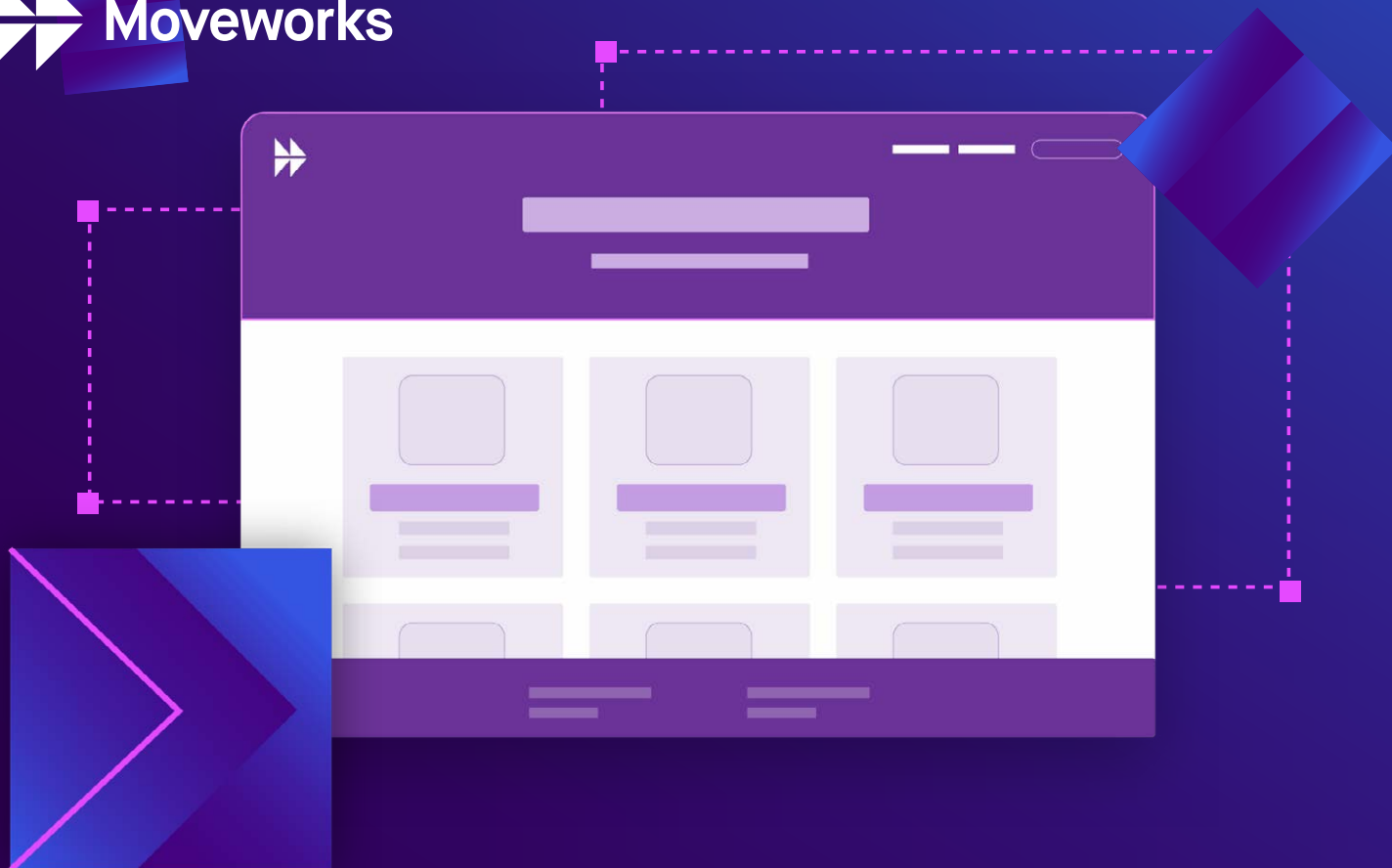




The Ultimate Guide to AI Agents

Intro

AI agents are the future

Automation is undergoing a radical transformation. Traditional systems that once handled predefined business processes with ease are now facing limitations when it comes to more complex, reasoning-based operations. Enter AI agents—a new era of automation driven by advanced Large Language Models (LLMs).

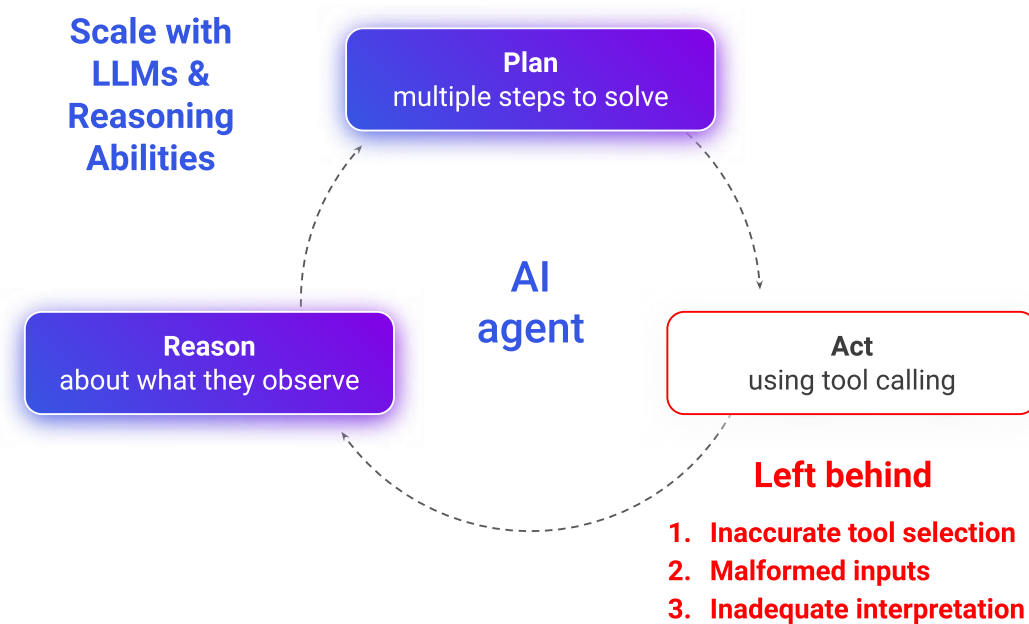
An “AI agent” is an autonomous system, powered by LLMs, which is capable of completing tasks on behalf of a user.

