

生成AIに手足を与える

Qumcumロボット連携から学ぶFunctionCallingとMCP

2025/11/22

サイオステクノロジー株式会社
Professional Service Service Line
清水 比路



清水 比路 (Hiro Shimizu)

所属：サイオステクノロジー

- 新卒1年目 エンジニア
- 業務内容
 - RAGアプリケーション開発

本セミナーの目的



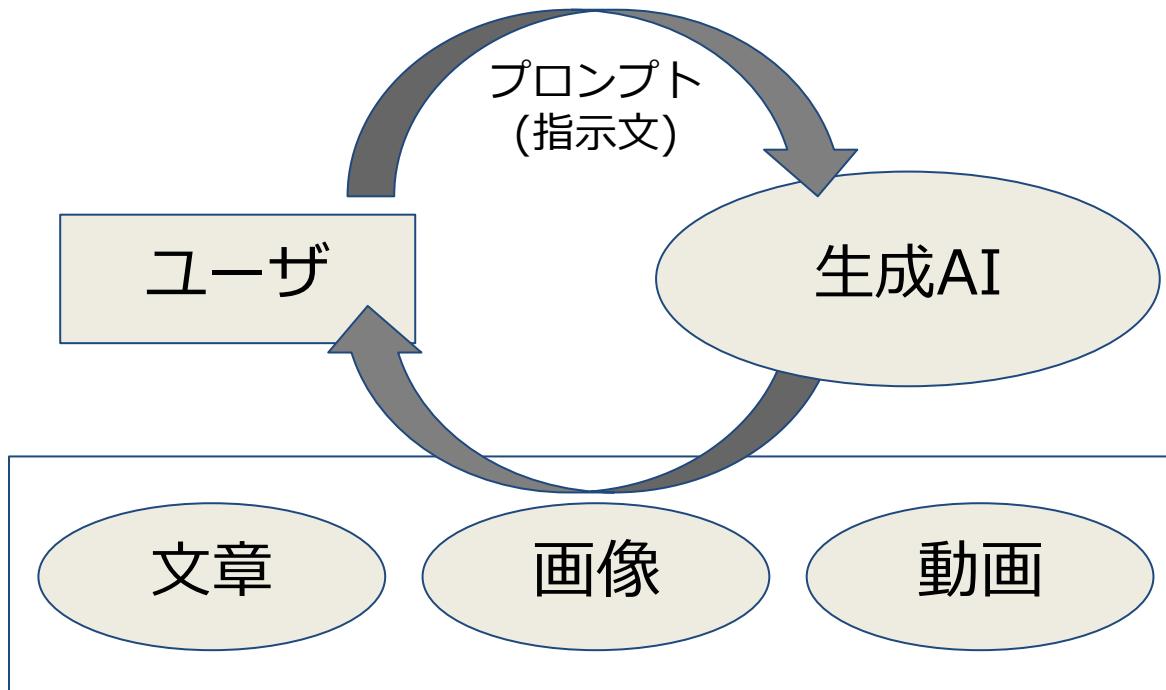
- ・生成AIの概要を理解できる
- ・生成AI×ツールで新たな価値を生み出せることを理解できる

-
- 1.生成AIって何？
 - 2.生成AI×ツールの実例
 - 3.生成AIにロボットを操作させる
 - 4.まとめ

-
- 1.生成AIって何？
 - 2.生成AI×ツールの実例
 - 3.生成AIにロボットを操作させる
 - 4.まとめ

生成AIって何？

コンテンツを新たに生み出してくれるAI(人工知能)



生成AIの例



テキスト生成

- Gemini 2.5Pro
- GPT-5
- Claude Sonnet 4.5

画像生成

- Stable Diffusion
- dall-e3
- imagen2

動画生成

- Sora
- Veo

テキストの生成



JSON形式からcsvに変換する関数の関数名のアイデアをください

◆ 思考プロセスを表示 ▾

JSONからCSVに変換する関数の関数名ですね。分かりやすく、一般的に使われるパターンをいくつか提案します。

1. シンプルで直接的な名前

最も一般的で、ひと目で内容がわかる名前です。

- jsonToCsv
- convertJsonToCsv
- transformJsonToCsv

2. 目的や動作を強調する名前

関数の具体的な役割（エクスポート、書き込み、解析など）を名前に含めるパターンです。

- exportJsonToCsv (エクスポート機能の場合)

Geminiへのプロンプトを入力

+

ツール

2.5 Pro ▾

指定した形式でテキストの生成



A screenshot of a web-based text generation interface. At the top, a dark gray box contains the instruction: "JSON形式で回答してください。fieldは、title,content,currenttime をお願いします。..." (Please respond in JSON format. The field is title, content, currenttime). Below this, a "思考プロセスを表示" (Show thought process) button with a dropdown arrow is visible. The main content area is a JSON editor with the title "JSON". It displays the following JSON code:

```
{  
  "title": "桃太郎、島へ行かず ~防衛と共生の道~",  
  "content": "桃太郎は、すくすくと育ち、村一番の力持ちになりました。\\n海のかなたにある鬼ヶ島から  
  "currenttime": "2025-11-10T17:14:49+09:00"  
}
```

At the bottom of the JSON editor, there are several small icons: a left arrow, a right arrow, a circular arrow, a double arrow, a square, and a colon. Below the JSON editor, a large input field is labeled "Geminiへのプロンプトを入力" (Input prompt for Gemini) and contains the text "桃太郎、島へ行かず ~防衛と共生の道~". At the bottom right of this input field, there are buttons for "ツール" (Tools) and "2.5 Pro" with a dropdown arrow. A microphone icon is also present.

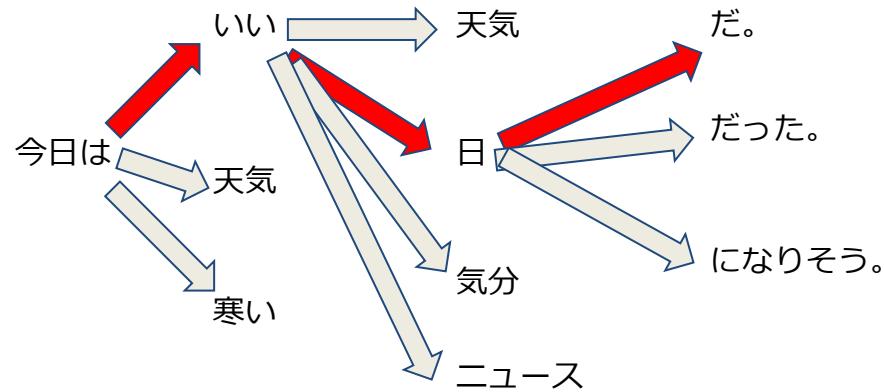
生成AIの原理

学習したパターンに基づいてプロンプトに沿ったものを生成

テキスト生成

- ・LLM(大規模言語モデル)
膨大なテキストデータから言語の
パターンを学習

確率の高い次の単語を予測し、候補
の中から次の単語を選択。
繰り返すことで文章を生成



今日はいい日だ。

LLM

機能

文章の生成

例

Gemini 2.5Pro
GPT-5

生成AIサービス

LLM搭載+

履歴保持

web検索

ファイルアップロード

Gemini



今日の福岡の天気は？

◆ 思考プロセスを表示 ▾

現在
12°C
曇

最高気温: 13° 最低気温: 8° 降水確率: 40%

水曜日・福岡市



ウェザーニュース

▲ ▾ C :

① Geminiへのプロンプトを入力

+

Thinking ▾



インコの画像を生成してください

◆ かしこまりました。インコの画像を生成します。



▲ ▾ C < > :

① Geminiへのプロンプトを入力

+

2.5 Pro ▾



福岡のおいしいラーメン屋さんを調査してください

◆ リサーチ計画をまとめました。変更が必要な箇所があればお知らせください。

福岡ラーメン調査計画

- ウェブサイトをリサーチ
 - (1) グルメサイトのランキング、レビュー記事、地元のブログなどを検索し、福岡県内（特に博多、天神、長浜エリア）で評価の高いラーメン店をリストアップする。
 - (2) 福岡ラーメンの主要なスタイル（博多豚骨、長浜ラーメン、久留米ラーメン、および最近トレンドの非豚骨系など）ごとに、代表的な店舗や人気店を分類・特定する。
 - (3) 福岡名物である「屋台」でラーメンを提供する店舗について、中洲・天神・長浜などのエリア別に人気店や利用しやすさを調査する。...
- その他
- ☰ 結果を分析
- ⟳ レポートを作成
- ⌚ 数分で完了予定

