Mejores prácticas para diseñar las bases de una experiencia de IVR dinámica y modular en Connect Customer

AWS Orientación prescriptiva

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Orientación prescriptiva: Mejores prácticas para diseñar las bases de una experiencia de IVR dinámica y modular en Connect Customer

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Introducción 1
Descripción general de 1
Funciones de Connect Customer para aplicaciones de IVR 3
Diseñador de flujos 3
Flujos 3
Módulos de flujo 3
AWS Lambda funciones 4
Principios de diseño de IVR 5
Identificación de los objetivos de su organización 5
Planificación del recorrido de contacto con el cliente 7
Creación de la arquitectura fundamental 9
Eliminar el diseño estático 10
Identificar los procesos repetibles 11
Ejemplo de caso de uso 13
Identificar las metas de la organización 13
Planificar el recorrido de contacto con el cliente 14
Crear la arquitectura fundamental 15
Conclusión 18
Recursos 19
Historial de documentos 20
..... xxi

Mejores prácticas para diseñar las bases de una experiencia de IVR dinámica y modular en Connect Customer

Ayesha Borker y Nicholas Pung, Amazon Web Services (AWS)

agosto de 2023 (historial de documentos)

Los arquitectos técnicos, los desarrolladores y los equipos empresariales tienen la oportunidad de reimaginar las experiencias de autoservicio automatizadas que ofrecen a sus clientes cuando migran a un centro de contacto basado en la nube mediante Connect Customer.

Los equipos empresariales suelen querer agregar funcionalidades y características, además de personalizar más la experiencia de autoservicio. Los arquitectos técnicos y los desarrolladores se centran sobre todo en cómo reducir la redundancia, permitir cambios rápidos y flexibilidad, modular los procesos repetidos y reducir los gastos generales de mantenimiento.

En esta guía se proporciona a los arquitectos técnicos un conjunto de prácticas recomendadas que pueden utilizar para crear el diseño fundamental de su aplicación de respuesta de voz interactiva (IVR) de manera modular y dinámica. Se proporcionan ejemplos de los sistemas de IVR (es decir, tecnologías de voz). Sin embargo, puede aplicar los mismos conceptos a cualquier otro canal. La guía se centra en crear un diseño de IVR modular y escalable mediante el uso de flujos y módulos de Connect Customer. Amazon Lex, que le permite agregar características de lenguaje natural (NL) a la aplicación de IVR, está fuera del ámbito de aplicación.

Descripción general de

Los métodos tradicionales para el diseño de experiencias de IVR pueden ser estáticos y complejos. Las organizaciones a menudo administrar experiencias independientes para distintos idiomas, entornos de implementación (como desarrollo, pruebas y producción), países y líneas de negocio (LOB). Con frecuencia, confían en el talento de la voz humana para crear peticiones, que luego se cargan y se hace referencia a estas en guiones codificados. Esto complica el proceso para hacer cambios y actualizaciones, sobre todo en tiempo real en el caso de emergencias o anuncios urgentes. Cuando las aplicaciones se separan de esta manera, a menudo se observa que los casos de uso habituales se repiten y se administran de manera redundante. Algunos ejemplos comunes son la recepción de pagos mediante un sistema de IVR o el envío de un SMS después de la transacción. Las integraciones de backend con bases de datos, soluciones de administración de

relaciones con los clientes (CRM) u otros sistemas de terceros suelen tener una codificación rígida, lo que significa que los cambios deben replicarse de manera manual en varias aplicaciones.

Este tipo de diseño de arquitectura monolítica genera procesos administrativos y gastos generales de administración excesivos, e impide la innovación. Los desarrolladores dedican más tiempo a implementar cambios pequeños en varias dependencias, lo que les dificulta seguir procesos ágiles. A menudo, esta complejidad se convierte en una carga y las empresas deben recurrir a organizaciones de servicios profesionales o consultores externos para que les ayuden a administrar estos cambios. Como resultado, la empresa aumenta el tiempo de respuesta de las actualizaciones habituales. Una arquitectura complicada que implica estos ciclos de desarrollo complejos y largos provoca un aumento de los costos.

Esta guía se centra en una arquitectura fundamental para una aplicación de IVR que ayuda a eliminar las redundancias y agiliza el proceso de administración de cambios. Esta arquitectura facilita a los desarrolladores el mantenimiento y la innovación. También proporciona a las empresas la agilidad que necesitan.

Contenido

- Funciones de Connect Customer para aplicaciones de IVR
- Principios de diseño de IVR
- Ejemplo de caso de uso
- Conclusión
- Recursos

Funciones de Connect Customer para aplicaciones de IVR

Connect Customer es un centro de contacto omnicanal en la nube que ayuda a las empresas a ofrecer un servicio de atención al cliente superior a un bajo coste. Connect Customer ofrece una experiencia fluida a través de voz y chat para clientes y agentes. Esto incluye un solo conjunto de herramientas para el enrutamiento según las habilidades, eficaces análisis históricos y en tiempo real, herramientas de administración intuitivas y fáciles de usar y precios de pago por uso.