AI agents have the unique ability to interpret natural language, develop a plan of action, and execute tasks with minimal human intervention. Using AI agents, businesses now have the opportunity to achieve unprecedented efficiency, allowing employees to focus on higher-level thinking while automation handles the mundane.

What are AI agents?

An “agent” is an entity that acts on behalf of another person or group
([Oxford English Dictionary](#))

An “AI agent” is an autonomous system, powered by Large Language Models, which is capable of completing tasks on behalf of a user.



Just imagine the power of AI agents to rapidly accelerate and streamline your automation development process, centralizing your automation tasks into a single platform, and offering built-in compliance as agents are programmed to adhere to company policies.

But, despite the early promise, most Agentic AI systems have yet to fully live up to the hype.

We want to help you understand Agentic Automation, the power of Moveworks Agentic Automation Engine, and show how it is able to make your development faster and easier than ever. That’s because Moveworks Agentic Automation doesn’t just connect APIs – it also enables reasoning, interpretation, and decision-making, using LLMs to take intelligent action.

Let's explore how Agentic Automation empowers developers to easily build and use AI agents to bridge the gap between human language and structured system APIs.

Witness firsthand how the Moveworks Agentic Automation Engine is able to seamlessly integrate with various systems, helping to enable your AI agents to understand complex requests, make informed decisions, and automate a wide range of tasks. Now keep reading and get ready to start unleashing the full potential of Agentic Automation and supercharging your development projects.

Table of Contents

Intro

AI agents are the future

04

How the Moveworks Reasoning Engine supercharges productivity with Agentic AI

01

The challenges of traditional automation platforms

02

Agentic automation –
A new class of tools

03

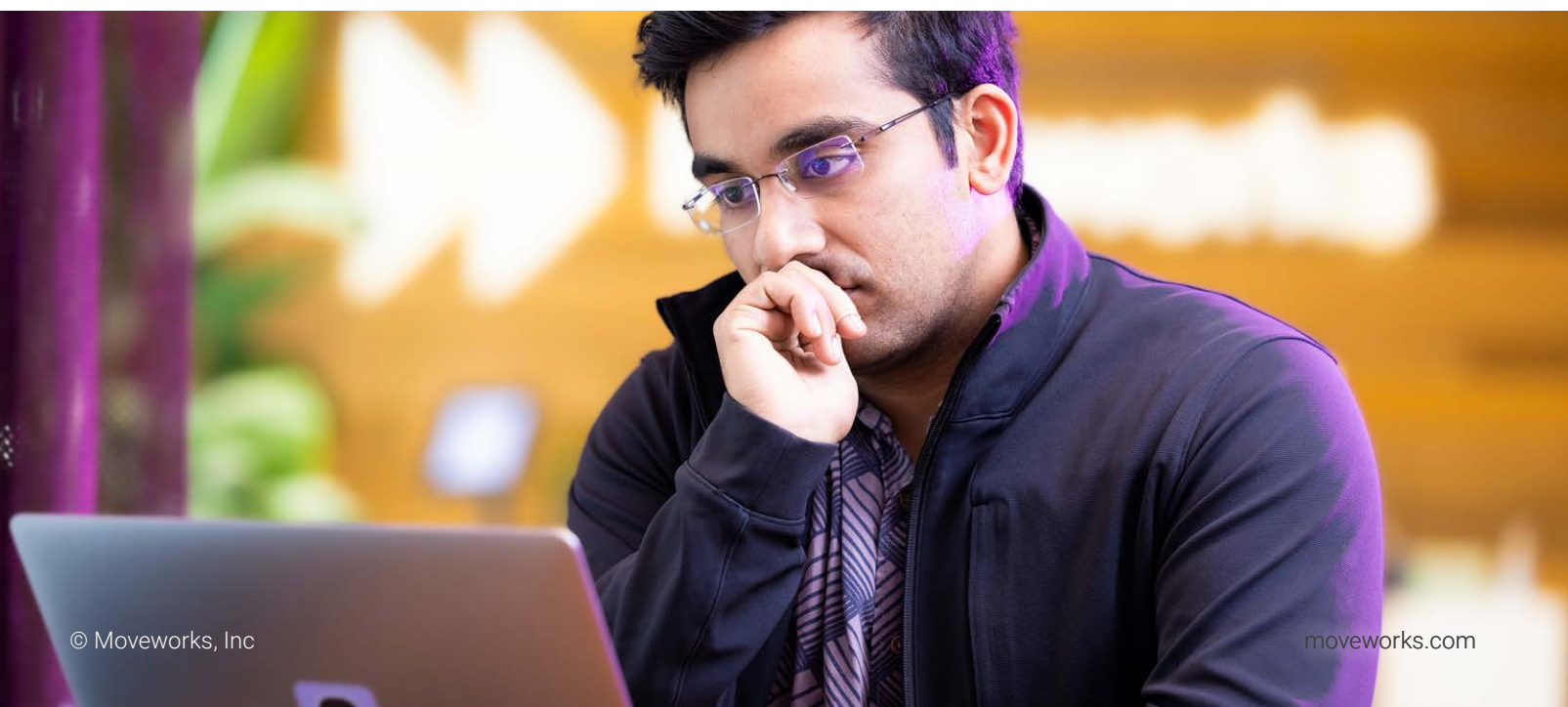
Unleash AI agents with Moveworks
Agentic Automation Engine

