



【Microsoft Copilot Studio】 サービス概要

2026年05月28日

改訂履歴

版数	発行日	改訂内容
第1版	2026年5月28日	初版発行

本資料の内容は 2026/5/28 時点のものです。製品のアップデートにより変更となる場合がございます旨ご了承ください。

Agenda

1. 前提情報
 1. 用語集
2. Microsoft Copilot Studio とは
 1. 背景・基本概念
 2. AIエージェントとは
 3. Microsoft Copilot Studio の特徴
3. Microsoft Copilot Studio の仕組み
 1. Microsoft Copilot Studio を構成する要素の全体像
 2. 情報基盤
 3. 判断基盤
 4. 実行基盤
4. 業務適用に向けた設計判断
 1. 導入における設計判断とガバナンス
 2. ナレッジ設計の判断軸
 3. 判断責務の分離と制御設計
 4. 業務実行における AI 活用範囲の定義
5. ユースケース
 1. 問い合わせ・検索型
 2. 業務処理支援型
6. 制約条件とコストの考え方
 1. コスト・課金モデルの考え方
 2. コスト設計



1. 前提情報

1.1. 用語集

本書で使用する用語及び略称を以下の通り定義します。

No.	用語	説明
1	大規模言語モデル (LLM)	インターネット上の膨大なテキストデータを学習し、人間の言葉を理解して自然な文章を生成する AI (人工知能) の基盤技術。
2	生成AI	ユーザーの入力に応じて、文章・要約・回答などのコンテンツを自動生成する AI。人の作業を支援する用途で広く利用される。
3	ローコード/ノーコード	コード入力によるプログラミングをほとんど行わずに、操作画面や自然言語でアプリやシステムを開発できる手法。
4	GUI	アイコン・ボタン・メニューなどの視覚的な要素を用いて、コンピュータを操作するユーザーインターフェースのこと。
5	プロンプト	AI に対して与える指示や質問のこと。内容や書き方によって AI の出力結果が大きく変わる。
6	ナレッジ	AI が回答の根拠として参照するデータ (マニュアル・FAQ・社内文書など)。ナレッジの質が回答精度に大きく影響する。
7	Dataverse	Microsoft が提供するデータベースサービス。業務データを一元管理し、Copilot から安全に参照・更新できる。
8	Microsoft Entra ID	ユーザー・デバイス・アプリの ID を一元管理し、認証・認可を通じてアクセス制御を行うクラウドベースの ID 管理サービス。Microsoft 365 や Azure などの各種サービスへの安全なアクセス基盤として機能する。
9	Bing	Microsoft の検索エンジン。Copilot Studio では外部 Web 情報を取得するためのデータソースとして利用される。

1.1. 用語集

本書で使用する用語及び略称を以下の通り定義します。

No.	用語	説明
10	クエリ	システムやデータベース、検索エンジンに対して、情報の取得・操作を要求するための問い合わせ指示。SQL や検索入力などの形で、必要なデータや結果を取得するために使用される。
11	チャンク	データを検索・生成用途に最適化するため、内容や構造に応じて任意の単位で分割したテキストの塊。一般的な固定分割ではなく、文脈や意味を維持するよう設計され、検索精度や応答品質を向上させる。
12	Azure AI Search	Microsoft Azure が提供する、データを検索・解析するためのフルマネージド型のクラウド検索サービス。全文検索・ベクトル検索・ハイブリッド検索とAI機能を統合し、アプリやデータの検索基盤として利用される。
13	インデックス	データやドキュメントを高速に検索するために、キーとデータの対応関係を整理したデータ構造。データベースや検索エンジンにおいて、必要な情報を効率的に特定するための仕組みとして利用される。
14	SQL (Structured Query Language)	リレーショナルデータベース (表形式のデータ) を操作するための標準的な言語。データの検索・追加・更新・削除など、データベース内の情報を管理・操作するために使われる。
15	ガバナンス	AI の利用範囲やルールを定め、安全かつ適切に運用するための管理の仕組み。
16	検索精度	ユーザーの質問に対して、必要な情報をどれだけ正しく・適切に見つけられるかを示す指標。関連性の高い情報が適切に取得されるほど検索精度は高くなり、AI の回答品質にも直結する。
17	応答生成	検索した情報やユーザーの入力内容をもとに、AI が自然な文章として回答を作成する処理。取得した情報を整理・要約し、ユーザーにとって理解しやすい形に変換する役割を担う。

1.1. 用語集

本書で使用する用語及び略称を以下の通り定義します。

No.	用語	説明
18	メタデータ	データを管理・検索・活用しやすくするために付与するデータに関する情報。
19	トリガーフレーズ	ユーザーの発話の中で、特定のトピックや処理を開始させるためのキーワードや文。会話AIにおいてユーザーの意図を示す入力例として定義され、対応する処理や応答を起動する。
20	Power Automate	Microsoft が提供する業務自動化サービスで、アプリやサービス間の処理をフローとして定義し自動実行できるプラットフォーム。トリガーとアクションを組み合わせることで、通知・承認・データ連携などの業務を効率化する。
21	ガードレール	AI の動作や利用可能範囲に制限を設ける仕組み。不正利用や誤動作、コスト増大を防ぐ。
22	監査ログ	システムやアプリケーションにおいて発生した操作やイベントを時系列で記録し、「誰が・いつ・何をしたか」を後から検証できるようにした証跡情報。
23	プレミアムコネクタ	Copilot Studio / Power Apps / Power Automate など利用できるコネクタのうち、追加（有償）ライセンスが必要な外部サービス連携用コネクタ。 以下、プレミアムコネクタ一覧記載 Microsoft 公式ページ。 https://learn.microsoft.com/ja-jp/connectors/connector-reference/connector-reference-premium-connectors



