The Lever

Putting AI Agents to Work

FROM COLUMBIA ENGINEERING

The Lever

Welcome

It's been nearly three years since ChatGPT stunned the world with its near-human ability to carry on a conversation. The real breakthrough was in the underlying technology, called large language models (LLMs).

Animating chatbots was just the start. LLMs can navigate the internet, fill out forms, send emails, and make payments. When LLMs are empowered to take action beyond the dialogue box, they're called AI agents.

If you gave it the right tools, an AI agent could plan an entire trip, picking (and booking) everything from the Lyft to the airport to your hotels and excursions. But just like ChatGPT, AI agents are prone to making wild and unpredictable mistakes.

In the pages below you'll hear from AI experts with experience conducting fundamental research, supporting enterprises as they add AI agents to their organizations, and training the first generation of full-stack AI engineers.

Tomorrow, Professor Lydia Chilton will lay out the first steps you have to take before weaving agents into your workflows.

Table of Contents

Map Out Your Workflows

Before you put AI agents to work, you have to know exactly what they need to accomplish.

To build an AI strategy, you have to understand how you, your team, or your whole organization are getting work done.

In most workplaces, knowledge about how things get done is buried in email threads, Slack messages, and informal agreements. When information is scattered and inconsistent, it's nearly impossible for AI to take on the drudgery that underpins complex operations.

That's why the first step is to understand how work gets done — by you, your team, or your entire organization. Especially the mundane stuff. I'm talking about filling out forms, chasing stalled conversations, and shepherding documents through layers of approval and revision.

AI agents aren't just suggesting text anymore. I've seen agents handle everything from simple scheduling to more complex coordination tasks, like planning resource allocation or accounting for unexpected hiccups that could send disruptive ripples across teams. There's more upside — and stronger competitive pressure — every day.

But agents can only help if your systems make sense to machines. That means defining inputs and outputs clearly, writing down rules, digitizing steps, and building APIs, which let digital tools share information automatically.

Here's how I suggest starting:

— **Define your workflows.** What tasks need to get done? What information is needed to do those tasks? What does success look like?

— **Remove manual blockers.** If someone has to write information by hand or transfer it manually, redesign the process.

— **Standardize context.** Agents shouldn't have to guess what "done" means. Make rules and data explicit.

— **Let agents learn.** If you show them a few examples, they'll remember what to do—and know when to ask for help.

Personally, I'm excited for a future where we don't waste time filling out forms or navigating clunky systems. The less time we spend pushing paper, the more time we can spend doing meaningful work. That's the real promise of this technology.

In the page below, my colleague Zhou (Jo) Yu will share her perspective on why starting with the right use case is key to deploying AI agents in your organizations.

If you'd like to learn more about partnering with Columbia researchers working at the forefront of applied research in AI, visit the DAPLab website or contact Lydia Chilton.

LYDIA CHILTON IS AN ASSOCIATE PROFESSOR OF COMPUTER SCIENCE AT COLUMBIA ENGINEERING AND A VISITING SCIENTIST AT AMAZON.

FOR MORE INFORMATION, VISIT DAPLAB.CS.COLUMBIA.EDU OR CONTACT LYDIA AT LC3251@COLUMBIA.EDU

Start by Solving One Problem

Finding the right initial use case is key to successfully deploying AI agents.

Leaders often talk about "adopting AI" like it's a vision statement that redefines everything happening in an organization. In my experience, success is far more likely when leaders identify one discrete use case where an AI agent can deliver measurable results. This "wedge" creates momentum, builds internal trust, and uncovers practical lessons that make future deployments easier and more impactful.

My colleagues and I have gained experience helping small and large organizations deploy agents to handle specific, repetitive jobs like responding to customer inquiries, basic bookkeeping, and managing customer records. Automating just one time-consuming task can show leaders across an organization the value that agents can bring.

For instance, startups like Relevance AI and Arklex.AI (founded by me and my students) are helping major corporations use AI sales agents to increase revenue by handling customer interactions faster and more consistently than their human team could.

