企業 / 店舗 問合せ回答 サービス

K合同会社

コンテンツ

シナリオ概要
アプリUI
アーキテクチャ
デプロイ方法
考慮事項

シナリオ概要

企業 / 店舗　問合せ回答サービス

Azure OpenAI Serviceを利用した高度な自然言語処理により、自然な会話が可能な問い合わせ回答サービスを作成します。

Azure Cognitive Searchを組み合わせることで、以下の機能を備えた問合せ回答サービスを作成することができます。

・Q&Aの事前登録
・音声応答
・多言語翻訳

これらの機能により、企業や店舗の問い合わせ回答をより多くのお客様に、自然な会話で提供します。

アプリUI

管理者画面 – ホーム画面設定

＜AIに関する設定＞
登録されたQ&Aを利用する度合いを設定します。
Q&Aの検索後の信頼度スコアと本設定値で回答の採否を決定します。

登録されたQ&Aに適切な回答がない場合に応答するメッセージを設定します。

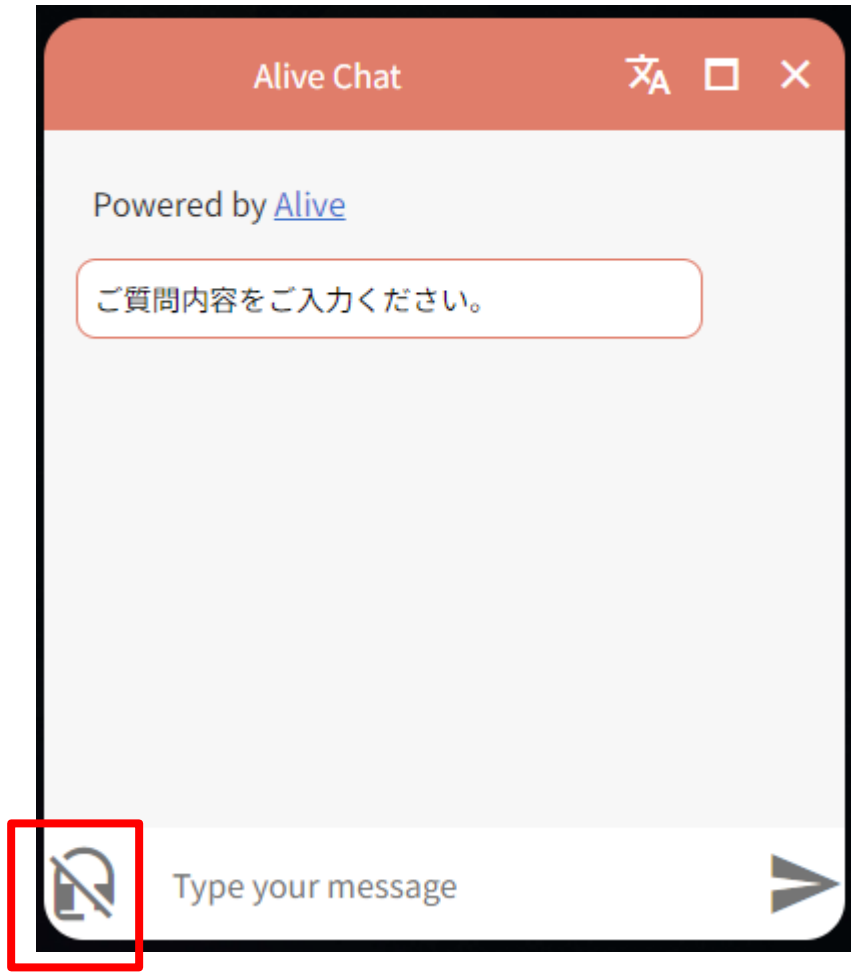
管理者画面 - サイト設定

＜AIに関する設定＞
AIに対して設定する指示内容を登録します。

管理者画面 - Q&A登録

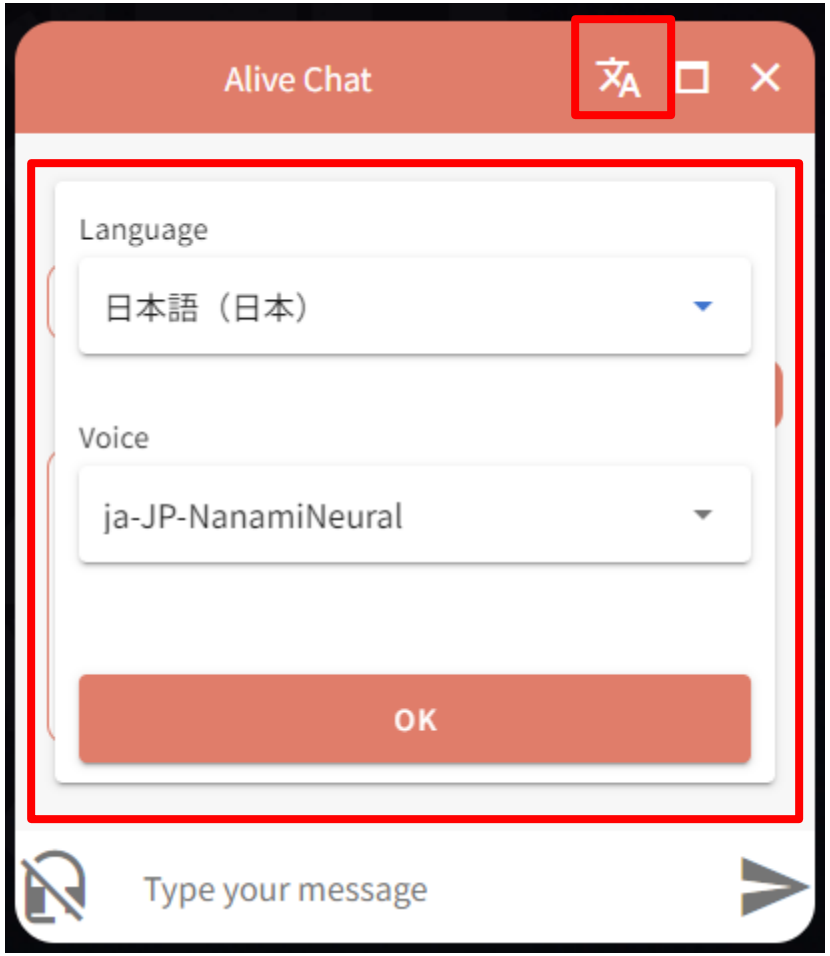
＜AIに関する設定＞
企業や店舗に予想されるQ&Aを登録します。
なるべくたくさん登録することで、AIの対応の幅が広がります。

