

AI-3026 Azure で AI エージェントを 開発する

Develop AI Agents with the Azure AI Foundry Agent Service

本講義は以下の **Microsoft Learn** 教材に準拠しています。

<https://learn.microsoft.com/ja-jp/collections/j06xt3t1kr4rd0>

公式コレクション

AI-3026 Azure で AI エージェントを開発する

講義用

編集日時 2025/8/5

作成者  Microsoft



本コースの概要・対象者

- Azure AI Foundry Agent ServiceとSemantic Kernelを使用してAIエージェントを開発する方法を解説します。
- 講義後半ではコードを使ったAIエージェント解説方法について解説します。
- 生成AIモデルの基礎知識、基本的なプログラミングのスキルがあると、本コースをスムーズにご受講いただけます。



本コースの構成

<https://learn.microsoft.com/ja-jp/collections/j06xt3t1kr4rd0>

モジュール

AzureでAIエージェント開発を始める

49 分

モジュール

Azure AI Foundry Agent Service を使用して AI エージェントを開発する

55 分

モジュール

カスタム ツールをエージェントに統合する

53 分

Azure AI Foundry Agent Serviceを
使った
AIエージェント開発

モジュール

セマンティック カーネルを使用して AI エージェントを開発する

55 分

モジュール

セマンティック カーネルを使用してマルチエージェント ソリューションを調整する

56 分

Azure AI Foundry Agent Serviceと
Semantic Kernelを組み合わせた
AIエージェント開発

講義

- モジュール1 AzureでAIエージェント開発を始める
- モジュール2 Azure AI Foundry Agent Service を使用して AI エージェントを開発する
- モジュール3 カスタム ツールをエージェントに統合する
- モジュール4: Semantic Kernelを使用して AI エージェントを開発する
- モジュール5: Semantic Kernelを使用してマルチエージェント ソリューションを調整する

モジュール1



AzureでAIエージェント開発を始める

AIエージェント、Azure AI Foundry、Azure AI Foundry Agent Serviceの概要を解説します。ラボでは、Azure AI Foundry Agent Serviceを使用して簡単なAIエージェントを開発します。

AIエージェントとは？

- **AI エージェント**は、特定のタスクを実行し、質問に回答し、ユーザーに代わりプロセスを自動化するために設計されています。
- **AIエージェント**には、単純なチャットボットから、Copilot や、デジタルまたはロボット システムの形で複雑なワークフローを自律的に実行できる高度な AI アシスタントに至るまで、さまざまなものがあります。
- **AIエージェント**は意思決定を行い、ツールを呼び出し、ワークフローに参加します。時には独立して、時には他のエージェントや人間と協力します。

AIエージェントってなんですか？ 初心者でもわかるように短くわかりやすく教えて！



AIエージェントとは、人工知能を利用して特定のタスクや目的を達成するために設計されたプログラムやシステムです。たとえば、チャットボットや音声アシスタントのように、ユーザーの質問に答えたり、仕事を効率化したりします。初心者でも簡単に使える技術が多く、人間のように対話したり決定を手助けすることができます。

...

AIエージェントの中ではGPTを使っていますか？



AIエージェントは、GPTのような自然言語処理モデルを使うことが多いです。GPTは特に、会話や文章生成などのコミュニケーションタスクで強力な性能を発揮します。そのため、質問への回答、情報検索、文章作成を行うAIエージェントの中核技術として利用されています。

...

AIエージェントが普及すると人間の仕事の仕方はどのように変わりますか？



...

AIエージェントが普及すると、人間の仕事には以下のような変化が見込まれます：

1. **効率化**: 繰り返しの多いタスク（データ入力、分析など）はAIに任せることで、人間は創造力や戦略的思考が要求される仕事に集中できるようになります。
2. **職種の多様化**: AI管理・運用に関する新しい職種が誕生し、従来の仕事の在り方が進化します。
3. **技能のシフト**: AIを扱えるスキルや技術が求められるようになり、教育や職業訓練の内容も変化していきます。
4. **協働の強化**: AIは人間のパートナーとして、仕事の質向上や新しいアイデア創出を支援します。