The challenges of traditional automation platforms

Tired of the endless cycle of building and managing complex automations and overwhelming developers? As your organization's SaaS landscape grows to 1000+ applications and beyond, the mounting pressure on your technical team to build efficient integrations is only getting stronger.

As a consequence, your business employs experts in the back office to get your processes moving again and relies on developers to fill the gaps by building countless automations. However, using traditional API-driven methods and iPaaS platforms to build automations is time-consuming, cumbersome, and just not cutting it.

Today's automation tools rely heavily on API-driven processes. While this works well for predictable, structured tasks, these platforms may falter when it comes to handling the dynamic nature of human language. While traditional iPaaS platforms excel in configuring predefined workflows, they may fall short in the era of AI agents. In short, these traditional automation approaches are slow, complex and often more time-consuming.



The rise of agentic AI

In this regard, **AI agents present a new automation paradigm.** Unlike traditional SaaS applications which expect users to discover and navigate interfaces, AI agents use large language models (LLMs) to perform automations for users.

A user simply describes the task they want to accomplish, and an AI agent is able to

understand the goal, develop a plan, and make relevant tool calls (aka function calls) to accomplish the user’s task. Such systems are often called agentic AI systems — as defined by the reasoning, planning, and tool-calling abilities exhibited by these systems.

If successful, agentic AI systems are able to make businesses significantly more efficient and employees more productive.

Traditional Applications	Agentic Applications	Business Outcomes
Developers build portals & hope employees find intake forms.	AI agents find the right automation based on the user’s need.	Downsize support staff required to navigate employees to resources
Employees have to go to sift through emails & portals to find pending tasks	AI agents aggregate a single view of all tasks	Fewer tasks are dropped or ignored
Developers build a user interface for every type of task	AI agents can dynamically generate the perfect interface for each task	Higher quality, more informed decisions are made
Developers pre-build & maintain static workflows.	AI agents combine smaller automations to fit user needs on-the-fly	Less developer time spent on editing workflows.

However, despite the advances in reasoning abilities of LLMs, **we have not yet seen agentic AI deliver against this vision.**

02

Agentic automation – A new class of tools

Agentic Automation brings a revolutionary set of tools that empower AI agents to bridge the gap between human language and structured system APIs. This platform allows developers to create AI agents that are able to understand vague instructions and transform them into more precise actions.

An AI agent needs to be able to reason, plan, and act.

The world's leading LLMs like OpenAI's o1, Anthropic's Claude 3.5, Google's Gemini Pro 1.5 are increasingly more powerful (larger context windows, better reasoning, more modalities). This enables reasoning and planning to scale, but [there's more to building AI agents](#).

Some developers mistakenly believe that LLMs could directly interact with business systems using the same APIs that other systems do.

While APIs let you interact with business systems and allow these systems to communicate, they aren't an integration in themselves.

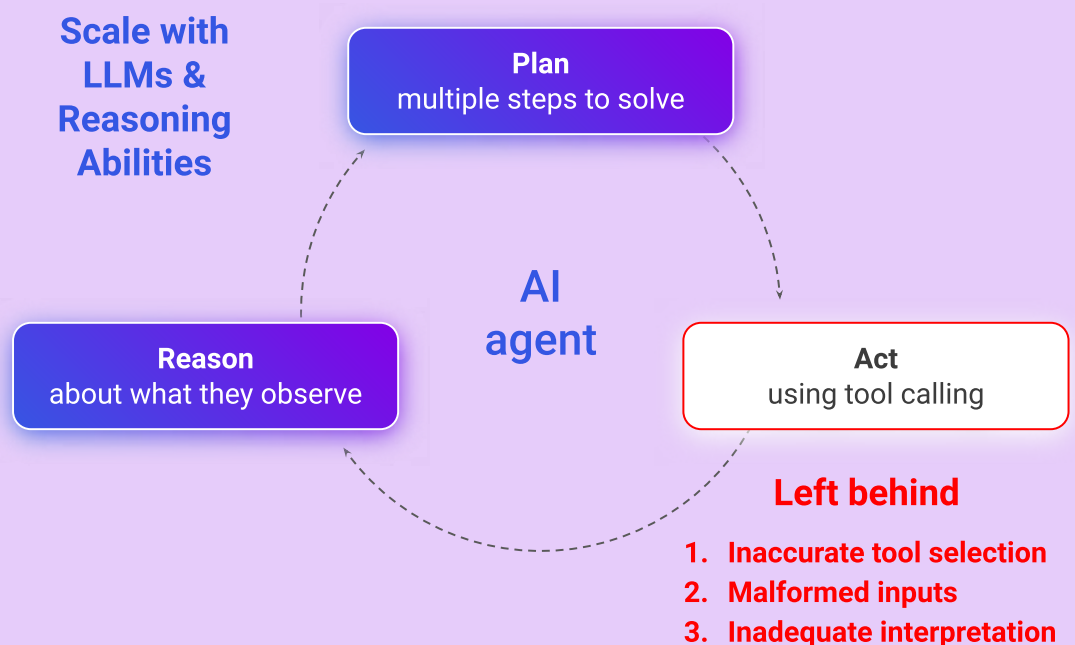
Every developer who has integrated with APIs before can probably understand that integrations need a little more than just an API – code needs to be written somewhere.