計画を編集 リサーチを開始

Deep Research なしで再試行

⌚ 調べたい内容を入力

+ 🔍 Deep Research × Thinking

こんにちは、

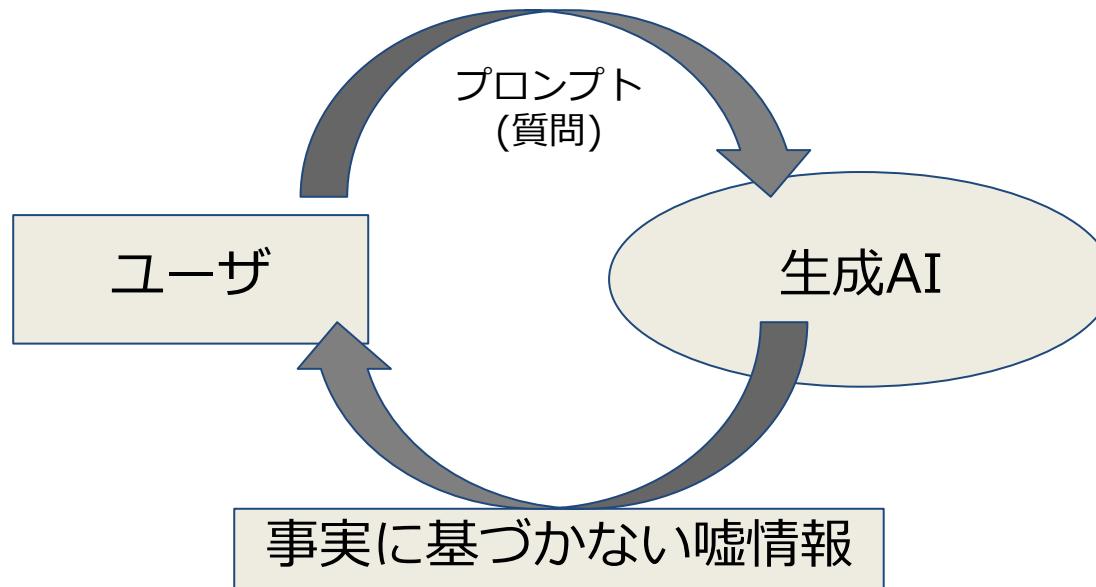
- ⌚ ファイルをアップロード
- ⌚ ドライブから追加
- ⌚ コードをインポート

Thinking

- ・文章の要約、翻訳
 - ・アイデア出し
 - ・プログラミングコードの生成
 - ・構造化データの生成
 - ・テキストによる日常的な会話
 - ・画像生成
 - ・動画生成
 - ・音楽生成
- etc

- ・最新の情報に関する回答
- ・独自の情報に関する回答
- ・学習データが少ない情報への回答
- ・情報の正確性の保証
- ・複雑な計算
- ・現実世界やデジタル環境の操作

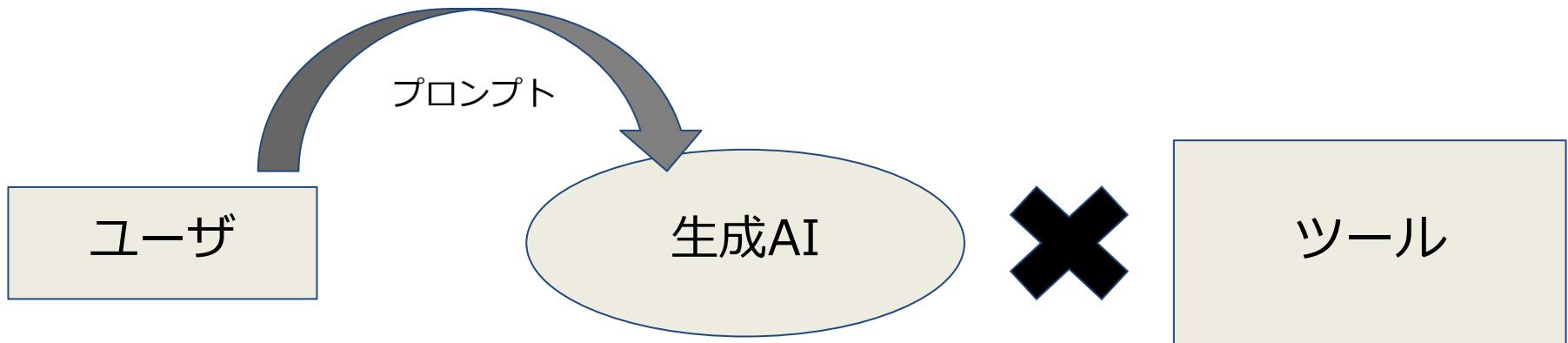
事実に基づかない嘘情報を出力してしまう現象



他のツールと組み合わせる



生成AIにツールを使わせて、
生成AI単体でできないことを可能にする



-
- 1.生成AIって何？
 - 2.生成AI×ツールの実例
 - 3.生成AIにロボットを操作させる
 - 4.まとめ

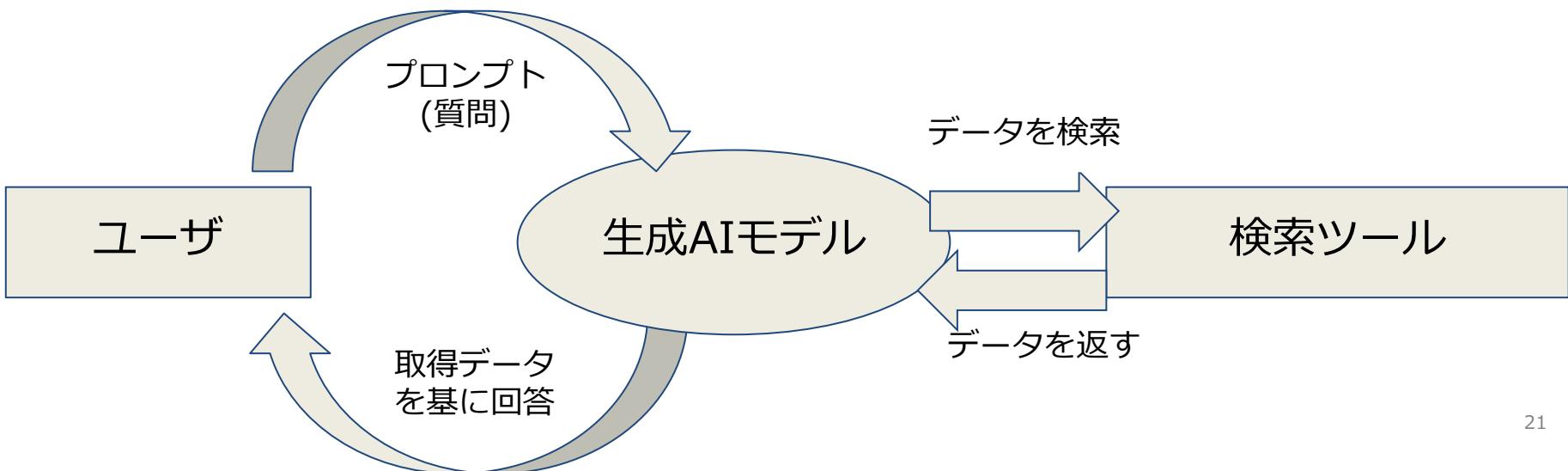
- RAG(Retrieval-Augmented Generation)
- FunctionCalling
- MCP(Model Context Protocol)