En esta sección se ofrece una descripción general de las funciones de Connect Customer que pueden ayudarle a crear una experiencia de autoservicio automatizada. Para ver los ejemplos de estos componentes, consulte la arquitectura de muestra que se describe más adelante en esta guía.

Diseñador de flujos

Connect Customer proporciona un diseñador de flujos de autoservicio basado en una interfaz gráfica de usuario (GUI). Esta característica incluye una interfaz para arrastrar y soltar que es útil para crear experiencias de cliente personalizadas de manera intuitiva y para hacer cambios sobre la marcha. Los bloques de flujos son los componentes básicos de la experiencia de IVR y enrutamiento. Cada bloque tiene una función específica (por ejemplo, reproducir una petición u obtener la entrada del cliente), lo que hace que los flujos sean intuitivos desde el punto de vista visual.

Flujos

Los flujos definen la experiencia del cliente con el centro de contacto de principio a fin. Ofrecen las opciones de la IVR y también dirigen a las personas que llaman a los agentes correctos según la información que recopilaron. Puede utilizar los flujos para interactuar con otros AWS servicios, por ejemplo AWS Lambda, para crear experiencias de cliente dinámicas y personalizadas. Los flujos también se pueden integrar con Amazon Lex para ofrecer interacciones realistas en lenguaje natural.

Módulos de flujo

Los módulos de flujos son secciones reutilizables de un flujo. Puede utilizarlos para extraer una lógica repetible en los flujos y llevar a cabo funciones comunes. Por ejemplo, puede crear un módulo que implemente pagos por IVR para las personas que llaman o que envíe notificaciones por SMS después de las transacciones.

AWS Lambda funciones

Los flujos le permiten interactuar con los sistemas secundarios, como la gestión de pedidos, los CRM, los sistemas de tickets y las bases de datos, mediante un bloque de AWS Lambda flujo. Esta integración permite las interacciones de autoservicio en el sistema de IVR con una mayor tasa de contención. Puede automatizar con facilidad los casos de uso habituales de los clientes, como obtener novedades sobre el estado de los pedidos o verificar los saldos de las tarjetas de crédito. También puede utilizar las API para actualizar las preferencias de los clientes en el sistema de CRM o crear tickets en su nombre.

Principios de diseño de IVR

El diseño efectivo de un IVR consta de dos principios principales:

- Centrarse en los impulsores de su empresa y en las metas de la organización
- Crear una arquitectura optimizada que pueda adaptarse a estos objetivos y escalarse para crecer e iterar en el futuro.

En esta sección se explican con más detalle estos principios en tres áreas de actuación clave, que ayudan a desglosar la planificación de IVR y a organizar las iniciativas en una secuencia lógica:

- Identificación de los objetivos de su organización
- Planificación del recorrido de contacto con el cliente
- Creación de la arquitectura fundamental

Identificación de los objetivos de su organización

La experiencia del cliente debe alinearse con los objetivos y resultados generales de su empresa. Podrían ser objetivos a largo plazo que dirige el equipo de liderazgo sénior, o iniciativas a corto plazo que impulsan los equipos individuales o las unidades de negocio. Estas metas deben estar alineadas con la mejora de la experiencia general del cliente.

Recomendación: Primero, trabaje con las partes interesadas correspondientes de la organización para conocer sus prioridades principales a corto y largo plazo. Recopile los datos de todos los departamentos correspondientes e identifique los patrones y las tendencias. Priorice estas iniciativas y asigne plazos razonables a cada meta.

En la lista siguiente se proporcionan ejemplos de metas empresariales. Pueden clasificarse como a corto o largo plazo según las necesidades de su organización.

- Mejorar la contención en la IVR

  Esto significa mejorar las funcionalidades del sistema de IVR para que los clientes puedan administrar tareas sencillas por sí mismos y dejar que los agentes se ocupen de los problemas más complejos. Esto podría incluir la automatización de las preguntas frecuentes, las consultas de

estado sencillas (como el estado de un ticket, vuelo, reclamación o pedido), las consultas de saldo, el restablecimiento de las contraseñas y la autenticación de los clientes.

- Impulsar las ventas y los ingresos

  Los centros de contacto pueden ayudar a impulsar las ventas y los ingresos de las empresas. Un diseño de IVR efectivo y una experiencia de escritorio eficiente para los agentes pueden ayudar a aprovechar las oportunidades de ventas adicionales y cruzadas. Ejemplo: ofrecer de manera proactiva una mejora a un cliente que compró hace poco un ticket de avión.

- Personalizar la experiencia del cliente

  Las empresas suelen ofrecer experiencias de autoservicio monótonas que no generan ningún compromiso de sus clientes. Los clientes reciben el mismo modelo de opciones y, sin más, las personas que llaman terminan por buscar un agente en lugar de buscar oportunidades de autoservicio. Estas experiencias deficientes también pueden afectar a las puntuaciones de satisfacción del cliente (CSAT). La personalización puede ayudar mucho a cambiar la opinión de los clientes. La identificación positiva del cliente es un requisito previo para ofrecer una personalización. Ejemplo: saludar a los clientes por su nombre u ofrecer opciones de menú de IVR personalizadas según su historial de usuario.

