

Gemini と Agent Development Kit で創る！ AI Agent 開発の最前線

Google Cloud Next Tokyo 25

株式会社スリーシェイク Sreake事業部 佐藤慧太

2025/08/06

佐藤 慧太@SatoJohn

- 2023/1 株式会社スリーシェイク入社
- Google Cloud Partner Top Engineer
'24、'25 選出
- お客様の労苦 <Toil>を減らす
- 娘のお世話を精一杯やっています



会社名 株式会社スリーシェイク

設立日 2015/1/15

代表者 代表取締役社長 吉田 拓真

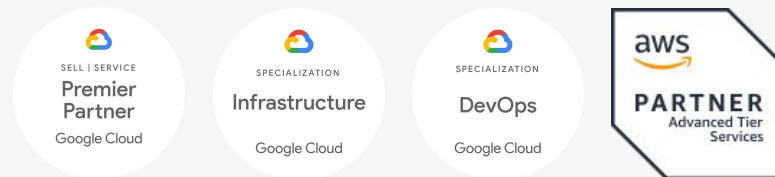
所在地 東京都新宿区大京町22-1
グランファースト新宿御苑3F・4F

沿革 2021年1月 JAFCOから総額5億円の資金調達
2022年8月 自動脆弱性診断ツール「Securify Scan」をリリース。JAFCO、MUCAPから総額8.48億円の資金調達

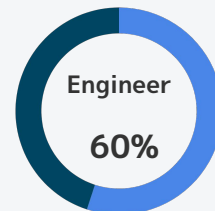
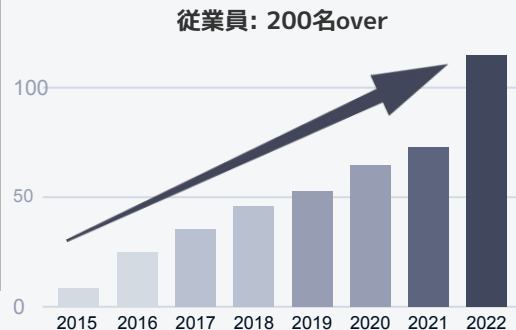
Mission: インフラをシンプルにして
イノベーションが起こりやすい世界を作る

Vision: 労苦〈Toil〉を無くすサービスを適正な価格で提供し続ける

Value: エンジニアリングレイヤーに横たわる人、手法、ツールが
サイロ化されて労苦が発生しているプロセスをシンプルにし
サービス機能開発に集中できるソリューション
(SRE、DevSecOps、DataOps、HROps) を提供する



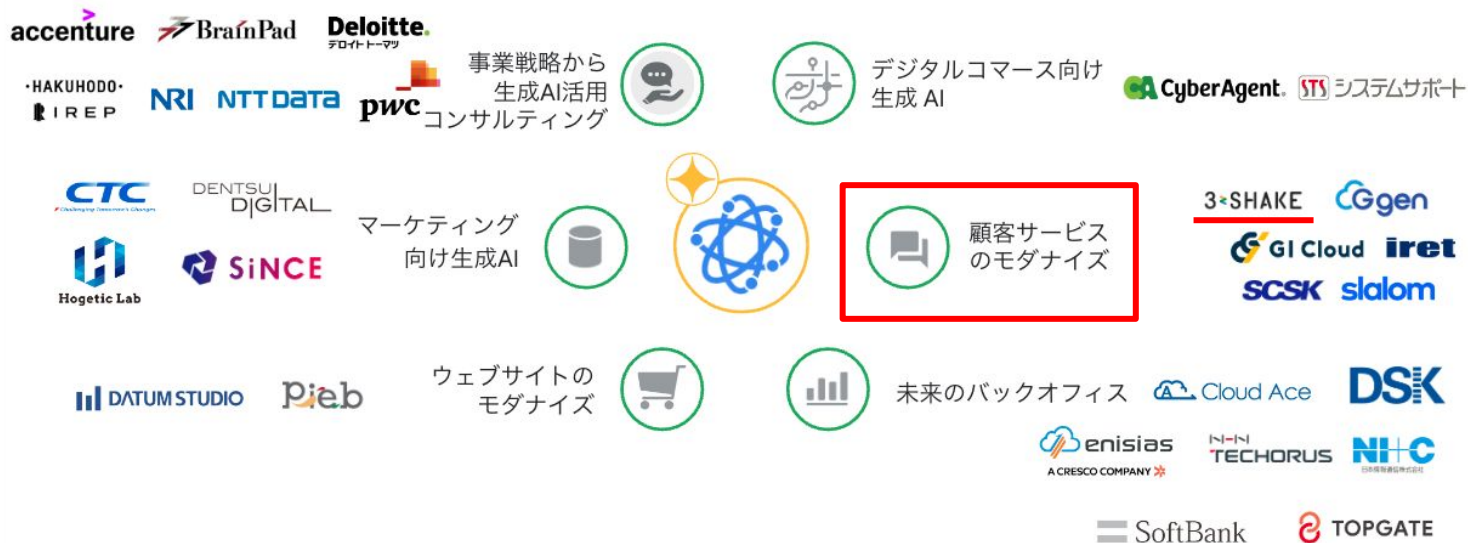
Googleクラウド・AWSの両方のエンジニアリングに強みを持つ
(2024年8月に国内2例目の、GoogleCloudのDevOpsスペシャライゼーションを取得)





2024 March

Google Cloud 生成 AI ソリューション ユースケース別支援パートナー



備考) パートナーロゴはアルファベット順に配置

Google Cloud

1. AI Agent の概要
 - a. Google Cloud での AI Agent
2. Agent Development Kit について
3. Agent Development Kit を使った AI Agent 開発運用
 - a. 構築、デプロイ、テスト、可観測性
4. まとめ

