5Star
Are you future-ready?

Agentic AI

GenAI Center of Excellence
IBM CIC Slovakia
04.12.2025

Speakers

RNDr. Renáta Ujháziová, PhD.

Senior Data Scientist,
AI Consultant,
Gen AI CoE Leader CIC Slovakia

RNDr. Štefan Bocko

GenAI & Agentic AI Solution Architect

Learning outcomes

1. **<u>Understand</u> the Foundations of Agentic AI.**
    - Understand the core principles of Agentic AI and how it differs from other AI systems.
    - Explore the key components that enable autonomy and reasoning.
    - Get familiar with leading Agentic AI frameworks and tools.
2. **<u>Develop</u> practical skills.**
    - Gain experience in designing and implementing basic Agentic AI workflows using Semantic Kernel.
3. **<u>Apply</u> knowledge through case studies.**
    - Examine a real-world case studies demonstrating Agentic AI in action.
4. **<u>Innovate</u> with confidence.**
    - Leave with the knowledge and frameworks to apply Agentic AI workflows in your own projects or research.

Agenda

1. **Welcome and Introduction to Agentic AI session** (10 min.)
2. **Introduction to Agentic AI** (25 min.)
   - What is Agentic AI and how it differs from traditional and generative AI systems.
   - Core elements and architectural patterns.
   - Framework examples, protocols, solution guidance and evaluation.
3. **Demo: Developing practical skills** (20 min.)
   - Introduction to Semantic Kernel.
   - Build a simple agentic AI workflow using Semantic Kernel.
4. **Case studies presentation: Banking use case / Consumer electronics** (20 min.)
5. **Future of Agentic AI** (5 min.)
6. **Key takeaways and Q&A** (10 min.)

Introduction to Agentic AI

Evolution of AI over the past few years

2023

Year of LLM

2024

Year of RAG

2025

Year of AI Agents

What is AI Agent?

An AI Agent is a system that ***autonomously (independently) reasons, plans, and takes actions*** to achieve a specific goal.

AI Agents key components

LLM
(Brains)

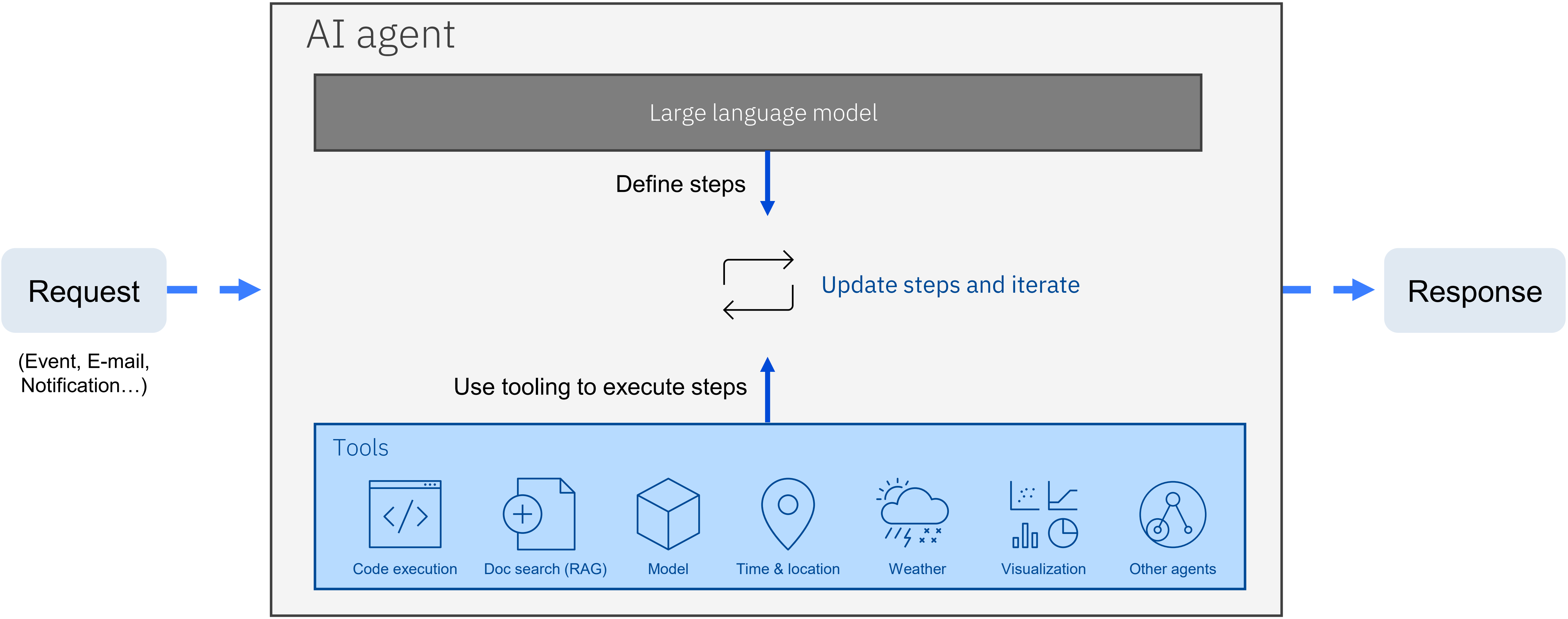
Memory
(Short-term &
Long-term)

Tools
(Hands)

Goals
(Tasks)

How do AI Agents work?

An AI agent is a program whose execution logic is controlled by a large language model.



Request

(Event, E-mail, Notification…)

AI agent

Large language model

Define steps

Update steps and iterate

Use tooling to execute steps

Tools

- Code execution
- Doc search (RAG)
- Model
- Time & location
- Weather
- Visualization
- Other agents

Response

How do AI Agents reason?

Planning

An agent receives a complex request and determines the most appropriate sequence of actions to execute and complete that tasks.

Tool Use

An agent can analyze, research, and generate content using web search, object detection, cloud storage, email and calendar entries, etc.

Reflection

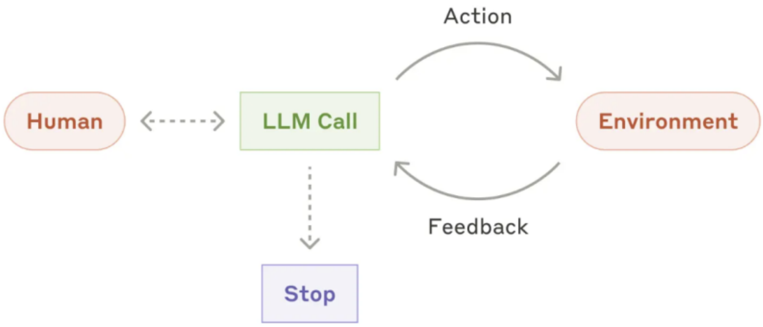
A workflow where an agent repeats a task, resulting in an improvement after each iteration.

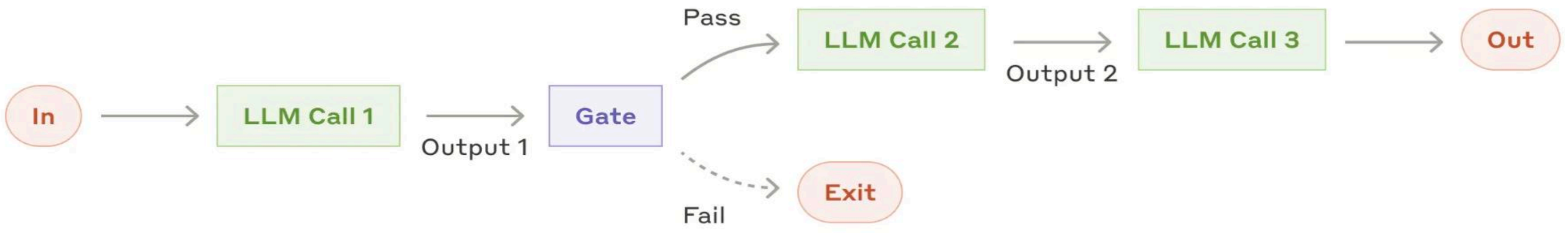
Collaboration

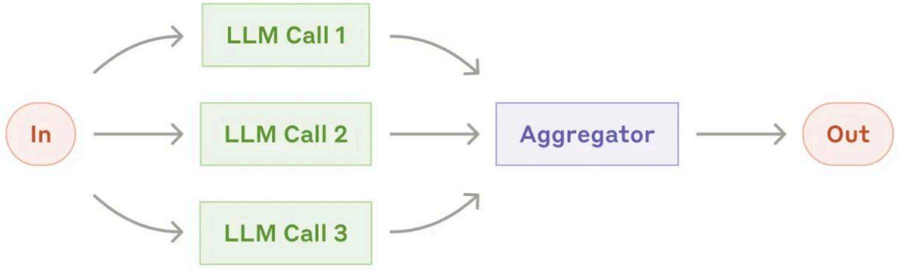
Multiple agents coordinate to perform differing tasks to complete a higher-level, complex set of requests.

