Patient Hospital Information Access via Telephone

January 2021

Notices

Contents

Abstract

The telephone is a powerful tool for interacting with and improving the lives of patients by providing easy access to patient data and care services. In this whitepaper, we describe how to integrate AWS services with existing electronic health record (EHR) architectures based on medical research-based outcomes.

Introduction

Interactive Voice Response (IVR) systems have existed since the 1980s. Originally expensive, multi-million-dollar systems based on custom hardware, these quickly shrank in cost and size during the 1990s and early 2000s. Migrating these systems to the cloud was the natural next step. With the arrival of Amazon Connect and its ability to link telephones to other AWS services, that capability became widely and inexpensively available. IVR has now become a simple way of providing access to complex patient data and patient services where a mobile phone with limited data capability is the best (or only) means of remotely accessing care.

Use Cases

One study covering Spanish speaking populations was able to achieve significant patient impact using IVR:

> "We used data from Interactive Voice Response (IVR) self-management support studies in Honduras, Mexico, and the United States (US) to determine whether IVR calls to Spanish-speaking patients with chronic illnesses is a feasible strategy for improving monitoring and education between face-to-face visits. 268 patients with diabetes or hypertension participated in 6–12 weeks of weekly IVR follow-up. IVR calls emanated from US servers with connections via Voice over IP. More than half (54%) of patients enrolled with an informal caregiver who received automated feedback based on the patient's assessments, and clinical staff received urgent alerts. Participants had on average 6.1 years of education, and 73% were women. After 2,443 person weeks of follow-up, patients completed 1,494 IVR assessments. Call completion rates were higher in the US (75%) than in Honduras (59%) or Mexico (61%; p<0.001). Patients participating with an informal caregiver were more likely to complete calls (adjusted odds ratio [AOR]: 1.53; 95% confidence interval [CI]: 1.04, 2.25) while patients reporting fair or poor health at enrollment were less likely (AOR:0.59; 95% CI: 0.38, 0.92). Satisfaction rates were high, with 98% of patients reporting that the system was easy to use, and 86% reporting that the calls helped them a great deal in managing their health problems. In summary, IVR self-management support is feasible among Spanish-speaking patients with chronic disease, including those living in less-developed countries. Voice over IP can be used to deliver IVR disease management services internationally; involving informal caregivers may increase patient engagement."[1]

There are several striking points in this study: the high satisfaction level, the significant positive impact in disease management, and how traditionally underserved the population of the study was. It is also interesting that the study mechanism worked across national boundaries and cultural groups. The mechanism described in the study can be implemented for under a dollar per patient contact, which is far less expensive than human-supported visits or even some mobile application-supported interactions.

In another study, patients with COPD were found to require 30 fewer hospital days with IVR-based follow-up. However, the same study noted that IVR-based follow-up did not materially impact congestive heart failure patients. IVR is not applicable in all circumstances, but for the right condition, where frequent follow-up is beneficial, IVR can provide significant benefits.[2]

The improvement in patient outcomes implied by 30 fewer hospital days is significant. Beyond the patient impact, this approach also helps patient care organizations that are judged and financially rewarded based on reducing patient hospital stays and readmissions.

Healthcare organizations can leverage IVR systems for patient portals. Patients expect to access their healthcare information using voice, chat, and the web. Using an IVR system to provide patient access to individual medical records is a good way to increase utilization of patient portals, while serving the most at-risk portions of the patient population.[3]

Business Case

in the 1980s, when IVR and its use in healthcare began, systems involved many servers, specialized telephone connections (T1 and T3 lines), and plenty of IT and telephony expertise. Although medical studies showed clear benefits to patients and providers using IVR, technical points of friction limited medical application. The introduction of Amazon Connect and its ability to seamlessly integrate with other Amazon services removes many of the barriers, such as cost, complexity, speed of implementation, and lack of agility to meet patient needs, that previously impeded adoption and utilization of IVR in healthcare. Amazon Connect makes IVR an extension of an organization’s cloud footprint. This means data can interact and flow seamlessly between the IVR system, a data lake, the EHR, and artificial intelligence (AI) services with little user effort. It also means architectures can be trialed for small amounts of money (hundreds or thousands of dollars), then scaled to support millions of patients and healthcare providers.