Retrieval・・・検索を意味する

Augmented・・・拡張されたという意味を持つ

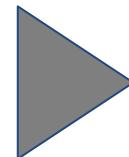
Generation・・・生成を意味する

生成AI×検索ツール

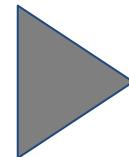


生成AI単体

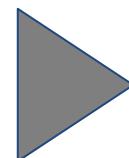
根拠のない回答



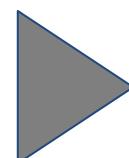
学習時点までの古い情報



学習したことのみ回答可



知識の更新が困難



RAG

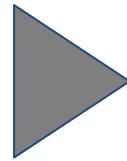
根拠のある回答

リアルタイムな情報

学習していない
独自の情報も回答可

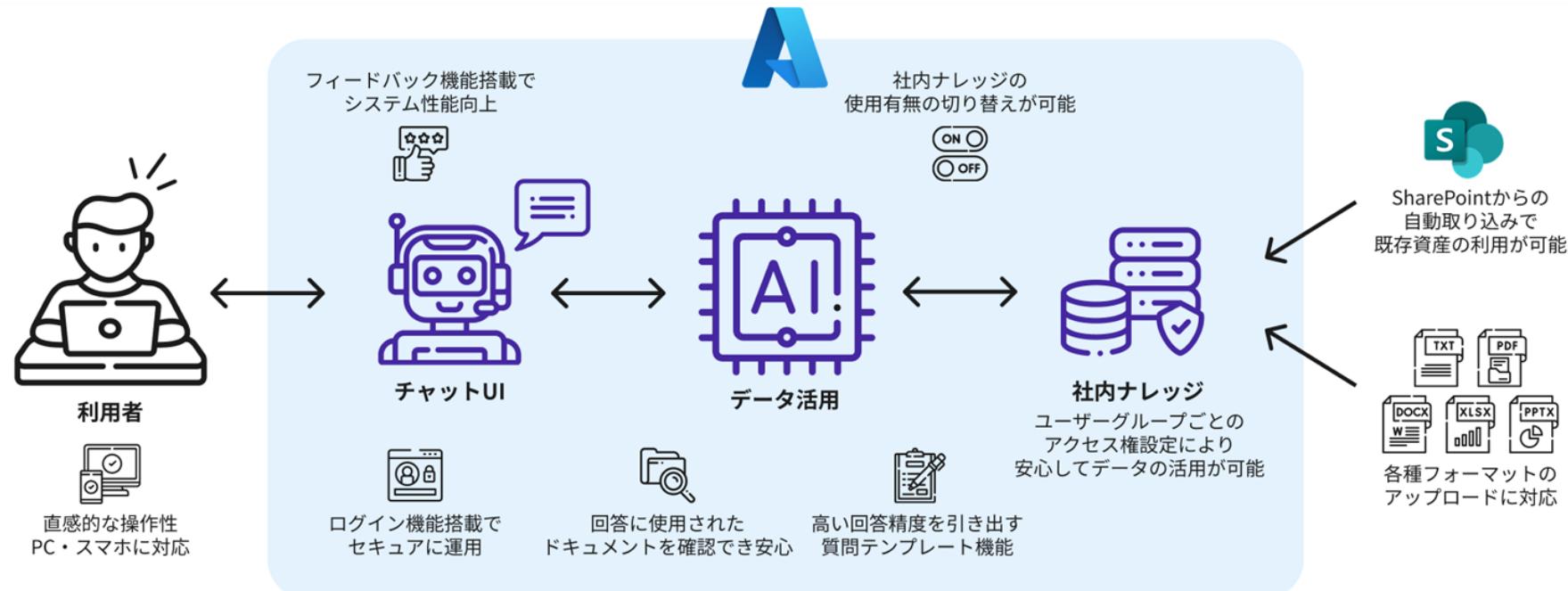
知識の更新が容易

- ・最新の情報に関する回答
 - ・独自の情報に関する回答
 - ・学習データが少ない情報への回答
 - ・情報の正確性の保証
-
- ・複雑な計算
 - ・現実世界やデジタル環境の操作



RAGを使用す
ることで対応可
能

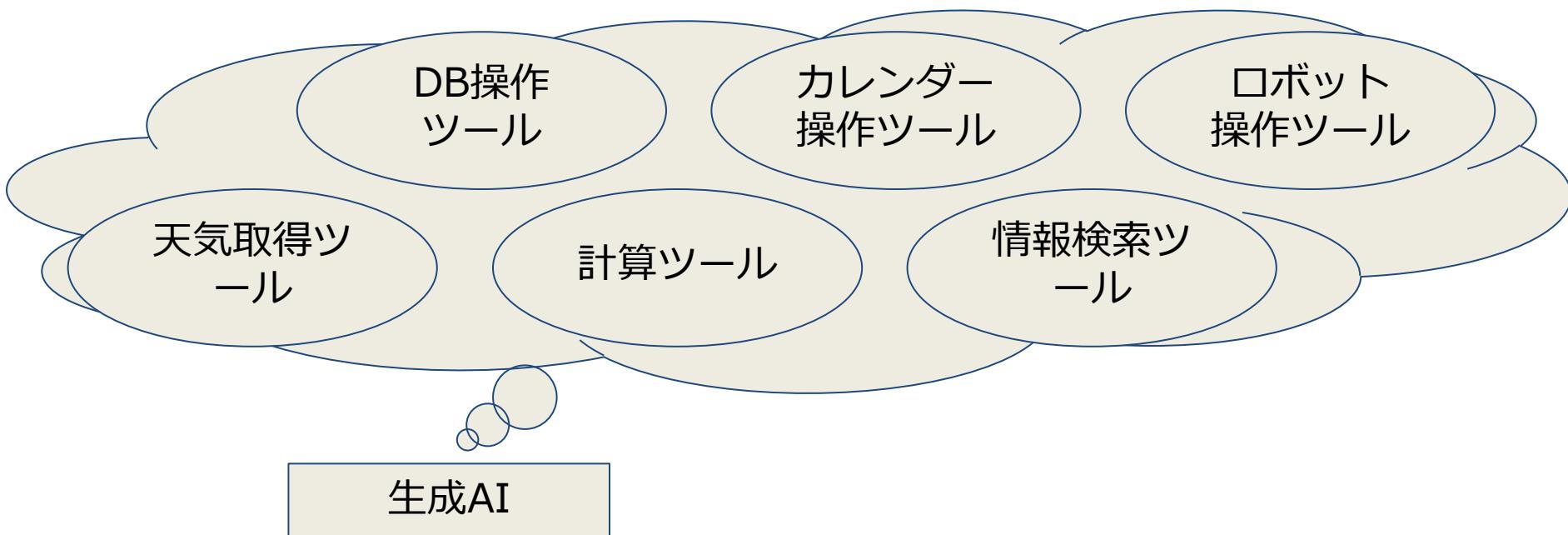
社内ナレッジ活用AIチャット導入サービス



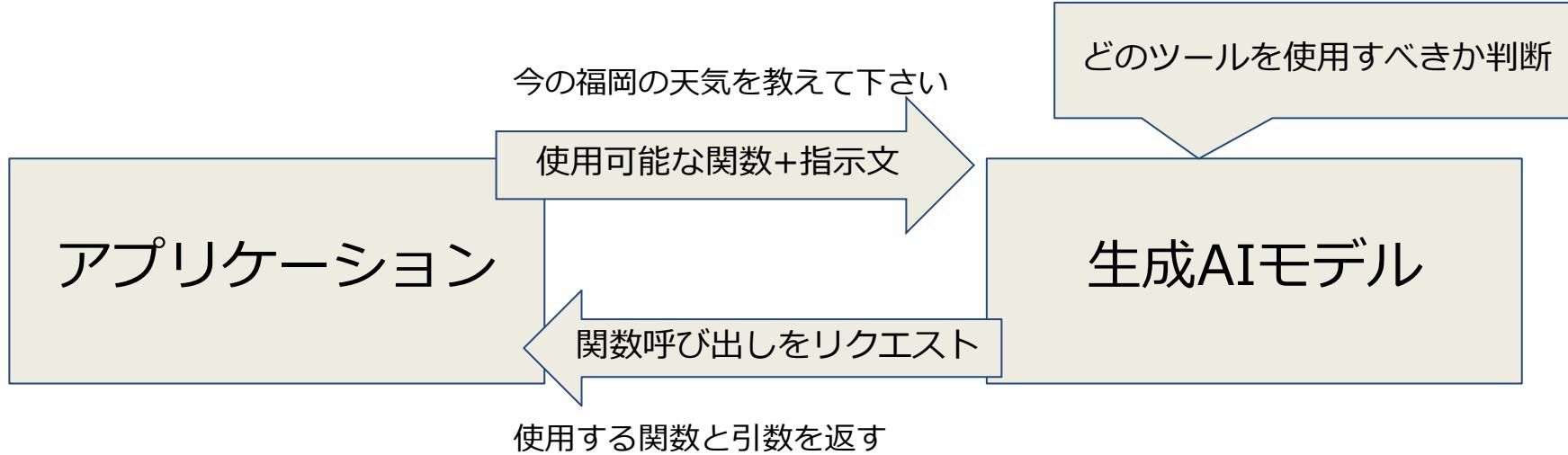
大規模な情報や社内のみで扱いたい情報等でご活用いただけます。

- RAG(Retrieval-Augmented Generation)
- FunctionCalling
- MCP(Model Context Protocol)