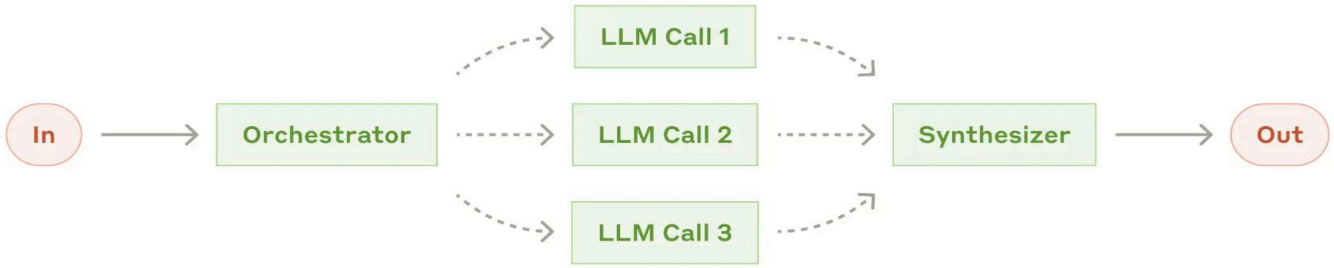
AI Agent patterns

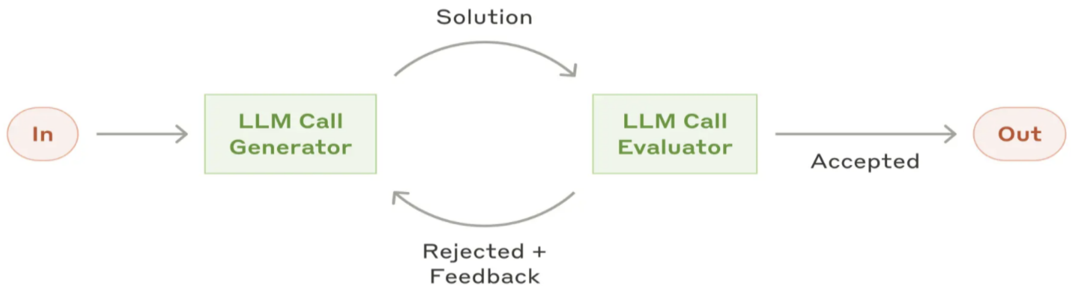
Agents



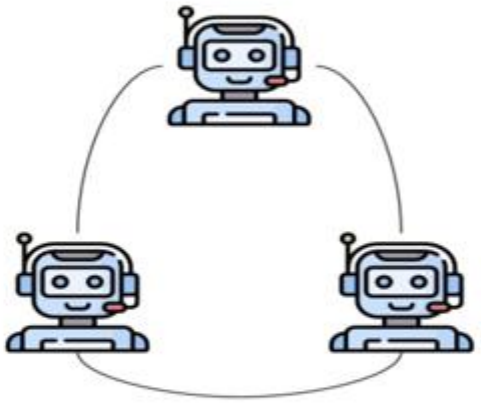






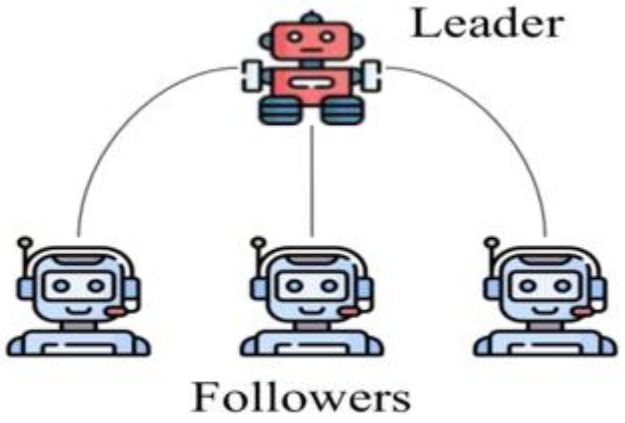


Structure of Multi-agent systems



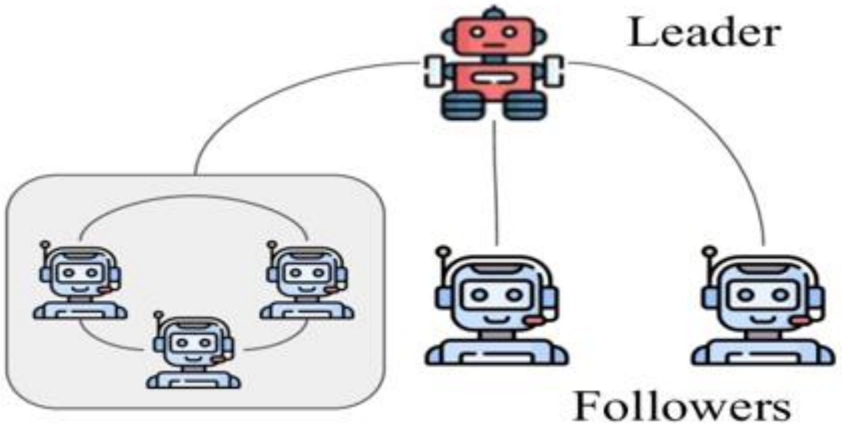
a. Equi-level structure

- Agents operate at the same hierarchical level.
- Collaborate or negotiate without a centralized leader.
- Shared responsibilities and collective decision-making.



b. Hierarchical structure

- Consists of leaders and followers.
- Leaders guide and plan; followers execute instructions.
- Used in scenarios requiring coordinated efforts.



c. Nested structure

- Combines equi-level and hierarchical structures.
- Complex tasks broken into sub-tasks managed by sub-systems.
- Interplay between hierarchy and peer-to-peer interaction.

Single agent vs Multi agent trade-offs

You can use multi-agent solutions where single agents were used - just consider the key trade-offs.

Single agent

Pros:

1. Lower and more predictable runtime cost.
2. Lower latency for time critical systems.
3. Low implementation complexity.

Cons:

1. Zero shot accuracy/quality limited by performance of selected model.
2. Less flexible for implementation of complex workflows.

Multi agent

Cons:

1. Increasing and less predictable runtime cost since the number of LLM invocations increase and is not fully deterministic.
2. Increasing latency especially for parts of workflows that cannot be parallelized due to sequencing of LLM calls.
3. Increasing implementation complexity where it is often harder to anticipate and evaluate the impact of changes to the solution.

Pros:

1. Significant empirical evidence that the accuracy/quality of multi agent systems outperforms single agent zero shot solutions.
2. Scales to automation of sophisticated workflows with state memory and human in the loop to support complex business processes.

Start simple and increase complexity when needed.

