# A generative AI powered self-service assistant for contact centers

# Generative AI in the contact center

## Many organizations operate contact centers to service requests from their customers

- Staffing for live agents in contact centers can be a significant contributor to an organization's operating expense
- In many cases, live agents spend time responding to customer issues that can easily be solved by referencing available information on the company's website or in knowledge base documents

## Generative AI can help reduce operating expenses, while streamlining and improving the customer experience

- LLMs on Amazon Bedrock, including Anthropic's Claude Haiku model, can use information in **Knowledge Bases for Amazon Bedrock** to answer questions within a few seconds, making it a viable solution for both text messaging interactions as well as voice calls
- AWS customers using Amazon Connect can **easily integrate with LLMs on Amazon Bedrock via Amazon Lex**

## AWS Generative AI Innovation Center solution

- The Contact Center Generative AI Assistant Solution provides an easy to deploy, easy to operate solution that can reduce call volumes requiring a live agent, by **answering customer questions based on content available in a knowledge base**
- The solution includes **automated testing, in-depth conversation analytics, hallucination detection and prevention, and monitoring**

## Solution benefits

- Reduction in operating expense, by deflecting inbound calls from human agents to automated agents
  - Costs for Bedrock-powered automated agents range from **$0.03 and $0.06 per call**
- Improved and streamlined customer experience
  - Customers can get **questions answered and issues resolved immediately**, without having to wait for a live agent
  - The generative AI model can be instructed to interact with customers in a **consistent, friendly, and informative manner that is always on-brand**, and stays within specified guardrails

# Building a Generative AI Contact Center Solution for DoorDash Using Amazon Bedrock

## Challenges

DoorDash receives hundreds of thousands of calls to its support center each day and wanted to help Dashers get answers to routine questions efficiently while freeing up live agents to handle more complex issues.

## Solutions

Using Anthropic's Claude Haiku on Amazon Bedrock, DoorDash achieved a response latency of 2.5 seconds or less. By handling common inquiries with generative AI, DoorDash has improved self-service workflows and reduced issue resolution speeds.

## Results

- 50x increase in testing capacity
- 50% reduction in response latency
- 2.5 seconds or less response latency
- 50% reduction in development time

> **Using AWS, we've built a solution that gives Dashers reliable access to the information they need, when they need it.**
>
> **Chaitanya Hari**
> Contact Center Product Lead, DoorDash

*The AWS case study is at https://aws.amazon.com/solutions/case-studies/doordash-bedrock-case-study*

CUSTOMER PROFILE

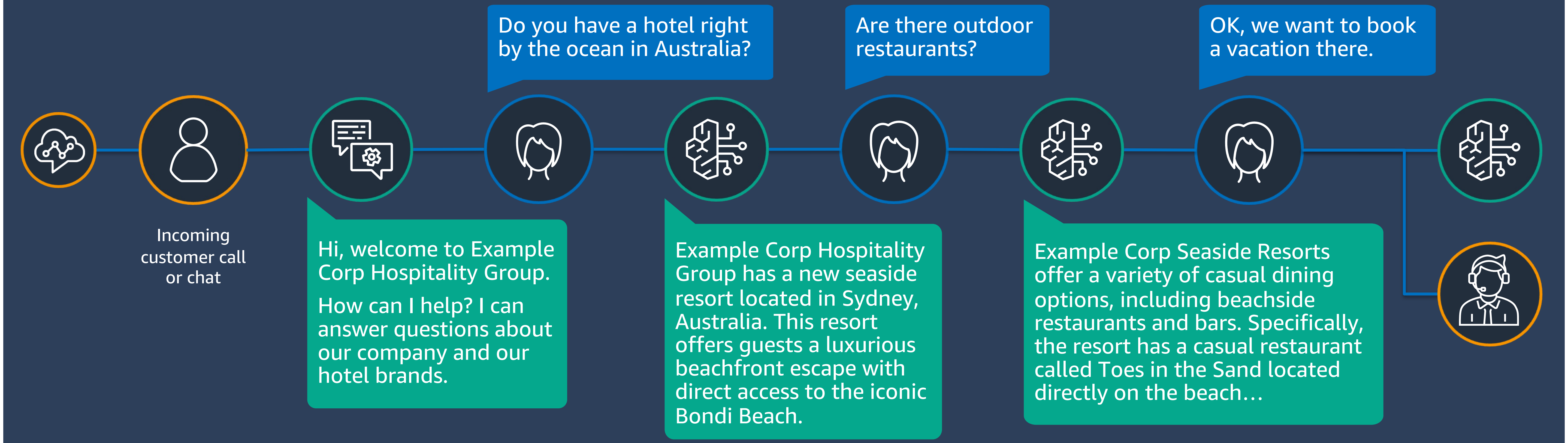
DOORDASH

INDUSTRY
**Technology**

REGION
**United States**

DoorDash is a local commerce platform dedicated to helping Merchants thrive in the convenience economy, giving consumers access to more of their communities, and providing work that empowers.

# Customer experience *(using a fictional hotel chain example)*



Do you have a hotel right by the ocean in Australia?

Are there outdoor restaurants?

OK, we want to book a vacation there.

Incoming customer call or chat

Hi, welcome to Example Corp Hospitality Group.

How can I help? I can answer questions about our company and our hotel brands.

Example Corp Hospitality Group has a new seaside resort located in Sydney, Australia. This resort offers guests a luxurious beachfront escape with direct access to the iconic Bondi Beach.

Example Corp Seaside Resorts offer a variety of casual dining options, including beachside restaurants and bars. Specifically, the resort has a casual restaurant called Toes in the Sand located directly on the beach…

## NATURAL

Provide a natural language experience with Amazon Connect and Lex

## AUTOMATED

Answer customer questions automatically, with information from Bedrock knowledge bases

## CONVERSATIONAL

Let the customer drive the conversation, while maintaining conversational context

