|  |  |
| --- | --- |
|  | ISD Department: [Consumer Applications]  Project ID/Title: PR10956: 102Ki ADR CX AI - Product Model |
| ISD Methodology : Solution Architecture Document | |

Solution Architecture Document

Version 1.1

10/1/2025

Table of Contents

[1 Solution Summary 4](#_Toc212462397)

[2 Overview 6](#_Toc212462398)

[2.1 Technical Assumptions 6](#_Toc212462399)

[2.2 Technical Risks 7](#_Toc212462400)

[3 Solution Architecture 8](#_Toc212462401)

[3.1 Architectural Capabilities 8](#_Toc212462402)

[3.1.1 Standard Capabilities 8](#_Toc212462403)

[3.1.2 Custom Capabilities 8](#_Toc212462404)

[3.2 Overview of Package Solution / Package Evaluation Results 8](#_Toc212462405)

[3.2.1 Owner and Salesperson AI Prototypes: 9](#_Toc212462406)

[3.2.2 Evaluation Results 10](#_Toc212462407)

[3.3 System Context Diagram 11](#_Toc212462408)

[3.4 Logical Architecture Diagram 12](#_Toc212462409)

[3.5 Operational Monitoring and Reporting 14](#_Toc212462410)

[4 Infrastructure Architecture 15](#_Toc212462411)

[4.1 Infrastructure Block Diagram 15](#_Toc212462412)

[4.1.1 Diagram 15](#_Toc212462413)

[4.1.2 End User Computing Device 15](#_Toc212462414)

[4.2 Technical Architecture Diagram 16](#_Toc212462415)

[5 Security Architecture 17](#_Toc212462416)

[5.1 User Provisioning 17](#_Toc212462417)

[5.2 User Authentication 17](#_Toc212462418)

[5.3 Data Security 22](#_Toc212462419)

[5.3.1 Production Data Security 22](#_Toc212462420)

[5.3.2 Electronic Signatures and Receipts 22](#_Toc212462421)

[5.3.3 Data Masking for Non-Production 22](#_Toc212462422)

[5.3.4 Database Activity Monitoring for Production 23](#_Toc212462423)

[5.3.5 Security Risk Assessment 23](#_Toc212462424)

[5.3.6 Security Risk Mitigations 27](#_Toc212462425)

[6 Enterprise Services 28](#_Toc212462426)

[6.1 Current Available Services used in Project 28](#_Toc212462427)

[6.2 Services to be created by Project 28](#_Toc212462428)

[7 Integration Architecture 29](#_Toc212462429)

[7.1 Integration Architecture Diagram 29](#_Toc212462430)

[7.1.1 Data Ingestion Architecture Diagram 29](#_Toc212462431)

[7.1.2 MyGarage Application Integration 30](#_Toc212462432)

[7.1.3 Hondaweb Application Integration 30](#_Toc212462433)

[7.2 Integration Points 30](#_Toc212462434)

[7.2.1 Data Ingestion Integration Points 30](#_Toc212462435)

[7.3 Real-Time and Batch Integration 31](#_Toc212462436)

[7.4 Internal Interfaces 31](#_Toc212462437)

[7.5 External Interfaces 31](#_Toc212462438)

[8 Application Architecture 32](#_Toc212462439)

[8.1 RAG Overview 32](#_Toc212462440)

[8.2 Owner AI Application Architecture 32](#_Toc212462441)

[8.2.1 End User Query and Response Sequence 32](#_Toc212462442)

[8.2.2 Internal User flow 34](#_Toc212462443)

[8.2.3 Data Ingestion Process 34](#_Toc212462444)

[8.3 Application Framework 34](#_Toc212462445)

[8.4 Security 34](#_Toc212462446)

[8.5 User Interface (UI) 35](#_Toc212462447)

[8.6 Portal 35](#_Toc212462448)

[8.7 Business Services 35](#_Toc212462449)

[8.8 Service Registry 35](#_Toc212462450)

[8.9 Business Process Engine 35](#_Toc212462451)

[8.10 Application/Core Components 35](#_Toc212462452)

[8.10.1 Amazon Cognito 36](#_Toc212462453)

[8.10.2 AppSync 37](#_Toc212462454)

[8.10.3 Lambda 37](#_Toc212462455)

[8.10.4 Amazon Simple Queue Service (SQS) 37](#_Toc212462456)

[8.10.5 Amazon Simple Notification Service (SNS) 37](#_Toc212462457)

[8.10.6 DynamoDB 37](#_Toc212462458)

[8.11 Persistence 37](#_Toc212462459)

[8.11.1 Amazon Simple Storage Service (S3) 37](#_Toc212462460)

[8.12 Integration - ESB 38](#_Toc212462461)

[8.13 Integration - ETL 38](#_Toc212462462)

[8.14 Database 38](#_Toc212462463)

[*8.14.1* Amazon Aurora PostgreSQL 38](#_Toc212462464)

[8.15 Scheduler 38](#_Toc212462465)

[8.16 Operational Reporting Tool 38](#_Toc212462466)

[8.17 Analytical Reporting Tool 38](#_Toc212462467)

[8.18 Fax 39](#_Toc212462468)

[8.19 Operating System 39](#_Toc212462469)

[8.20 Backup / Recovery 39](#_Toc212462470)

[9 Data Architecture 39](#_Toc212462471)

[9.1 Conceptual Data Model (J0/J1) 39](#_Toc212462472)

[9.2 Logical Data Model (J1) 40](#_Toc212462473)

[9.3 Physical Data Model (J1) 40](#_Toc212462474)

[10 Revision History 40](#_Toc212462475)

# Solution Summary

**Owner AI Assistant**

**Product Value Proposition   
A00:** Provide vehicle owners with an AI-enhanced owner resource that allows them to easily find information, receive assistance and accomplish tasks using natural language – improving the customer experience and driving improved customer satisfaction.

**A01:** The AI Assistant provides accurate answers to vehicle-related questions.

**A02:** Users can easily verify the source of provided information by following links to cited resources.

**A03:** Users interacting with the AI Assistant are more satisfied than with their traditional MyGarage experience

**Primary Users**

Current US Acura vehicle owners visiting Acura Help Center on MyGarage:

* 2 million US owners overall
* 2,700 visits per month to Acura Help Center on MyGarage
* May or may not be logged into MyGarage (70% not logged in)

Product users will expand to both Acura and Honda vehicle owners reaching additional areas of MyGarage and/or other owner channels in future releases.

**Business Problem**

**Problem Statement of Customer:** Owners have difficulty finding needed information and resources due to a range of performance, search and navigation issues on resources like MyGarage, causing frustration and impairing their ability to resolve issues.

**Problem Statement of Honda:** Dissatisfied customers (as reflected in low MyGarage NPS scores) have a negative impact on loyalty and brand advocacy. Inability to self-serve may also lead to higher CR support costs.

**Competitor/Market Landscape:** Increasing number of competitors (Ford, Volkswagen, Toyota, etc.) offering AI-enhanced owner manuals and similar advanced owner resources. Customer expectations and behaviors rapidly shifting to use of generative AI/semantic search for improved information retrieval.

**Opportunity**

**Owner AI Assistant:** Provide owners with an intelligent AI search and information retrieval feature that centralizes and streamlines access to data sources and formats, allowing owners to get **immediate answers to any question** using **natural language.** Initial use cases will focus on vehicle information retrieval and can expand over time to include service appointment scheduling, order status, personalized recommendations and other features.

**Value to Customers:** A more usable and intelligent owner resource that allows owners to find immediate answers to their vehicle-related questions.

**Value to Honda:** More satisfied owners who feel supported, leading to stronger NPS, reduced reliance on support channels, and improved loyalty.

**Key Performance Indicators (KPI):**

**AI Response Accuracy:**

* Measures the AI’s ability to provide factually accurate and reliable answers based on knowledge grounding
* Target: 90%

**AI Reference Accuracy:**

* Measures the AI’s ability to accurately reference source documentation from the knowledge base
* Target: 90%

**NPS:**

* Measures NPS of customers interacting with the AI Assistant in comparison to customers interacting with current search feature on Acura Help Center.
* Target: +20 vs. control group

# Overview

The proposed solution for this project is based on the recent advancements in the field of Generative Artificial Intelligence (aka “Gen AI”). Solutions that are based on Gen AI capabilities allow the end users to communicate with a digital assistant, such as a chatbot on a website, using natural language. The Gen AI solutions can use different Large Language Models (LLM) which respond to specific instructions (or “prompts”) to generate their output. The LLMs are pre-trained with publicly available information and have different levels of output generation abilities including text, image, and videos (LLMs can be trained with private information by customers, however, we will be using LLMs that are publicly available through the AI platform providers)..