2. Microsoft Copilot Studio とは

2.1. 背景・基本概念

近年、生成AIの進化は飛躍的に進み、ビジネスシーンの利活用も増えてきています。実際、多くの企業がAIを導入しているものの、本格的に業務へ組み込み、成果を出せている企業はまだ一部にとどまっているのが現状です。

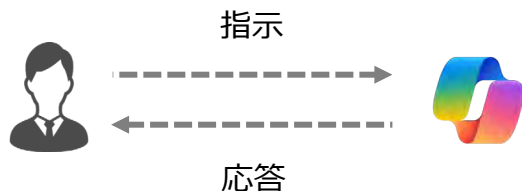
このような流れの中で重要なのが、「AIを使う」ことではなく「AIを業務に組み込む」ことです。生成AIは、ユーザーの質問や指示に対して返答する「個人支援ツール」としての位置づけで、業務との連携や継続的な活用には限界がありました。

一方で、Microsoft Copilot Studio（以下、Copilot Studio）は、自社の業務プロセスやデータと連携した「企業専用のAIエージェント」を構築できるプラットフォームです。これにより、業務時間の大幅な削減や処理スピードの向上といった成果が報告されており、単なるコスト削減ではなく業務そのもののあり方を変革するインパクトを持っています。

As-Is

生成AIの個人活用

議事録や画像生成など
個人業務レベルを効率化



To-Be

AIエージェントとの協業

AIエージェントが複数のシステムと
横断的に連携し、業務全体を革新



2.2. AIエージェントとは

一般的にビジネスシーンで使われる生成AIと、Copilot Studio で作成できる AI エージェントとの違いについて解説します。

生成AIは前述のとおり、「個人の作業を支援するパーソナルアシスタント」であることに対し、AI エージェントは「企業が運用するビジネスプロセスに導入される自立型ソフトウェア」という位置づけになります。

	生成AI	AIエージェント
主な役割	個人の作業を支援	企業の業務の代替
活用場面	議事録・資料作成・ 画像生成・データ分析	社内 FAQ・CRM 登録・ データベース検索・社外ヘルプデスク
自立性	ユーザーの指示のみ	目的によって自己計画、 自己修正のループ
複雑なタスク処理	単一応答のみ	外部ツールと連携し 計画・実行・観察・修正

生成 AI
「人 × AI」の1対1で個人の作業効率を上げる



AI エージェント
「複数人 × AI」で企業全体の業務効率化を広げる

2.3. Microsoft Copilot Studio の特徴

Copilot Studio は、ローコード/ノーコードで業務に特化したAIエージェントを構築出来るプラットフォームです。主に以下のような特徴があります。

Copilot Studio の特徴

AIエージェントの構築

- GUI ベースでドラッグ&ドロップなどによる簡単な構築
- 自社が保有する業務データ（マニュアル、FAQ、チャットなど）を取り込み、自社専用AIエージェントを作成

高機能なカスタマイズ

- Microsoft 製品以外の CRM、ERP など多数の外部アプリと連携
- トピック機能やエージェントフロー機能により、ユーザーとの対話の中で業務処理を判断&実行

ローコード/ノーコードで高機能な**自社専用AIエージェント**が構築できる

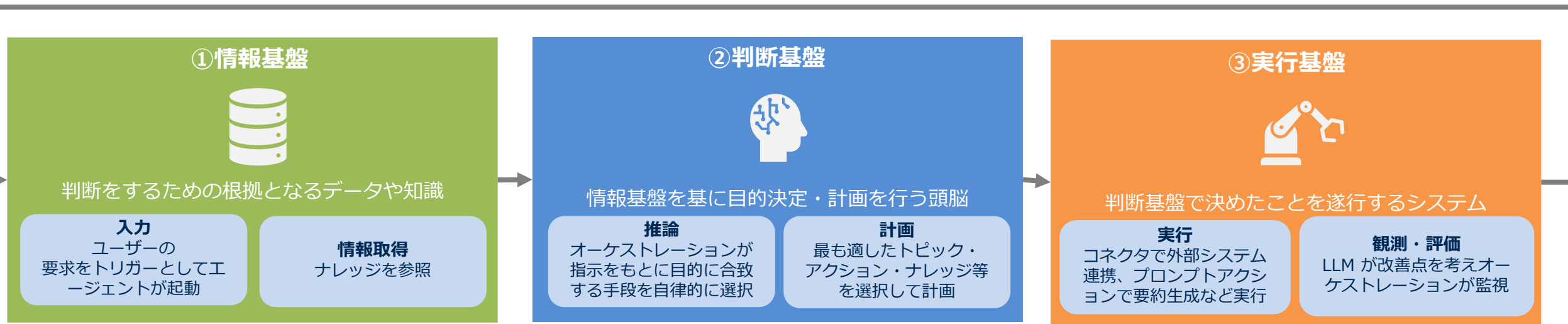


3. Microsoft Copilot Studio の仕組み

3.1. Copilot Studio を構成する要素の全体像

AIエージェントとは、大規模言語モデル (LLM) を中核に据え、情報参照 → 判断 → 実行を分離して処理することで、意思決定・実行を自律的に行うシステムです。複合的な仕組みにより、ループ的に試行錯誤しながら解答や処理を進めます。

