eBook

Navigating the Era of Agentic AI

Agents

Authors

- Prashant Atri
- Prabhjot Kaur
- Anil Goswami
- Abhishek Bhardwaj
- Hoc Phan

ABOUT THE AUTHORS

Prashant Atri
Strategic Advisor – Data & AI, Microsoft

He focuses on leveraging cutting-edge technologies, including AI and data analytics, to drive customer transformation and large-scale enterprise modernization. He has led multiple cloud migration and transformation projects across various clouds and with large global pharmaceutical customers in highly regulated environments.

His experience spans across multiple industries and global SI/ISV partners. He is passionate about leading ideation projects, establishing structure and framework, driving community growth and authoring content.

His ongoing knowledge sharing and contributions to the community, positions him as a trusted advisor and visionary leader. He has served as a key panelist on several webinars and round table discussions.

Outside of work, he enjoys music, dance, reading and walking.

Hoc Phan has over 20 years of experience in the tech industry, driving innovative products for Amazon, EMC, Dell, and Microsoft. He is also a book author and have written three books covering topics on mobile app development and the Internet of Things, sharing insights and expertise with professionals and enthusiasts alike.

In 2025, Hoc joined Okahu to lead product management and pre-sales, focusing on AI observability. The mission is to help customers understand their large language model (LLM) behavior while enhancing AI app performance, reliability, and quality. It's been an exhilarating journey so far!

Prior to that, he was part of Microsoft's data governance transformation through the BlueTalon acquisition which became Microsoft Purview (Data Catalog and Policy). He led many internal R&D projects including cyber security re-platform and data governance, trained early adopters, and gathered customer insights through various EBC. Hoc has shared his knowledge at many industry events as well such as RSA Conference, Strata Hadoop, Dell EMC World and VMworld, cementing my passion for both technology and community engagement.

Feel free to contact me to learn more or if you have question on **AI observability - Hoc@okahu.ai**

Anil Goswami
Principal Data Scientist – Cummins Inc

Anil Goswami is an accomplished engineering and digital transformation leader with over 20 years of experience in driving innovation at the intersection of data science, AI/ML, and product development. As Principal Data Scientist at Cummins Inc, he specializes in integrating advanced AI/ML solutions to enhance product reliability, optimize manufacturing processes, and accelerate simulation workflows. He has significantly contributed in process efficiency improvement, reduced troubleshooting challenges, and streamlined decision-making through data-driven insights. A mentor and thought leader, he is passionate about leveraging AI to advance engineering practices and shape the future of digital innovation.

Outside of work, he enjoys exploring nature, travel and playing pickleball.

Prabhjot Kaur - Principal Cloud Solution Architect
Partner Success Organization- Microsoft

With an in-depth understanding of industry trends and best practices, she aims to align business goals with IT capabilities by delivering scalable and sustainable solutions that encourage innovation and support long-term organizational success. Her role entails collaborating with partners and customers to comprehend their challenges and devising customized cloud strategies and solutions to enhance performance, security, and cost-efficiency.

She is passionate about writing technical blogs and whitepapers, having co-authored whitepaper and several blogs based on real-world partner engagements. In her free time, she enjoys discussing emerging technologies.

Outside of work, she enjoys cooking, kayaking, and hiking.

Abhishek Bhardwaj
Associate Data Engineer – Publicis Sapient

Abhishek Bhardwaj is an experienced professional with over 10 years in IT, specializing in Data Science, Data Analysis, and Data Engineering. With a Master's in Applied Data Science from Indiana University, he has expertise in AWS services, Python, SQL, and machine learning. Abhishek has designed efficient data pipelines, implemented big data solutions, and created automated workflows.

His technical acumen is complemented by strong communication skills and a passion for solving complex data challenges. Abhishek enjoys exploring cutting-edge technologies, mentoring teams, and continuously learning to stay at the forefront of the data engineering landscape.

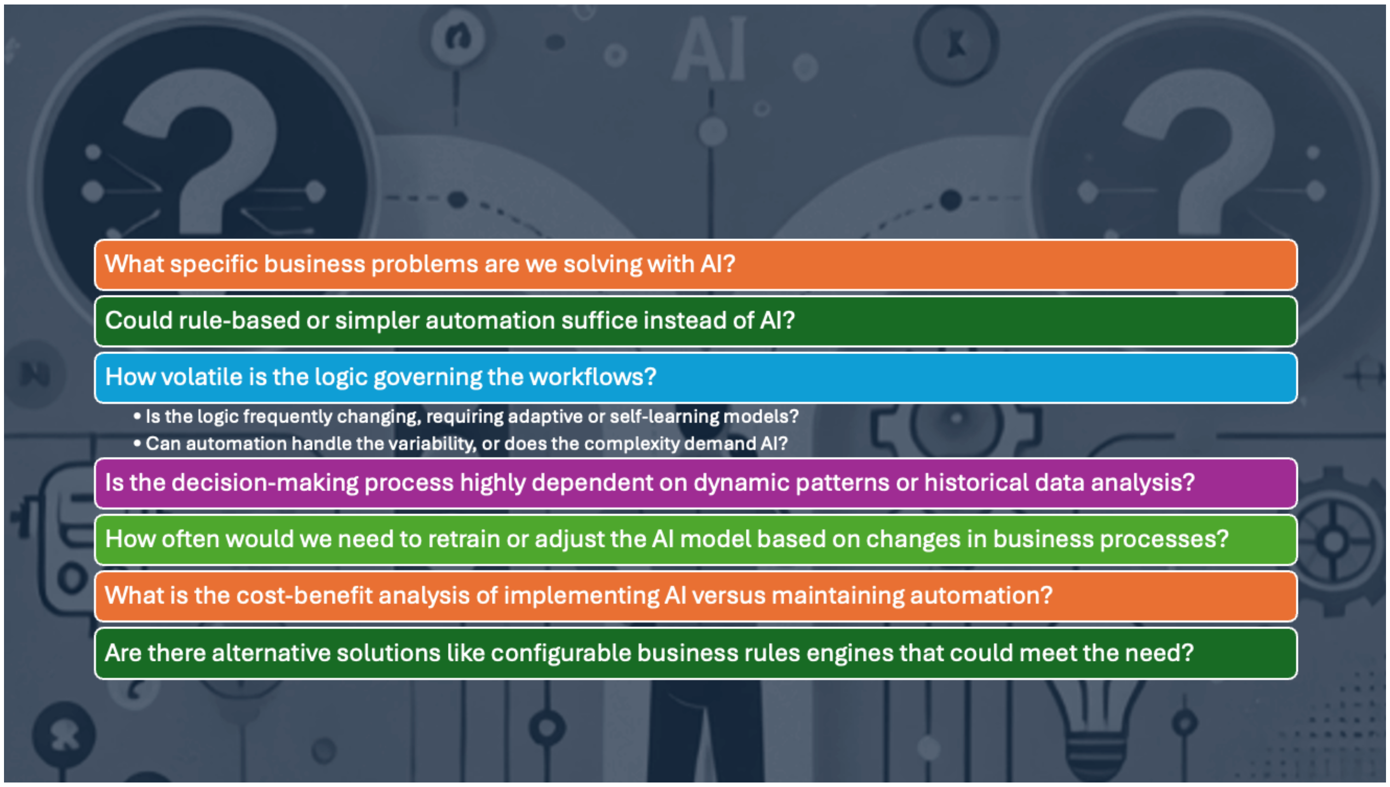
Table of Contents

Introduction

The Generative AI (GenAI) landscape is evolving rapidly with the integration of Large Language Models (LLMs) and advanced prompt engineering. These innovations, when combined with Agent-based architectures, are transforming user experiences by automating complex tasks and enabling actions with minimal human intervention. This synergy extends the capabilities of models to not only generate contextually relevant outputs but also access external knowledge bases and execute decisions autonomously.