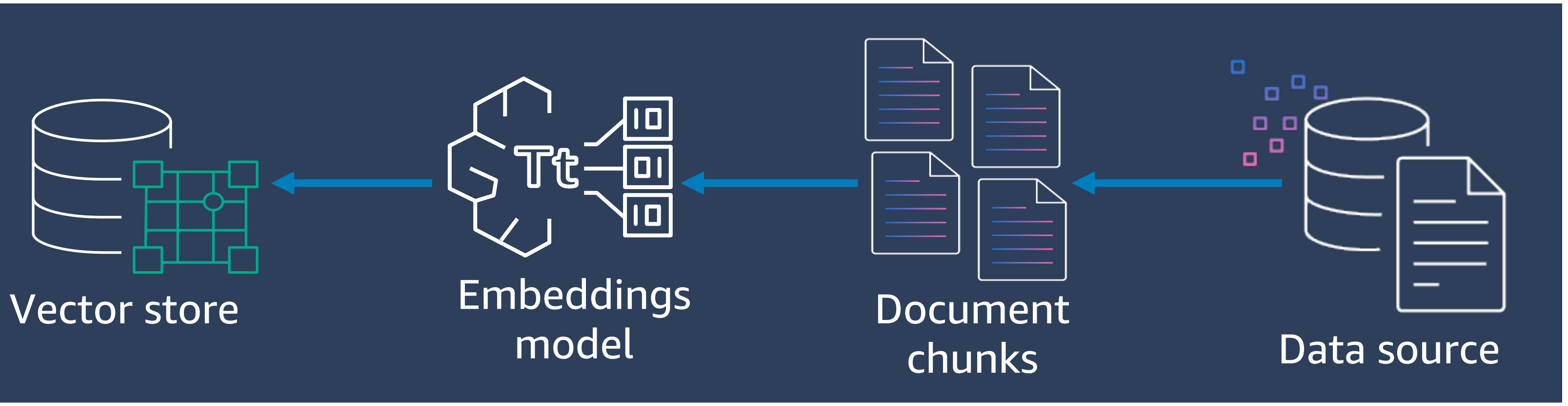
## INTEGRATED

Hand conversatons off to live agents, or automate further with Bedrock Agents

# Retrieval Augmented Generation (or, in-context learning)

Data Ingestion

*Content to be used to answer questions is ingested into a knowledge base. The ingestion process “chunks” the content, generates numerical vector embeddings for each chunk using an embeddings model, and stores the resulting vectors in a vector store.*



