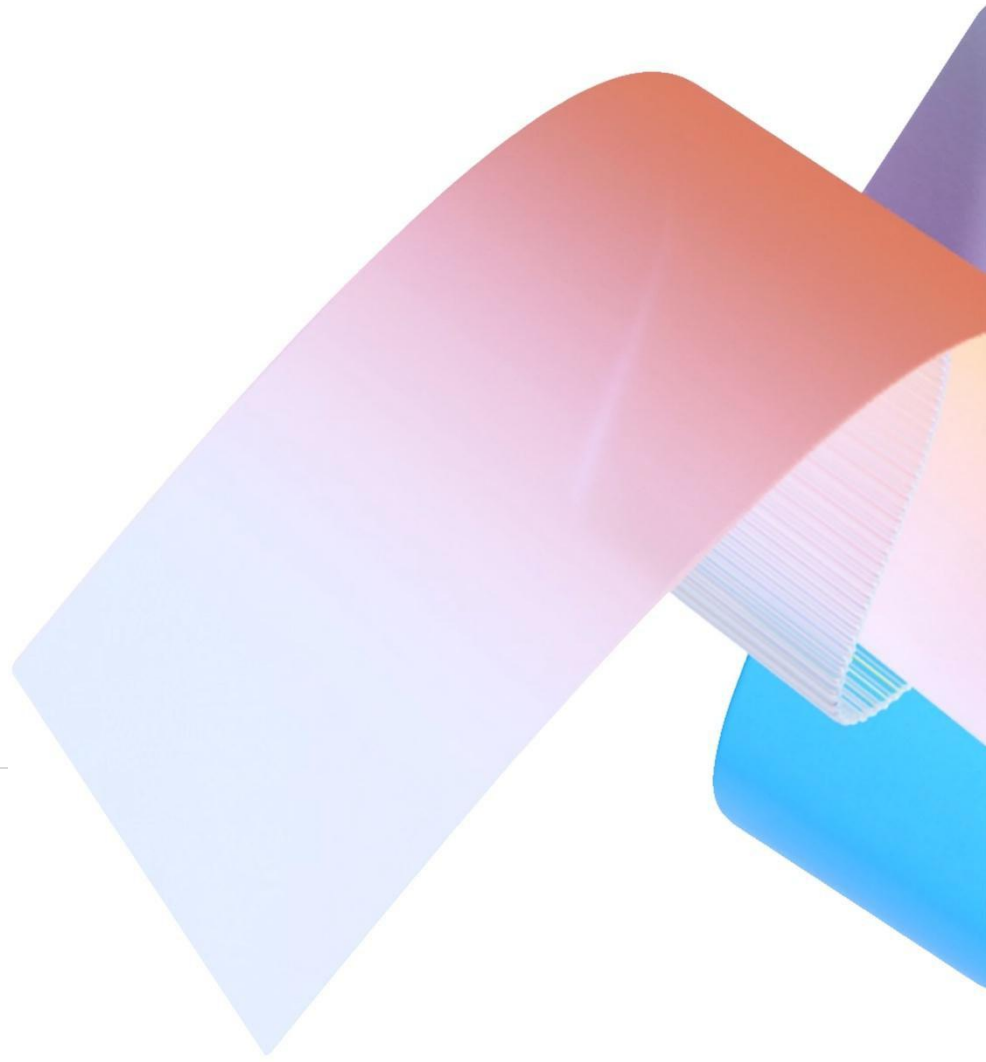




How to Build Agents with Microsoft: An 8-Step Framework



Tear Sheet: 8 Steps to Success

<p>Step 1 Define the business problem</p>	<ul style="list-style-type: none"> Choose a process and identify goals; prioritize existing workflows with organizational readiness for adoption Determine preliminary governance requirements such as agent role boundaries, misbehavior response, and safe and accurate data retrieval Define evaluations and before-and-after ROI metrics
<p>Step 2 List the data needed</p>	<ul style="list-style-type: none"> Identify data sources, systems, and connectors Assess data quality and completeness Plan for human-in-the-loop based on quality and risk; automate later
<p>Step 3 Identify the workflow steps and agents</p>	<ul style="list-style-type: none"> Map the end to end process spine: triggers, steps, user channels, decisions, handoffs, and failure points Assign agentic vs. deterministic roles Prefer modular steps/agents
<p>Step 4 Decide the interaction patterns</p>	<ul style="list-style-type: none"> Pick the pattern: conversational, workflow automation, or proactive Set user/agent engagement rules: when to act, ask, or escalate Choose channels based on user needs and where they work
<p>Step 5 Choose an identity model</p>	<ul style="list-style-type: none"> Select the identity for runtime permissions and governance Use on-behalf-of (OBO) for assistants acting with user's ID permissions Use dedicated agent accounts for shared agents needing stable, scoped access such as their own mailbox, calendar and/or files
<p>Step 6 Choose tools and build</p>	<ul style="list-style-type: none"> Pick tools by builders' technical skill levels and runtime control needs Build workflow and interactions and attach data/tools, ensuring robust observability, traces, and logs Centralize registration and governance for security and control
<p>Step 7 Test and refine</p>	<ul style="list-style-type: none"> Run scenario-based evaluations Improve in order: instructions, tools/data, workflow, then split agents Iterate for quality and reliability; continually re-test to catch regressions and raise the bar as evaluations improve
<p>Step 8 Deploy, govern, and operate</p>	<ul style="list-style-type: none"> Integrate the agent into business systems and user channels Monitor ROI metrics and compliance Train users on when to trust, review, and adjust outputs; reinforce responsible AI

About this framework

This document provides a conceptual framework and tool guidance for power users, IT teams, and developers building agents on Microsoft platforms. For IT leaders, business stakeholders, and builders seeking an overview of the Microsoft Agent Factory program, please refer to the [white paper](#).