In this whitepaper, we will explore the mechanics of Agentic architecture, detailing its foundational principles and illustrating how it simplifies the deployment of GenAI systems. By understanding this architecture, organizations can unlock the full potential of Generative AI to deliver impactful, scalable, and efficient solutions.

Pause and Reflect: Is AI the Right Fit for Your Agent



- What specific business problems are we solving with AI?
- Could rule-based or simpler automation suffice instead of AI?
- How volatile is the logic governing the workflows?
  - Is the logic frequently changing, requiring adaptive or self-learning models?
  - Can automation handle the variability, or does the complexity demand AI?
- Is the decision-making process highly dependent on dynamic patterns or historical data analysis?
- How often would we need to retrain or adjust the AI model based on changes in business processes?
- What is the cost-benefit analysis of implementing AI versus maintaining automation?
- Are there alternative solutions like configurable business rules engines that could meet the need?

The Evolution of Digital Interaction

From Assistants and Chatbots to Copilots and AI Agents

Assistants

Imagine you have a personal assistant who can only follow straightforward instructions, like "Set a reminder" or "What's the weather today?" Assistants like Siri, Alexa, and Google Assistant fall into this category. They are helpful but limited to predefined tasks.

- **Analogy:** Think of a secretary who only answers basic questions or takes notes but doesn't offer advice.
- **Example:** "What's the time in New York?" or "Remind me to call Dad at 6 PM."

Chatbots: Conversational Problem-Solvers

Chatbots were the next step, designed to communicate through text or voice. They were a bit smarter than assistants, capable of having basic conversations and answering specific queries. However, their scope was still limited—they relied on predefined scripts or simple rules.

- **Analogy:** A customer service agent at a help desk who answers frequently asked questions but struggles if you ask something unexpected.
- **Example:** Chatbots on e-commerce websites that answer, "Where's my order?" or provide product recommendations

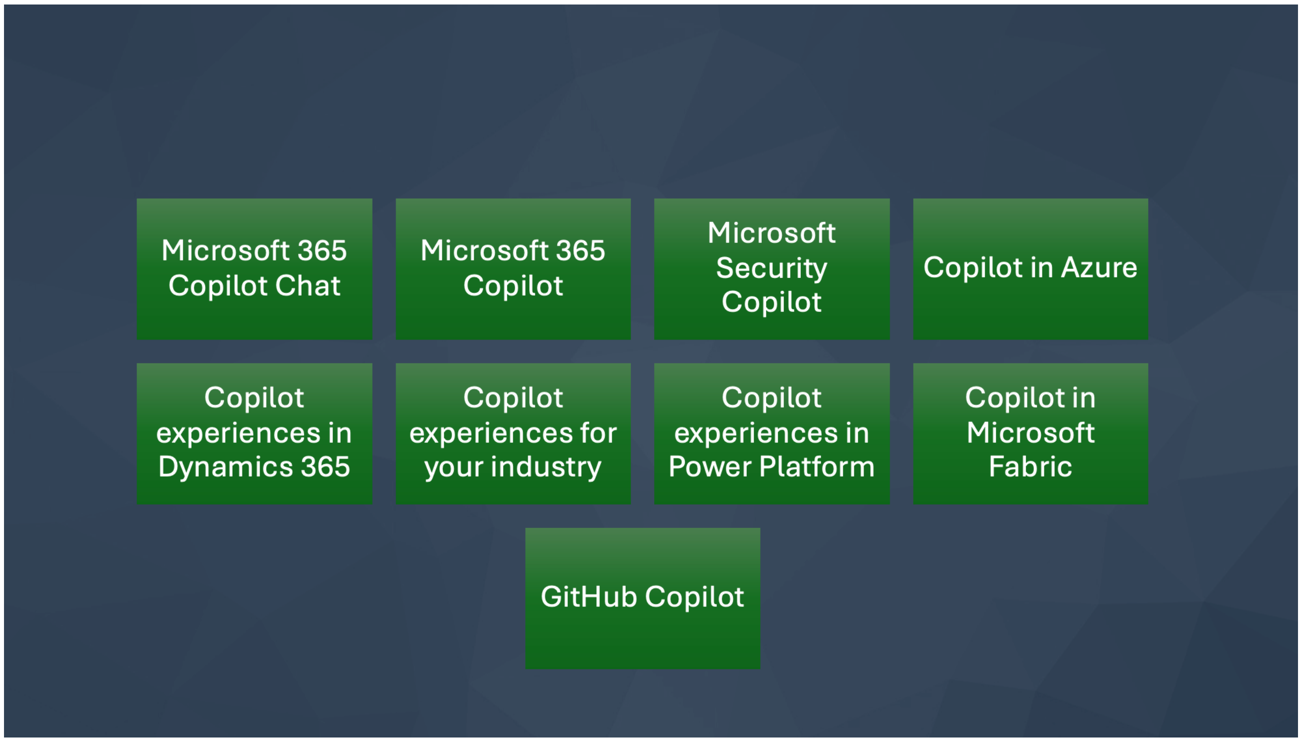
Copilots: Working *With* You

Copilots take things a step further. Instead of just following instructions or answering questions, they actively assist users in accomplishing complex tasks. They work *alongside* you like a teammate, providing real-time suggestions, recommendations, and even solutions.

- **Analogy:** A co-driver in a car rally who not only tells you the route but also provides tips on how to navigate tricky turns and avoid obstacles.
- **Examples:**
  - **Email Writing:** When drafting an email, a copilot might suggest better phrases, find attachments, or summarize relevant information from past emails.
  - **Coding Assistance:** Developers use copilots like GitHub Copilot to get suggestions for writing or debugging code.
  - **Security Analysis:** A copilot could assist a security analyst by automatically analyzing logs for potential incidents and highlighting critical patterns.

Copilots work *with you*, not independently, enhancing productivity while still requiring user involvement.

For instance, see the list of Copilots in the **Microsoft ecosystem** below.


Microsoft 365 Copilot Chat
Microsoft 365 Copilot
Microsoft Security Copilot
Copilot in Azure
Copilot experiences in Dynamics 365
Copilot experiences for your industry
Copilot experiences in Power Platform
Copilot in Microsoft Fabric
GitHub Copilot

-------------------------------------Welcome to Journey of Agentic AI ---------------------------------

AI Agents: Autonomous Operators

AI Agents are the future. Unlike copilots, they don't just assist—they can operate independently and execute complex workflows without constant human input. These agents are like having a fully autonomous system that can make decisions, solve problems, and even improve itself over time.

- **Analogy:** A self-driving car that doesn't need a human driver at all—it navigates, plans routes, and reacts to traffic automatically.
- **Examples:**
    - Automating financial transactions by analyzing patterns, detecting fraud, and transferring funds.
    - Managing an entire production line in a factory, from monitoring machinery to optimizing output.
    - Handling customer service queries end-to-end, from understanding the problem to resolving it autonomously.