プロンプトに対して受動的に回答する生成AIチャットでは、処理の制御・再現性が難しいという課題がありますが、AIエージェントは各基盤が分業されているためAIの挙動を可視化・制御でき、業務として成立させられる点が特徴です。



主要コンポーネント

トリガー：エージェントが動き出すきっかけとなる条件

ナレッジ：SharePoint、Dataverse、Web サイトなど連携するデータソース

オーケストレーション：

状態管理を統括し、安定した実行を担う仕組み

指示：AIエージェントのプロンプト

トピック：ユーザーの会話に応じた処理や流れの定義

ノード：トピック内の次のステップを定義するパーツ

コネクタ：外部ツールやシステムを連携するツール
プロンプトアクション：プロンプトで内容要約や返信文作成などを実行

3.2. 情報基盤

情報基盤は、「AI が判断をするための根拠となるデータや知識」の部分です。情報基盤の中でも、ナレッジはAIが参照する情報ソースです。信頼性のあるナレッジを、適切な粒度や鮮度で提供することがAIエージェントの精度を高めます。Copilot Studio でのナレッジについて、以下4つの要素で解説します。

ナレッジの範囲

Copilot Studio がサポートするナレッジは以下の通りです。内部データは Microsoft Entra ID によるアクセス制御でフィルタリングされ、さらに回答時にソースを手動で指定するクラシックモードとAIが自動で参照する生成モードを設定可能です。

- ・公開Webサイト（外部）：Bing でクエリ入力を検索
- ・Documents（内部）：Dataverse 内にアップロードされたドキュメント
- ・SharePoint（内部）：SharePoint 内を横断的に検索
- ・Dataverse（内部）：Dataverse の検索拡張生成技術を使用
- ・連携コネクタ先データ（内部）：外部連携できるエンタープライズシステム

ナレッジの信頼性

Copilot Studio におけるナレッジの信頼性は、以下設定の組み合わせで決定されます。これらの技術的制約と運用リスクを理解し、業務要件に応じた信頼性設計を行う必要があります。

- ・Grounding 設定：「選択ナレッジのみ参照」+「一般知識オフ」により、AIの推測や創作を防げる
- ・Web Search 設定：未承認の公開Web混入を防ぎ、公式性・鮮度を担保
- ・権限フィルタリング：Microsoft Entra ID 権限に基づき内部ソースを制限
- ・コンテンツモデレーション：暴力、性的表現などの不適切なコンテンツを検出・ブロック

ナレッジの粒度

ドキュメントなどの非構造化データでは自動的にチャンク化され、開発者は標準機能では制御できません。Dataverse ではレコード/列単位が粒度となり、同義語/用語集で意味的な粒度を補正できます。粒度が不適切だと、文脈の欠落、誤解釈、断片的回答が発生し、AI の信頼性が低下します。運用上は以下のような戦略を推奨します。

- ・ハイブリッドアプローチ：粗い粒度と細かい粒度のナレッジソースを併用
- ・テスト：想定される質問パターンに対して、AI の応答を事前にテストし、粒度が適切かを検証
- ・Azure AI Search 活用：Azure AI Search でカスタムチャンク戦略を設計

ナレッジの更新

Copilot Studio におけるナレッジの更新頻度については、以下の通りナレッジの種別により異なります。現在ナレッジのバージョンや同期日時を確認する機能が標準搭載されていないため、変更管理ワークフローを導入や、ナレッジに有効期限情報を持たせるなどの運用でカバーする必要があります。

- ・非構造化データ（SharePoint/OneDrive）：同期頻度は非公開かつ手動同期不可
- ・Web Search：外部インデックスに依存するためリアルタイム性を確保不可
- ・Dataverse：レコード変更はほぼリアルタイムで反映されるが同義語/用語集は遅延あり

3.3.判断基盤

判断基盤は、「AIが情報基盤をもとに、目的を決定して計画を行う頭脳」の部分です。具体的には、指示によってAIの行動ルールを決め、オーケストレーションで処理の流れや手順を設計します。さらに、トピックやノードを使って会話や処理の具体的なステップを定義することで、状況に応じた最適な判断と実行計画を自立的に組み立てます。この判断基盤となる各要素が曖昧だと、AIの挙動が予測出来なくなり、結果として想定通りの成果が得られなくなる可能性があります。

指示

AIエージェント作成時に与えるプロンプトです。ここを曖昧にしまうとAIが意図しない動作をしてしまうため、具体的に詳細を書くことが重要です。役割・目的・条件と行動指針を段階的に記載する構成で記述します。