Overview

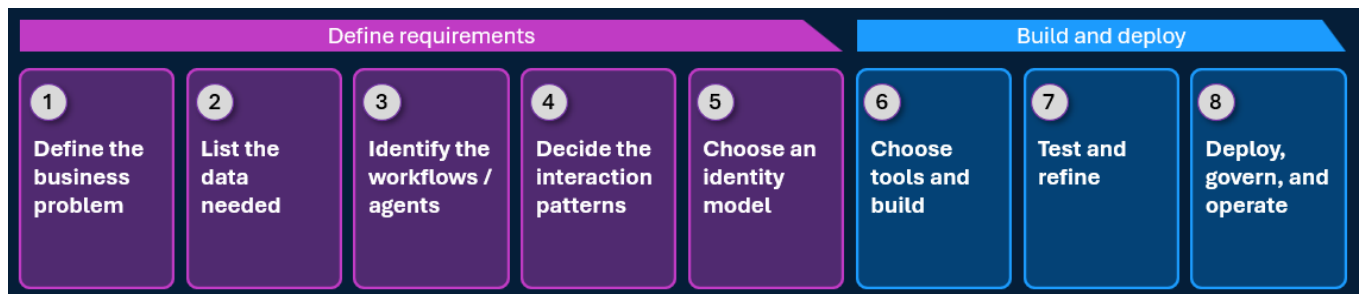
Agents extend organizations' capabilities beyond the limits of human-only speed and scale. They interpret information, make decisions, and create high-quality outputs that previously required significant human effort. Combined with deterministic workflows, they deliver intelligent, predictable business processes—scaling operations, reasoning over more content and variables, and enabling deep personalization.

Organizations are building agents through both bottom-up employee experimentation and top-down deployment by central teams. These agents are expected to act as digital workers: operating autonomously where appropriate, but also working in multi-agent teams, being measured and improved against real business outcomes, and being governed and supervised like any other enterprise worker or system. Scaling from experimentation to production therefore requires not just agent-building tools, but a disciplined end to end operating model.

Microsoft offers a range of tools so builders of various skill levels can build and run agents efficiently and securely—from personal productivity agents with Microsoft 365 Copilot Agent Builder to agents that address business process and department or enterprise-wide scenarios with Microsoft Copilot Studio and Microsoft Foundry. [Microsoft Agent Factory](#) reduces friction and time-to-value for enterprise scenarios with flexible licensing and enterprise support. Microsoft Agent 365 provides central identity and governance for all agents, with explicit controls for data protection, auditability, risk governance, and operational resilience built into every step.

8 Steps to infuse business processes with AI Agents

This guide offers a practical eight step approach to building agents, with an emphasis on design and best practices. Each step includes simple summaries of key design choices and tools. Deeper guides to build, deploy, and govern agents are in the [appendix](#).



Example Scenario: Claims support - a concrete example to ground the 8 steps

This claims support agent scenario blends conversational help, knowledge grounding, and a governed workflow, from the first employee question through submission and approval.

At the center is a **front-door agent** employees can access in Microsoft Teams or via an internal HR web portal. This agent is the single, friendly entry point: it answers questions, collects information, sets expectations, and routes work. On the backend, it coordinates two connected agents purpose-built for other parts of the process:

- **Claims expert agent:** a lightweight, high-accuracy **retrieval-augmented generation (RAG) agent** grounded in official policy and documents. It gives the front-door agent approved, policy-backed answers for questions like “Is this covered?”
- **Claims submission agent:** a rich agent that runs the **end to end submission and decision** workflow. It validates required fields, cross-checks policy guidance from the Claims Expert Agent, performs fraud and risk checks, and enforces guardrails. It auto-approves claims that are low risk and meet clear thresholds. If not, it routes to the right human reviewer with a structured summary, evidence, and a recommendation.

Together, these deliver a modern experience: employees get quick, accurate answers and guided intake, and the business gets a repeatable workflow with the right mix of automation, human oversight, and governance.

Step 1: Define the business problem and how to measure access

The fastest path to value is to start with an existing workflow and ask what has historically prevented automation or scale. Many processes rely on humans to interpret inputs, make decisions, and produce outputs. Agents can address all three. Starting with a well-understood or well-defined process allows organizations to measure clear “before and after” performance indicators as they build ambitious new agent-first processes over time.

Teams should determine:

- a. **Goal:** The outcome to improve, such as reducing time or improving responses/consistency).
- b. **Scope:** The tasks or user goals the agent will handle, such as creating and managing tickets or drafting responses.
- c. **Boundaries:** When the agent should hand off tasks to a human and how the agent should be supervised.
- d. **Basic functional or security requirements:** For instance, standards like safe or grounded responses.
- e. **Success metrics:** Measures of agent and business process success, including baselines for comparison.