Chatbots are a long-standing concept, but AI agents are advancing beyond basic human conversation to carry out tasks based on natural language.

AI Agents help solving below areas,

- Operating the task independently
- Execute complex workflow autonomously
- Perform each of the tasks with or without human approval
- Perform specific tasks, answer questions, and automate processes for user

The Evolution of Complexity

AI agents come in varying levels of complexity:

1. **Simple Agents:** Perform basic tasks independently, like sending reminders or automating simple workflows.
2. **Semi-Autonomous Agents:** Work on moderately complex tasks and occasionally require human approval, such as approving a financial transaction.
3. **Advanced AI Agents:** Fully autonomous systems, like robotic process automation or self-driving cars, that execute intricate workflows without human intervention.

AI Agent Autonomy Scale

The level of autonomy provided by AI agents can be ranked on a 0-10 scale, where 0 is completely manual, and 10 is fully autonomous. This scale helps organizations determine the suitability of AI agents for specific use cases based on their requirements for automation and decision-making.

Score	**Description**
0-2	Basic automation or manual systems with limited AI assistance.
3-6	Semi-automated systems where AI agents handle repetitive tasks and provide recommendations.
7-8	Highly autonomous systems that execute end-to-end tasks but escalate critical decisions.
9-10	Fully autonomous systems capable of managing all operations without human intervention.

What is an Agent?

An agent is a powerful AI companion that can handle a range of interactions and tasks, from resolving issues requiring complex conversations to autonomously determining the best action to take based on its instructions and context. Agents coordinate a collection of language models, along with instructions, context, knowledge sources, topics, actions, inputs, and triggers to accomplish your desired goals.

Data engineering and Data and AI Governance are integral components that significantly contribute to the success of an agent. These areas are becoming increasingly important rather than diminishing in relevance.

Foundational components

Models - an AI model that is trained on broad data such that it can be applied across a wide range of use cases to support tasks like language processing, visual comprehension, text generation, code writing, and more. Different types of models (LLM, SLM etc..)

Since Agents more specialized task executor, leverage **industry specific models** or appropriate model for cost optimization and scaled deployment.

Microsoft provided benchmarking data helps choosing the appropriate model. Mostly customers start with GPT-4o or GPT-4o-mini, do internal testing / evaluations to determine the right-fit for the use case.

Typical Model selection parameters

- Multilingual support
- Quality output
- Context window
- Cost
- Latency
- Specialized areas like Math and Code

Tools

- Tools helps AI agents interact with the world and solve tasks effectively.
- Tools includes **knowledge base (Data Stores) and actions (Functions).**

Preparing your <u>knowledge base</u> for grounding:

Data-driven decision making – Every business is focusing on collecting and utilizing right data to be more competitive and to achieve efficient decision making. Applying **data engineering, data management and data quality, security guardrails** techniques are essential for preparing a knowledge base. Leverage Microsoft Fabric, Microsoft Purview, M365 and Azure cloud capabilities for the same.

Real-world trends:

- Most of the organizations are heavily focusing on Data-driven and data quality investments
- Customers are actively focusing on **unifying the data**, building enterprise level data hubs / decentralized architectures etc..

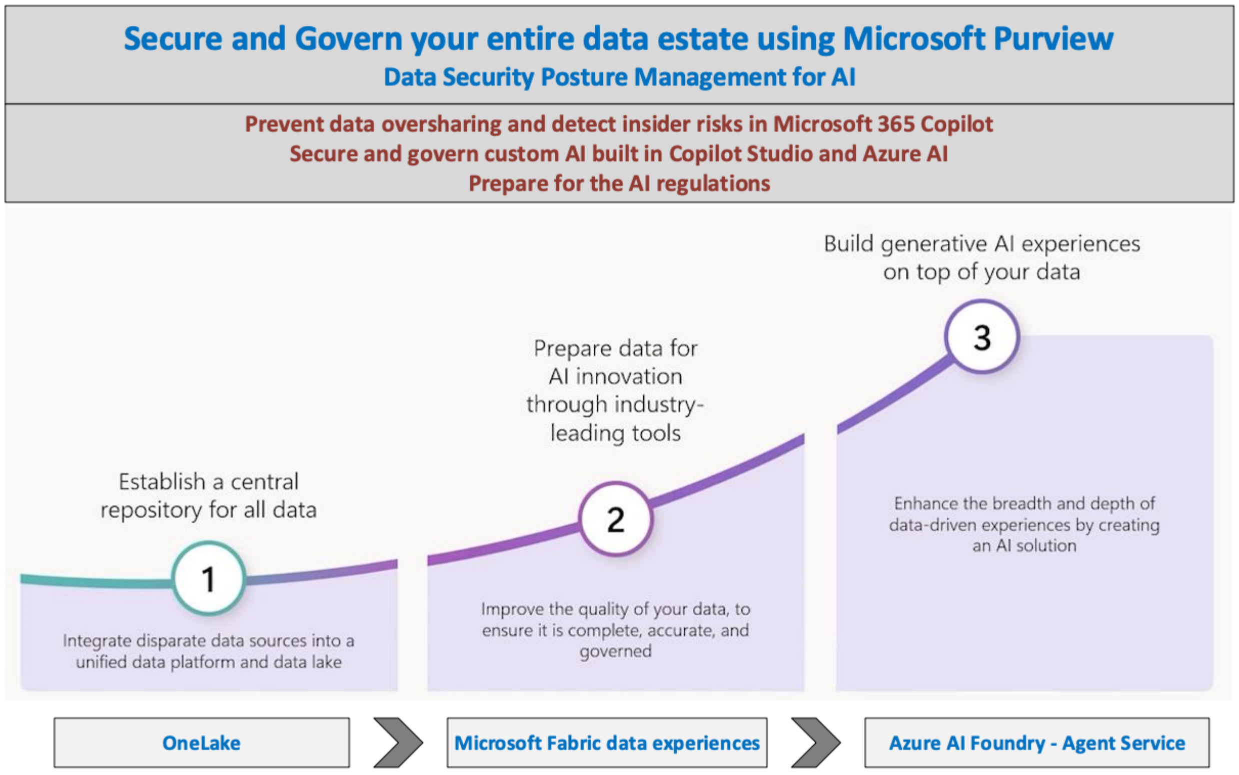
- Also explore partner based integrated data sources to enhance Agent responses.

Typical Knowledge base / data stores:
- Azure AI search
- Bing search
- M365, SharePoint and Graph connectors
- Storage systems (Blobs / local files)
- Partner integrations or data providers
- Microsoft Fabric
- Dynamics 365 Finance & Operations
- Azure Cosmos DB
- Clients or ISV owned data stores

Patterns:
- For example, ISV companies build and expose APIs to extend their knowledge base to custom copilots / agents.

Fuel your journey to AI innovation by getting your culture AI ready:

Depending on your use case and data strategies, different tech stacks can be utilized. **Here is one way to achieve this.**


Secure and Govern your entire data estate using Microsoft Purview
Data Security Posture Management for AI
Prevent data oversharing and detect insider risks in Microsoft 365 Copilot
Secure and govern custom AI built in Copilot Studio and Azure AI
Prepare for the AI regulations
Build generative AI experiences on top of your data
3
Prepare data for AI innovation through industry-leading tools
Establish a central repository for all data
2
1
Enhance the breadth and depth of data-driven experiences by creating an AI solution
Improve the quality of your data, to ensure it is complete, accurate, and governed
Integrate disparate data sources into a unified data platform and data lake
OneLake
Microsoft Fabric data experiences
Azure AI Foundry - Agent Service

Establish a central repository for all data

A strong data foundation is essential for successful AI transformation in any organization. OneLake, integrated into Fabric, serves as a centralized data lake that consolidates all your data into one accessible location for comprehensive management.