The Integration Platform-as-a-Service (or iPaaS) industry was born to solve this problem. iPaaS tools today (Zapier, Workato, Microsoft Power Automate, ServiceNow Flow Designer, Salesforce Flow Builder, Mulesoft, etc.) allow developers to write code that connects APIs across business systems.

The reality is that existing APIs and integration platforms were hardly designed for use by LLM-based AI agents.

iPaaS and middleware were designed to connect APIs (which are precise and deterministic) to other APIs. **But AI agents need to connect human language (which is imprecise and ambiguous) to APIs.**

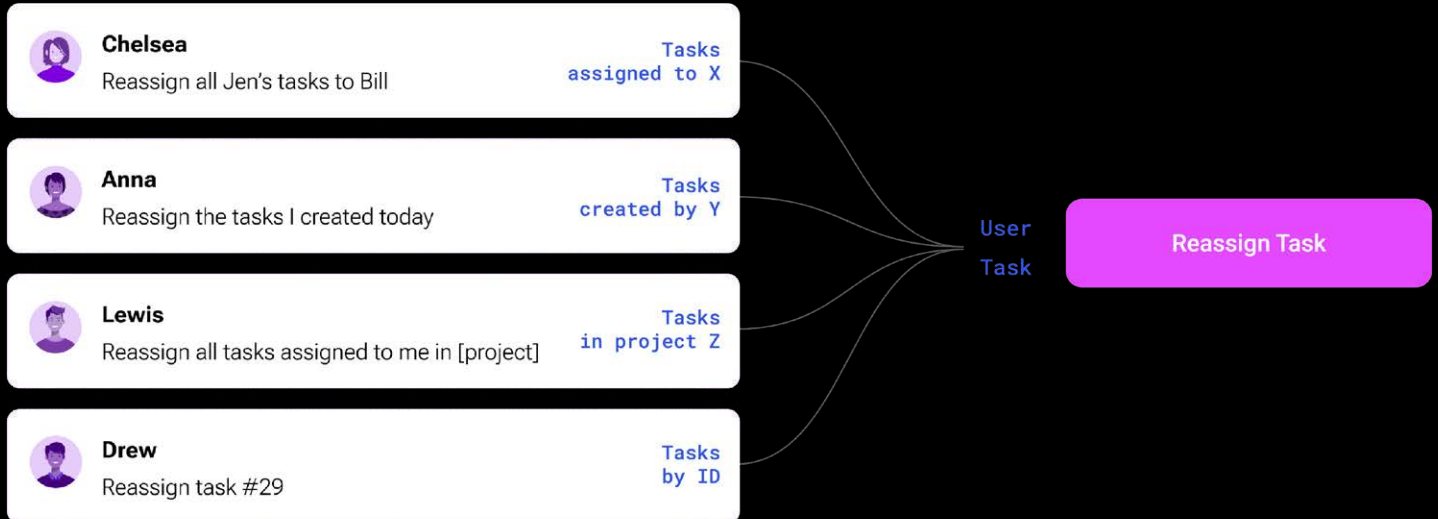
How AI agents work



Let's take a simple example.

Say I want an agent that is able to help me manage tasks (create new tasks, update existing tasks, reassign tasks, etc.). To build such an agent, you can imagine that we need to expose a **Reassign Task API**. Which needs (1) a task ID and (2) a user ID to assign it to.

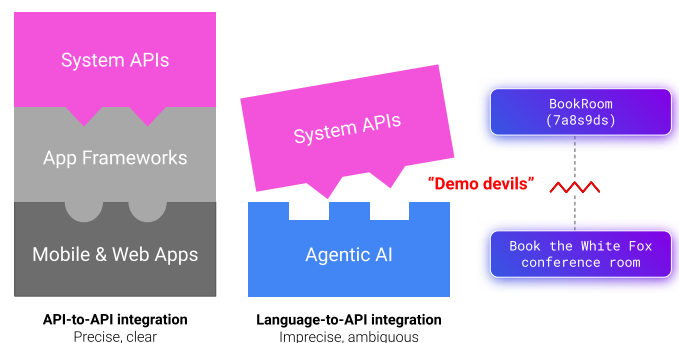
But human language is messy. There are dozens of ways that your users might specify these inputs:



Limitations like these are why REST APIs and traditional automation platforms were not built to be used by AI agents. They may often lack critical capabilities required to overcome challenges with inaccurate tool selection, malformed inputs, and incoherent responses.