To ensure that users of Honda’s AI assistants receive accurate and reliable information from official sources, the solution will leverage the Retrieval Augmented Generation (RAG) design which uses the official Honda content such as the content available on Honda and Acura websites, Owner’s manuals (in PDF files) as the primary source of information for the generated response.

For the Owner AI Pilot project Release 1, the solution will utilize the Frequently Asked Question (FAQ) content in the Acura Help Center section of the MyGarage website ([Acura Help Center](https://mygarage.honda.com/s/help-center-acura)). In releases 2 and 3, additional content (i.e. knowledgebases) i.e. Owner’s Manual (PDF files), will be added. How to Videos (hosted on You Tube) may be added at a future release.

The solution will be accessible as a chatbot feature available on the MyGarage website and will be available to all users (Authentication not required). Internal users will have access to dashboards and performance evaluation metrics from Hondaweb. While MyGarage is currently hosted on the Salesforce platform, the Owner AI assistant’s backend logic will be hosted on the AWS platform.

## Technical Assumptions

| **TA #** | **Category** | **Assumption** |
| --- | --- | --- |
| 1 | Hosting Platform | AWS and Amazon Bedrock were selected as the AI platform provider and will provide the services and LLMs required to enable this solution. It is assumed that this platform will be aligned with the future AI strategy for AHM and Honda Japan. |
| 2 | Source of Data | It is assumed that the source system for Acura FAQ, where they are authored and owned is the CRRS team. Once created, a copy is shared with MyGarage, where they are made accessible to users (in HTML format). As of the time of writing this document, the source data is neither available in any other format nor in any other location. However, this may change in the future. In addition, a request has been submitted to add a custom field to the Knowledge Article object which will be used to store relevant meta data about the question and answer. This meta data will be helpful to the LLM to respond with accurate answers. |

## Technical Risks

1. End user performance may be perceived as too slow due to the need for a high level of response accuracy. Appropriate server capacity and tuning, along with User Interface techniques that keep the user engaged while the repose is being generated must be implemented.
2. This solution includes AWS services that are new to Honda NA environment (e.g. AppSync, Cognito, AppFlow). While the B2C team will provide production support for the immediate future, the Enterprise Shared Services team is responsible for providing cloud support and they are not yet ready to provide the required support. A plan to transition long-term production support to the Enterprise Shared Services team must be put in place.
3. Applications using Gen AI are dependent on a technical ecosystem that is rapidly changing. New services and tool sets are being constantly introduced. The solution presented in this document must be reevaluated on a periodic basis to ensure it remains efficient, secure, and in sync with the latest advances in the field of artificial intelligence.

See Section [Architectural Decisions](#ArchitecturalDecisions)

# Solution Architecture

## Architectural Capabilities

This solution will be a classic RAG application using Amazon Bedrock. As Amazon enhances Bedrock’s capabilities, it will be enhanced to take advantage of such capabilities including “Agentic-AI” architectures.

### Standard Capabilities

| # | Standards ID # | Capability | Description |
| --- | --- | --- | --- |
| 1 | SC-01 | AWS services will be used which is Honda’s standard Cloud Provider | AWS standard services used in this solution include S3, Lambda, SQS, SNS, CloudWatch, KMS, Secrets Manager, OpenSearch, Aurora DB, DynamoDB. New non-standard AWS services introduced with this project include Cognito, AppSync, and AppFlow |
| 2 |  |  |  |

### Custom Capabilities

| # | Capability | Description | Justification |
| --- | --- | --- | --- |
| 1 | N/A |  |  |

## Overview of Package Solution / Package Evaluation Results

Three AI platform providers were evaluated as part of this project:

* AWS
* Microsoft
* Salesforce

The platform providers were evaluated against 72 evaluation criteria across 10 categories. Evaluations were based on vendor presentations, prototypes, and Honda staff's architecture and code reviews. The same set of Honda artifacts (knowledgebases) were used for all platforms. Scoring was a team effort with multiple review rounds.

### Owner and Salesperson AI Prototypes:

All three platform providers built working prototypes in Honda’s accounts. The following requirements were given to all vendors:

* Response Accuracy: Highest priority for recommending LLMs by each of the platform providers.
* Managed Services: Focus on LLMs supported by the platform’s managed AI services (e.g., AWS Bedrock, Azure AI Foundry, Salesforce Einstein) to leverage built-in features like guardrails, evaluation, and monitoring.
* Flexibility: LLMs used in prototypes can be easily switched as new models are introduced.

Prototypes developed on each of the three platforms based on PRDs defining basic feature requirements:

* Chat interface
* Knowledge retrieval (FAQs, OMs, videos, ODSX assets)
* Guardrails
* Citations with links

The prototypes were evaluated to understand:

* Actual platform features (vs. claims)
* Data preprocessing, dev process, UIs
* Skills needed, out of the box capabilities
* Service/partnership provided
* Token usage and pilot operating costs.

### Evaluation Results

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | AWS Bedrock | Microsoft Azure AI | Salesforce Agentforce |
| Strengths | * Highest quality responses due to thorough prompt engineering and LLM selection * Developed web scraping tool to ingest all 750+ MyGarage FAQs * Flexible design system for building UI * Strong prototype elements transferrable to prod (prompt engineering, FAQ index) * UI for side-by-side LLM comparison * Lowest operating cost | * Strong video indexing and deep linking * Intuitive demo interface | * Low-code platform / user-friendly dev * Compelling Testing Center feature (but non-functional) * Late in process provided strong developer resource implemented improved data pre-processing methods and response quality |
| Weaknesses | * High latency due to multiple retrievals (addressable) * Confusing initial UI (later fixed) * Citation links partially functional (addressable) | * Minimal effort made to develop and iterate * Poor indexing/inability to differentiate across vehicle models * Several key features in “Preview” status * Fewer LLM Providers (mostly OpenAI) | * Many claimed features missing or non-functional, even months after release * Limited out-of-the-box chat interface * Poor initial development effort/support * Highest operating cost |
| Prototype Rating |  |  |  |

## System Context Diagram



## Logical Architecture Diagram

The Owner AI application is composed of 5 logical layers as described below:

| **Layer Name** | **Layer Purpose** |
| --- | --- |
| **User Interface** | The User Interface contains logic that operates from the user’s browser, capturing user input and displaying system out. The Owner AI application includes two types of user Interface: 1) End User Chatbot, and 2) Internal user monitoring and reporting. They are both built using the popular React front-end framework and are distributed to the user’s browser as a Single Page App (SPA) using Akamai Content Delivery Network (CDN) service for improved performance and security.   * Chatbot user interface captures the user’s queries and displays the response as it is streamed back from the back end. * The internal user interface is used to access various AI metrics designed to measure the application’s performance. |
| **Security & API Gateway** | Responsible for receiving the API calls from the User Interface and invoking the appropriate logic depending on the type of user, and type of request. In the future, this layer will perform user authentication logic which will allow for a more personalized user experience based on user profile and relationship with the company. |
| **Integration** | This layer will provide synchronous and asynchronous integration mechanisms including queuing and event processing which will be specially required for future extendibility to a multi-agenetic architecture in which different agents perform various tasks including interaction with each other based on their reasoning capability |
| **Model Orchestration** | In this layer, the core feature of user input augmentation from corporate knowledge bases and vector database are implemented. |
| **Knowledge & Persistence** | In this layer, the knowledgebases, vector database, and the Large Language Model(s) are hosted. The LLMs are accessed through the Bedrock APIs (i.e. not self-hosted within Honda’s account) |

The following diagram depicts the various components and AWS services used in each layer:



## Operational Monitoring and Reporting

Operational monitoring and reporting for this solution include the following capabilities:

**Amazon CloudWatch Logs:**

* Provides logs that record API behavior, assist in performance bottlenecks, and record LLM token generation and usage counts
* Log data can be queried using Log Insights Query Language
* Logs can be streamed to other services and stored for further processing, reporting, and visualization if needed (future requirement)
* Amazon Bedrock provides for Invocation logging which is disabled by default (can be enabled if needed). With invocation logging, you can collect the full request data, response data, and metadata associated with all calls performed. Logging can be configured to provide the destination resources where the log data will be published. Supported destinations include Amazon CloudWatch Logs and Amazon Simple Storage Service (Amazon S3).

**AWS AppSync:**

AWS AppSync is a managed GraphQL service and is needed to provide subscription to streaming data coming back from the LLM. It provides robust monitoring and reporting capabilities through integrations with Amazon CloudWatch and AWS X-Ray, providing insights into API usage, performance, and errors. For this project, we can:

* Track key metrics,
* enable detailed logging, and
* leverage tracing to diagnose issues and optimize GraphQL APIs.