- OneLake is designed to help you simplify data management and reduce data duplication and is automatically wired into every Microsoft Fabric workload.
- You can bring data in using shortcuts which virtualizes data into OneLake from across clouds, accounts, and domains—all without duplication, movement, or changes to metadata or ownership.
- And OneLake's open data format means you only need to load the data into the lake once and you can use the single copy across every Fabric workload.
- Once in OneLake, you can use domains and workspaces to organize your data by business functions into a logical data mesh. You also get industry-leading governance and security to ensure only the right people have access to the right data.
- You can then empower everyone to search across this mesh using an intuitive, personalized data hub

Prepare data for AI innovation through industry-leading tools
For generative AI solutions to be as accurate as possible, they need to be built with clean data and in a semi-structured way. Fabric is the perfect platform to prepare your data before building custom AI experiences.

Build generative AI experiences on top of your data
And once you have a cleaned, robust dataset, it's time to start building generative AI experiences on top of your data.

Actions / Code execution and Scalability:
- Leverage Functions to implement stateless or stateful code-based actions.
- Take actions with Connectors (i.e. Azure Logic Apps, Power platform)
- Code interpreter
- Integration with existing infrastructure

Orchestration layer

The **Orchestration Layer** in an AI agent acts like a **traffic controller**. It manages how the agent takes in information, thinks about it, decides what to do, and then acts. It ensures that everything runs smoothly, step by step, until the agent reaches its goal.

Key Functions of the Orchestration Layer

• ***Input Handling***: It takes in user queries or requests and passes them to the agent for processing.

Example: User asks, "help me book a flight to New York?"

- ***Reasoning***: The layer helps the agent think through the problem using frameworks like ReAct, Chain-of-Thought (CoT), or Tree-of-Thoughts (ToT).

Example: The agent decides, "I need to use a **Tool or API** to answer this."

- ***Tool Selection***: The orchestration layer chooses the right tools (Functions, or Data Stores) the agent needs to complete the task.

Example: Let's dive deep on **Intent-to-function Mapping design approaches** a) Static mapping b) Dynamic mapping

Static mapping

```
# AgentTools.py

def search_flight(destination, date):

   return f"Searching for flights to {destination} on {date}."

def book_hotel(location, nights):

   return f"Booking a hotel in {location} for {nights} nights."

# init.py

from TravelAgentTools import search_flight, book_hotel

# Static intent-to-function mapping

intent_to_function_map = {

   "search_flight": search_flight,

   "book_hotel": book_hotel,

}
```

Limitations of Static Mapping

1. **Scalability**:
   - Becomes cumbersome as the number of intents grows.
2. **Flexibility**:
   - Requires manual updates for each new intent or function.
3. **Less Dynamic**:
   - Cannot handle fuzzy or ambiguous intent recognition as effectively as an LLM-based approach.

Dynamic mapping

In complex systems with many intents, maintaining a static mapping can become unwieldy. A dynamic approach allows the agent to manage these mappings efficiently.

You don't need to explicitly define the intent-to-function mapping, because frameworks like LangChain and systems using AgentExecutor provide a more dynamic and flexible way of handling this mapping. Use appropriate framework best-suited for your use cases.

```
tools = [

    Tool(name="search_flight", func=TravelAgentTools.search_flight, description="Search for flights to a destination."),

    Tool(name="book_hotel", func=TravelAgentTools.book_hotel, description="Book a hotel for a given location."),

]
```

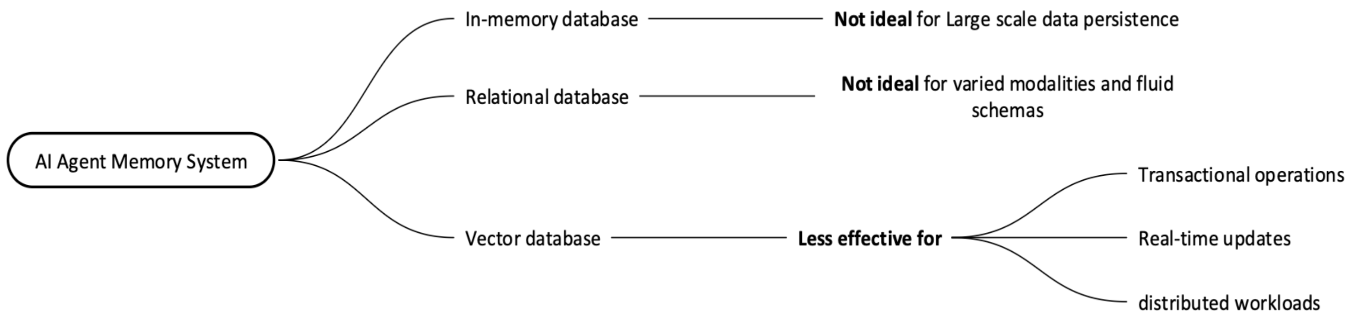
- ***Memory and Context***: Maintains a history of interactions so the agent can handle multi-step tasks or follow-up questions.

Example: User asks, "send me the itinerary?" The layer ensures the agent remembers the context of the previous query.

Agent Memory system

For Agent to execute the task, it needs specific context, structure or relationships that are relevant to task. Providing relevant and useful information to Agent is very crucial. Robust memory systems enable organizing and storing various kinds of information that the agents can retrieve at inference time.

As most of the agent use cases involve varied data types (varied modalities and fluid schemas of the metadata, relationships, entities, summaries, chat history, user preferences, sensory data, decisions, facts learned, or other operational data), using above standalone databases may adds complexity and performance bottlenecks.



Recommendation - Consider using unified / managed solution for AI agent memory systems. Below are the high-level parameters to help with choosing appropriate solution for Agent memory system.

Latency - rapid data access and management, single-digit millisecond latency
Scale - Global distribution and horizontal scalability
Simplicity - fully managed systems, flexibility in supporting varied data types etc..
Real-time - react to new information promptly.
Availability - Support for multi-master writes and high availability and resilience
Consistency levels - cater to support various distributed workloads depending on the requirements.

- ***Looping Until Goal***: The orchestration layer keeps cycling through reasoning, tool use, and actions until the agent reaches a stopping point or final answer.

Example: The agent uses multiple functions until it books the flight.

Technical Terminology

- **State Management**:

Tracks the current task progress and context.

Example: Keeping a session history in memory for multi-turn conversations.

- **Reasoning Frameworks**:

Guides how the agent thinks and plans (e.g., ReAct, CoT, ToT).

Example: Using CoT for step-by-step reasoning in solving math problems.

- **Execution Engine**:

Handles the actual running of tools, APIs, or functions chosen by the agent.

Example: Making a GET request to an external API.

- **Feedback Loop**:

Collects results from tools/actions and decides the next step.

Example: If a tool fails, it re-evaluates and tries another approach.