- Reducir el abandono de las llamadas

  La falta de opciones de autoservicio o de personalización a menudo puede provocar tiempos de espera más largos, lo que podría hacer que se abandonen las llamadas y se obtengan puntuaciones bajas en la satisfacción del cliente. Esto también puede suponer una pérdida de ingresos, en especial para un centro de contacto de ventas. Un diseño de IVR eficaz puede ayudar a mitigar este problema de varias maneras. Por ejemplo, mediante la automatización de preguntas frecuentes, la verificación del estado de las colas y al ofrecer opciones de interacción alternativas.

- Mejorar la resolución en el primer contacto (FCR)

  La capacidad de resolver el problema del cliente la primera vez que se pone en contacto con el centro de contacto se denomina resolución en el primer contacto. Hay varios factores que pueden ayudar a mejorar esta métrica, pero la clave es contar con las opciones y el diseño correctos de la IVR. Ejemplo: identificación del autor de la llamada en la IVR, identificación de la intención del autor de la llamada y enrutamiento según el contexto a un agente debidamente cualificado.

- Reducir el tiempo medio de gestión (AHT)

  El tiempo medio de gestión se refiere al tiempo medio que dedica un agente a un cliente en una conversación en tiempo real. Por lo general, las empresas quieren reducir este tiempo para poder

resolver las consultas de los clientes con más rapidez, reducir el tiempo de espera de los clientes en la cola y mejorar las puntuaciones generales de la satisfacción del cliente. Una IVR inteligente que recopila la información del cliente con antelación ayuda a reducir el tiempo medio de gestión, ya que permite al agente llegar a la resolución con mayor rapidez. Ejemplo: automatizar la autenticación de los clientes, la clasificación de los clientes y la recopilación o búsqueda de datos.

Planificación del recorrido de contacto con el cliente

Una vez definidas las metas de su organización, trabaje hacia atrás para crear un plan del recorrido del cliente. Este plan ayuda a identificar los patrones de interacción típicos y los datos necesarios para resolver las consultas de los clientes. También ayuda a definir la experiencia que se presenta al cliente, las respuestas a los eventos y las emergencias sin planificar y el volumen y la escala de los contactos con los clientes necesarios para la implementación.

Este paso es fundamental para planificar los tres pilares AWS Well-Architected del marco: excelencia operativa, seguridad y confiabilidad.

Estas son algunas de las preguntas que se deben tener en cuenta al crear el plan de recorrido del contacto del cliente:

- ¿Cuál es el motivo del contacto del cliente?

  La respuesta a esta pregunta ayuda a identificar las preguntas más frecuentes, como las consultas de saldo, las actualizaciones del estado de los pedidos y el restablecimiento de contraseñas.

- ¿Cómo se valida la identidad del cliente?

  La seguridad es la prioridad máxima y validar la identidad del cliente es un paso fundamental en el recorrido del contacto. Entre las opciones de autenticación se incluyen los datos biométricos por voz, las contraseñas de un solo uso y la información personal, como el número de tarjeta de crédito y la fecha de nacimiento. Esta información también ayuda a identificar los requisitos de cumplimiento normativo, como la Ley de Portabilidad y Responsabilidad de Seguros Médicos (HIPAA), el Estándar de Seguridad de Datos del Sector de las Tarjetas de Pago (PCI DSS) y los Controles de Sistemas y Organizaciones (SOC).

- ¿Qué información tiene ya sobre este cliente?

  Una vez identificado un cliente, los datos correspondientes pueden ayudar a personalizar su experiencia y resolver problemas de manera proactiva. Entre estos datos se pueden incluir

el nombre del cliente, los casos pendientes, el historial y las opiniones de los contactos y el comportamiento de compra.

- ¿Qué información es necesaria para resolver la consulta del cliente?

  Determinar la información necesaria para responder a las consultas del cliente ayuda a definir los datos del cliente que se deben recopilar en el sistema de IVR. Por ejemplo, si un cliente quiere consultar su saldo, será necesario su número de cuenta y su código de seguridad, y la aplicación de IVR tendrá que integrarse con los almacenes de datos de backend que contienen datos bancarios. Para verificar el estado de su pedido, es necesario el número de pedido.

- ¿Es necesaria alguna integración para cumplir con la solicitud del cliente?

  Por lo general, para las solicitudes de autoservicio en la IVR son necesarias las integraciones de backend para poder atenderlas de manera correcta. Según el caso de uso, estas integraciones pueden hacerse con bases de datos, CRM o sistemas de generación de tickets de la organización. Por ejemplo, para comunicar el estado de un pedido podría ser necesario consultar la información desde un sistema de administración de pedidos. Es posible que para las compras sean necesarias las integraciones con una puerta de enlace de pago. Es fundamental identificar estas dependencias con antelación, para poder planificar la disponibilidad de las API pertinentes.