これにより、単なる置き換えではなく、仕事の「質」や「役割」が変化することが期待されています。

Azure AI Foundryとは？

- **Azure AI Foundry** は、エンタープライズ AI 運用、モデルビルダー、およびアプリケーション開発のための統合された Azure サービスとしてのプラットフォーム。
- **Azure AI Foundry** は、運用グレードのインフラストラクチャとわかりやすいインターフェイスを組み合わせたもの。
- **Azure AI Foundry** を使用することで、開発者はインフラストラクチャを管理するのではなく、アプリケーションの構築に集中できる。
- <https://ai.azure.com> からアクセスできる。



Azure AI Foundryとは何ですか？ AIエージェント開発に関係がありますか？



Azure AI Foundryは、MicrosoftのAzureプラットフォームに基づくエコシステムの一部で、人工知能ソリューションを開発・実装するためのツール群を提供します。AIエージェントの開発においては、Azure AI Foundryは以下のような役割を果たします：

- **AIモデルの構築と展開**：大規模なAIモデル（例えばGPTなど）をカスタマイズして特定のアプリケーションに適用可能。
- **クラウドインフラの活用**：データストレージや計算リソースを、Azureのクラウドサービスを通じて効果的に利用。
- **開発ツールの提供**：開発者がAIアプリケーションを迅速に構築できるよう支援。

したがって、Azure AI FoundryはAIエージェントの開発プロセスに必要なプラットフォームとリソースを提供することで、関係性が強いといえます。

Azure AI Foundry Agent Serviceとは？

- **Azure AI Foundry** は、モデル、ツール、フレームワーク、ガバナンスを統合されたシステムに組み合わせてインテリジェントなエージェントを構築するプラットフォームですが、その中心には **Azure AI Foundry Agent Service** があり、開発、デプロイ、運用全体でエージェントの操作を可能にします。
- **Azure AI Foundry Agent Service** は、モデル、ツール、フレームワークなどの **Azure AI Foundry** のコア部分を1つのランタイムに接続します。

Azure AI Foundry Agent Serviceとは何ですか？ AIエージェント開発に使用できますか？



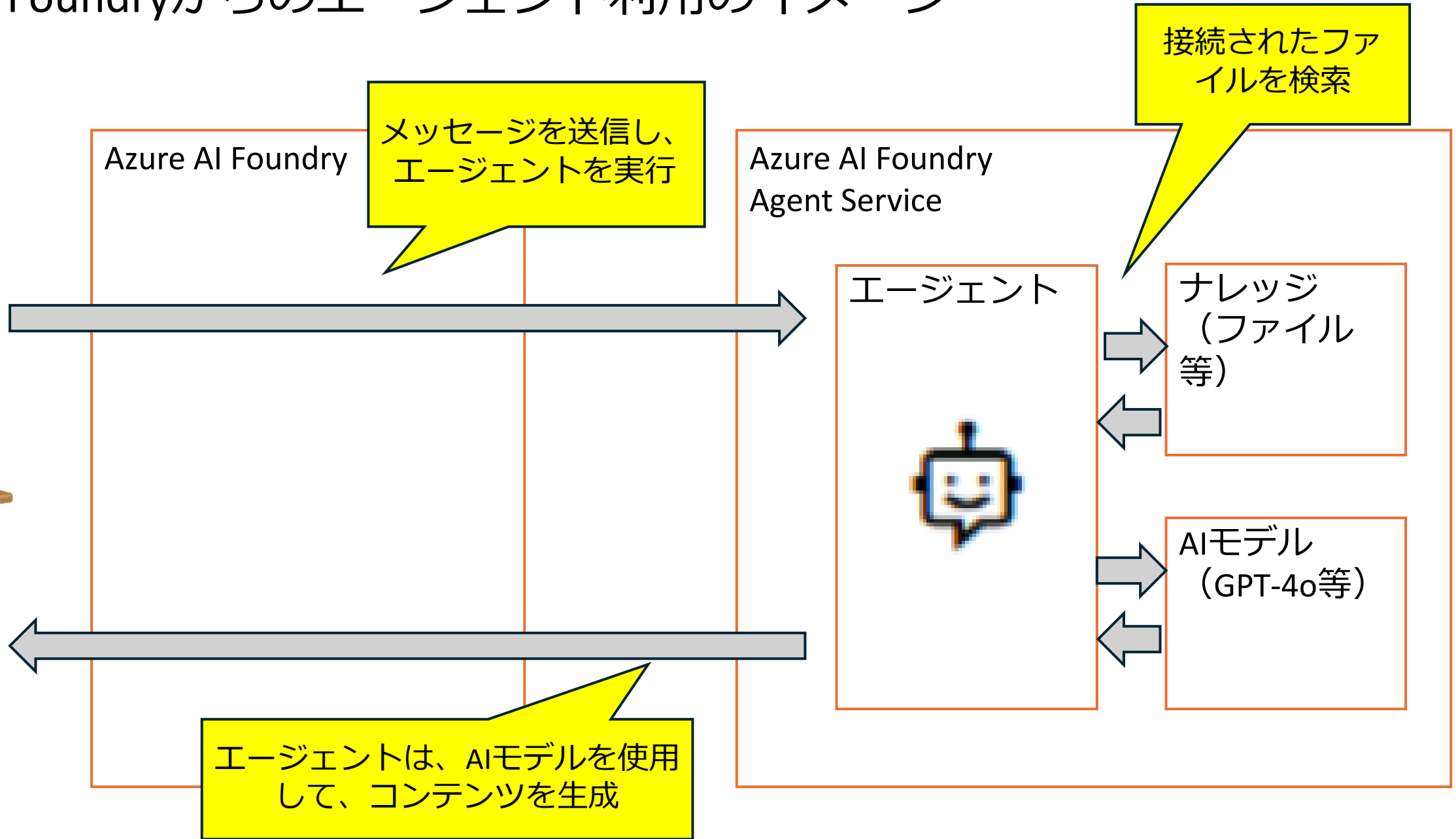
Azure AI Foundry Agent Serviceは、MicrosoftのAzure AI Foundryの一部として提供されるサービスで、AIエージェントの開発と展開をサポートします。このサービスを活用することで、カスタマイズ可能なAIエージェントを簡単に作成して実際のアプリケーションに応用できます。