To better determine items (c) through (e) above, the team should outline:

- **Governance and audit:** Who can create or share agents, what data and tools agents can access, what policies apply, and the response to agent misbehavior, such as blocking, escalating, quarantining. Agent governance tools should capture traces and transcripts to review incoming requests, agent actions, and final outputs.
- **Evaluations (evals):** Benchmarks for accuracy, reliability, latency/cost, and compliance in real-world conditions, covering common paths, edge cases, and likely or high-risk failures. Determining evaluations up front anchors development in the right goals, informs agent design and trade-offs, and avoids builders compromising on evaluations to fit the agent they've built.
- **ROI measurement:** Business metrics, telemetry, and dashboards with “before and after” baselines to track value over time, such as metrics for customer adoption, time saved, cycle time, cost-to-serve, and quality/satisfaction.

Example Scenario: Claims support

The goals here are to **reduce time-to-submission, speed up low-risk decisions via automatic approval, and improve policy-aligned accuracy**. Evaluations measure grounded accuracy, completion rate, and correct escalation, plus ROI metrics like cycle time, manual touch rate, and cost per claim before vs. after deployment.

Step 2: List the data needed and assess its quality

At this stage, the point is to identify the knowledge and systems a human would need to perform the tasks the agent needs to take on. Assess the quality and accuracy of this data—if it is outdated or incorrect the agent will be as well. This might mean it's best to start with a conversational pattern (so a human reviews the agent's suggestions), and once the data quality improves, and the right guardrails, escalation paths, and ongoing validation are in place, the agent can move to an automated pattern. Then tailor human-in-the-loop (HITL) to the data quality and sensitivity of the agent scenario.

This table lists some tools and data sources, starting with those that have the most out-of-the-box value at the top.

Tool	Description
Web data	Access to internet searches or scoped data from websites.
Copilot connectors	Simple, safe, and governed read access to Microsoft 365 and other sources in Copilot.
Power Platform connectors	Prebuilt read and write connectors for hundreds of enterprise and software as a service (SaaS) systems.
MCP Servers Model Context Protocol	Model- and vendor-agnostic access to tools and data across MCP-compatible agents.
Vector Stores E.g., Azure AI Search	Retrieval index for unstructured content (such as documents and webpages), used for read access when the system has no APIs; ideal for grounding and RAG.

Computer-use agent (CUA)	UI interaction via clicking, typing, and navigating. Used when systems lack APIs and connectors and the workflow needs UI interaction, and in legacy or closed systems.
Code Interpreter	Sandboxed Python environment for structured data analysis (such as spreadsheets, CSV files, logs, and metrics) before taking action or generating output.

Microsoft includes several of these sources in a unique intelligence layer to give agents rich enterprise context. Microsoft **Work IQ** provides Microsoft 365 data with memory and inferencing based on user/team collaboration patterns and organization models. Microsoft **Fabric IQ** provides a single semantic model across OneLake, Power BI, and operational systems so agents can act with a live, connected view of the business. Microsoft **Foundry IQ** creates reusable knowledge bases across sources in a company, via MCP, and the public web, eliminating the need for individual connectors.

Example Scenario: Claims support

Today, a human claims processor looks up coverage in policy documents, checks submission requirements, pulls employee details from HR systems, and reviews past claims and risk signals across multiple tools. For the AI solution, knowledge is accessed in similar ways: **policy and guidance documents are indexed in a retrieval store, and structured systems like HR, claims, and fraud checks are accessed via governed APIs.** This process replaces manual search and copy and paste that a person would do with consistent and auditable knowledge retrieval and system access.