**RAG Evaluations:**

Amazon Bedrock provides built-in and custom capabilities to evaluate the RAG application’s performance using metrics including:

**Quality metrics:**

1. Helpfulness
2. Correctness
3. Logical coherence
4. Faithfulness
5. Completeness
6. Citation precision
7. Citation coverage

**Responsible AI metrics:**

1. Harmfulness
2. Refusal
3. Stereotyping

The metrics values can be customized as appropriate. Below is a screenshot of the prototype’s evaluation results:

# Infrastructure Architecture

## Infrastructure Block Diagram

The solution will be completely cloud-based and hosted in a new AWS Account in us-east-1 region. The underlying infrastructure is abstracted. The us-east-1 region provides the highest set of available features for Amazon Bedrock.

### Diagram

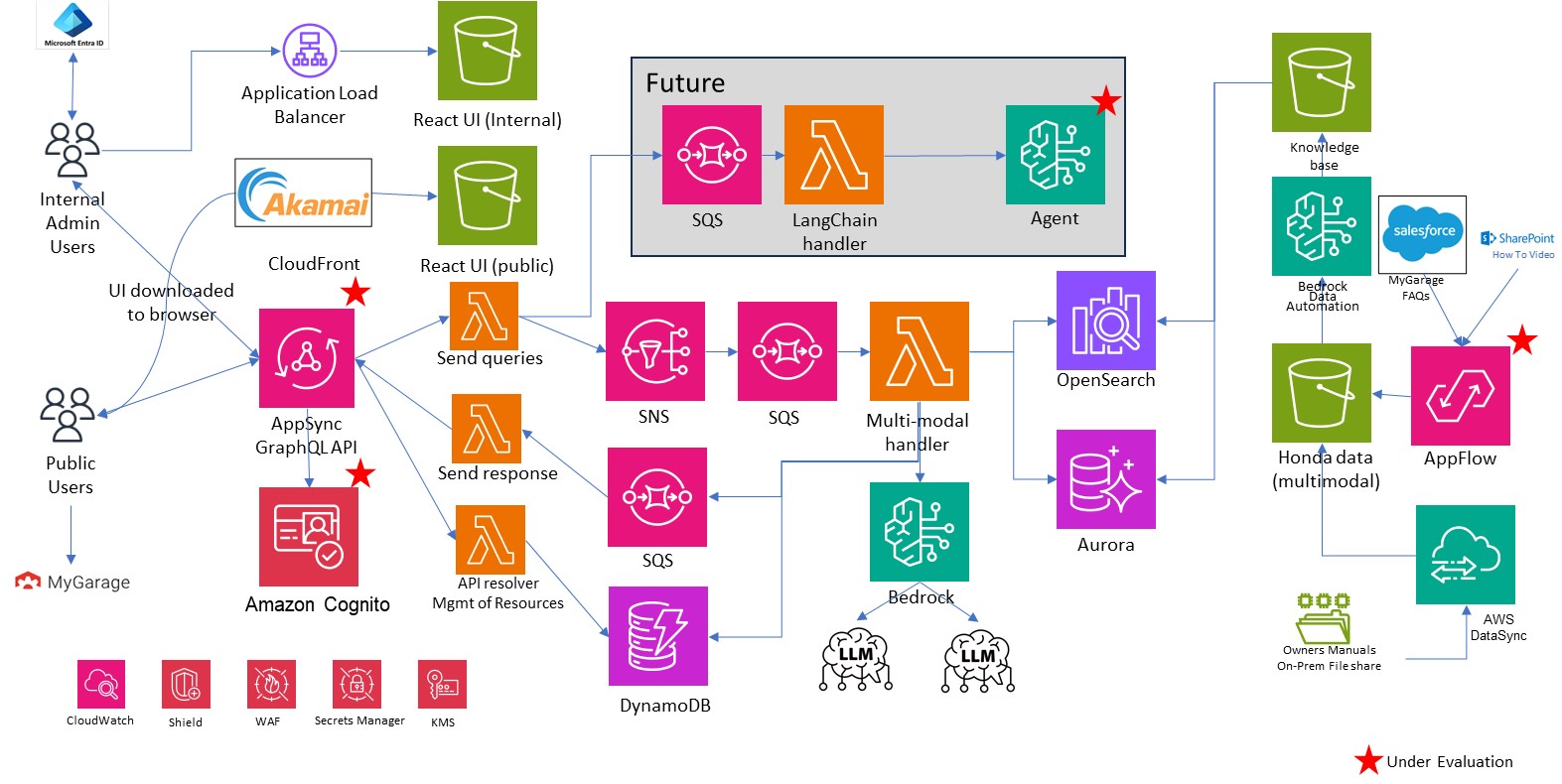
N/A

### End User Computing Device

* MyGarage users will use desktop and mobile devices.
* Internal Users will use Honda issued laptops.

## Technical Architecture Diagram

The following diagram shows the Owner AI application using AWS services:



# Security Architecture

## User Provisioning

Internal users (admins, analysts) will be given the required role in Saviynt and the application icon/link will be made available for them on Hondaweb (or MyApps).

## User Authentication

There will be two types of users accessing the Owner AI chatbot:

1. End users (visitors of MyGarage website)
2. Internal Business and/or IT administrators

For the Owner ID Pilot, the end users may be authenticated or unauthenticated. The chatbot’s functionality and user experience will be the same for both groups for the Pilot period. It is conceivable that in the future, authenticated users will have a more personalized experience and more extensive feature set.

No end-user authentication is needed to access this application (it is open to the public). However, we need to incorporate features that ensure the chatbot is only used by active visitors to MyGarage and it cannot be shared to bad actors. The protection approach for unauthenticated users must be implemented at all layers of the application to provide adequate protection:

|  |  |  |
| --- | --- | --- |
| Layer | Components Being Protected | Protection Approach |
| 1 | Chatbot endpoint | The chatbot User Interface is protected using the Akamai Content Delivery Network which provides defenses against DDoS attacks.  The chatbot’s APIs (exposed by AppSync) are private and only reachable through the Enterprise API Gateway (IBM DataPower). |
| 2 | API Gateway (AppSync) | AWS WAF (Web Application Firewall) can be integrated with AppSync to create web ACLs (Access Control Lists). These web ACLs can include rate-based rules to throttle requests based on criteria like IP address, preventing single sources from overwhelming the API. This step may be skipped if DataPower can perform this function.  GraphQL Execution Limits: AppSync allows configuring limits on GraphQL query complexity, depth, and the maximum number of resolvers per query. This helps prevent overly complex or deeply nested queries that could consume excessive resources or lead to denial-of-service (DoS) scenarios. |
| 3 | Application functions and data | Role Based Access (RBAC) will be used limiting access to functionality and data based on the role of the user. For guest (unauthenticated) users limits will can be used in access to tools (and later to AI Agents), knowledgebases (only public data will be made available to unauthenticated users), and usage limits (unauthenticated users may be subject to a limited number of LLM tokens used).  Furthermore, application will leverage temporary credentials (created by Amazon STS service) which will expire after a pre-determined period of time (default is 1 hour). Usage of temporary credentials is a security best practice and protects against the risk of permanent credentials being distributed for misuse. |

The chatbot solution will leverage a CDN (probably Akami which is the product used by Honda B2C applications) to defend against DDOS attacks. These features include:

* **Traffic Distribution & Absorption:**

Akamai's global network of edge servers absorbs the massive amount of traffic from a DDoS attack, distributing it across thousands of nodes so that no single server is overwhelmed.

* **Traffic Filtering:**

The [Akamai Connected Cloud](https://www.google.com/search?rlz=1C1GCEA_enUS1124US1125&cs=0&sca_esv=2fdb4996cfdf24b0&sxsrf=AE3TifPKLq3yc_4NCU23268efcTQV9fcXA%3A1756249730682&q=Akamai+Connected+Cloud&sa=X&ved=2ahUKEwi0r7KKzKmPAxV2IEQIHTjYCYsQxccNegQIJhAB&mstk=AUtExfAoItQTHg4CNSOEIUT75xyKlZ9iSbNun-RsfAWMRzR3GU3Trlq6APFOtjlrwJFZGSqiyX6SuREzitDjbjYTFG2ZmvdbaM2n0Ug6tFkGBnKmkn7WMxw1_8xklBPht2v8-XMD-lJ7lvQiv6ZKqNGOUL0368KUCcXA9bWESQq-52y4xcg&csui=3) acts as a reverse proxy, dropping traffic that isn't on standard HTTP/S ports (80/443) and filtering out known malicious IP addresses.