- ¿Cuáles son los requisitos de cumplimiento normativo relacionados con el recorrido o los datos del cliente?

  Es fundamental definir los objetivos de cumplimiento e incorporarlos en el diseño de la IVR. Los programas de cumplimiento se suelen aplicar a un nivel de la organización. Le recomendamos que involucre al equipo de seguridad de la información en estas conversaciones y sesiones de planificación.

  Por ejemplo, en el caso de la información de pago que figura en la IVR podría ser necesario el cifrado y el cumplimiento normativo de PCI DSS. Para la información de identificación personal (PII) recopilada de cualquier cliente podría ser necesario el cifrado. Su sistema de IVR podría incluir información adicional que tendría que proteger.

- ¿Cómo responde al alto volumen de llamadas o a la falta de agentes disponibles?

  Es esencial identificar las situaciones en las que un volumen alto de llamadas podría provocar tiempos de espera más largos. En estas situaciones, el sistema de IVR suele actuar como una puerta de enlace para redirigir de manera eficaz los contactos de los clientes. Entre algunas estrategias de desvío habituales se incluyen devolver las llamadas durante tiempos de espera excepcionalmente largos, diseñar escenarios de saturación que amplíen el número de agentes

objetivo, reproducir mensajes promocionales y redirigir a un cliente a otro canal (SMS o chat). Para obtener más información sobre estas estrategias, consulta la entrada del blog Cómo gestionar los picos de contacto inesperados con Connect Customer.

- ¿Cuál debe ser la experiencia del cliente durante los días festivos o en caso de emergencia?

  Las emergencias imprevistas y los eventos no planificados pueden afectar a la capacidad de su organización para atender a los clientes. Planificar la experiencia del cliente durante estos eventos es una práctica recomendada para garantizar la fiabilidad. Además, es posible que en los días festivos haya personal limitado. El sistema de IVR puede ser un lugar ideal para mantener informados a los clientes y establecer expectativas.

  Por ejemplo, puede diseñar marcadores de posición de emergencia y mensajes de días festivos en el sistema de IVR, actualizarlos y reproducir las actualizaciones pertinentes cuando sea necesario.

- ¿Cuál debe ser la experiencia del cliente cuando tenga problemas técnicos?

  La experiencia de un cliente siempre debe ser completa y significativa cuando sigue el flujo de IVR, incluso cuando se producen errores de integración de la aplicación o del backend. Le recomendamos diseñar la experiencia del autor de la llamada en caso de errores de integración. Por lo general, puede implementar reintentos y repeticiones, reproducir un mensaje que comunique la dificultad técnica o transferir al cliente a un agente. En cualquiera de los dos escenarios, la experiencia del autor de la llama no debe provocar una finalización abrupta e inexplicable de la comunicación, lo que, en general, daría lugar a más llamadas.

Tras describir el recorrido del contacto del cliente, busque procesos repetibles o patrones comunes. Estos serán los puntos clave en los que centrarse cuando diseñe la base de la arquitectura de la IVR.

Creación de la arquitectura fundamental

Una vez definidos los objetivos de su organización y descrito el recorrido del contacto del cliente, podrá utilizar esa información para definir una arquitectura fundamental para el sistema de IVR. Estos puntos de datos son fundamentales para crear una arquitectura de IVR que sea modular y escalable. Los principios clave para crear un buen marco fundamental son los siguientes:

- Eliminar el diseño estático
- Identificar los procesos repetibles

Eliminar el diseño estático

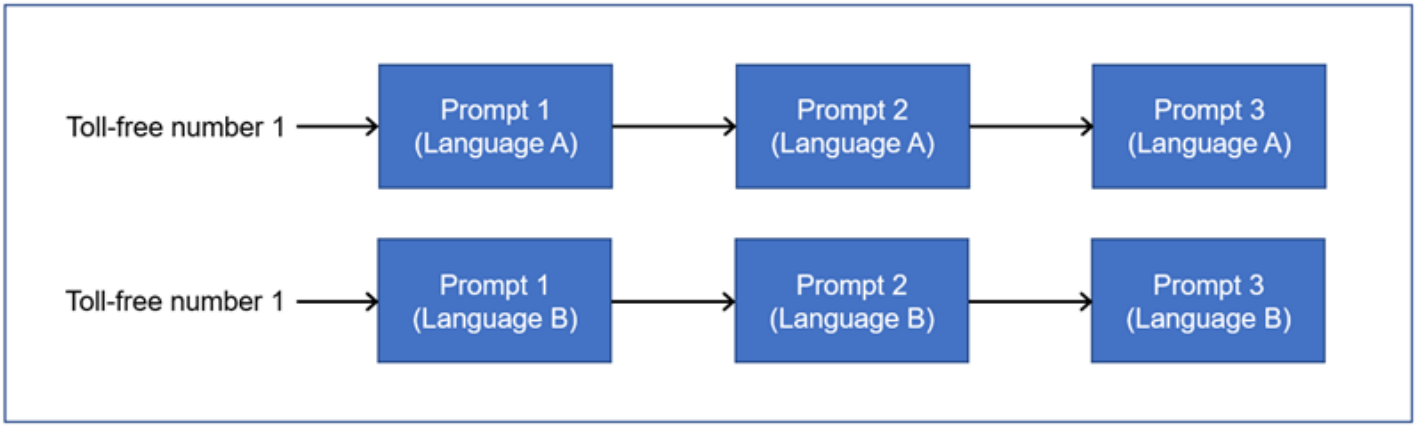
Un error común que cometen los desarrolladores al diseñar un sistema de IVR es introducir información estática o lógica de codificación rígida en el flujo. Entre esta información estática se incluyen los mensajes de petición, el valor del siguiente flujo de llamadas que se va a invocar o el grupo de habilidades al que dirigir la llamada. También podría ser la dirección de los puntos de conexión de la API que se invocan desde una IVR o las variables devueltas, que luego se utilizan para ramificar la lógica en el sistema de IVR.

