

BizCopilot Powered By ChatGPT  **BizCopilot**

株式会社ディープコム

Copyright © 2023 DeepCom Co., Ltd.

全体概要

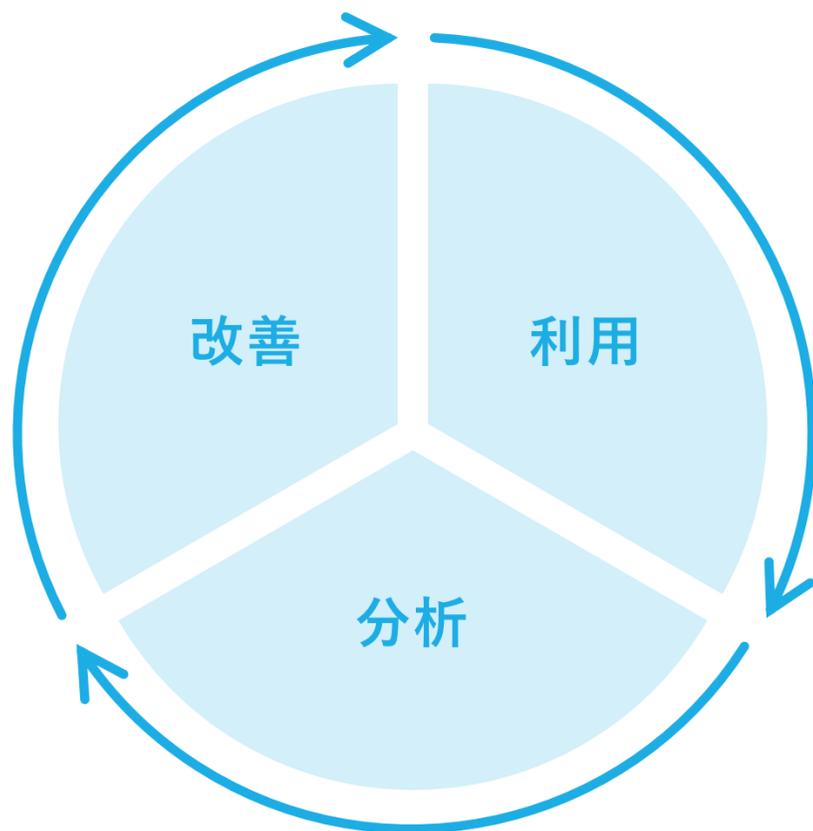
- STEP1 : Azure OpenAI Service を用いた企業利用向けのセキュアなチャットボットアプリケーション PoC
- STEP2 : Azure Cognitive Service を用いた社内情報検索
- STEP3 : 要望に応じて特定業務へ特化

ソリューションLP : https://deepcom.co.jp/enterprise_chatgpt/

サービスLP : https://deepcom.co.jp/bizcopilot_services/

BizCopilot Powered By ChatGPT

生成AI 活用サイクル



①利用

セキュアな環境で
最先端のAIテクノロジーを
従業員に付与することで
業務生産性を向上

②分析

利用状況を分析し、
利用価値を把握
より効率的なプロンプトを検討

③改善

利用状況の分析結果から
日々の利用からフィードバックを
得て更なる業務生産性向上

ソリューション

STEP
01

- セキュアな環境構築
- PoC

STEP
02

- 実運用向け開発
(STEP1の実用化)

STEP
03

- ログ分析
利用価値把握

STEP
04

- 業務プロセス適用
- 社内データ活用

STEP
05

- 特定高度業務向け検討

本アーキテクチャ範囲

アプリUI

The image displays the BizCopilot application interface. On the left is a dark sidebar with a 'BizCopilot' logo and a '+ New Chat' button. Below it are several chat session thumbnails, with the first one selected and highlighted in blue. The main area shows a chat session titled 'Current Chat Session'. At the top, it says 'You' and 'C#でFizzBuzzをお願いします。' with a timestamp of '9/6/2023 1:17:20 PM'. Below this, the BizCopilot assistant responds: 'もちろんです。以下にC#でのFizzBuzzの実装例を示します。' and provides a code block for C#. The code is as follows:

```
csharp
using System;

class Program
{
    static void Main(string[] args)
    {
        for (int i = 1; i <= 100; i++)
        {
            if (i % 3 == 0 && i % 5 == 0)
            {
                Console.WriteLine("FizzBuzz");
            }
            else if (i % 3 == 0)
            {
                Console.WriteLine("Fizz");
            }
            else if (i % 5 == 0)
            {
                Console.WriteLine("Buzz");
            }
            else
            {
                Console.WriteLine(i);
            }
        }
    }
}
```