オーケストレーション

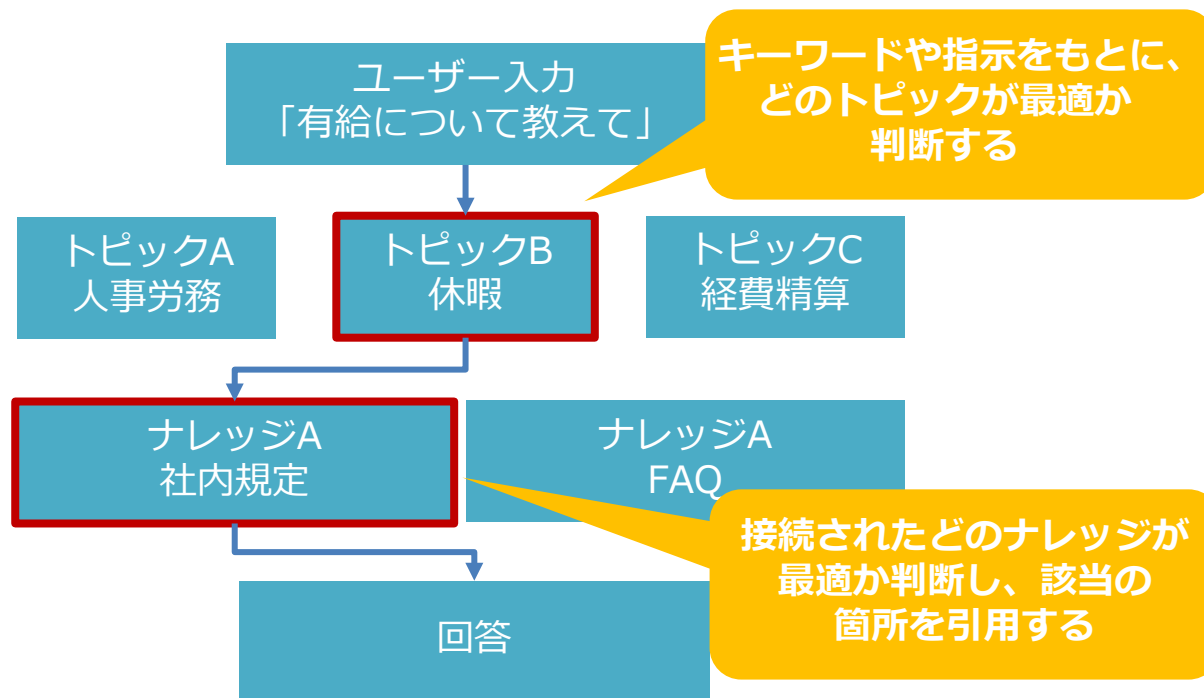
AIがどのような処理をどの順番で行うかを制御する仕組みです。LLMが生成した意図や判断に基づいて、事前に指示などで定義されたルールを踏襲して、指定された設定やツールの中でAPI呼び出しやデータ処理、外部ツール連携などを自動的にトリガーします。

トピック

トピックはユーザーからの会話に応じた処理や流れをエージェントが定義するための機能です。ユーザーの会話の中で、予め定義したキーワードをトリガーとして、ナレッジを呼び出す、情報を聞き出す、外部ツールを呼び出すなどの次の処理ステップを行います。

動作フローイメージ ■ : AIが判断する箇所

指示：ナレッジを利用して質問に回答する



3.3.判断基盤（補足）

トピック機能についての補足説明を記載します。トピックとは、ユーザーの会話に応じた処理や流れをエージェントが定義するための機能です。トピックを設定せずともAIエージェントはユーザーとの会話に LLMで場面に合った返答をしますが、開発側がユーザーの会話に合わせて特定の処理を行いたい場合はトピックを設定します。以下にその例を記載します。

シナリオ：小売店ECサイトのヘルプチャットボットで特定の商品情報を案内したい

トピック設定例

- ①トリガー設定：「おすすめは？」「新商品について教えて」など想定される会話を入力する
- ②ノード設定：案内したい会話を入力して回答させる

トピックを設定した場合



新商品について教えて

トピックのトリガーに該当するため、設定した会話ノードを行う

おすすめの新品は〇〇です
URLはこちら
→<https://xxxxx>

売りたい商品情報を案内することができる



トピックを設定していない場合



新商品について教えて

トピックが設定されていないため、一般的な Copilot の回答のみ

どのような商品をお探しですか？

Copilot が LLM で判断するため開発側が意図しない商品が案内されることがある



3.3.判断基盤（補足）

設定画面イメージ

The screenshot shows a configuration interface for a chatbot topic. It consists of three main sections:

- トピック (Topic):** A header bar at the top.
- トリガー (Trigger):** A section containing a trigger icon, the text "エージェントが選択するもの" (Selected by agent), and a description box. The description box contains the text: "このトピックでは、「新商品のおすすめは?」「どんな新商品がありますか?」「最近入荷したものは?」「新しい商品について教えて」などの質問に回答します。" (In this topic, we answer questions such as "What are the recommended new products?", "What new products are there?", "What new products have recently arrived?", and "Tell me about new products.")
- メッセージ (Message):** A section containing a text message node. The message text is: "おすすめの新商品は〇〇です。URLはこちら→https://xxxx~" (Recommended new products are 〇〇. URL is here → https://xxxx~).
- 現在のトピックを終了する (End current topic):** A button at the bottom to end the current topic.

Labels on the left side of the screen identify these sections: "トピック", "トリガー", "メッセージ", and "現在のトピックを終了する".