El uso de valores estáticos en el sistema de la IVR dificulta su seguimiento y cambio en tiempo real, e introduce redundancias en el flujo. La práctica recomendada es tener un diseño dinámico siempre que sea posible. Veamos un ejemplo de cómo diseñar un sistema de IVR en varios idiomas para comprenderlo mejor.

Enfoque estático:

Por lo general, al diseñar un sistema de IVR multilingüe, los desarrolladores identifican el idioma preferido del cliente y, a continuación, crean dos versiones del mismo flujo. La primera versión tiene los mensajes de la petición en codificación rígida en el idioma A y la segunda versión podría tener los mensajes en codificación rígida en el idioma B, como se muestra en el diagrama siguiente.

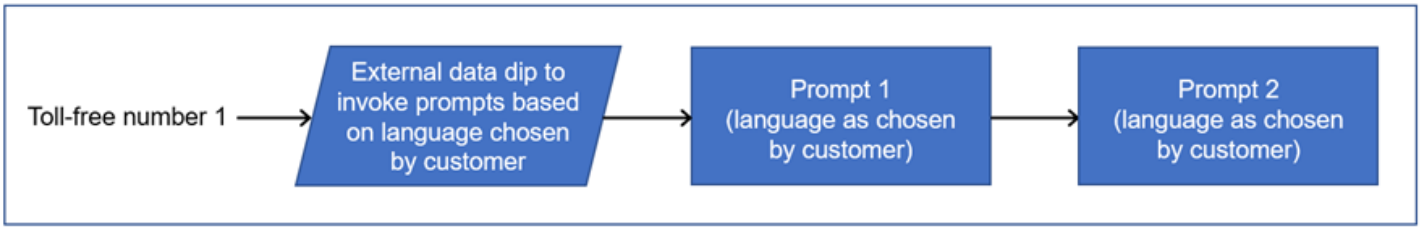
Este enfoque es difícil de escalar porque los desarrolladores tienen que mantener versiones distintas del mismo flujo. También tienen que crear una versión nueva del flujo cada vez que agregan un idioma nuevo.



Además, cualquier cambio que hagan en la lógica empresarial en un flujo debe replicarse de manera manual en las demás versiones. Esto hace que el diseño de la IVR sea confuso y no modular. También, aumenta el tiempo de comercialización de la innovación.

Enfoque dinámico:

El enfoque recomendado para resolver este problema es asignar mensajes de petición a las variables en lugar de utilizar valores estáticos en los bloques de las solicitudes del diseñador de la IVR. Según el idioma que elija el cliente, invoque los valores de las peticiones de una base de datos externa y rellene las variables para reproducir esos mensajes, como se muestra en el diagrama de flujo siguiente. Al utilizar este enfoque, puede mantener un solo flujo de IVR y aplicarlo de manera fácil a un número n de idiomas. Puede agregar las peticiones del idioma a la base de datos externa e invocarlas (mediante una API) según la selección del cliente en el sistema de IVR o de los datos que devuelve una consulta de CRM. De este modo también se protege al sistema de IVR de cualquier cambio en la lógica empresarial. Cuando utilice este modelo, asegúrese de tener una rama predeterminada con un conjunto mínimo de mensajes estáticos que se puedan utilizar en caso de que se produzcan errores en la base de datos o en las consultas.