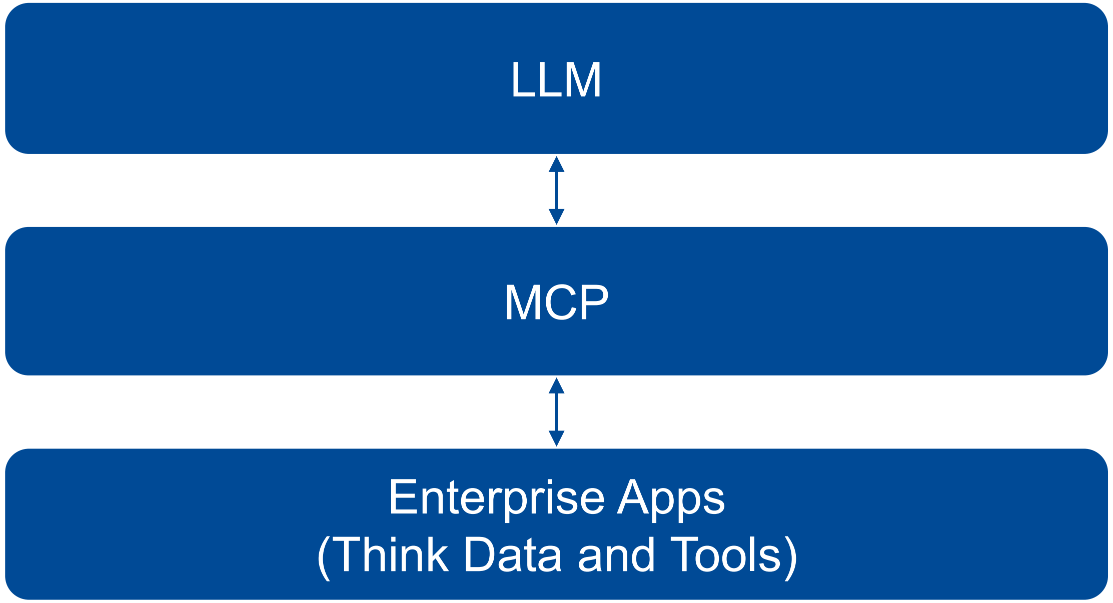
Agentic AI Frameworks

Feature	LangGraph	CrewAI	Autogen	Semantic Kernel
Architecture	Graph-based, visualizing dependencies and managing agent interactions. 🟢	Role-based, assigning specific roles and goals to agents. 🟢	Consensus based. 🟢	Plugin-based, planner-executor architecture enabling orchestration of AI functions and native code. 🟢
Task Execution	Supports cyclic graphs, allowing agents to revisit previous steps and adapt to changes. 🟢	Supports sequential and hierarchical task execution modes. 🟢	Leverages the collective behaviour of simple agents for decentralized problem-solvi 🟢	Supports sequential and goal-oriented task execution via planners (sequential, stepwise, function-calling) 🟢
Agents	Has built-in ReAct agent and a selection of community-built agents. Can easily be integrated with other agentic platforms. 🟢	Has a wide selection pre-built agents. 🟢	Couple of pre-defined agents for code execution, mulitmodel websurfing, file surfer and video surfer. 🔴	Enables custom agent behaviour through Skills (functions), Memories (context), and Planners. Integrates with OpenAI, Azure OpenAI, HuggingFac 🟢
User-Friendliness	May have a learning curve due to its complex, graph-based architecture 🔴	Provides a user-friendly platform for creating and managing multi-agent systems. 🟢	Complexity depends on the specific implementation and use case. 🔴	Developer-friendly for C#, Python and Java users, with clear modular structure; some learning curve for orchestration logic. 🟢
Integration	Integrates with tools like LangChain and LangSmith for enhanced functionality. 🟢	Various tools and models, including LangChain interoperability. 🟢	Integrates with several models, tools and can be extended 🟢	High extensibility: Integrates with LLMs, external APIs, custom functions, and supports memory (embedding/vector stores) and planner integration 🟢
UI/GUI	LangGraph Studio 🟢	CrewAI UI Studio 🟢	Autogen Studio 🟢	No official GUI yet; primarily a code-first SDK approach. Visual tooling may require external dashboards. 🔴

🟢 = Pros 🔴 = Cons

Agentic AI protocols

Model Context protocol (MCP)



LLM

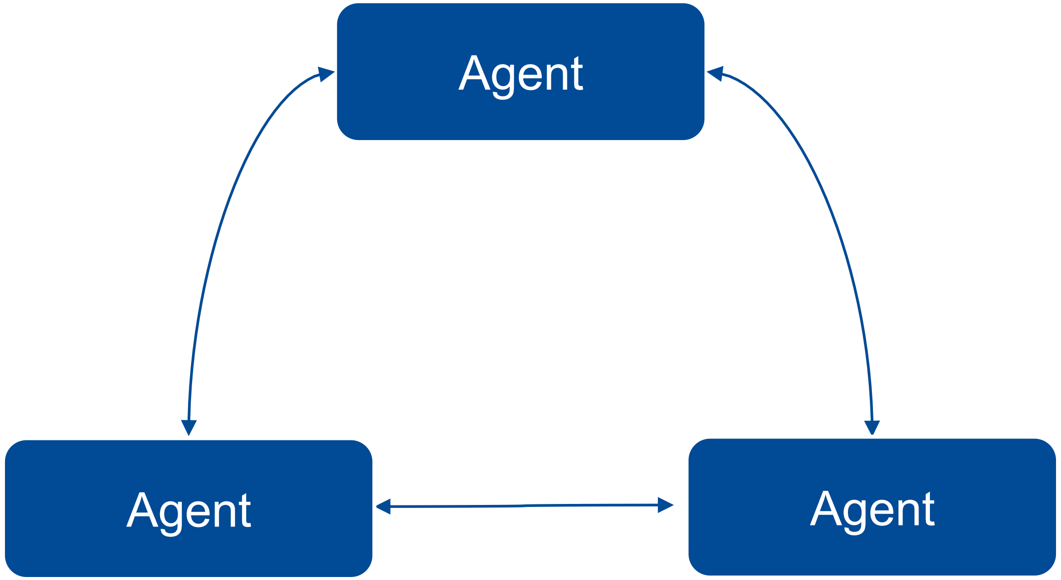
MCP

Enterprise Apps
(Think Data and Tools)

Vertical

(Model to App)

Agent2Agent protocol (A2A)



Agent

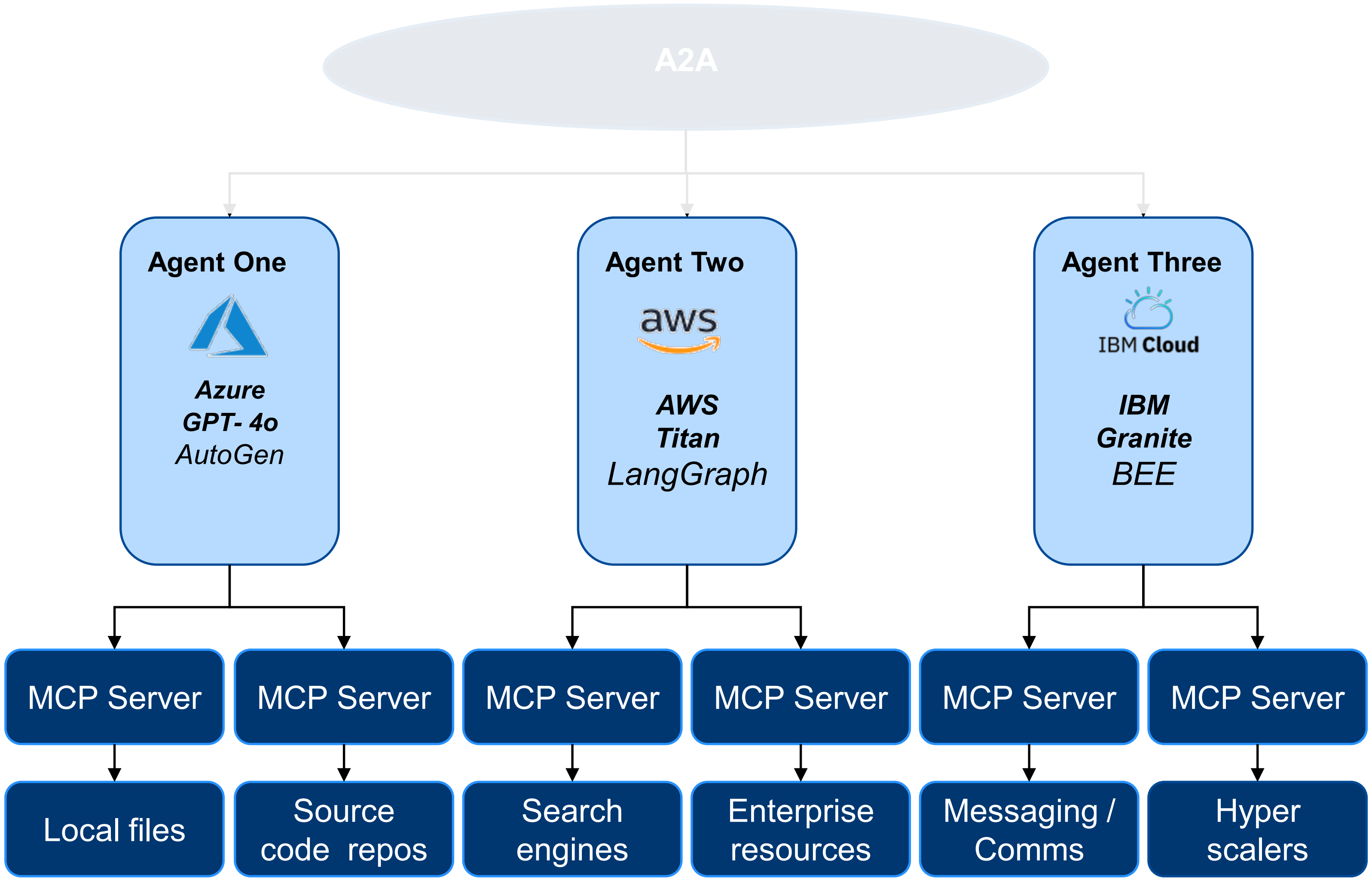
Agent

Agent

Horizontal

(Agent to Agent)