Analogy:

- Think of the orchestration layer as the **conductor of an orchestra**:

- The **musicians** are the tools (Functions, Data Stores).
- The **conductor** (orchestration layer) decides when and how each musician (tool) should play to create a harmonious performance (achieve the task).

***Why It's Important….*?**

The orchestration layer is the backbone of how an agent coordinates its thinking and actions to achieve a goal efficiently and effectively.

Without the orchestration layer:

- The agent wouldn't know which tool to use or when to use it.
- Tasks would remain incomplete, or responses could be inconsistent.
- Multi-step or complex workflows would be impossible.

Observability

Observability, in general, can happen in various app development lifecycle. Most organizations blend in traditional Machine Learning Operations (MLOps) to apply the same concepts for Gen AI apps in production. However, AI Observability serves a very different purpose. MLOps focuses on building, deploying and maintaining ML models while AI Observability's goal is to provide insights into LLM behavior, optimize performance, and ensure models are functioning correctly in production (e.g. reduce hallucination).

It's very crucial to have clear objectives and goals in mind when you bring your app into production. Here are the top questions to ask:

1. How to evaluate the performance of my app?
    a. This is not purely about application performance monitoring because in an agentic world, you could have several agents getting triggered and sent back results. The key area to understand is how the chain affects total user experiences.
2. How to detect and address master prompt, ingested data and model drift over time?
    a. The world of LLM evolves very quickly. Once you have new data ingested and your engineers decided to modify the prompt and/or upgrade model, how do you ensure the quality of the app and avoid unknown issues?
3. How to ensure security and privacy of users?
    a. Your users might send in sensitive data which you could detect. What if these sensitive data being queried by another user, etc…?
4. How to run evaluation based on prompts and responses and come up with good metrics for quality?

a. This is a very important question as you want to know if your AI responses hallucinating over time, becoming in-responsive / rejecting, getting worse on grounding, etc..

It’s tempting to deprioritize AI observability and treat it as an afterthought. However, once your AI application is in production, having robust monitoring in place becomes essential. It’s important to start on the journey early because you need to coordinate your internal team on various components:

1. **Logs**: Detail of events, function and API calls.
2. **Metrics**: Things like response time, latency, token usage.
3. **Traces**: Actual app traces such as which methods getting invoked and inputs / outputs which include information of prompts and responses.

Some examples of goals for AI observability system:

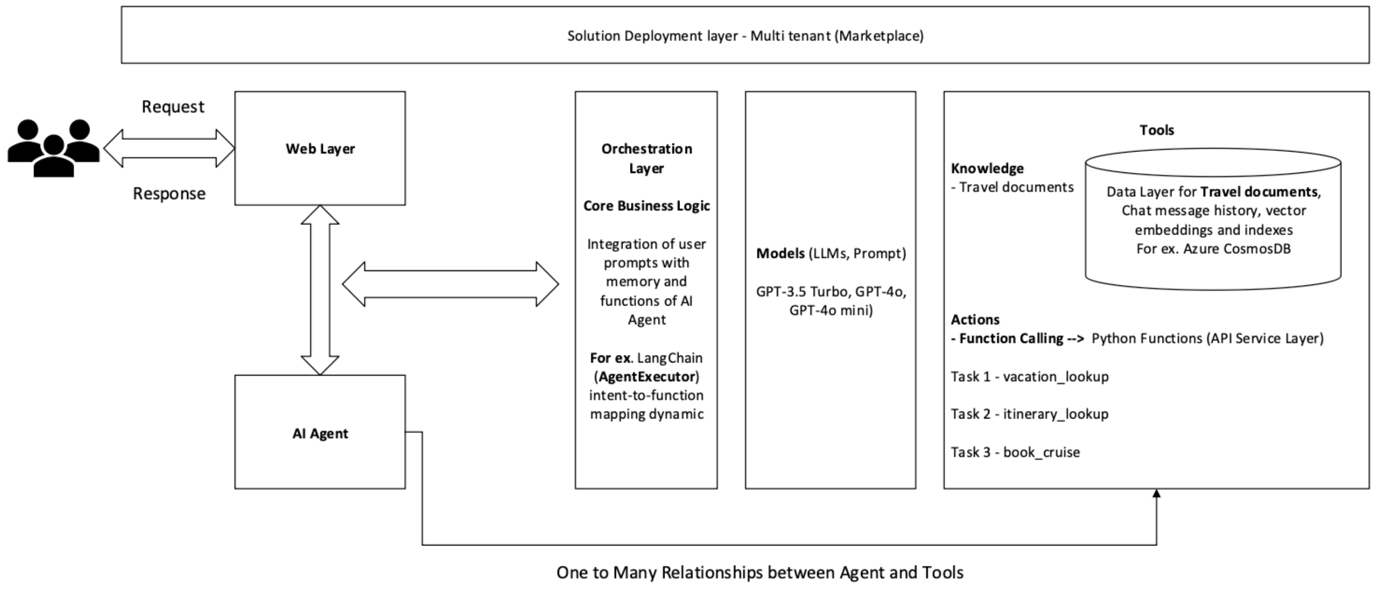
1. **Development Lifecycle**: Criterial to help graduate a development to pre-prod then prod.
2. **Deployment**: CI/CD so that if a certain tests failed, the system should not be promoted to prod.
3. **Executive dashboard:** detailed user experiences, customer-oriented metrics (e.g. happiness, satisfaction…)

Market research and development: LLM can result in very useful information based on natural language inputs from the users. Depending on your apps, you could derive great insights and help roadmap development.

Walk through the AI Agent components using example

Now that you have an understanding of the foundational components, let's review these components using an example.

Let’s say, you are developing a Travel Agent that assists users with travel-related inquiries and tasks, **for example**, Flight booking inquiry, hotel reservation inquiry, travel-itinerary assistance, book a flight or cruise etc..


Solution Deployment layer - Multi tenant (Marketplace)
Request
Response
Web Layer
AI Agent
Orchestration Layer
Core Business Logic
Integration of user prompts with memory and functions of AI Agent
For ex. LangChain (AgentExecutor) intent-to-function mapping dynamic
Models (LLMs, Prompt)
GPT-3.5 Turbo, GPT-4o, GPT-4o mini)
Tools
Knowledge
- Travel documents
Data Layer for Travel documents, Chat message history, vector embeddings and indexes For ex. Azure CosmosDB
Actions
- Function Calling --> Python Functions (API Service Layer)
Task 1 - vacation_lookup
Task 2 - itinerary_lookup
Task 3 - book_cruise
One to Many Relationships between Agent and Tools

Component	Description
Frontend – Web layer	It is responsible for routing requests and managing communication. For example, build agent using Python FastAPI which facilitates integration with the front-end user interface.
AI Agent	Agent uses the orchestration layer for selecting a sequence of actions to execute and communicates to users via API web layer.
Orchestration layer	It is responsible for primary business logic and interaction with the data layer, Agent and Tools. For example, its using LangChain AgentExecutor for intent-to-function mapping and does the overall function invocation. The AgentExecutor is responsible for orchestrating the interaction between the **agent** (usually powered by an LLM) and the available **tools** (functions or APIs). **AI Agent memory - history for each chat session** is stored in CosmosDB
Models	gpt-3.5-turbo-16k
Tools	**Actions -** Python functions for each travel task, for example vacation_lookup, itinerary_lookup, book_cruise etc.. **Knowledge base -** CosmosDB to store travel documents, chat session history etc..