* [**Behavioral**](https://www.google.com/search?rlz=1C1GCEA_enUS1124US1125&cs=0&sca_esv=2fdb4996cfdf24b0&sxsrf=AE3TifPKLq3yc_4NCU23268efcTQV9fcXA%3A1756249730682&q=Behavioral&sa=X&ved=2ahUKEwi0r7KKzKmPAxV2IEQIHTjYCYsQxccNegQIDxAB&mstk=AUtExfAoItQTHg4CNSOEIUT75xyKlZ9iSbNun-RsfAWMRzR3GU3Trlq6APFOtjlrwJFZGSqiyX6SuREzitDjbjYTFG2ZmvdbaM2n0Ug6tFkGBnKmkn7WMxw1_8xklBPht2v8-XMD-lJ7lvQiv6ZKqNGOUL0368KUCcXA9bWESQq-52y4xcg&csui=3)**Analysis:**

The Behavioral DDoS Engine uses machine learning to monitor traffic patterns, establish a baseline of normal activity, and identify deviations that indicate a DDoS attack.

* **DNS Infrastructure Protection:**

[Edge DNS](https://www.google.com/search?rlz=1C1GCEA_enUS1124US1125&cs=0&sca_esv=2fdb4996cfdf24b0&sxsrf=AE3TifPKLq3yc_4NCU23268efcTQV9fcXA%3A1756249730682&q=Edge+DNS&sa=X&ved=2ahUKEwi0r7KKzKmPAxV2IEQIHTjYCYsQxccNegQIJBAB&mstk=AUtExfAoItQTHg4CNSOEIUT75xyKlZ9iSbNun-RsfAWMRzR3GU3Trlq6APFOtjlrwJFZGSqiyX6SuREzitDjbjYTFG2ZmvdbaM2n0Ug6tFkGBnKmkn7WMxw1_8xklBPht2v8-XMD-lJ7lvQiv6ZKqNGOUL0368KUCcXA9bWESQq-52y4xcg&csui=3) protects your DNS services by distributing queries across its highly available network and routing legitimate requests while blocking malicious DNS amplification attacks.

* [**Prolexic**](https://www.google.com/search?rlz=1C1GCEA_enUS1124US1125&cs=0&sca_esv=2fdb4996cfdf24b0&sxsrf=AE3TifPKLq3yc_4NCU23268efcTQV9fcXA%3A1756249730682&q=Prolexic&sa=X&ved=2ahUKEwi0r7KKzKmPAxV2IEQIHTjYCYsQxccNegQIEhAB&mstk=AUtExfAoItQTHg4CNSOEIUT75xyKlZ9iSbNun-RsfAWMRzR3GU3Trlq6APFOtjlrwJFZGSqiyX6SuREzitDjbjYTFG2ZmvdbaM2n0Ug6tFkGBnKmkn7WMxw1_8xklBPht2v8-XMD-lJ7lvQiv6ZKqNGOUL0368KUCcXA9bWESQq-52y4xcg&csui=3)**for Network-Layer Defense:**

Akamai's Prolexic solution offers large-scale network-layer DDoS mitigation through its cloud-based network and dedicated security operations center (SOC). This feature is not included in the current subscription for Honda.

* [**Collaborative Defense**](https://www.google.com/search?rlz=1C1GCEA_enUS1124US1125&cs=0&sca_esv=2fdb4996cfdf24b0&sxsrf=AE3TifPKLq3yc_4NCU23268efcTQV9fcXA%3A1756249730682&q=Collaborative+Defense&sa=X&ved=2ahUKEwi0r7KKzKmPAxV2IEQIHTjYCYsQxccNegQIERAB&mstk=AUtExfAoItQTHg4CNSOEIUT75xyKlZ9iSbNun-RsfAWMRzR3GU3Trlq6APFOtjlrwJFZGSqiyX6SuREzitDjbjYTFG2ZmvdbaM2n0Ug6tFkGBnKmkn7WMxw1_8xklBPht2v8-XMD-lJ7lvQiv6ZKqNGOUL0368KUCcXA9bWESQq-52y4xcg&csui=3)**:**

Akamai participates in collaborative defense initiatives, sharing threat intelligence and attack signatures with other organizations to enhance collective protection.

* [**Cross-origin resource sharing (CORS) Defense**](https://www.google.com/search?rlz=1C1GCEA_enUS1124US1125&cs=0&sca_esv=2fdb4996cfdf24b0&sxsrf=AE3TifPKLq3yc_4NCU23268efcTQV9fcXA%3A1756249730682&q=Collaborative+Defense&sa=X&ved=2ahUKEwi0r7KKzKmPAxV2IEQIHTjYCYsQxccNegQIERAB&mstk=AUtExfAoItQTHg4CNSOEIUT75xyKlZ9iSbNun-RsfAWMRzR3GU3Trlq6APFOtjlrwJFZGSqiyX6SuREzitDjbjYTFG2ZmvdbaM2n0Ug6tFkGBnKmkn7WMxw1_8xklBPht2v8-XMD-lJ7lvQiv6ZKqNGOUL0368KUCcXA9bWESQq-52y4xcg&csui=3)**:**

Akamai will set up Access-Control-Allow-Origin header, indicating which origins are permitted to access its resources.  In this case it will be the MyGarage website.

Amazon WAF features will provide application layer defense for requests originating from the Chatbot User Interface. In addition, temporary credentials to access AWS will be utilized and will be encrypted, randomized tokens that are generated with every user’s request.

The End user (customer) flow is described herein:

1. User Clicks on the link to navigate to the Acura Help Center page
2. MyGarage application generates a token (this token may include the CIAM JWT token in its payload). The rendered page will include a button for chatbot invocation that includes the token.
3. The user clicks on the Owner AI Assistant (chatbot) button
4. A request is sent to the chatbot’s endpoint in the Content Delivery Network (CDN), i.e. Akamai
5. The User Interface code is downloaded to the user’s machine from Akamai’s origin server (S3 bucket) and displays the welcome message
6. The user enters their query and clicks on the chatbot’s Enter icon. The logic in the User Interface sends a request to the Enterprise shared service (Data Power), including the identity token
7. DataPower receives the request, validates the token against the corresponding Identity Provider, and forwards the validated request, including the identity token and user query to the private endpoint of Amazon AppSync, invoking a GraphQL API on AppSync. If the token validation fails, an error message is returned to the UI, indicating invalid (or expired) token and asking the user to try again.
8. AppSync examines the request received from DataPower for an identity token which also includes the source application.
9. Amazon Cognito is used to federate with external Identity providers. It uses the centralized logic in the identity pool to determine the IAM role. It also provides additional security for unauthenticated identities with a scope-down policy that sets an upper limit on IAM permissions. The Cognito enhanced flow is the most secure choice with the lowest level of developer effort. To learn more about these options, see [Identity pools authentication flow](https://docs.aws.amazon.com/cognito/latest/developerguide/authentication-flow.html).
10. AppSync assumes the appropriate role returned from Cognito and invokes the required Lambda for backend processing.



The Unauthenticated user flow is shown in the diagram below:



Internal Business and/or IT administrators will access the chatbot’s dashboard and configuration features to ensure it meets accuracy and performance requirements. The internal users are authenticated using EntraID.

The internal user is provided access to the Chatbot application through existing user role provisioning tools (NASA or Saviynt). To access the application:

1. The internal user clicks on the “Owner AI Chatbot administration” application icon displayed on Hondaweb (or the new myapps.microsoft.com page)
2. The application sends the request to DataPower. The request includes the identity token indicating the user is an internal user.
3. The rest of the flow is similar to the above starting with step 6.

The internal user flow is similar to the authenticated end-user (customer) flow. The only difference is that for internal users, EntraID is the Identity Provider.

## Data Security

All Knowledgebase data used in the Pilot phase is publicly available on MyGarage (i.e. Acura Help Center Frequently Asked Questions and answers). All data in transit will be encrypted using TLS 1.2.

All data in knowledgebases will be encrypted using the default S3 encryption key.

### Production Data Security

Amazon Aurora PostgreSQL does not have encryption by default. Instead, it uses the Salted Challenge Response Authentication Mechanism (SCRAM) for password encryption, which is recommended for enhanced security. Additionally, Amazon Aurora supports AES-256 encryption for data at rest, but this requires users to enable encryption for their database instances. Therefore, while encryption is supported, it must be explicitly enabled using KMS Customer managed Keys (CMK) for the database cluster.

DynamoDB automatically encrypts all table data at rest, including indexes, streams, global tables, and backups. This is enabled by default and cannot be disabled. It is recommended to use KMS Customer Managed Keys (CMK).

CloudWatch Logs provides data protection for log data by encrypting it at rest. By default, CloudWatch Logs uses server-side encryption with 256-bit Advanced Encryption Standard Galois/Counter Mode (AES-GCM) to encrypt log data. This encryption is managed by the CloudWatch Logs service.

For enhanced security and control, it is recommended to encrypt CloudWatch Logs using customer-managed AWS Key Management Service (KMS) keys. This allows for greater control over the encryption keys and their access policies.