At the bottom of the chat area, there are buttons for 'Send', 'Stop', 'JA⇔EN', and 'Share'. The interface also shows 'Plugins: OrganizationData' and 'Model: gpt-35-turbo-16k'. On the right side, there is a zoomed-in view of the chat history, showing the previous messages and the assistant's response in Python:

```
python
for i in range(1, 101):
    if i % 3 == 0 and i % 5 == 0:
        print("FizzBuzz")
    elif i % 3 == 0:
        print("Fizz")
    elif i % 5 == 0:
        print("Buzz")
    else:
        print(i)
```

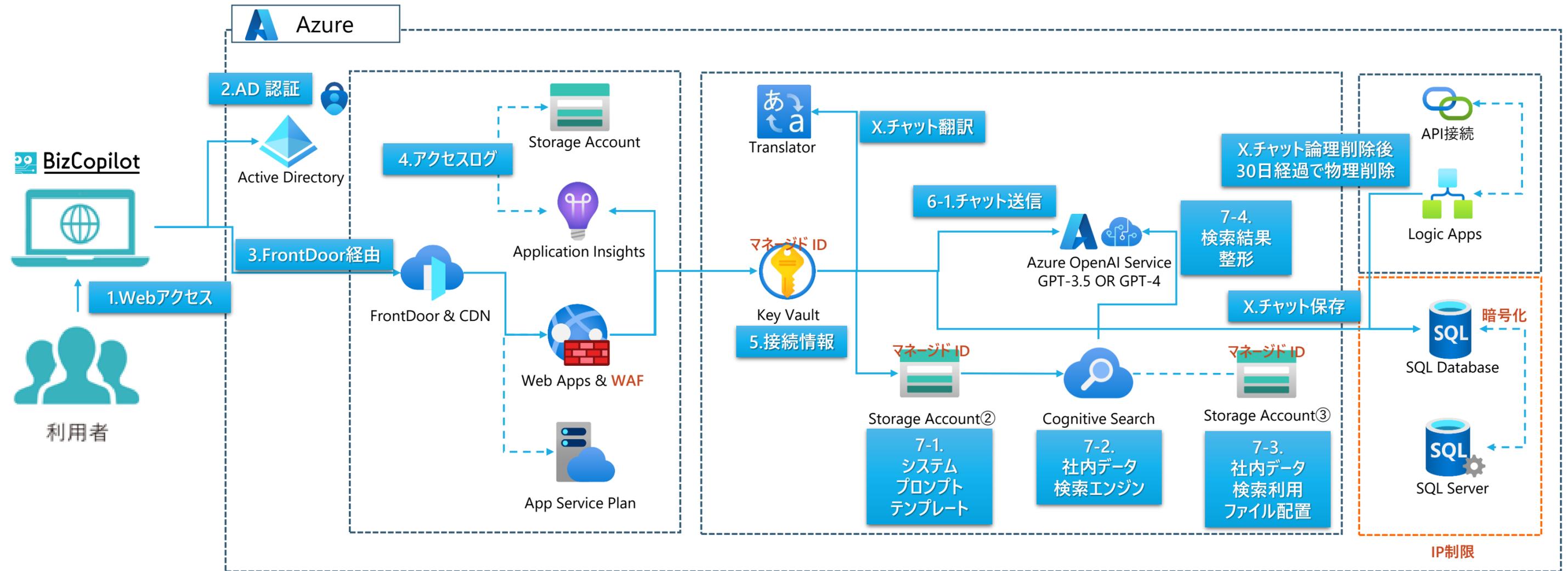
The zoomed-in view also shows the 'BizCopilot' logo, the user's message, the assistant's response, and a 'Copy code' button. At the bottom of the zoomed-in view, there are buttons for '送信', '停止', and 'JA⇔EN'.