Identificar los procesos repetibles

La identificación de patrones repetibles en la experiencia del cliente puede eliminar los componentes redundantes del sistema de IVR. Puede crear módulos o lógicas reutilizables para casos de uso repetidos e invocarlos según sea necesario en varios flujos de llamadas. Por ejemplo, si el equipo de ventas y el de servicio al cliente deben enviar una notificación por SMS en el sistema de IVR, puede crear un módulo de envío de SMS único y reutilizable que se pueda utilizar en ambas líneas de negocio. Con este enfoque, puede mantener un flujo único y las actualizaciones de ese módulo se reflejarán en todas las LOB de manera simplificada. Algunos otros ejemplos comunes de procesos repetibles son los siguientes:

- Asignación de la selección del idioma al flujo de la IVR según las entradas del cliente.
- Implementación del procesamiento de pagos en el sistema de la IVR.
- Lógica de autenticación de llamadas.
- Identificación de un cliente mediante una búsqueda en el sistema de CRM. Al modularizar el módulo de búsqueda de clientes en su sistema de CRM, puede convertir las conexiones personalizadas, que suelen utilizar lenguajes patentados, en bloques sencillos de IVR. Cualquier

persona que desarrolle experiencias de cliente podrá entonces acceder a estos bloques de IVR y utilizarlos.

- Realización del enrutamiento de mensajes de emergencia.

Ejemplo de caso de uso

Ahora, se examinará un caso de uso que ilustra cómo aplicar estos principios y obtener una arquitectura fundamental modular. Example Corp, una empresa global ficticia, es el punto central de nuestro ejemplo.

Example Corp quiere migrar a un centro de contacto basado en la nube para reducir el mantenimiento del centro de datos, centrarse en la innovación y reducir los costos. Una prioridad clave para ellos es el autoservicio, con el objetivo de mejorar la experiencia del cliente como parte de la migración. Example Corp es una organización global que administra varios números gratuitos (TFN) y números de llamadas entrantes directas (DID) según los países (Estados Unidos, Canadá, Italia, Países Bajos, etc.) y líneas de negocio (ventas, atención al cliente, productos nuevos, etc.). Esto significa que deben disponer de varios idiomas (como el inglés estadounidense, el francés canadiense y el italiano). Example Corp también quiere ofrecer experiencias personalizadas a las personas que llaman según el número marcado o del país de origen, lo que podría dar lugar a cambiar las opciones del menú u ofrecer mensajes de petición distintos en el sistema de IVR. Además, para cada país hay un conjunto concreto de habilidades (según el idioma, el ámbito de la actividad, etc.) que deben asignarse a los agentes para el desvío de las llamadas. En total, la empresa admite 25 países y 12 idiomas. También tienen previsto ofrecer el procesamiento de pagos y la posibilidad de comprar productos y complementos sin tener que hablar con un agente. Como práctica recomendada, Example Corp mantendrá los entornos de desarrollo, ensayo y producción para su centro de contacto.

En esta sección se muestra cómo los principios de diseño de la IVR que analizamos anteriormente en esta guía se pueden aplicar al caso de uso de Example Corp. Repasamos el proceso de identificar los objetivos de la organización, crear un recorrido de contacto con el cliente y crear un marco arquitectónico que cumpla con los requisitos de la empresa mediante AWS el uso de los servicios.

Identificar las metas de la organización

En el caso de Example Corp, entre las metas organizativas principales se incluyen las siguientes:

- Ahorro de costos: migre a un centro de contacto basado en la nube para reducir los costos de mantenimiento y mejorar la innovación.
- Autoservicio personalizado: ofrezca una experiencia personal y personalizada a los clientes según su número o país de origen, tal como las opciones de idioma.

- Tasa de contención de la IVR mejorada: simplifique el procesamiento de pagos en el sistema de IVR para que los clientes puedan comprar productos y complementos sin que tenga que intervenir un agente.
- management/innovation Centro de cambio mejorado: mantenga entornos separados (desarrollo, puesta en escena y producción) para el sistema IVR, a fin de gestionar los cambios y experimentar de forma eficaz.
- Mejora de la resolución en la primera llamada: implemente estrategias de enrutamiento eficaces según el idioma, el país y la intención del cliente.

Planificar el recorrido de contacto con el cliente

Para crear un recorrido de contacto con el cliente para Example Corp, debemos tener en cuenta los aspectos siguientes:

- Origen de la llamada: identifique el país y la línea de negocio asociados a la llamada entrante para ofrecer una experiencia personalizada.
- Selección de idioma: ofrezca un menú de idiomas adecuado según el país de origen, el perfil y los idiomas en los que habla Example Corp.
- Autenticación del cliente: autentique a la persona que llama mediante los detalles de su cuenta, datos biométricos de voz u otro método seguro.
- Personalización: recopile y utilice los datos de los clientes para personalizar la experiencia de la IVR. Por ejemplo, salude a la persona que llama por su nombre o proporcione opciones de menú personalizadas según el historial de su cuenta.
- Self-service opciones: ofrezca opciones de autoservicio para responder a las consultas habituales de los clientes, como consultas de saldo, actualizaciones del estado de los pedidos o restablecimiento de contraseñas.
- Procesamiento de pagos: integre una puerta de enlace de pago segura para que los clientes puedan comprar en el sistema de la IVR.
- Integraciones de backend: tenga en cuenta las integraciones de la API de backend para implementar los procesos de autoservicio, como saludos personalizados, novedades de los pedidos y pagos. Compruebe la disponibilidad y preparación de estas API.
- Seguridad y cumplimiento normativo: determine la información para la que es necesario un cifrado superior, enmascaramiento o inhabilitación de los registros (por ejemplo, la información de las tarjetas y la información de identificación personal del cliente).

