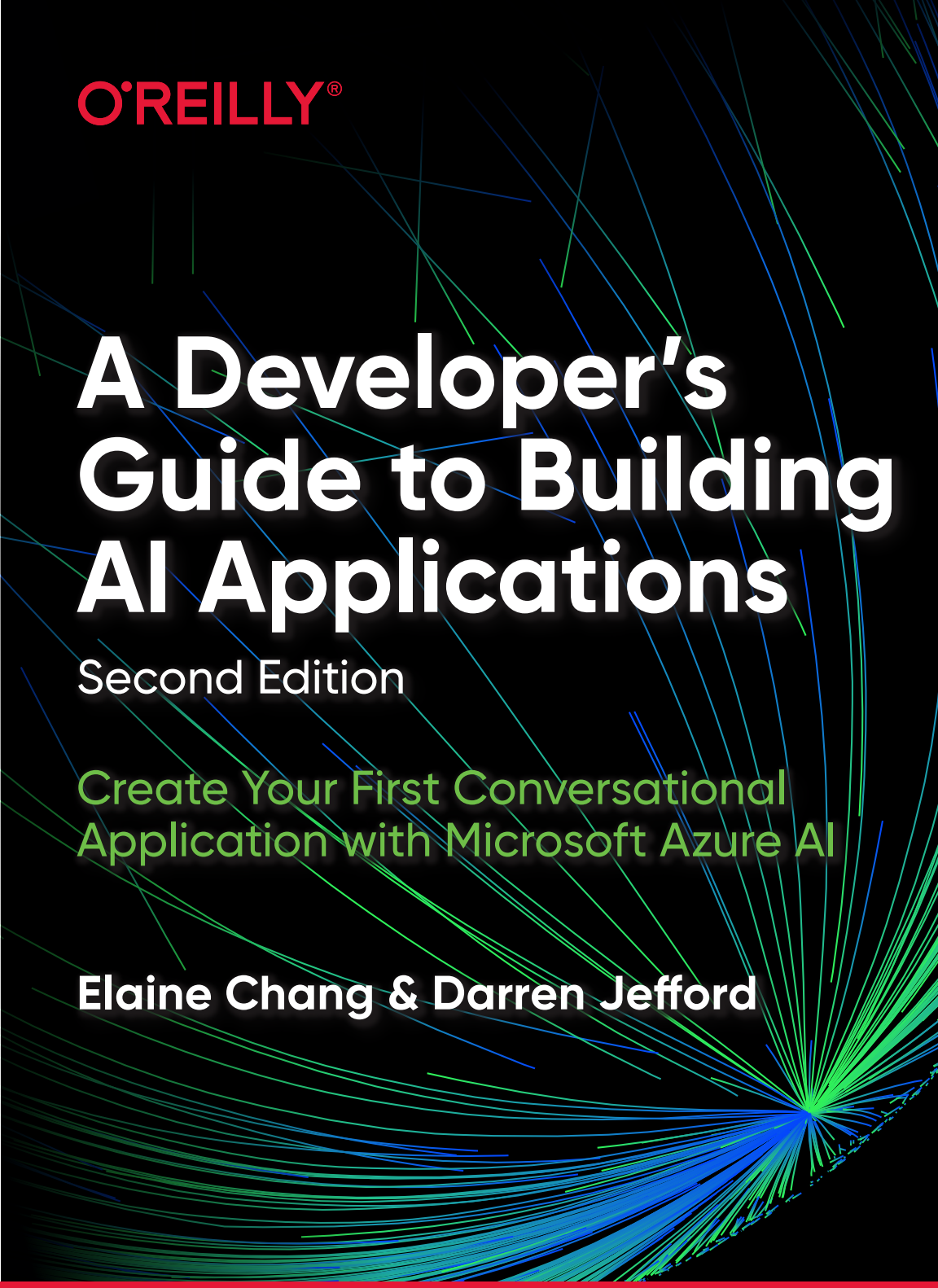
O'REILLY®
A Developer's Guide to Building AI Applications
Second Edition
Create Your First Conversational Application with Microsoft Azure AI
Elaine Chang & Darren Jefford
REPORT

SECOND EDITION

# A Developer's Guide to Building AI Applications

## *Create Your First Conversational Application with Microsoft Azure AI*

*Elaine Chang and Darren Jefford*

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

**A Developer's Guide to Building AI Applications**
by Elaine Chang and Darren Jefford



Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

**Acquisitions Editor:** Rebecca Novack
**Development Editor:** Nicole Taché
**Production Editor:** Christopher Faucher
**Copyeditor:** Charles Roumeliotis
**Proofreader:** Athena Lakri
**Interior Designer:** David Futato
**Cover Designer:** Karen Montgomery
**Illustrator:** Rebecca Demarest

July 2018: First Edition
April 2020: Second Edition

**Revision History for the Second Edition**
2020-04-17: First Release



978-1-492-08060-2

[LSI]

# Table of Contents

# Acknowledgments

We would like to thank the following individuals at Microsoft who contributed to the report as advisors and technical reviewers:

Lili Cheng, Anand Ramen, Ben Brown, Chris Mullins, Cindy Noteboom, Deborah Harrison, Dewain Robinson, Em Ivers, Lauren Mills, Patrick Volum, Robert Standefer, Ryan Lengel Isgrig, Steve Sweetman, Ted Li, Tyler Mays-Childers, William Mendoza, and Prem Prakash.

# Foreword

The creators of the modern computer wanted to augment human intellect and expand our capabilities beyond the limitations of our collective brainpower. When researchers and computer scientists in the 1950s conceived of the "thinking computer," they started a rapid evolution toward AI, laying the groundwork for technology that would have a tremendous impact on the world for decades to come.

AI is now everywhere. People do not even realize that AI is powering their experiences. While AI is now present in practically everything we use, from thermostats to sprinkler systems, it is also powering opportunities in new frontiers, such as autonomous vehicles, pharmaceutical research, and precision agriculture. AI is an exciting frontier for developers to create powerful and innovative ways to approach hard-to-solve problems.

While AI was once solely the domain of researchers and institutions, the recent availability of AI infrastructure, platforms, and services means that all the power of AI is now available to developers to build solutions with intelligence. Cloud AI services and tools make developing AI-powered solutions possible—including machine learning, knowledge mining, intelligent agents, and pretrained models—without requiring specialized knowledge.

The past few years have seen a leap in the adoption of AI, catalyzed by vast amounts of digital data, application services, and enormous computing power. As a result, AI technologies such as natural-language understanding, sentiment analysis, speech recognition, and image understanding can now power applications across a broad range of industries.