アプリUI

■ 社内データ検索
「OrganizationData」ボタン選択時は、自身や自社が所有する情報を用いた回答が返却可能。

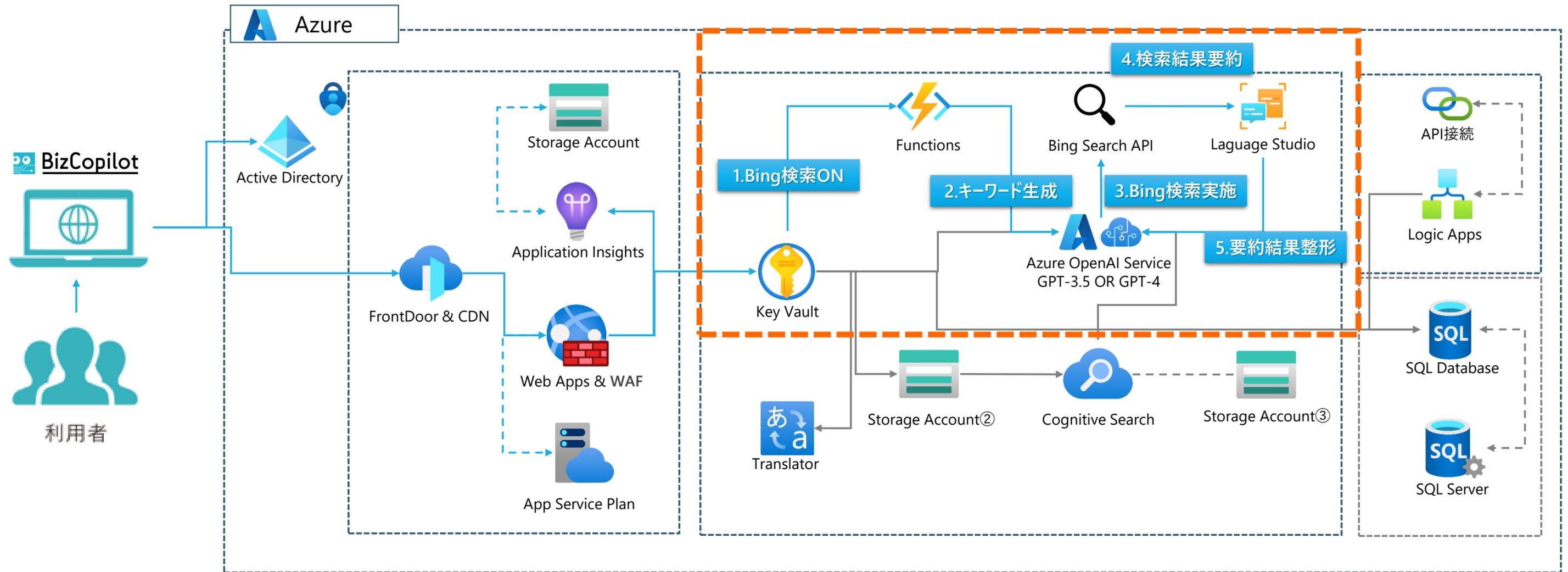
■ モデル切り替え
各社デプロイ済のモデルを指定して利用
※利用可能なモデルは下記
【gpt-35-turbo/ gpt-35-turbo-16K】
【gpt-4/ gpt-4-32K】

アーキテクチャ図



- X.チャット翻訳：画面上で翻訳ボタン押した場合のみ
- 6-1.~：社内データ検索しない場合のルート
- 7-1.~：社内データ検索をする場合のルート 検索結果は一度モデルを通し、ユーザーライクな形式で返却
- X.チャット保存：社内データ検索有無に関わらず暗号化のうえ保存
- X.チャット論理削除後...：スケジュール起動
- 文字色オレンジ：セキュリティ関連の対応