AIエージェント開発においては、以下のように役立ちます：

- **高度なAIモデルを活用**：Azureが提供するAIモデルを利用し、エージェントにインテリジェントな機能を付加可能。
- **迅速な開発**：クラウドベースのインフラを活用することで、効率的に開発を進められる。
- **カスタマイズ**：特定のニーズに合わせたエージェントを設定可能。

つまり、エージェントサービスは、AIエージェント開発のプロセス全体を支援し、開発者が必要とするツールや機能を提供します。

Azure AI Foundryからのエージェント利用のイメージ



ラボ1: Azure AI Foundryでのエージェント作成

- ラボ1を起動
- <https://ai.azure.com/> にアクセス
- プロジェクトを作成
- 「エージェント プレイグラウンド」でエージェントをテストする
- Wordファイルをエージェントの「ナレッジ」に追加する

モジュール1 まとめ

- AIエージェント
- Azure AI Foundry
- Azure AI Foundry Agent Service
- Azure AI Foundry Agent ServiceによるAIエージェント開発（ラボ1）

モジュール2



Azure AI Foundry Agent Service を使用 して AI エージェントを開発する

C#などのプログラミング言語を使用して、AIエージェントを作成する方法を解説します。また、エージェントの中で「コードインタープリター」を使用する方法を解説します。

コードインタープリター (code interpreter) とは

- **コード インタープリター**により、エージェントはセキュリティで保護されたサンドボックス実行環境内で、生成したPythonコードを実行できる。
- これにより、エージェントは、データ分析、数学的計算、ユーザー要求に基づくファイル操作などのタスクを実行できる。
- たとえば、**数値データを分析し、グラフを作成する**、といった処理を実行できるようになる。



エージェントのコードインタープリターとは何ですか？ Pythonが使用されますか？ 簡単に説明してください



「コードインタープリター」とは、エージェントによってコードを実行するためのツールやサービスを指します。このツールは、データ処理や分析、アルゴリズムの実施を可能にし、特にPythonプログラミング言語を使用することがよくあります。コードインタープリターは通常、複雑な計算やデータの操作を自動化し、エージェントがインタラクティブなタスクを効率的に行えるようにします^{1 2}。