Solution deployment layer	Based on use case, you need to determine what's the best deployment strategy for your environment. For example, you can choose to embed end-to-end agent solution in Teams app or M365 Copilot interface. Also, think about how you monetize the solution using Marketplace services.

Build your own Agent

Let's explore some scenarios and their respective guidelines.

Scenario 1: Enhancing SaaS Integration with M365 Copilot using Agentic Architecture

ABC Company: The Challenge
ABC Company's flagship product, "Product1," is a widely adopted SaaS solution tailored to streamline business operations for diverse industries. However, as customers increasingly embraced the Microsoft 365 (M365) ecosystem and its AI-driven capabilities like M365 Copilot, the demand for deeper integration between Product1 and M365 became evident.

The goal was clear: enhance the end-user experience by leveraging the power of Agentic architecture to enable seamless interaction across Product1, M365 SharePoint, and Microsoft Graph data sources. To achieve this, ABC Company embarked on a pilot journey to design vertical-specific AI Agents tailored to their customers' unique needs.

The Pilot Journey: Goals and Strategy

1. **Building Vertical-Specific Agents**
   ABC Company aimed to create agents optimized for specific industries, such as healthcare, finance, and retail. These agents would combine:
   - Domain expertise from Product1's knowledge base (retrieved via APIs).
   - M365 data sources, such as SharePoint and Microsoft Graph.
2. **Seamless Data Correlation**
   The agents needed to intelligently correlate data across multiple sources to answer complex queries and deliver actionable insights.

The Whiteboarding Exercise: Key Considerations

1. **User Interface**
   A fundamental question arose:
   - Should the end-users interact with the agent through the familiar M365 UI, such as Microsoft Teams, or through ABC Company's proprietary application?
   - The decision required balancing user familiarity with customization needs.
2. **Sample Scenarios**
   The team brainstormed potential user queries and workflows to test the agent's reasoning abilities, such as:

    - "Retrieve the latest project updates from Product1 and SharePoint."
    - "Summarize cross-department collaboration trends using Product1 and Microsoft Graph data."
3. **LLM and Prompt Engineering**
   They explored how Large Language Models (LLMs) and prompt engineering could enable the agent to:
    - Perform contextual reasoning.
    - Connect the dots between diverse data points.
    - Handle client-specific business logic effectively.
4. **API Design**
   A critical decision was whether to:
    - Build more APIs for improved retrieval and RAG architecture.
    - Or, consolidate responses via a single API for simplicity.
5. **Scaling Questions**
   The team debated between:
    - Low-code approaches (using Copilot Studio for rapid deployment).
    - Pro-code solutions for greater customization and scalability.
6. **Connector Strategy**
   Identifying the right connectors was crucial:
    - Power Platform connectors for automation.
    - Microsoft Graph for deep integration.
    - Custom connectors as a gateway for unique retrieval needs.
7. **Agent Quality Evaluation**
   Defining quality release criteria involved:
    - Metrics for response accuracy and relevance.
    - Rigorous testing for edge cases and performance under load.
8. **Retrieval Strategy**
   Ensuring efficient data retrieval from:
    - Product1 APIs.
    - Microsoft Graph connectors, while monitoring index quota usage.
9. **Agent Design Choices**
    - Should they adopt a Declarative agent for easier deployment?
    - Or a Custom engine agent for fine-grained control?
10. **Deployment and Licensing**
    Deployment options and licensing requirements varied:

- Declarative agents offered quicker setups.
- Custom engine agents required more resources but delivered flexibility.
- Licensing considerations for connectors like Microsoft Graph and Power Platform influenced overall costs.

Additional reference data points:

Declarative agents are easier to build with minimal coding, while custom engine agents require more technical skills.

- **Technical Expertise**: Declarative agents are easier to build with minimal coding, while custom engine agents require more technical skills.
- **Flexibility**: Custom engine agents offer more advanced capabilities and customization options.
- **Use Cases**: Declarative agents are suitable for simpler, predefined tasks, whereas custom engine agents are better for complex, dynamic interactions.

Recommendations:

- Analyze, what's the orchestration layer looks like? Are you planning to use Microsoft Copilot foundation model / orchestration **OR** need more advanced orchestration customization? Doing a quick POC on both approaches - Declarative and Custom engine agent would help identifying best orchestration option.
- Assess your architecture based on these parameters: end-user experience, developer effort, data freshness, compliance, trust and security, total cost of ownership, and licensing.
- To get more flexibility in deployment, management and updates across tenants, go for custom engine agent.
- Scaling aspects - maintain the agent's individuality and focus, use specific data, assign specific roles, keep the data targeted, and concentrate on particular tasks.

Scenario 2: Automating Customer Enrollments with AI Agents

Company XYZ: The Challenge
XYZ is a growing service provider known for its customer-centric approach. However, the company's enrollment process was becoming a bottleneck. Every day, XYZ received hundreds of customer enrollment requests via email. These emails contained essential information such as names, contact details, preferences, and required documents attached as PDFs or images.

An operator was responsible for reading these emails, extracting data, and manually entering it into XYZ's proprietary enrollment system. The process was labor-intensive, prone to errors, and delayed responses to customers. With increasing demand, XYZ sought a smarter solution.

Enter the AI Agent
XYZ turned to AI technology to tackle this inefficiency. They implemented an AI-powered automation agent designed specifically for the task. Here's how it transformed their operations:

1. **Email Monitoring**
   The AI agent was integrated with XYZ's email system. It continuously monitored the inbox for new customer enrollment requests. Using natural language processing (NLP),

the agent identified relevant emails based on keywords like "enrollment request" or "new customer."

2. **Data Extraction**
   Upon detecting an enrollment email, the agent extracted structured information from the email body and attachments. For example:
   - From the email body: Name, email, and phone number.
   - From attached PDFs: Address, preferences, and any scanned IDs.
   - Using OCR (Optical Character Recognition), it also extracted text from image-based attachments.
3. **Data Validation**
   Before proceeding, the AI validated the extracted data. It cross-checked for missing fields, verified the format of email addresses and phone numbers, and flagged inconsistencies, such as expired documents. In some cases, it even sent an automated reply to customers requesting corrections or additional documents.
4. **System Integration**
   After validation, the agent logged into XYZ’s proprietary enrollment system using API access. It auto-filled the necessary fields with the extracted data, mimicking the operator’s manual process but completing it within seconds.
5. **Confirmation and Logging**
   Once the submission was complete, the AI agent sent a confirmation email to the customer, thanking them for their application and providing an estimated processing time. Simultaneously, it logged the entire interaction for compliance and auditing purposes.

Scenario 3: Streamlining Business Process Approvals with AI Agents

Company ABC: The Challenge
ABC Corporation operates a complex, multi-step business process that requires frequent approvals. Each step involves decision-making based on predefined rules, stakeholder inputs, and occasionally, manual escalations.

For instance, the finance department manages vendor payments, where:

- Low-value payments (< $5,000) follow a fast-track auto-approval process.
- Payments between $5,000 and $50,000 require a manager's approval.
- Payments exceeding $50,000 need director-level approval and sometimes a special review.

The existing system required manual oversight at every level, leading to delays, missed deadlines, and unnecessary overhead. ABC needed a solution to optimize its workflows without compromising compliance or accuracy.