アーキテクチャ図：オプション「Bing検索機能」



オプションとしての位置づけで、「Bing検索機能」を追加した場合のアーキテクチャのご紹介。

※オプション関連以外のアーキテクチャは前述のアーキテクチャ同様の為、説明を割愛。

- 1. チャット送信時、Bing検索ON（画面でチェック☑）の場合「Functions」で処理を統括
- 2. チャット内容から、「Azure Open AI」にて検索キーワード生成
- 3. 検索キーワードを元に、「Bing Search API」にてインターネット検索処理を実施
- 4. 検索結果を、トークン数節約の為「Language Studio」にて要約
- 5. 要約結果を「Azure Open AI」にて整形しユーザーへ返却

考慮事項

信頼性(可用性)

- Azureのサービスでは「可用性ゾーン」や「リージョン」といった単位で可用性を設計しており、これらを適切に組み合わせることで、ビジネス クリティカルなワークロードの信頼性を実現するように設計することが可能です。詳細は「[Azure リージョンと可用性ゾーンとは](#)」をご参照ください。
- このシナリオで用いられているApp Service、Azure OpenAI Service、Azure Cache for Redis (Enterprise) 等のコンポーネントはそれぞれゾーン冗長、リージョン冗長、geoレプリケーションなど高可用性のオプションや構成を利用可能です。
必要となる可用性に応じて導入を検討してください。
複数リージョン間/複数ゾーン間でAct-Act構成を取る場合にはAzure Front Door、Azure Application Gatewayなどの利用をご推奨します。

信頼性(回復性)

- アプリケーションの正常性を監視するために、Application Insights を使用すると、カスタマー エクスペリエンスや可用性に影響を及ぼすパフォーマンスの問題についてアラートを生成し、対応することができます。
詳細については、「[Application Insights とは何か?](#)」を参照してください。
- 回復性に関するその他の記事については、「[信頼性の高い Azure アプリケーションを設計する](#)」を参照してください。

セキュリティ

- セキュリティは、重要なデータやシステムの意図的な攻撃や悪用に対する保証を提供します。
詳細については、「[セキュリティの重要な要素の概要](#)」を参照してください。
- セキュリティで保護されたソリューションの設計に関する一般的なガイダンスについては、「[Azure のセキュリティのドキュメント](#)」を参照してください。
- エンドユーザの認証認可について、社外ユーザはAzure AD B2C、社内ユーザはAzure ADのご利用をご検討ください。

考慮事項

コスト最適化

- 不要な費用を削減し、運用効率を向上させる方法を検討することです。詳しくは、[コスト最適化の柱の概要](#)に関する記事をご覧ください。

オペレーショナルエクセレンス

- システムの健全性の担保、トラブルの解決、利用動向の監視を行うためには適切な監視とログ収集が必要となります。詳細は「[ワークロードの監視](#)」をご参照ください。
API Managementを利用することで、API利用の監視やトレースを行うことが容易になります。
- ソフトウェアのアップデートや脆弱性への対応など、ソフトウェア/インフラ設計の改修を円滑に進められるよう、DevOpsプロセスを確立してください。詳細は「[リリース エンジニアリングの継続的インテグレーション](#)」をご参照ください。

パフォーマンス効率

- アプリケーションの負荷が高まることを見越し、スケーラビリティの確保をあらかじめ検討することは重要です。詳細は「[スケーリング用のアプリケーションを設計する](#)」をご参照ください
- App Serviceは負荷に応じて水平にスケールさせることが可能です。詳細については「[自動スケーリングを有効にする方法](#)」をご参照ください。
- Enterpriseレベル以上のRedis Cache for Azure（Search機能のご利用に必要）は優れたパフォーマンスを発揮しますが、さらなる性能を必要とする場合にはRedis Cacheインスタンスを増やし負荷分散することも可能です。
また、頻出のクエリについてはアプリ側でキャッシュする等のキャッシュ戦略もご検討ください。詳細は「[キャッシュを使用する](#)」をご参照ください。
- 特定のユーザーにAzure OpenAI Service の利用が集中することを避けたい場合には API Management によるスロットリング導入などをご検討ください。詳細は「[Azure API Management を使用した高度な要求スロットル](#)」をご参照ください。



株式会社ディープコム

Copyright © 2023 DeepCom Co., Ltd.

<https://deepcom.co.jp/>