### Electronic Signatures and Receipts

N/A

### Data Masking for Non-Production

N/A. There is no PII in the knowledgebases used by the application. Amazon Bedrock Guardrail must be configured to detect and mask any PII data the user may enter into the chatbot.

### Database Activity Monitoring for Production

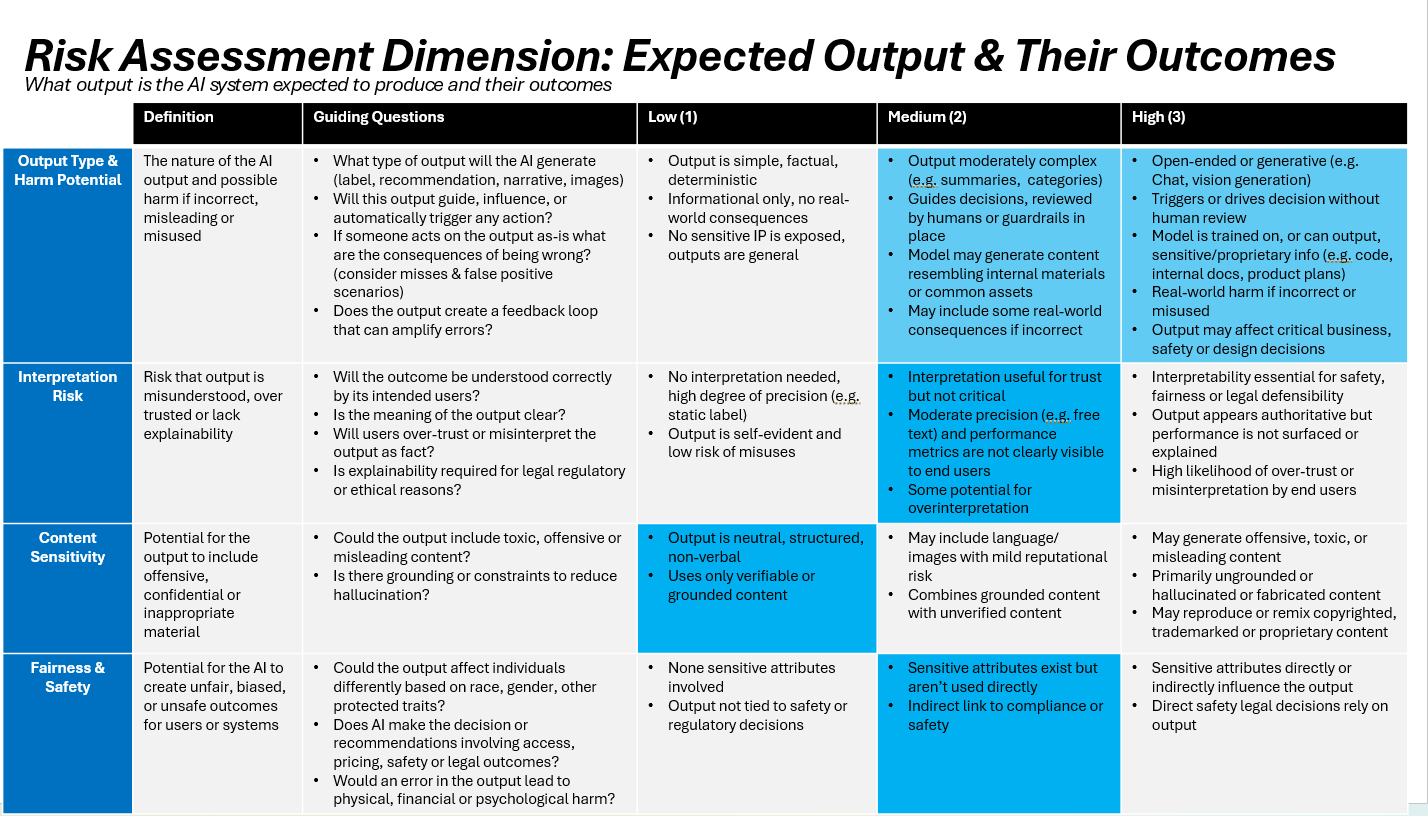
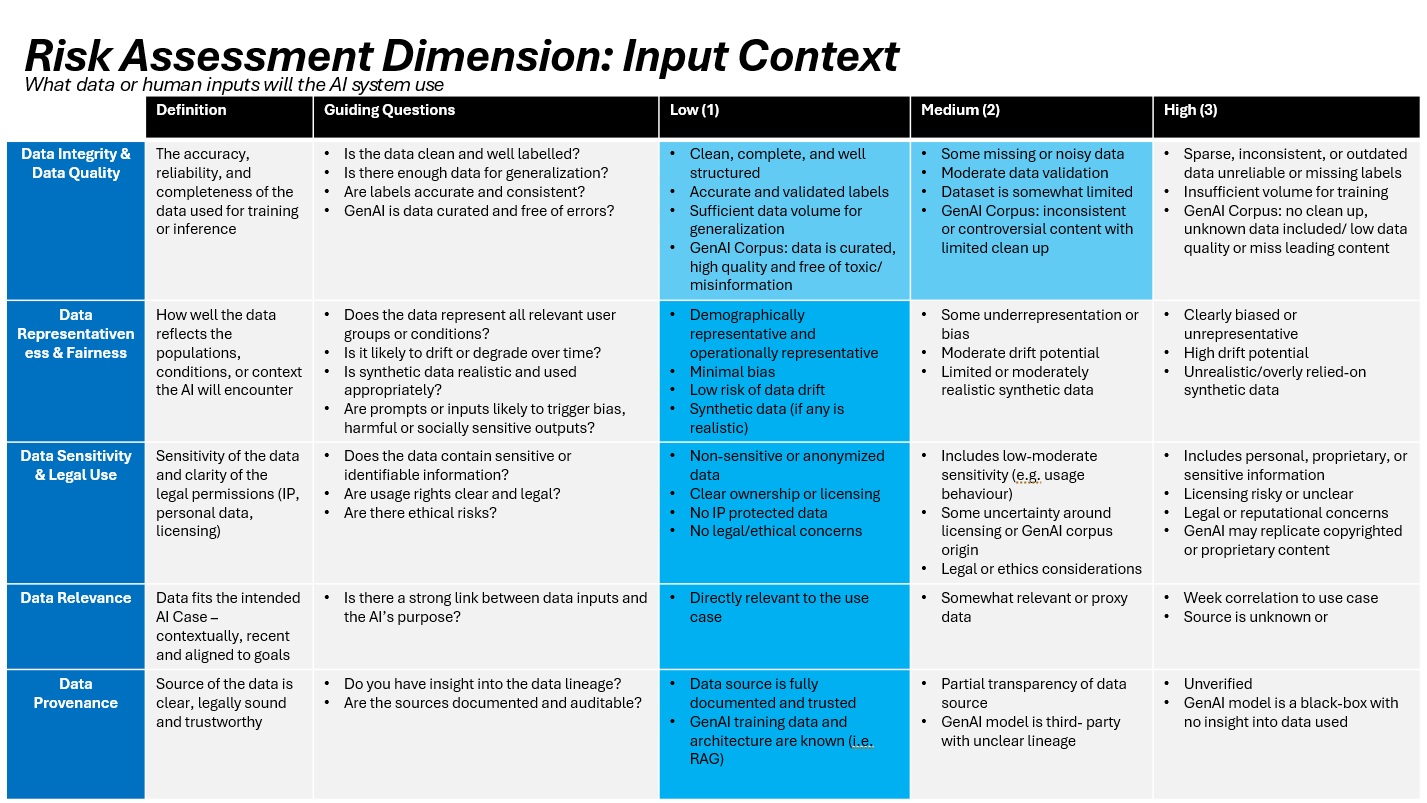
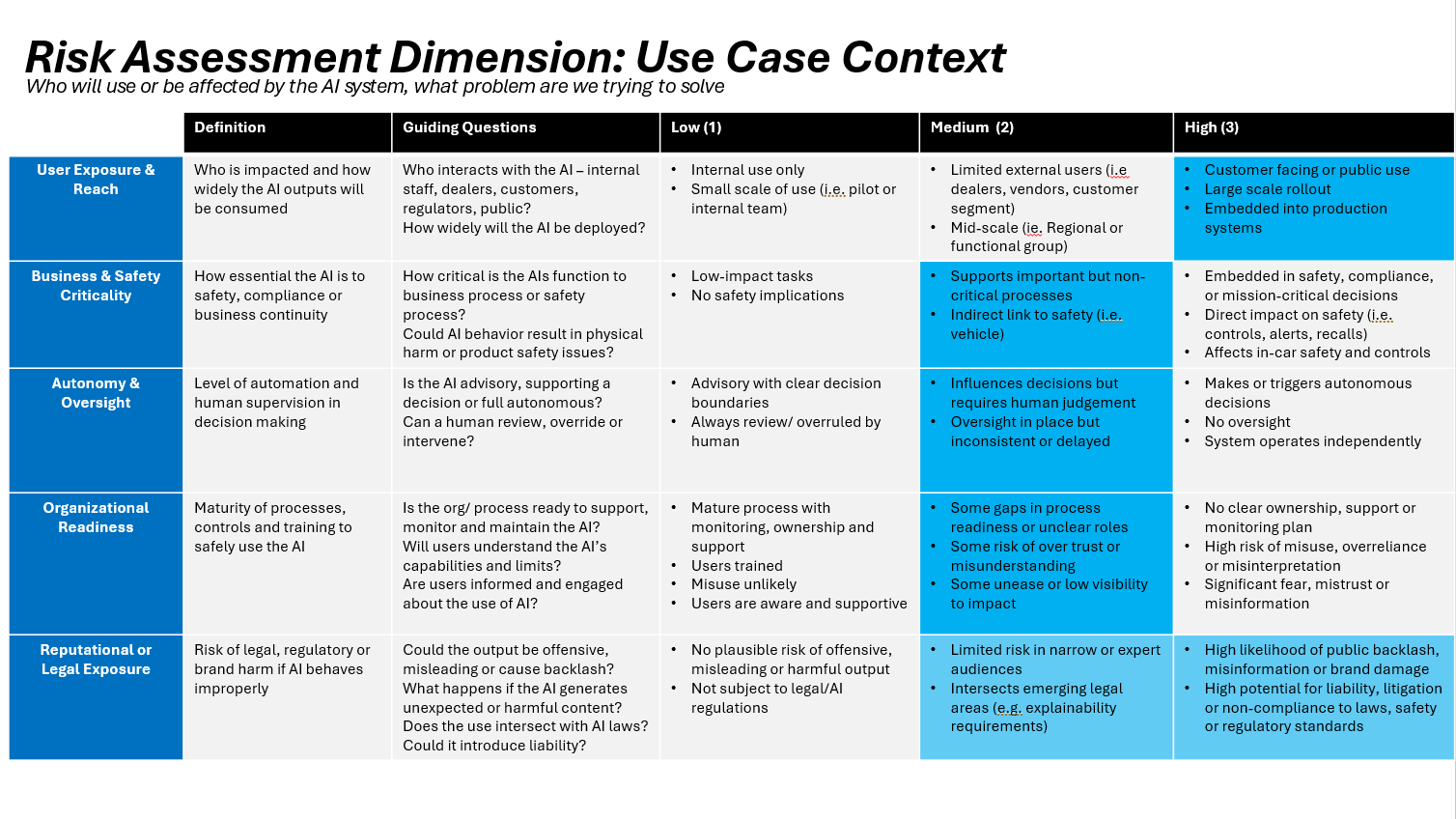
Amazon Aurora activity logs will be stored in CloudWatch.