Communication protocols: A2A vs MCP



A2A

Agent One

Azure
GPT- 4o
AutoGen

Agent Two

aws

AWS
Titan
LangGraph

Agent Three

IBM **Cloud**

IBM
Granite
BEE

MCP Server

MCP Server

MCP Server

MCP Server

MCP Server

MCP Server

Local files

Source code repos

Search engines

Enterprise resources

Messaging / Comms

Hyper scalers

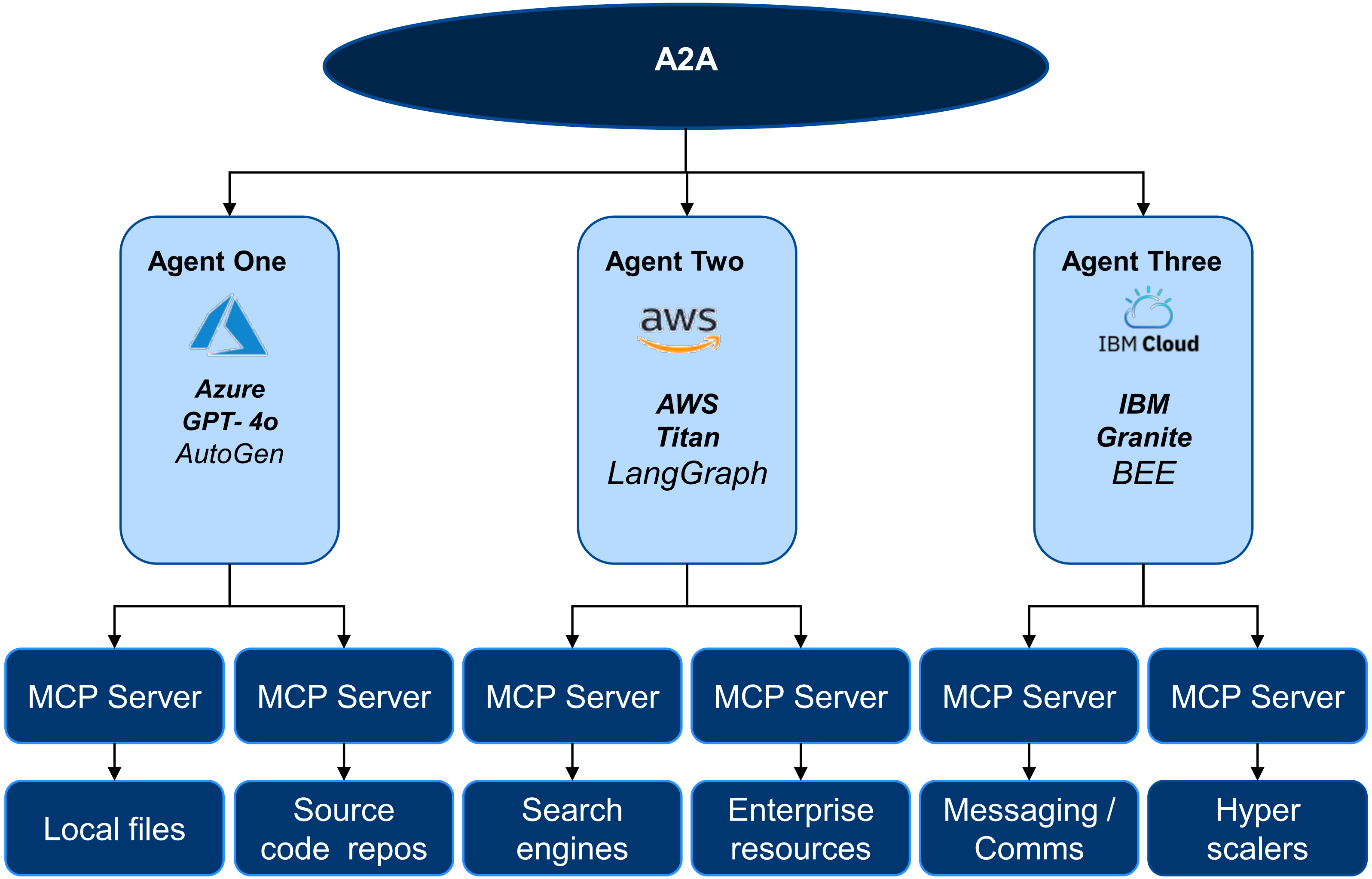
A2A

- Standardise communication between agents over secure web protocols.
- Works like a conference room for agents sharing task-oriented information while protecting its own data and processes.
- Focus on long running workflows driven by agents sharing context in conversational style.

MCP

- Standardize tool usage leveraging common communication protocol with your data products over secure protocols like HTTPS.
- Structured input and output based on schemas.
- Focus on synchronous short lived API calls similar to calling REST APIs based on request/response interactions.

Communication protocols: A2A vs MCP



A2A

Agent One

Azure
GPT- 4o
AutoGen

Agent Two

AWS
Titan
LangGraph

Agent Three

IBM
Granite
BEE

MCP Server

Local files

MCP Server

Source code repos

MCP Server

Search engines

MCP Server

Enterprise resources

MCP Server

Messaging / Comms

MCP Server

Hyper scalers

A2A

- Standardise communication between agents over secure web protocols.
- Works like a conference room for agents sharing task-oriented information while protecting its own data and processes.
- Focus on long running workflows driven by agents sharing context in conversational style.

MCP

- Standardize tool usage leveraging common communication protocol with your data products over secure protocols like HTTPS.
- Structured input and output based on schemas.
- Focus on synchronous short lived API calls similar to calling REST APIs based on request/response interactions.

Best practices for deploying AI agents at scale

How can an enterprise manage 100s of autonomous agents?

Flexible tools for developing AI agents

- Low-code tools for knowledge workers
- Development frameworks and APIs for coders

Management and governance

- Lifecycle management
- Collect agent decision telemetry data
- Monitoring and alerts

Security

- Manage authentication across systems
- Secure data exchange between tools and services
- Monitor for attacks and vulnerabilities

Guardrails and Explainability

- Set conditions for human intervention
- Guardrails to ensure agents meet expectations
- All agent actions and decisions to be explainable

Performance evaluation

Why LLM and agent’s evaluation matters?

Task Performance

1. How well does the model handle our specific use case?
2. How to evaluate prompt engineering and hyperparameter tuning outcomes?

Risk Management

1. Can we trust that the model’s output doesn’t contain bias or harmful content?
2. Can we switch to another LLM provider without jeopardizing performance?

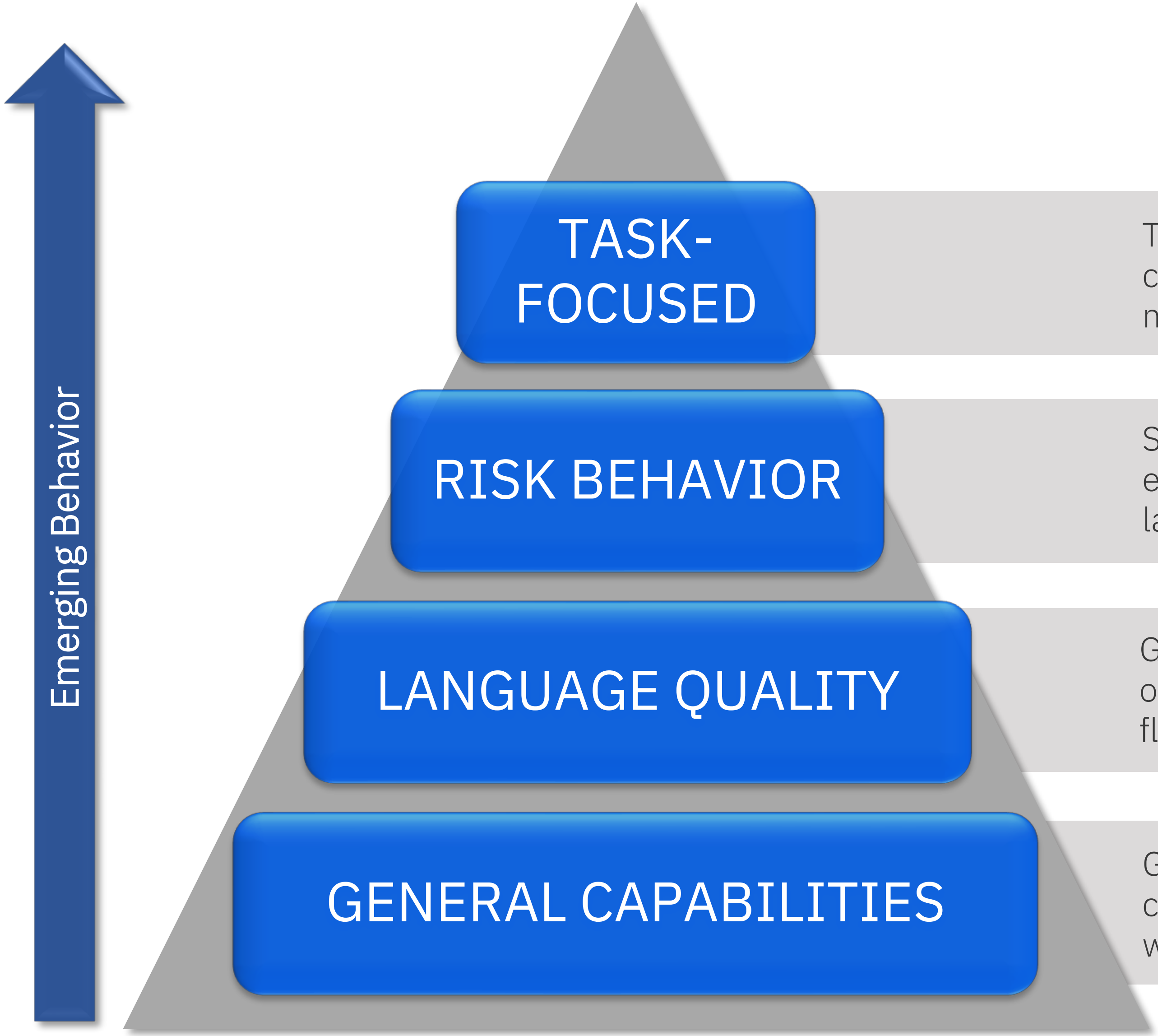
LLMops

1. Models and data can change over time – does this impact our use case?
2. How to best automate the evaluation and incorporate it in our Ops practice?