Without the right set of primitives, developers are forced to make iPaaS' round peg fit into agentic AI's square hole. As a result, developers may try to patch the gap with an inordinate amount of excess, unmaintainable, unscalable code (tech debt), just to produce inferior results.

That whitespace between System APIs and agentic AI systems represent the "not quite ready" reality of developing AI agents today. This is where the Demo Devils live.



When it comes time to deploy an AI agent to hundreds or thousands of employees, “It worked in my sandbox” isn’t a valid answer when things go wrong. The investments that businesses are making in AI agents seldom see the light of day – they often remain trapped as science projects and demos because they lack the robustness and maturity to be confidently deployed across the business.

In short, trying to build an agentic AI system with iPaaS tools often fails to deliver on the desired agentic capabilities. Further, this development process may challenge even advanced technical teams and consume significant resources, diverting attention away from your strategic initiatives.

Navigating beyond iPaaS constraints involves adopting platforms engineered to address the complex needs of agentic AI. This means

prioritizing tools that grant developers the freedom [to intricately design](#) and refine AI systems with advanced logic and autonomy. The whitespace is the reason why AI agents have yet to proliferate. AI agents may not reach their full potential while they are limited by traditional automation platforms.

Agentic AI automation: A path forward

Agentic automation is an approach to creating AI agents designed to enable seamless integrations between business platforms and enterprise copilots, overcoming the limitations of traditional automation tools. It enables direct conversation-to-API translations, utilizes advanced reasoning engines, and accesses vital data, paving the way for the development of smarter, more responsive AI systems.

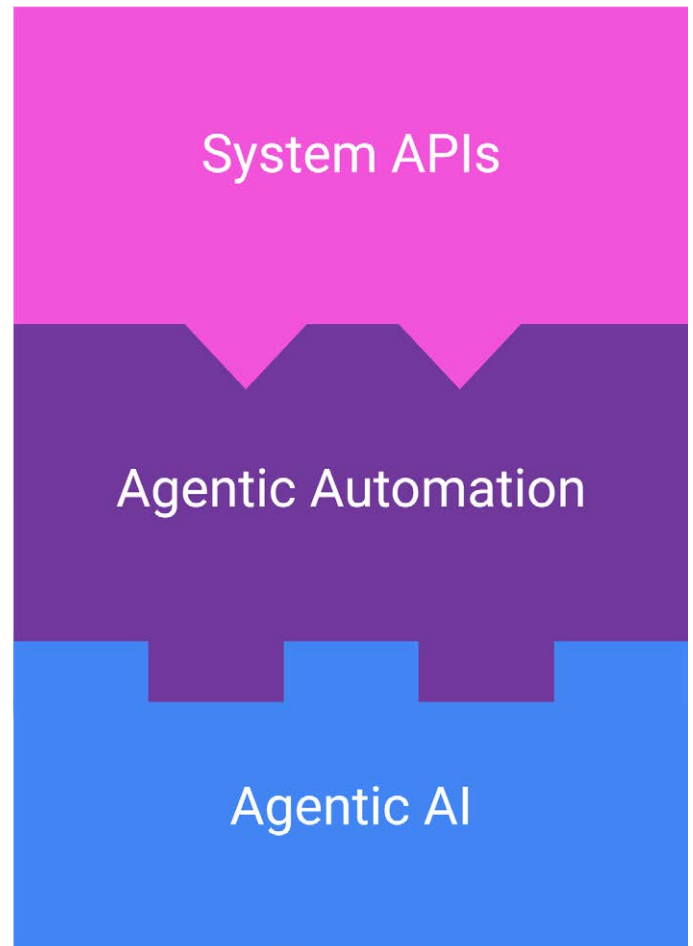


Unleash AI agents with Moveworks Agentic Automation Engine

To empower developers to break through the limitations of traditional automation platforms, Moveworks has designed a new Agentic Automation Engine. By enabling developers to build powerful AI agents with a fraction of the code, this engine unlocks a new level of efficiency, transforming how businesses automate complex workflows.

The Agentic Automation Engine is engineered from its core to build automations that are able to bridge language from agentic AI and structured data from business systems. This Agentic Automation Engine is the new heart of [Creator Studio](#), where developers build AI agents on the Moveworks platform.

With this, developers are able to build powerful AI agents that are able to handle more complex tasks than AI agents powered by a traditional automation engine. Developers should be able to build AI agents faster, with as little as a tenth of the code compared to traditional approaches. Best of all, developers are able to centralize all development in one platform, eliminating code sprawl and simplifying deployment practices.



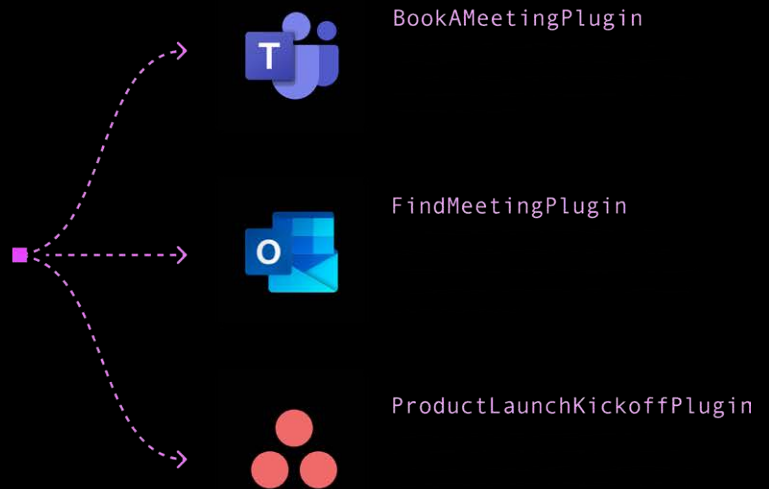
Powered by the Agentic Automation Engine, developers are able to build plugins which the Moveworks Copilot uses. The Agentic Automation Engine comes packed with some key features:

Manifest Generator

A novel approach to interfacing with LLMs that goes beyond just prompts and utterances. It analyzes plugins (agentic automations), generates a precise manifest of plugins, and dynamically adjusts prompts to help the AI agent understand the nuances between plugins and how to map them to user requests more effectively.