- Procesos repetibles: identifique los procesos, como el pago y la autenticación de las personas que llaman, que se repetirán en la mayoría de los países y líneas de negocio.

Crear la arquitectura fundamental

Para crear un marco arquitectónico modular y dinámico para el sistema de IVR de Example Corp, considere la posibilidad de eliminar el contenido estático y modular los procesos repetibles.

Eliminar el contenido estático

- Considere la posibilidad de utilizar las variables para invocar las indicaciones de manera dinámica según los números gratuitos (TFN) asignados a cada país. Por ejemplo, si TFN1 pertenece a Canadá, puede invocar la petición correspondiente de la base de datos externa para proporcionar al cliente el inglés y el francés canadiense como opciones de idioma. Según la selección del cliente, puede invocar las peticiones en inglés (o en francés canadiense) de la base de datos externa y reproducirlas durante todo el flujo de llamadas de la IVR. Este enfoque utiliza un solo flujo de IVR y elimina las redundancias en todos los TFN.
- Considere almacenar las colas de agentes y las habilidades a las que se debe dirigir la persona que llama, el siguiente árbol de IVR por el que debe pasar la llamada y los valores de los puntos de conexión de la API, como las URL y los nombres de recursos de Amazon (ARN), en una base de datos externa. También puede invocarlas según el TFN asignado a un país o línea de negocio.
- Para simplificar aún más la experiencia, almacene los detalles del entorno de implementación en la base de datos externa. Por ejemplo, si el TFN2 está asignado a un entorno de desarrollo, puede invocar las peticiones de idioma correspondientes, los puntos de conexión de la API, las colas de agentes y las habilidades que sean pertinentes solo para ese entorno. Si el TFN3 está asignado a un entorno de ensayo, puede invocar los detalles de las llamadas que se asignaron al entorno provisional, etc. Este enfoque agiliza de manera considerable el mantenimiento y el desarrollo del flujo de llamadas. Los desarrolladores pueden mantener un solo flujo de llamadas y pueden modificar la experiencia de manera fácil según los parámetros clave, como el TFN asignado a un país, la línea de negocio o el entorno.

