

# ADK + toolbox を使ってデータマネジメントやってみた話

hiracky16 (ひらき)

# 自己紹介

- 名前: Noriaki Hiraki (X: [hiracky16](#))
- 所属: ファインディ株式会社 データエンジニア
- 興味: サッカー観戦🏈、ハロプロ🌸、ポッドキャスト🎙️、子育て👶
- ひとこと: 9月に福岡に引っ越してきました🍜よろしくをお願いします！（出身は熊本🐻）



# いきなりデモ

- 題材として企業マスタのメンテナンスを想定
- ユースケースとして以下を考える
  - 新しい企業の登録
  - 正しい法人番号の更新（法人番号の検索）
  - 重複レコードの削除
  - 所在地による営業リストの抽出
- QR コードは GitHub リポジトリへのリンク
  - <https://github.com/hiracky16/adk-data-management-agent>
- スターお待ちしております！！



# データマネジメントとは？

企業が保有するデータを、目的の利用目的に応じていつでも活用できる状態を維持し、継続的に管理していく組織的な取り組みです。データの登録、更新、蓄積、アーカイブ、破棄など、データのライフサイクル全体を管理し、品質向上と効率的な利活用を目指します。

データマネジメントを継続的に回すためデータガバナンスがあり、現場の Ops に対してガバナンスを効かせることが非常に重要

# データライフサイクルにおける課題感

- 登録: 決めたルールで入力されない
- 更新: 一度入力して最新化されない
- 蓄積: 決められた場所に保存されない
- 破棄: 古く使われていないデータがそのまま残っている
- 決めたルールやオペレーションを運用するのに様々なコストが掛かってしまう
  - 実際に人手でやるのは難しい
  - AIによって簡単にできないだろうか...? 🤔

# ADK (Agent Development Kit) とは？

- Google 製の AI エージェント開発・展開のための柔軟なフレームワーク
- Gemini と Google エコシステムに最適化されつつ、モデルやデプロイメントに依存しない
- エージェント開発をソフトウェア開発のようにし、複雑なワークフローを容易に構築可能にする



# たった 14 行でエージェントが開発できる

```
1 from google.adk.agents.llm_agent import Agent
2
3 # Mock tool implementation
4 def get_current_time(city: str) -> dict:
5     """Returns the current time in a specified city."""
6     return {"status": "success", "city": city, "time": "10:30 AM"}
7
8 root_agent = Agent(
9     model='gemini-2.5-flash',
10    name='root_agent',
11    description="Tells the current time in a specified city.",
12    instruction="You are a helpful assistant that tells the current time in cities. Use the
    'get_current_time' tool for this purpose.",
13    tools=[get_current_time],
14 )
```

```

1 from google.adk.agents import LlmAgent
2 import os
3 from google.adk.tools.mcp_tool.mcp_toolset import MCPToolset, StdioConnectionParams,
  StdioServerParameters
4
5 GITHUB_PERSONAL_ACCESS_TOKEN = os.getenv("GITHUB_PERSONAL_ACCESS_TOKEN", "hogefugapiyo")
6 INSTRUCTION = "あなたはGitHubを操作するエージェントです"
7
8 mcp_toolset = MCPToolset(
9     connection_params=StdioServerParameters(
10         command='/usr/local/bin/github-mcp-server',
11         args=["stdio"],
12         env={
13             "GITHUB_PERSONAL_ACCESS_TOKEN": GITHUB_PERSONAL_ACCESS_TOKEN,
14         },
15     ),
16 )
17
18 root_agent = LlmAgent(
19     model="gemini-2.5-flash",
20     name="github_agent",
21     description="github agent",
22     instruction=INSTRUCTION,
23     tools=[mcp_toolset]
24 )