Cross-functional benchmark

1. How to ensure consistent performance in multilingual applications?
2. How to compare performance across tasks?

Levels of LLM evaluation



Emerging Behavior

TASK-FOCUSED

Targeted metrics selected with respect to the given use case the LLM is applied to. A custom evaluation data needs to be used.

RISK BEHAVIOR

Strategies evaluating potential risk behavior of the LLM, either use-case-specific or general, such as harmful language detection, demographic bias.

LANGUAGE QUALITY

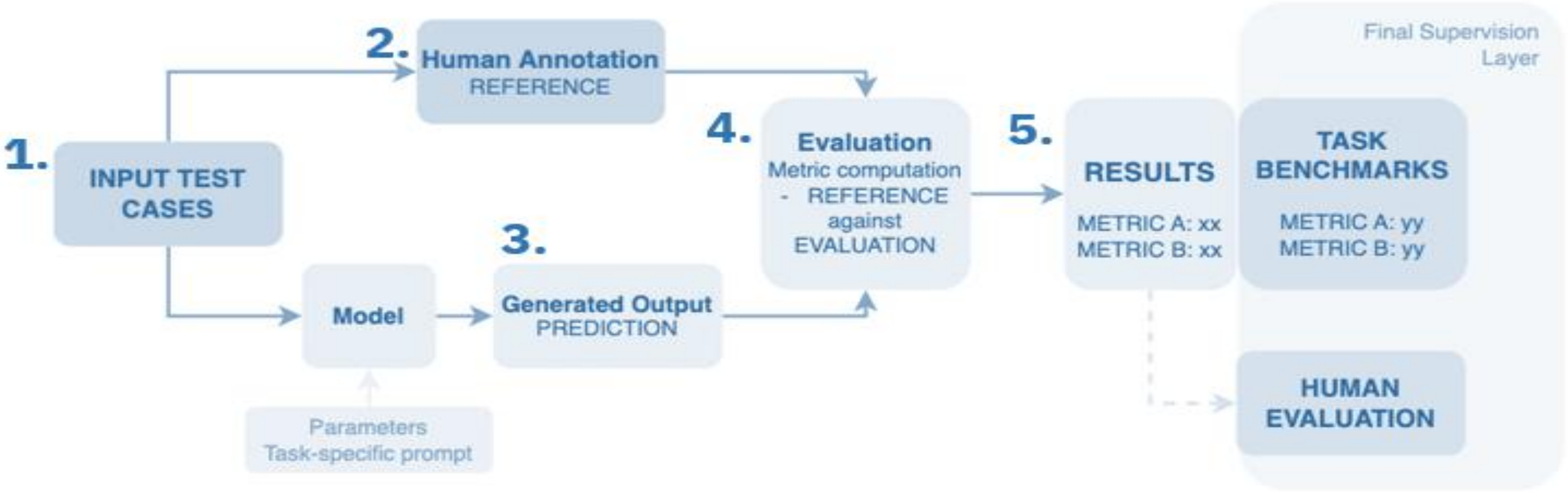
Generic metrics aimed at evaluating the generated output's text quality or general text properties, such as fluency (perplexity), grammar, readability.

GENERAL CAPABILITIES

Generic evaluations on a broad scale of LLMs' capabilities. Crucial for building LLMs, not sufficient when assessing performance on a specific task.

Task-focused model evaluation: General process

1. Collect a list of test records (Custom datasets or OSS datasets)
2. Gather expected output (Ground truth)
3. Generate model output
4. Compare model output with the expected output
5. Select a metric and use it to measure the overall performance



Evaluation metrics for model’s performance measurement

Evaluation Task	Top 3 Common Metrics	Additional Common Metrics	Benchmark Data set
Image Classification	Accuracy, Precision, Recall	F1 Score, Area Under ROC Curve (AUC), Confusion Matrix	CIFAR-10, CIFAR-100, ImageNet, MNIST
Object Detection	Average Precision (AP), Mean Average Precision (mAP), Intersection over Union (IoU)	Precision-Recall Curve, Recall at 10 False Positives (R@10FP), Mean Average Recall (mAR)	COCO, Pascal VOC, KITTI, WIDER Face
Semantic Segmentation	Intersection over Union (IoU) Mean Accuracy	F1 Score, Precision, Recall, Pixel Error Rate	Cityscapes, Pascal VOC, ADE20K, CamVid
Instance Segmentation	Average Precision (AP), Mean Average Precision (mAP), Intersection over Union (IoU)	F1 Score, Precision, Recall, Pixel Accuracy	COCO, Cityscapes, Pascal VOC
Text Summarization	ROUGE (Recall-Oriented Understudy for Gisting Evaluation), Cosine Similarity	F1 Score, Precision, Recall, Human Evaluation	CNN/DailyMail, Gigaword, XSum
Machine Translation	BLEU	TER (Translation Error Rate), GTM (Global Translation Measurement), NIST (National Institute of Standards and Technology)	WMT, IWSLT, TED Talks
Speech Recognition	Word Error Rate (WER), Character Error Rate (CER), Sentence Error Rate (SER)	Phoneme Error Rate (PER), Frame Error Rate (FER), Confusion Matrix	LibriSpeech, Common Voice, TED-LIUM
Sentiment Analysis	Accuracy, F1 Score, Area Under ROC Curve (AUC)	Precision, Recall, Cohen's Kappa, Pearson Correlation Coefficient	IMDB, Amazon Reviews, Yelp Reviews, SST
Topic Modeling	Coherence Score, Perplexity	Topic Homogeneity, Topic Separation, Topic Coherence	20 Newsgroups, Reuters, NIPS
Named Entity Recognition	F1 Score, Precision, Recall	Micro-Averaged F1 Score, Macro-Averaged F1 Score, Accuracy	CoNLL-2003, OntoNotes, ACE 2005
Question Answering	F1 Score, Exact Match (EM), Precision	Recall, BLEU, ROUGE	SQuAD, TriviaQA, BioASQ

Benchmarks: Comparing models

- **Benchmarks** are a **collection of evaluation tasks** that **form a framework to compare different models**.
- They are **informative** and serve well as a method to **compare models against a baseline**.

General benchmarks are great, BUT…

They don't measure the model performance on
- your use cases,
- your prompting,
- your fine-tuning!

Use custom benchmark dataset with your own data as well!!!

Building a simple agentic AI workflow with Semantic Kernel

Semantic Kernel

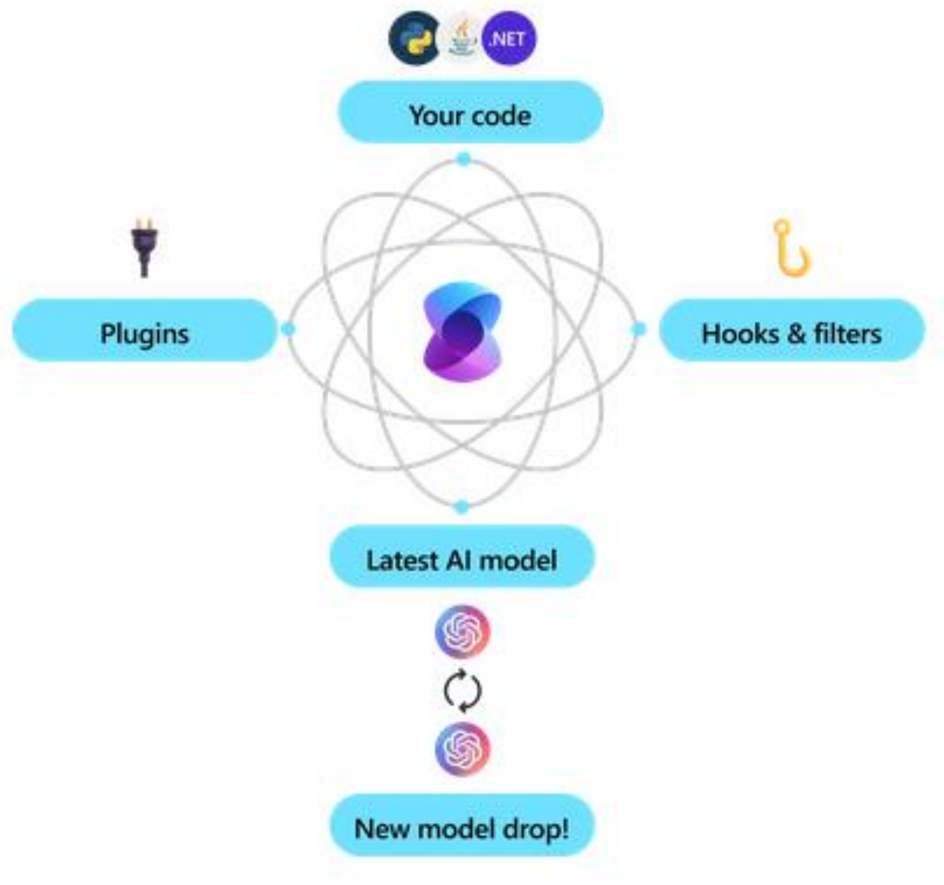
- is a **lightweight, open-source development kit** for integrating AI models into **C#, Python, or Java** codebases.
- acts as **middleware** for rapid delivery of enterprise-grade AI solutions.