One of the most compelling applications of AI is in making our everyday lives better and easier. Since the development of computing, people have envisioned having meaningful dialogs with computers, expressing our needs and ideas in the ways we communicate with each other using natural language: say something to the computer, and it would respond to you. Conversational AI shifts the interaction model from domain-specific, machine-driven commands to conversational interfaces that focus on people and expression. With conversational AI, developers can make computers communicate like people, by recognizing words, understanding intent, and responding in ways that feel natural and familiar.

*— Lili Cheng, Corporate Vice President*
*Microsoft Conversational AI*

# A Developer's Guide to Building AI Applications

## Introduction

In this book, we look at the requirements for applying well-tested AI solutions to everyday problems. To help you explore the possibilities of AI, we will show you how to create a Virtual Assistant, a conversational AI application that can understand language, perceive vast amounts of information, and respond intelligently. Along the way, we will share the many AI resources and capabilities that are available to developers.

Here is a roadmap to the contents of this book:

*"The Intersection of Data, AI, and the Cloud"*
: This section explains the technological basis for this book and why these technologies are increasingly offered in the cloud.

*"Microsoft Azure AI"*
: This section introduces the Microsoft Azure AI platform with a variety of services, infrastructure, and tools to empower developers to build AI apps and agents and add knowledge mining and machine learning capabilities. This book focuses on conversational AI applications and provides pointers to additional resources for other areas of Azure AI.

*"Conversational AI"*
: This section discusses the evolution of natural language processing, Microsoft's Language Understanding service (formerly named LUIS) and Bot Framework ecosystem, common use

cases of conversational AI, and the development life cycle of conversational AI applications.

*"Core Features of Virtual Assistants "*
: This section highlights core features of Virtual Assistants, including complete control for developers, prebuilt and reusable Conversational Skills, flexible integration and contextual awareness, business insights captured and distributed, multimodal input, Aaptive Cards, and enterprise capabilities. Bot Framework Virtual Assistant solution accelerator brings together best practices to enable organizations to deliver advanced conversational Assistant experiences tailored to their brand, personalized to their users, and made available across a broad range of apps and devices.

*"Developing Your Virtual Assistant"*
: This section provides guidance for creating your conversational AI application with Virtual Assistant solution accelerator, with pointers to online tutorials.

*"Connecting Assistants to Clients and Channels"*
: This section provides guidance for connecting your conversational AI application to clients and channels. A wide variety of channels and adapters are offered by Microsoft letting your conversational experiences reach your end users wherever they are.

*"Optional: Adding Intelligence to Your Assistant with Skills"*
: This section introduces Skills, a way to plug in features from the platform to your conversational experiences, with pointers to online tutorials.

*"Enabling Analytics for Your Virtual Assistant "*
: This section provides guidance for setting up analytics tools for your conversational AI application. Insights from the analytics dashboard of your conversational AI application can suggest important improvements for its effectiveness and performance.

*"Roadmap and More Resources"*
: Learn more about Microsoft's future investment in conversational AI.

*"Building Responsible AI "*
: We will conclude with a discussion of how the Azure AI platform encourages developers to create responsible and trustworthy solutions that treat people fairly.

## The Intersection of Data, AI, and the Cloud

Today we are enabling computers to learn from vast amounts of data and to interact and respond more naturally with the world, rather than following preprogrammed routines.[1] Consider the following capabilities of modern software:

*Computer vision*
: The ability to "see" by recognizing objects and their relationships within a picture or video, creating data from the physical environment.

*Speech recognition and synthesis*
: The ability to "listen" by understanding the words that people say and to transcribe them into text, and in reverse, to read text aloud in a natural voice.

*Language understanding*
: The ability to "comprehend" the meaning of words and respond, considering the many nuances and complexities of language (such as slang and idiomatic expressions). When computers can effectively participate in a dialog with humans, we call this *conversational AI*.

*Knowledge*
: The ability to "reason" by representing and understanding the relationship between people, things, places, and events.

How do these capabilities feature in enterprise applications? Through machine learning, AI-powered applications *reason* by unlocking large amounts of varied data—data that has been collected over time across repositories and held in massive datasets. These AI systems *understand* and create meaning in unstructured data such as email, chats, and handwritten notes, all of which could not have been previously processed by computers. More important,

1 Lili Cheng,"Why You Shouldn't Be Afraid of Artificial Intelligence," *Time*, January 4, 2018, *https://ti.me/2GEkknZ*.

these systems are now *interacting* with customers and engaging them in different channels, in ways that can be hyperpersonalized.

By applying these capabilities, businesses are using AI-powered applications to digitally transform every aspect of their organization. They are transforming their products through insights from customer data. They are optimizing business operations by predicting anomalies and improving efficiencies. They are empowering their employees through intelligent tools and engaging their customers through conversational agents that deliver more customized experiences.

To design technology for humans, it is important to understand the context of how people work, play, and live. Current AI solutions complement and unlock human potential and creative pursuits. And such tailored solutions must also be able to learn from and adapt to new external conditions, just as humans do.

One of the most fascinating areas of research is bridging emotional and cognitive intelligence to create conversational AI systems that model human language and have insight into the sometimes illogical and unpredictable ways humans interact. According to Lili Cheng, Microsoft's corporate vice president of conversational AI, "This likely means AI needs to recognize when people are more effective on their own—when to get out of the way, when not to help, when not to record, when not to interrupt or distract."[2]

Because datasets are growing, and because they vary wildly in size, it is becoming more important to offer developers quick access to flexible options for both storage and processing. Thus, organizations are turning more and more toward the cloud, which provides this range and flexibility. In addition, cloud vendors pack a rich and powerful toolbox to enable the AI capabilities we have discussed. Enormous connectivity allows any type of connected device to bring massive amounts of data into the cloud on a real-time basis for analysis and intelligent processing at scale. For developers, the cloud provides the necessary infrastructure and tools to offer enterprise-grade security, availability, compliance, and manageability for the business applications and services.

2 Lili Cheng,"Why You Shouldn't Be Afraid of Artificial Intelligence," *Time*, January 4, 2018, *https://ti.me/2GEkknZ*.

# Microsoft Azure AI

Microsoft's Azure AI platform aims to bring AI to every developer and to empower developers to innovate and accelerate their projects with a variety of services, infrastructure, and tools. Azure AI supports a variety of use cases and targets different levels of expertise and desired ways of working. For example, Azure offers the Azure Bot Service and Bot Framework SDK enabling developers to build rich conversational experiences. Additionally, Azure Cognitive Services offers developers domain-specific AI services available as APIs to build applications that can see, hear, and understand. Azure Machine Learning allows developers and data scientists to build custom AI models, with investments in the necessary hardware and infrastructure to support deep learning and machine learning framework and tools.

*AI apps and agents*

The industry-leading AI models being used today in Microsoft products such as Office 365, Teams, Dynamics 365, Cortana, Xbox, HoloLens, and Bing are available to your own apps through a platform of democratized cognitive services. Some of these models can be customized with your own data and run both offline and online.