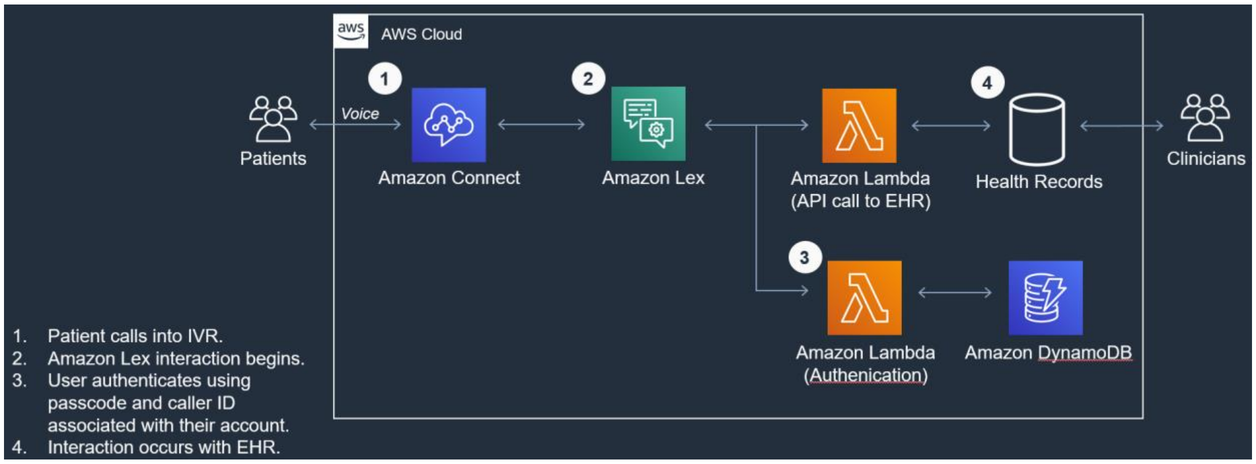
Services that can be Used for the Architecture

There are four key services used for this architecture:

- **Amazon Connect** is an omnichannel cloud contact center that enables you to build your own IVR system within minutes.
- **Amazon Lex** is a service for building conversational interfaces using voice and text.
- **AWS Lambda** enables you to run code without provisioning servers.
- **Amazon DynamoDB** is a key-value and document database that delivers single-digit millisecond performance at any scale.

Implementing the Architecture

The overall architecture consists of four main components to enable voice interactions with an EHR system. Connect handles the call flows for interacting with the patients, and uses an Amazon Lex bot to facilitate interactions with backend systems. Amazon DynamoDB stores information to match a patient to a registered phone number and voice passcode. Interactions with the EHR are handled with Lambda functions. While the architecture diagram illustrates one Lambda function making a single API call, you can add as many Lambda functions and API calls as required to build additional interactions from Lex to an EHR.



1. Patient calls into IVR.
2. Amazon Lex interaction begins.
3. User authenticates using passcode and caller ID associated with their account.
4. Interaction occurs with EHR.

Figure 1 – Architecture for patient hospital information access via telephone

To implement the architecture:

1. Set up an Amazon Connect instance. This creates a fully managed, pay-as-you-go, cloud IVR system that patients can call in to. For now, set up the instance with an administrator account and without customizing contact flows. You will create a contact flow in a later step. See Get started with Amazon Connect.
2. Create a Lex bot to facilitate user interactions with your backend systems. When setting up your Lex bot, create an intent called `VerifyPatientCaller` and leave it empty. Create a second intent called `GetCurrentMedications` and leave it empty as well. You will use these intents later.
3. If you want to allow interactions with a backend system, the caller must be authenticated. In this example, you create an Amazon DynamoDB table to store phone numbers and passcodes registered to patients. For production, consult with your organization's legal team to meet the security requirements for authenticating patients.
4. With your data store in place, create a Lambda function to take in a parameter for "phone number". The Lambda function checks your data store and return a custom attribute that tells you whether the user is validated. (This step is simplified for illustrative purposes. Work with your organization's security team to determine what constitutes a valid authentication mechanism for your users.)
5. Create another Lambda function to fulfill your Lex intent for retrieving medication data. This function should handle all necessary intents for a bot, so use an Amazon Lex blueprint when creating your function.
6. Create an Amazon Connect contact flow to prompt a user for their passcode (or other identifying attributes), validate against your data store, and then prompt the user for utterances that trigger your Lex intents. In this case, your prompt might say something like "Ask me about your current medications." See Figure 2 for an example contact flow.