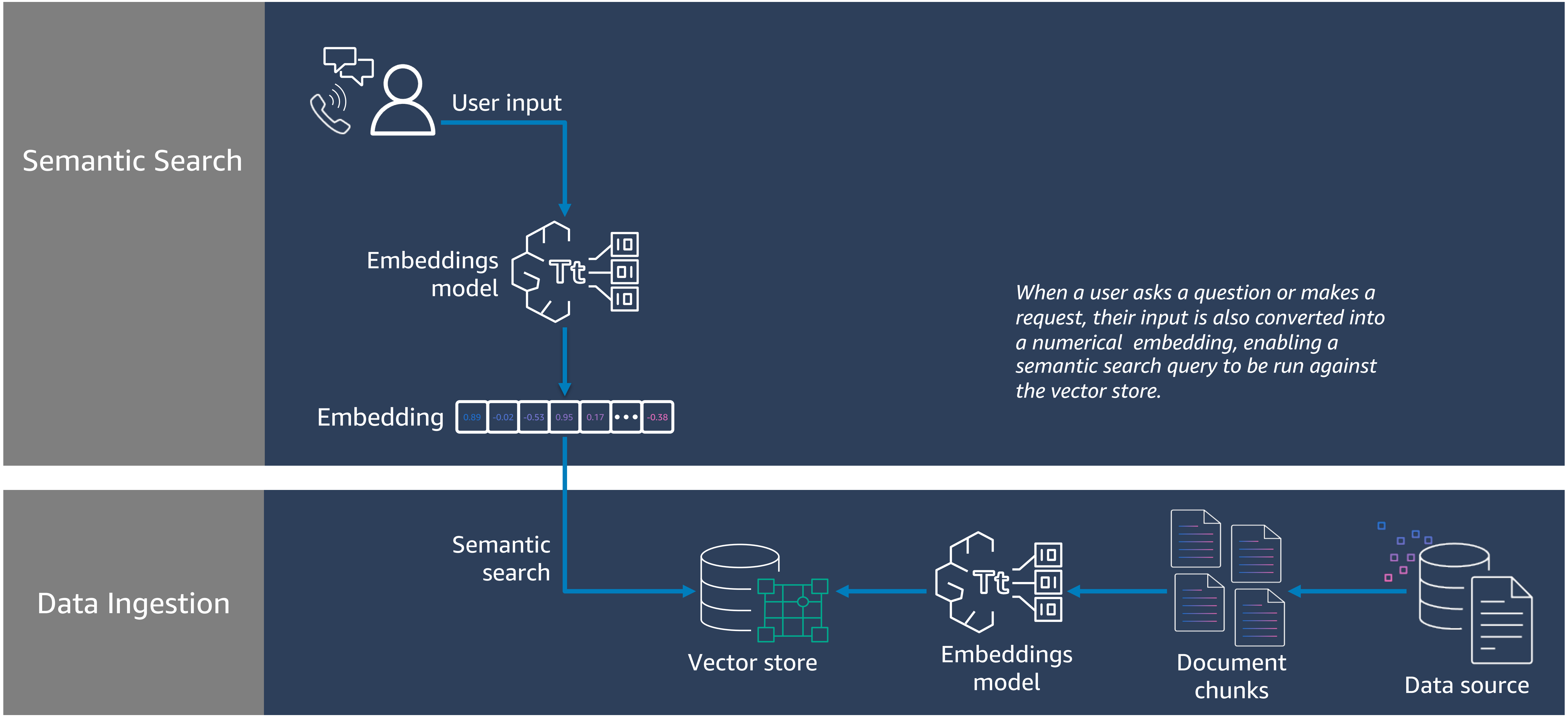
Vector store

Embeddings model

Document chunks

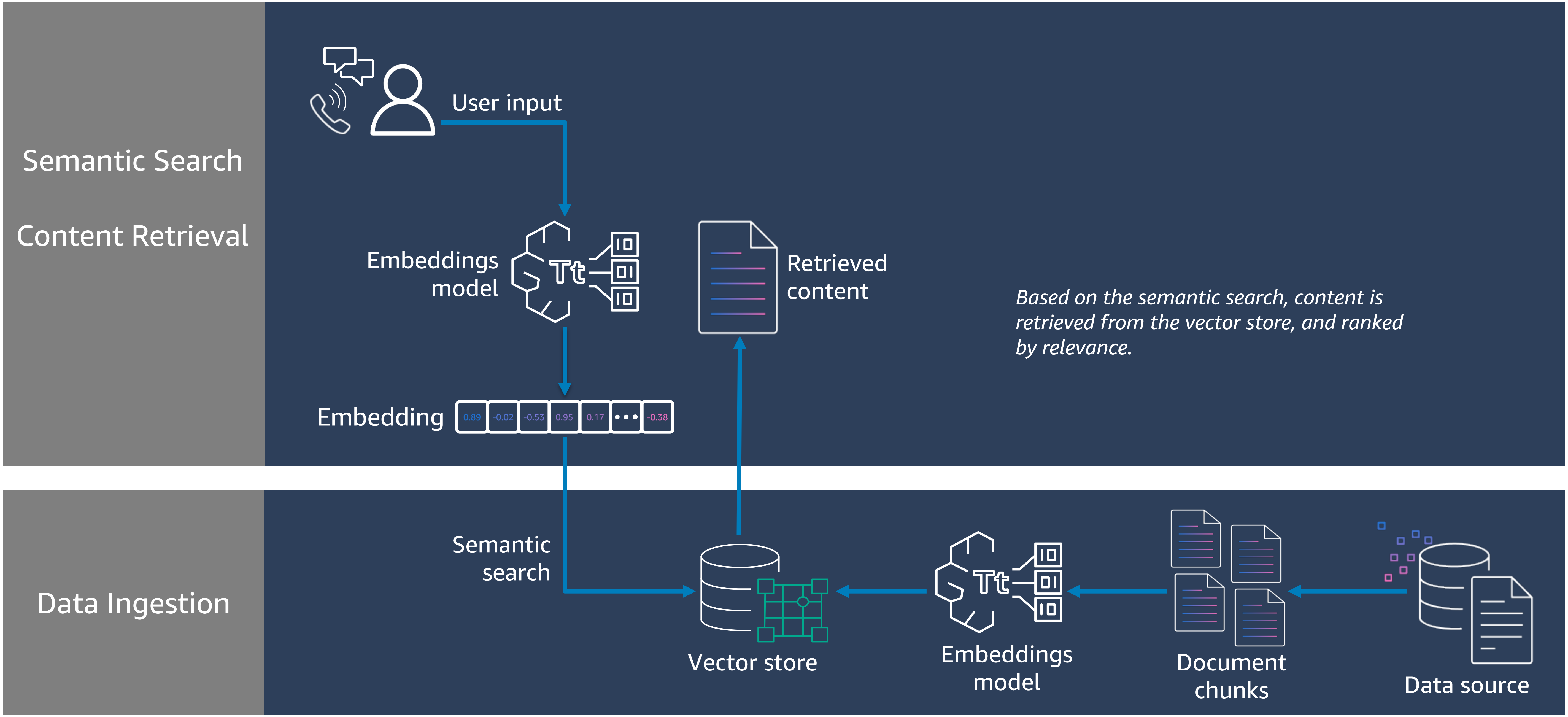
Data source

# Retrieval Augmented Generation (or, in-context learning)



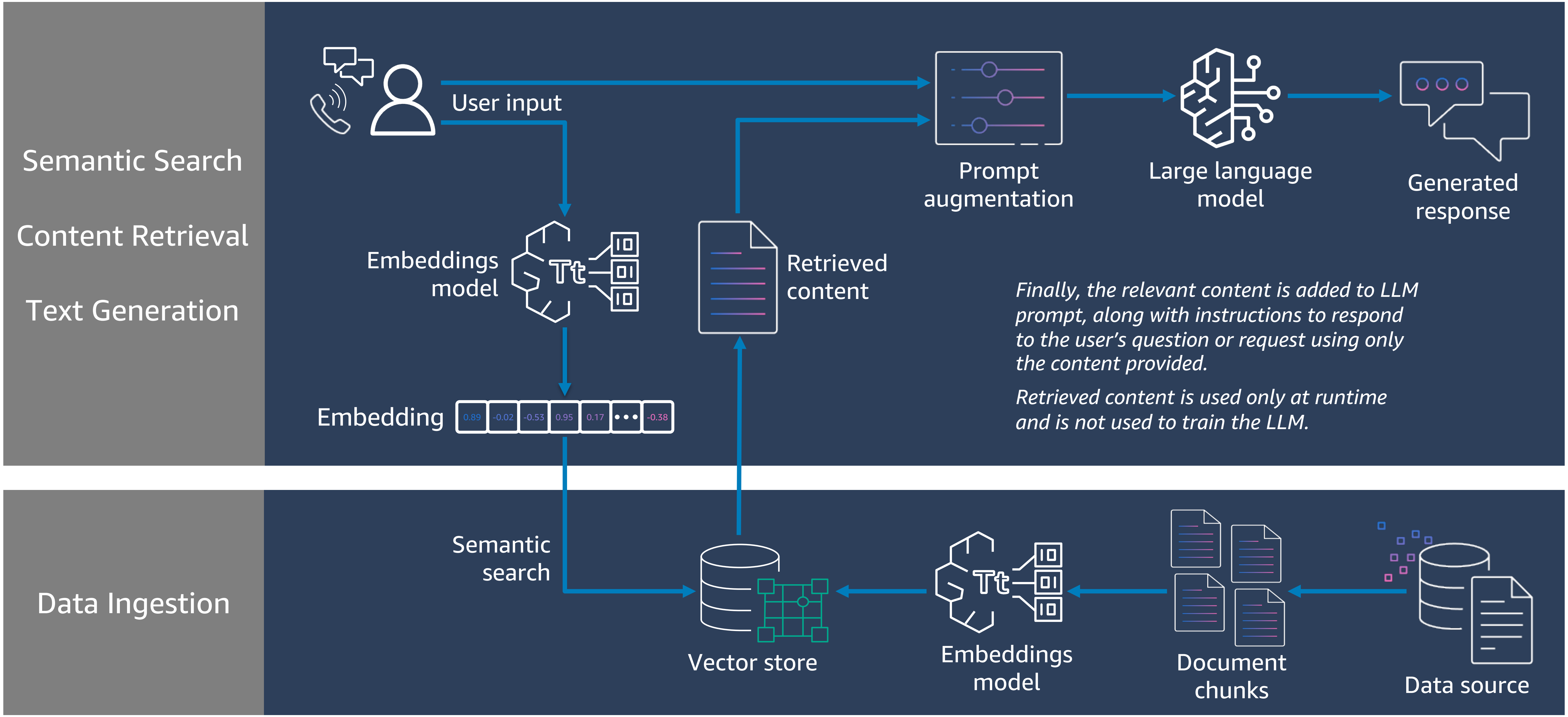
*When a user asks a question or makes a request, their input is also converted into a numerical embedding, enabling a semantic search query to be run against the vector store.*

# Retrieval Augmented Generation (or, in-context learning)



*Based on the semantic search, content is retrieved from the vector store, and ranked by relevance.*

# Retrieval Augmented Generation (or, in-context learning)



Semantic Search

Content Retrieval

Text Generation

User input

Embeddings model

Embedding

0.89 | -0.02 | -0.53 | 0.95 | 0.17 | ••• | -0.38

Retrieved content

Prompt augmentation

Large language model

Generated response

*Finally, the relevant content is added to LLM prompt, along with instructions to respond to the user's question or request using only the content provided.*

*Retrieved content is used only at runtime and is not used to train the LLM.*

Data Ingestion

Semantic search

Vector store

Embeddings model

Document chunks

Data source

# Knowledge Bases for Amazon Bedrock

Fully-managed native support for retrieval augmented generation (RAG)

- Fully managed support for end-to-end RAG workflow
- Securely connect LLMs and agents to data sources
- Automatically converts text documents into embedding vectors
- Stores embeddings in your vector database
- Retrieves content based on semantic search and augments LLM prompts
- Provides source attribution