The book *Building Intelligent Apps with Cognitive APIs* provides a closer look at what's behind apps that see, hear, speak, understand, and interpret people's needs.

The Azure Bot Service, along with the Bot Framework, enables developers to build advanced conversational experiences. The new Power Virtual Agents offering, available as part of the Power Platform, is built on top of Bot Framework and empowers nondevelopers to create conversational experiences and compose with other Bot Framework components.

*Knowledge mining*

Azure Cognitive Search works across many types of data to transform unstructured information into searchable content. Extract insights and structured information, discover patterns and relationships, reveal sentiment, and more.

The whitepaper "Extracting Actionable Insights from All Your Content" covers how knowledge mining works, use cases,

industry-leading solutions, and additional resources for those who want to get started with knowledge mining.

*Machine learning*
: Developers can get access to the advanced machine learning capabilities of the Azure AI through Azure Machine Learning (AML) services. AML is a managed cloud service where you can train, manage, and deploy models in the cloud or to edge devices using Python and tools like Jupyter notebooks. You can even deploy TensorFlow image classification and recognition models, using a variety of deep neural networks, to Microsoft's Project Brainwave FPGA hardware in Azure for inference and training, which offers extremely high-scale throughput and low latency.

  The book *Thoughtful Machine Learning with Python: A Test-Driven Approach* provides a starting point for AI programming that can be useful for readers interested in using AML.

To help you get started with Azure AI, you can leverage the resources available on the Azure AI website.

In this book, we will be focusing on showing how you can build a conversational AI application using Bot Framework.

## Conversational AI

Natural language processing (NLP) gives computers the ability to read, understand, and derive meaning from human language. Since the 1950s, computer scientists have been working on the challenges of NLP, but limitations in computing power and data sizes hindered advancements in processing and analyzing textual components, sentiments, parts of speech, and the various entities that make up natural language communication.

That changed in the 2010s. Advances in cloud computing, machine learning, and the availability of vast amounts of text and conversational data from messaging systems, social media, and web chats have helped us make immense progress in NLP. The advancements in NLP have made it possible for computers to not only identify words in text but also to understand the meaning behind those words and the relationships between them.

NLP works by analyzing a large body of human-generated text and turning it into machine-readable data. NLP identifies and extracts key metadata from the text, including:

*Entities*
: NLP identifies entities in text like people, places, and things. Entities can also be pieces of information requiring special extraction, such as dates and times.

*Relations*
: NLP identifies how entities are related using semantic information.

*Concepts*
: NLP extracts general concepts from the body of text that do not explicitly appear. For example, the word "excel" might return concepts like "productivity tools" and "numbers," even if these terms do not appear in the text. This is a powerful tool for making connections that might not seem obvious at first glance.

*Sentiment*
: NLP scores the level of positivity or negativity in the text. This is useful, for example, to gauge sentiment related to a product or service. Or, in a customer support context, this functionality is helpful when determining whether to route a chat to a human (upon detecting negativity).

*Emotions*
: This is sentiment analysis at a finer granularity. In this case, NLP classifies not just "positive" and "negative" but "anger," "sadness," and "joy."

*Keywords*
: NLP extracts keywords and phrases to use as a basis for indexing, searching, and sorting.

*Categories*
: NLP creates a hierarchical taxonomy for what the data is about and places this taxonomy in a high-level category (text classification). This is useful for applications like recommending relevant content, generating ads, organizing emails, and determining the intent of a user.

In the past, you might have tried to simulate NLP-style capabilities through rule-based approaches, such as regular expressions or

decision trees, which struggled at scale to understand the intent of questions from a human. Or you might have used custom machine learning models, which required access to specialized expertise, large datasets, and complex tools, limiting their implementation to only large organizations with the resources to invest.

Now, consider where we are today. Easy-to-use APIs in the cloud provide NLP capabilities that are powering the widespread use of conversational AI. From the rise of open source tools to the arrival of cloud APIs, NLP capabilities that were once solely in the domains of academia and the research community are now available to a wider audience across industries.

## Language Understanding (formerly named LUIS)

Language Understanding, a service developed by Microsoft, enables developers to build applications that can take user input in natural language and extract structured information, including meaning and intent. Language Understanding is a machine learning–based service to build natural language experiences, and it enables you to quickly create enterprise-ready, custom models that continuously improve.

With Language Understanding, you can use a prebuilt model (e.g., weather, calendar), customize an existing one, or build your own from scratch. A model begins with a list of general user intents that represent the tasks or actions the user would want to perform, such as "book a flight," "schedule meeting," or "contact help desk." After you identify the intent, you provide example phrases, called utterances, for the intent. Then, you label the utterances with the specific details you want Language Understanding to pull out of the utterance. The data that is pulled out of the utterance is an entity.

An entity represents detailed information that is relevant in the conversation. By recognizing and labeling the entities that are mentioned in the user's input, Language Understanding helps you choose the specific action to take to answer a user's request. You can define your own entities, such as domain-specific terminology, or extract prebuilt common entities, such as dates and times, proper names, measurements, and numbers. With prebuilt domains, you have a set of entities and utterances for common categories like calendar, entertainment, communication, and home automation.

Language Understanding also allows developers to continuously improve the app through active learning. Language Understanding stores user queries and selects utterances that it is unsure of. You can then review the utterances, select the intent, and mark entities for real-world utterances. This retrains the language model with more data.

The service integrates with other AI tools in the cloud to power natural language processing and understanding in apps, bots, and Internet of Things (IoT) devices. Through its Bot Framework, Microsoft incorporates Language Understanding and other cognitive services for the development of bots.

## Bot Framework Ecosystem

Microsoft Bot Framework (Figure 1) has an ecosystem of tools and services that provide a comprehensive experience for building conversational AI applications.

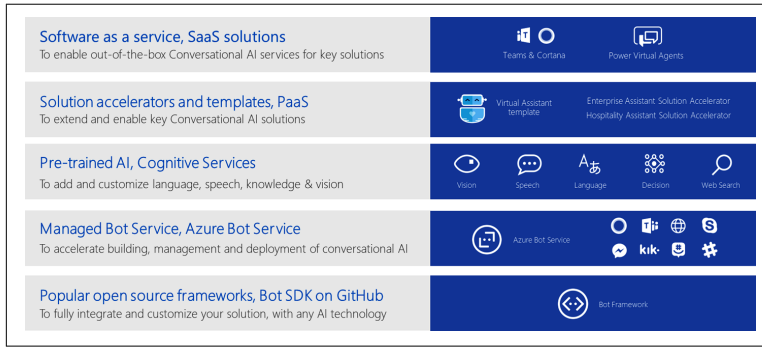


*Figure 1. Bot Framework ecosystem*

With Bot Framework SDK, developers can easily model and build sophisticated conversation using their favorite programming languages. Developers can build conversational AI applications that converse free-form or can also have more guided interactions where the application provides user choices or possible actions. The conversation can use simple text or more complex rich cards that contain text, images, and action buttons. Developers can add natural language interactions and questions and answers that let users interact with bots in a natural way.