Before this, developers had to learn how to do prompt engineering and prompt tuning to get the desired behavior.

Can you find 30min early next week in Mountain View HQ to review the launch plan



Slot Resolvers

A new technique that empowers agentic AI systems to convert natural language into reliable, API-friendly IDs. For example, it is able to convert "the ACME account" into its Salesforce ID "00ORd0000040V9ZMAU". This allows AI agents to correctly identify business objects and execute tasks successfully.

Before this, developers had to try to parse text and manipulate strings inside their iPaaS which was unreliable and led to unexpected results.

"me"	<person>_Ava Chang
"my boss"	<person>_Diana Icard
"early next week"	<date>_20241007-20241010
"Mountain View"	<OfficeSpace>_MTV Office
"Rajat"	<person>_Rajat Patel

Policy Validators

LLMs manage ambiguity well, but their instructions may be misunderstood, or worse, jailbroken. Business rules are rules. You don't want anyone booking a 20-person room for a 2-person meeting. You don't want employees submitting expenses over \$100 without a business justification. Our policy validators are able to add an additional layer into our Reasoning Engine that enforces these policies — no matter how many you add.

Before this, developers would try to add instructions in natural language, but the model would often misinterpret their instructions or fail to enforce it properly.

Policy validators help to enforce your specific policies



Meetings with execs require a meeting agenda	✓
Exec meetings must be scheduled 4 hours in advance	✓
Special rooms can not be reserved for more than 2 hours	✓
Meetings with virtual employees need a room with working VC	✗

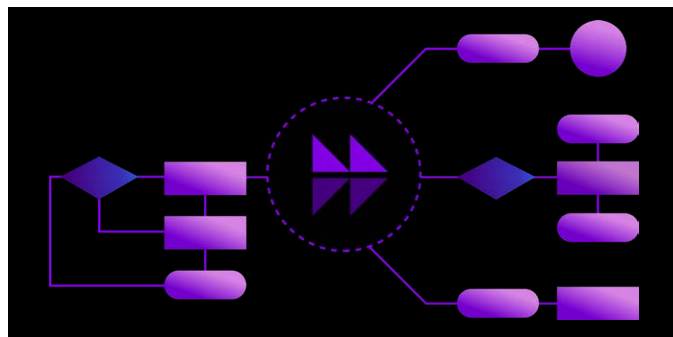
Action Orchestrator

A deeply-connected integration technology that informs the Reasoning Engine about the actions that are planned, in-progress, and complete.

This approach lets developers focus on building a working prototype, and lets our Reasoning Engine handle all dialog generation, edge cases, and failures.

Before this, developers had to build all the error handling logic and come up with custom dialogs for every scenario.

To learn more about these incredible technologies and how they stand apart from their traditional automation counterparts, [dig into our whitepaper here.](#)



How the Moveworks Reasoning Engine supercharges productivity with Agentic AI

The field of AI has long grappled with a fundamental question: Can machines truly reason? Recent advancements in agentic AI are providing compelling answers. Unlike traditional AI systems that follow predefined rules, agentic AI is designed to think and act autonomously. At the core of this approach is the reasoning engine — a sophisticated AI system that goes beyond data processing to tackle complex challenges through logical analysis.

A reasoning engine enables AI to understand problems, develop strategies, and execute solutions independently. This capability is transforming how businesses approach task automation and problem-solving. In this blog, we'll explore the mechanics of reasoning engines, their role in agentic AI, and how they're changing the landscape of enterprise operations.

What is a reasoning engine?

A reasoning engine is an AI system that understands a user's objective, develops a plan to achieve the objective, executes function calls according to its plan, evaluates the success of this execution, iterates on the plan, till it successfully achieves its original objective. In some ways, it attempts to mimic how human beings solve problems.

In practical terms, a reasoning engine allows the system to dynamically figure out how to help a user, rather than follow a set of pre-defined rules or decision trees. It serves as the brain of AI assistants.



```
raise Exception("Input language could not be determined")
return None
parsedInput = self.parseInputToLanguageModel(inputString, inputLanguage, context)
if not parsedInput or not self.model:
    return None
context.append(parsedInput) # Add new conversation entry to context
return (self.model.generateLLMOutput(parsedInput), context)

def parseInputToLanguageModel(inputString, inputLanguage, context):
    if self.model is None or self.model.language != inputLanguage:
        # LLM is not initialised or has wrong language, load LLM
        self.model = self.loadAILanguageModelFromDatabase(inputLanguage)
```



Understanding agentic reasoning

Agentic AI is an AI system that can act via autonomous “agents” to achieve complex goals on behalf of users. In this framework, a combination of ML models is used to create an AI that can independently create and execute a reasoning-based plan that can achieve the user’s goal instead of being limited to simply following programmed logic or dialog flows.