Step 3: Identify the workflow steps and agents needed

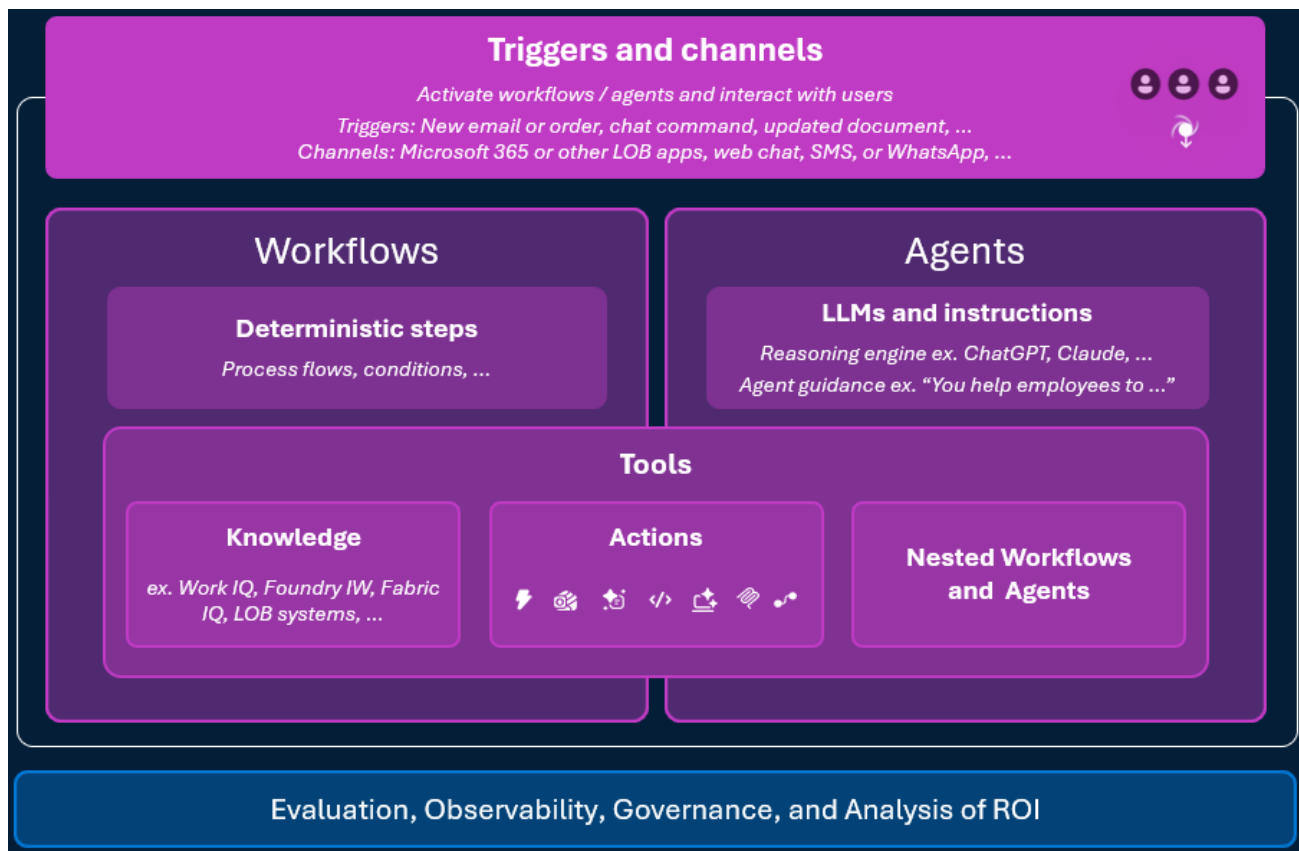
Successful solutions typically start with a deterministic workflow and then introduce agents only where required. Begin by mapping the existing process end to end to create a clear **workflow spine**. Consider:

- What **triggers** the work, such as a user request, event, or schedule
- The major **steps** from intake to completion
- **Systems** that are read from or written to
- **Decision** points, approvals, and handoffs
- Where **errors**, delays, or inconsistency occur today

Next, identify which steps require **agentic capabilities** versus **deterministic logic**. Agents are best

for interpreting unstructured inputs (such as emails, documents, and chat), applying judgment or policy context, or generating content like summaries, recommendations, and explanations. However, they introduce non-deterministic behavior and cost and that is unnecessary for many simple scenarios. Deterministic workflows and simpler indexed retrieval are well suited to static knowledge access or structured and predictable tasks like routing, approvals, and error handling.

Finally, for each workflow step, define inputs and outputs, when agents stop or escalate, and where human touchpoints are required.



It can be tempting to try to address an entire process with one super-agent. However, sophisticated solutions are typically multi-agent systems with **modular workflow steps** and **specialized agent roles**. This modular approach is powerful: it keeps agents simple, makes the process easier to reason over, allows teams to evolve or replace individual pieces, and lets organizations mix agents created by different users and teams. It mirrors how organizations operate through scoped roles and handoffs, so processes can be executed in clear, auditable steps that scale reliably and can be evaluated for quality.

Example Scenario: Claims support

Start by writing down the current claims workflow. Then map each part of that workflow to a clear owner. For example, the **front-door agent** owns intake, status, and handoffs; the **claims expert agent** owns coverage and policy lookups; and the **claims submission agent** owns the end to end submission flow. This keeps the process recognizable to the business, while using modular specialized agents.

Step 4: Decide the interaction patterns and channels

Clarifying an agent's behavior model sets expectations for predictability and required oversight.

Conversational agents act on demand, interacting in natural language. They serve as collaboration partners. Responses may be immediate, like answering policy questions, or span longer-running tasks such as research, analytics, or case handling—but the agent always returns to the end user to share progress or results. Use these agents when humans control and initiate requests, such as for clarification, reasoning, or personalized assistance.

Automation agents are event-driven and operate independently once triggered, involving humans only when needed. Use these agents for tasks that must run end to end, such as classifying or routing items, enriching records, orchestrating multistep business processes, or ensuring compliance. For instance, an agent could extract relevant information from unstructured data to kick off a ticketing process, and apply policies, rules, history, and risk signals to generate decisions or recommendations.

Proactive agents combine conversation and automation to monitor signals, act autonomously, and engage humans at the right time. For instance, a proactive agent could track contract expirations and notify owners with spend summaries and performance insights. Use these agents for timely reaction, such as catching issues before they escalate and generating event summaries. Agents must know whether to act or wait for approval.

The agent channel should be optimized for end users where they already are. Agents can be published to, and triggered by, Microsoft 365 Copilot, Microsoft Teams, email, or business-to-consumer (B2C) channels like web chat, Slack, Twilio, SMS, or WhatsApp.

Example Scenario: Claims support

The **front-door agent** is conversational, answering questions and handholding claim intake in Teams or the HR portal. The **claims submission agent** is autonomous, running end to end when the **front-door agent** hands off a claim, or when triggered by a claim form submission. In either case, it only involves humans when needed.

Step 5: Choose an agent identity model

Every agent should receive an **Agent ID** in a shared directory, which functions as an app identity for registration, discovery, governance, and permissions. A separate **user identity** determines what data agents have access to at runtime and whose permissions it uses for actions.