The key is to find a clear need that's suited to automation. That usually means an operation that relies on repetitive communication and well-defined goals, like communicating with customers who run into common issues. Once you manage to solve that initial problem, it's much easier to extend the model to other internal processes and departments.

Given the significant progress in LLMs and other AI technologies, the real challenge in deploying an agent isn't technical — it's organizational. People are protective of their work. They don't always want to write down what they know or change how things get done. That's why success usually requires top-down commitment. In some cases, leaders had to insist: write down your process or risk being left behind.

If you're preparing to bring agents into your organization:

— **Start with a wedge.** Choose a task that's high-volume and well understood. We've seen successful implementations center on sales follow-up, customer service, and internal onboarding.

— **Design for flexibility.** Different teams and functions have different needs. Start with this in mind, and build templates that can work across teams with minimal rewiring.

— **Plan for culture change.** People need to see agents as support, not surveillance. Make productivity gains visible and reward the humans who made them possible.

— **Enlist leadership.** Cultural resistance is real. Signals from the top of the org chart can make or break implementation.

Using agents to solve one problem (especially if it drives revenue or brings down expenses) can create the momentum to roll out the technology across an organization.

In the page below my colleague Vishal Misra will discuss the problems that agents can cause — and how you can mitigate that risk.

If you'd like to learn more about partnering with Columbia researchers working at the forefront of applied research in AI, visit the DAPLab website or contact co-director Zhou (Jo) Yu.

ZHOU (JO) YU IS AN ASSOCIATE PROFESSOR OF COMPUTER SCIENCE AT COLUMBIA ENGINEERING, A CO-DIRECTOR OF DAPLAB, AND THE FOUNDER AND CEO OF ARKLEX.AI

FOR MORE INFORMATION, VISIT DAPLAB.CS.COLUMBIA.EDU OR CONTACT ZHOU AT ZY2461@COLUMBIA.EDU

Don’t Get Lost in the Hype

There’s a lot to consider before letting agents write in critical databases.

For years, people have predicted that AI agents would handle tasks like booking a vacation, taking care of everything from buying flights to reserving hotels. But I don’t think anyone is ready to hand over their credit card without checking the itinerary first. I feel the same way about applying AI agents to mission-critical systems in enterprise settings.

Agents that write code demonstrate why. There are already plenty of models that can scan a codebase, understand complex requests, and generate sophisticated software. But their output often includes hallucinations and code that falls short of professional standards. That software could cause significant problems if implemented without serious oversight.

Imagine a company giving that same kind of agent access to its sales database. One hallucinated instruction could delete every customer record — wiping out years of business data and leaving the sales team scrambling. It would be catastrophic.

Importantly, this isn’t a shortcoming of the training data. In fact, software engineering is one of the best-documented domains, with many large repositories of code available. If agents still struggle in this ideal environment, it’s unrealistic to expect dramatic improvements for agents that are trained to perform more niche tasks, much less ones that are trained on a single company’s data.

That’s why a growing ecosystem of tools and practices — called “scaffolding” — is emerging to keep agents in check. These safety layers help constrain agents’ permissions, check their outputs before deployment, and prevent mistakes from cascading across critical systems. While developers and researchers are often enthusiastic about what agents could do, many people in industry remain cautious. The chance of a particularly consequential mistake is a huge business risk.

Instead of hoping for perfect models, we need better guardrails. That means designing safety layers that check agents’ outputs, constrain their permissions, and ensure that failures don’t cascade through key systems.

In the page below my colleague Kostis Kaffes will dig deeper into what you need to think about when building those guardrails.

If you’d like to learn more about partnering with Columbia researchers working at the forefront of applied research in AI, visit the DAPLab website or contact Vishal Misra.

VISHAL MISRA IS A PROFESSOR OF COMPUTER SCIENCE AND VICE DEAN OF COMPUTING AND ARTIFICIAL INTELLIGENCE AT COLUMBIA ENGINEERING.

FOR MORE INFORMATION, VISIT DAPLAB.CS.COLUMBIA.EDU OR CONTACT VISHAL AT VISHAL.MISRA@COLUMBIA.EDU

Imagine Everything That Could Go Wrong

Agents are a powerful tool, but it's essential to consider the damage they can do.