AI Agent とはなにか
LLM との違いを
知りたい人

AI Agent を
開発する人
開発したい人

Google Cloud での
AI Agent 作成
について知りたい人

質問事項は Ask the speakerで

01

AI Agent とは

Google Cloud

AI を使用してユーザーの代わりに目標を追求し、タスクを完了させるソフトウェア システムです

Anthropic

LLM が独自のプロセスとツールの使用を動的に指示し、タスクの達成方法を制御するシステムです

AWS

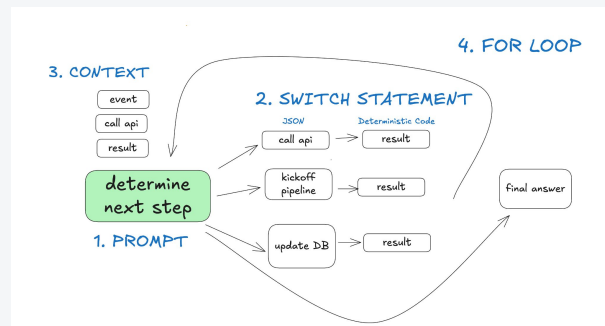
環境と対話し、データを収集し、そのデータを使用して

自己決定タスクを実行して、事前に決められた目標を達成するためのソフトウェアプログラムです

<https://cloud.google.com/discover/what-are-ai-agents?hl=ja>
<https://www.anthropic.com/engineering/building-effective-agents>
<https://aws.amazon.com/jp/what-is/ai-agents/>

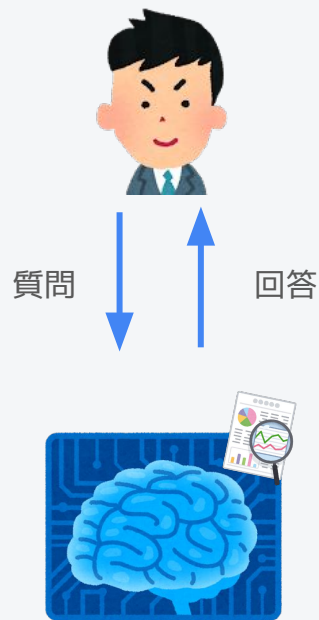
12 Factor Agents

- **Prompt**
 - LLMの動作方法と利用可能な「ツール」を指示します
- **switch ステートメント**
 - LLM が返す JSON に基づいて、それをどう処理するかを決定します
- **蓄積されたコンテキスト**
 - 発生したステップとその結果のリストを保存する
- **for ループ**
 - LLM が何らかの「ターミナル」ツール呼び出し (またはプレーンテキスト応答) を発行するまで、switch ステートメントの結果をコンテキスト ウィンドウに追加し LLM に次のステップを選択するように要求します



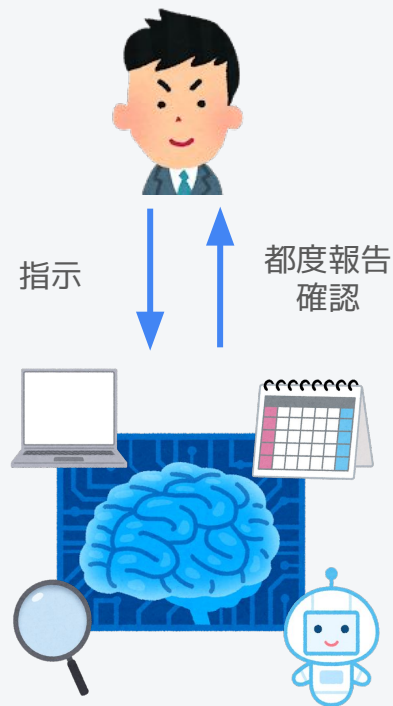
	Bot	AI Assistant	AI Agent
目的	単純なタスクや会話を自動化	ユーザーの作業をサポート	自律的にプロアクティブにタスクを実行
機能	事前定義されたルールに従う 学習は限定的 基本的なやり取り	リクエストやプロンプトに応答し、情報を提供して簡単なタスクを完了する アクションを推奨できるが、決定はユーザーが行う	複雑な複数ステップのアクションを実行できる、 学習して適応する、 独立して意思決定できる
インタラクション	受動的 トリガーやコマンドに応答する	リアクション型 ユーザーのリクエストに応答する	予防型、目標指向