Amazon DynamoDB supports activity logging for both control plane operations (like table creation) and data plane operations (like item-level changes) through AWS CloudTrail. By enabling data events in CloudTrail, you can log actions such as item reads, writes, and deletes, allowing for detailed auditing, security monitoring, and compliance tracking of all item activity within tables. These logs are delivered to an Amazon S3 bucket and can be further analyzed in Amazon CloudWatch Logs.

### Security Risk Assessment

AHM AI-specific risk assessment under development by AI Governance Working group / AI CoE was completed. It evaluates risk across 14 dimensions, including pre-launch and monitoring checklist evals.

Highlighted cells are how the Owner AI project maps to the framework:



### Security Risk Mitigations

* Design and implement fallback mechanisms for high-risk queries, especially those related tosafety and legal exposure. ​
* Develop and automate monitoring metrics for chatbot performance, accuracy, and userfeedback. ​
* Investigate AWS and Anthropic guardrails and sentiment analysis tools for enhancedoversight. ​
* Establish a process for regular review of flagged or negative feedback, with thresholds forescalation. ​
* Coordinate with security teams to test for prompt injection and jailbreaking vulnerabilities. ​
* Clarify and document organizational ownership for ongoing monitoring and incidentresponse. ​
* Review and update FAQ and dealer documentation to ensure accuracy and completeness. ​
* Consider adding call center escalation options for queries the chatbot cannot safely answer.

# Enterprise Services

The Data Power API Gateway Shared Service will be used in this project

## Current Available Services used in Project

| # | Standards ID # | Capacity | Description |
| --- | --- | --- | --- |
| 1 |  | TBD. All requests for the chatbot will be routed through the Data Power. See section 1 for the estimate number of users. | Data Power API Gateway will be used as the public endpoint of the CX AI chatbot. It will provide security (firewall), as well as identity token validation with the respective Identity Provider (i.e. CIAM, EntraID, or Custom MyGarage IdP created for anonymous users). Once validated, the request is forwarded to AWS AppSync (private endpoint) for processing. |

## Services to be created by Project

N/A

| # | Service | Description | Justification |
| --- | --- | --- | --- |
| 1 | N/A |  |  |

# Integration Architecture

## Integration Architecture Diagram

### Data Ingestion Architecture Diagram

The diagram below shows the high-level integration architecture for data ingestion used in the Owner AI chatbot application:



### MyGarage Application Integration

End users access the chatbot from the MyGarage Help Center (Acura Autos). The details of the integration between MyGarage and the Chatbot are described in section 5.2.1 and 5.2.2

### Hondaweb Application Integration

Internal users (who are authenticated by definition) access the chatbot’s admin pages using the Chatbot’s icon on MyApps.microsoft.com page

## Integration Points

There are three main integration points:

* 1. Data ingestion from Honda knowledge sources
  2. MyGarage application (External Users)
  3. Hondaweb (Internal Users)

### Data Ingestion Integration Points

At this time, there are three (3) known Data Ingestion integration points as described below. A potential forth integration point is with the TRACS application to report customer input related to complaints or reports of death or injury. This requirement is currently under development.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Integration Point No. | Description | Data Format | Source | Target | Frequency | Tool Used | Batch or online | Estimated Size |
| 1 | Import of Frequently Asked Questions from MyGarage Help Center into Bedrock Knowledgebase | HTML, JSON | MyGarage web site hosted on Salesforce | CX AI Knowledgebase S3 bucket | Once a day | Amazon AppFlow, or IBM ACE, or Bedrock Knowledgebase’s built-in connector to Salesforce (requires OAUTH credentials provided by SF) | Scheduled Batch | 1-2 MB |
| 2 | Import of Owners Manuals from Digital Asset Management (DAM) system into Bedrock Knowledgebase | PDF | On-prem File share | CX AI Knowledgebase S3 bucket having a folder structure such as (OM-> Make -> Model) | Once a day | Amazon DataSync | Batch file transfer | 4-5 MB |
| 3 (Future scope) | Import of How To Videos from Social Media Team’s SharePoint into Bedrock Knowledgebase | Mp4 | SharePoint | CX AI Knowledgebase S3 bucket | Once a day | Amazon AppFlow | Batch | TBD |

## Real-Time and Batch Integration

See section 7.2.1

## Internal Interfaces

See sections 7.2.1 and 7.2.3

## External Interfaces

None

# Application Architecture

## RAG Overview

The Owner AI application architecture is based on the popular Retrieval-Augmented Generation (RAG) application architecture. It is based on

optimizing the output of a Large Language Model (LLM), so it references an authoritative knowledge base outside of its training data sources before generating a response. LLMs are trained on vast volumes of data and use billions of parameters to generate original output for tasks like answering questions, translating languages, and completing sentences. RAG extends the already powerful capabilities of LLMs to specific domains or an organization's internal knowledge base, all without the need to retrain the model. It is a cost-effective approach to improving LLM output, so it remains relevant, accurate, and useful in various contexts.

