

# INTRO TO AI AGENTS & DATA COLLECTION

## A HANDS-ON WORKSHOP

Deepak Babu  
Founder/CEO  
Sabhya Technologies



February 11, 2025



# Agenda

- Introduction to AI Agents
- Data Collection Strategies
- Ethical Considerations and Challenges
- Tools & Techniques
- Hands-on Workshop
- Key Takeaways

# INTRODUCTION TO AI AGENTS

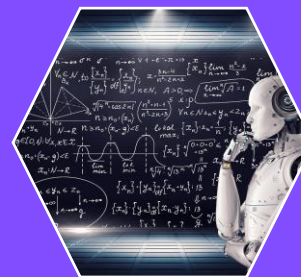
# HOW DID WE GET HERE?



Machine learning  
algorithms



Computer vision  
systems



Natural Language  
Processing (NLP)

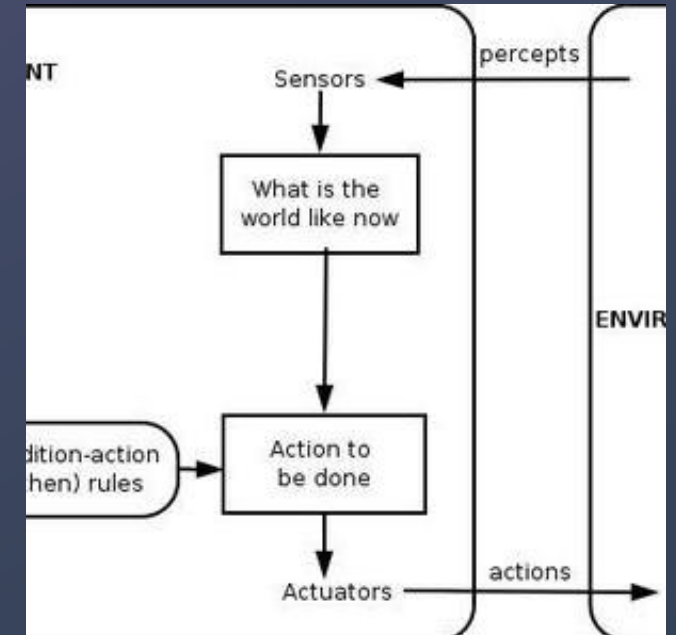


Generative AI

# DEFINITION

“An artificial intelligence (AI) agent is a software program that can interact with its environment,  
 -collect data,  
 -and use the data to  
 -perform self-determined tasks to  
 meet predetermined goals.”

• +  
○



# Type of AI Agents

## Simple Reflex Agents

These agents act solely on the basis of current percepts without considering past experiences. They use **condition-action rules** (e.g., a thermostat turning on the heater when the temperature drops).

## Model-Based Reflex Agents

These agents maintain an internal model of the world, allowing them to track changes and make better decisions. They consider the **state of the environment** when choosing actions.

## Goal-Based Agents

These agents take actions **based on predefined goals**. They evaluate different possibilities and select actions that help them achieve their objectives (e.g., GPS navigation systems).

## Utility-Based Agents

These agents go beyond goal-based behavior by incorporating a utility function, which helps them make optimal decisions by **considering different outcomes and their desirability** (e.g., self-driving cars optimizing for speed and safety).

## Learning Agents

These agents improve their performance over time by **learning from experiences**. They use machine learning techniques to adapt and refine their behavior (e.g., AI chatbots, recommendation systems).

# RPA/BOT VS AI AGENT

**Rule-Based**

**Non-Learning**

**Designed For Repetitive Tasks**

**No Intelligence**

Data entry, invoice processing, form filling, rule-based workflows.

**Adaptive**

**Intelligent**

**Capable Of  
Decision-Making**

Chatbots, self-driving cars, fraud detection, recommendation systems.



# DATA COLLECTION STRATEGIES

- **Web Scraping:** Use cases and tools
- **Sensor Data Integration:** IoT applications
- **Human-in-the-Loop Techniques:** •  
Crowdsourcing and feedback systems ○





# INTERNET OF THINGS (IOT)

- **Smart sensors:** Collecting real-time data on equipment performance, temperature, pressure, etc.
- **Connected devices:** Integrating data from machines, robots, and other equipment.
- **Edge computing:** Processing data at the source for immediate insights and decision-making.

IOT FOR **BEYOND**  
REAL-TIME DATA  
CAPTURE

# SUPERVISED AND UNSUPERVISED LEARNING

## Supervised

Goal: Predict outcomes based on **past examples**.