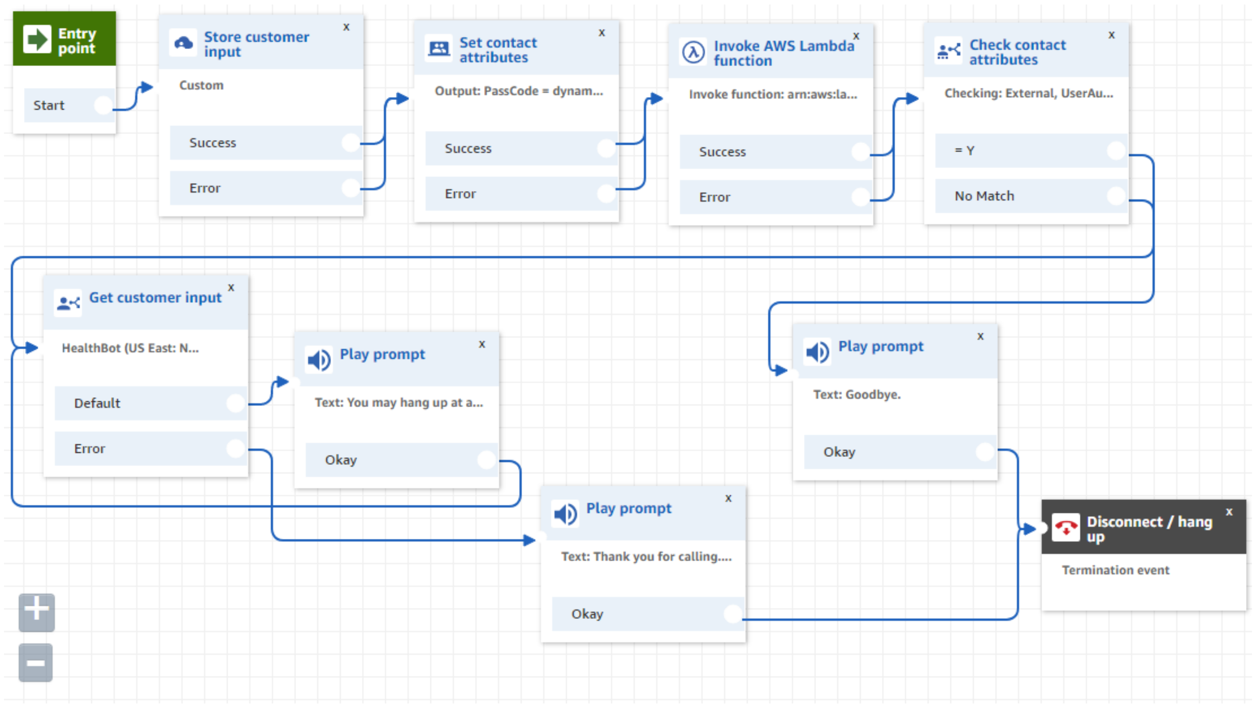
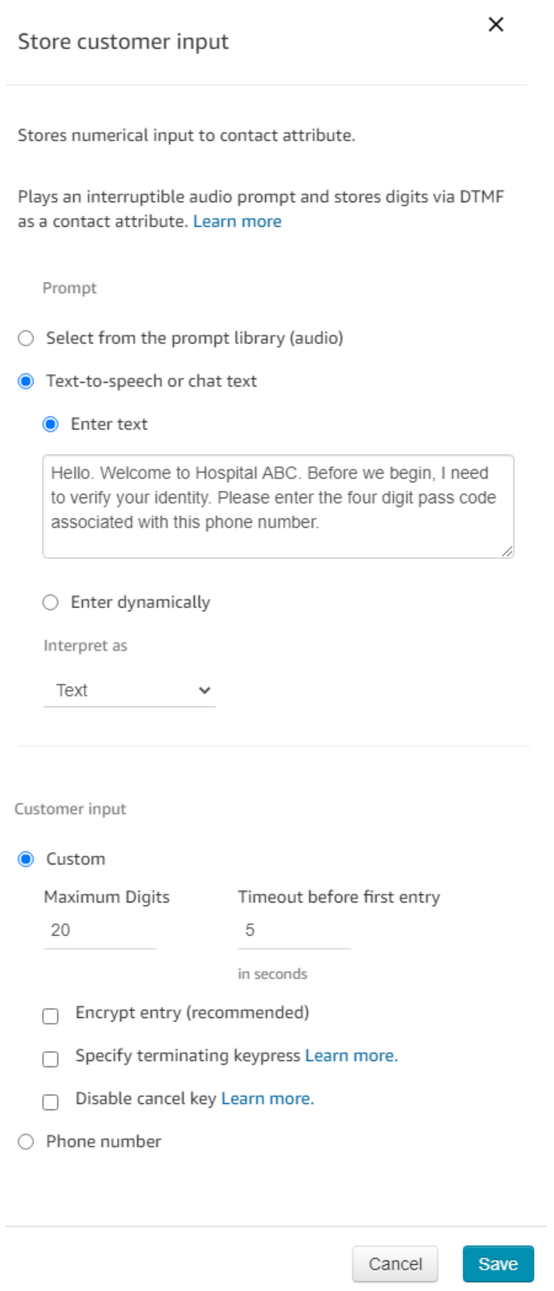