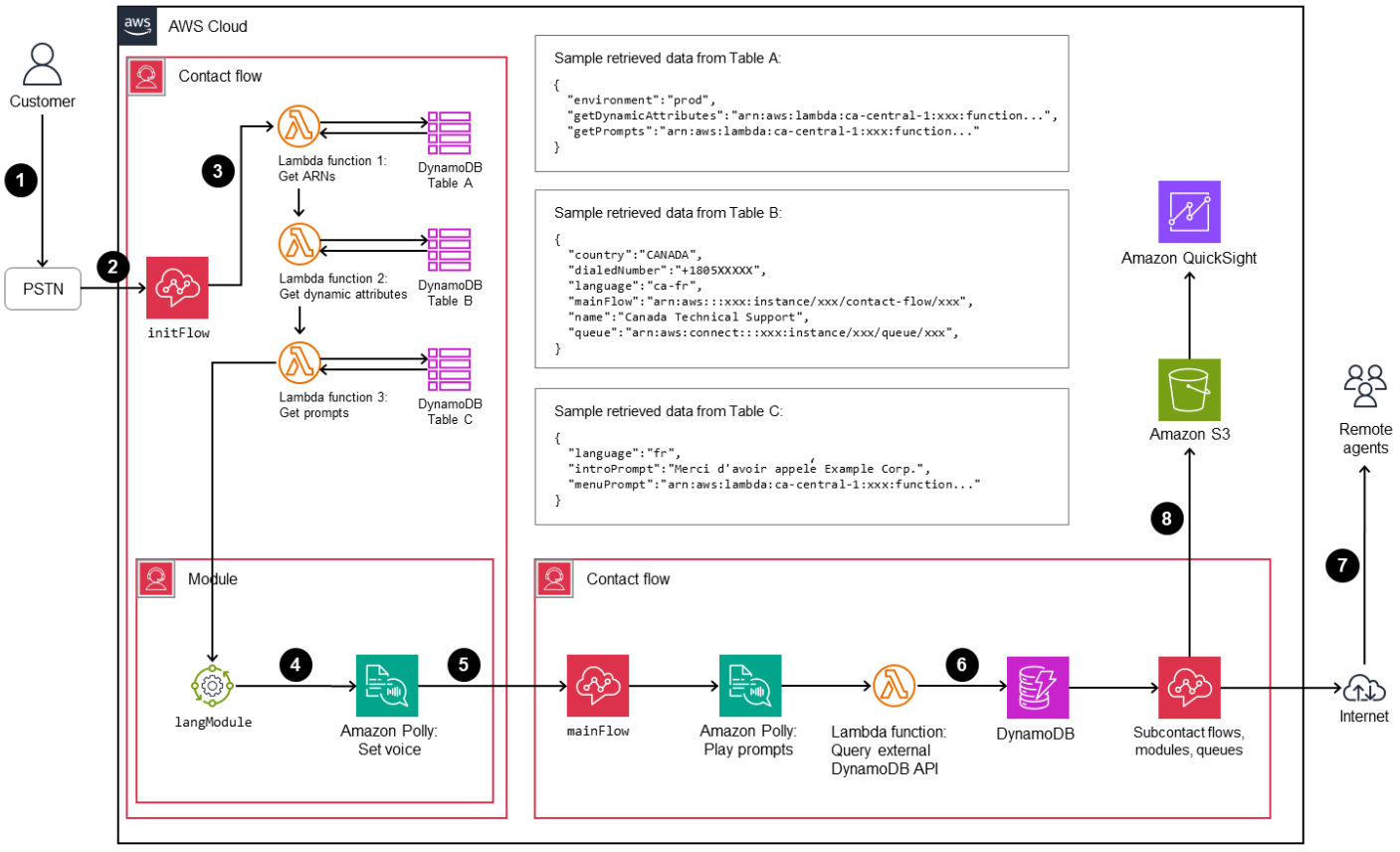
Identificar los procesos repetibles

En el caso de Example Corp, algunos de los procesos repetibles identificados en todos los países y líneas de negocio son la selección del idioma de las personas que llaman y el procesamiento de

pagos. Puede crear estos procesos como módulos para que sea fácil centralizarlos, mantenerlos e invocarlos en todos los flujos de llamadas.

En el diagrama siguiente se muestra un ejemplo de arquitectura basado en la metodología de diseño de la IVR que se describe en esta guía. Esta arquitectura utiliza los flujos y módulos de Connect Customer con los siguientes AWS servicios adicionales:

- Las tablas de Amazon DynamoDB almacenan los ARN, los atributos dinámicos y las peticiones.
- AWS Lambda las funciones recuperan información de las tablas de DynamoDB.
- Amazon Polly convierte las peticiones de texto en voz.
- Amazon Simple Storage Service (Amazon S3) almacena los registros de contacto de los clientes de Connect.



El sistema de IVR implementa los pasos siguientes:

1. Un cliente llama al DID o TFN específico del país de la empresa.
2. La llamada entra en un flujo de inicialización de Connect Customer (`initFlow`).

3. El flujo de inicialización llama a tres funciones de Lambda que recuperan los datos de tres tablas de DynamoDB:
   - La primera función recupera los ARN de las instancias de Connect Customer en función del entorno al que se realizó la llamada (como producción, puesta en escena o desarrollo).
   - Una segunda función recupera los atributos dinámicos, como el país, el idioma y la cola de servicio al cliente, según el número marcado.
   - Una tercera función recibe los mensajes de petición de la VR según el idioma que seleccione el cliente.
4. El módulo de idioma Connect Customer (`langModule`) establece una voz de Amazon Polly en función del idioma seleccionado.
5. A continuación, la llamada pasa al siguiente flujo de contacto (`mainFlow`), que reproduce peticiones dinámicas, como un mensaje de bienvenida o las opciones del centro de llamadas, que se recuperaron de la tercera tabla de DynamoDB.
6. Una función de Lambda llama a una API de DynamoDB para invocar los flujos, los módulos y las colas de subcontactos.
7. La llamada podría enviarse a un agente del centro de atención al cliente lo que seleccione este en el sistema de la IVR. La pantalla del agente se rellena de manera dinámica con los datos que se capturaron, como el país del cliente, su identidad y el motivo de la llamada.
8. Los datos de contacto del cliente también se almacenan en un bucket de S3 como parte de los registros de contacto de Connect Customer y se pueden utilizar para generar informes personalizados con Amazon Quick.

Para obtener información sobre cómo se utilizó este diseño en un escenario real, consulte la presentación de re:Invent 2020 Cómo Best Western creó un centro de contacto modular y dinámico mediante Connect Customer. Para ver un ejemplo de código, consulte el proyecto Dynamic Contact Center en. GitHub

Conclusión

Esta guía proporciona los principios de diseño de IVR para ayudar a las organizaciones a optimizar y escalar su enfoque de la arquitectura de IVR, además de cumplir los objetivos organizacionales. También proporcionó una arquitectura de muestra basada en un caso de uso específico. Puede utilizarla como plantilla de inicio para crear su propio diseño de IVR.

Recursos

- Documentación de Connect Customer
- AWS Lambda Guía para desarrolladores
- Guía para desarrolladores de Amazon DynamoDB
- Guía para desarrolladores de Amazon Polly
- Guía para desarrolladores de Amazon Lex
- Ejemplo de código para implementar un IVR modular (GitHub proyecto)
- Cómo Best Western creó un centro de contacto modular y dinámico mediante Connect Customer (Troy Tracy y Ayesha Borker, presentación sobre re:Invent 2020)

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones RSS.

Cambio	Descripción	Fecha
Publicación inicial	—	28 de agosto de 2023

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.