

Project Design Phase-I Solution Architecture

Date	19 September 2022
Team ID	PNT2022TMID36907
Project Name	Project – DemandEst-AI powered Food Demand Forecaster
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:

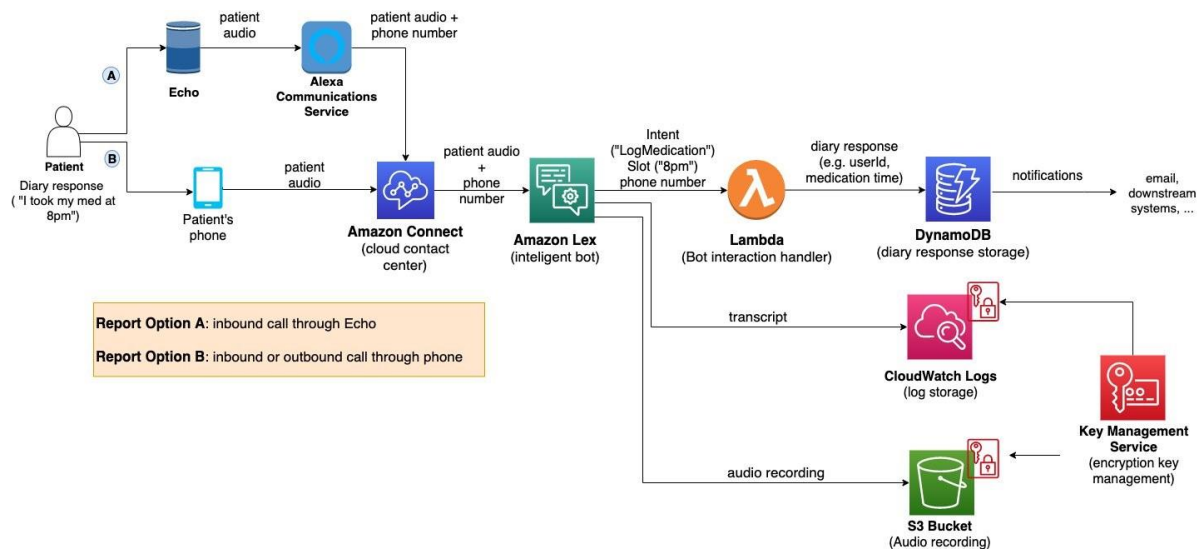


Figure 1: Architecture and data flow of the voice patient diary sample application

Reference: <https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/>

The following image shows a high-level architecture and data flow diagram of the voice-enabled patient diary sample implementation. Note that the interaction is two-way in nature:

The application supports three different ways for patients to reach the patient diary voice reporting interface:

- Making an inbound call through their smart phone or landline
- Making an inbound call through a smart speaker such as [Amazon Echo](#)
- Receiving an outbound call on their smart phone or landline, initiated by an automated patient outreach scheduler that uses the [StartOutboundContact API of Amazon Connect](#).

Regardless of which method the patient uses to reach the reporting application, the patient call is handled by [Amazon Connect](#). In this application, the contact flow first greets the patient, then verifies their identity. If successfully authenticated, the flow then checks which questionnaires the patient needs to complete, and collects responses for each questionnaire accordingly.

To support natural language interactions and interpret the patient's response, the Amazon Connect contact flow delegates each questionnaire to a corresponding [Amazon Lex bot](#). The Amazon Lex service allows you to build conversational interfaces using voice and text. It provides developers the same deep-learning based, conversational AI building blocks that power [Amazon Alexa](#): automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text. You can build a conversational interface for your application in minutes by supplying a few example phrases for each intent you want to support.

After the Amazon Lex service interprets the patient's response and parse it into semi-structured data (e.g. `medicationTime='20:00'`), it invokes an [AWS Lambda](#) function where you can define custom business logic to react to this information, such as storing in a clinical research database or an EHR system or notifying the principal investigator or the patient's caregiver if the patient has missed their medication. In the prototype application, the reported information is stored in a set of [Amazon DynamoDB](#) tables, and notifications to caretaker/providers are implemented using [Amazon Pinpoint](#).