Figure 2 – Example contact flow

Contact Flow Details

The first block welcomes the patient and prompts for a passcode.



Store customer input

Stores numerical input to contact attribute.

Plays an interruptible audio prompt and stores digits via DTMF as a contact attribute. Learn more

Prompt

( ) Select from the prompt library (audio)

(●) Text-to-speech or chat text

(●) Enter text

Hello. Welcome to Hospital ABC. Before we begin, I need to verify your identity. Please enter the four digit pass code associated with this phone number.

( ) Enter dynamically

Interpret as: Text

Customer input

(●) Custom

Maximum Digits: 20

Timeout before first entry: 5 in seconds

☐ Encrypt entry (recommended)

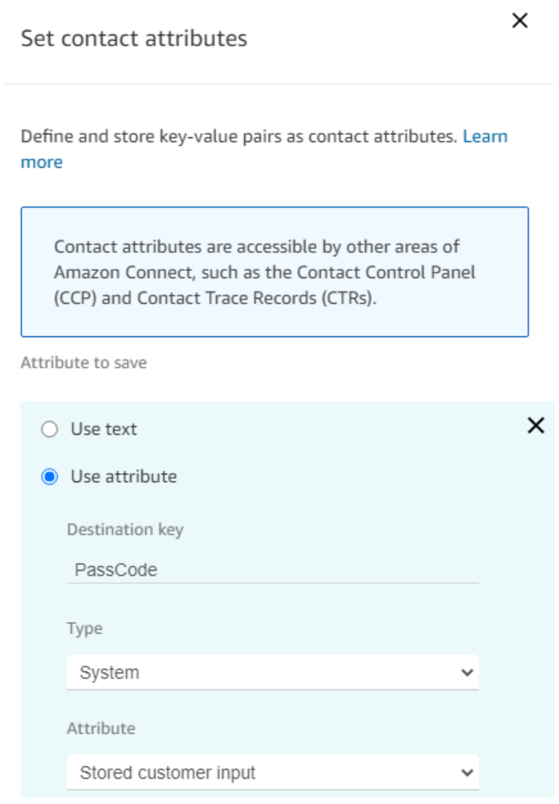
☐ Specify terminating keypress Learn more.

☐ Disable cancel key Learn more.

( ) Phone number

Figure 3 – Patient welcome and passcode prompt

The second block captures the patient's passcode and stores it to an attribute that you can later pass on to your Lambda function.


Set contact attributes

Define and store key-value pairs as contact attributes. Learn more

Contact attributes are accessible by other areas of Amazon Connect, such as the Contact Control Panel (CCP) and Contact Trace Records (CTRs).

Attribute to save

Use text [✓ Use attribute]