Bot



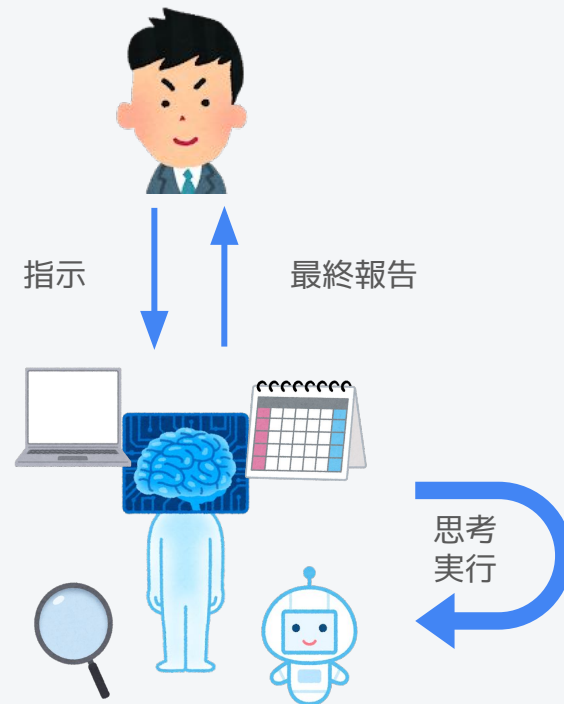
物知りな対話相手

AI Assistant



ツール利用の補助

AI Agent



自律的に動く執事

Web or Slack Chat実装



Chat 形式での AI Agent の開発
Agent Development Kit / Mastra.ai

Agent 作成調査

Agent Development KitとAgent Engineを使ってVertex AI Agent Builderに入門してみる

Keita Sato
2025.7.17



Agent Development Kit を使った Agent Engine
Agent 開発調査

ノーコード

ex. **Agentspace**

ローコード

ex. **Conversational
Agents**

フルコーディング

ex. **Agent Engine**

構築難易度低

構築難易度高

Pros

- 非エンジニアでも構築が簡単である。メンテナンスコストも低い

Cons

- コネクタ、ツールにないものは使えない
- Multi Agent などの複雑なフローは作れない

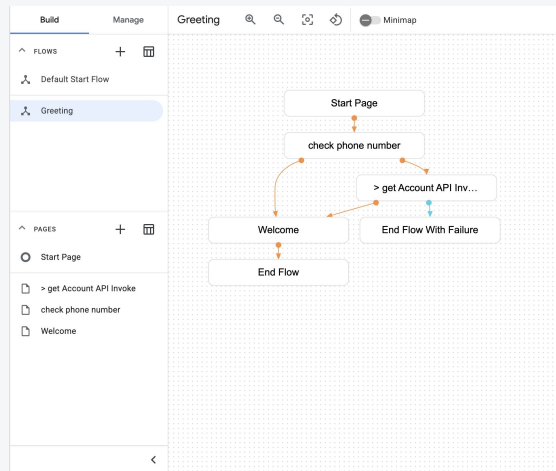
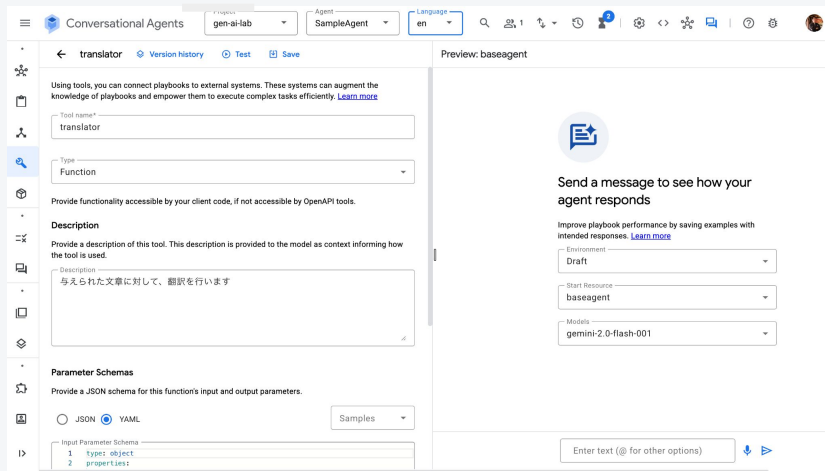


Pros

- ノーコードよりできることは多い、外部との接続は比較的簡単
- Agentspace から呼び出す事もできる

Cons

- Conversational Agents 独自の設定や流儀などを覚える必要がある(クセがある)



Pros

- 柔軟性が高い
- 運用面でのサポートが厚い。Session, O11y の設定がデフォルトでついている

Cons

- 実装が必要である。保守を自分たちで実施する

エージェント エンジン

更新

この Cloud プロジェクト内のエージェント インスタンスを表示、管理します。

エージェントを開発

デプロイガイド

リージョン

us-central1 (アイオワ)

フィルタ

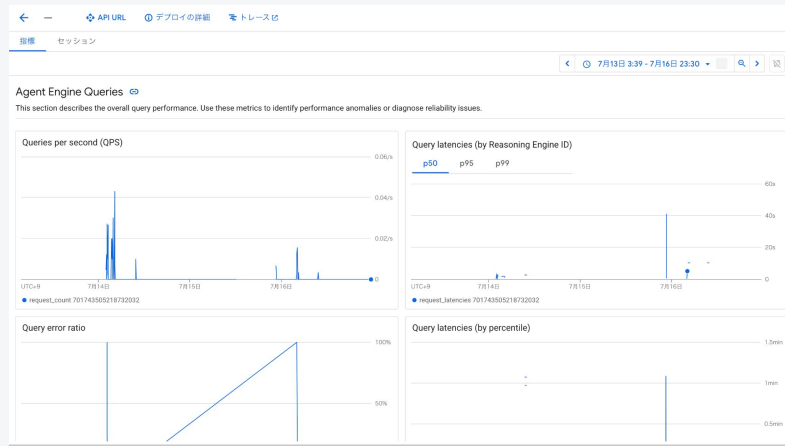
名前作成フレームワーク説明

二	2025/07/14 1:53:37	google-adk	⋮
二	2025/06/22 13:47:57	google-adk	⋮
二	2025/06/22 13:45:14	google-adk	⋮