```

```

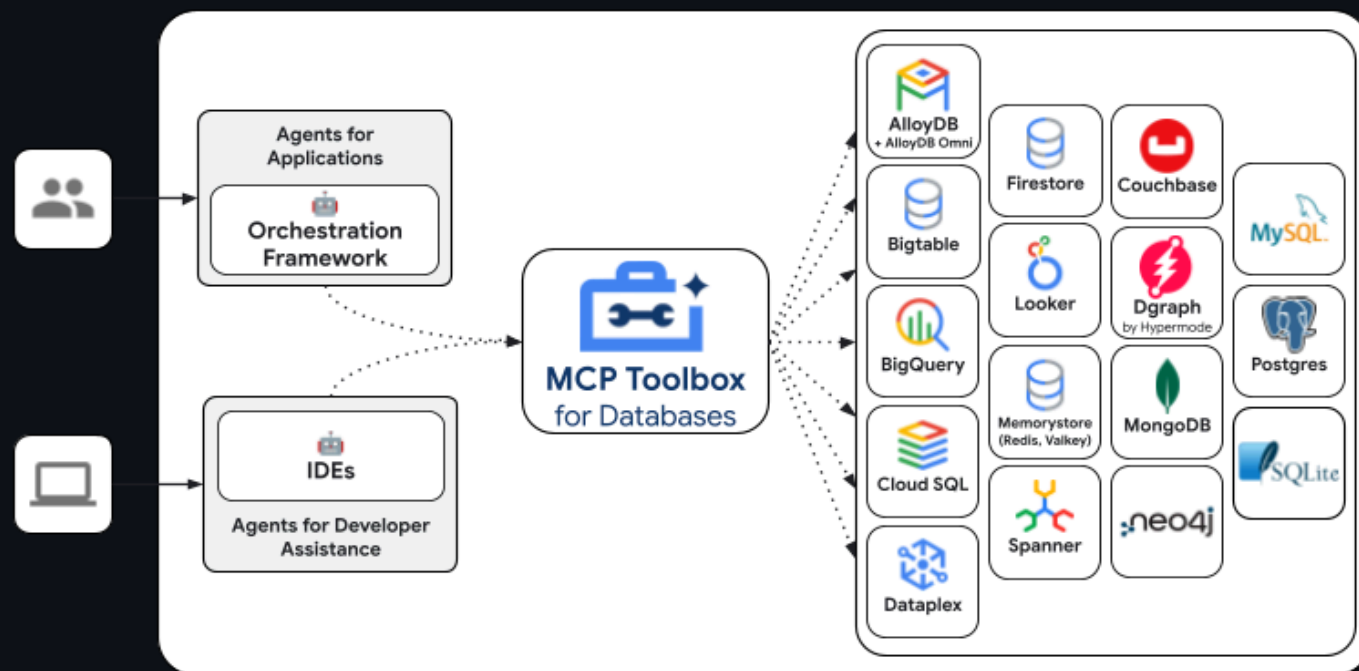
1 from google.adk.agents import LlmAgent
2
3 SUB_AGENT_INSTRUCTION = 'サブエージェントとして親エージェントを支えてあげて'
4
5 INSTRUCTION = '親らしくサブエージェントと仲良くやりなさい'
6
7 sub_agent = LlmAgent(
8     model="gemini-2.5-flash",
9     name="sub_agent",
10    description="sub agent",
11    instruction=SUB_AGENT_INSTRUCTION,
12    tools=[mcp_toolset]
13 )
14
15 root_agent = LlmAgent(
16     model="gemini-2.5-flash",
17     name="github_agent",
18     description="github agent",
19     instruction=INSTRUCTION,
20     sub_agents=[sub_agent]
21 )

```



# MCP Toolbox for Database とは？

- Google 製の Database のための MCP サーバー
- MySQL や PostgreSQL, BigQuery, Looker など対応サービスが豊富
- prebuild されたツールや自前のツールを容易に開発可能



# YAML でツールを用意

```
1 sources:
2   my-bigquery-source:
3     kind: bigquery
4     project: YOUR_PROJECT_ID
5     location: us
6 tools:
7   search-hotels-by-name:
8     kind: bigquery-sql
9     source: my-bigquery-source
10    description: Search for hotels based on name.
11    parameters:
12      - name: name
13        type: string
14        description: The name of the hotel.
15    statement: SELECT * FROM `YOUR_DATASET_NAME.hotels` WHERE LOWER(name) LIKE LOWER(CONCAT('%',
16      @name, '%'));
17   search-hotels-by-location:
18     kind: bigquery-sql
19     source: my-bigquery-source
20    description: Search for hotels based on location.
21    parameters:
22      - name: location
23        type: string
24        description: The location of the hotel.
25    statement: SELECT * FROM `YOUR_DATASET_NAME.hotels` WHERE LOWER(location) LIKE LOWER(CONCAT('%',
26      @location, '%'));
27   book-hotel:
28     kind: bigquery-sql
29     source: my-bigquery-source
30    description: >-
31      Book a hotel by its ID. If the hotel is successfully booked, returns a NULL, raises an error if
32      not.
33    parameters:
34      - name: hotel_id
35        type: integer
36        description: The ID of the hotel to book.
37    statement: UPDATE `YOUR_DATASET_NAME.hotels` SET booked = TRUE WHERE id = @hotel_id;
```

# ADK + Toolbox の連携

- Toolbox でデータのライフサイクル（登録～破棄）を行うツールを開発し MCP サーバーとしてホスティング
  - BigQuery を操作するための SQL をあらかじめ記述したツールをいくつか用意
- ADK を使って Toolbox MCP のクライアントとして自然言語でデータを操作できるエージェントを開発
  - `adk web` コマンドでチャット UI を提供