User identity model	Description and when to use it
On-behalf-of (OBO) Using a User ID	<p>The agent inherits a user's permissions for what it can see or do.</p> <p>Use when: The agent is a personal assistant - reading email, accessing files, or taking actions on their behalf.</p>
Agent User ID Dedicated account	<p>The agent has its own alias, email, files, permissions, and more. This stabilizes access across users and lets users collaborate with agents just like any other user.</p> <p>Use when: The agent supports a team or shared process, and schedules, shares, or collaborates with its own permissions, or if it needs more restricted access than a user.</p>

Example Scenario: Claims support

The **front-door agent**, **claims expert agent**, and **claims submission agent** run **OBO** the employee to retrieve their context and draft a claim using the same permissions as the employee. The **auto-approver** portion of the process uses a **dedicated Agent User ID** so approvals come from a stable, auditable service identity with tightly scoped permissions, rather than from whichever employee initiated the request.

Step 6: Choose tools and build

All agents and workflows, regardless of which Microsoft tool they are built with, use the same underlying architecture described in Step 1. The products can be optimized for users with different technical skill levels, the scope and scale of the business process, and the control and flexibility needed over the runtime. **Many organizations use a mix of solutions across processes or sometimes even within a single process.**

Information worker	Power user	Developer
Agent Builder in Microsoft 365 Copilot, Workflows (Frontier)	Copilot Studio <i>Including Copilot Studio agent starters like Employee Self Service</i>	Microsoft Agent Framework (SDK), Agents Toolkit (VS Code), and Foundry Agent Service

Information workers typically start with **Microsoft 365 Copilot Agent Builder** and **Workflows (Frontier)** for the fastest personal automation. Power users (non-developers in business-led or departmental teams who build shared processes) often start with **Copilot Studio**, with **Employee Self-Service (ESS)** as the quickest path for HR, IT, and workplace scenarios. Teams that are comfortable with a hosted environment and do not need deep control over networking, scaling behavior, or custom runtimes are well suited to **Copilot Studio**. Developers (traditional software developers or data scientists) typically need more control, so **Microsoft Foundry** and pro-code SDKs like **Agents Toolkit** and the **Microsoft Agent Framework** are well-suited for them. For example, with Foundry, agents can run in isolated networks, integrate deeply at the API or data layer, and operate with full DevOps and observability practices.

As mentioned above, most customers will use a mix of solutions: personal agents for bottom-up momentum, Copilot Studio for departmental scale, and Foundry for enterprise-grade services. Agent 365 provides observability for all agents through Microsoft Entra Agent ID, with security and governance by Microsoft Defender, Microsoft Purview, and Microsoft Entra. Refer to the section on [Microsoft's workflow and agent tools](#) for details on these solutions.

With the design decisions made in steps 1-5, building should be straightforward: attach the tools and connectors selected in Step 2, set up the workflow steps and agents from Step 3, and implement the interaction patterns from Step 4. Depending on the product chosen, specialized toolkits can expedite developments like the [Copilot Studio Kit](#) for low-code agents and [Microsoft 365 Agents Toolkit](#) for pro-code agents.

Example Scenario: Claims support

Pick products based on who will **own and operate** each part long-term. The business team that owns the claims policies and documents—who may not be deeply technical—would build and maintain the **claims expert agent** in **Copilot Studio**, so they can manage the knowledge experience directly. The business-critical **claims submission agent** would be built and owned by **developers** in **Foundry**, because it requires advanced workflow logic, deeper integrations, and production-grade operations.

Step 7: Test and refine

Before rollout, agents must show consistent, correct behavior. Build **evaluations** to the specifications outlined in Step 1. If performance falls short, consider refinements in this order:

1. Instructions: Sharpen the role, constraints, and definition of quality.
2. Tools and data: Adjust what the agent can see or do to ensure proper context.
3. Workflow clarity: Add deterministic steps where needed.
4. Multi-agent design: Split responsibilities if an agent is overloaded.

Continuous, rapid test-and-iterate cycles with evaluations after each change can help catch regressions early and improve agent quality. When scores approach 90%, more rigorous evaluations can help maintain steady improvement.

Example Scenario: Claims support

Test each agent against the evaluations defined in Step 1. Run **scenario-based evaluations** for the **claims expert agent** to ensure policy answers are consistently grounded in the indexed documents, and for the **claims submission agent** to ensure it reliably collects required details, applies deterministic approval thresholds correctly, and escalates to humans when risk or ambiguity is present.

Step 8: Deploy, govern, operate, and secure

Deploy agents into existing systems and workflows so they are accessible to the right people and processes. Channels like Microsoft 365 Copilot and Microsoft Teams, which have differentiated data from Microsoft's intelligence layers, are optimized for organization- or department-wide AI scenarios. Govern agents with Agent 365, leveraging its unified control plane for deployment, governance, security, and ongoing operations. Native integration with Microsoft Defender, Microsoft Entra, and Microsoft Purview give organizations central management of agent access and identity, protection against AI-specific threats and sensitive data exposure, and agent auditing to meet security, compliance, and regulatory requirements as they evolve.

Support adoption by helping users understand how the agent fits into their workflows, when agents can be relied on, when human review is required, and how teams can responsibly operate and evolve agent-driven processes.