1  Azure AI Foundry2.html

2  Azure AI Foundry.html



コードインタプリターを使用して、数値データからグラフを作図できますか？

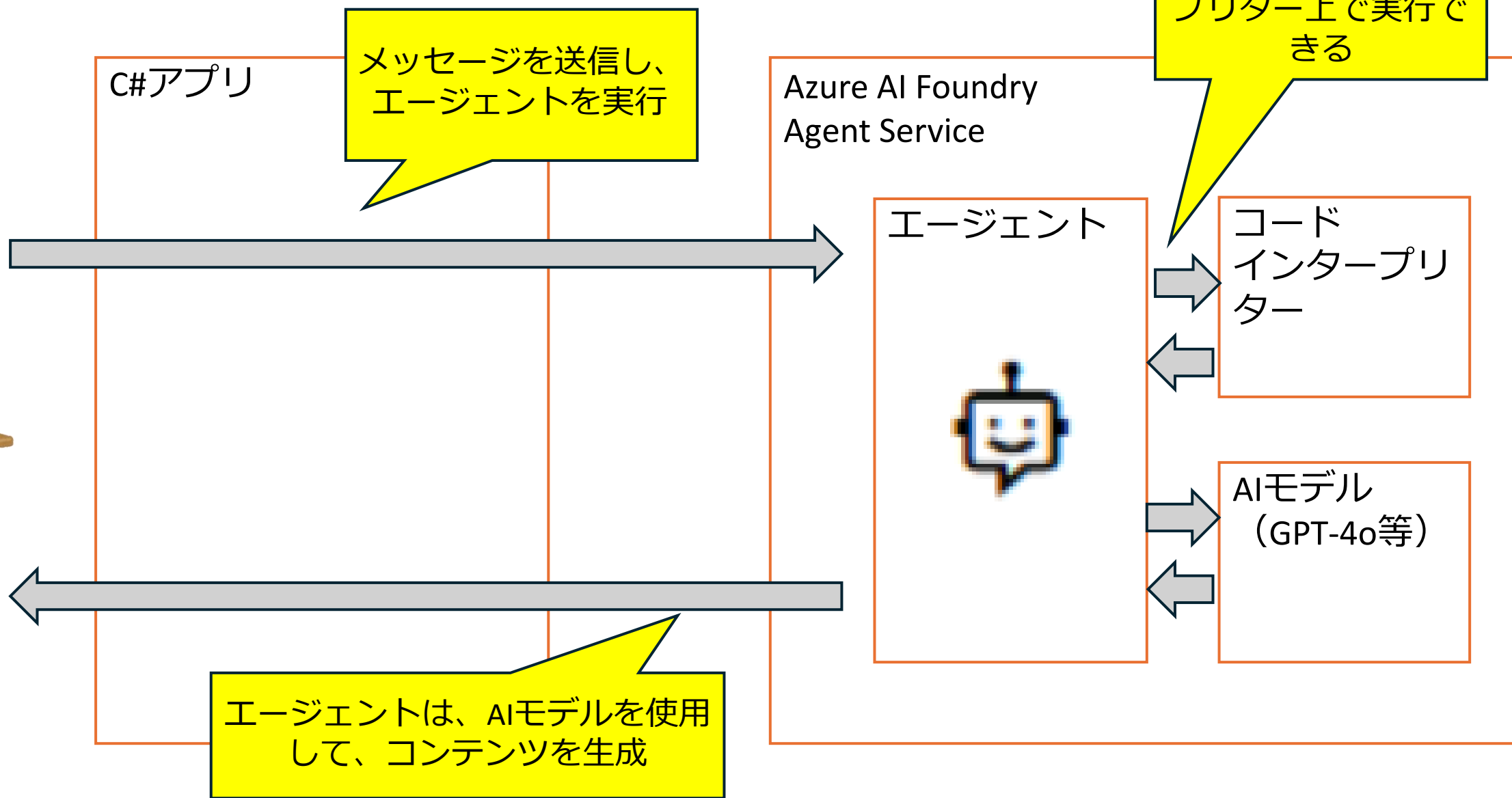


はい、コードインタプリターを使用すれば、数値データからグラフを作図することが可能です。主にPython言語を活用する場面では、例えば以下のようなライブラリが一般的に使用されます：