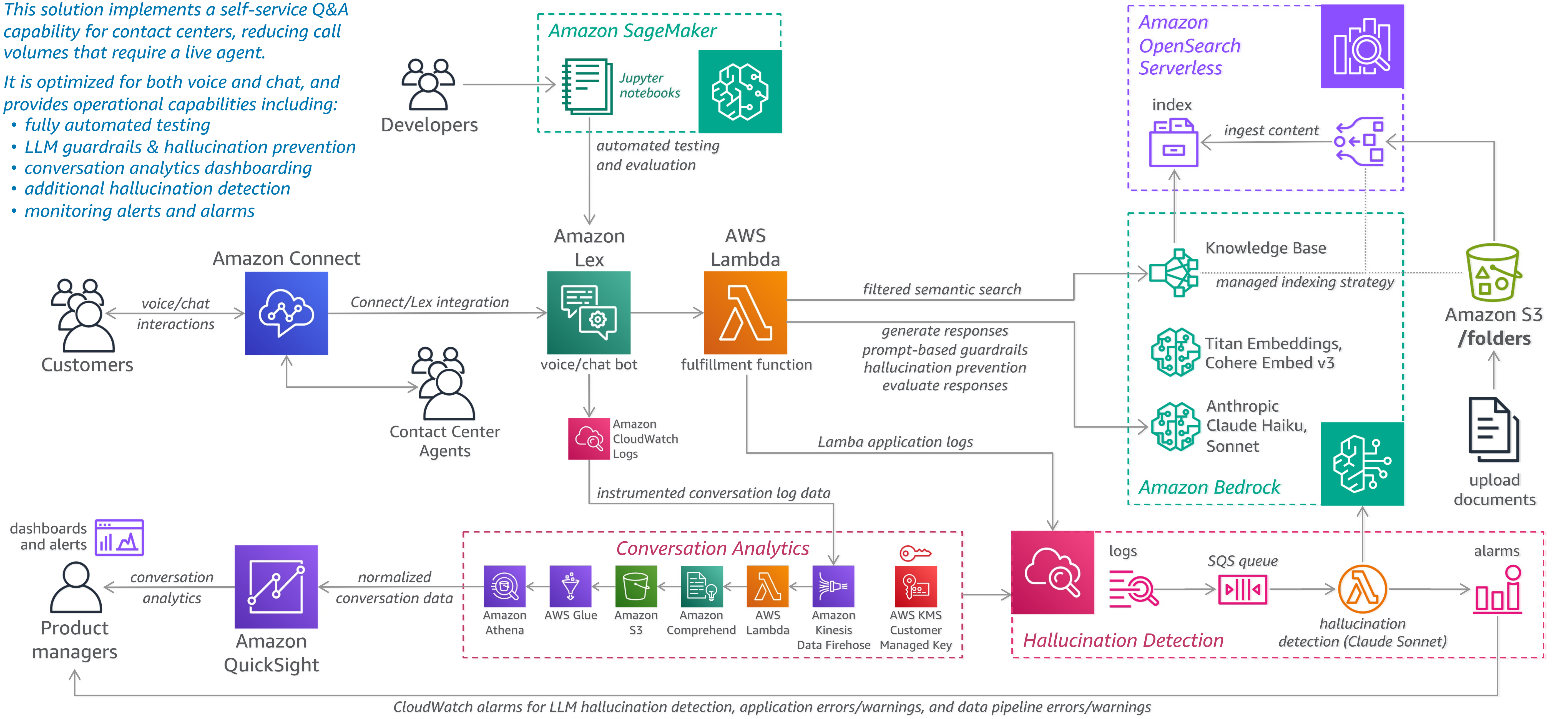
# Solution Architecture

# Contact center RAG solution architecture

*This solution implements a self-service Q&A capability for contact centers, reducing call volumes that require a live agent.*

*It is optimized for both voice and chat, and provides operational capabilities including:*

- *fully automated testing*
- *LLM guardrails & hallucination prevention*
- *conversation analytics dashboarding*
- *additional hallucination detection*
- *monitoring alerts and alarms*



Developers

Amazon SageMaker

Jupyter notebooks

automated testing and evaluation

Amazon OpenSearch Serverless

index

ingest content

Customers

voice/chat interactions

Amazon Connect

Connect/Lex integration

Contact Center Agents

Amazon Lex

voice/chat bot

AWS Lambda

fulfillment function

filtered semantic search

generate responses
prompt-based guardrails
hallucination prevention
evaluate responses