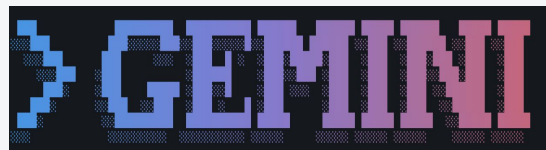
Vertex AI Agent Engine を試してみる

Vertex AI と CI / CD の統合のためのエージェント コードと構成のパッケージングを合理化します。

スターターパックを入手

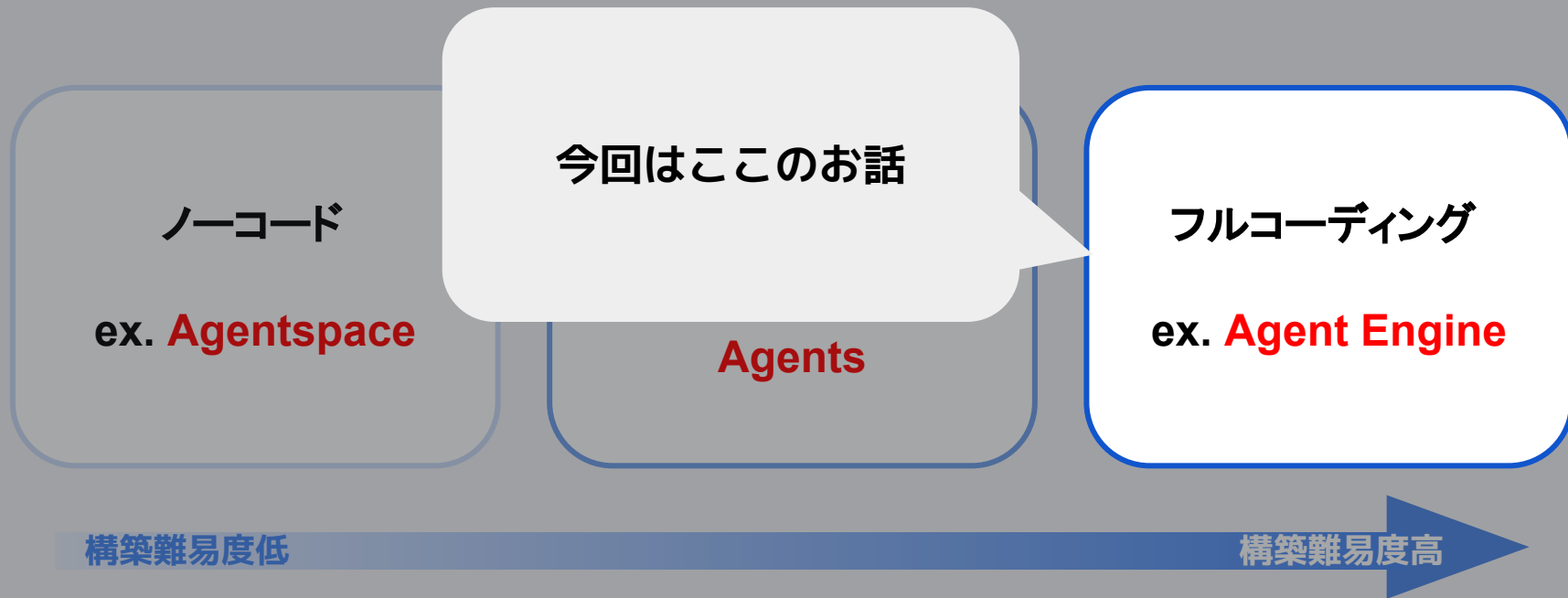


- 仕様などのコンテキストがそのフレームワーク内で閉じている
ローコードを使った開発は少なくなる
- プロンプトのみで非エンジニアも使えるという意味で
ノーコードはまだまだ需要が大きい
- コーディング用 AI Agent を利用し、デプロイまでできるため
フルコードは、ハードルが下がる



社内開発で Vibe Coding → Agent Engine への deploy が多くなる

観点	Google Cloud Agent Engine - Agentspace	Microsoft Azure AI Foundry - Copilot Studio	Amazon Web Services Amazon Bedrock Agent
構築容易性	中 ノーコード UI とプロコードSDK の両方を提供し、幅広い開発者層に対応 ・記憶領域も内部で実装済み ・VPC Service Controls や IAM, CMEK	中 プロ開発者向けの「Foundry」とビジネスユーザー向けの「Copilot Studio」で明確に分離 ・外部に記憶領域を配置 ・VNet、Microsoft Entra ID, コンテンツフィルター	高 自然言語による指示で迅速に構築可能 AWS サービスとの連携が前提 ・外部に記憶領域を配置 ・IAM, VPC, Guardrails
金額感・コスト	実行環境のコンピュータ課金とクエリ課金(Gemini利用率)	Agent Service自体は無料 利用したAzureサービスの料金	Agents 機能自体は無料 基盤となる Bedrock モデルと Lambda などのサービス利用料が発生
扱える言語・フレームワーク	ADK や主要な OSS フレームワーク(LangChain 等)をサポート	主に独自 SDK(Azure AI SDK)と Semantic Kernel LangChain 等も利用可能	主に Lambda 関数、その他の AWS のサービスとの連携を利用する



02

フルコーディング

Agent Development Kit について

AI Agent を作るためのフレームワーク

Google Cloud Next 25 にて発表

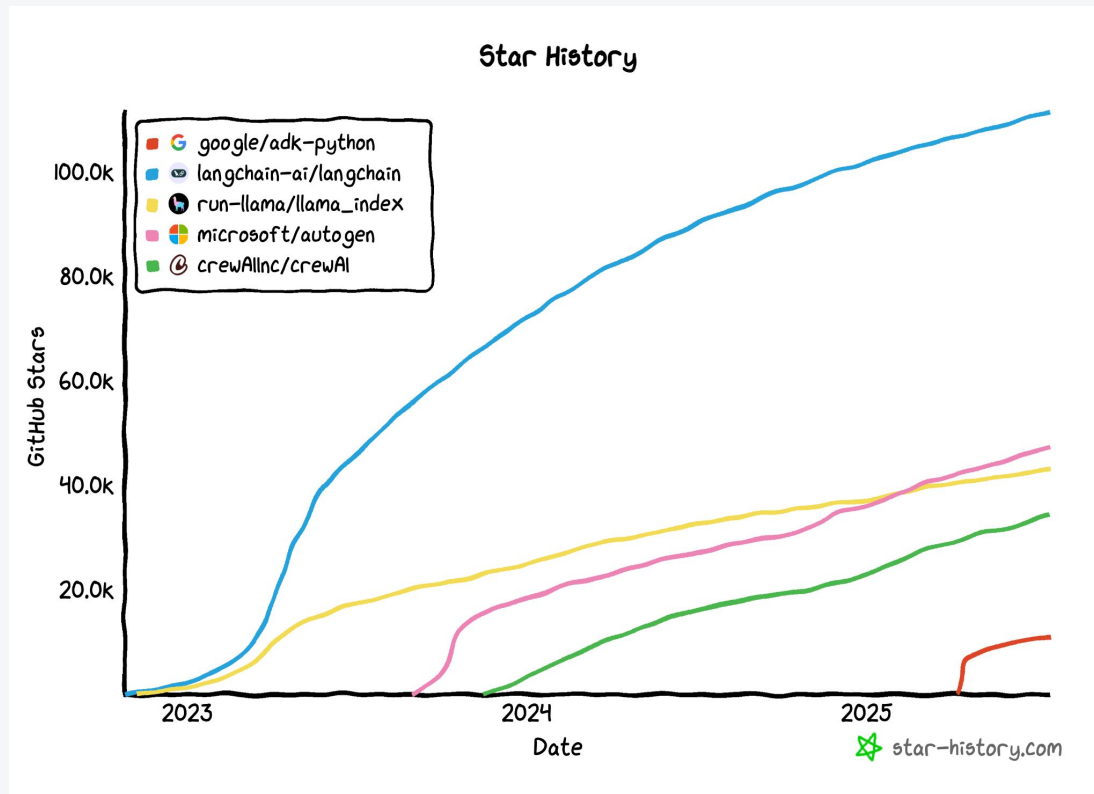
- Python 1.7.0 (<https://github.com/google/adk-python>)
- Java 0.2.0 (<https://github.com/google/adk-java>)



Agent Development Kit

- LiteLLM を使った別モデルの利用
- Session などの interface の提供
- Sequence や Loop など Workflow 的な Agent の構築も可能
- MCP との連携
- Agent Engine でのリリースが容易
- Enterprise Ready