チャット画面イメージ

The screenshot shows a chat conversation:

- User Message:** "こんにちは、testと申します。ご用件をお申し付けください。" (Hello, my name is test. Please let me know what you need.)
- Bot Response:** "おすすめの新商品は〇〇です。URLはこちら→https://xxxx~" (Recommended new products are 〇〇. URL is here → https://xxxx~)

Annotations in the image show a dashed orange box around the user's question "新商品について教えて" (Tell me about new products) and a dashed blue box around the bot's response "おすすめの新商品は〇〇です。URLはこちら→https://xxxx~". Arrows indicate the flow from the user's question to the bot's response.

3.4. 実行基盤

実行基盤は、「判断基盤で決めたことを現実の業務として安全に遂行するシステム」部分となります。Copilot Studio では、複数の実行処理が選択できます。

主な実行ツール

コネクタ：外部のシステムやサービスと連携するための仕組みです。API の呼び出しを簡単に扱えるようにラップした機能であり、データの取得・更新・通知などの処理を実行できます。

プロンプトアクション：自然言語で記述した指示をもとに、生成AIが処理や回答を生成します。内容要約や返信文作成などを実行できます。

エージェントフロー：エージェント内で実行する自動化ワークフローです。複数の処理を組み合わせることができます。

用途	利用実行ツール	具体処理	使用コネクタ
在庫確認	コネクタ	在庫データベースから数量取得して回答	SQL / Dataverse
レポート生成	コネクタ+プロンプトアクション	データベースからデータ取得 → 要約して回答	SharePoint /Excel
申請登録	コネクタ + エージェントフロー	申請作成 → 通知	Dataverse / Outlook / Teams



4. 業務適用に向けた設計判断

4.1.導入における設計判断とガバナンス

Copilot Studio における基本設計では、「どこを固定化し、どこをAIに任せるか」を意識しながら各設計を分離することが重要です。その際、制御可能な最小範囲からスモールスタートで段階的に設計を進めることが重要です。以下に基本設計の例を記載します。

	概要	範囲	確認内容
制御設計	エージェントの役割・目的を定義	指示の設計・オーケストレーションモードの選定	—
ナレッジ設計	AIが何を根拠にするかを定義	ナレッジの準備、精査	FAQ検索で正しく回答できるかなどを確認
会話設計	ユーザーの会話による処理の流れを定義して固定化	トピック構造の設計・ノード設計	必須入力項目に漏れがないか、会話分岐、ユーザー誘導の確認
実行設計	AIの判断を実際のシステム処理にどう繋げるかを定義	各トピックから想定される必要な実行処理を設計	権限、エラー処理含め正しく実行されたか確認
ガバナンス設計	安全で信頼できるAI基盤を提供できるかを評価する	データ・権限・監査の面からセキュリティ対策	各設定で想定通り作動するか確認

ポイント スモールスタートで設計し、都度テストをしてから次のステップへ

4.2. ナレッジ設計の判断軸

Copilot Studio におけるナレッジ設計とは、LLM が、意図解釈に基づき適切な情報ソースを参照し、正確かつ一貫性のある応答を生成できるように、知識データを構造化・最適化する設計を指します。単にドキュメントを格納するのではなく、検索精度と応答生成の品質を高めるために、「対象情報の選定」「情報の粒度設計」などを体系的に設計することが重要です。

ナレッジ設計の重要要素

Chunking (チャンク分割)

概要：文書を検索単位として適切なサイズに分割する

目的：意味単位・章単位・業務単位などで分割することで、検索精度を向上

例：「目的」「手順」「問い合わせ先」などの見出しごとに区切り、長い見出しはさらに一定の文字数（例：800～1,200文字程度）で分割など

Freshness (鮮度)

概要：ナレッジの最新性を維持する

目的：古い情報参照による誤回答リスクを防止

具体例：ドキュメントのメタデータで更新日や有効期限を管理し、定期的にナレッジを更新・置換する

Scoping (スコープ)

概要：参照する文書や情報源をある程度制御する

目的：ユースケースに合わせた文書を用意することでノイズを除去し、検索精度を向上

具体例：全ての社内文書を一気に詰め込むのではなく、問い合わせの多いものから順に参照させる



ポイント

**ナレッジ設計は初期構築で終わらず、運用で継続的に改善することが重要
実運用前には必ずテストを実施し、想定質問に対する回答品質を検証する**

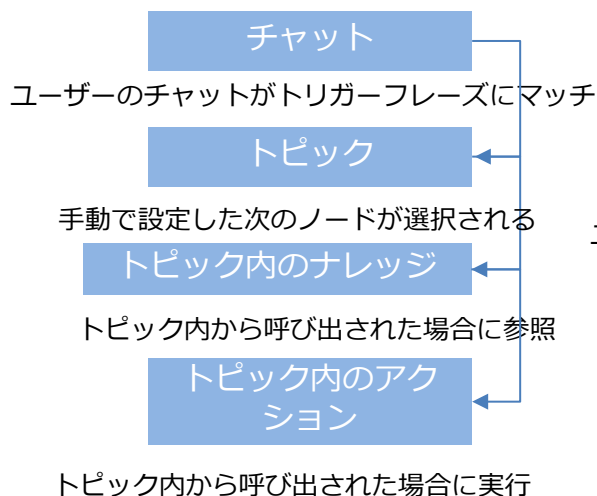
4.3.判断責務の分離と制御設計

Copilot Studio のように生成AIを組み込んだシステムでは、どの部分をAIに任せ、どの部分を明示的なルールで制御するかが設計の肝となります。これを初めから明確に設計レベルで整理しておくことで、後々システムが複雑化するのを防ぎ、保守性が高い構成を実現できます。