### Common Algorithms

- Linear Regression
- Decision Trees
- Support Vector Machines (SVM)
- K-Nearest Neighbor

## Unsupervised

Goal: Discover **hidden structures** in data.

### Common Algorithms

- K-Means Clustering
- Principal Component Analysis (PCA)
- Hierarchical Clustering
- Autoencoders

# ETHICAL CONSIDERATIONS IN DATA COLLECTION

Privacy and  
consent

Bias in data

Data  
security

Legal &  
Regulatory  
Compliance

• +  
○



**WHAT'S  
IMPORTANT  
TO YOU?**

# CHALLENGES IN DATA COLLECTION

+  
•  
○



Data Quality &  
Accuracy



Data Integration  
& Compatibility



Scalability &  
Storage



Cost of Data  
Acquisition



Regulatory &  
Ethical  
Concerns



Data Privacy &  
Security

# TOOLS & TECHNIQUES

# TOOLS



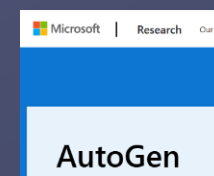
**LangChain,  
LangSmith, LangGraph**



**LlamaIndex**



**Semantic  
Kernel**



**AutoGen**

**Azure AI Bot  
Service**



**Microsoft Bot  
Framework**



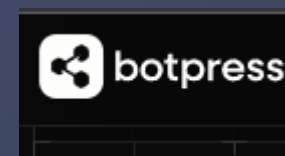
**Langflow**



**Rasa**



**Prompt  
flow**



**Botpress**



**Haystack**

# SEMANTIC KERNEL

// Create a kernel with Azure OpenAI chat completion

```
var builder = Kernel.CreateBuilder().AddAzureOpenAIChatCompletion(modelId,
endpoint, apiKey);
```

// Enable planning

```
OpenAIPromptExecutionSettings openAIPromptExecutionSettings = new() {
FunctionChoiceBehavior = FunctionChoiceBehavior.Auto() };
```

// Get the response from the AI

```
var result = await chatCompletionService.GetChatMessageContentAsync( history,
executionSettings: openAIPromptExecutionSettings,
kernel: kernel);
```