比較項目	Google Agent Development Kit	LangChain / LangGraph	LlamaIndex	Microsoft AutoGen
主要ユースケース	エンタープライズ プロセス自動化	汎用/複雑なワークフロー	データ拡張型Q&A	研究/動的な問題解決
エコシステムと統合	Google Cloudに最適化 /成長中	最大/最も成熟	RAGに特化/LlamaHub	Microsoft支援/研究センター
本番環境への準備度	高	高	高	中
マルチエージェント手法	階層的委任 & ワークフロー	エージェントの明示的な グラフ	エージェント的 クエリルーター	創発的なグループチャット
主要な強み	制御と信頼性	柔軟性	RAG のパフォーマンス	動的な コラボレーション
主要な弱み	振興とユーザ数	急な学習曲線	ニッチな焦点	GA ではない



	Agent Engine	Cloud Run Functions	Cloud Run Service	GKE
構築容易性	◎	○	○	×
運用容易性	◎	△	△	×
マルチエージェント構成	○	△ *2	△	○
Agent2Agent 構成	×	△ *2	○	○

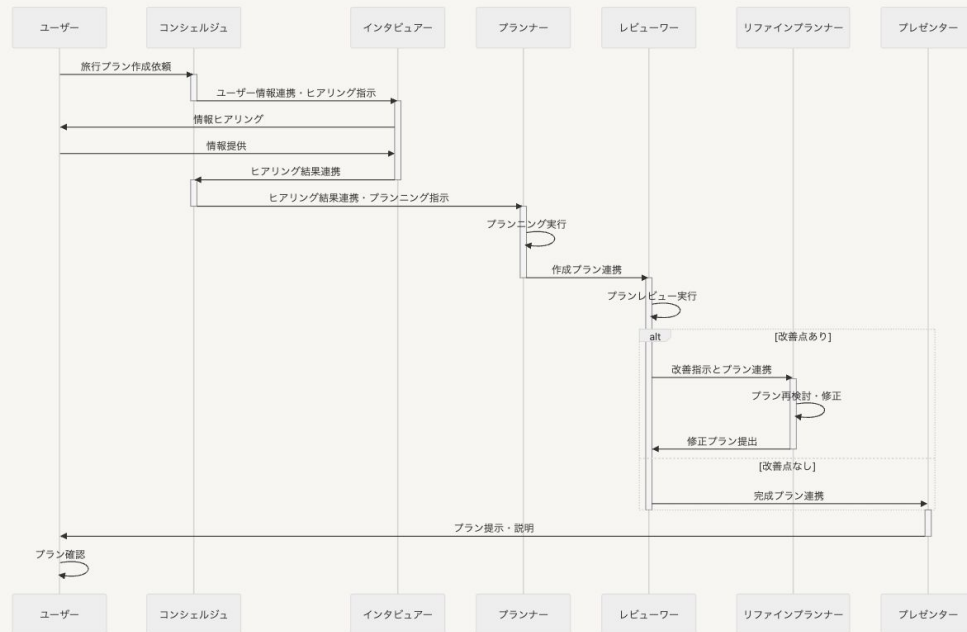
*1. HOST からの接続時にProtocolが限られている

*2. 実行時間が限られている

03

Agent Development Kit での開発運用

今回の AI Agent 想定



```
from constraints import MODEL_GEMINI, PLANNER_OUTPUT_KEY
from google.adk.agents import Agent
from google.adk.tools.google_search_tool import google_search
```

```
root_agent = Agent(
    name="planner",
    model=AGENT_MODEL,
    description="ユーザの旅行計画を手伝うエージェントです。 ",
    instruction="""
```

あなたは旅行計画のアシスタントです。指定された情報からユーザの旅行計画を **1つ**作成してください。

■ユーザから得られた情報 :

```
-----
{{interviewer_output}}
-----
```

```
""",
    tools=[google_search],
    output_key=PLANNER_OUTPUT_KEY,
)
```

Agent または
LLM Agent class を使って
AI Agent を定義

AI Agent の振る舞い(description)と指示
(instruction)の定義
他 Agent での output を変数として埋め込み
が可能

```
root_agent = Agent(  
    name="planner",  
    model=AGENT_MODEL,  
    description="ユーザの旅行計画を手伝うエージェントです。 ",  
    instruction="""
```

あなたは旅行計画のアシスタントです。指定された情報からユーザの旅行計画を **1つ**作成してください。

■ユーザから得られた情報 :

```
{{interviewer_output}}
```

```
""",  
    tools=[google_search],  
    output_key=PLANNER_OUTPUT_KEY,  
)
```

利用するツールを定義
Google 提供のツールも存在

```
loop_agent = LoopAgent(  
    name="loop_planer",  
    sub_agents=[  
        reviewer_agent,  
        refine_planner  
    ],  
    max_iterations=3  
)  
sequential_agent = SequentialAgent(  
    name="travel_planer",  
    sub_agents=[  
        planner_agent,  
        loop_agent,  
        presenter_agent  
    ],  
)
```

繰り返し実行する
または、直列に実行するようにワークフローを定義

```
loop_agent = LoopAgent(  
    name="loop_planer",  
    sub_agents=[  
        reviewer_agent,  
        refine_planner  
    ],  
    max_iterations=3  
)  
sequential_agent = SequentialAgent(  
    name="travel_planer",  
    sub_agents=[  
        planner_agent,  
        loop_agent,  
        presenter_agent  
    ],  
)
```

最大のループ回数を指定する

Agent Engine に Deploy

1. Google-cloud-aiplatform[adk,agent_engines]
をインストール
2. コードで記載する

```
from agents.agent import root_agent
from vertexai.preview import
reasoning_engines
app = reasoning_engines.AdkApp(
    agent=root_agent,
    enable_tracing=True,
)
```

コンテナとして Deploy

1. Runner の定義
2. Docker File の作成

```
def create_agent_runner() -> tuple[Runner,
BaseSessionService]:
    session_service = InMemorySessionService()
    return (Runner(
        agent=root_agent,
        app_name=APP_NAME,
        session_service=session_service,
    ), session_service)
```