Example Scenario: Claims support

Deploy the agents into the **channels and systems where claims work already happens** (Microsoft Teams and the HR claims portal), and **operate them as a single governed process**. Ongoing operations would focus on monitoring the same success metrics from Step 1 in addition to operational signals like failures, escalations, and cost, to continuously improve quality and compliance over time.

Next steps

Put these steps into action! Get started by picking a high-impact scenario that has implementation support. For inspiration, review the case studies at the end of this paper that bring the eight step framework to life, and explore where and how other organizations are investing today to transform their processes at aka.ms/frontierfirmsscenarios. These scenarios can be particularly helpful for organizations that are awash in use cases but are struggling to prioritize what to pilot and scale—they represent the top functional AI solutions being pursued today, that have a demonstrated history of being core to successful transformation, and are pressure tested against current and emerging Microsoft AI Business Solutions capabilities.

Microsoft's workflow and agent tools

Personal productivity agent and workflow builders

Microsoft 365 Copilot Agent Builder

Value: An immediate, interactive AI development experience that is perfect for quick and straightforward projects for employees to automate their work. Agents are created directly in Microsoft 365 Copilot and grounded in the user's Microsoft 365 and Work IQ data across emails, documents, meetings, and chats, plus any organizational content that IT enables via Copilot connectors. It requires no coding and delivers personal ROI without IT involvement. It can be shared with teammates and other coworkers.

Use when: Best for one-off or user/team workflows like research, summarization, weekly reports, inbox triage, or project coordination. Also good for prototyping enterprise agents.

Availability: Available to Microsoft 365 Copilot users with access to Work IQ. *Note: Capabilities available for agents differ based on the user's license. For details, see [agent capabilities and licensing models](#).*

Workflows (Frontier) in Microsoft 365 Copilot

Value: A natural-language workflow builder inside Microsoft 365 Copilot that helps users automate repetitive work across Microsoft 365 without manually wiring connectors or designing flows from scratch. Describe the desired outcome, for example, "Every weekday, summarize important emails and post to Teams," and Workflows generates a multistep workflow that can run on a schedule or in response to events, while keeping users in control of the sequence and logic.

Workflows is optimized for personal automation, but is built on the same infrastructure used in Copilot Studio agent flows, bringing enterprise-grade reliability and governance alignment to everyday workflow creation.

Use when: Best for personal automations connecting common Microsoft 365 services (Outlook, Teams, SharePoint, Planner, and Approvals in Microsoft Teams), such as inbox triage, recurring status updates, reminders and follow-ups, lightweight approvals, or signal monitoring workflows that send notifications to users when something changes. Ideal for fast value with minimal setup when the automation is within Microsoft 365's supported actions/connectors.

Availability: Available as a Workflows (Frontier) agent in the [Microsoft 365 Copilot Agent Store](#) for customers in the Frontier program. *Note: currently with limited connectors, a personal automation focus, and only in English.*

Enterprise agent and workflow builders

Enterprise agents built by professional makers in conjunction with their IT partners transform how work gets done by individual employees, across departments or the entire organization.

Microsoft Copilot Studio

Value: A fully managed SaaS environment to build custom agents, workflows, and integrations. [Copilot Studio](#) combines a broad connector ecosystem with conversational design tools, evaluations, analytics, 3P channels, and enterprise-grade governance. No infrastructure to run; everything is hosted.

Use when: Best for departmental and cross-team processes like HR onboarding, finance approvals, customer support triage, sales qualification, and operations workflows - where speed, connectors, tools, and rapid iteration matter more than deep code-level control and data management. Ideal for professional makers, business analysts, and IT admins supporting line-of-business automation.

Availability: Licensed Microsoft 365 Copilot users can access Copilot Studio at no additional cost and use it to build agents that work inside Microsoft 365 services such as Teams, SharePoint, and Outlook. Organizations can fund usage through Copilot Credits purchased as prepaid capacity packs or pay-as-you-go meters; also available through the [Microsoft Agent prepurchase plan \(P3\)](#) for a single metered pool across Copilot Studio and Foundry.

Microsoft Copilot Studio templates

Value: [Templates](#) optimize for specific scenarios and domains and speed up building in Copilot Studio. They are configurable and extensible to accommodate unique needs and preferences. The Employee Self-Service agent is an example of a template for high-traffic HR, IT, and workplace operations scenarios. It provides a secure and centralized experience in Microsoft 365 Copilot for employees to get the answers and actions they need, without navigating to multiple apps or agents. With ESS organizations can access Microsoft documentation that is finely tuned to this domain (ex. typical HR standards and legal requirements), review out-of-the-box dashboards (ex. tickets deflected) and use key capabilities that only light up for that domain (ex. agent handoff to systems like Workday, adding per response disclaimers that are critical for HR).

Use when: Templates are purpose-built for specific scenarios. For the ESS agent, scenarios are scoped to Employee Services like HR, IT, and internal Q&A and fulfillment. Organizations can connect their knowledge, HRIS and IT systems and flexibly configure their internal processes.

Availability: The Employee Self Service (ESS) agent is [generally available](#); after it is configured and enabled by IT in Microsoft 365 Copilot, employees can access the agent if they have a Microsoft 365 Copilot license or by using Copilot Credits. Other Copilot Studio templates are still in preview and vary in maturity and availability.