Mistakes aren't the problem. Humans make them all the time.

For people and organizations that are implementing AI agents, the real threats are mistakes that happen quietly, quickly, and at scale. A hallucination sends funds to the wrong account or deletes important records could cause enormous damage.

By definition, AI agents take actions that can't necessarily be undone. When an agent edits a database, sends an email, or initiates a credit card transaction, it makes a real-world change with real-world consequences.

When something goes wrong, it might take hours (or days) to notice.

To implement a responsible AI strategy, you have to work backward by imagining what can go wrong and taking steps to prevent unacceptable outcomes. But catching mistakes quickly isn't enough. We also need to give agents structured ways to explore, experiment, and learn safely, so they can reduce the number of mistakes they make in the first place.

As a systems researcher, I've seen problems propagate through a network faster than the human brain could begin to understand what was going on. These issues can compromise fundamental requirements like data integrity, customer privacy, and legal obligations.

From my perspective, anyone implementing AI agents should take a few basic steps to prevent the worst outcomes:

— **Track what changes.** You need full lineage: what changed, when, by which agent, and who relied on it next. Without this, even basic troubleshooting becomes impossible.

— **Simulate first.** Don't let agents take just one action. They need to test dozens before choosing. Your infrastructure should support isolated, fast simulations so they're not experimenting directly on systems with real-world implications.

— **Spread out your safeguards.** There's no single "safety layer." You need checks at different levels of the stack that work together to catch problems early.

At Columbia Engineering, we're working with partners across sectors to help systems handle these edge cases before they become headlines. Because in this next phase of AI, a robust system isn't a plus, it's essential.

In the next page and final issue of this series, my colleague Eugene Wu will explore what it takes to build the kind of infrastructure that makes AI agents trustworthy.

If you'd like to learn more about partnering with Columbia researchers working at the forefront of applied research in AI, visit the DAPLab website or contact Kostis Kaffes.

KOSTIS KAFFES IS AN ASSISTANT PROFESSOR OF COMPUTER SCIENCE AT COLUMBIA ENGINEERING.

FOR MORE INFORMATION, VISIT DAPLAB.CS.COLUMBIA.EDU OR CONTACT KOSTIS AT KKAFFES@CS.COLUMBIA.EDU

Build Systems Worth Trusting

AI agents need the right guardrails to put humans at ease.

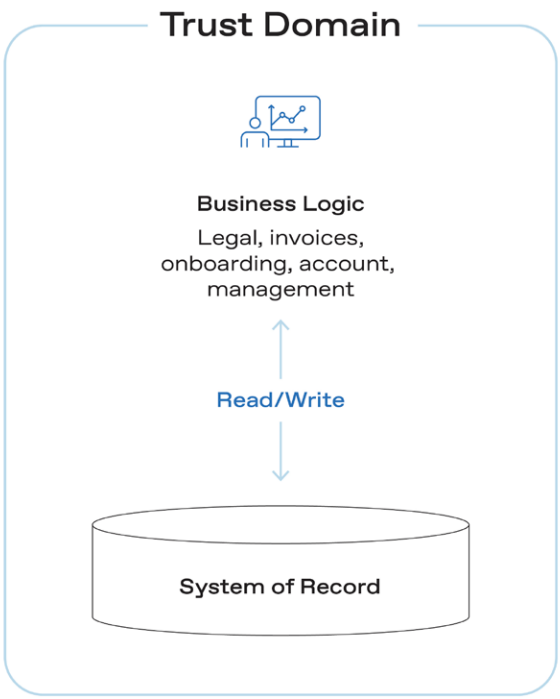
Imagine a new technology that eliminates office drudgery by making business operations faster and more efficient, but at a cost: it's prone to mistakes, hard to trust, and hungry for energy.

This isn't just the story of AI in 2025. It was also the story of relational databases from the 1980s.

From a systems perspective, the challenges are similar — and so are the solutions.