Flexible, Modular & Observable

Security Enhancement with Telementry, Hooks & Filters

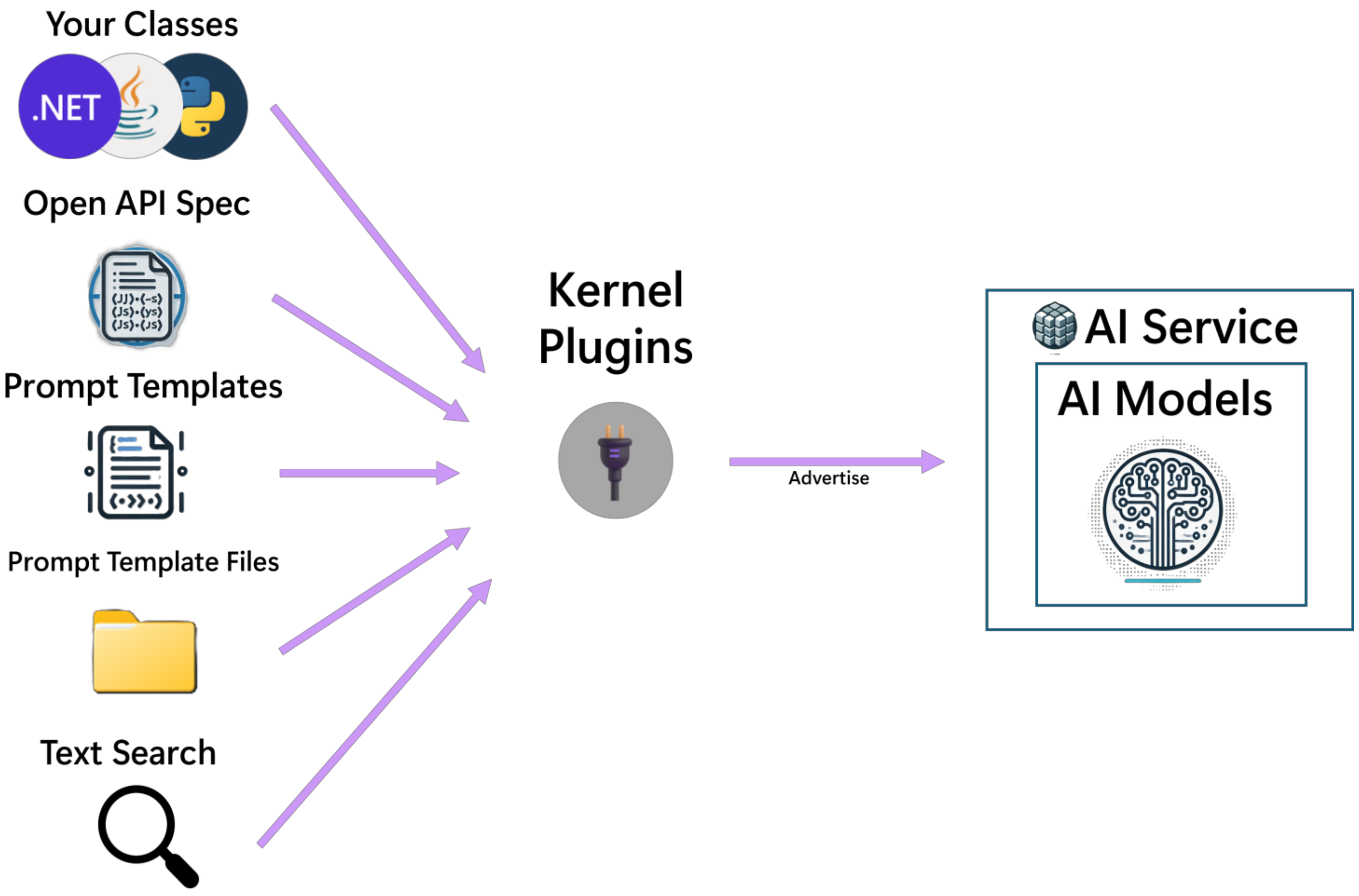
Future-proof Design with Easy Model Updates

Extensible APIs



Resource: Semantic Kernel Components

Key components of Semantic Kernel



Your Classes

.NET

Open API Spec

Prompt Templates

Prompt Template Files

Text Search

Kernel Plugins

Advertise

AI Service

AI Models

- AI Service Connectors
- Vector Store (Memory) Connectors
- Functions and Plugins
- Prompt Templates
- Filters

AI Service Connectors

Abstraction Layer

Provides a common interface for multiple AI service types from various providers.

Supported Services

Chat Completion
Text Generation
Embedding Generation
Text to Image
Image to Text
Text to Audio
Audio to Text

Default Behavior

When registered with the Kernel, Chat Completion or Text Generation services are used by default for method calls.

Manual Use Required

Other supported services must be explicitly called; they are not used automatically.

Vector Store (Memory) Connectors

Abstraction Layer

Provides a common interface for vector stores from various providers.

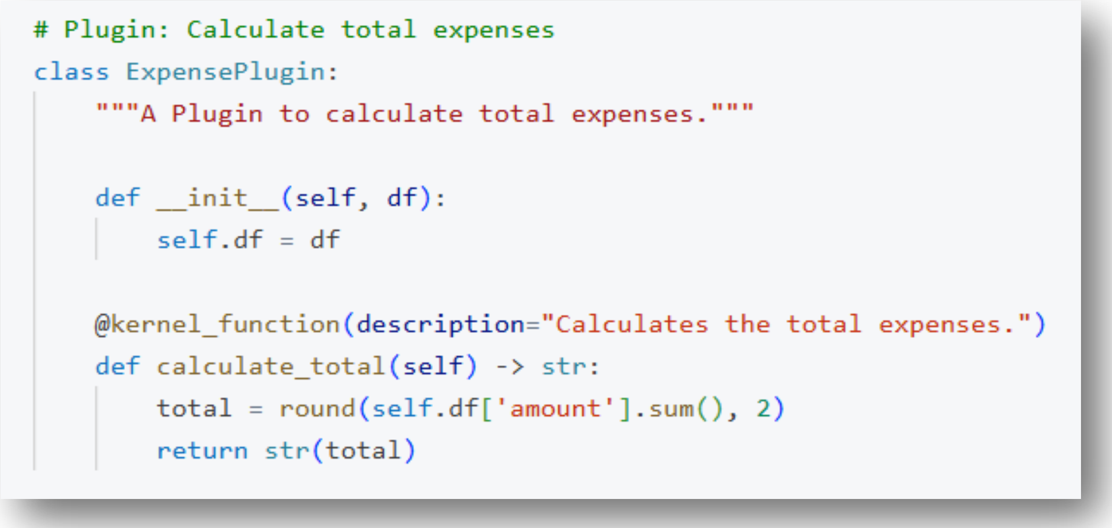
Vector search plugin

Can be easily exposed as a plugin to the Kernel.

No automatic usage

Registered vector stores are not used automatically by the Kernel.

Functions and Plugins



```python
# Plugin: Calculate total expenses
class ExpensePlugin:
    """A Plugin to calculate total expenses."""

    def __init__(self, df):
        self.df = df

    @kernel_function(description="Calculates the total expenses.")
    def calculate_total(self) -> str:
        total = round(self.df['amount'].sum(), 2)
        return str(total)
```

1. Definition

Plugins are named function containers that can hold one or more functions.

2. Registration

Plugins can be registered with the kernel.
This registration enables two main functionalities:

- **AI Integration**: Advertise plugins to the chat completion AI for invocation.
- **Template Availability**: Make plugins callable from templates during rendering.