Knowledge Base

managed indexing strategy

Titan Embeddings, Cohere Embed v3

Anthropic Claude Haiku, Sonnet

Amazon Bedrock

Amazon S3 /folders

upload documents

Amazon CloudWatch Logs

Lamba application logs

instrumented conversation log data

dashboards and alerts

Product managers

conversation analytics

Amazon QuickSight

normalized conversation data

Conversation Analytics

Amazon Athena

AWS Glue

Amazon S3

Amazon Comprehend

AWS Lambda

Amazon Kinesis Data Firehose

AWS KMS Customer Managed Key

logs

SQS queue

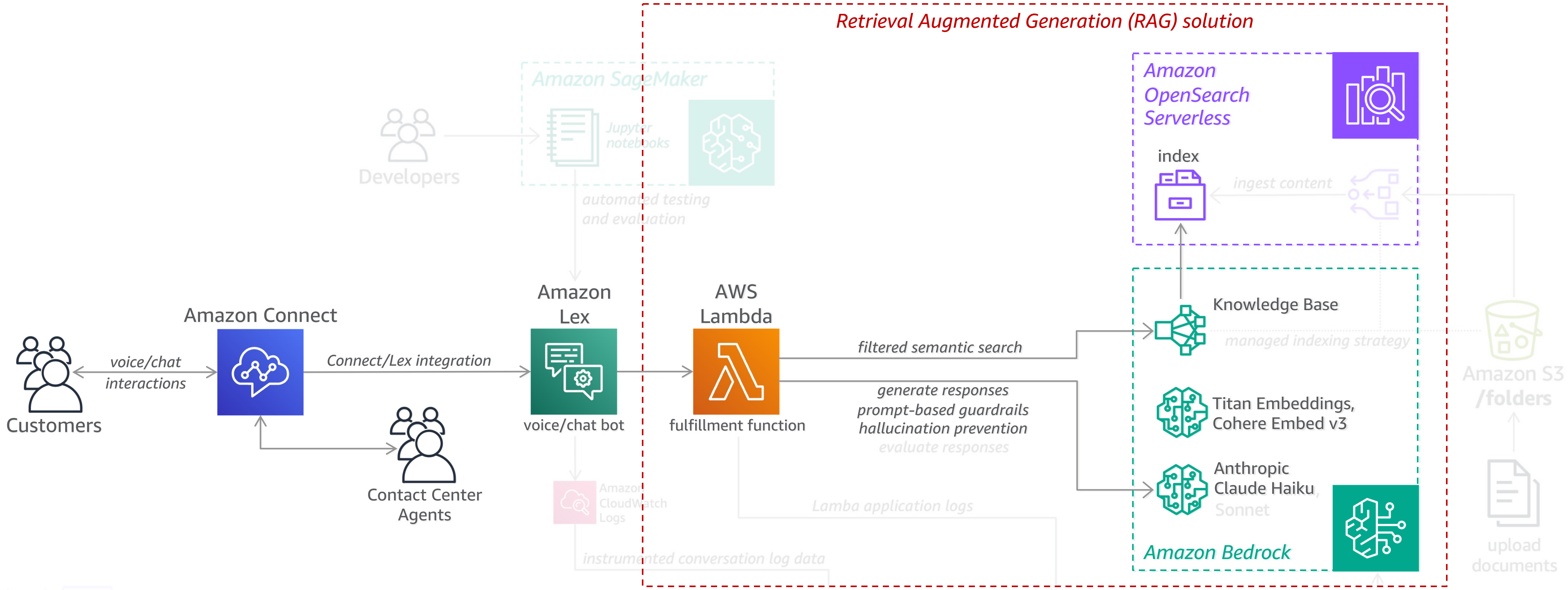
alarms

hallucination detection (Claude Sonnet)

Hallucination Detection

CloudWatch alarms for LLM hallucination detection, application errors/warnings, and data pipeline errors/warnings

# Core RAG solution

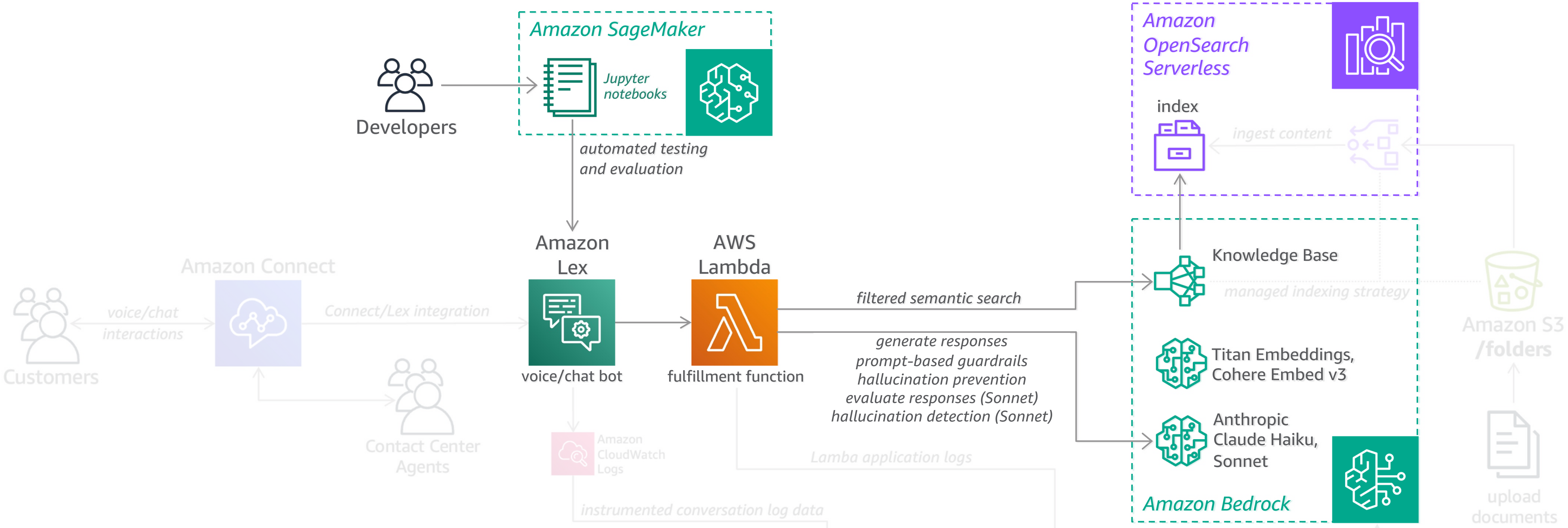


*When a customer calls in or initiates a chat session, they're greeted by an Amazon Lex bot that asks them: "how can we help you today?"  Based on the customer's response, Lex either routes the customer to the appropriate Amazon Connect live agent queue, or initiates the RAG solution to answer their questions.*

*The RAG solution uses Anthropic's Claude 3 Haiku LLM, which answers questions based on content available in an Amazon Bedrock Knowledge Base. The content resides in a single collection in Amazon OpenSearch Serverless. To zero in on the most relevant content, Knowledge Base uses a hybrid query that employs a metadata prefilter based on customizable attributes (such as customer type/persona, the specific customer intent/question type, and country), along with a semantic search based on the customer's specific question and conversation history (cosine similarity/k-nearest neighbors) .*

*The solution makes it easy to use any LLM available on Amazon Bedock, and any type of content repository supported by Bedrock Knowledge Bases. LLM prompts are provided for Claude Haiku and managed in an Amazon DynamoDB table, and can be customized as needed by customer persona, customer intent, country, etc. Guardrails are employed to redirect inappropriate or off-brand topics.*

# Automated testing



*The automated testing component consists of a multi-threaded test script in a Jupyter notebook. The test script can execute 100s of test cases per minute, and can be incorporated into the test stage of a CI/CD deployment pipeline.*

*Sets of test cases are captured in an Excel workbook or .CSV file, and each test case can contain one or more steps (i.e., for multi-turn conversations, follow-on questions, etc). Each step of a test case includes a user input, a correct "ground truth" answer, and any configuration parameters (session attributes) needed for the test case. The test script sends the test case to the RAG solution, and then uses Anthropic Claude Sonnet to compare the response from the RAG solution to the ground truth answer. If the RAG solution's answer has the same semantic meaning as the ground truth answer, the test case passes. If the RAG solution's answer is semantically different than the ground truth answer, or is incomplete, the test case fails. Either way, a detailed explanation is provided by the LLM as to why the test case passed or failed.*

*In addition, the RAG solution's answer is evaluated by Claude Sonnet against the RAG content from the knowledge base, to detect any hallucinations (information provided in the answer that is not present in the knowledge base. Test results and hallucination detection results are saved to an output Excel workbook, and also to the conversation analytics dashboard.*