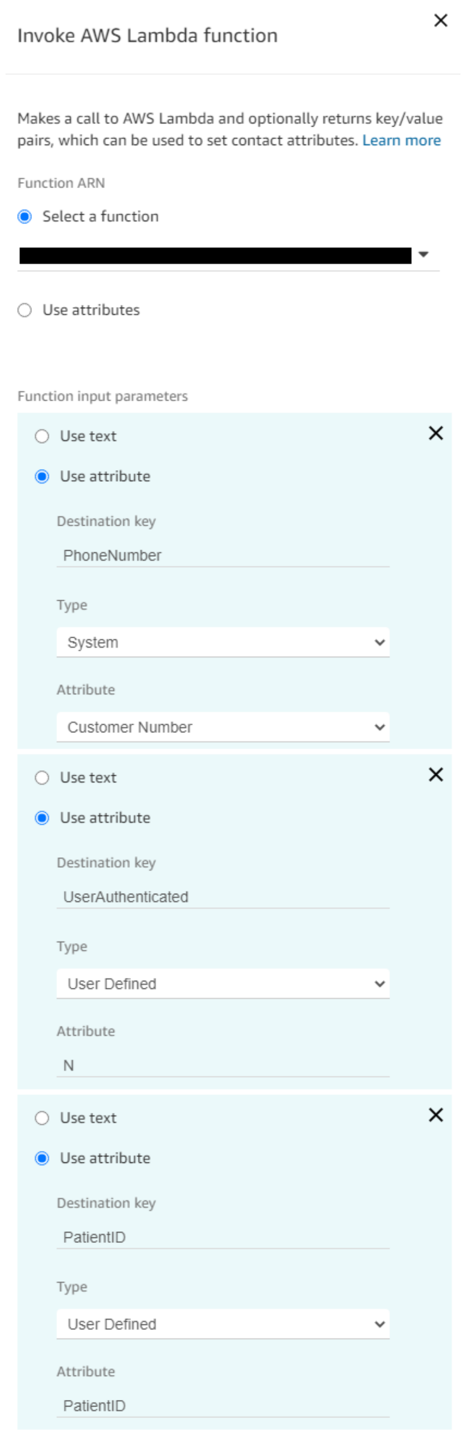
Destination key: PassCode

Type: System

Attribute: Stored customer input

Figure 4 – Patient's passcode is captured and stored to an attribute

The third block invokes the Lambda function to validate the user's information.



Invoke AWS Lambda function

Makes a call to AWS Lambda and optionally returns key/value pairs, which can be used to set contact attributes. Learn more

Function ARN

● Select a function

[redacted]

○ Use attributes

Function input parameters

○ Use text
● Use attribute

Destination key: PhoneNumber
Type: System
Attribute: Customer Number

○ Use text
● Use attribute

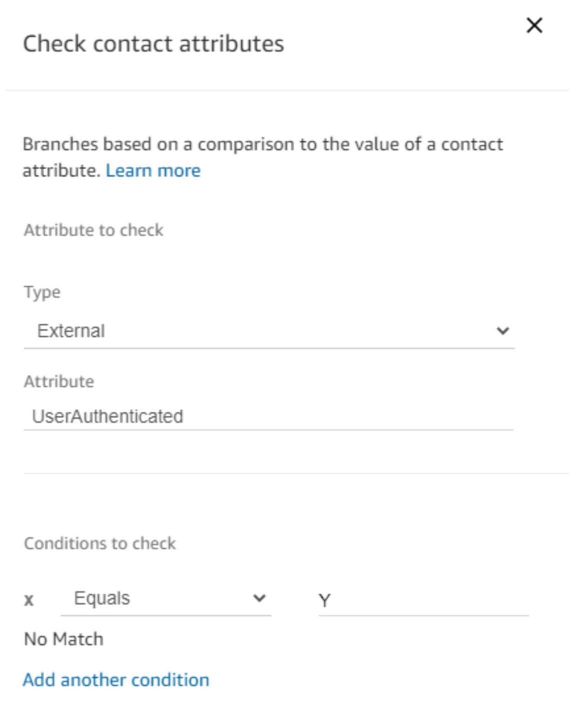
Destination key: UserAuthenticated
Type: User Defined
Attribute: N

○ Use text
● Use attribute

Destination key: PatientID
Type: User Defined
Attribute: PatientID

Figure 5 – Invoking the Lambda function to validate the user's information

After validating the caller, the contact flow can branch to have different conversation paths based on whether the patient is authenticated. This concept can be used for a variety of other branching logic you might consider as you build more complex contact flows.


Check contact attributes

Branches based on a comparison to the value of a contact attribute. Learn more

Attribute to check

Type: External

Attribute: UserAuthenticated

Conditions to check

x Equals Y

No Match

Add another condition

Figure 6 – Contact flow branches

The last component of the contact flow is the bot interaction. The user loops back to the same bot and continues to interact with it after completing an intent.