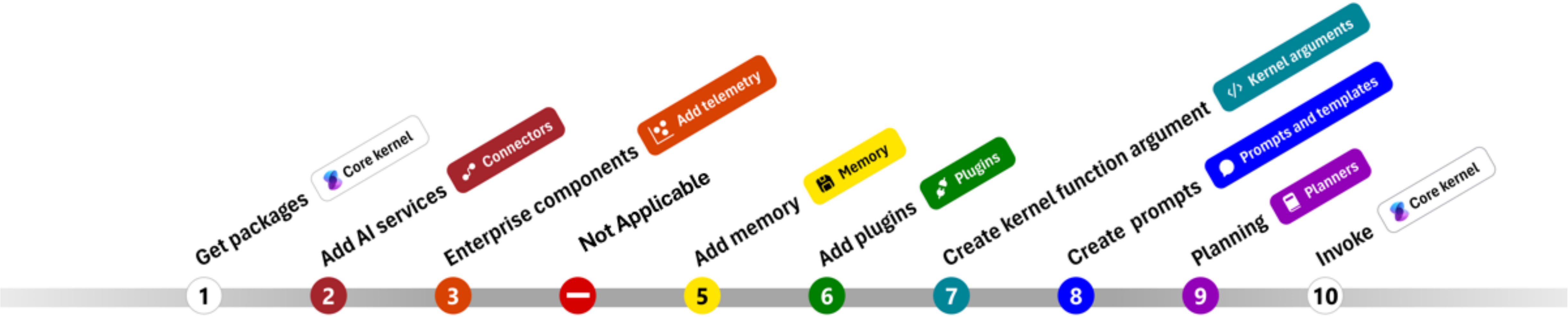
3. Function creation

Functions can be derived from various sources, including:

- Native code
- OpenAPI specifications
- ITextSearch implementations for RAG
- Prompt templates

Building an Agentic AI workflow

Step by step guide to get started
building enterprise apps with Semantic Kernel



1 Get packages — Core kernel

2 Add AI services — Connectors

3 Enterprise components — Add telemetry

Not Applicable

5 Add memory — Memory

6 Add plugins — Plugins

7 Create kernel function argument — Kernel arguments

8 Create prompts — Prompts and templates

9 Planning — Planners

10 Invoke — Core kernel

McDonald's
SO GOOD EVEN COMPUTERS WANT A BYTE

Demo:

Building a simple Expense AI agent

Use case overview

AI Agent overview

Build a simple Expense AI agent to perform essential tasks.

Key Functions

1. Calculate total expenses.
2. Simulate sending an email.

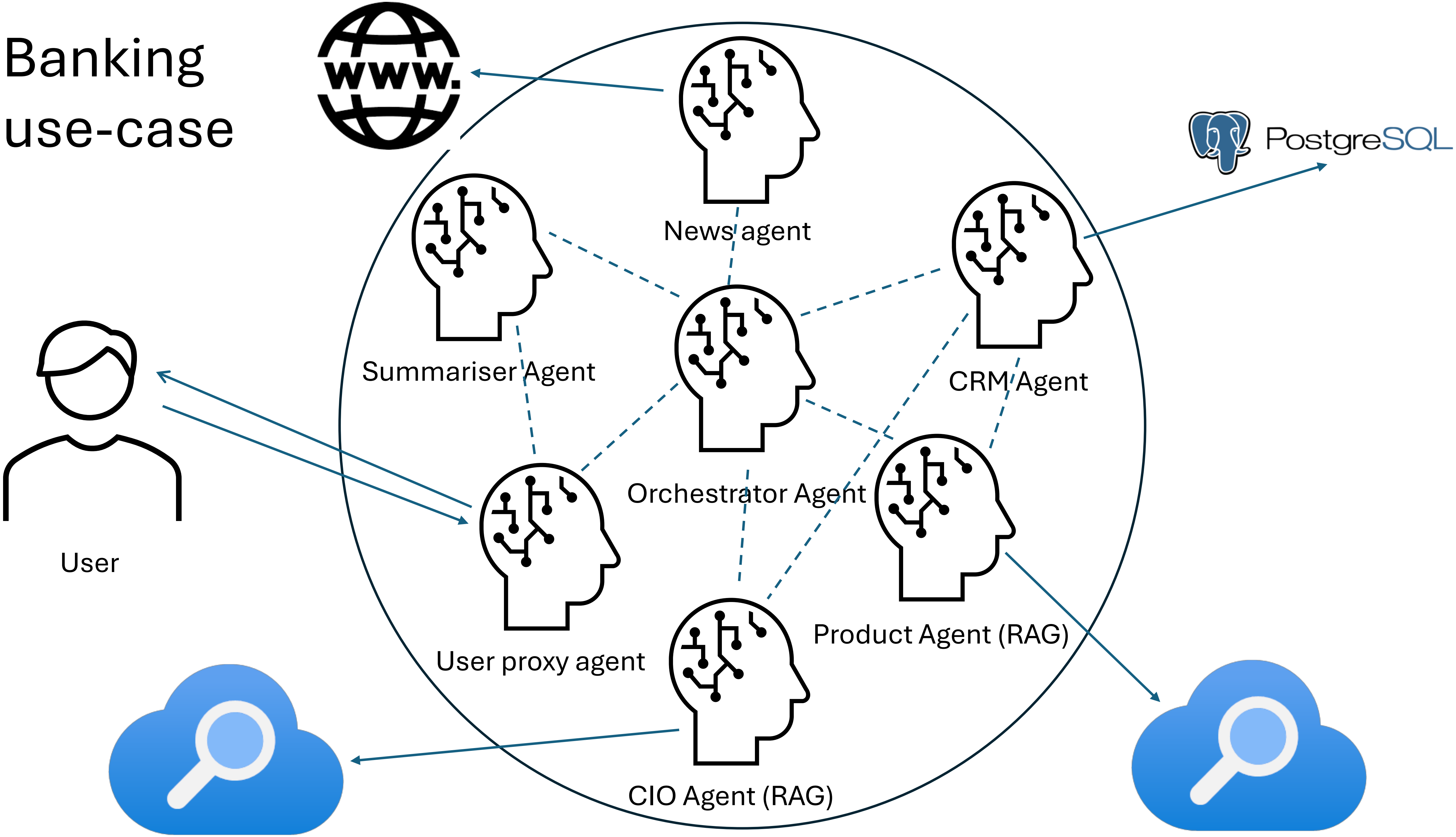
Expected Outcomes

Automate tasks in the expense management workflow and improve the processes.

Case study 1:

Banking agentic CRM

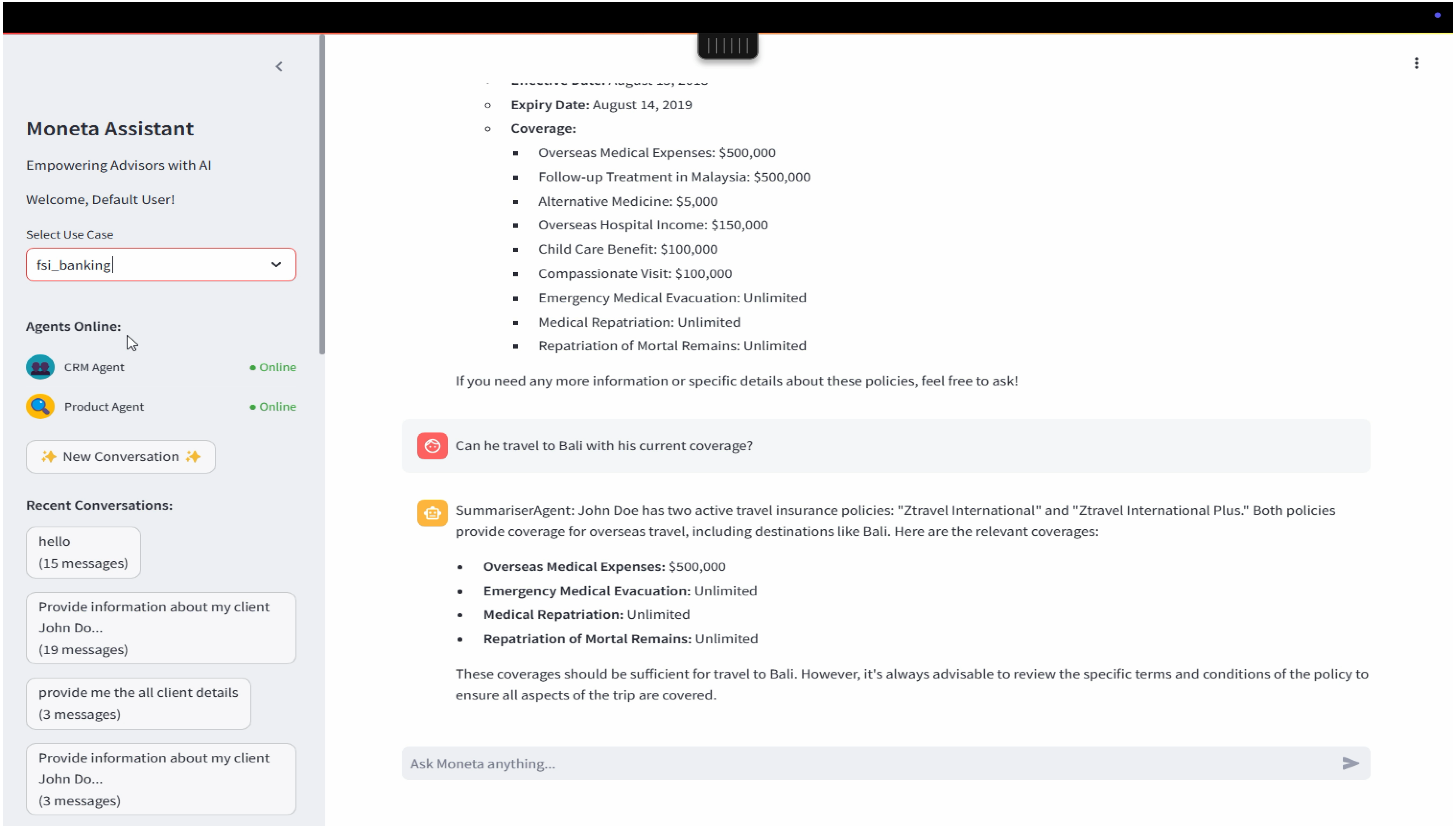
Banking use-case



```
crm_agent = self.create_agent(service_id="gpt-4o",
                              kernel=self.kernel,
                              definition_file_path="sk/agents/banking/crm.yaml")
funds_agent = self.create_agent(service_id="gpt-4o",
                              kernel=self.kernel,
                              definition_file_path="sk/agents/banking/funds.yaml")
cio_agent = self.create_agent(service_id="gpt-4o",
                              kernel=self.kernel,
                              definition_file_path="sk/agents/banking/cio.yaml")
news_agent = self.create_agent(service_id="gpt-4o",
                              kernel=self.kernel,
                              definition_file_path="sk/agents/banking/news.yaml")
responder_agent = self.create_agent(service_id="gpt-4o",
                              kernel=self.kernel,
                              definition_file_path="sk/agents/banking/responder.yaml")

agents = [crm_agent, funds_agent, cio_agent, news_agent, responder_agent]

agent_group_chat = AgentGroupChat(
    agents=agents,
    selection_strategy=self.create_selection_strategy(agents, responder_agent),
    termination_strategy=self.create_termination_strategy(
        agents=[funds_agent, crm_agent, responder_agent],
        final_agent=responder_agent,
        maximum_iterations=8
    )
)
```