# Conversation analytics

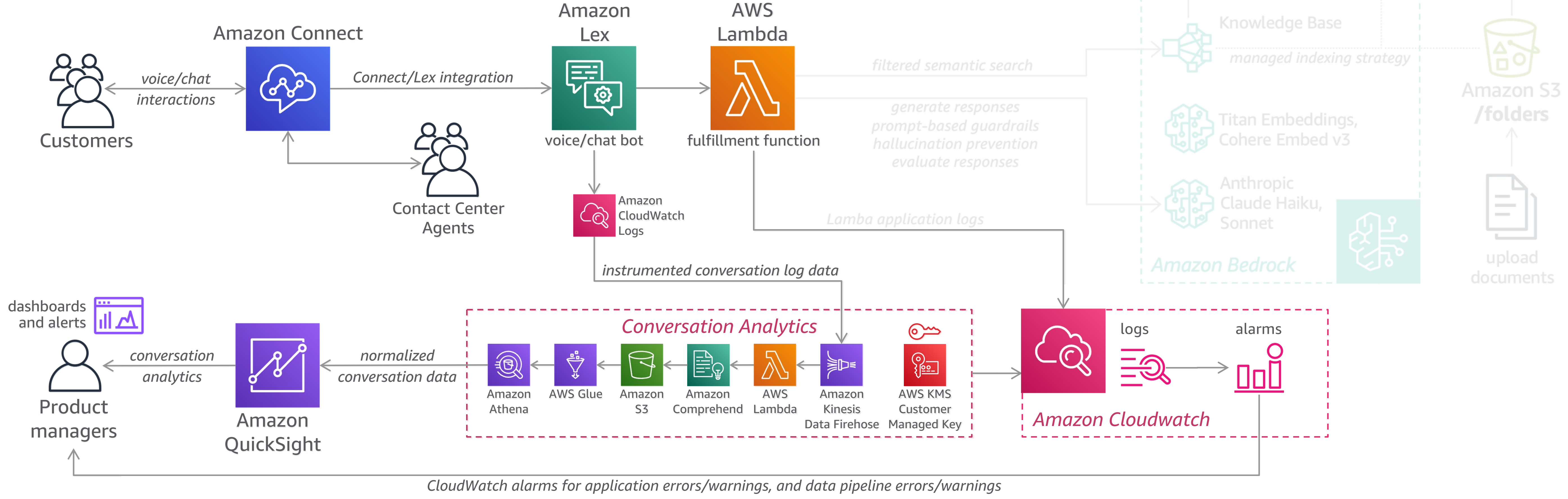
*The conversation analytics susbsystem consists of a data pipeline stack and an Amazon QuickSight dashboard, together with Amazon CloudWatch alarms.*

*The data pipeline extracts Lex conversation logs from Amazon CloudWatch Logs, "flattens" them into a tabular, name/value pair structure to simplify queries and analytics, and stores them in an S3 bucket. AWS Glue crawls the S3 bucket on a scheduled basis (defaulting to every 5 minutes) and updates a data catalog schema to make the data accessible via Amazon Athena via SQL queries.*

*For protecting sensivite data, the data pipeline can optionally redact PII data using Amazon Comprehend, and can apply an AWS Key Management Service customer-managed key (CMK) to limit access to both the CloudWatch Logs log group and the S3 bucket to specifically identified principals.*

*The resulting dataset includes hundreds of attributes for every conversation turn, including the user question, the RAG solution's answer, Lex slot values and session attributes, the LLM model version used, the knowledge base identifyer, retrieval and RAG latency, input/output token counts, and much more, providing fine-grained observability into the solution.*

*All testing data, including pass/fail results and explanations, as well as hallucination detection results and explanations, are also available for evaluating test runs.*