Tips

**Google 提供の Tools が Root Agent かつ 一つでないと利用できないため
Agent を Tool として利用する**

- Google Search の場合は規約として検索結果も出力する必要があるので工夫が必要

**ローカル開発, ADK web, Cloud Run, Agent Engine でも動かせるようにするために
__init__.py を使い Root Agent を閲覧できるようにする**

- {Agent name}
 - __init__.py
 - agent.py

成功の定義

Define Success

エージェントにとっての成功とは
どのような結果でしょうか？

重要なタスクの特定

Identify Critical Tasks

エージェントが実行する必要がある
重要なタスクは何ですか？

関連指標の選択

Choose Relevant Metrics

パフォーマンスを測定するために
どのような指標を追跡しますか？

軌道とツールの使用の評価

Evaluating Trajectory and Tool Use

ツール、戦略の選択、アプローチの効率など、エージェントが解決策に到達するまでに実行する手順を分析します

最終応答の評価

Evaluating the Final Response

エージェントの最終出力の品質、関連性、正確性を評価します

1. JSON の作成

a. テストデータ

- i. Input Query
- ii. 呼び出し期待される tool
 - 1. 引数

b. 期待結果スコア

- i. Tool 利用の順番などの一致度
- ii. Agent の Response の一致度

2. Pytest の作成

AgentEvaluator.evaluate

```
[ {  
  "query": "xxxx",  
  "expected_tool_use": [  
    {  
      "tool_name": "transfer_to_agent",  
      "tool_input": {  
        "agent_name": "interviewer"  
      }  
    }  
  ],  
  "expected_intermediate_agent_responses": [],  
  "reference": "xxxx"  
},  
]
```

```
{  
  "criteria": {  
    "tool_trajectory_avg_score": 0.5,  
    "response_match_score": 0.8  
  }  
}
```

ADK web でのテスト

- テストデータの JSON を配置することで
実施可能

The screenshot displays the ADK web interface. On the left, the 'agents' dropdown is set to 'agents'. Below it, the 'evalset252727' is selected, and the 'Run Evaluation' button is visible. A table shows the evaluation results for 'case0b7dd5', which is marked as 'Fail'. A modal titled 'EVALUATION METRIC' is open in the foreground, showing two sliders: 'Tool trajectory avg score' set to 1 and 'Response match score' set to 0.7. The modal also includes 'Cancel' and 'Start' buttons.

Agent Development Kit

agents

Trace Events State Artifacts Sessions Eval

< evalset252727

Run Evaluation

Case ID	Result
case0b7dd5	Fail

+ Add current session to evalset252727

EVAL CASE ID case0b7dd5

こんにちは旅行に行きたいです。相談に乗ってください。ドイツには行きたいです。高すぎない

transfer_to_agent

transfer_to_agent

旅行の目的はヨーロッパの街並みの散策とこのことですか。美術館巡り、グルメ、ショッピングなど、興味のあ

EVALUATION METRIC

Tool trajectory avg score: 1

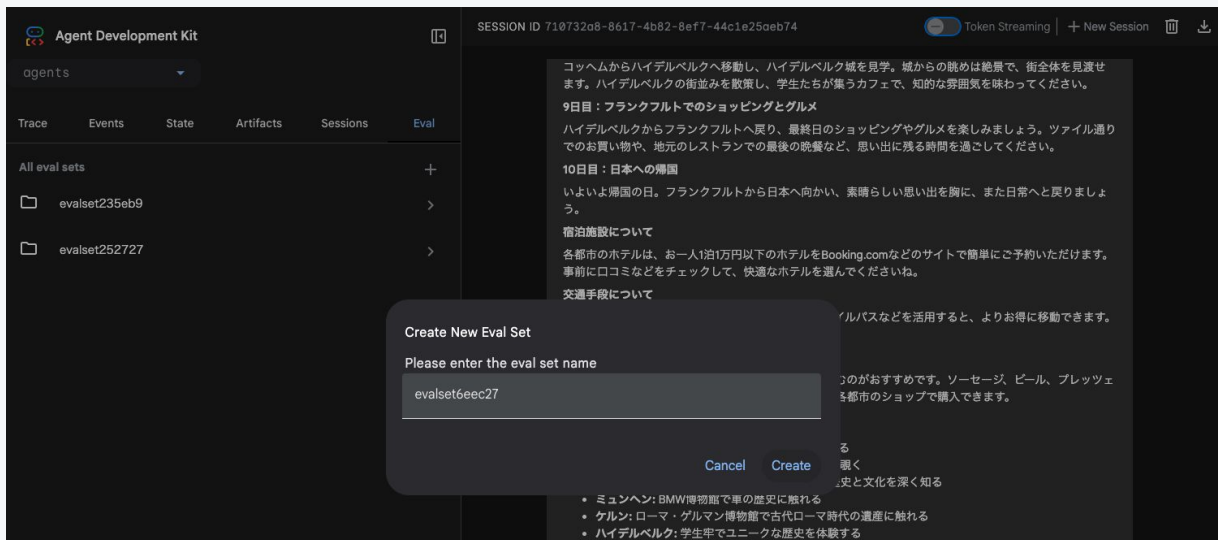
Response match score: 0.7

Cancel Start

Tips

ADK web で Agent のフローを json として出力できるのでそれをテストデータとする

- 保存しておき、デグレのチェック

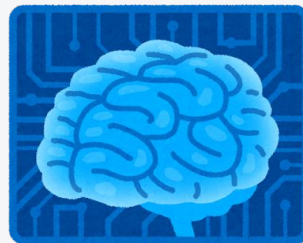


UX 起因

- レスポンスの時間
- AI Agent、Tool の利用順、引数

SLI/SLO

- トータルコスト、トークン数
- エラー率



ADK web での Event の可視化

google-adk[eval]をインストール

agents

SESSION ID fbb69204-c22a-4573-965e-9f77ef37a795

Events State Artifacts Sessions Eval

Conversations with agent

- model: functionCall: 0: transfer_to_agent
- user: functionResponse: 1: transfer_to_agent
- model: 2旅行の目的はヨーロッパの街並みの散策とのことですが、
- model: 3ありがとうございます。お城に興味があるのですか、
- model: 4承知いたしました。他に旅行の予定や同行者について...
- model: 5ありがとうございます。予算と人数について承知...
- model: functionCall: 6: transfer_to_agent
- user: functionResponse: 7: transfer_to_agent
- model: 8承知いたしました。ドイツを中心に、お城巡りと...
- model: 9旅行プランをレビューします。"良い点:" "希望..."
- model: 10承知いたしました。レビュー内容と提案結果に...
- model: 11旅行プランを再度レビューします。"改善点:" "...
- model: functionCall: 12: exit_loop
- user: functionResponse: 13: exit_loop
- model: 14はい、承知いたしました。お客様のために、ド...

Type a Message...