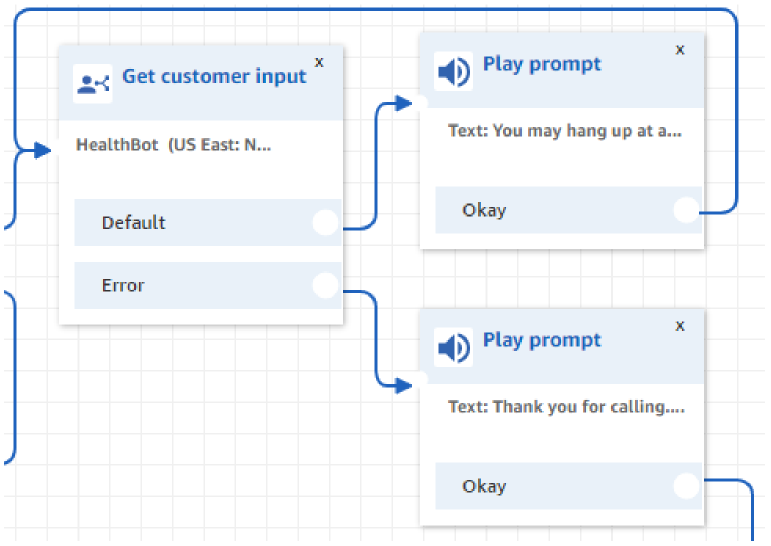


Figure 7 – Bot interaction

Reading Data from the EHR

Reading data from your EHR is possible by calling a Lambda function and returning the result back to Lex or Amazon Connect through JSON. This means that your Lambda fulfillment function can call your EHR to get a value, and then that value can be appended as a custom JSON attribute to be passed back to Lex and Connect. This enables you to create conversations that read and write data between Connect and your backend data store, and change conversations based on values from the EHR. In this example, suppose you have a Lambda function that queries your EHR to decide if a patient has had a recent surgery. You can call the Lambda function directly from Amazon Connect.

Review the following sample code (Node.js). A custom JSON value of `[‘recentsurgery’] = ‘Y’` is appended to the JSON response to simulate reading from an EHR system’s patient history and returning a result. In an actual implementation, you use an API call to query the EHR and return the result. You can append any custom JSON attribute to your response so it can be used in later points in your architecture.

```
'use strict';
const AWS = require("aws-sdk");

function close(sessionAttributes, fulfillmentState, message) {
    return sessionAttributes;
}

// --------------- Events -----------------------
function dispatch(intentRequest, callback) {
    const sessionAttributes = intentRequest.Details.Parameters;
    // Query EHR here and determine if recent surgery logic
    // ...
    // For demo, we'll assume EHR returned
    sessionAttributes['recentsurgery'] = 'Y'
    callback(close(sessionAttributes, 'Fulfilled',  {'contentType':
'PlainText', 'content': + 'Success' }));
}

// --------------- Main handler -----------------------
exports.handler = (event, context, callback) => {
    try {
        dispatch(event,
            (response) => {
                callback(null, response);
            });
    } catch (err) {
        callback(err);
    }
};
```

Because the JSON response now contains an attribute, you can create a contact flow branch based on whether the patient recently had surgery. This enables you to create different conversation paths, such as asking the patient to rate their pain level on a scale of 1 to 5. If they didn’t have surgery, the contact flow can proceed to another set of options.