Azure Bot Service enables you to host intelligent, enterprise-grade conversational AI applications with complete ownership and control

of your data. Developers can register and connect their bots to users on Microsoft Teams and Web Chat, Facebook Messenger, and more.

To add more intelligence to a conversational AI application, you can add and customize pretrained API models and Cognitive Services, including language, speech, knowledge, and vision capabilities.

Bot Framework also provides a set of solution accelerators and templates to help build sophisticated conversational experiences. The Virtual Assistant solution accelerator brings together all the supporting components and greatly simplifies the creation of a new project including basic conversational intents, dispatch integration, QnA Maker, Application Insights, and an automated deployment.

The Power Virtual Agents offering builds on top of the Bot Framework platform, providing a no-code graphical interface to create conversational experiences.

## Conversational AI Use Cases

Customers familiar with virtual assistants and messaging apps are engaging more and more with conversational interfaces, which can present a more natural experience where humans express their needs through natural language and quickly complete tasks. For a lot of companies, conversational AI applications are becoming a competitive differentiator. Many organizations are strategically making bots available within the same messaging platforms in which their customers spend time. Organizations around the world are transforming their businesses with conversational AI, which can promote more efficient and natural interactions with both their customers and their employees. Here are a few common use cases:

*Customer support*

Organizations are using conversational AI to transform their customer relationships by providing easy and natural interactions across multiple channels and platforms, such as home devices, mobile apps, social channels like Facebook Messenger, and websites. Conversational experiences not only allow organizations to reach their customers wherever they are but also to personalize and continually improve their interactions.

Insurance companies, for example, are making it easier for customers to get quick answers to commonly asked questions, submit claims, or even generate a quote for an insurance plan.

Retail companies are also allowing users to quickly track packages and get order status updates, while still allowing for a customer to be transferred to chat with a human agent. Telecommunications companies are using virtual assistants with AI capabilities to learn more about customers to deliver rich customized interactions, grow revenue, and increase customer support teams' productivity.

*Enterprise Assistant*

Organizations are using conversational AI to improve employee engagement, connecting people, tasks, information, and services more effectively with more natural and intuitive interfaces. By integrating employee assistants with voice and text interfaces into enterprise devices and existing conversation canvases (e.g., Microsoft Teams, Slack, and Web Chat), organizations speed up the process of managing calendars, finding available meeting rooms, finding people with specific Skills, or contacting HR. Integration with Dynamics, PowerApps, ServiceNow, and other IT providers simplifies accesses for employees and allows them to easily find the data and perform the tasks that they are looking for. Integration into searches adds the power to provide enterprise data in a natural way for users as well.

*Call center optimization*

Integrating a conversational experience into a call center telephone communications system can reduce call times with human agents by clarifying information in advance or resolving simple requests without the need for a human agent. In addition, the solution replaces classic interactive voice response (IVR) solutions with a modern conversational experience and enables a consistent user experience through the duration of the call, or until hand-off to a human agent.

Post-call analysis assesses call quality and customer feedback, with insights available to improve the call flow and optimize the user experience, increase first contact resolution, and meet other key performance indicators (KPIs).

The same assistant can be exposed through additional text-only channels, enabling end users to interact through their channel of choice and increasing the payoff of the investment by ensuring all users—whether they are using SMS or richer channels—can participate.

*In-car voice assistant*

Voice-enabled assistants integrated into cars provide drivers and passengers the ability to perform traditional car operations (e.g., navigation, radio) along with productivity-focused scenarios such as moving meetings when your running late, adding items to your task list, and proactive experiences where the car can suggest tasks to complete based on events such as starting the engine, traveling home, or enabling cruise control. Other use cases include scheduling service for a vehicle based on a user's preferences for service provider, vehicle location, provider schedule availability, severity of issue, loaner preference, both personal and work schedules, and many more variables. This is the power of bringing an automotive supplier's data into the picture and illustrates the fully integrated experience possible through the Virtual Assistant solution.

*Hospitality assistant*

A Virtual Assistant integrated into a hotel-room device can provide a broad range of hospitality-focused scenarios: extending a stay, requesting late checkout, room service, concierge services, and finding local restaurants and attractions. The app can be linked to a productivity account, opening up more sophisticated experiences such as alarm calls, weather warnings, and learning patterns across stays.

These are some examples of the types of conversational AI applications we will be focusing on building in this book. Let's now look at the typical workflow for developing a conversational AI application.

## Development Workflow of Conversational AI Applications

The typical workflow for developing a conversational AI application resembles other kinds of projects: the major phases are *design*, *build*, *test*, *deploy*, *connect*, and *evaluate* (Figure 2).[3]

3 These phases are described further in line Azure documents.

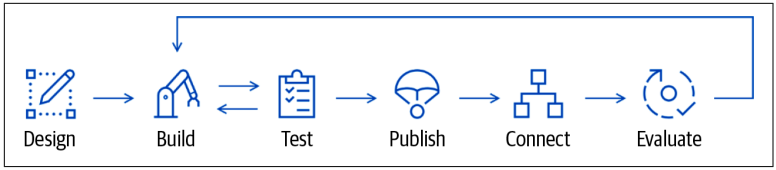


*Figure 2. The typical workflow for developing a conversational AI application*

Let's look at each phase in this workflow.

### Design

Developing a bot, like developing websites and applications, should start with a design for a great experience. When humans interact with bots, we expect that what we say is understood, what we receive as a response is appropriate, and what we get as a service is delightful. We expect that, if we leave mid-conversation, the bot will remember where we left off.

Your bot represents your brand, products, and services for your customers and employees, so it is imperative to start with a design-led approach to ensure that the goal of the bot meets the explicit or latent need of the human it serves. To design a delightful experience, we recommend the best practices of researching targeted users, defining bot personas, storyboarding bot scenarios, designing conversation flow, and defining an evaluation plan, *without* specifying technical development details.

For each of these design activities, here are the key questions to answer:

*Researching targeted users*
: Who are your users? What are their objectives, needs, and expectations? What is the context for their interaction with the bot? What does their environment look like? How will your bot help them? What services should your bot provide them?

*Defining bot personas*
: What should your bot look like (for instance, an avatar)? What should it be named? Does the bot carry out your organization's values? What is your bot's personality? Does your bot have a gender? Can it respond to off-topic questions? What tone of voice should your bot use? How would your bot handle

different situations? How should your bot respond (with proactive, reactive, or exception management)?

*Storyboarding bot scenarios*
: What is the user journey for your bot's targeted users? What should your bot do and not do? What are the goals and priorities of your bot's use cases?

*Designing conversation flow*
: What conversation flows can you expect for your main use cases? Simple Q and A, push notifications, step-by-step instructions, or more complex interactions?