Enter the AI Agent
ABC deployed an AI agent to streamline and automate its business approval process. Here's how it worked:

1. **Workflow Integration**
   The AI agent was embedded into the company's process management software. It monitored incoming approval requests and classified them based on the predefined rules and thresholds.
2. **Fast-Track Approvals**
   For approvals below the $5,000 threshold, the agent automatically processed the requests:
   - Verified the accuracy of associated data (e.g., vendor ID, invoice details).
   - Checked compliance with internal policies (e.g., no duplicate payments).
   - Approved and initiated payment workflows within seconds.
3. **Conditional Approvals and Escalations**
   For mid-tier requests ($5,000–$50,000):
   - The agent prepared a summary of key details (e.g., budget allocation, payment urgency).
   - Forwarded the request to the assigned manager with contextual insights (e.g., flagged risks or potential policy violations).
   - Sent follow-up reminders if approvals were delayed beyond set timeframes.

   For high-value requests (> $50,000):

   - The agent escalated them to the director and flagged them as high-priority.
   - Generated a detailed report with all supporting documentation, including expense trends and past approvals for similar cases.
4. **Customizable Logic**
   The AI's approval logic was customizable, allowing stakeholders to add new conditions, such as:
   - Escalating payments above $75,000 to the CFO for review.
   - Applying auto-approvals for specific trusted vendors.
5. **Proactive Communication**
   The agent sent real-time notifications to stakeholders at every step:
   - "Request approved."
   - "Pending manager approval for payment ID #12345."
   - "Escalation initiated due to amount exceeding approval limits."
6. **Performance Insights**
   The AI tracked workflow bottlenecks and generated analytics dashboards for decision-makers:
   - Average approval times.
   - Escalation frequency.
   - Trends in approval requests by department or vendor.

Typical customer patterns

Customer 1 - I don’t want to manage memory, context history and dynamic interactions.
Customer 2 - I am looking for more control in orchestration and output generation.
Customer 3 - Start small and grow – let’s go with low-code approach and phase 2 will be more customized or pro code approach.
Customer 4 - I want data insight in plant operations which automate multi-step workflows and generate detailed reports.
Customer 5 - I need something that blends automated and manual workflows seamlessly.
Customer 6 - I want instant data analysis with actionable insights and visualizations.
Customer 7 - I need a tool that predicts potential risks and suggests mitigation strategies.
Customer 8 - I want to make sure the data go into my AI app is secured and within private network boundary
Customer 9 - I want intelligent recommendations from AI but want human in the loop as always.

Unpack agent use cases by considering the following questions

User Experience:

- What is the primary interface for the user (e.g., M365, proprietary application)?
- Should the agent be embedded in an existing platform or be standalone?
- How do users interact with the agent (chat, dashboard, API)?

Knowledge Base:

- What data sources constitute the knowledge base?
- Is the knowledge base static or continuously updated? If updated, how often?
- Are there privacy and compliance considerations for storing/using knowledge?

Actions in Workflow:

- What key actions will the agent perform in the business process?
- Which steps require user validation versus full automation?
- Are there dependencies between actions or external systems?

Data Consumption from Client’s System:

- What data sources do we integrate with (e.g., databases, APIs, file systems)?
- How do we ensure data security during retrieval and processing?
- Are there existing APIs or connectors? If not, what needs to be developed?

Data Retrieval from M365:

- Is Microsoft Graph API direct query sufficient for your use case? If not, why?
- What are the performance and scalability requirements for data retrieval?
- Are there alternative methods (e.g., webhooks, direct database access)?
- How do you handle authentication and authorization for M365 data?

Simplifying AI Development with Azure AI Agent Service

Introduction

- Developers face significant challenges when building custom AI agents, including:
  - Managing conversation state and chat threads.
  - Tool integrations and retrieval systems.
  - Executing code manually.
- Stateless APIs, such as Azure OpenAI's chat completions API, require developers to handle these complexities independently.

Azure AI Agent Service: A Game-Changer

- Azure AI Agent Service addresses these challenges, offering:
  - Persistent, automatically managed threads.
  - Built-in conversation state management.
  - Solutions to work around model context window constraints.
- Designed to enhance developer productivity and streamline AI agent workflows.

Key Features

1. **Built-In Conversation State Management**
   - Automatically manages and retrieves conversation history from end-user interactions.
   - Ensures consistent context within user sessions, enabling:
     - Retention of conversation history.
     - Delivery of personalized and richer interactions.
     - Improved performance of AI agents over time.
   - Avoids manual thread management and mitigates context window limitations of AI models.
2. **Persistent Threads**
   - Threads are optimized automatically to stay within the model’s maximum context window.
   - Developers can append new messages to ongoing threads, ensuring fluid conversations.
3. **Multi-Tool Access**
   - Seamless access to multiple tools simultaneously.

   - Enhances flexibility for developers by integrating various capabilities into one system.

Key considerations

AI Agents operate in environments that demand strict adherence to enterprise InfoSecurity and Quality standards. Ensuring the reliability of AI Agents requires addressing several critical aspects, such as data privacy, cost efficiency, performance quality, and scalability. To guide organizations on this journey, here’s a comprehensive cheat sheet to help you design, deploy, and manage AI Agents effectively.

1. **Data Sensitivity and Compliance**

- ***Sensitivity Labeling:*** Ensure data accessed by Agents is labeled appropriately. Agents should recognize and honor sensitivity labels to protect sensitive data.
- ***Retention and Deletion Policies:*** Apply retention and deletion policies to prompts and responses for compliance.
- ***Access Control:*** Restrict sensitive data access to authorized users only.

For example, by incorporating SharePoint sites as a knowledge base, you can build agents that provide Microsoft Purview data security and compliance controls on the data within those sites. For instance, when a user interacts with the agent, it accesses sensitive data by recognizing and honoring sensitivity labels.

2. **Visibility and Traceability**

- *Implement strategies to monitor the sensitivity of data in user prompts and Agent responses.*
- *Enable traceability to detect risky or non-compliant AI use.*

For data security admins, Microsoft Purview offers the ability to discover the sensitivity of data in user prompts and agent responses within DSPM for AI. Additionally, it can detect user activity anomalies and risky AI usage, and govern user prompts and agent responses with audit, eDiscovery, retention policies, and non-compliant usage detection.

3. **Agent Orchestrate, Development and Architecture**

- Select the right model, orchestration layer, plugins, and actions based on the deployment plan.
- Define roles, including Agent developers, AI councils, security teams, and business unit stakeholders, to create a robust architecture.

Leverage Azure AI Agent Service, it is a **fully managed** service designed to empower developers to securely build, deploy, and scale high-quality, and extensible AI agents without needing to manage the underlying compute and storage resources.

Azure AI Agent Service works out-of-the-box with multi-agent orchestration frameworks that are wireline compatible with the Assistants API, such as **AutoGen**, a state-of-the-art research SDK for Python created by Microsoft Research, and **Semantic Kernel**, an enterprise AI SDK for Python, .NET, and Java.

4. **Prompt Engineering**

- Develop a structured prompt engineering framework to ensure accurate Agent responses.
- Combine prompt engineering with the chosen AI model for improved performance.

5. **Technical Considerations**

- Understand quotas and limits for the AI models being used, such as OpenAI models.
- Differentiate between chat completions APIs and Agent services to align with your use case.
- Plan for context window truncation and set appropriate parameters for optimal outcomes.