An agentic AI Copilot that is able to reason is much more capable of helping employees throughout the day with a plethora of tasks. Similar to the role a personal assistant would play, this AI, too, has the capacity to understand challenges and problem-solve on your behalf. State the challenge you’re facing and Copilot can do the diligence and come up with the best solution for the situation — that’s the value of reasoning in action.

The Moveworks Reasoning Engine

A reasoning engine powers the AI to meaningfully achieve an employee’s goals. It does so with abilities to:

- Understand a user’s objective
- Develop a plan to achieve that objective
- Execute function calls according to its plan
- Evaluate the success of this execution
- Iterate on the plan until it successfully achieves its original objective

Importantly, a reasoning engine is able to check back with the user wherever necessary to seek additional context about the goal, confirm its direction, and take user input for adapting the plan on the fly.

In some ways, it attempts to mimic how human beings solve problems, which is why having a [reasoning Copilot](#) is so powerful.

The Moveworks Copilot's reasoning engine is an [agent-based AI system](#) that showcases many of these human-like problem solving capabilities. Our engine uses reasoning, decision-making and plugin calling capabilities of multiple LLMs to accomplish specific elements of the bullets we touched on above.

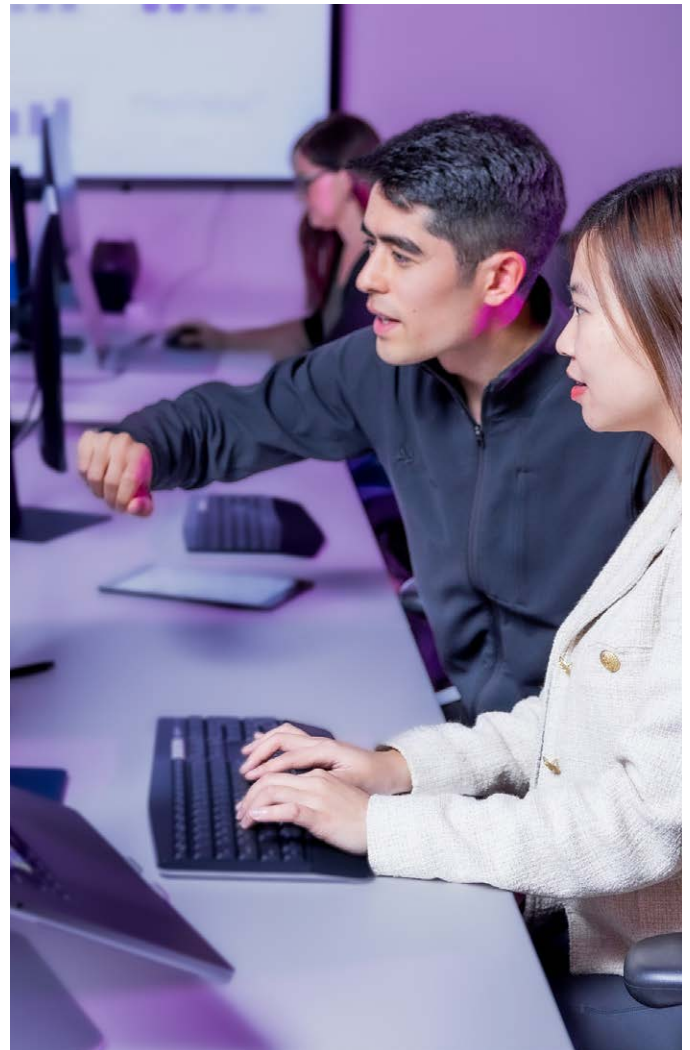
And from a universal support perspective, our reasoning engine scales to reliably operate within the specifics of any of our customers' unique environments (i.e. system integrations, available plugins, business logic, processes, etc.).

Success for the Moveworks Copilot means our reasoning engine is able to create business-specific plans using a distinct set of available tools and processes, and consistently execute whatever the task may be in a goal oriented way—for millions of users across the world, and all in real-time. Let's talk further about how that happens.

Reasoning in action

When users interact with Copilot, their queries, questions, or conversations are run through a reasoning process that informs how it responds back to them. For easy understanding, we can think of that process unfolding in three distinct steps: understanding, planning, and executing.

In each step, we've fine-tuned the Reasoning Engine's various LLMs to apply the best reasoning and decision-making techniques. This approach aims for each employee interaction to be met with a thoughtful, high-quality response.



Understanding user needs

The first stage, understanding, begins as soon as a query is received. The first step is to enrich the text of the query with extracted metadata that serve as critical signals for the reasoning process. The reasoning engine uses a collection of machine learning models, including LLMs, to determine properties of the utterance such as topic, intent, sentiment, domain and language.

Next, it establishes the query in the specific business context of the customer by leveraging the Moveworks knowledge graph to map entities mentioned by the user to their standard or canonical forms. This ensures that all the components of the Moveworks Copilot are able to understand the variations of how an entity is referenced, including very unique and organization-specific entities that only exist within the customer's environment. In this stage, the reasoning engine also checks for whether the query is toxic or offensive and prevents such requests from being served and entering the system.

The next important piece of understanding is conversational context. The reasoning engine incorporates previous interactions between the user and the bot, which provide a rich vein of situational information to help identify the most effective solution. This includes identifying whether the user is starting a new topic or continuing a previous one, deciphering implicit references such as "it," "that," and so on to

understand follow-up questions and to check what other resources and options were previously offered to the user.