*Defining an evaluation plan*
: How would you measure success? What measurements do you want to use to improve your service, and where should you insert instrumentation?

Before writing code, review the bot design guidelines from Microsoft's Bot Framework documentation for best practices.

The Bot Framework provides a set of tools for the design phase, including:

- *.chat* files to create a mockup of conversations between the user and the bot for specific scenarios
- The `bf chatdown` command to convert *.chat* files into rich transcripts
- Bot Framework Emulator, which opens a *.transcript* file to view a realistic rendering of the conversations (Figure 3)

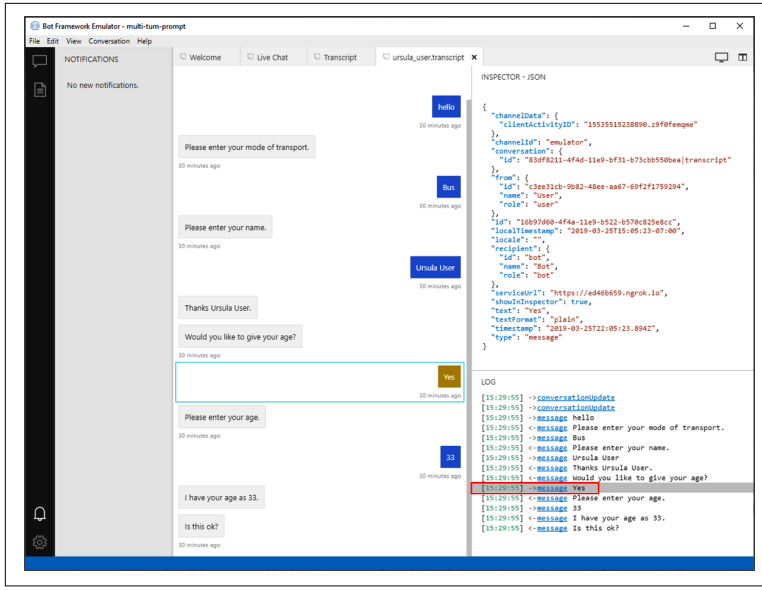


*Figure 3. View of a* .transcript *file in Bot Framework Emulator*

## Build

A bot is a representational state transfer (REST) web service that communicates with the user by sending and receiving messages and events from conversational interfaces like chat rooms or Web Chat widgets. With Microsoft's Azure Bot Service and Bot Framework, you can create bots in a variety of development environments and languages. You can start your bot development in the Azure portal or use one of the Bot Framework SDK templates for local development. The templates support the C#, JavaScript, and Python languages with Java support in early preview at the time of writing.

After you build the basic bot, extend its functionality in the ways your design calls for. You can add NLP capabilities using Language Understanding, add a knowledge base to answer common questions using QnA Maker, add capabilities to manage complex conversation flows and multiple knowledge domains using the Dispatch tool, and add graphics or menus using Adaptive Cards. Additionally, Microsoft provides command-line tools to help you create, manage, and test these bot assets as part of a DevOps process.

You may access a variety of samples that showcase the conversational capabilities available through the SDK, including basic dialog

capabilities such as multiturn dialogs through more advanced capabilities such as proactive messaging and authentication.

In addition, Microsoft provides a more advanced Virtual Assistant template, which is recommended as a starting point for building a more sophisticated conversational experience. It brings together many best practices for building conversational experiences and automates the integration of components that have been found to be highly beneficial by Bot Framework developers.

For example, a conversational experience built on the Virtual Assistant template allows developers to handle multiple languages, NLP models for base conversational intents, custom personalities to answer more general questions, integrated language generation for more natural responses, an introduction experience for new users, context switching, and Skill support.

In the next section of this book, we will use the Virtual Assistant template to create a conversational AI application.

#### Test

To test your conversational AI application, Microsoft provides the Bot Framework Emulator enabling developers to test conversations quickly and easily. You can also write unit tests using the Bot Framework SDK, which can focus on functionality testing of specific dialogs. Once configured through the Azure portal, your bot can be reached through a web chat interface, enabling broader testing by end users early in your development process.

#### Publish

When you are ready for your bot to be available on the web, either publish your bot to Azure or to your own web service or data center —wherever a normal web application can be hosted.

#### Connect

Azure Bot Service does most of the work necessary to connect your bots to a range of channels and devices. Configured through the Azure portal, you can connect your bots to Facebook Messenger, Slack, Microsoft Teams, Cortana, email, Telegram, Twilio, LINE, and other channels. You can also use Web Chat widgets to embed your bots in your websites or mobile applications.

You can use the Direct Line channel to connect your bot to your own client application, or the Direct Line Speech channel that enables low-latency speech interfaces with client applications using the Microsoft Speech SDK. Thus, you can embed text and speech experiences into desktop applications, mobile apps, and devices such as cars, speakers, and alarm clocks.

Bot Framework and members of the open source community also provide code-based adapters to connect your bots to other channels, such as Google Assistant, Amazon Alexa, Webex Teams, websockets, and webhooks.

#### Evaluate

Recordings of conversations between bots and users provide valuable business insights to help you evaluate your bot's performance. At this phase, best practices include evaluating success metrics that you defined during the design phase, reviewing instrumentation logs, collecting user feedback, refining, and iterating. Bot Framework provides sample Application Insights queries and a Power BI dashboard to help you grasp the full breadth of your bot's conversations with users and gain key insights into your bot's health and behavior.

## Core Features of Virtual Assistants

Building on the Bot Framework SDK, Microsoft's open source Virtual Assistant solution (available in C# and TypeScript) is a project template that encompasses the best practices for developing a bot on the Microsoft Azure platform.

Organizations are seeing a greater need to deliver advanced conversational assistant experiences tailored to their brand, personalized to their users, and made available across a broad range of apps and devices. With the Virtual Assistant, you control the name, voice, and personality to suit your needs. Bot Framework provides solutions that simplify the creation of a Virtual Assistant, enabling you to get started and extend your bot with a broad range of end-to-end development tooling.

The Virtual Assistant brings together Bot Framework, Azure Bot Service, and Language Understanding inside the Azure AI platform to simplify building your own Virtual Assistant (refer to Figure 1). Bot Framework and Azure Bot Service provide core conversational

capabilities for the Virtual Assistant, including dialog management, natural language prompts, context switching, memory, and language generation. The Virtual Assistant provides additional capabilities and prepackaged sets of domain-specific interactions called *Skills* to help organizations build their own assistant experiences in a variety of languages—for example Calendar and ToDo.

The Virtual Assistant intends to make the developer's job easier and more productive. In this section, we will walk through some core features of the Virtual Assistant.