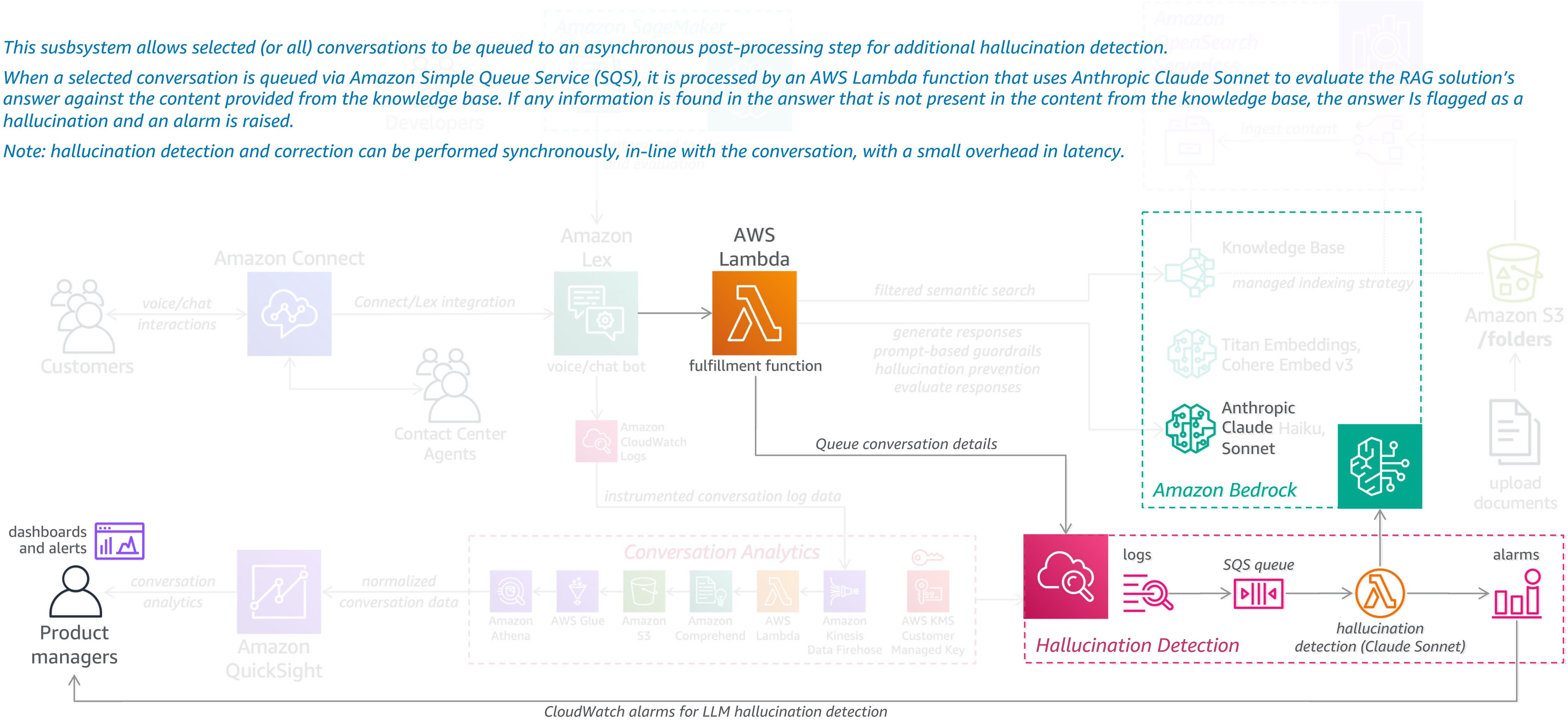


# Hallucination detection

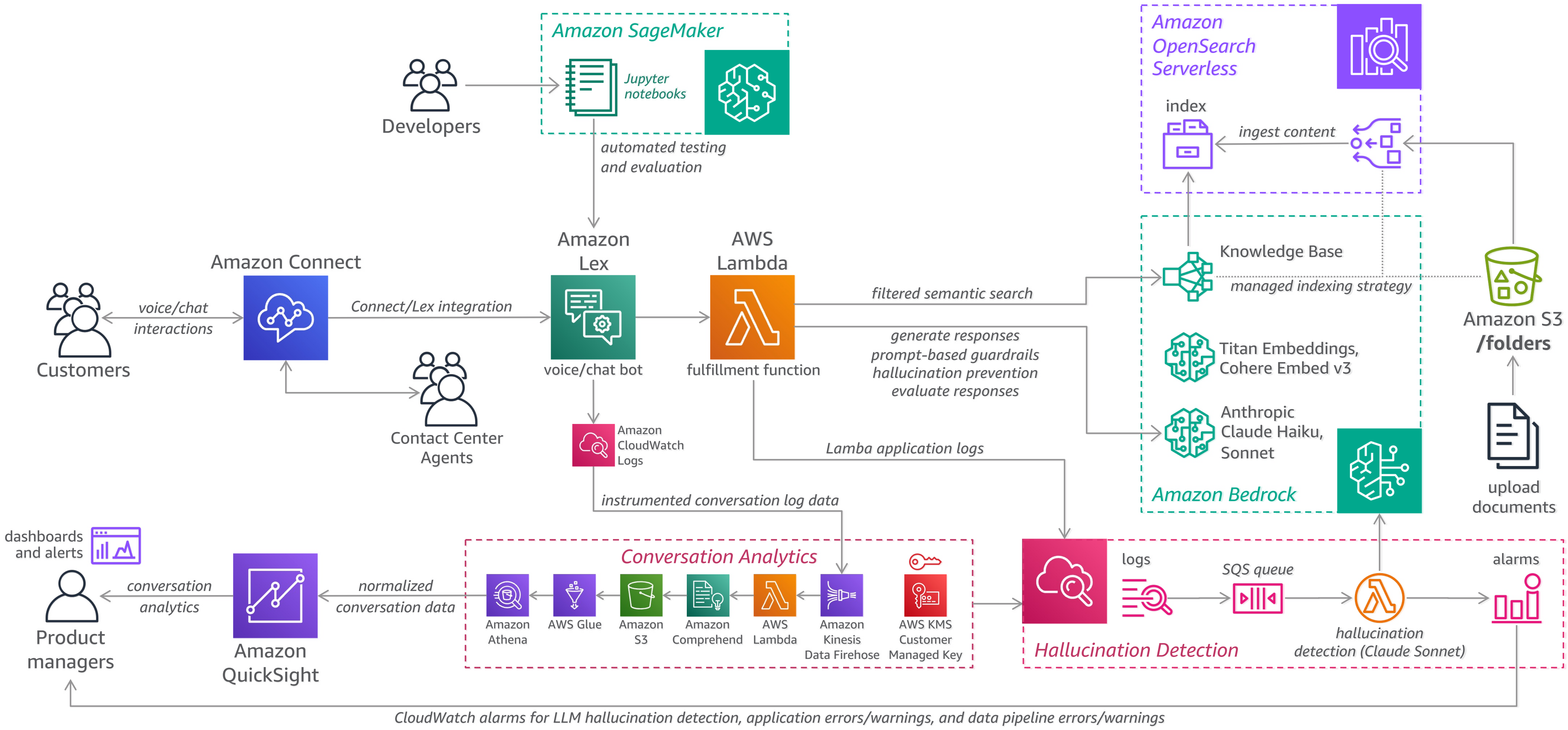
*This susbsystem allows selected (or all) conversations to be queued to an asynchronous post-processing step for additional hallucination detection.*

*When a selected conversation is queued via Amazon Simple Queue Service (SQS), it is processed by an AWS Lambda function that uses Anthropic Claude Sonnet to evaluate the RAG solution's answer against the content provided from the knowledge base. If any information is found in the answer that is not present in the content from the knowledge base, the answer Is flagged as a hallucination and an alarm is raised.*

*Note: hallucination detection and correction can be performed synchronously, in-line with the conversation, with a small overhead in latency.*



# Full end-to-end contact center RAG solution architecture



Amazon SageMaker

Developers

Jupyter notebooks

automated testing and evaluation

Amazon OpenSearch Serverless

index

ingest content

Customers

voice/chat interactions

Amazon Connect

Connect/Lex integration

Amazon Lex

voice/chat bot

AWS Lambda

fulfillment function

filtered semantic search

generate responses
prompt-based guardrails
hallucination prevention
evaluate responses