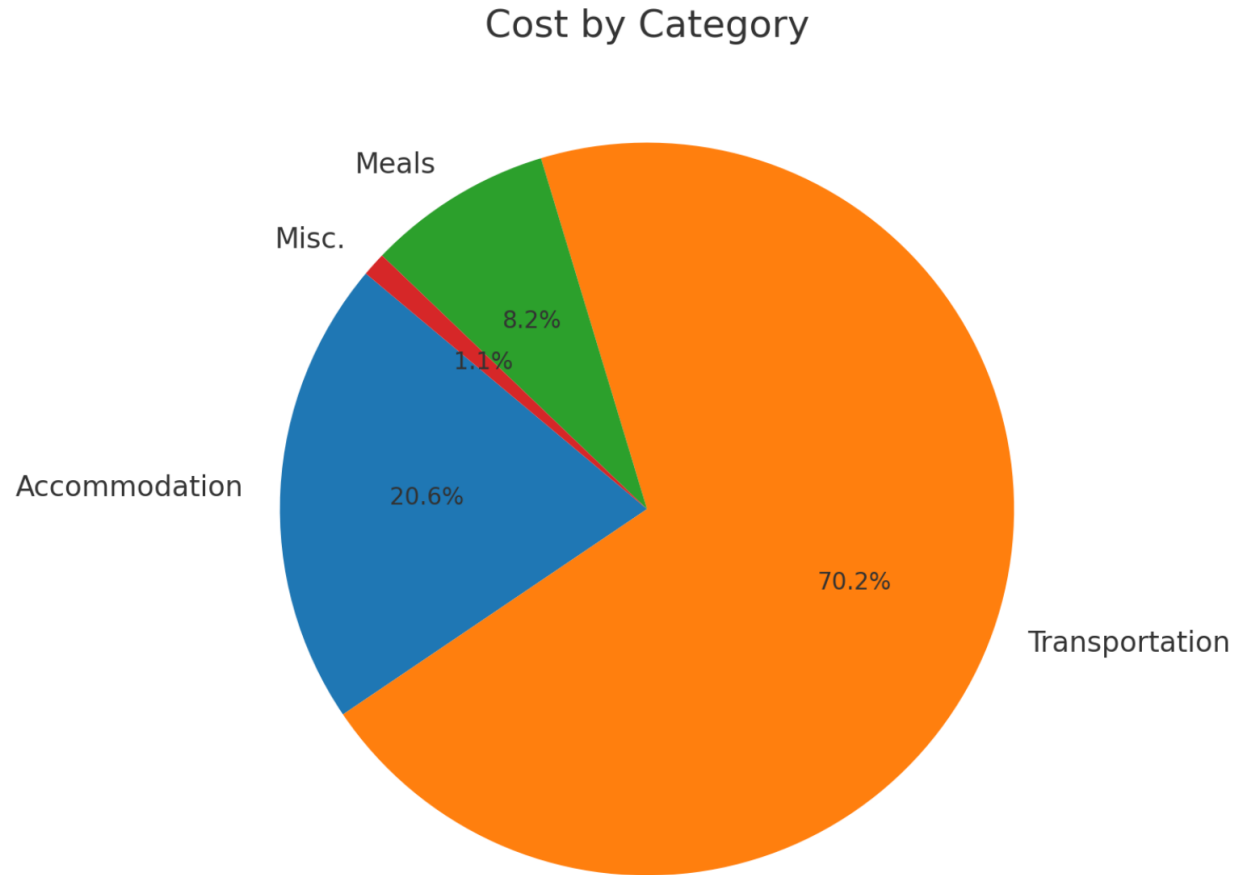
1. **Matplotlib**: 基本的なプロットやカスタマイズが可能。
2. **Seaborn**: データの統計的可視化に特化。
3. **Plotly**: インタラクティブなグラフ作成。

これにより、線グラフ、棒グラフ、散布図、ヒストグラムなどのさまざまな種類のグラフを作ることが出来ます。コードインタプリターと連携すれば、リアルタイムでデータを取り込み、可視化を行うことも簡単です。