## Complete Control for Developers

With Virtual Assistant, all aspects of the user experience are owned and controlled by you. This includes the branding, name, voice, personality, responses, and avatar. Microsoft provides five chat personalities based on the Azure Cognitive Service QnA Maker, enabling you to tailor the bot's personality. The source code to the Virtual Assistant and supporting Skills are provided as samples for you to customize. Your Virtual Assistant will be deployed within your Azure subscription. Therefore, all data generated by your Assistant (questions asked, user behavior, etc.) is entirely contained within your Azure subscription. See Cognitive Services compliance and privacy details and the Azure section of the Trust Center for more information.

## Prebuilt and Reusable Skills

Common Virtual Assistant scenarios are provided as reusable Conversational Skills and include tasks like finding nearby points of interest, checking off an item on a to-do list, and replying to an email. Skills—delivered in source code form—are fully customizable and consist of language models for multiple natural languages, dialogs, and integration code. Additional Skills can be created and made available either through your own Assistant or through a broader Skill ecosystem. This enables you to curate the capabilities that make sense for your scenario and that work across industries. Because the Virtual Assistant leverages Azure Bot Service, you can give users access to your Assistant through any of the supported channels and adapters, letting you reach your end users wherever they are, and using UI/UX experiences with which they are already familiar and comfortable.

## Flexible Integration and Contextual Awareness

The Virtual Assistant architecture is flexible and can be integrated with existing speech or NLP capabilities, back-end systems, APIs, and devices to enable personalization and adapt responses as required for the users' context. The Assistant can also be aware of the device or channel through which the user is communicating, which can be used to optimize the experience (e.g., voice versus text) and enable the user to switch between different channels if needed.

## Business Insights Captured and Distributed

Bot Framework provides a rich set of out-of-the-box metrics enabling any conversational experience to collect data at the levels you desire, to let you understand better what your customers are asking and where you might have opportunities to improve the service.

Azure Application Insights capture end-to-end analyses across the entire architecture. Microsoft has also provided sample Power BI dashboards that serve up easy-to-read graphs showing popular conversations, conversation length, unique users, and other key statistics (Figures 4 and 5). You can further extend your insights pipeline with your own machine learning to provide customized AI experiences to your users. Like all data that is part of your bot, the metrics remain under your control, helping you comply with privacy regulations such as the General Data Protection Regulation (GDPR).

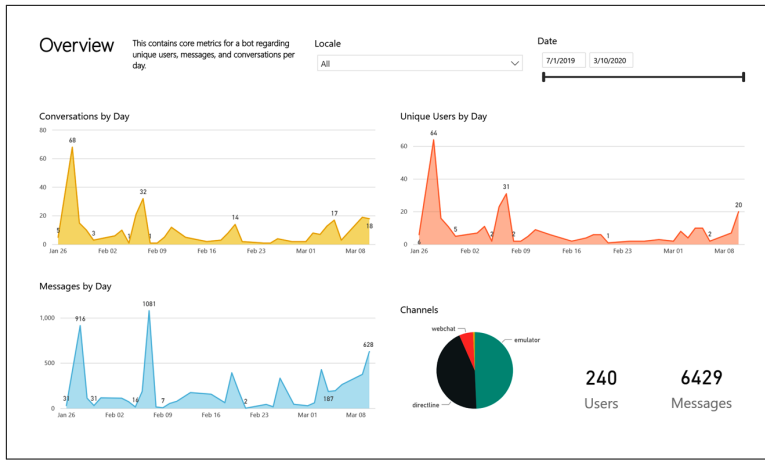


*Figure 4. Power BI dashboard for Bot Framework Virtual Assistant—Overview Tab*

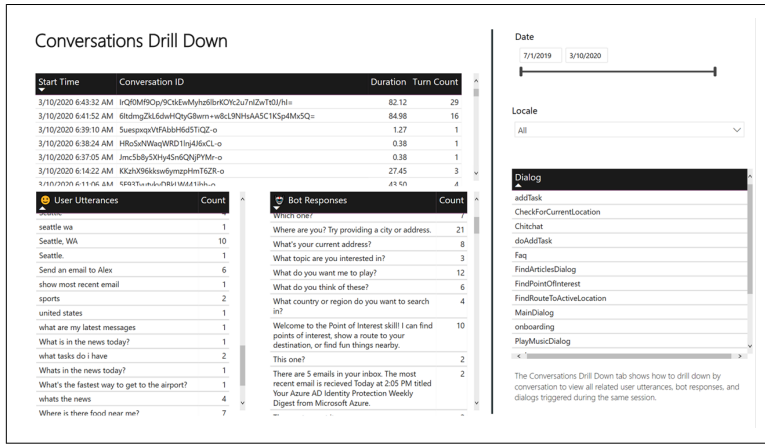


*Figure 5. Power BI dashboard for Bot Framework Virtual Assistant—Conversations Drill Down*

## Multimodal Input

The Virtual Assistant provides a range of input mechanisms: text, tap, and speech. This can be extended as needed to include vision through the integration of vision cognitive services. Additional input types can easily be integrated, depending on device or canvas capabilities. A Bot Framework–based conversational experience can

also be extended to support gestures (if available from the end user device), enabling users to switch between input types as desired.

The Virtual Assistant also integrates tightly with Speech service, part of the Cognitive Services family, along with NLP and dialog management—to enable contextual awareness of conversations. Real-time streaming of user audio enables NLP and dialogs to start as soon as the user finishes speaking, delivering a low-latency, more natural experience. The custom neural voice capability in the Speech service enables customers to develop highly realistic custom voices for natural conversational interfaces, starting with just 30 minutes of audio.

## Adaptive Cards

Adaptive Cards provide graphic capabilities such as cards, images, and buttons inside your Assistant. The cards are platform-agnostic pieces of UI, authored in JSON, that supported apps and services can exchange. When delivered to a specific app, the JSON is transformed into native UI that automatically adapts to its surroundings. It enables you to design and integrate lightweight UI for all major platforms and frameworks.

If the conversation canvas has a screen, these cards can be rendered across a broad range of devices and platforms, thus providing a UX that is consistent with the service or context in which the card is embedded. Devices that do not have screens can make use of the speech-friendly responses provided alongside the Adaptive Cards or any combination of delivery mechanisms appropriate to the context.

The Virtual Assistant and related Skills work comprehensively with Adaptive Cards, and their design and branding can be fully customized to suit your scenario. Figure 6 shows a few examples.