Knowledge Base

managed indexing strategy

Titan Embeddings, Cohere Embed v3

Anthropic Claude Haiku, Sonnet

Amazon Bedrock

Amazon S3 /folders

upload documents

Contact Center Agents

Amazon CloudWatch Logs

Lamba application logs

instrumented conversation log data

dashboards and alerts

Product managers

conversation analytics

Amazon QuickSight

normalized conversation data

Conversation Analytics

Amazon Athena

AWS Glue

Amazon S3

Amazon Comprehend

AWS Lambda

Amazon Kinesis Data Firehose

AWS KMS Customer Managed Key

Hallucination Detection

logs

SQS queue

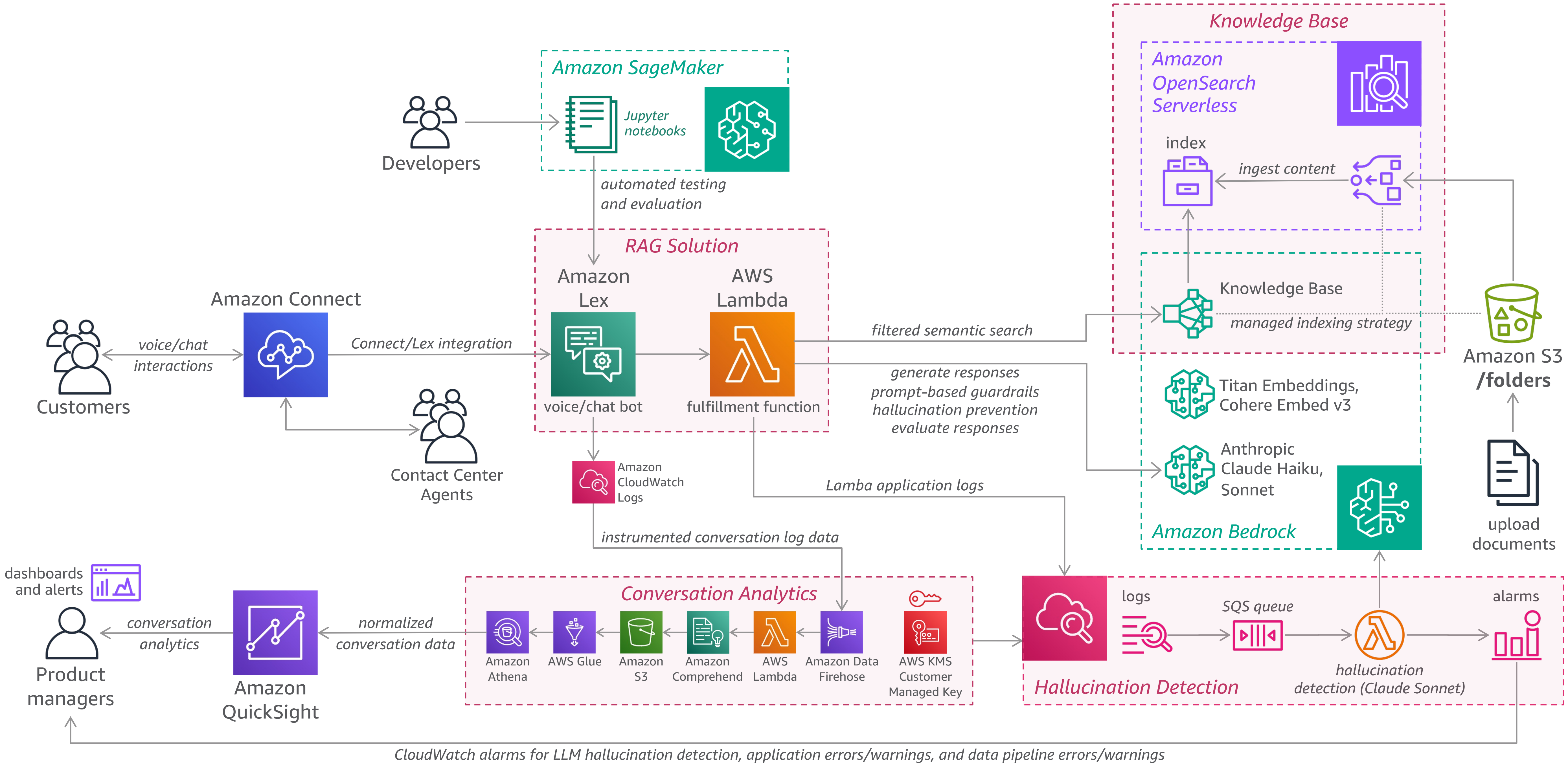
hallucination detection (Claude Sonnet)

alarms

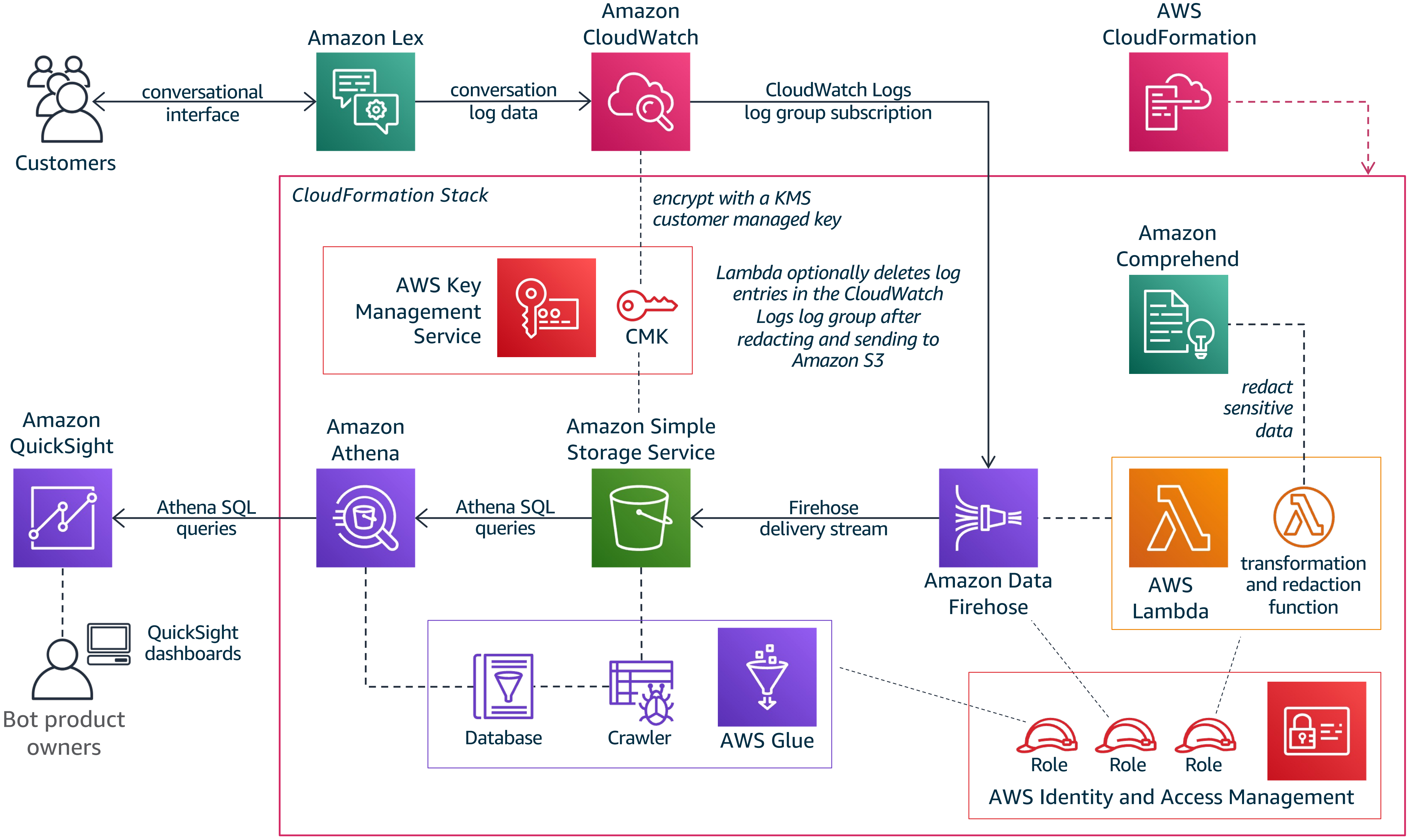
CloudWatch alarms for LLM hallucination detection, application errors/warnings, and data pipeline errors/warnings

# Appendix

# CloudFormation stacks



# Conversation analytics data pipeline – detail view


Customers
conversational interface
Amazon Lex
conversation log data
Amazon CloudWatch
CloudWatch Logs log group subscription
AWS CloudFormation
CloudFormation Stack
encrypt with a KMS customer managed key
AWS Key Management Service
CMK
Lambda optionally deletes log entries in the CloudWatch Logs log group after redacting and sending to Amazon S3
Amazon Comprehend
redact sensitive data
Amazon QuickSight
Amazon Athena
Amazon Simple Storage Service
Athena SQL queries
Athena SQL queries
Firehose delivery stream
Amazon Data Firehose
AWS Lambda
transformation and redaction function
QuickSight dashboards
Bot product owners
Database
Crawler
AWS Glue
Role
Role
Role
AWS Identity and Access Management