アプリからのエージェント利用のイメージ



コードインタープリターを使用して作成された円グラフの例



プロジェクトエンドポイント

アプリからAzure AI Foundryプロジェクトに接続するには、
Azure AI Foundryプロジェクトの「プロジェクトエンドポイント」を利用する

The screenshot shows the Azure AI Foundry interface for a project named 'project53553528'. The left sidebar contains a navigation menu with the following items: 概要 (Overview), モデル カタログ (Model Catalog), プレイグラウンド (Playground), ビルドとカスタマイズ (Build and Customize), エージェント (Agents), テンプレート (Templates), 微調整 (Fine-tuning), コンテンツの 解釈 (Content Interpretation), 監視と最適化 (Monitoring and Optimization), トレース (Traces), 監視 (Monitoring), 保護と管理 (Protection and Management), 評価 (Evaluation), ガードレールとコントロール (Guardrails and Controls), リスク + アラート (Risk + Alerts). The '概要' (Overview) item is highlighted with a red box. The main content area shows the project name 'project53553528' and a description 'プロジェクトの説明を追加してください (省略可能)'. Below this is the 'エンドポイントとキー' (Endpoints and Keys) section, which has a link 'すべてのエンドポイントを表示する' (Show all endpoints). The 'API キー' (API Key) section shows a masked key with a copy icon. The 'ライブラリ' (Library) section lists 'Azure AI Foundry', 'Azure OpenAI', and 'Azure AI サービス'. The 'Azure AI Foundry' item is highlighted with a red box. To the right of the library list, there is a text block: 'デプロイされたすべての基本モデルを呼び出すには、次のエンドポイントを使用します:' (To call all deployed basic models, use the following endpoint:). Below this text, the 'Azure AI Foundry プロジェクト エンドポイント' (Azure AI Foundry Project Endpoint) is displayed in a box with a red border, showing the URL 'https://project53553528-resource.services.ai.azure.com/api/'. At the bottom, there is a button labeled '{x} API ドキュメント' (API Documentation).

認証

- Azure AI Foundryプロジェクトに接続するには、**Entra ID認証**が利用できる
- アプリを実行するコンピュータで、**az login コマンド**を実行して、サインインしておく
- アプリのコードで**Azure Identityライブラリ**を使用する

クライアントを作成する

エージェントを操作するための
「クライアント」を作成する

```
PersistentAgentsClient client = new(connectionString, new DefaultAzureCredential());
```

✓ 0.1s

プロジェクトエンド
ポイントを指定

Entra ID認証を使用し
て認証

エージェントを作成する

「クライアント」を使用して、
エージェントを作成する

```
PersistentAgent agent = await client.Administration.CreateAgentAsync(  
    model: deployName,  
    name: "data-agent",  
    instructions: "You are an AI agent that analyzes the data in the file that has been uploaded.  
    tools: [ new CodeInterpreterToolDefinition() ],  
    toolResources: new ToolResources() { CodeInterpreter = codeInterpreter }  
);
```

✓ 0.7s

「コードインタプリター」を使用
するための設定を行う

ラボ2: C#を使用してAIエージェントを作成する / コードインタープリターを使用する

- ラボ1を起動（どのラボでもよい）
- Azure AI Foundryで、プロジェクトを作成
- サンプルコードのリポジトリにアクセスし、GitHub Codespacesを起動
 - <https://github.com/hiryamada/AI-3026-notebook/tree/main>
- .env ファイルを作成、Azure AI Foundryプロジェクトのエンドポイントを設定
- az login を実行 (setup.dib)
- ラボ2のサンプルコードを実行 (lab02.dib)

モジュール2 まとめ

- コードインタプリター
- C#プログラムからのAzure AI Foundryプロジェクトへの接続
- 認証
- クライアントの作成
- エージェントの作成
- エージェントの実行
 - スレッドの作成、メッセージの追加、実行（Run）、実行結果の取得と表示、不要なスレッドとエージェントの削除