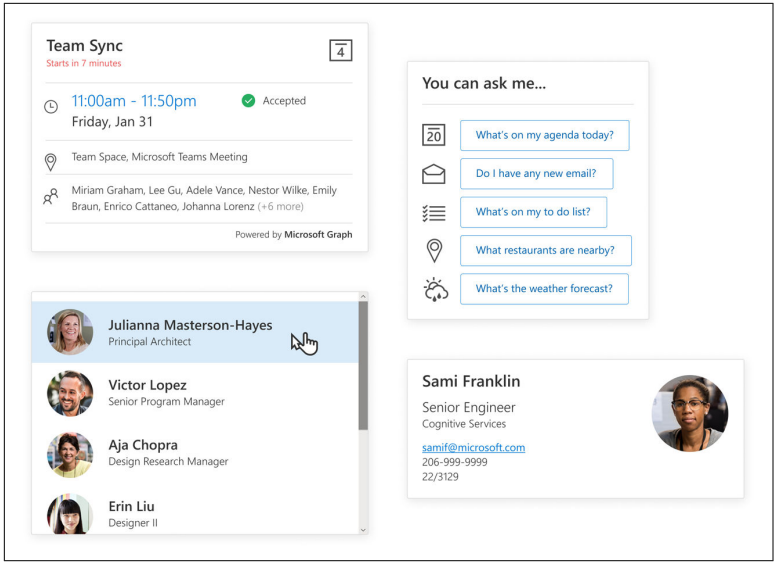


*Figure 6. Adaptive Card examples*

## Capabilities for the Enterprise

A typical Bot Framework–based conversational experience leverages a range of Azure capabilities—for example Azure Bot Service, Language Understanding, and Speech Cognitive Services, along with a broad set of supporting Azure components. This means you benefit from the Azure global infrastructure that includes ISO 27018, HIPAA, PCI DSS, and SOC 1, 2, and 3 certification. In addition, Language Understanding supports many languages. The Translator service provides machine translation capabilities to extend your Virtual Assistant even further.

Now that we know what the Virtual Assistant can do, let us create one. In the next few sections, we will guide you through the process of creating and customizing your Virtual Assistant, adding intelligence to your Assistant with Skills, connecting your Assistant to clients and channels, and enabling analytics for your Assistant. Along the way, we will provide links to online tutorials, which can be done using either C# or TypeScript.

# Developing Your Virtual Assistant

In this section, we will guide you through how to create a highly functional Virtual Assistant in your own Azure subscription.[4]

## The Virtual Assistant Template

The Virtual Assistant template,[5] as we noted earlier, is an open source tool that provides a preconfigured starting point for building a custom Assistant. Using the template, you will generate an Assistant project, which follows the recommended structure for a bot project. You are free to restructure this as necessary but bear in mind that the provided deployment scripts expect some files to be in consistent locations.

Building a conversational experience using the Virtual Assistant template requires the following resources:

- Azure Bot Registration (to configure Bot endpoint and channel configuration)
- Azure Web App (to host your Bot application)
- Azure Storage Account (to store transcripts)
- Azure Application Insights (telemetry)
- Azure Cosmos DB (Conversational State and User State—this can be swapped for Azure Storage for development purposes)
- Language Understanding
- QnA Maker (including Azure Cognitive Search and an Azure Web App)

To enable you to get started quickly and prevision these resources, Microsoft provides an Azure Resource Manager (ARM) template and a set of PowerShell scripts (supported cross-platform), along with Language Understanding models for common base domains, QnA Maker knowledge bases (personality and example FAQs), and the dispatcher tools.

---

4 In addition, you can follow the Bot Framework quick-start documentation to create a simpler experience that can be extended for additional scenarios.

5 The architecture and capabilities of the template are described in the online documentation for the Virtual Assistant template.

## Online Tutorial: Create a Virtual Assistant

Follow the online tutorial (in C# or TypeScript) to create your first Virtual Assistant app that greets a new user and handles basic conversational intents. During this tutorial, you will:

1. Ensure you have an Azure subscription (or get a free Azure account if you don't have one yet).
2. Download and install Bot Framework development prerequisites and the Virtual Assistant template.
3. Create your Virtual Studio project using the Virtual Assistant template.
4. Provision your Assistant using the provided ARM template and a PowerShell script.
5. Run and test your Assistant.

## Online Tutorial: Customize Your Assistant

Now that you have an Assistant, you can optionally customize it to personalize the experience for your brand and users. Dialogs can be added directly to your Assistant or through Skills if you wish to build a more complex solution—this is covered below. Follow this online tutorial (in C# or TypeScript) to learn how to perform the following tasks:

- Edit the greeting by customizing the Adaptive Card.
- Edit responses by customizing Language Generation (*.lg*) files.
- Edit cognitive models by, for example, updating knowledge bases (FAQ and/or chit-chat), adding an additional knowledge base, and updating local LU files for Language Understanding and QnA Maker.

# Connecting Assistants to Clients and Channels

Clients and channels are the ways that users can interact with a conversational AI application. The Azure AI platform does most of the work necessary to connect your bots to a range of channels and devices.

Configured through the Azure portal, you can connect your bots to Facebook Messenger, Slack, Microsoft Teams, Cortana, email, Telegram, Twilio, LINE, and other channels. You can use Web Chat widgets to embed your bots in your websites or mobile applications.[6]

## Online Tutorial: Speech-Enable Your Assistant

Direct Line Speech is a robust end-to-end solution for creating a flexible, extensible voice Assistant that is optimized for speech interaction with bots. Direct Line Speech offers a high level of customization and sophistication for voice assistants.

Follow the online tutorial to perform the following tasks to connect your Assistant to the Direct Line Speech channel and build a simple application integrated with the Speech SDK to demonstrate speech interactions.

1. Create a Speech Service resource.
2. Add the Direct Line Speech channel.
3. Use the Bot Framework Emulator or Speech sample client application and connect to your Assistant.
4. Change the voice.

## Online Tutorial: Surface Your Virtual Assistant to Microsoft Teams

If you're are looking at building an Enterprise Assistant, you can follow the online tutorial to connect your Assistant to Microsoft Teams and create the application manifest required to install your Assistant within Teams.

1. Add the Microsoft Teams Channel.
2. Install the Teams App Studio.
3. Create the application manifest for Teams.

6 You can find details on how to connect to channels in the Bot Framework channels documentation, with the table of contents on the left linking to additional channel-specific instructions. In addition, you have the option to connect your Assistant to Amazon Alexa, Google Home, and others through integration done by the Bot Builder open source community.

4. Test in Teams.
5. Add Commands.

## Optional: Adding Intelligence to Your Assistant with Skills

A Bot Framework Skill provides a conversational component model enabling developers to split their Assistant experience into a set of conversational building blocks, which can be developed independently of each other and brought together into one unified experience. This is a common pattern for larger conversational experiences, whereby there is one "parent bot" that users interact with which then hands them off to various "child" Skills to handle certain tasks.

Think about the broad set of common capabilities and dialogs that developers have traditionally built themselves. Productivity scenarios are a good example, where each organization would need to create its own language models, dialogs, API integration, and responses. The job is then further complicated by the need to support multiple languages, resulting in a large amount of work required for any organization building their own assistant experience.