Event 10 of 15

Event Request Response

travel_concierge interviewer travel_planner loop_planner presenter planner refine_planner reviewer google_search researcher exit_loop

content:

```

{
  parts:
    0: {
      text: "旅行プランをレビューします。 **良い点** **希望に合致** ドイツのお城巡りと街並み散策という希望に沿ったプランである。 **具体的な都市名** フランクフルト、ローテンブルク、ニュルンベルク、ミュンヘン、ケルンといった具体的な都市名が挙げられており、イメージしやすい。 **ホテルの提案** 各都市のホテルが提案されており、予算内で実現可能である。 **移動手段の提案** ドイツ国内の移動手段として鉄道が提案されており、ジャーマンレイルパスの活用も示されている。 **その他** City Cardの利用やノイシュヴァンシュタイン城の事前予約など、有益な情報が含まれている。 **予算内訳** 予算内訳が示されており、費用の目安が分かりやすい。 **改善点** **移動時間の考慮** 各都市間の移動時間が考慮されているが、鉄道の遅延なども考慮して、もう少し余裕を持ったスケジュールにすると、より安心して旅行できる。 **観光地の詳細** 各都市の観光地の詳細な情報が不足している。例えば、ローテンブルクであれば市街のどの部分まで歩くのか、聖ツプス教会の見どころは何かなど、具体的な情報があると、より旅行が楽しくなる。 **食事の提案** 食事に際する提案が少ない。各都市の各名料理やおすすめのレストランなどを紹介すると、旅行の満足度が向上する。 **ノイシュヴァンシュタイン城** ミュンヘンから日帰りとなっていますが、移動時間などを考慮すると、ノイシュヴァンシュタイン城の観光に十分な時間を確保できない可能性があります。 **点数** ** 85点 **理由: ** 全体的に良くまとまっているが、観光地の詳細情報や食事の提案が不足している点、ノイシュヴァンシュタイン城の日帰り観光に無理がある可能性がある点が減点対象。"
    }
  role: "model"
  invocation_id: "e-77c84312-7b49-461c-8847-7ced4dc8ba3a"
  author: "reviewer"
  actions:
    {
      state_delta: {
        interviewer_output: "旅行プランをレビューします。 **良い点** **希望に合致** ドイツのお城巡りと街並み散策という希望に沿ったプランである。 **具体的な都市名** フランクフルト、ローテンブルク、ニュルンベルク、ミュンヘン、ケルンといった具体的な都市名が挙げられており、イメージしやすい。 **ホテルの提案** 各都市のホテルが提案されており、予算内で実現可能である。 **移動手段の提案** ドイツ国内の移動手段として鉄道が提案されており、ジャーマンレイルパスの活用も示されている。 **その他** City Cardの利用やノイシュヴァンシュタイン城の事前予約など、有益な情報が含まれている。 **予算内訳** 予算内訳が示されており、費用の目安が分かりやすい。 **改善点** **移動時間の考慮** 各都市間の移動時間が考慮されているが、鉄道の遅延なども考慮して、もう少し余裕を持ったスケジュールにすると、より安心して旅行できる。 **観光地の詳細** 各都市の観光地の詳細な情報が不足している。例えば、ローテンブルクであれば市街のどの部分まで歩くのか、聖ツプス教会の見どころは何かなど、具体的な情報があると、より旅行が楽しくなる。 **食事の提案** 食事に際する提案が少ない。各都市の各名料理やおすすめのレストランなどを紹介すると、旅行の満足度が向上する。 **ノイシュヴァンシュタイン城** ミュンヘンから日帰りとなっていますが、移動時間などを考慮すると、ノイシュヴァンシュタイン城の観光に十分な時間を確保できない可能性があります。 **点数** ** 85点 **理由: ** 全体的に良くまとまっているが、観光地の詳細情報や食事の提案が不足している点、ノイシュヴァンシュタイン城の日帰り観光に無理がある可能性がある点が減点対象。"
      }
    }
  artifact_delta: {
    requested_auth_configs: {
      "keychain":
    }
  }
}

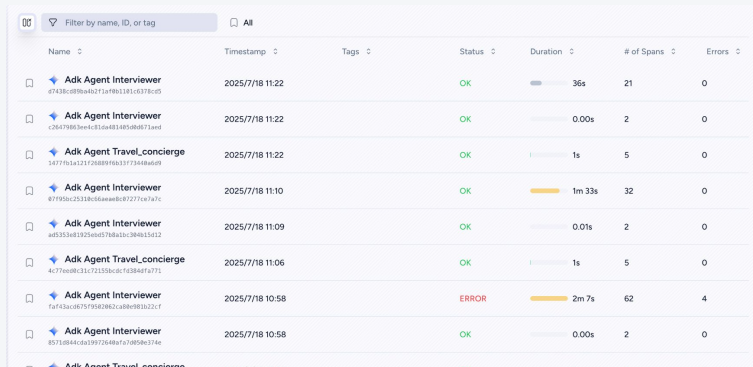
```

AgentOps

1. 環境変数の設定と、ライブラリのimport
 - a. 自動で既存の otel の tracer を NoOpTracer にする
2. エージェントの run_async 実行
 - a. 自動で非同期に送信される

```
AGENTOPS_API_KEY=xxxxxx
```

```
import agentops
agentops.init()
```



The screenshot shows the AgentOps observability dashboard. It features a table with columns for Name, Timestamp, Tags, Status, Duration, # of Spans, and Errors. The table lists several traces, including 'Adk Agent Interviewer' and 'Adk Agent Travel_concierge'. The 'Adk Agent Interviewer' traces show a duration of 36s and 0.00s, while the 'Adk Agent Travel_concierge' traces show a duration of 1s. The 'Adk Agent Interviewer' traces have a status of 'OK', while the 'Adk Agent Travel_concierge' traces have a status of 'ERROR'.