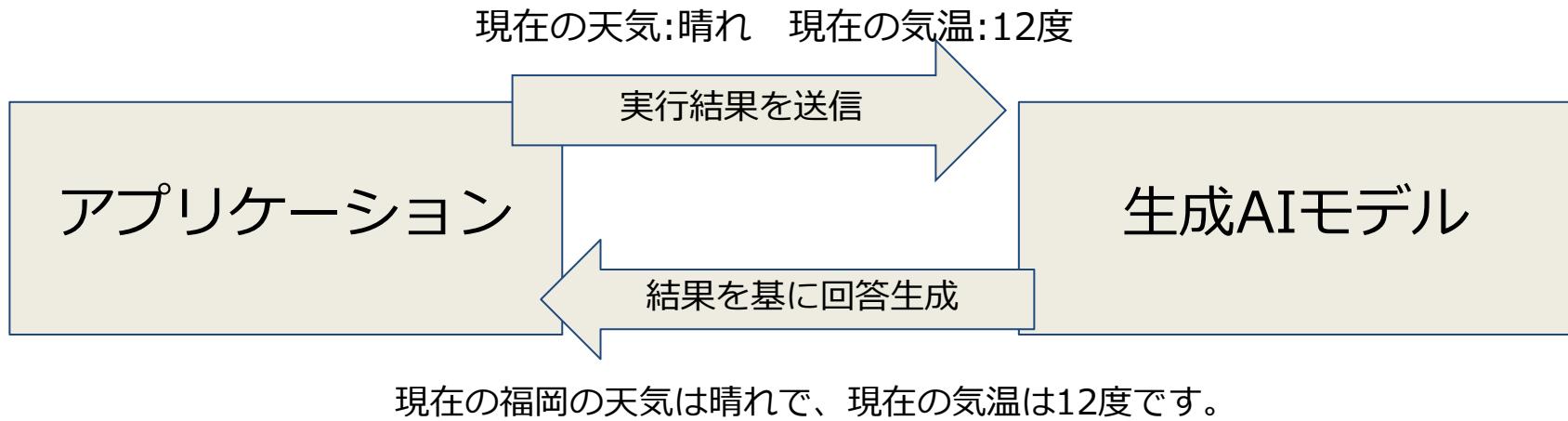
生成AIに自律的に関数を実行させることが可能



生成AIにツールを使用させられる。



外部APIやデータベースと連携させることができる



FunctionCallingでできること

リアルタイムな情報を生成AIに
回答させる
天気や最新のニュース等

IoTと連携して現実に干渉させる
ロボットを動作させる、電気つける等

カレンダーへのイベント登録や
メール送信させる
Gmailやnotionなど

計算処理させる
AIモデルの苦手な計算だけ関数にさせる等

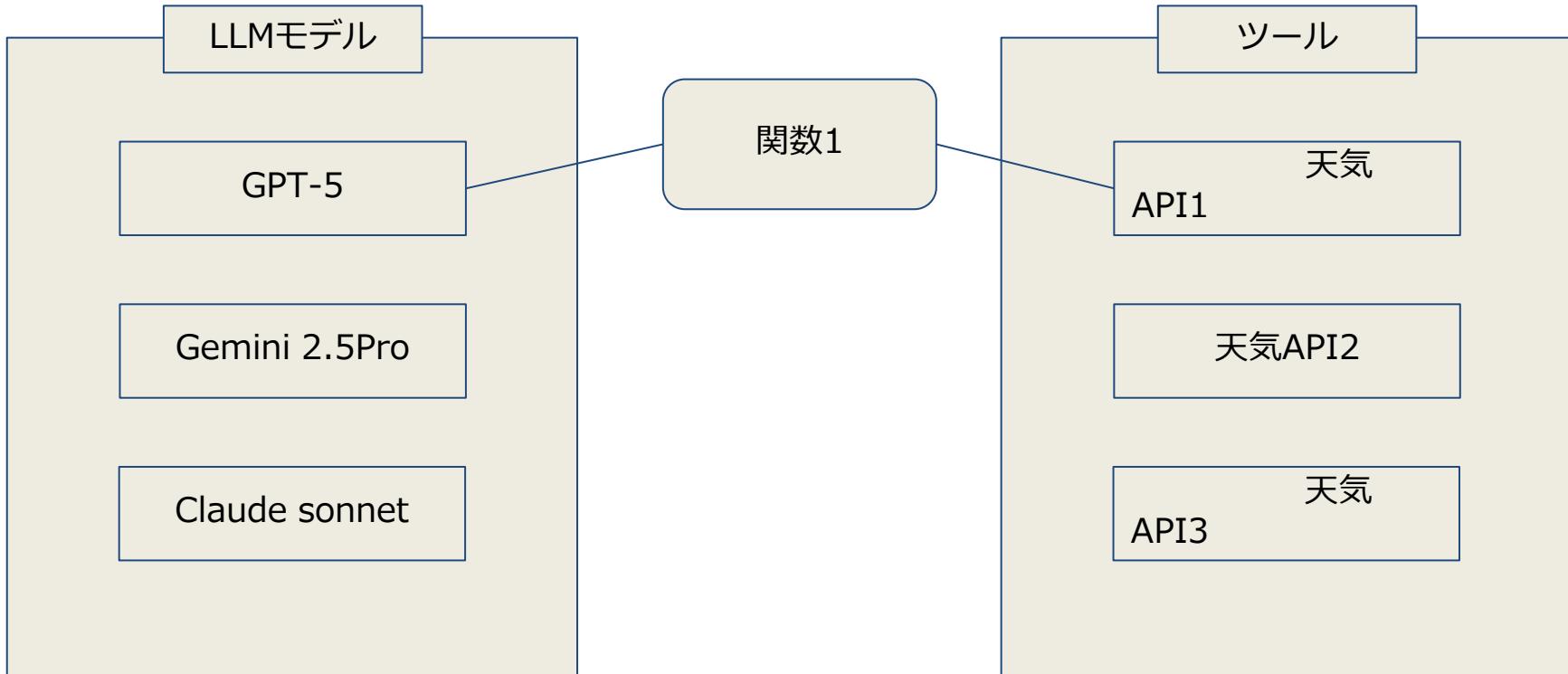
MCP(Model Context Protocol)