Moneta Assistant
Empowering Advisors with AI
Welcome, Default User!
Select Use Case
fsi_banking
Agents Online:
CRM Agent ● Online
Product Agent ● Online
✨ New Conversation ✨
Recent Conversations:
hello (15 messages)
Provide information about my client John Do... (19 messages)
provide me the all client details (3 messages)
Provide information about my client John Do... (3 messages)

- **Expiry Date:** August 14, 2019
- **Coverage:**
  - Overseas Medical Expenses: $500,000
  - Follow-up Treatment in Malaysia: $500,000
  - Alternative Medicine: $5,000
  - Overseas Hospital Income: $150,000
  - Child Care Benefit: $100,000
  - Compassionate Visit: $100,000
  - Emergency Medical Evacuation: Unlimited
  - Medical Repatriation: Unlimited
  - Repatriation of Mortal Remains: Unlimited

If you need any more information or specific details about these policies, feel free to ask!

Can he travel to Bali with his current coverage?

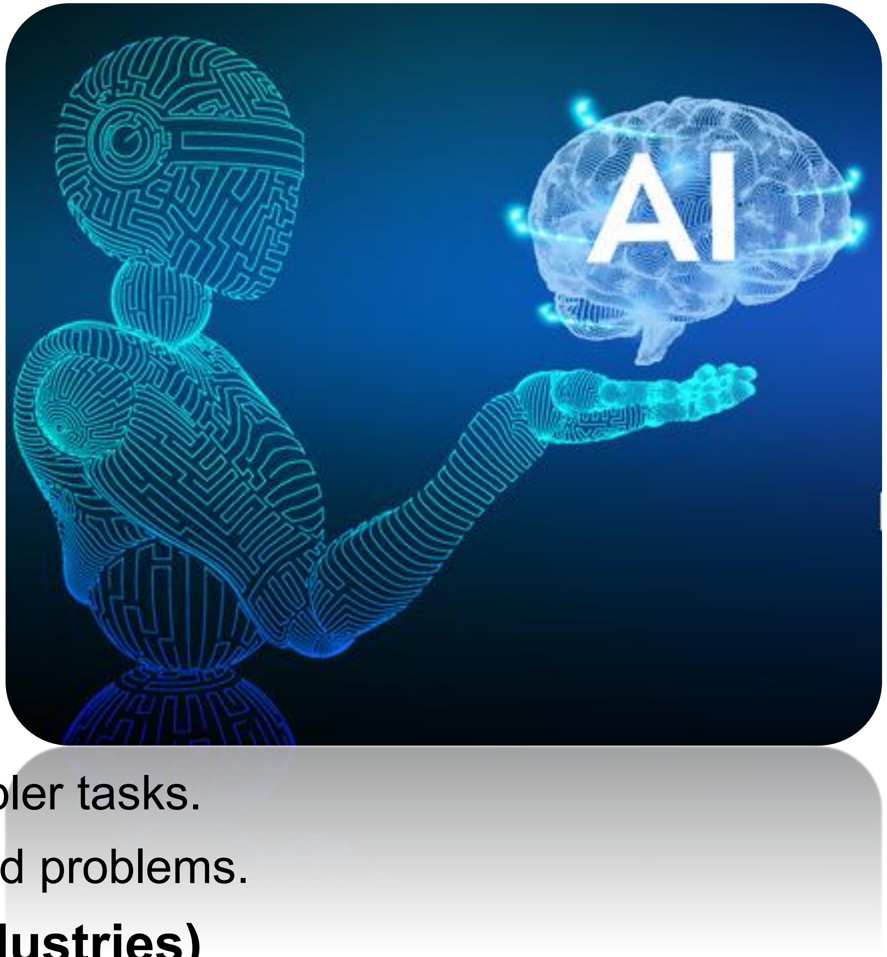
SummariserAgent: John Doe has two active travel insurance policies: "Ztravel International" and "Ztravel International Plus." Both policies provide coverage for overseas travel, including destinations like Bali. Here are the relevant coverages:

- **Overseas Medical Expenses:** $500,000
- **Emergency Medical Evacuation:** Unlimited
- **Medical Repatriation:** Unlimited
- **Repatriation of Mortal Remains:** Unlimited

These coverages should be sufficient for travel to Bali. However, it's always advisable to review the specific terms and conditions of the policy to ensure all aspects of the trip are covered.

Future of Agentic AI

Agentic AI: What is coming next?



1. **Better understanding of language**
   - AI will understand words and context more deeply.
   - It will adjust actions to make conversations more natural.
2. **Working together (Multi-agent collaboration)**
   - AI agents will work together like human teams.
   - They will share tasks to reach complex goals.
3. **Clear and honest AI (Explainable AI – XAI)**
   - AI will explain how it makes decisions.
   - This builds trust and ensures fairness.
4. **Smaller and smarter models (SLMs vs. LLMs)**
   - *Small Language Models (SLMs):* Faster and cheaper for simpler tasks.
   - *Large Language Models (LLMs):* Better for complex and broad problems.
5. **Expended application (Agentic AI will impact many industries)**
   - *Education:* Helps create personalized learning for each student.
   - *Cybersecurity:* Detects and responds to online threats in real time.

How Agentic AI will transform business and organizations?

1. **More efficiency**
   - Automates repetitive tasks to save time and reduce costs.
2. **Independent decision-making**
   - Agentic AI will make some business decisions.
3. **New ways of organizing work**
   - Companies may shift to flexible AI “networks” focused on results.
4. **Smarter technology connections**
   - Agentic AI will work both on devices (edge) and in the cloud for quicker responses.
5. **Better customer experience**
   - AI agents will manage most customer service interactions.
   - They will offer personalized, proactive help, while humans stay important for empathy and connection.

Key takeaways

Best practices for building Agentic AI

1. **Start simple – don’t over-engineer.**
    - Not every problem needs an agent. Apply agentic design only where it adds real value.
2. **Understand the fundamentals before going multi-agent.**
    - Understand key ideas like **RAG** (Retrieval-Augmented Generation) and **function calling**.
    - Begin with single-agent systems before trying multi-agent setups.
3. **Use agents to scale high-value use cases.**
    - Use agents to automate important or repetitive tasks that save time and effort.
4. **Leverage the right tools and frameworks.**
    - Get hands-on with proven frameworks like **LanGraph** or **Semantic kernel** to build and orchestrate agents effectively.
5. **Plan for security and infrastructure.**
    - Make sure your system is safe, with secure data access and reliable setup from the start.
6. **Follow structured approaches.**
    - Use structured methods to build scalable, flexible, and secure AI applications.

Questions & Answer

We are looking forward to your questions!

Thank you!

IBM®