Choose a programming language

C#

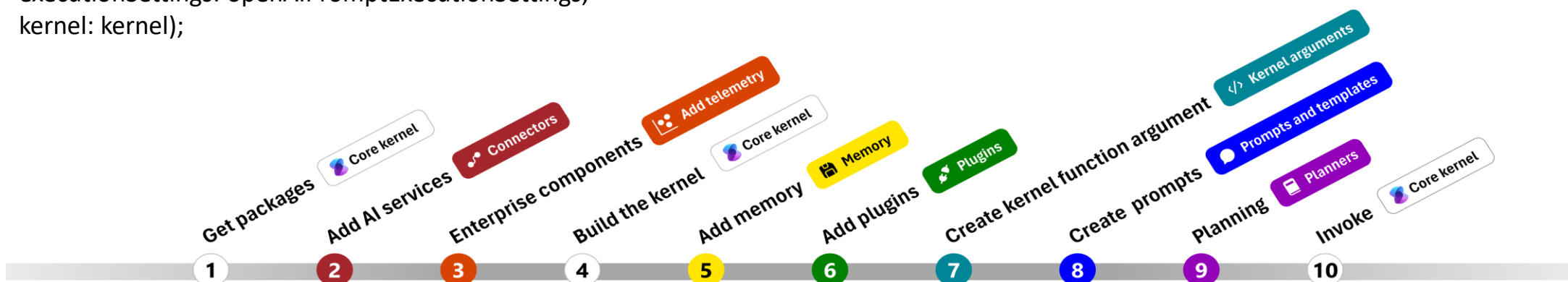
Python

Java

<https://github.com/microsoft/semantic-kernel>

*Install the following .NET dependencies*

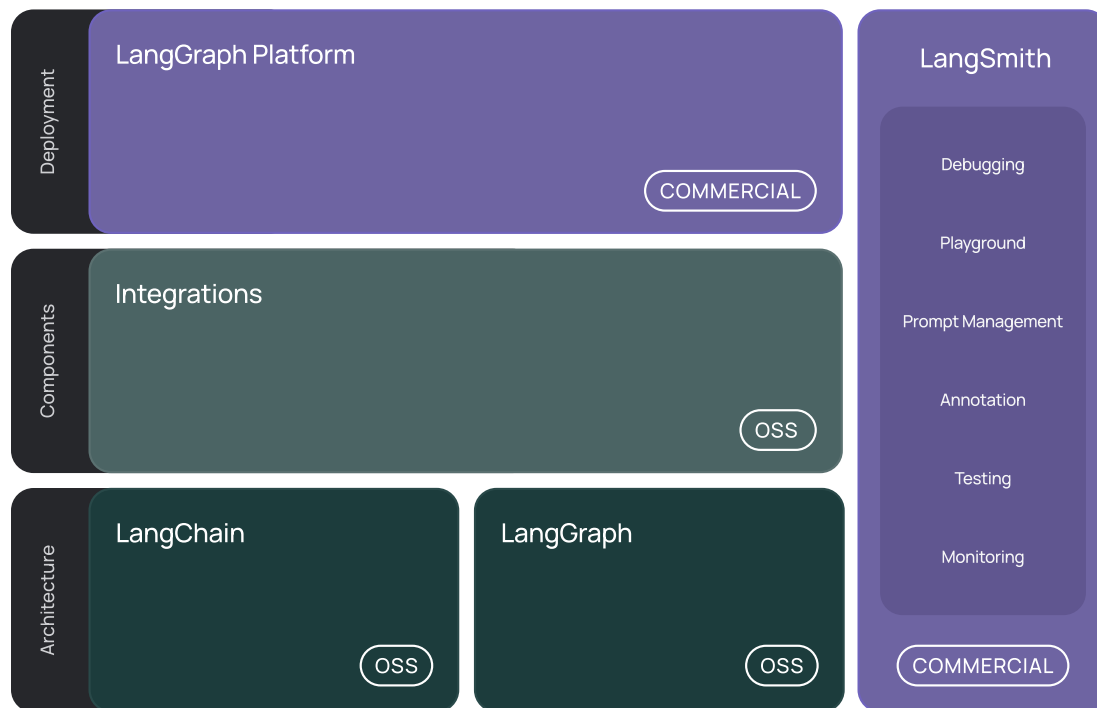
```
dotnet add package Microsoft.SemanticKernel
dotnet add package Microsoft.Extensions.Logging
dotnet add package Microsoft.Extensions.Logging.Console
```



<https://learn.microsoft.com/en-us/semantic-kernel/get-started/quick-start-guide?pivots=programming-language-csharp>



# LANGCHAIN, LANGGRAPH, LANGSMITH



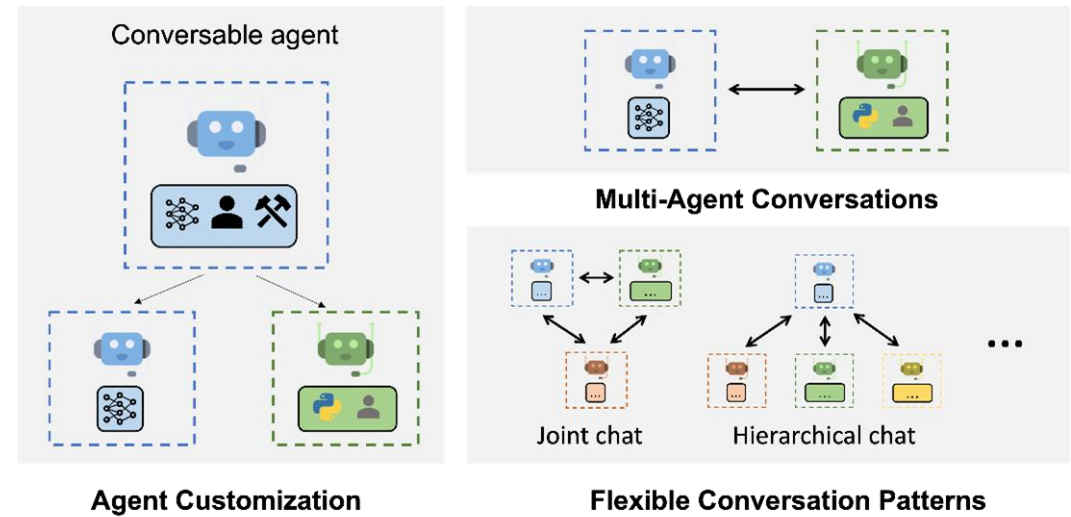
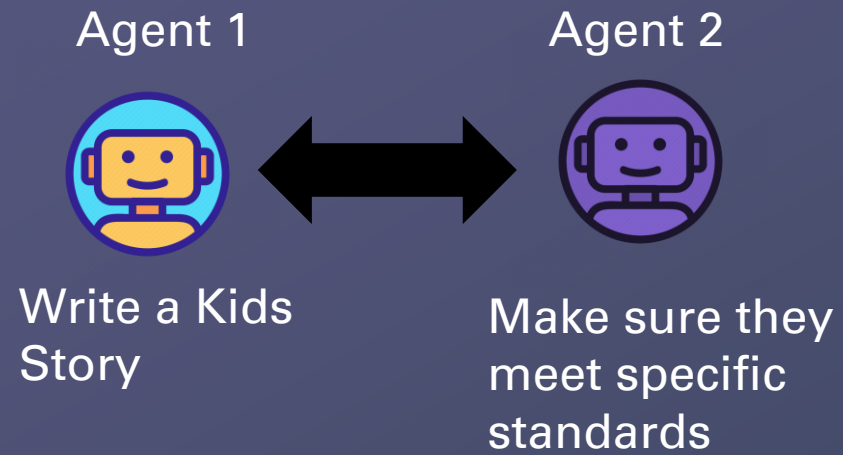
```
# Import relevant functionality
from langchain_anthropic import ChatAnthropic
from langchain_community.tools.tavily_search import TavilySearchResults
from langchain_core.messages import HumanMessage
from langgraph.checkpoint.memory import MemorySaver
from langgraph.prebuilt import create_react_agent

# Create the agent
memory = MemorySaver()
model = ChatAnthropic(model_name="claude-3-sonnet-20240229")
search = TavilySearchResults(max_results=2)
tools = [search]
agent_executor = create_react_agent(model, tools, checkpoint=memory)

# Use the agent
config = {"configurable": {"thread_id": "abc123"}}
for chunk in agent_executor.stream(
    {"messages": [HumanMessage(content="hi im bob! and i live in sf")]}, config
):
    print(chunk)
    print("----")

for chunk in agent_executor.stream(
    {"messages": [HumanMessage(content="whats the weather where I live?")]}, config
):
    print(chunk)
```