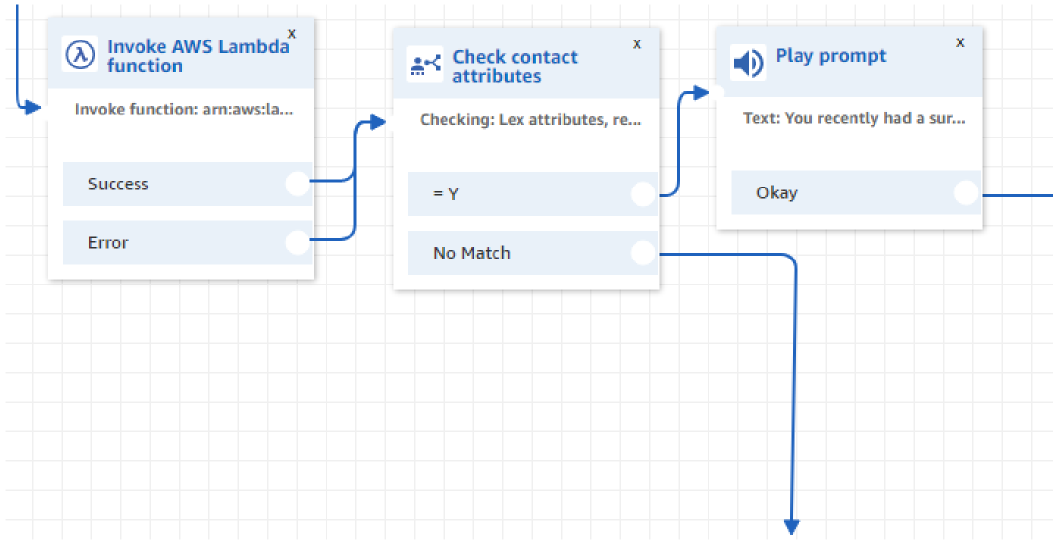
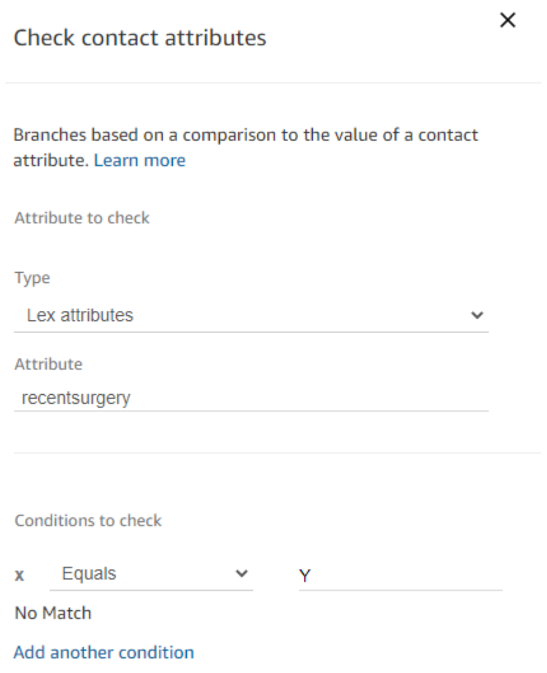

Invoke AWS Lambda function
Invoke function: arn:aws:la...
Success
Error
Check contact attributes
Checking: Lex attributes, re...
= Y
No Match
Play prompt
Text: You recently had a sur...
Okay

Figure 8 – Further contact flow branches

Use a **Check contact attributes** box (shown in Figure 8) to look at the returned variable for recent surgeries. The details of this are shown in Figure 9. Note that you can customize branching based on any value you pass back from the Lambda function.



Check contact attributes

Branches based on a comparison to the value of a contact attribute. Learn more

Attribute to check

Type: Lex attributes

Attribute: recentsurgery

Conditions to check

x Equals Y

No Match

Add another condition

Figure 9 – Check contact attributes

Writing data back to the EHR is a similar process. The only difference is that you read the JSON attributes within your Lambda fulfillment function, and then write them to the EHR via Lambda.

Best Practices

The gateway to success in all clinical care scenarios, especially when using IVR, is closely tracking and responding to patient satisfaction metrics. Delivering a satisfying experience to patients increases their engagement and helps make their healthcare experience a success. Amazon Connect enables sentiment analysis as well as collecting survey information. Both of these are good sources for tracking patient satisfaction in real-time.

We also encourage healthcare innovators to leverage existing studies and information. There are a large number of studies available that help identify mechanisms that have substantial positive impact on patient outcomes. This research can, at a reasonable cost, be turned into a scalable implementation that benefits large patient populations.

Conclusion

This paper demonstrated how, with a small investment in both time and money, clinically significant mechanisms can be implemented using AWS Services and an existing EHR system. The example provided demonstrates how simply and quickly the mechanisms in the referenced medical research can be turned into implemented architectures. If you are interested in implementing such an architecture within your organization, AWS has a team of Solution Architects and Partners to assist you.

Contributors

Contributors to this document include:

- Brian Niemeyer, Senior Partner Solutions Architect, HCLS
- Brian Warwick, Senior Partner Solutions Architect, HCLS

Document Revisions

Date	Description
December 2020	First publication

Notes

[1] Spanish-Speaking Patients' Engagement in Interactive Voice Response (IVR) Chronic Disease Self-Management Support Calls: Analyses of Data from Three Countries https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3778441/

[2] e-Coaching: Interactive Voice Response-Enhanced Care Transition Support for Complex Patients (Alabama) https://digital.ahrq.gov/ahrq-funded-projects/e-coaching-interactive-voice-response-enhanced-care-transition-support-complex

[3] University of Southern Alabama Remote Patient Monitoring Solution RMEDE https://www.southalabama.edu/centers/cshi/sol_rmede.html