オーケストレーションパターン

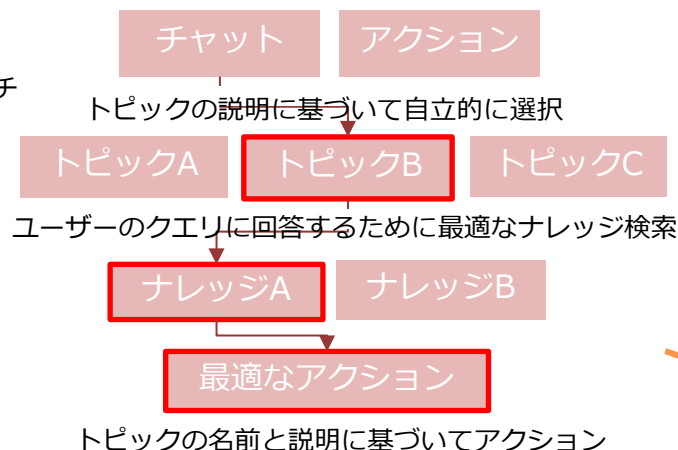
AIエージェントには2種類の動作パターンがあります。クラシックオーケストレーションは、ユーザー入力がトピックに登録した「トリガーフレーズ」の一致によって次のノードを開始します。確実な制御が可能ですが、ユーザーのあらゆる言い回しを予測して登録する手間がかかります。一方、生成AIオーケストレーションはAIが文脈を理解し、最適なトピックやナレッジを自律的に選択します。詳細なトリガー設定が不要で、AIがナレッジや指示に基づいて柔軟に対応しますが、結果の再現性が低いなどのリスクもあります。デフォルトでは、エージェント新規作成時に生成オーケストレーションが有効になっています。

クラシックオーケストレーション



生成オーケストレーション

※アクショントリガーからも実行可能



生成オーケストレーションのリスク

- ✓ 間違った申請処理などの意図しないツール選択
- ✓ 生成AI利用による消費クレジットが増大
- ✓ 会話コンテキスト崩壊
- ✓ 結果の再現性が低い

ポイント AIエージェントの「生成AIによる判断」と「ルールベースによる判断」の使い分けが重要

※次ページ参照

4.3.判断責務の分離と制御設計

オーケストレーションの使い分け例

クラシックオーケストレーションが向いているケース

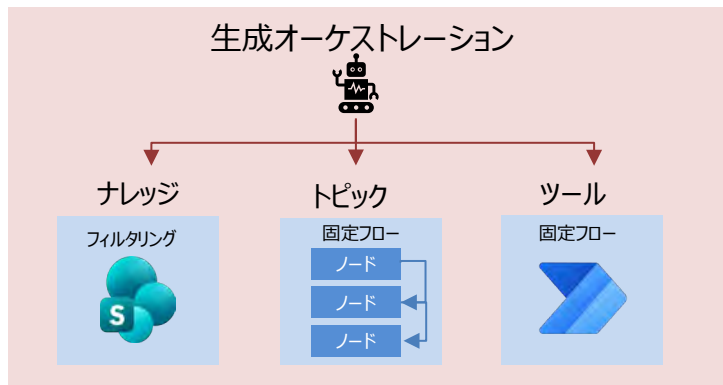
ユースケース	内容	理由
登録処理	ユーザー情報登録	漏れなく固定した情報を取得するフローが可能
定型FAQ	営業時間やパスワードリセット案内	AI不要でコストパフォーマンス向上
分岐型サービス案内	自然言語ではなく分岐型によるサービス案内	設計通りに誘導可能

生成オーケストレーションが向いているケース

ユースケース	内容	理由
FAQ、ナレッジ検索	ユーザーからの自然言語による問い合わせ	AIが意図を解釈して柔軟に処理できる
ITヘルプデスク	FAQ回答+チケット発行	AIがトピック、ナレッジ、ツール、を横断的に実行
複数要求	経費ルールが知りたい+申請など複数の要求	複数の意図を同時に処理し、動的に実行計画を生成

ハイブリッド構成の重要設計

オーケストレーションのハイブリッド構成は、生成AIの柔軟な意図理解力と、固定フローによる安全な業務設計を組み合わせる考え方です。どのトピック、ツール、ナレッジを使うかはAIが選択しますが、以降の制御はトピックやツール連携先の固定フローが優先され、各コンポーネント内で定義された処理が実行されます。そのため、ハイブリッド構成では以下の設計が重要になります。



プロンプト（指示）設計：

エージェントの「振る舞い」を最初に定義します。回答方針、制約条件、優先動作、禁止事項などを指示します。この設計が曖昧になると、AIが挙動に迷い意図しない動作を行うリスクがあります。

トピック設計：

トピック単位で業務処理や会話フローを固定化します。承認処理・条件判定・システム登録など厳密な制御が必要な処理はトピック内で収束する設計が必要です。トピックは業務単位で小さく分割することでAIが適切なトピックを選択しやすくなります。

ポイント ハイブリッド構成でAIが安全に業務を実行できる構造を設計する

4.4.業務実行における AI 活用範囲の定義

Copilot Studio の実務設計では、機能選定から考えるのではなく、「業務をAIにどこまで任せたいか」という業務レベルから整理することが重要です。AIの活用範囲によって、必要な制御や構成が変わってくるためです。ここでは、1つのシナリオをベースに実務設計を行う例を説明します。