## Owner AI Application Architecture

The Owner AI application is designed to satisfy two main use cases:

### End User Query and Response Sequence

1. User Query: The process begins with a user submitting a query to the application
2. Application to Amazon Bedrock Knowledge Base: The application sends the user's query to the Amazon Bedrock Knowledge Base for processing.
3. Amazon Bedrock Knowledge Base to Embedding Model: The Knowledge Base uses an embedding model (like Amazon Titan Embeddings) to convert the user's query into a vector representation (embedding)
4. Embedding Model to Vector Store: The generated query embedding is used to search for semantically similar documents within the configured vector store
5. Vector Store to Amazon Bedrock Knowledge Base: The vector store returns the retrieved relevant documents (or document chunks) to the Amazon Bedrock Knowledge Base.
6. Amazon Bedrock Knowledge Base to Foundation Model: The Knowledge Base augments the user's original query by incorporating the retrieved documents as context and passes this augmented prompt to the selected Foundation Model within Amazon Bedrock
7. Foundation Model to Application: The Foundation Model generates a response based on the augmented prompt and returns it to the application.
8. Application to User: The application displays the generated response to the user, potentially including citations to the original sources if configured

The following diagram depicts the typical user flow:



### Internal User flow

Internal users access the chatbot to monitor the application’s performance accuracy, and effectiveness. They are provided with a exclusive User Interface showing dashboards containing operational and guardrail metrics (see section 3.5 for more details on metrics)

### Data Ingestion Process

1. Frequently Asked Questions (source data) may be are retrieved from Salesforce via a connector between AWS and Salesforce (Amazon AppFlow connector or IBM ACE), or a new built-in feature of Amazon Bedrock Knolwdgebases that connects to Salesforce
2. Application Creates Knowledge Base: The application (or an administrator) creates a knowledge base in Amazon Bedrock, specifying the S3 bucket as the data source, the embedding model to use, and the vector store (i.e., OpenSearch Serverless, Aurora) for storing the embeddings.
3. Bedrock Knowledge Base Processes Data: Amazon Bedrock Knowledge Bases automatically handles the processing of the documents, including splitting them into smaller chunks and generating embeddings using the specified embedding model
4. Bedrock Knowledge Base to Vector Store: These embeddings are then stored in the configured vector store.

## Application Framework

The Owner AI chatbot will be a productionalized version of Open Source project provided by AWS. Please see [aws-samples/aws-genai-llm-chatbot: A modular and comprehensive solution to deploy a Multi-LLM and Multi-RAG powered chatbot (Amazon Bedrock, Anthropic, HuggingFace, OpenAI, Meta, AI21, Cohere, Mistral) using AWS CDK on AWS](https://github.com/aws-samples/aws-genai-llm-chatbot)

## Security

Both authenticated and unauthenticated users may use the chatbot. Application is available to the public through a button on the MyGarage website. Platform capabilities (i.e. AWS WAF) are used to detect and prevent malicious users from launching attacks.

See section 5.2 for more details on user flows.

Initially, there will only be IAM Roles created for end users:

* Unauthenticated (or guest) role
* Authenticated end user role

For the pilot implementation, both roles provide equivalent access to resources and features (i.e. same set of permissions in access policies). In the future releases of the chatbot, there may be features available to authenticated users only which would require access to data related customer profile and their ownership relationship (sales, vehicle details, service history, etc.) with Honda. In addition, there may be further segmentation of authenticated users (e.g. owners of Honda versus owners of Power Sports) with separate access to functions/tools and knowledgebases. The IAM service has a tight integration with Cognito which allows implementation of least privilege principle from the beginning to the end of a use case.

## User Interface (UI)

A React-based front-end is used for the chatbot. The UI will be hosted on S3 bucket and distributed via Content Delivery Network (Akamai).

The MyGarage UI will be modified to include a button that invokes the Owner AI chatbot when pressed.

The Chatbot UI will be displayed in a separate tab of the user’s browser.

## Portal

Hondaweb will be used as the entry point for the chatbot application’s admin view (or AWS console) where reporting and LLM metrics are accessed by internal users (business analysts and/or IT administrators)

## Business Services

N/A

## Service Registry

N/A

## Business Process Engine

N/A

## Application/Core Components

The solution is based on the merging of the code developed during the Prototype phase with the open-source AWS Samples code indicated here: [aws-samples/aws-genai-llm-chatbot: A modular and comprehensive solution to deploy a Multi-LLM and Multi-RAG powered chatbot (Amazon Bedrock, Anthropic, HuggingFace, OpenAI, Meta, AI21, Cohere, Mistral) using AWS CDK on AWS](https://github.com/aws-samples/aws-genai-llm-chatbot/tree/main)

Some of the AWS services used in this solution have been approved as an exception since they are not included as part of the enterprise standards. The table below describes then:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **AWS Service used in the solution (currently not an Enterprise Standard)** | **Alternative enterprise standard service** | **Gaps/ Differences** | **Cost Difference** |
| **1** | OpenSearch | None exists | N/A | N/A |
| **2** | Aurora PostgreSQL DB | RDS PostgreSQL | Aurora offers superior throughput, faster failovers, and better scalability for high-demand, enterprise-grade applications, whereas RDS provides broader PostgreSQL version support and is a good option for general-purpose or smaller workloads. | Aurora is generally less expensive than RDS |
| **3** | DynamoDB | No equivalent key/value database exists | DynamoDB is listed under “Emerging” category | N/A |
| **4** | AppSync | Amazon API Gateway (emerging standard), IBM DataPower | * AppSync offers GraphQL support (API Gateway does not support GraphQL natively). * IBM DataPower does not support tight integration with IAM and Cognito which is needed for this application. * IBM DataPower is being used in this solution as the public endpoint of the chatbot, providing security functions | AppSync uses pay as you go subscription, DataPower requires a license |
| **5** | AppFlow | IBM ACE | Both offer equivalent features for connectivity, but IBM ACE will not have tight integration with other Amazon services used in the solution including IAM, S3, KMS, CloudTrail, CloudWatch, PrivateLink among others. | AppFlow uses pay as you go subscription, DataPower requires a license |
| **6** | Bedrock | None. Bedrock is listed as an “emerging” standard | N/A | N/A |

More details about the proposed services and components is given below:

### Amazon Cognito

* User Pool
* User Pool Client
  + Attach an existing Cognito Domain for Federation, adds Permission for User Pool to leverage existing Cognito Domain / Creates OIDC or SAML Identity Provider in Cognito
* User Groups

The Owner AI chatbot will have three User Groups defined in Cognito User Pool (extendable to more groups):

* 1. Guest (or Unauthenticated users)
  2. Authenticated user
  3. Internal User

Each User Group will be assigned an IAM role, each having their own Trust Policy and Permission Policy.

Admin and workspace manager roles allow applications (configurable chatbots) and workspace management retrospectively.

### AppSync

The project relies on AWS AppSync which creates serverless GraphQL and Pub/Sub APIs that simplify application development through a single endpoint to securely query, update, or publish data. One of the key advantages is GraphQL subscriptions which is used to receive the chatbot responses in real time using streaming.

The GraphQL Schema defining the possible operations is available on the AWS console Once deployed, AWS AppSync provides a Querying tool which can be used to explore and test the APIs.

### Lambda

AppSync GraphQL API Handler: Function responsible for handling all inbound GraphQL API requests and processing them.

### Amazon Simple Queue Service (SQS)

* Outgoing Messages Queue: Queue for managing outgoing messages
* Outgoing Messages Dead-Letter Queue (DLQ): Queue for handling Dead-Letter messages from the Outgoing Messages Queue

### Amazon Simple Notification Service (SNS)

Messages Topic: Topic for managing message bus

### DynamoDB

**Session Table**: Houses user chat sessions with chatbot

**WorkSpaces Table**: Table that houses data about the RAG Workspaces. Does not house the actual RAG data, but rather the metadata/configuration data for the Workspace.

**Documents Table**: Table that houses the data about documents that have been or need to be ingested into RAG workspaces (websites, PDFs, text files, RSS feeds, etc.)

## Persistence

### Amazon Simple Storage Service (S3)

* File Uploads Bucket: Files uploaded into a RAG workspaces
* Processing Bucket: Bucket to house files being processed for a RAG workspace

## Integration - ESB

IBM ACE may be used to connect AWS to Salesforce. A POC was conducted to confirm AppFlow’s ability to retrieve data from SharePoint, along with the associated metadata. However, Amazon AppFlow provides a less costly alternatives.

## Integration - ETL

N/A

## Database

### Amazon Aurora PostgreSQL

RAG Workspaces: users can create PostgreSQL w/ pgvector data stores for Retrieval Augmented Generation (RAG) workspaces. Must be enabled

A caching mechanism may be used to limit accessing the LLMs for common queries submitted by the users

## Scheduler

CloudWatch will be used to schedule data ingestion pulls from data sources. See section 7.2

## Operational Reporting Tool

CloudWatch Dashboard will be used to display key Bedrock metrics. Specific reporting requirements are still in development

## Analytical Reporting Tool

Bedrock Evaluations provides metrics related to the Owner AI application’s performance (see section 3.5)

## Fax

N/A

## Operating System

All services used in this application are serverless.

## Backup / Recovery

Backup and recovery processes used for the Owner AI application take advantage of the high resiliency, reliability, and availability features of the AWS Cloud platform. Specifically:

1. Knowledgebases are stored in S3 buckets which are highly resilient storage technology
2. Vector Database containing vectorized data which is the output of the embeddings model is replicated in another Availability Zone and backed up in another region.

