggesting courses or summarizing research. While these systems are still developing, their trajectory is clear. They will become increasingly proactive, learning to take initiative on your behalf when highly confident that the action will be helpful. Ultimately, the agent becomes an indispensable ally, helping you discover and achieve ambitions you have yet to fully articulate.

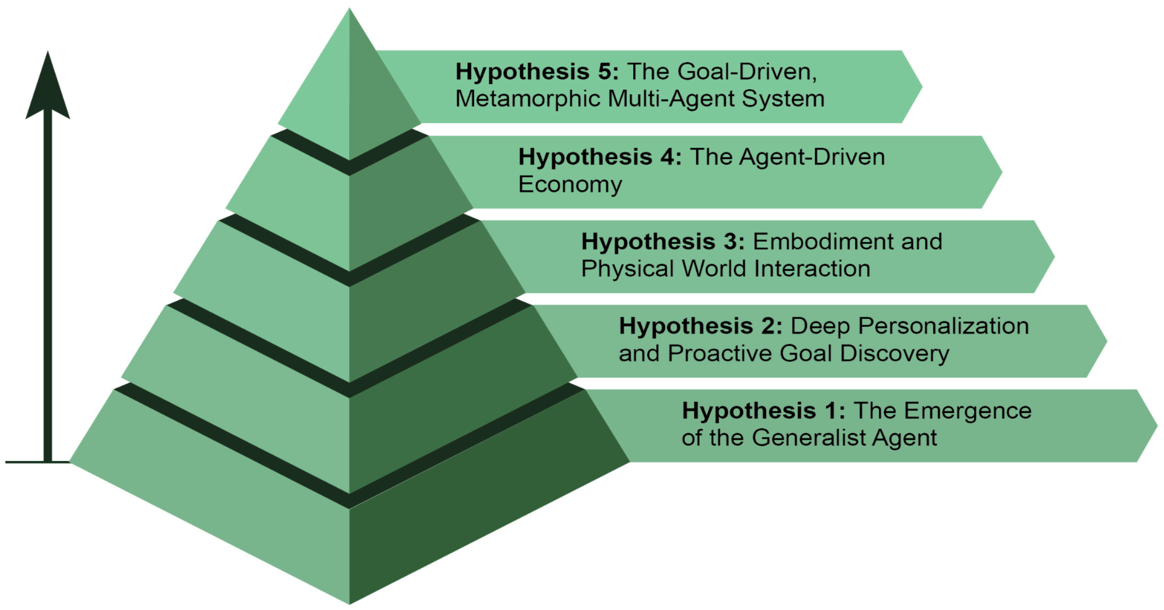

**Hypothesis 5:** The Goal-Driven, Metamorphic Multi-Agent System
**Hypothesis 4:** The Agent-Driven Economy
**Hypothesis 3:** Embodiment and Physical World Interaction
**Hypothesis 2:** Deep Personalization and Proactive Goal Discovery
**Hypothesis 1:** The Emergence of the Generalist Agent

Fig. 4: Five hypotheses about the future of agents

## Hypothesis 3: Embodiment and Physical World Interaction

This hypothesis foresees agents breaking free from their purely digital confines to operate in the physical world. By integrating agentic AI with robotics, we will see the rise of "embodied agents." Instead of just booking a handyman, you might ask your home agent to fix a leaky tap. The agent would use its vision sensors to perceive the problem, access a library of plumbing knowledge to formulate a plan, and then control its robotic manipulators with precision to perform the repair. This would represent a monumental step, bridging the gap between digital intelligence and physical action, and transforming everything from manufacturing and logistics to elder care and home maintenance.

### Hypothesis 4: The Agent-Driven Economy

The fourth hypothesis is that highly autonomous agents will become active participants in the economy, creating new markets and business models. We could see agents acting as independent economic entities, tasked with maximising a specific outcome, such as profit. An entrepreneur could launch an agent to run an entire e-commerce business. The agent would identify trending products by analysing social media, generate marketing copy and visuals, manage supply chain logistics by interacting with other automated systems, and dynamically adjust pricing based on real-time demand. This shift would create a new, hyper-efficient "agent economy" operating at a speed and scale impossible for humans to manage directly.

### Hypothesis 5:  The Goal-Driven, Metamorphic Multi-Agent System

This hypothesis posits the emergence of intelligent systems that operate not from explicit programming, but from a declared goal. The user simply states the desired outcome, and the system autonomously figures out how to achieve it. This marks a fundamental shift towards metamorphic multi-agent systems capable of true self-improvement at both the individual and collective levels.

This system would be a dynamic entity, not a single agent. It would have the ability to analyze its own performance and modify the topology of its multi-agent workforce, creating, duplicating, or removing agents as needed to form the most effective team for the task at hand. This evolution happens at multiple levels:

- Architectural Modification: At the deepest level, individual agents can rewrite their own source code and re-architect their internal structures for higher efficiency, as in the original hypothesis.
- Instructional Modification: At a higher level, the system continuously performs automatic prompt engineering and context engineering. It refines the instructions and information given to each agent, ensuring they are operating with optimal guidance without any human intervention.

For instance, an entrepreneur would simply declare the intent: "Launch a successful e-commerce business selling artisanal coffee." The system, without further programming, would spring into action. It might initially spawn a "Market Research" agent and a "Branding" agent. Based on the initial findings, it could decide to remove the branding agent and spawn three new specialized agents: a "Logo Design" agent, a "Webstore Platform" agent, and a "Supply Chain" agent. It would constantly tune their internal prompts for better performance. If the webstore agent becomes a bottleneck, the system might duplicate it into three parallel agents to work on different parts of the site, effectively re-architecting its own structure on the fly to best achieve the declared goal.

## Conclusion

In essence, an AI agent represents a significant leap from traditional models, functioning as an autonomous system that perceives, plans, and acts to achieve specific goals. The evolution of this technology is advancing from single, tool-using agents to complex, collaborative multi-agent systems that tackle multifaceted objectives. Future hypotheses predict the emergence of generalist, personalized, and even physically embodied agents that will become active participants in the economy. This ongoing development signals a major paradigm shift towards self-improving, goal-driven systems poised to automate entire workflows and fundamentally redefine our relationship with technology.

## References

1. Cloudera, Inc. (April 2025), 96% of enterprises are increasing their use of AI agents.https://www.cloudera.com/about/news-and-blogs/press-releases/2025-04-16-96-percent-of-enterprises-are-expanding-use-of-ai-agents-according-to-latest-data-from-cloudera.html
2. Autonomous generative AI agents: https://www.deloitte.com/us/en/insights/industry/technology/technology-media-and-telecom-predictions/2025/autonomous-generative-ai-agents-still-under-development.html
3. Market.us. Global Agentic AI Market Size, Trends and Forecast 2025–2034. https://market.us/report/agentic-ai-market/