# AUTOGEN



# AUTOGEN FOR .NET

```
// create teacher agent
// teacher agent will create math questions
var teacher = new OpenAIChatAgent(
    chatClient: gpt4oMini,
    name: "teacher",
    systemMessage: @"You are a teacher that
create pre-school math question for student
and check answer.
If the answer is correct, you stop the
conversation by saying [COMPLETE].
If the answer is wrong, you ask student to
fix it.")
```

1

```
// start the conversation
var conversation = await student.InitiateChatAsync(
    receiver: teacher,
    message: "Hey teacher, please create math
question for me.",
    maxRound: 10);
```

3

```
// create student agent
// student agent will answer the math questions
var student = new OpenAIChatAgent(
    chatClient: gpt4oMini,
    name: "student",
    systemMessage: "You are a student that answer
question from teacher")
    .RegisterMessageConnector()
    .RegisterPrintMessage();
```

2

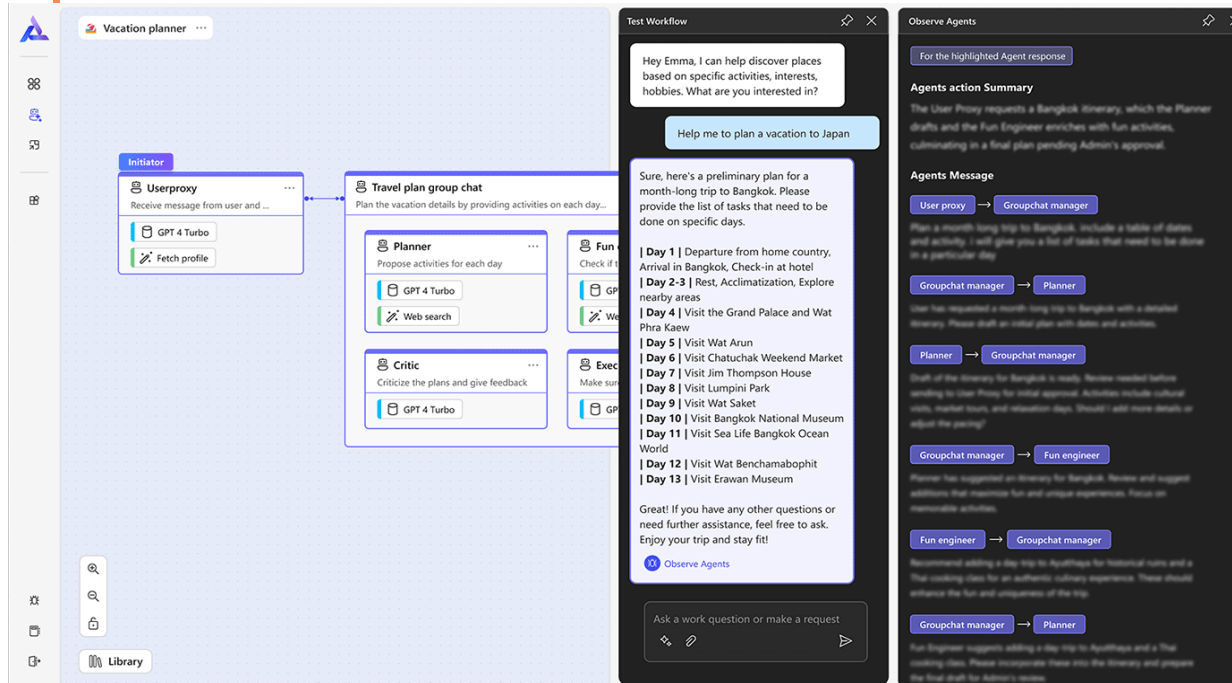
```
/ output
// Message from teacher
// -----
// content: Of course!Here's a math question
for you:
// What is 2 + 3 ?
// -----
// Message from student
// -----
// content: The sum of 2 and 3 is 5.
```