# Data Architecture

## Conceptual Data Model (J0/J1)

The primary data source for the Owner AI application is data that is extracted from the MyGarage website’s Help Center section containing Frequently Asked Questions for Acura Vehicles which are stored in the form of free form text, tables, images, and links to web pages.

The relational database used to store vectorized data as the output of the embeddings model from the data ingestion process is designed by the Bedrock service and does not include the traditional textual structured data.

In order to improve accuracy of the output of the LLM, metadata must be added to knowledgebase documents. Metadata may be imported to Bedrock (see [https://docs.aws.amazon.com/bedrock/latest/userguide/kb-metadata.html](https://jpn01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.aws.amazon.com%2Fbedrock%2Flatest%2Fuserguide%2Fkb-metadata.html&data=05%7C02%7C%7C9d97565f8e8e4b5ac7ff08ddd4fa8904%7Ccc138ffb9c4d433bb211d4549ae62e8b%7C0%7C0%7C638900894691752052%7CUnknown%7CTWFpbGZsb3d8eyJFbXB0eU1hcGkiOnRydWUsIlYiOiIwLjAuMDAwMCIsIlAiOiJXaW4zMiIsIkFOIjoiTWFpbCIsIldUIjoyfQ%3D%3D%7C0%7C%7C%7C&sdata=XdcD1bv0YMT4vfZKaSW3T9mq6iSccAS0yLqTwxge1b0%3D&reserved=0)) For example, the following metadata model can be incorporated into the knowledgebase:

* Document type (Owners Manual, FAQ, Video.)
* Language
* Year
* Make/Brand
* Vehicle model (or year/model/trim triplet)
* Original url to be used for citations
* Dates, e.g. Created, Last Iimported, etc.

The FAQ data is stored in Knowledge Article Object in Salesforce. A custom field will be added to this object which will be used to include the required meta data. This field must be examined by the LLM to improve accuracy of response that are based on FAQs.

## Logical Data Model (J1)

## Physical Data Model (J1)

# Revision History

| **Date** | **Author** | **Version** | **Initiating Project/Enhancement** | **Sections Impacted** | **Change Summary** |
| --- | --- | --- | --- | --- | --- |
| 9/8/2025 | Kamyar Varzandeh | 1.0 | PR10956: 102Ki ADR CX AI | All | Submitted to the ARB |
| 10/1/2025 | Kamyar Varzandeh | 1.1 | PR10956: 102Ki ADR CX AI | 3.3, 5.2 | Incorporating DataPower API Gateway |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Appendix A – References

| **Term or Document Name** | **Description** |
| --- | --- |
| [ISD Glossary](http://hondaweb2.ahm.am.honda.com/wiki/index.php?title=Category:ISD_Glossary) | Glossary of ISD terminology. |
| Enhancement Requiring Funding Flowchart (MSR-FC-0002) | Flowchart depicting the process for Enhancements which require funding. |
|  |  |
|  |  |

Appendix B – Architectural Decisions

|  |  |
| --- | --- |
| **Detailed Architectural Decision** | |
| Decision ID | AD-01 |
| Issue | Which API Gateway to be used: API Gateway or AppSync. Neither one is on the EA Standards list. AppSync is used in the reference architecture proposed by AWS ProServe. API Gateway was used as part of the Owner AI prototype built with the help of AWS Innovation center |
| Business Justification |  |
| **Alternatives** | |
| Option 1 Description | Amazon AppSync |
| Pros | * Managed GraphQL service, allows for fewer API calls (compared to REST), and the returning payload is more likely to be optimized (i.e. not over or under fetching) * It is more modern, allowing for extension of the current chatbot architecture to support more features, knowledgebases, tools, agents, etc. * It is part of the reference architecture proposed by AWS ProServe. There is running code that is proven which should accelerate development timeframe and reduce cost. * AWS WAF (Web Application Firewall) can be integrated with AppSync to create web ACLs (Access Control Lists). These web ACLs can include rate-based rules to throttle requests based on criteria like IP address, preventing single sources from overwhelming the API. * GraphQL Execution Limits: AppSync allows configuring limits on GraphQL query complexity, depth, and the maximum number of resolvers per query. This helps prevent overly complex or deeply nested queries that could consume excessive resources or lead to denial-of-service (DoS) scenarios. * Robust Authorization Mechanisms: Using stronger authorization types like AWS IAM, Amazon Cognito User Pools, or OpenID Connect (OIDC) helps prevent unauthorized access and ensures only legitimate users can interact with the API, reducing the surface area for abuse. |
| Cons | * Not as widely understood or adopted by Honda IT. It is currently listed under “Non Standard” in EA’s ‘AWS Cloud Service Catalog” * Cost per invocation is higher than that of API Gateway (but that may be offset by fewer calls to achieve the same objective) |
| Option 2 Description | Amazon Api Gateway |
| Pros | * More established API Gateway service supporting REST * Supports a rich set of features including built-in API throttling which is needed for public APIs * Cost per API is less than AppSync, but UI may may need to invoke more APIs compared to GraphQL APIs * On the Honda’s “Emerging technology” list (however, it has been widely in use since 2015) |
| Cons | * Applications may experience performance issues due to over-fetching or under-fetching of data needed by the UI * Not part of the reference architecture which is the basis of the chatbot solution proposed by AWS ProServe team |
| Decision | **Use Amazon AppSync** |
| Justification | Best technical fit for the chatbot application. The accelerator code provided by AWS already uses AppSync |
| Pre-Requisites | Production support team must obtain required skill set for AppSync |
| Assumptions | None |
| Implications | Requires ARB approval using an Exception condition, or to be added to the list of standard AWS services |

|  |  |
| --- | --- |
| **Detailed Architectural Decision** | |
| Decision ID | AD-02 |
| Issue | How to handle unauthenticated user traffic from MyGarage website to Owner ID chatbot |
| Business Justification | Need to allow both authenticated and unauthenticated users to use the Owner ID chatbot |
| **Alternatives** | |
| Option 1 Description | Use Static link to the chatbot on the MyGarage Help Center, Acura Autos chatbot launch page |
| Pros | * Simplest approach to implement * There are other defenses against bad actors using Akamai and AWS security services |
| Cons | * Not as secure as other options utilizing a dynamic link containing a token |
| Option 2 Description | Generate a Token on MyGarage and include it in the chatbot launch button |
| Pros | * More secure than option 1 * Ensures request is coming from an unauthenticated user from MyGarage |
| Cons | * Complex token generation on MyGarage – more time and cost to build and test may impact schedule * Additional token validation step required on Amazon Cognito * Cost to value ratio is high. Chatbot is serving public information anyway |
| Option 3 Description | Generate a Token on AWS and include it in the chatbot launch button |
| Pros | * More secure than option 1 * Ensures request is coming from an unauthenticated user from MyGarage |
| Cons | * Requires using Cognito API – more time and cost to build and test may impact schedule * Cognito is being used as an Identity Provider. CIAM is the Enterprise standard for IdP service. * Additional token validation step required on Amazon Cognito * Cost to value ratio is high. Chatbot is serving public information anyway |
| Decision | **Use Option 2 – Dynamic Link** |
| Justification | Superior security and protection against bad actors through usage of dynamic links that expire. Also, this options fits in with the authenticated user flows all leveraging identity tokens |
| Pre-Requisites | Akamai and AWS security services must be configured to guard against DDoS and support CORS |
| Assumptions | None. See here for more details. |
| Implications | None |

Appendix C – Non-Functional Requirements



Appendix D – Architectural Decisions Template

|  |  |
| --- | --- |
| Decision ID |  |
| Category | DB |
| Issue | Cannot perform join or sophisticated SQL with SharePoint JSOM making it difficult to determine participation in the review in a performant way. |
| Decision | Add the 6 fields to the request form database to simplify the query. |
| Assumptions | The reviewer assignment will only be done once per request form.  The reading of the participation will be done many times per request form.  The read operations should be optimized. |
| Implications | Must save the reviewer assignments in both the request form and the reviewer assignments lists. |
| Related Decisions |  |
| Related Requirements |  |

Table 1 - Short Form

|  |  |
| --- | --- |
| **Detailed Architectural Decision** | |
| Decision ID |  |
| Issue |  |
| Business Justification |  |
| **Alternatives** | |
| Option 1 Description |  |
| Pros |  |
| Cons |  |
| Option 2 Description |  |
| Pros |  |
| Cons |  |
| Decision |  |
| Justification |  |
| Pre-Requisites |  |
| Assumptions |  |
| Implications |  |
| **Security** | |
| User Provisioning |  |
| User Authentication and Authorization |  |
| **Data Security** | |
| Production Data Security |  |
| Electronic Signatures and Receipts |  |
| Data Masking Solutions |  |
| **Monitoring and Reporting (Optional)** | |
|  |  |