6. **Security and Responsible AI**

- Implement multi-cloud security posture management solutions.
- Apply content filters and adhere to Responsible AI principles to maintain ethical standards.

Tip: Leverage Microsoft Purview DSPM (Data Security Posture Manager) for AI, Insider Risk Management, and data compliance controls for agents built in Copilot Studio. Microsoft Defender for Cloud to protect AI agent (GenAI risks and threats, jailbreak attempts). To reduce the risk of harmful use of the Azure AI Agent Service, use the content filtering feature.

7. **Debugging and Performance**

- Utilize tracing capabilities for troubleshooting complex Agents.
- Adjust default chunk size and overlap settings as required by your tools.
- Ensure your vector store readiness before your agent runs the search.

Tip: Leverage tracing using Application insights (Azure AI Foundry)

8. **Integration and Scalability**

- Use function triggers and bindings to simplify Agent interaction with external systems and services.
- Test APIs for authentication and determine calling behavior.

AI Agent Suitability Across Industries

AI agents can revolutionize operations across industries by enabling automation, enhancing decision-making, and streamlining workflows. They leverage advanced machine learning models, real-time data processing, and automation capabilities to perform tasks ranging from simple data retrieval to complex decision-making with minimal or no human intervention.

The adoption of AI agents depends on the industry's complexity and the degree of autonomy required. For instance, industries like manufacturing, logistics, finance, healthcare, and retail each have unique workflows and challenges where AI agents can make a significant impact. Most industries employ AI agents in semi-automated or highly autonomous roles, where they handle routine tasks autonomously and escalate critical decisions or edge cases to humans.

Here is an example of AI Agent use case application in industry.

Use case: Agent role in Manufacturing Industry

The use of AI agents has the potential to transform industry operations by automating tasks, enhancing decision-making, and optimizing processes across various domains. These agents act as specialized Subject Matter Experts (SMEs) in specific operational areas, enabling industries to handle complex workflows with increased efficiency and precision. Below is an overview of how AI agents can be applied to different domains and their relevance to modern industrial applications

1. **Agent@Production**
   - Acts as a specialize agent for production line management by autonomously scheduling production runs, identifying bottlenecks, and dynamically allocating resources to meet demand.
   - **Capabilities**:
     - Optimizes line balancing based on real-time data from IoT sensors.
     - Detects inefficiencies and suggests or implements corrective actions.
     - Predicts output based on current production conditions.
2. **Agent@SupplyChain**
   - Specializes in supply chain optimization by managing procurement, tracking shipments, and ensuring on-time delivery.
   - **Capabilities**:
     - Tracks supplier performance and predicts delays.
     - Recommends alternate suppliers during disruptions.
     - Maintains optimal inventory levels by analyzing demand patterns.
3. **Agent@Maintenance**
   - Focuses on asset reliability by predicting failures, scheduling maintenance, and ensuring equipment uptime.
   - **Capabilities**:
     - Analyzes equipment sensor data to predict failures.
     - Creates maintenance schedules that minimize downtime.
     - Orders spare parts proactively, integrating with procurement systems.
4. **Agent@Quality**
   - Ensures product quality and compliance by automating defect detection and root-cause analysis.
   - **Capabilities**:
     - Analyzes production data for patterns leading to defects.
     - Uses computer vision to detect defects in real time.
     - Provides insights into improving process quality and meeting compliance standards.
5. **Agent@Energy**
   - Optimizes energy usage to reduce costs and improve sustainability.
   - **Capabilities**:

- Monitors energy consumption patterns in real time.
- Suggests process changes to reduce energy waste.
- Assists in reporting sustainability metrics for compliance

Looking Ahead…

Everyone Will Pilot Vertical and Horizontal Agents - As the AI landscape continues to mature, organizations across industries will increasingly pilot both vertical and horizontal AI agents. Vertical agents are specialized in handling domain-specific tasks, such as healthcare diagnostics or financial portfolio management, while horizontal agents operate across domains, enabling seamless collaboration and task automation. The adoption of these agents will empower businesses to experiment with tailored solutions, optimize workflows, and identify opportunities for innovation. With competition intensifying, rapid experimentation with these agents will become a strategic imperative for organizations aiming to remain competitive.

Adoption Challenges - Despite their potential, the adoption of AI agents will face several challenges:

- **Trust**: Building trust among users and stakeholders will be critical. AI agents must demonstrate reliability, transparency, and ethical behavior to gain widespread acceptance.
- **User Personalization**: For AI agents to be truly effective, they must cater to individual preferences, contexts, and workflows. Achieving this level of personalization will require sophisticated algorithms and seamless integration with user data.
- **Fully Autonomous Capabilities**: While autonomy is a key promise of AI agents, achieving full autonomy in complex environments will be a gradual process. Striking the right balance between autonomy and human oversight will be crucial.
- **Governance**: Establishing governance frameworks to monitor AI agents' actions, ensure compliance with regulations, and address potential biases will remain a pressing challenge. Organizations will need to focus on ethical AI practices to avoid unintended consequences.

ISVs Focusing on Product-Level Integrations - Independent Software Vendors (ISVs) will play a pivotal role in simplifying the end-user experience by developing product-level integrations with cloud providers. These integrations will ensure that AI agents can seamlessly interact with cloud-based infrastructure, services, and tools. By leveraging cloud-native capabilities, ISVs will enable AI agents to scale efficiently, process data in real-time, and deliver actionable insights. This ecosystem will significantly reduce complexity for end-users, paving the way for broader adoption of AI-driven solutions.

SaaS Is Dead: The Rise of AI Agents - The traditional Software-as-a-Service (SaaS) model is undergoing a paradigm shift as AI agents take center stage. Instead of relying on standalone SaaS applications, businesses will increasingly leverage AI agents that manage business logic and directly perform Create, Read, Update, and Delete (CRUD) operations on backend systems. These agents will act as dynamic intermediaries, eliminating the need for rigid interfaces and enabling more intuitive, conversational interactions. As a result, the lines between SaaS, PaaS, and IaaS will blur, creating a unified, agent-driven ecosystem that prioritizes efficiency, flexibility, and user-centric design.

Team Collaboration and Restructuring - The journey toward adopting and scaling AI agents across organizations is **not just a technological transformation**—it's a fundamental shift in how teams collaborate and operate. For organizations to unlock the full potential of AI transformations, significant restructuring and a renewed focus on collaboration across multiple domains are imperative. Ultimately, restructuring organizational processes and strengthening collaboration across these groups will unlock the full transformative potential of AI agents and lead to sustained success in the AI journey.

Few examples,

Partners: Bridging Data, Analytics, and AI Practices - The ecosystem of partners—consultancies, system integrators, and independent software vendors—must also evolve to meet the demands of the AI-driven future. Data, analytics, and AI practices that were traditionally independent must converge to provide holistic solutions to customers.

End Customers - For end customers, the effective deployment of AI agents requires strong collaboration among diverse stakeholders, including IT teams, InfoSecurity, Privacy, Legal, AI Governance Councils, and business-aligned stakeholders. This entails:

- Defining a **solid data foundation** with clear ownership and accountability.
- Establishing robust **AI governance** frameworks to ensure ethical and responsible AI use.
- Encouraging **cross-departmental collaboration** to align AI strategies with broader business objectives.