利用者側画面 – 音声応答



＜音声応答機能＞
ヘッドセットアイコンを押下することで音声応答を可能にします。

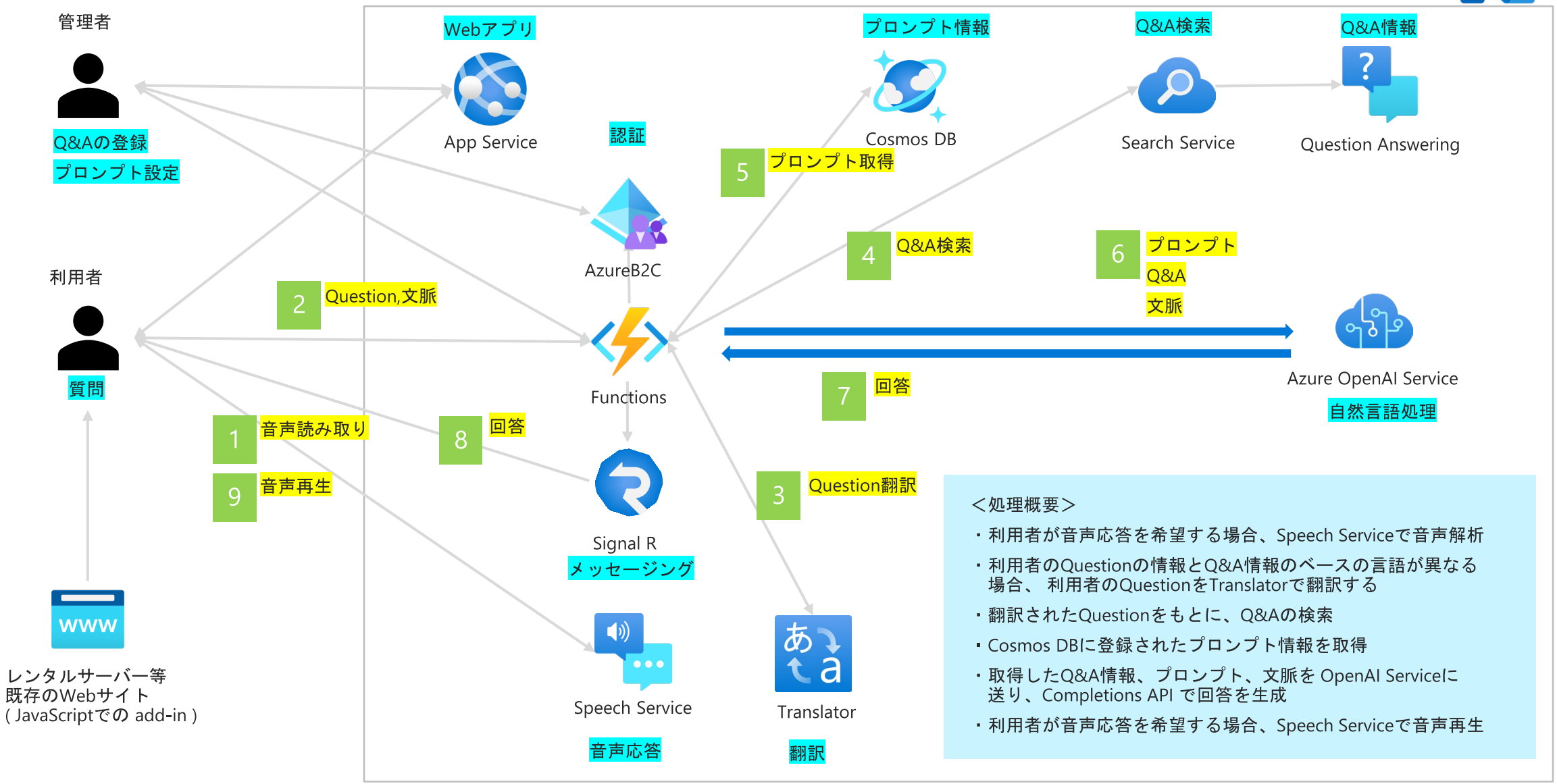
利用者側画面 – 翻訳機能



＜翻訳機能＞
翻訳アイコンを押下することで言語を選択可能にします。
また、音声を選ぶことも可能にします。

アーキテクチャ

コンポーネントとデータフロー



デプロイ方法

デプロイ方法

開発

サーバー側	Azure Function	Node.jsで開発
フロント側	Webサーバー	Html, Javascript (Vue.js)で開発

デプロイ方法

サーバー側	VSCodeよりデプロイ
フロント側	FTPクライアントよりアップロード

考慮事項

各コンポーネント別の考慮事項

Question Answering

- Search Service にはベースの言語を設定する必要があり、検索の際にはその言語で検索する必要があります。
  そのため、Q&Aのデータが複数言語にまたがる場合は、それぞれにプロジェクトを作成するか、あらかじめ
  １つの言語変換してからQ&Aデータを作成してください。

Azure OpenAI Service

- API１回あたりのトークン（≒文字数）のサイズが決められているため、プロンプト、文脈、Q&Aの情報を適切に混合してサイズ内に収めてください。
  例）プロンプトは300文字までに固定、文脈は直近から順に、Q&Aは一致度の高いものから順に、交互に設定。

- 音声応答による会話をしたい場合、応答速度が高い事が望まれます。
  トークンの上限サイズを小さくする、応答の速いモデル（gpt-3.5-turbo）を使う、短い回答をするようにプロンプトで指示するなど、対策を講じてください。

- 誤った回答を生成する可能性を減らしたい場合、Question Answeringでの検索結果の confidenceScore（信頼度スコア）が低い場合は生成AIを利用しないなど、対策を行ってください。

Azure Well-Architected Framework観点での考慮事項 (1)

Microsoft Azure Well-Architected Framework

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 Azure App Service 、Azure OpenAI 等のコンポーネントはゾーン冗長、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。複数リージョン間で Act-Act 構成を取る場合には Azure Front Door や Traffic Manager などの利用を推奨します。

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。
- このシナリオでは、Azure AD B2Cを使用してユーザーを認証します。
- 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。

Azure Well-Architected Framework観点での考慮事項 (2)

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。
- 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。

K合同会社