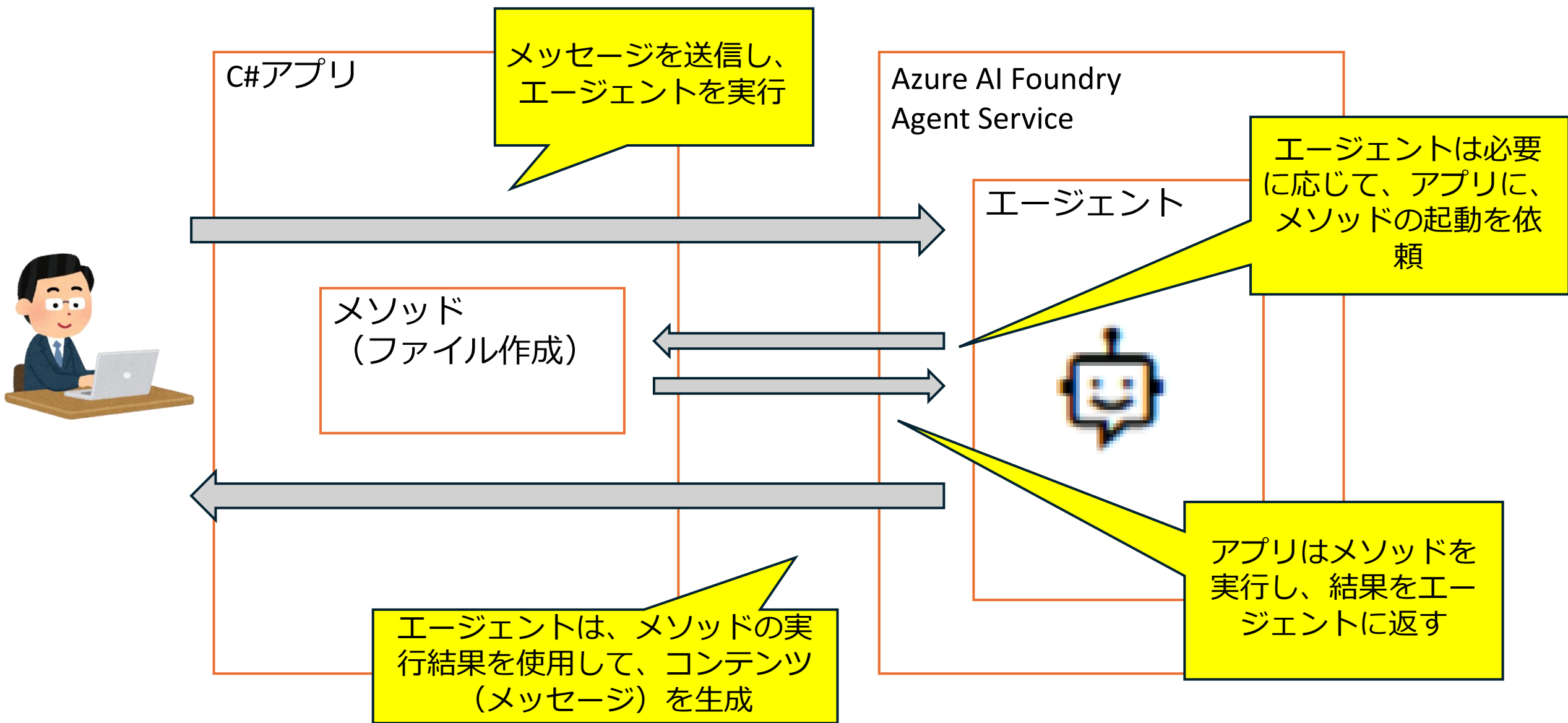
モジュール3



カスタム ツールをエージェントに統合する

アプリのコードをエージェントから利用できるようにする方法を解説します。

Function Calling（関数呼び出し）のイメージ



サポートチケットを作成するコードの例

```
async Task<string> SubmitSupportTicket(string emailAddress, string description)
{
    string ticketNumber = Guid.NewGuid().ToString("N").Substring(0, 6);
    var fileName = $"output/ticket-{ticketNumber}.txt";
    var text = $""
        Support ticket: {ticketNumber}
        Submitted by: {emailAddress}
        Description: {description}
        "";
    var messageJson = $$""
        {"message": f"Support ticket {{ticketNumber}} submitted. The ticket file is saved as {{fileName}}"}
        "";
    await File.WriteAllTextAsync(fileName, text);
    return messageJson;
}
```

作成されたチケットの例

```
output > ≡ ticket-550fa7.txt
```

```
1   Support ticket: 550fa7
2   Submitted by: alex@contoso.com
3   Description: my computer won't start.
```

ラボ3: C#を使用してAIエージェントを作成する / コードインタープリターを使用する

- ラボ1を起動（どのラボでもよい）
- Azure AI Foundryで、プロジェクトを作成
- サンプルコードのリポジトリにアクセスし、GitHub Codespacesを起動
 - <https://github.com/hiryamada/AI-3026-notebook/tree/main>
- .env ファイルを作成、Azure AI Foundryプロジェクトのエンドポイントを設定
- az login を実行 (setup.dib)
- ラボ3のサンプルコードを実行 (lab03.dib)