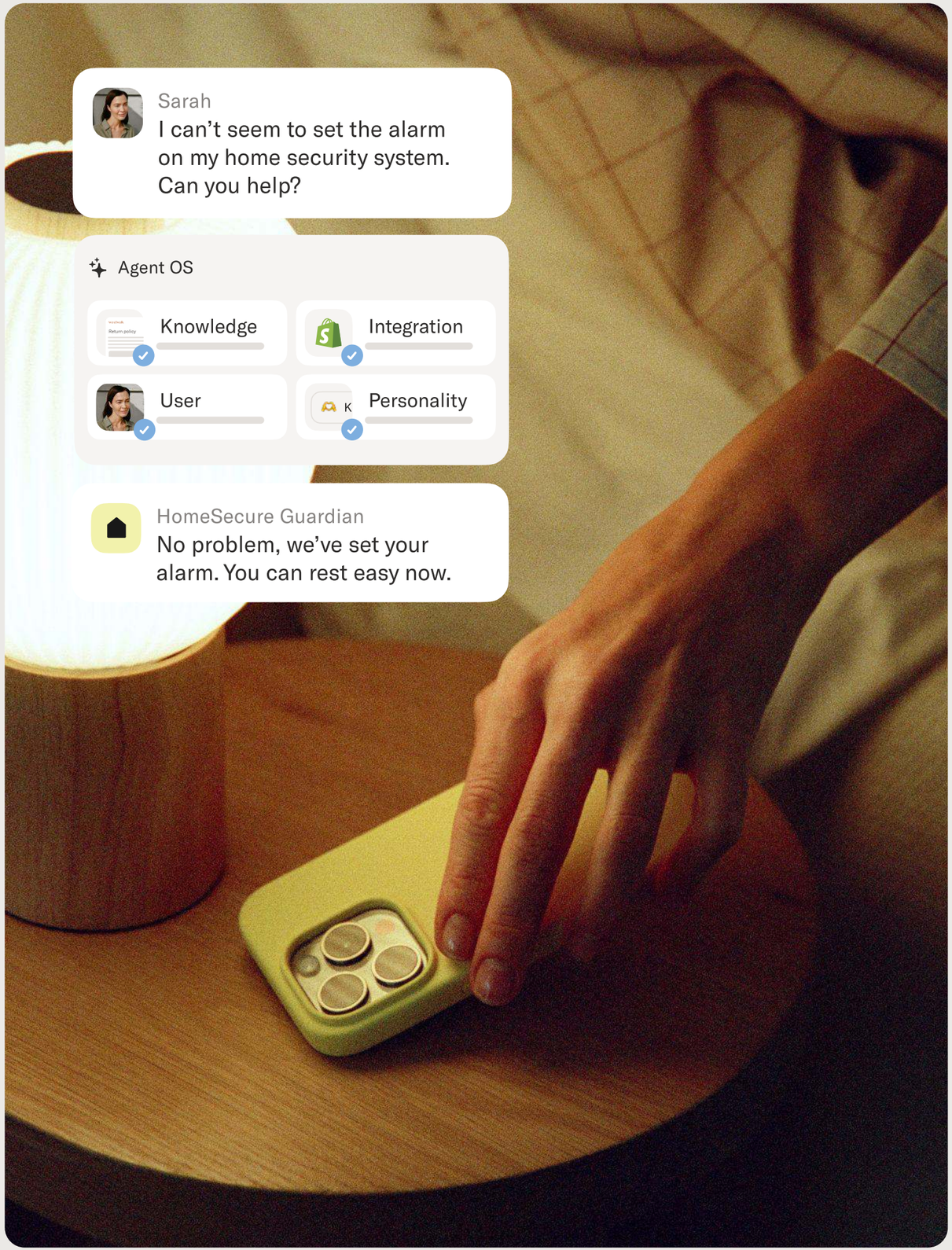
SIERRA

The Guide to AI Agents

Sierra enables every company in the world to build its own branded, customer-facing AI agent for everything from customer service to commerce. You can read more about our mission, our platform, and our early customers here.

This book is all about AI agents: what they are, how they work, and why we're excited about them. It's intended for a general audience, so whether you have a background in computer science or not, if you're curious about agents, read on.



What are agents?

Agents are a new kind of software.

Traditional software applications act as tools to help you get a job done—to look up information, to analyze some data or write a report, to prepare a budget. Applications make the work easier, but you still have to do the work.

Agents are different. Agents are autonomous software systems that can reason, make decisions, and pursue goals with creativity and flexibility, all while staying within the bounds that have been set for them. Whereas applications help you do the work, agents get the work done for you.

The three types of agents

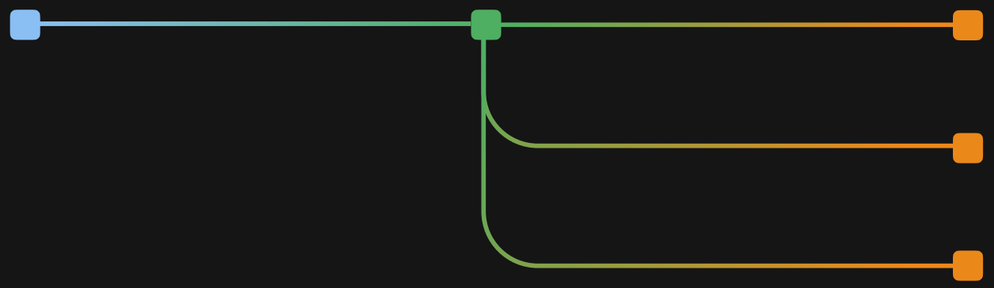
AI agents have the potential to be so useful that we believe that they will soon be everywhere.