4

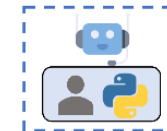
19

# AUTOGEN STUDIO

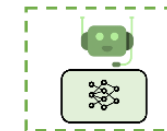
## A LOW-CODE INTERFACE



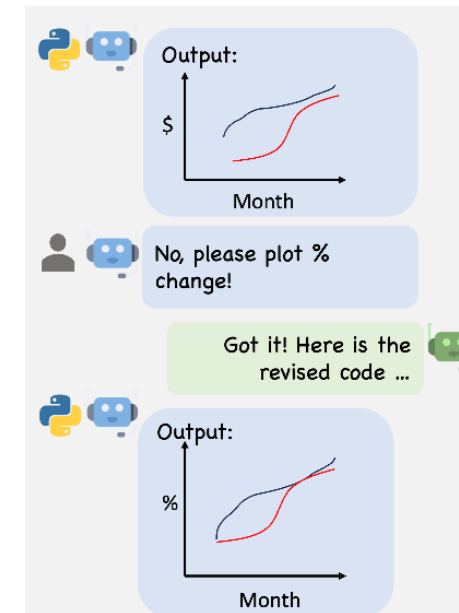
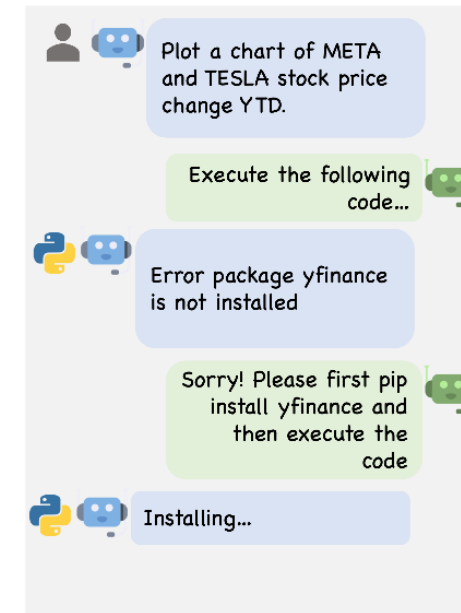
Uses shell with human-in-the-loop  
User Proxy Agent



Assistant Agent



LLM configured to write python code



## A New Paradigm



# CONVERSATION DRIVEN DEVELOPMENT(CDD)

(CDD) is the process of  
**listening** to your users and  
using those insights to  
**improve** your **AI assistant**



# WORKSHOP

# HANDS-ON WORKSHOP

- Using a Sensor to Collect Data
- Processing and Visualizing Data

## Setup and Tools

- <https://www.docker.com/products/docker-desktop/>
- <https://docs.influxdata.com/influxdb/v2/install/>
- Github: <https://github.com/sabhyadb/workshop.git>
- Sensor hardware



# KEY TAKEAWAYS



Is Your  
Job Safe?



AI is here-  
No need to  
hide



Ride the  
AI Beast



# THANK YOU

Deepak Babu

262-352-3591

[deepak@sabhyatech.com](mailto:deepak@sabhyatech.com)

[www.sabhyatech.com](http://www.sabhyatech.com)