LLMが外部ツール等を使用するための共通規格

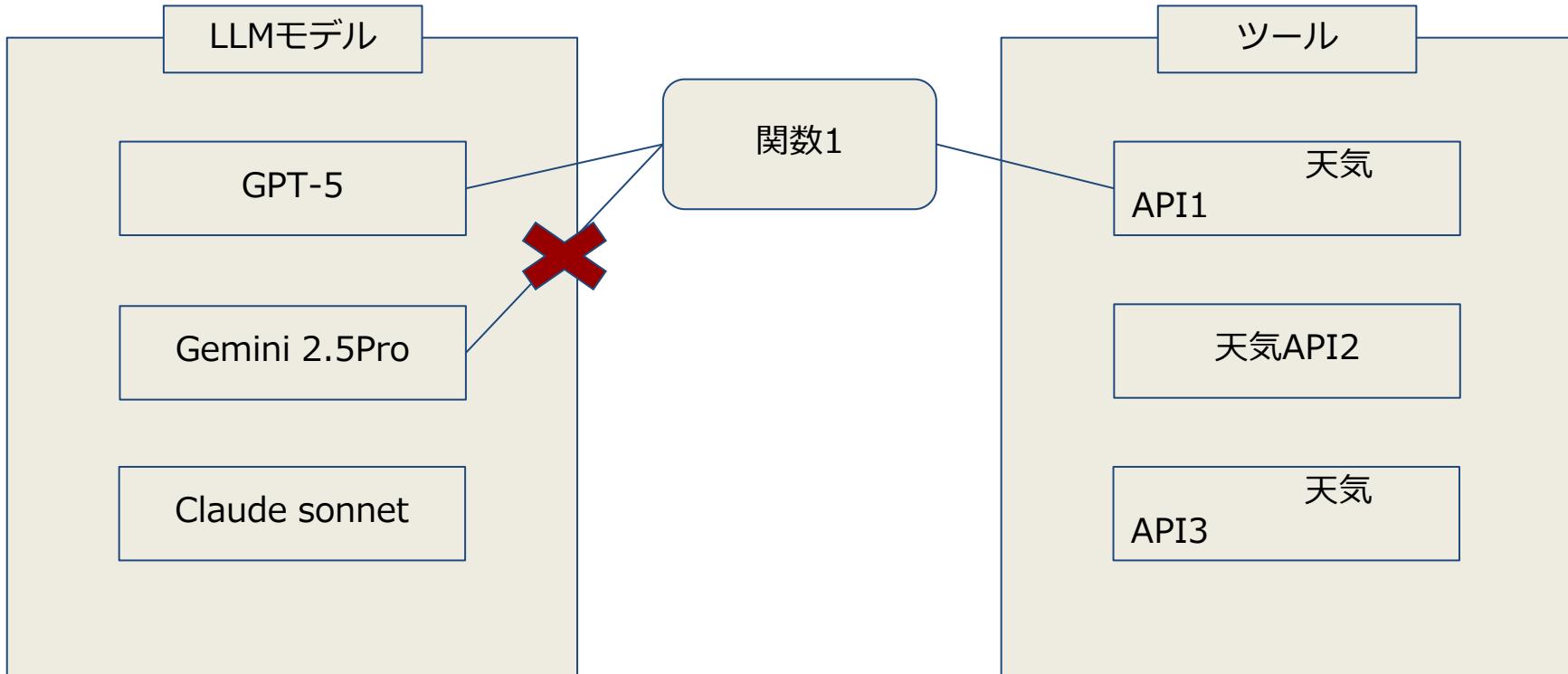
Claudeの開発を行っているAnthropic社が提唱

LLMとツールの連携が容易になる

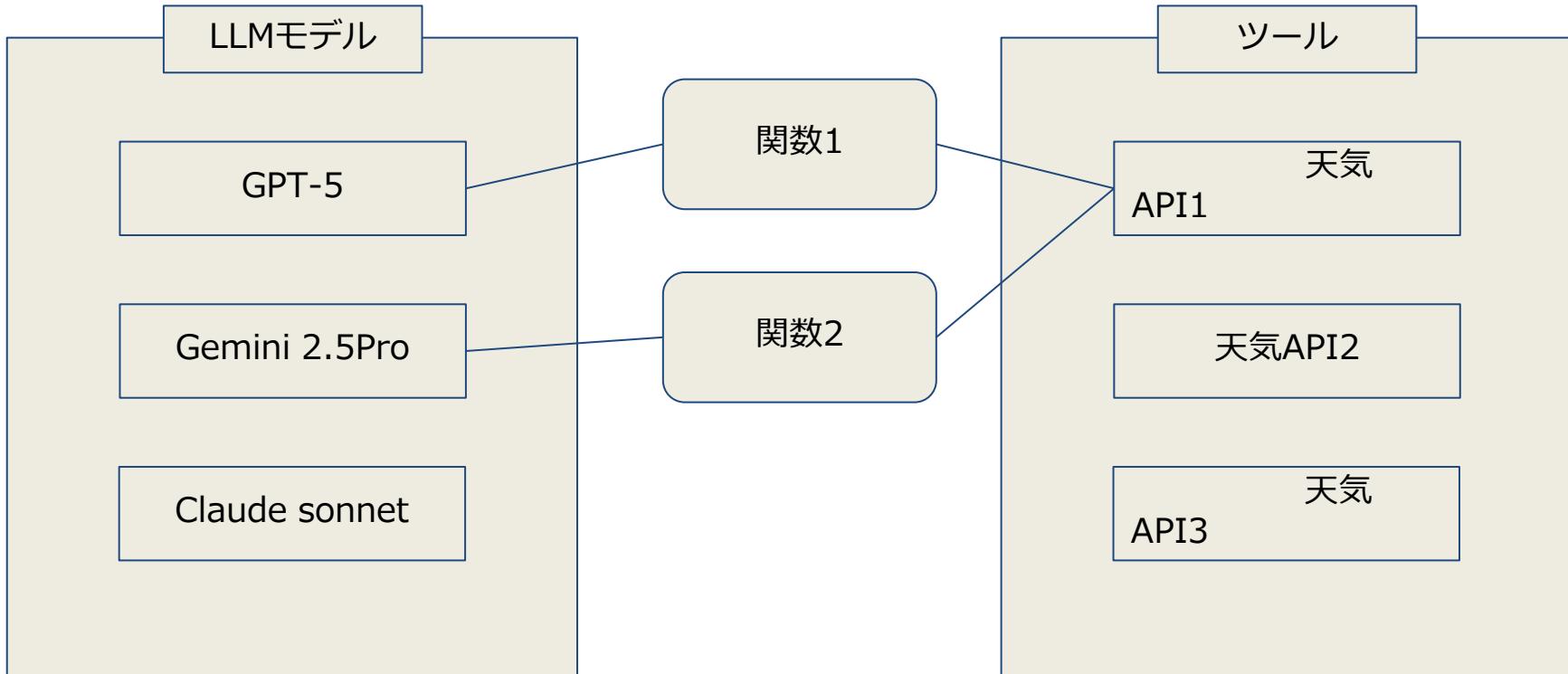
MCPがない場合



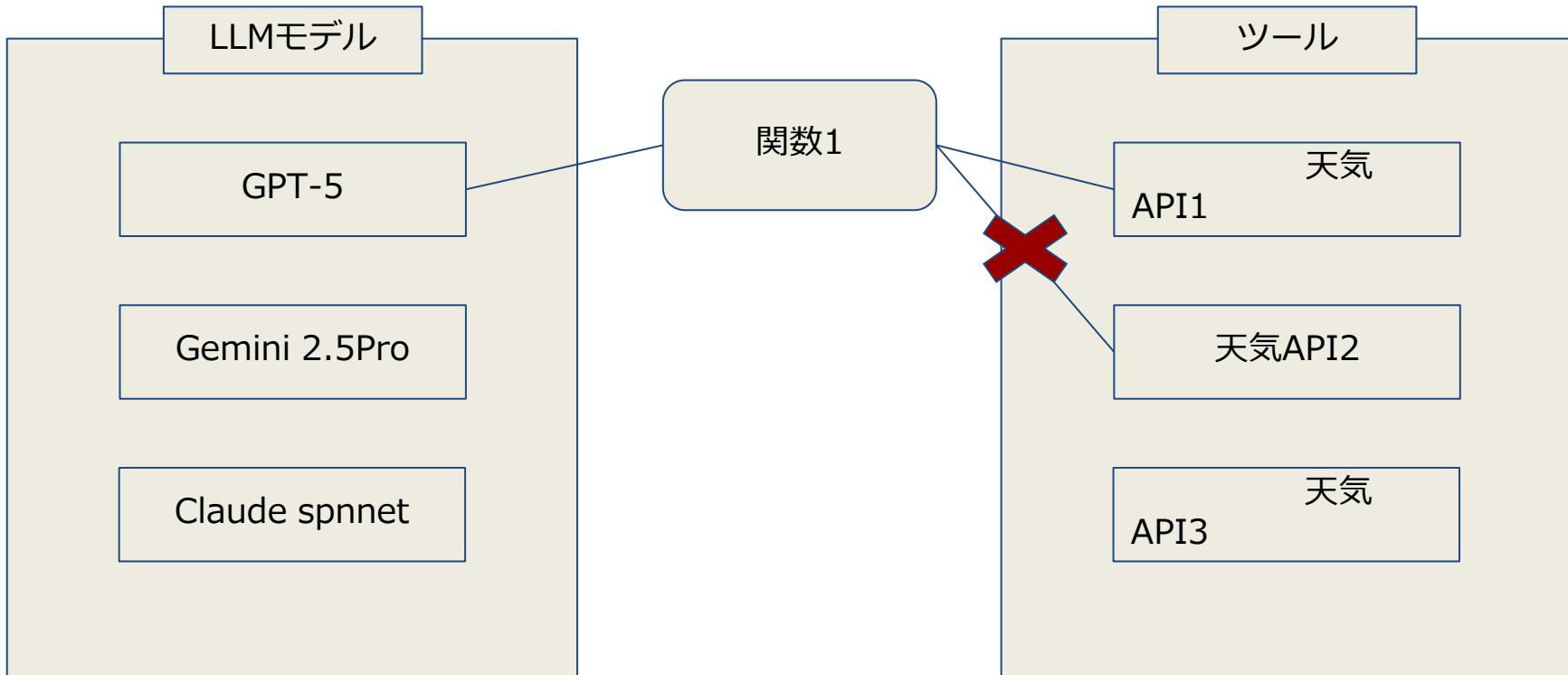
MCPがない場合



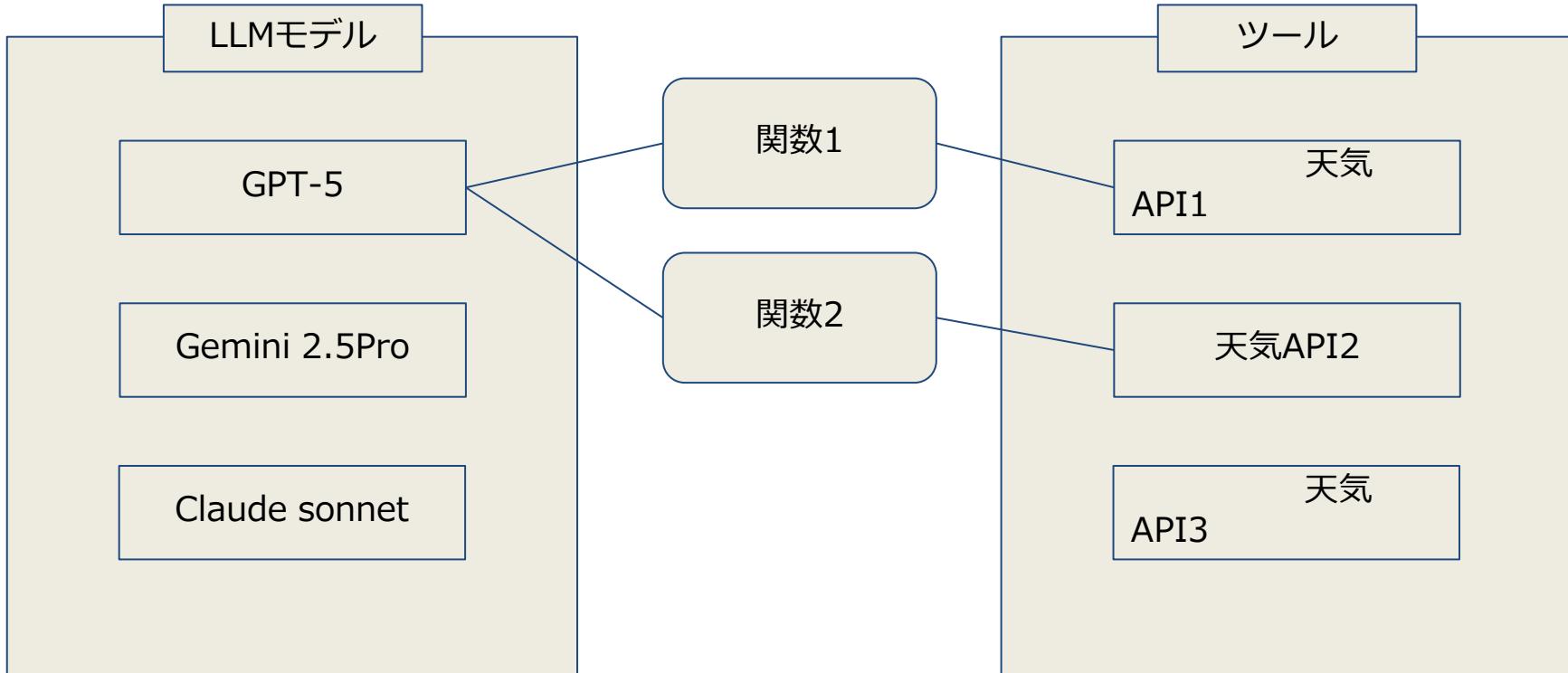
MCPがない場合



MCPがない場合



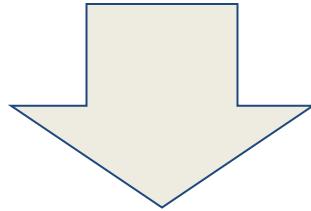
MCPがない場合



MCPがない場合