In your personal life, you'll use an AI agent as an assistant to book meetings and draft emails.

Within your company, you'll use role-specific AI agents to act as software developers, data analysts, or even paralegals to streamline and automate all sorts of routine internal tasks and processes.

And outside your company, your customers will use your company's conversational AI agent to make purchases, answer questions, manage subscriptions, and more. To a customer interacting with your company's agent, it will feel a lot like conversing with a field of experts at the company—the veteran friendly sales associate, the genius customer support person, even the company historian. Your company's agent will soon be as important as its website or mobile app, and will become the ultimate manifestation of your company's brand.

TYPE	BUILT FOR	EXAMPLE USES
Personal Agents	Consumers	Scheduling meetings Planning trips
Role-based Agents	Employees	Reviewing contracts Developing software
Company Agents	Businesses	Customer service Subscription management Sales assistance



Large language models: The brains of AI agents

Large language models (LLMs) are the brains of AI agents.

LLMs like Google's Gemini and OpenAI's GPT-4, which powers ChatGPT, are remarkable, and have enabled three important and foundational new capabilities that are relevant to building AI agents.

First, LLMs enable sophisticated natural language understanding. These models can make sense not just of the content of sentences, but can understand context, tone, sarcasm, and jargon. It's a stark difference from the days of trying to wrestle the weather or sports scores out of Alexa. This natural language capability is key in enabling an agent to take instructions and make sense of information that it comes across.

Second, LLMs can generate fluent language in the form of text and audio, using context and instructions to create nuanced answers. In the context of building an AI agent that can "talk" and take on a particular tone or voice, this capability is central.

Finally, LLMs can reason and solve problems. Incredibly, GPT-4 has aced high school Advanced Placement exams in biology, statistics, and economics. And LLMs aren't only capable of answering questions in broad areas of general knowledge. Used properly, LLMs can solve problems that are highly context-specific.

It's these capabilities of LLMs—natural language understanding, language generation, and reasoning—that have for the first time made it possible to build sophisticated AI agents. But an LLM by itself is not an AI agent. To build an agent, you need more.

Beyond LLMs: Tools, memories, and plans

Sophisticated AI agents consist of a number of components, all working together to enable an agent that can converse fluently, answer nuanced questions, and solve complex problems. It's a bit like an orchestra, with each component serving a distinct and important role and, by working in concert with others, enabling something greater than the sum of the parts.

To start, an agent might use an LLM many times to break down a problem and perform subtasks. It might first call an LLM to take stock of the current conversation, summarizing what has happened so far into a sort of working memory. Then, it might call another LLM to plan out possible next actions, and a third to evaluate the quality of each candidate action. Then, it might make a final LLM call to generate an answer for the end user. By combining many separate LLM calls, each for a special purpose, agents can achieve far higher performance than would be possible with a single LLM call.

Agents can also use long-term memory to answer questions or solve problems. For instance, a retailer could give their agent access to detailed knowledge of each of its products, its warranty policy, and even its company history, enabling it to answer a broad range of customer questions.

Of course, reasoning, planning, and answering questions isn't enough. In order to be able to do anything, agents need to be able to take action and interact with systems and services around it. But no matter how powerful the latest LLM is, it can't possibly know where a package is, or the status of a subscription, or the last time a customer wrote in. This type of information lies in your company's own internal systems.

Giving an agent access to "tools to use" via APIs—interfaces that enable one piece of software to talk to another—enables an agent to interact with these systems and take action. Sierra agents, for instance, can look up a customer's order history in an order management system, process a return, search a channel lineup, pull diagnostic information for a customer's device to help troubleshoot, change a subscription plan, and more.

Finally, to be able to reliably perform complex, multi-step tasks, an agent needs a repository of "plans" which capture procedural knowledge and how those tasks are accomplished, similar to a handbook or instruction manual that you'd hand a new team member. These plans act as a sort of scaffolding for the agent, ensuring that it performs steps in the right order, stays on task, and gets the job done.

Trust, safety, and agents monitoring agents

A key attribute of AI agents is that they're autonomous. They can converse, reason, and take action—all on their own. As a technology in the abstract, that's really powerful. But if you're putting an AI agent in front of your customers, a key question is: how can you trust it?

Trust starts with being able to define and enforce guardrails for an agent. The Sierra platform makes it possible to express and enforce deterministic rules and business logic where needed. For example, if your company's exchange policy specifies a 30-day window, an agent should never be able to process an exchange if a customer's purchase falls outside that range.

Tools for monitoring and auditing are also part of the solution. For instance, a well architected agent should be able to log reasoning traces for each decision it makes, making it possible to correct and improve its performance and behavior over time.

Finally, it turns out that a key ingredient in ensuring trust and safety in AI agents is... more AI agents. LLMs can often identify and fix their own errors, similar to how you might proofread an email before hitting send. This property can be applied to improving the performance, reliability, and safety of an agent by creating secondary agents to do things like spot and correct errors, keep an agent on topic, or escalate to a person when needed. Sierra agents for instance come paired with a set of specialist "supervisors" that can do things like monitor answers for factuality, ensure that agents don't dispense medical advice, and detect if an agent is being misused by an end user.

By layering agents in this way, it's possible to create an agent that is far more capable, more reliable, and more trustworthy.