Back then, business software was expensive and brittle, with no reliable infrastructure for managing data. The lack of trust in the underlying data systems, which we call a "trust wall," meant developers had to rewrite vast swaths of the application logic anytime they wanted to optimize the data layout, slightly change the data model, or simply add a new feature. Relational databases made it reasonable to trust these digital systems. Once that infrastructure existed, entire industries scaled. That shift underpins nearly every enterprise system in use today.

AI agents now face a similar "trust wall." Many organizations let them read data or draft documents, but hesitate to let them take action, like submitting a form or updating a record. And that's not unreasonable: while agents are powerful, they're also unpredictable, prone to hallucination, and tough to monitor in real time. Even a single mistake could be catastrophic because the systems agents operate within weren't built with those shortcomings in mind.



CONTINUED ON NEXT PAGE →

To unlock the next wave of automation, we don't just need better agents. We need to build systems that make trust possible.

So, what does it take to make agents trustworthy? We need systems that let agents plan, simulate, and act in controlled environments that are designed to take advantage of their strengths while mitigating their weaknesses:

— **Sandboxed environments** where agents can explore hundreds of actions before committing to one.

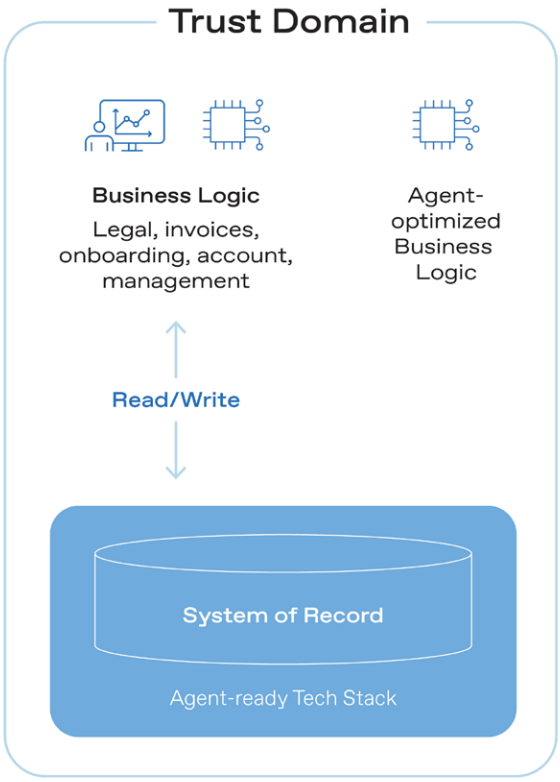
— **Forkable infrastructure** that isolates mistakes, preventing them from cascading through the system.

— **Built-in safeguards** at the data layer that block unverified actions from affecting live operations.

When that kind of infrastructure exists, agents will move beyond offering suggestions and start taking action. These components will also fundamentally shape how agents and models are designed. They will guide what assumptions are safe to embed into models, how to optimize for efficiency and reliability, and how to structure agent interaction patterns. Moreover, this infrastructure will inform the design of user experiences that give people the right level of visibility, control, and trust in the agents acting on their behalf.

To unlock the next wave of automation, we don't just need better agents. We need to build systems that make trust possible.

If you'd like to learn more about partnering with Columbia researchers working at the forefront of applied research in AI, visit the DAPLab website or contact co-director Eugene Wu.



EUGENE WU IS AN ASSOCIATE PROFESSOR OF COMPUTER SCIENCE AT COLUMBIA UNIVERSITY AND A CO-DIRECTOR OF THE NEW DATA, AGENTS, AND PROCESSES LAB (DAPLAB), WHICH BRINGS TOGETHER FACULTY ACROSS AI, SYSTEMS, HCI, AND ALGORITHMS TO BUILD HOLISTIC, END-TO-END SOLUTIONS TO THESE AND OTHER CHALLENGES TOWARDS WIDE-SPREAD ADOPTION OF AGENTIC AUTOMATION. THE LAB WORKS CLOSELY WITH INDUSTRY PARTNERS TO SHAPE THE RESEARCH AGENDA. FOR MORE INFORMATION, SEE DAPLAB.CS.COLUMBIA.EDU OR CONTACT EUGENE AT EWU@CS.COLUMBIA.EDU