Microsoft Agents Toolkit, Microsoft Agent Framework, and Microsoft Foundry

Value: [Foundry](#) is the enterprise pro-code platform-as-a-service (PaaS) for agents, providing a full control plane for building, running, and governing deeply integrated, multi-agent systems. It gives developers end to end control over architecture, networking (virtual networks and private endpoints), security, observability, and evaluations, with first-class support for multi-agent orchestration, model flexibility (including multiple providers and fine-tuned models), and rich integration at the API and data layers. [Agents Toolkit](#) and [Agent Framework](#) bring this into familiar .NET and Python workflows so teams can treat agents like any other mission-critical service.

Use when: Choose Foundry for agents that are part of the core application platform, not just a single workflow. It's the right fit when you need to embed agents into existing apps and microservices, run them in tightly controlled or regulated networks, support complex multi-agent topologies, or operate with full CI/CD, environment promotion, and site reliability engineering (SRE) practices. Foundry is the natural next step when Copilot Studio hits limits around runtime control, integration depth, or scale - especially for platform teams that want agents as durable, reusable microservices rather than one-off automations.

Availability: Token and compute-based Azure consumption; also available through the [Microsoft Agent prepurchase plan \(P3\)](#) for a single metered pool across Copilot Studio and Foundry.

Agent operating model and governance

These are common across the agent-building tools described above:

Microsoft Agent Factory: A program designed to help organizations accelerate from pilots to production-grade AI agents. A single plan covers Foundry, Copilot Studio, GitHub Copilot, and Microsoft Fabric, simplifying licensing and removing up-front provisioning so users have flexibility to build and deploy agents across tools with predictable costs and fewer bottlenecks. Role-based training builds employee confidence and adoption, while hands-on support from Microsoft forward deployed engineering (FDE) and partners supports rapid deployments. [Learn more.](#)

Microsoft Agent 365: A unified control plane and single registry for all agents, whether enterprise agents or personal agents, built on Microsoft platforms or third-party platforms. IT admins can govern agents using the same identity, security, and compliance used for people, across Microsoft 365 admin center, Defender, Entra, Intune, and Microsoft Purview. Agent 365 has interoperability with Microsoft 365 Work IQ as well as MCP. It is built into Agent Builder in Microsoft 365 Copilot and Copilot Studio, with an SDK for custom-hosted agents. [Learn more.](#)

Foundry Control Plane: A developer-focused control plane inside Foundry that helps engineering teams build, evaluate, operate, and ship production-ready agents. It provides a unified interface for managing agents, models, and tools; monitoring and optimizing agent quality, cost, performance, and compliance; and applying deterministic guardrails and policies. [Learn more.](#)

Customer case studies across agent solutions

The following case studies illustrate how teams have applied the agent-building methodology described in this paper, moving from manual processes to scalable, AI-powered systems.

Customer: NTT DATA

Business problem: NTT DATA, an IT services and consulting firm with operations in over 50 countries, wanted to help clients automate complex business processes and transform IT operations to improve efficiency, reliability, and customer experience.

Workflow mapping: Broke down the IT service desk process into modular steps with specialized agents assigned to each stage. This approach ensured routine tasks were automated, while complex cases were escalated to human experts, enabling seamless handoffs and operational transparency.

Tool selection: Microsoft Foundry offered scalability, flexibility, and advanced automation and integration.

Agent design: Agents were developed to handle IT service desk requests, automate ticket management, and provide real-time support for both customers and employees. Modular agent architectures allowed for clear communication between agents and defined roles for data retrieval, analysis, and customer interaction.

Governance and measurement: The solution was deployed and monitored for operational efficiency, reliability, and customer satisfaction. Continuous improvement was achieved through faster resolution of service desk requests and reduced manual workload.

[Read the full story](#)

Customer: CSX

Business problem: CSX, one of the largest freight railroad transportation companies in the United States, needed to modernize its supply chain and field operations.

Workflow mapping: Identified manual, time-consuming processes in shipment tracking and field crew support as key pain points. Automation could help employees access information faster and resolve issues more efficiently.

Tool selection: Used Copilot Studio and Microsoft Foundry together to build and deploy agents:

- Built custom agents with Copilot Studio to automate shipment tracking, streamline case

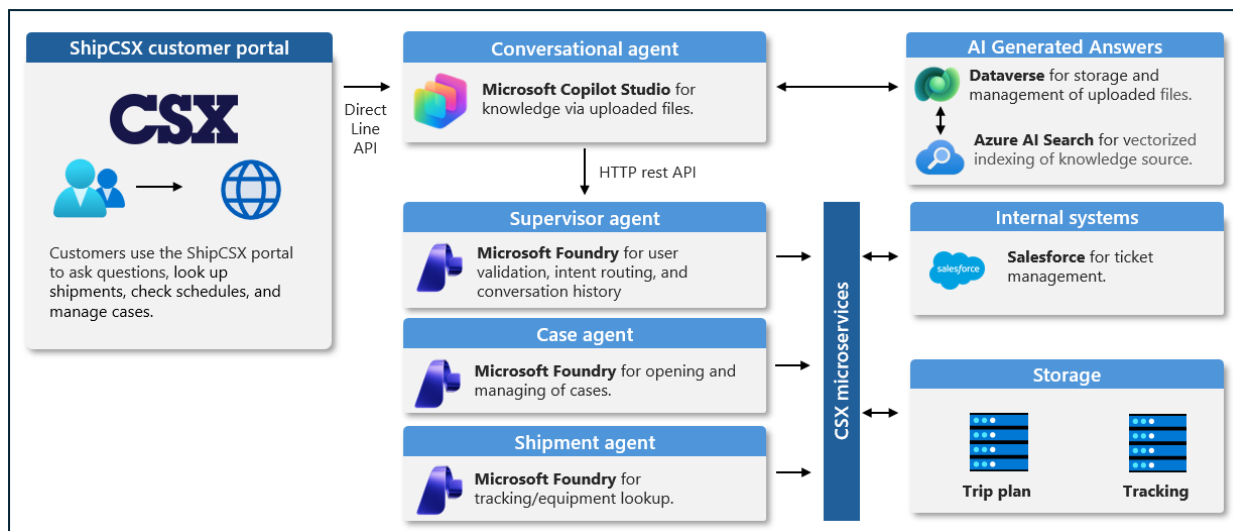
management, and provide real-time support for field operations.