シナリオ

経費精算をAIエージェントで自動化する場合

経費についてのFAQ回答のみなのか、承認フローまで含めるのか、によって設計とAI実行責務が大きく変わります。最初は最小業務レベルからPoCを行い、徐々に範囲を拡大していくことが推奨されます。



ポイント 最初に「どの機能を使うか」ではなく、「業務をどこまで自動化したいか」を考える

4.4.業務実行における AI 活用範囲の定義

シナリオ

経費精算をAIエージェントで自動化する場合

AI活用範囲が広がるほど、設計や運用範囲が拡大していきます。特に、承認・条件判定・システム更新など厳密性や再現性が求められる業務については、固定フローで制御してハイブリッド構成にしてAIの柔軟性とシステムの安全性を両立する設計が求められます。

	AI操作	構成要素	必要設計
レベル1 FAQ	<ul style="list-style-type: none">ナレッジ検索回答生成文書要約	ナレッジ 生成AI	プロンプト設計 ナレッジ設計
レベル2 入力支援	<ul style="list-style-type: none">申請内容整理下書き文生成	生成AI トピック	～ トピック設計
レベル3 実行支援	<ul style="list-style-type: none">Outlook 送信Teams 通知承認フロー起動	トピック ツール Power Automate	～ ツール設計 Power Automate 設計 連携コネクタ設計 権限・セキュリティ設計
レベル4 業務実行	<ul style="list-style-type: none">経費システム登録Dataverse 更新チケット起票など	トピック コネクタ Power Automate	～ ガードレール設計 監査ログ
レベル5 自立実行	<ul style="list-style-type: none">条件判断複数ツール呼び出し業務全体を自立実行	マルチエージェント API連携	～ AI責務定義 停止条件 リスク制御 監視 運用ガバナンス



設計・運用・クレジット消費などの運用コストを踏まえて費用対効果を考慮した適用範囲を選択する



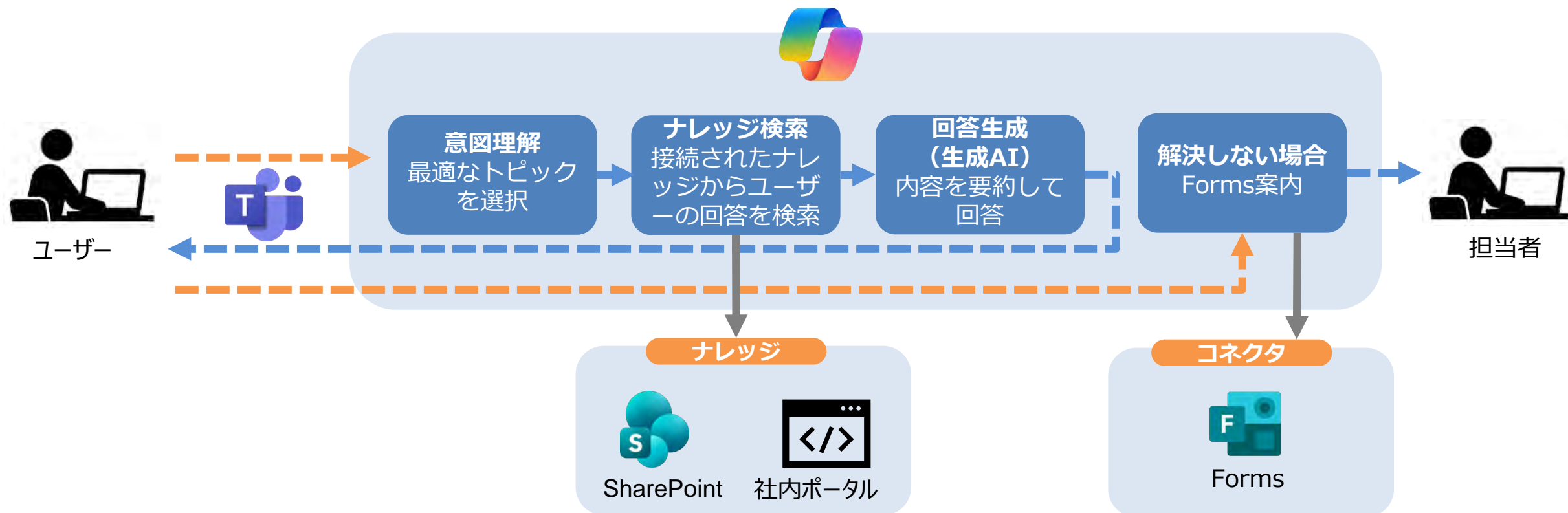
5. ユースケース

5.1.問い合わせ・検索型

概要 社内ナレッジ検索エージェント

構成 ナレッジ(SharePoint) / トピック/プロンプトアクション / コネクタ (Forms)