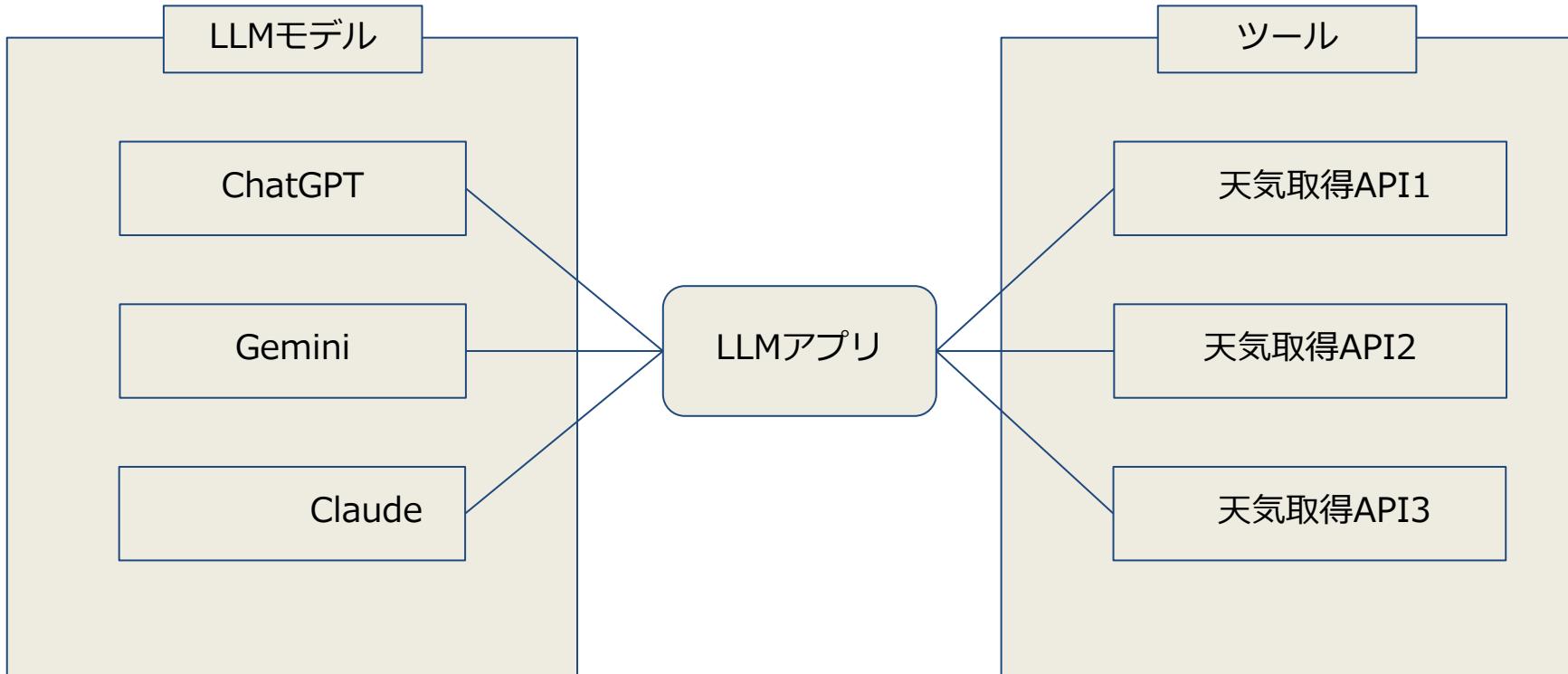


- FunctionCallingの仕様はモデル毎に異なる
- 外部API、ツールの仕様もツール毎に異なる



- モデルとツールの組み合わせの数だけ個別に連携部分を実装する必要がある

MCPがある場合



- ・MCPという共通のプロトコルがあるので、モデル、ツールの切替、追加が容易になる
- ・多くのツール(MCPサーバ)が公開されている

- notion: ドキュメントの作成・検索等
- blender: 3Dモデルの作成
- Googleカレンダー: イベントの追加や検索
- Googleスプレッドシート: スプレッドシートの編集
- ファイルシステムMCPサーバ: ローカルファイルの読み取り、書き込み、検索等