- Created agents with Microsoft Foundry to drive real-time data retrieval through custom logic across systems.

Agent design: Agents were developed to help employees quickly access shipment information, resolve field crew issues, and reduce manual effort in daily workflows. The solution was integrated into CSX's operational environment to support real-time decision-making.

Governance and measurement: Agent adoption leads to measurable business impact, including faster resolution of field crew issues, improved accuracy in shipment tracking, and enhanced operations in supply chain workflows.

[Read the full story](#)



Customer: KPMG

Business Problem: KPMG, a global leader in professional services, specializing in audit, tax, and advisory, needed to modernize its internal operations and client-facing services by leveraging AI to automate routine tasks, enhance decision-making, and drive greater value for both employees and clients. The goal was to reduce manual effort, improve accuracy, and enable scalable, repeatable processes that could adapt to evolving business needs.

Workflow Mapping: Identified areas in key business workflows where agentic automation could deliver the most impact, such as automating document review and data extraction for audit and advisory engagements, streamlining client onboarding and support processes, or integrating agents into existing systems for seamless handoffs and collaboration.

Tool Selection: A mix of Microsoft tools to address varying technical requirements and business scenarios:

- Microsoft 365 Copilot Agent Builder for personal productivity and quick automations within employee workflows
- Copilot Studio for departmental and cross-team processes, enabling business users to build and manage custom agents for HR, finance, and client support
- Foundry for enterprise-grade, pro-code agent development, supporting deep integration, advanced orchestration, and robust governance across mission-critical applications

Agent Design: Agents were designed to operate as digital workers, each with a scoped role:

- Personal assistants for employees, leveraging Microsoft 365 Copilot Agent Builder to automate research, reporting, and coordination tasks
- Departmental agents built in Copilot Studio to handle complex workflows like onboarding, compliance checks, and client interactions
- Enterprise agents developed in Foundry to manage large-scale data processing, multi-agent orchestration, and integration with KPMG's core systems

Governance & Measurement: Adopted Microsoft Agent Factory for disciplined governance and measurable impact.

[Read the full story](#)

Customer: Engie

Business Problem: Engie, a French company with around 100,000 employees and offering low-carbon energy and services, needed an efficient employee solution for IT issues, policy lookups, and facilities requests. Fragmented employee access to information and support was resulting in slow resolutions and increased operational costs.

Workflow Mapping: Mapped high-traffic HR and IT scenarios. The agent is meant to become the centralized entry point for Engie employees to resolve common IT issues, access HR policies, and submit facilities requests—all within the Microsoft 365 Copilot Chat interface.

Tool Selection: ESS was chosen for its customizable template, including domain-specific topic flows, and pre-built workflows for HR and IT. ESS's built-in connectors and workflows eliminated the integration, governance, security, and lifecycle overhead of stitching together multiple agents or building a DIY solution.

Agent Design: A single agent by customizing ESS to integrate data, knowledge, services, and tools. The agent uses connectors to Microsoft 365 content and trusted sources, ensuring accurate and compliant answers.

Governance & Measurement: Deployed and is testing the Employee Self-Service Agent to curate information across multiple HR and IT systems, bringing answers and actions from these systems into a single, secure M365 Copilot experience.

[Read the full story](#)

Additional resources

- [Microsoft Agent Factory](#): Learn more about how Microsoft partners with customers to deliver results with AI.
- [The Microsoft Agent Factory white paper](#): See how Microsoft Agent Factory helps customers scale agents from pilot to production.
- [Microsoft AI use cases](#): Explore real-world use cases to help you accelerate your business goals in the AI era.
- [Microsoft Copilot Studio](#): Explore the SaaS platform for building custom conversational and autonomous agents.
- [Microsoft Foundry Agent Service](#): Explore the platform-as-a-service for agents that provides full control over architecture and orchestration.
- [Microsoft Agent 365](#): Learn more about the centralized control plane for managing agent identity, security, compliance, and telemetry.
- [Responsible AI Principles](#): Discover best practices and guidelines to support ethical and trustworthy AI agent development.
- [AI Agents Hub – Microsoft Adoption](#): Explore the tools available in creating agents and how to get started.
- [Microsoft Cloud Adoption Framework](#): Build a business plan for agents aligned to your company priorities.
- [Microsoft 365 Copilot Extensibility](#): Learn how to build custom Microsoft 365 Copilot experiences.