## 工夫点

- YAML にデータ登録や更新用の SQL をあらかじめ定義できるためハルシネーションなどの心配がない
- ADK の実装と `adk web` を使った Web UI によって爆速で提供可能
  - `adk run` による CLI ベースでの実行も可能なため全自動でも可能
- 自然言語による操作が可能なため Ops への負担が少ない

# まとめ

- 価値あるデータを蓄積するためにデータマネジメントが必要だが、ルールを決めて運用し続けるにはコストが掛かる
- ADK でエージェントを開発するのがすごく簡単
- Toolbox を使って BigQuery を操作するツール開発するのもすごく簡単
- 2つを組み合わせると爆速でデータマネジメントツールが開発でき、登録から破棄までのコストが抑えられる（かもしれない） ... !

ご清聴ありがとうございました！

# Appendix

## 企業リストの表示

データを初期化して



⚡ init-company-master



✓ init-company-master



データが初期化されました。

企業リストを表示して



⚡ list-companies



✓ list-companies

現在登録されている企業は以下の通りです:

- A株式会社 (ID: 43ded87b-a12c-43b0-a675-3662bb8e70ba, 法人番号: 0000000000000, 所在地: 東京都港区1-1-1)
- B株式会社 (ID: 8ca3ddf1-c3ba-4321-ad23-18cbb502c0a2, 法人番号: 00000000000001, 所在地: 東京都港区2-2-2)
- C株式会社 (ID: 94dff6c9-e9e2-4ec3-a805-e55ac08ddf1d, 法人番号: 00000000000002, 所在地: 東京都港区3-3-3)
- C株式会社 (ID: 164c660a-8f1b-411f-8d38-0959a3de37ee, 法人番号: 00000000000002, 所在地: 東京都港区3-3-3)
- D株式会社 (ID: 85032cd6-d925-4012-bb41-a3388923a087, 法人番号: 00000000000002, 所在地: 東京都品川区3-3-3)



# Appendix

## 企業の追加

ファインディ株式会社を追加したいです  
法人番号と所在地はわからないので Google で検索してからその値を埋めて



ファインディ株式会社の法人番号と所在地をGoogleで検索します。



⚡ google\_search\_agent



✓ google\_search\_agent



ファインディ株式会社の法人番号と所在地が見つかりました。  
法人番号: 8011001098775  
所在地: 東京都品川区大崎1-2-2 アートヴィレッジ大崎セントラルタワー 5階  
これらの情報を使って、ファインディ株式会社を追加します。



⚡ add-company



✓ add-company



ファインディ株式会社が追加されました。



# Appendix

## 重複企業の削除

企業を表示して



⚡ list-companies



✓ list-companies

現在登録されている企業は以下の通りです:

- A株式会社 (ID: 43ded87b-a12c-43b0-a675-3662bb8e70ba, 法人番号: 0000000000000, 所在地: 東京都港区1-1-1)
- B株式会社 (ID: 8ca3ddf1-c3ba-4321-ad23-18cbb502c0a2, 法人番号: 00000000000001, 所在地: 東京都港区2-2-2)
- C株式会社 (ID: 94dff6c9-e9e2-4ec3-a805-e55ac08ddf1d, 法人番号: 00000000000002, 所在地: 東京都港区3-3-3)
- C株式会社 (ID: 164c660a-8f1b-411f-8d38-0959a3de37ee, 法人番号: 00000000000002, 所在地: 東京都港区3-3-3)
- D株式会社 (ID: 85032cd6-d925-4012-bb41-a3388923a087, 法人番号: 00000000000002, 所在地: 東京都品川区3-3-3)
- ファインディ株式会社 (ID: 9b6911bb-290b-4a9b-8d6a-f692b8e5b7e3, 法人番号: 8011001098775, 所在地: 東京都品川区大崎1-2-2 アートヴィレッジ大崎セントラルタワー 5階)



重複した企業を削除して



重複している企業を削除します。

C株式会社 (ID: 164c660a-8f1b-411f-8d38-0959a3de37ee) を削除します。



⚡ delete-company



✓ delete-company



重複した企業が削除されました。

# Appendix

## 所在地による営業リストの抽出

品川区の企業を抽出して



⚡ search-companies-by-location



✓ search-companies-by-location



品川区に所在地のある企業は以下の通りです。

- D株式会社 (ID: 85032cd6-d925-4012-bb41-a3388923a087, 法人番号: 00000000000002, 所在地: 東京都品川区3-3-3)
- ファインディ株式会社 (ID: 9b6911bb-290b-4a9b-8d6a-f692b8e5b7e3, 法人番号: 8011001098775, 所在地: 東京都品川区大崎1-2-2 アートヴィレッジ大崎セントラルタワー 5階)