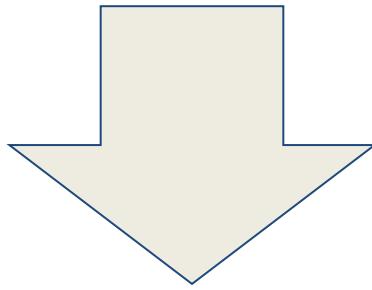
他にも多くのMCPサーバが提供されています。

<https://github.com/modelcontextprotocol/servers>

MCPのトークン消費問題



MCPを使用して多くのツールを生成AIに使用させられるようになった



ツールを追加するにしたがってトークン消費量が多くなるという問題点も存在

プロンプトを生成AIモデルはどう理解するのか

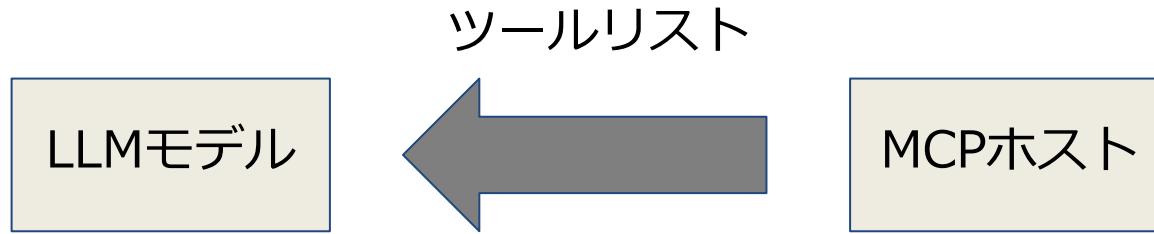
今日はいい日だ

トークンに分割

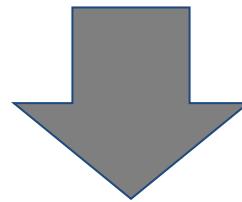
今日/は/いい/日/だ/

トークン=分割の最小単位

生成AIモデルはトークン数に応じてコストがかかる
処理できるトークン数はモデル毎に上限がある



ツールが増えれば生成AIが処理するトークン数が増加



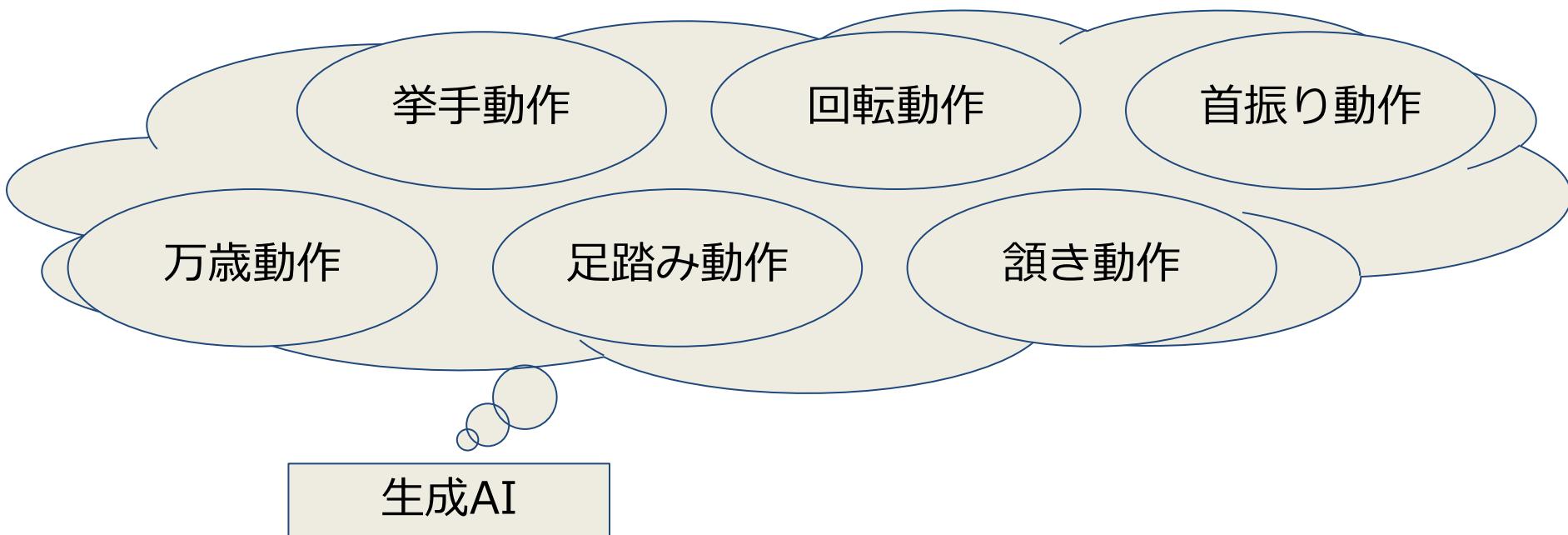
コスト増加・生成AIの性能低下

生成AIにツールを使用させることで、生成AI単体ではできなかった

- ・最新の情報に関する回答
 - ・独自の情報に関する回答
 - ・学習データが少ない情報への回答
 - ・複雑な計算
 - ・外部ツールや現実世界への干渉
- をさせることが可能

-
- 1.生成AIって何？
 - 2.生成AI×ツールの実例
 - 3.生成AIにロボットを操作させる
 - 4.まとめ

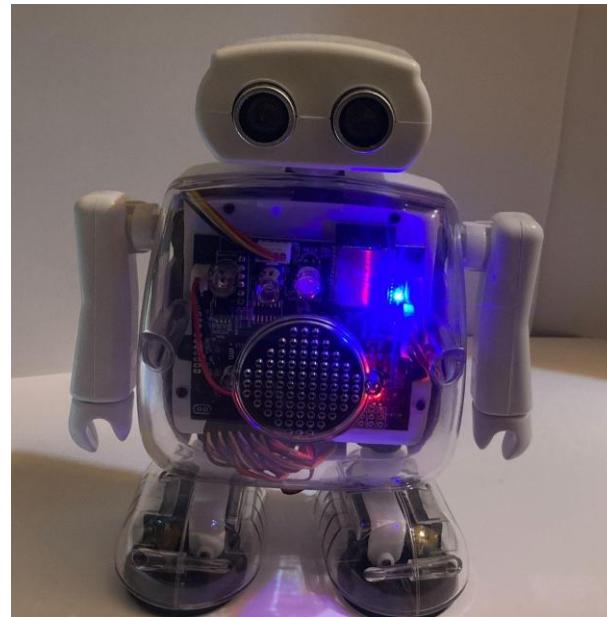
FunctionCallingで生成AIにロボットを操作させる



小型ロボット。PCとbluetooth接続可能

主な機能

- ・距離センサ
- ・頭、腕、足の回転
- ・音検知
- ・発声
- ・LEDの発光



MicrosoftのクラウドサービスAzure上でOpenAI社のAIモデル
を利用できるようにしたサービス
生成AI機能をアプリに取り込むことが可能

```
response =  
client.chat.completions.create(  
    model=deployment_name,  
    messages=messages,  
    max_tokens=1000,  
    temperature=0.7,  
    stream=False,  
)
```

messagesに具体的なプロンプトを含む
会話履歴を渡すことでresponseに
生成AIからの回答が返ってくる

ARIA - インタラクティブAIロボット

音声認識状態

• 待機中

ロボット状態

未接続

[QumCumに接続](#)

リアルタイム文字起こし

AI分析結果

感情:

重要度:

ロボット表情:

生データ:

宝くじやかた動画

アプリケーションUI



AI分析結果

感情:

重要度:

ロボット表情:

生データ:

実行された動作:

テキスト入力

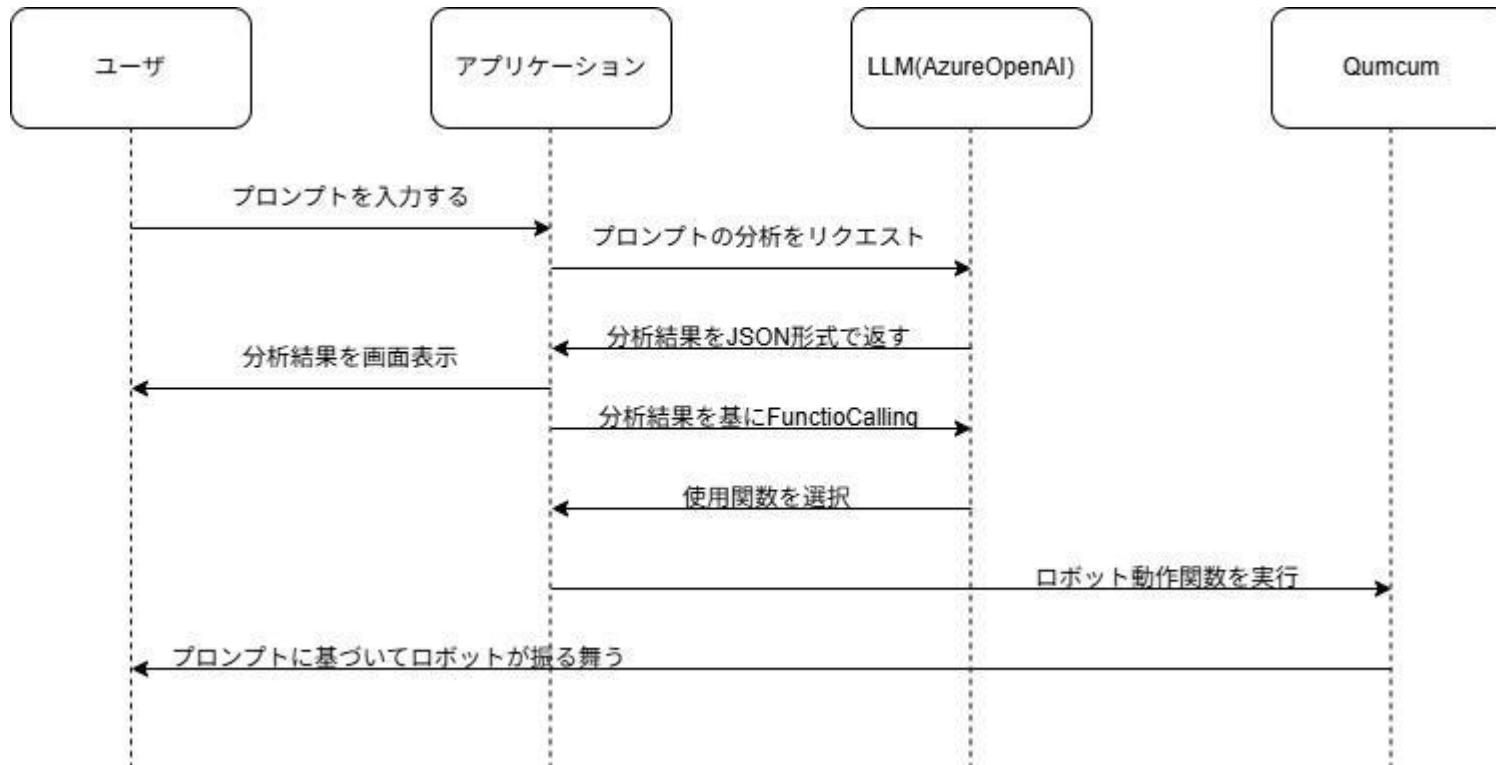
テキストを入力してください

分析実行

▶ 設定

▶ 統計・ログ

シーケンス図



AzureOpenAIで構造化レスポンス



```
prompt = f"""
```

以下の文章を分析し、指定されたスキーマに従ってJSON形式で出力してください。

文章: "{text_analysis}"