用途 ユーザーの質問を理解して、社内ナレッジを検索。回答として生成AIで要約し、手順やリンクを案内する。

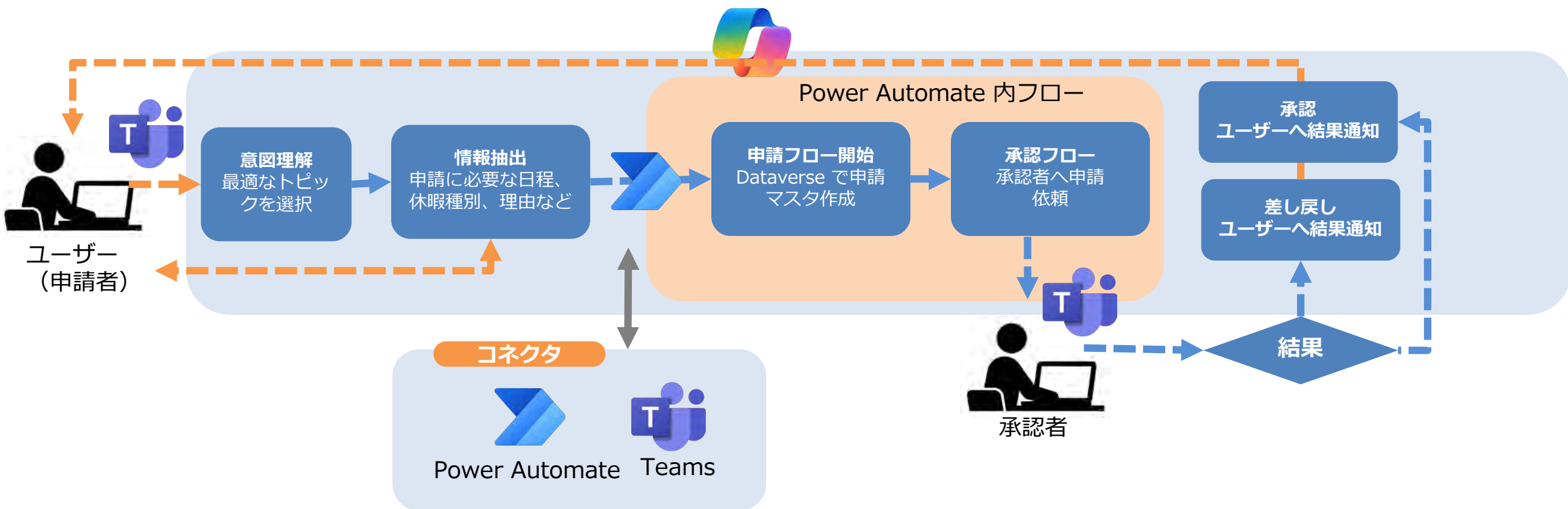


5.2.業務処理支援型

概要 人事労務エージェント

構成 トピック/コネクタ(Power Automate/Teams) / エージェントフロー

用途 ユーザーの有給休暇申請に対して自動で情報を聞き出し、Power Automate 連携で申請フローの実行を行う。





6. 制約条件とコストの考え方

6.1.制約条件とコストの考え方

Copilot Studio のライセンス体系において重要なのが「Copilot クレジット」という概念です。

これは、Copilot Studio における全機能共通の「通貨」のようなもので、AIエージェントの処理の複雑さや負荷に応じてクレジットが消費されます。

※Copilot クレジット消費が発生しないケースもあります。詳細は以下 Microsoft 公式ページをご確認ください。

<https://learn.microsoft.com/ja-jp/microsoft-copilot-studio/requirements-messages-management>

AIエージェントが処理を行うための通貨 = Copilot クレジット

従来型の回答

トピックにより開発者が手動で設定した回答

1

クレジット
/回

生成回答

AIがLLMによって生成した回答

2

クレジット
/回

エージェント アクション

AIが判断して外部APIを呼び出すなどのアクション

5

クレジット
/回

テナントグラフ検索

AIがSharePointに格納されたファイルなどを検索

10

クレジット
/回

AIツール (Premium)

高度な推論を含むレポート生成や長文要約など

100

クレジット
/10回

AIが行う処理の複雑性や負荷により Copilot クレジット消費レートが増大

※上記レートは一例です。詳細は以下 Microsoft 公式ページをご確認ください。

<https://learn.microsoft.com/ja-jp/microsoft-copilot-studio/requirements-messages-management>

6.2.コスト設計

Copilot Studio のコスト設計について解説します。Copilot Studio のコストは、固定費ではなく利用量（クレジット）を前提とした“変動型コスト”として設計することが重要です。

推奨見積もり

平均（通常利用）伸長（利用増加）上振れ（繁忙期・全社展開）の3パターンを見積り
→上振れパターンで予算を確保しておき、上振れ時に制限する機能などを予め決めておく。

見積もり例：ヘルプデスクAIエージェント平均

- ・対象ユーザー数（例：500名）
- ・月間問い合わせ件数（例：1人あたり0.6件 → 300件/月）
- ・1件あたり平均会話ターン（例：6往復）
- ・生成AIを使う割合（例：30%）
- ・外部連携（チケット起票・アカウントロック解除など）の割合（例：20%）

予算設計

ガードレール設定

- ・**機能ガードレール**：生成AIは特定トピックのみ、外部コネクタ連携は承認制にするなど機能制限
- ・**データガードレール**：SharePoint検索は特定サイトのみ、ナレッジ接続時は承認制などデータ制限

エラー見直し

- ・例外処理時の定形フローを用意しておく
- ・フローのループ発生を防止



すべてAI化はNG。高コストの利用（横断検索・要約分析・プレミアムコネクタ）を見直し、低コストの利用（固定化されたフロー・条件分岐を用意・データのスコーピング）へ置き換える