Solution Design Document

Project Title: AI Chatbot Using RAG, Azure OpenAI & Azure Cloud Infrastructure

Date: 24 June 2025

---

1. Purpose of the Document

This document provides a comprehensive solution design for implementing an **AI-driven customer support chatbot** using **Retrieval-Augmented Generation (RAG)**, **Azure OpenAI**, **Hugging Face models**, and **Azure-native infrastructure**.

The objective is to automate customer support by enabling users to interact with a smart, responsive chatbot that understands natural language, provides relevant answers using company knowledge, and reduces the number of support calls routed to human agents. The solution will lower operational costs, enhance customer experience, and provide real-time visibility into support operations.

---

2. Business Objective

The organization currently handles:

- **500 customer support calls on weekdays**
- **100 calls daily on weekends**
- Each call costs the company approximately **AUD 20 for a 10-minute interaction**

This leads to **over AUD 240,000 in monthly support expenses**.

This project aims to:

- **Reduce these costs** by deflecting a significant volume of queries to an automated chatbot
- **Deliver faster, 24/7 responses** to customers
- **Free up human agents** to handle complex, high-value queries
- **Track performance and ROI** through real-time analytics

---

3. High-Level Architecture Overview

The solution consists of the following key components, all orchestrated within **Microsoft Azure Cloud**:

Component	Description
Azure OpenAI	Powers natural language understanding and generative responses
Hugging Face RAG	Retrieves knowledge-based answers using internal documents
Gemini Embeddings	Converts documents and queries into high-dimensional vectors
Chroma Vector DB	Stores embeddings for similarity search
Azure Data Lake Storage (ADLS)	Stores raw documents, interaction logs, and audit data
Azure Synapse Analytics	Analyzes chatbot usage, trends, and business KPIs
Power BI	Presents dashboards and performance insights
Azure Functions / AKS	Hosts orchestration logic between components
Azure Active Directory (AAD)	Enforces user access controls and authentication

---

4. RAG-Based Chatbot Flow

The solution uses a **Retrieval-Augmented Generation (RAG)** pipeline for accurate, grounded answers.

Workflow Steps:

1. **Document Embedding**
   - Internal resources like FAQs, policies, and guides are processed through **Gemini Embeddings** to convert them into vector format.
2. **Indexing to Vector Store**
   - These embeddings are stored in the **Chroma vector database**, enabling similarity search.
3. **User Query Encoding**
   - When a user enters a question in the chatbot, the query is converted to an embedding using the same Gemini model.
4. **Similarity Search**
   - The query embedding is matched against the stored document embeddings to find the most relevant sources.
5. **Document Retrieval**
   - The top results (top-K documents) are retrieved from Chroma.
6. **Prompt Construction**
   - These retrieved documents are appended to the user query and sent to **Azure OpenAI's GPT model**.
7. **LLM Response Generation**
   - The LLM generates a context-aware response grounded in company knowledge and returns it to the user.

This architecture ensures responses are not only accurate but also verifiable against company-approved data.

---

5. System Component Mapping

Function	Technology
Chat UI	Existing Web and Mobile App UI
Natural Language Model	Azure OpenAI (GPT-4 / GPT-4o)
Embedding Model	Gemini Embeddings
Vector Storage	Chroma
Document Source	Azure Data Lake Storage (ADLS)
Processing & Orchestration	Azure Functions or Azure Kubernetes Service (AKS)
Authentication	Azure Active Directory (AAD)
Analytics Pipeline	Azure Synapse Analytics
Dashboarding Tool	Power BI

---

6. Audit and Logging

For accountability, debugging, and compliance, the system captures detailed logs of every chatbot interaction.

Data Captured Per Session:

- **User ID and Role** (from AAD)
- **User Query** and detected **intent**
- **Conversation Start and End Time**
- **Interaction Duration (minutes)**
- **Number of Back-and-Forth Exchanges**
- **Embedding and Retrieval IDs**
- **LLM version used**
- **Resolution Status**: Resolved or Escalated
- **Escalation Reason** (if applicable)
- **Feedback Score** (if collected)
- **Date of Interaction**

Logs are stored in **Azure Data Lake Storage**, processed in **Synapse**, and visualized in **Power BI**.

---

7. Security and Access Control

Security is enforced at every level of the architecture:

- **Data Encryption**: All data is encrypted at rest and in transit
- **Authentication & Authorization**: Managed by **Azure AD**

- **Access Control**: Role-based permissions for users, agents, and admins
- **Monitoring**: All access and activities tracked using **Azure Monitor** and **Log Analytics**

Only authorized personnel can access chatbot data, audit logs, or model configurations.

---

8. Monitoring & Analytics

The solution provides deep observability and insights using Azure-native tools:

- **Azure Monitor** for real-time system health
- **Application Insights** for request/response tracing
- **Power BI Dashboards** for:
  - Number of users and sessions
  - Percentage of queries resolved without escalation
  - Escalation rate trends
  - Response time metrics
  - Estimated operational cost savings

Dashboards help stakeholders make data-driven decisions and continuously improve the solution.

---

9. Continuous Integration and Deployment (CI/CD)

CI/CD Goals:

- Ensure rapid and secure deployments
- Eliminate manual deployment errors
- Automate rollback and rollback detection

Tools Used:

Function	Tool Used
Source Control	GitHub / Azure Repos
CI/CD Automation	Azure Pipelines / GitHub Actions
Container Registry	Azure Container Registry (ACR)
Infra Provisioning	Terraform / Bicep
Monitoring Deployments	Application Insights

Workflow:

1. Developer pushes code
2. CI Pipeline builds, tests, and packages artifacts

3. Artifacts deployed to **Dev > Staging > Production**
4. Manual approval for Production
5. Post-deployment testing and health checks
6. Rollback initiated automatically if failure occurs

All stages are logged and monitored for auditability.

---

10. Benefits of the Solution

- ✅**Cost Reduction**: Shifts thousands of call center conversations to automated chat, cutting expenses dramatically.
- ✅**Scalability**: Azure-native components allow scaling to any load.
- ✅**24/7 Availability**: Chatbot can serve users anytime without downtime.
- ✅**Accuracy**: RAG ensures responses are based on real internal data.
- ✅**Observability**: Track KPIs with real-time analytics and audit logs.
- ✅**Security and Governance**: Enterprise-grade access control and data protection.

---

11. Future Enhancements

To evolve with user needs and technological advances, the system will be extended to include:

- □ **Voice Integration** via Azure Speech Services
- □ **Multilingual Support** for global users
- □ **Feedback-based Model Tuning** using real interaction data
- □ **Proactive Messaging** via WhatsApp, SMS, or email
- □ **Continual Learning** from unresolved or escalated queries