スキーマ: "{schema_string}"
"""

```
"emotion": {  
    "description": "文章全体の感情的なトーン",  
    "enum": [  
        "ポジティブ",  
        "ネガティブ",  
        "ニュートラル"  
    ],  
    "title": "Emotion",  
    "type": "string"  
}
```

emotion	プロンプト全体の感情的なトーン
robot_emotion	プロンプトに対してロボットが表現すべき感情の種類(嬉しい、悲しい等)
importance	文章の重要度(1~5で評価)
summary	文章の要約

AzureOpenAIで構造化レスポンス



```
response = client.chat.completions.create(  
    model=deployment_name,  
    messages=[  
        {  
            "role": "system",  
            "content": system_message,  
        },  
        {"role": "user", "content": prompt},  
    ],  
    response_format={"type": "json_object"},  
    max_tokens=1000,  
    temperature=0.2,  
    stream=False,  
)
```

response_formatを指定することで
JSON形式で返答させることを保証

AI分析結果

感情:

重要度:

ロボット表情:

生データ:

実行された動作:

テキスト入力

テキストを入力してください

福岡でおいしいものを食べたいと思います

分析実行

› 設定

› 統計・ログ

AI分析結果



感情: ポジティブ

重要度: ★★★★☆

ロボット表情: 喜び 😊

生データ: { "emotion": "ポジティブ", "robotEmotion": "喜び", "importance": 4, "summary": "福岡でおいしい食べ物を楽しみたいという願望" }

実行された動作: banzai

AzureOpenAIでFunctionCalling



```
response =  
client.chat.completions.create(  
    model=deployment_name,  
    messages=messages,  
    tools=tools,  
    tool_choice=tool_choice,  
    temperature=0.7,  
)
```

functionCallingで選択可能な関数リスト

none,autoを指定可
autoにすると独自の判断でツールを呼び出すか否か、どのツールを呼び出すか選択してくれる

ツール

```
{  
  "type": "function",  
  "function": {  
    "name": "nod",  
    "description": "傾き動作を実行する",  
    "parameters": {  
      "description": "引数なし用の空スキーマ",  
      "properties": {},  
      "title": "EmptyArgs",  
      "type": "object"  
    }  
  }  
},
```

プロンプト

```
messages = [  
  {  
    "role": "system",  
    "content": "あなたはロボットの動作選択  
AIアシスタントです。分析結果に基づいて最適な  
動作を選択してください。",  
  },  
  {  
    "role": "user",  
    "content": f"以下の分析結果に基づいて、  
最適な推奨動作を1つだけ選択してください。分  
析結果: {analysis_response.model_dump_json()}"  
  }  
]
```

AI分析結果



感情: ポジティブ

重要度: ★★★★☆

ロボット表情: 喜び 😊

生データ: { "emotion": "ポジティブ", "robotEmotion": "喜び", "importance": 4, "summary": "福岡でおいしい食べ物を楽しみたいという願望" }

実行された動作: banzai



音声認識をさせてロボットを自然に動作させる
→プレゼンテーションの練習等に活用

Qumcumの発声機能を使用して、音声認識と合わせて自然な会話をさせることもできるかもしれない

-
- 1.生成AIって何？
 - 2.生成AI×ツールの実例
 - 3.生成AIにロボットを操作させる
 - 4.まとめ

生成AI単体ができる

- ・文章の要約、翻訳
- ・アイデア出し
- ・プログラミングコードの生成
- ・構造化データの生成
- ・テキストによる日常的な会話
- ・画像生成
- ・動画生成
- ・音楽生成

生成AI単体でできない

- ・最新の情報に関する回答
- ・独自の情報に関する回答
- ・学習データが少ない情報の回答
- ・情報の正確性の保証
- ・複雑な計算
- ・現実世界やデジタル環境の操作

生成AIとツールをかけ合わせる技術が存在する

- RAG(Retrieval-Augmented Generation)
- FunctionCalling
- MCP(Model Context Protocol)

生成AI×ツールで生成AI単体でできなかつたことが可能になる

- ・最新の情報に関する回答
- ・独自の情報に関する回答
- ・学習データが少ない情報への回答
- ・情報の正確性の保証
- ・複雑な計算
- ・現実世界やデジタル環境の操作

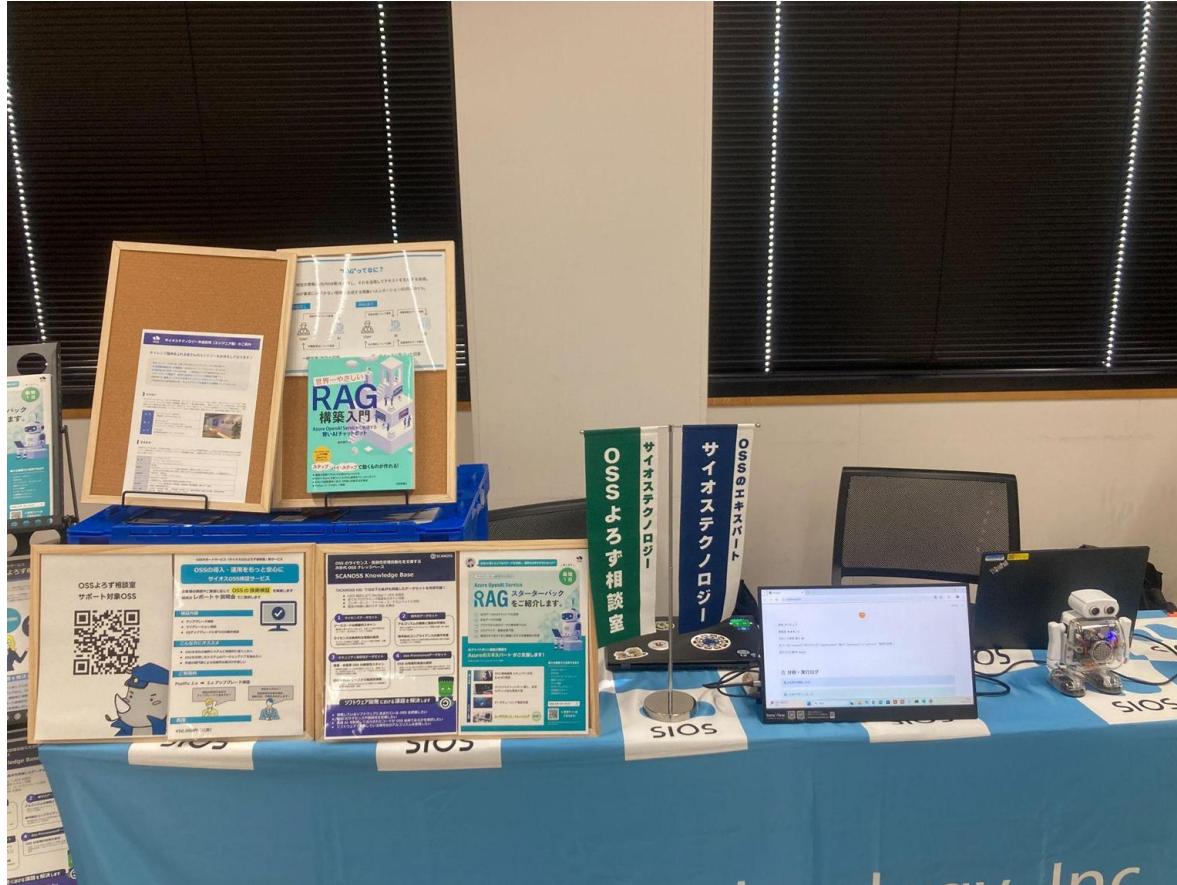
展示紹介

展示ブースレイアウト

※当日変更になっている場合もあります



展示紹介



ご清聴ありがとうございました