Name	Timestamp	Tags	Status	Duration	# of Spans	Errors
Adk Agent Interviewer	2025/7/18 11:22		OK	36s	21	0
Adk Agent Interviewer	2025/7/18 11:22		OK	0.00s	2	0
Adk Agent Travel_concierge	2025/7/18 11:22		OK	1s	5	0
Adk Agent Interviewer	2025/7/18 11:10		OK	1m 33s	32	0
Adk Agent Interviewer	2025/7/18 11:09		OK	0.01s	2	0
Adk Agent Travel_concierge	2025/7/18 11:06		OK	1s	5	0
Adk Agent Interviewer	2025/7/18 10:58		ERROR	2m 7s	62	4
Adk Agent Interviewer	2025/7/18 10:58		OK	0.00s	2	0

Tips

ツールとして自作関数を使うと

'function' object has no attribute 'name'

と言われて agent が動かないので

FunctionTool(exit_loop) としてラップ

The screenshot displays the ADK interface with a workflow view on the left and LLM details on the right. The workflow shows a sequence of steps: 'travel_concierge' (Agent), 'travel_planner' (Tool), 'transfer_to_agent' (Tool), 'interviewer' (Agent), and 'reviewer' (Agent). A tooltip for the 'travel_planner' tool is visible, showing its configuration and usage.

default
07b9c1f8410fa3c69e2c75482e8f582cc • 2025/7/18 12:38

gcp.vertex.agent • gemini-2.0-flash • Duration: 38.39s • Total Cost: \$0.0038828 • LLM Calls: 21 • Tool Calls: 7 • Errors: 0 • Total Tokens: 24,734

Waterfall View • Agents • Terminal Logs

Agent • Llm • Tool • Workflow • Session

travel_concierge
travel_planner is 旅行計画を作成するためのエージェントです..
transfer_to_agent
interviewer
海外旅行のご希望ありがとうございます！いくつか確認させてください。*..
travel_planner
旅行計画の作成が完了しました。旅行の準備ができていますので、旅行プランの作成に進みます..
transfer_to_agent
adk.llm.gemini-2.0-flash... agentops
travel_planner
Timings:
Duration: 43087.216ms
Start - End: 12:38:39 - 12:39:22
Usage:
Prompt tokens: 654
Completion tokens: 29
Total tokens: 683
ETC:
ID: 766a3a35..
Type: agent

Show Bar Text
Show tooltip

LLM: gemini-2.0-flash

Time: 12:38:35 - 12:38:39	Cost	Tokens
Duration: 3.93s	Prompt: \$0.0000451	Prompt: 451
Model: gemini-2.0-flash	Completion: \$0.0000040	Completion: 10
	Total: \$0.0000491	Total: 461

Prettify • Raw JSON

travel_planner は 旅行計画を作成するためのエージェントです。interviewer は ユーザから旅行計画に必要な情報を聞き出すエージェントです。

あなたは旅行計画のアシスタントの窓口です。ユーザの旅行計画を手伝ってください。ユーザの旅行計画を手伝うために、interviewer を使用してユーザから情報を聞き出してください。その後 travel_planner を使用して旅行計画を作成し、ユーザに伝えてください。

旅行計画以外のことを質問された場合は、「申し訳ありません。それ以外はおの方に聞いて下さい」と答えてください。

You are an agent. Your internal name is "travel_concierge".

The description about you is "ユーザに対して旅行計画を手伝うエージェントです。"

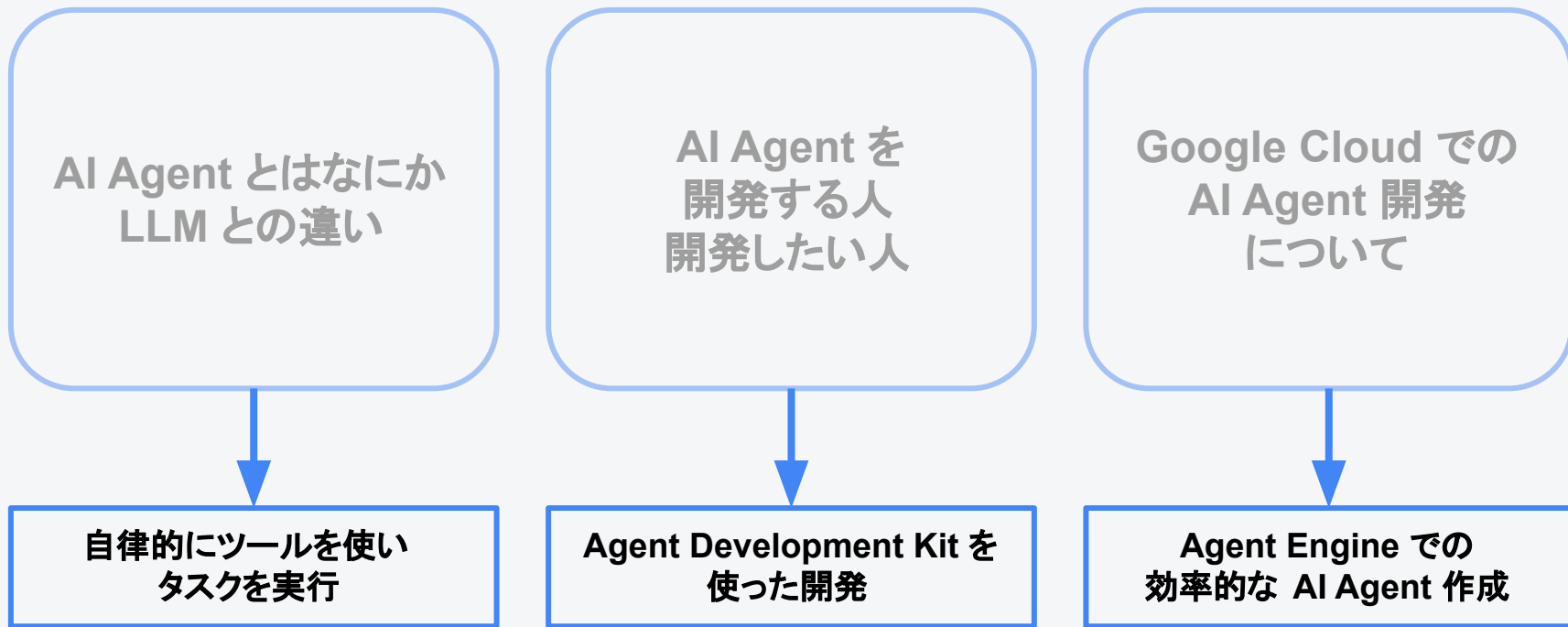
04

まとめ

AI Agent とはなにか
LLM との違い

AI Agent を
開発する人
開発したい人

Google Cloud での
AI Agent 開発
について



ご清聴ありがとうございました