From the combination of all of these inputs, the reasoning engine rewrites the user's query and is able to piece together a complete and well-defined problem statement to solve for.

Planning for success

The next step is planning a course of action. This is where the reasoning engine maps its understanding of the problem and the prior context to select the best plugins or tools to serve the request. You can think of plugins as specialized tools custom-built to handle specific use cases and integrate with certain systems.

For example, plugins may reset passwords in Okta, provision users in Google Workspace, onboard staff in BambooHR, manage access in AccessHub, deploy servers, update Salesforce, pull reports from analytics platforms, and sync product data between ERP systems.

The list goes on and on. Each organization has a unique set of plugins, based on their chosen configuration and the custom plugins they have developed using the Creator Studio extensibility platform.

The reasoning engine reviews the list of available plugins and quickly filters the plugin catalog down to the most relevant options with the previously extracted information.

It then enters what is called a plan-evaluate loop. In this loop, the reasoning engine uses its planning module in conjunction with an autonomous evaluator module.

The planner and evaluator rapidly iterate on approaches using Chain-of-Thought to break down the problem using the available plugins, and this independent iteration produces a high quality first pass that eliminates wasted effort and unnecessary burden on the user to guide the reasoning engine. From a performance perspective, we have also optimized this to be very low latency to ensure that the plan can be charted out quickly.

But how do we know that a given plan makes sense in the business context? This is enforced with the use of policies in the reasoning engine, which provide guidance on how to think about the problem in the context of employee support, and to lay out the standard objectives and best principles for how an employee support assistant should work. This imbues the reasoning with a form of grounding that focuses

the solution space and allows the planner and evaluator a way to determine the quality and efficacy of a candidate plan.

Executing with purpose

The reasoning engine is now ready to start executing its identified plan. Just like a human problem solver, it is able to attack the problem but pays close attention to the outputs at each stop, so that it can adapt the plan, if required.

It invokes the plugin executor component, which is responsible for making the appropriate function calls to each plugin selected in the plan and for monitoring the responses sent back. As the plugins perform their specific tasks, they return results or error messages, which the plugin executor analyzes and sends to the reasoning engine, which determines what to do next.

At every juncture, Copilot analyzes the latest available information and decides whether to summarize the output as a final response, or to ask the user for more information, or to try



another approach by calling other plugins. This iterative approach ensures that the reasoning engine dynamically tries to solve the problem in a variety of ways without any human intervention. This flexibility is built into its core reasoning approach, mimicking how humans would handle similar situations.

As an example, for a complex IT troubleshooting scenario, the plan could involve using search plugins that scour knowledge base articles or files, or serve pertinent support forms, or workflows that can guide the user through a designed resolution process. Therefore, we can see that the reasoning and planning processes work in lockstep to remain focused on the user's goal and find the fastest path towards providing a meaningful solution.

All these steps are taken by the reasoning engine and plugins in a seamless, autonomous way, in a matter of seconds. The reasoning engine produces user-facing responses with a concise, referenced summary of the most relevant findings. What's more, at every step the reasoning engine continues to provide the user with reasoning steps for explainability, which also helps the user understand what steps it has tried, and how the response was generated.

If none of the available plugins can help the user, Copilot enables the user to easily seek human support by filing a ticket or speaking with a live agent.



Better reasoning, better support

The Moveworks Copilot puts powerful AI directly at the fingertips of employees to aid them in their day to day. The Reasoning Engine is able to perform similar diligence as a human agent when it comes to comprehending a user's issue, developing a plan to fix it, and executing the appropriate actions to do so with the same degrees of success—only AI can do it faster, for cheaper, and around the clock.

The next era of automation

These capabilities are available to all Moveworks customers at no additional cost.

This release brings forward our vision for Moveworks as the platform for AI agents. When combined with the industry-leading Agentic Reasoning Engine, this new Agentic Automation Engine supercharges a developer's ability to bring new agents to life. Every persona, every SaaS application, and every line of business deserves to have an AI agent powered by Moveworks. Further, this release eliminates the need for traditional automation platforms when building AI agents with Moveworks.

As we look to the future, we look forward to seeing our developer ecosystem multiply manifold and truly deliver what enterprises need most: one AI agent for every business system.

Agentic automation: The future of automation

Agentic Automation is not just a new tool—it represents a paradigm shift in how businesses interact with technology. By leveraging the advanced reasoning capabilities of AI agents, businesses may achieve levels of efficiency and accuracy that were previously unimaginable.

The Moveworks Agentic Automation Engine empowers developers to bring powerful, complex AI agents to life, with minimal effort and maximum flexibility. As more enterprises adopt this technology, we are witnessing the dawn of a new era in automation—one where businesses no longer need to navigate rigid systems, but are able to instead speak their needs into existence.

Discover how Agentic Automation can revolutionize your business operations

Download our [white paper](#) or watch our pre-recorded [webinar](#) to learn more about:

- How Agentic Automation leverages AI agents to achieve unprecedented levels of efficiency and accuracy.
- Why traditional integration tools fall short in building intelligent, adaptable AI agents.
- The key components of the Moveworks Agentic Automation Engine.
- How to build and deploy powerful AI agents across your enterprise applications.

Don't miss this opportunity to be at the forefront of the automation revolution.