Bot Framework provides a range of multilanguage open source Conversational Skills—including Calendar, Email, To Do, and Point of Interest—to reduce this effort. The framework also offers a number of experimental Skills, including Phone, News, Weather, Music, and IT Service Management.

These Conversational Skills are themselves bots and incorporate language models, dialogs, and integration code. They are built in the same way as any bot but can be incorporated through easy configuration into an existing conversational experience to extend their capabilities. All aspects of each Skill are completely customizable by developers, and the full source code is provided on GitHub alongside the Virtual Assistant.

Organizations can also create Skills for their private use or to share with other organizations to compose into their own experiences. For example, a conversational app developed by a meal delivery service for their own channels (mobile apps, websites, and conversational

canvases) can also be exposed as a Skill for household IoT devices and cars to integrate as appropriate. This highlights a core capability of Bot Framework and Azure Bot Service: they enable you to write a Skill once and then provide it through many different channels (including Alexa and Google Assistant) with a single code base to reduce duplication across different ecosystems.

### Online Tutorial: Connect to a Sample Skill

As required, you can add any of the sample Skills provided in the Bot Framework Solutions repository to your Assistant. The Skills are available only in C# at this time but can be added to a JavaScript or Python-based Assistant.

Follow the online tutorial to perform the following tasks:

1. Deploy a sample Skill project.
2. Add the sample Skill to your Assistant and republish your Assistant to Azure.
3. Test the sample Skill.

### Online Tutorial: Create a Custom Skill

If you wish to build your own custom Skill to extend your Assistant, you can follow the online tutorial, during which you will perform the following tasks:

1. Create your Skill project using the Skill template in Visual Studio.
2. Provision your Azure resources using the provided ARM template and PowerShell script.
3. Run your Skill.
4. Add your Skill to an Assistant.
5. Invoke your Skill.

## Enabling Analytics for Your Virtual Assistant

Developers can gain key insights into their Assistant's health and behavior with the Bot Framework analytics solutions, which include

sample application queries and a dashboard (leveraging Microsoft Power BI) to understand the full breadth of your Assistant's conversations with users. The underlying data captured can be used by a broad range of data analysis tools, as needed.

## Online Tutorial: View Analytics with Power BI

Follow the online tutorial here to hook up your Assistant with the dashboards provided in the Power BI Virtual Assistant analytics template:

1. Configure your Assistant's telemetry logging with the Application Insights application.
2. Open the Virtual Assistant analytics template (a Power BI template) and connect it with your telemetry logging.

The Power BI template provides a comprehensive dashboard for you to gain valuable insights into how your Assistant is performing and which areas need improvement. It provides you insights including:

*Overall usage*
: Understand core metrics such as unique users, messages and conversations per day, and channels (Figure 4)

*Dialogs*
: Review the popularity of all dialogs as well as outcomes (abandoned, canceled, done, or started)

*Language Understanding*
: Gain insights on Language Understanding intents, which are useful for the purpose of monitoring what your users are talking about

*Conversations*
: View data about conversations per user by day and average duration, with the ability to drill down by conversation to view all related user utterances, bot responses, and dialogs triggered during the same session (Figure 5)

*Transcripts*
: Show interactions, sessions, and transcripts from conversations between the Assistant and users

*QnA Maker insights*
: Review insights on matched user queries with QnA Maker, which is useful for identifying user's knowledge-base gaps

*User feedback insights*
: Review explicit user feedback (positive, negative) and corresponding user and bot utterances if your bot has enabled feedback middleware in its telemetry

Now that you have learned how to view a sample of Virtual Assistant analytics, you can also:

- Add telemetry to your bot: learn what specific code components are required for out-of-the-box telemetry.
- Analyze your bot's telemetry data.
- Work with events generated by the Bot Framework Service telemetry.

## Roadmap and More Resources

Microsoft continues to invest in its Azure AI platform to make it easier for organizations and developers to build robust conversational solutions, and to deploy them wherever their customers are.

For example, Microsoft continues to improve the process for building conversational experiences through regular releases of the Bot Framework SDKs and tools. As best practices evolve, they are packaged into the Virtual Assistant template and sample Skills.

Microsoft is committed to making bot building easier with the Bot Framework Composer, which provides a low-code visual interface for creating, editing, testing, and refining bots. Microsoft is also connecting bots to more users by adding more channels and support for human hand-off and telephone-based communications.

At the same time, Microsoft is enabling more natural, dynamic, and sophisticated conversations with Adaptive Dialogs. These dialogs enable a more natural interaction, with the user being able to seamlessly move around all stages of a dialog, changing their mind on a previous answer or providing additional information addressing later questions automatically.

Additionally, Microsoft is improving and enhancing Language Understanding capabilities and providing support for document understanding.

For additional resources, please check out:

- GitHub's Bot Framework SDK documentation
- Microsoft's Azure Bot Service documentation
- Microsoft's Bot Framework documentation
- GitHub's Bot Framework news

## Building Responsible AI

The Capgemini Research Institute, in their July 2019 report, identified that nearly nine in ten organizations have encountered unintended consequences resulting from the use of AI. The authors of this report identified their top concerns, which included:

- Overreliance on machine-led decisions without disclosure
- Collecting and processing personal data in AI algorithms without consent or for purposes other than which it was collected
- Biased and unclear recommendations resulting in discriminatory access and pricing of products or services
- Citizens objecting to mass surveillance, collection, and the use of personal data including biometrics
- Customers demanding reasoning and clarity behind a decision taken by an AI algorithm

As we've described in this book, breakthrough advances in AI technologies over the last five years are beginning to transform products and services, which affects each of our lives. While many of these changes are good, they also raise concerns about unintended consequences resulting from bias, erosion of privacy, misinformation, and automation.

Designing AI to be trustworthy requires creating solutions that reflect ethical principles that are deeply rooted in important and timeless values. At Microsoft, we follow principles of fairness, relia-

bility and safety, privacy and security, inclusiveness, transparency, and accountability to build responsible AI systems.

> As is true for any technology, trust will ultimately depend on whether AI-based systems can be operated reliably, safely and consistently—not only under normal circumstances but also in unexpected conditions or when they are under attack.
>
> —Microsoft President Brad Smith, *The Future Computed*

Building responsible AI applications is critical, but it ultimately rests on AI developers and the organizations deploying AI technologies to ensure responsible behavior. To help developers and organizations to build responsible solutions, Microsoft has proposed 18 design guidelines for human-AI interaction. For conversational AI systems, Microsoft also released a set of 10 guidelines covering impact, transparency, inclusiveness, reliability, privacy, security, and the detection of bias or unfair exclusion.

To take a responsible approach with AI, it is essential to consider the needs of the people who will use the solution during the initial design. For example, a conversational agent designed to assist citizens gaining access to public healthcare services must consider and cater to the full range of abilities of those individuals. Designing a conversational interface that supports audio and text input helps those with visual impairments, as well as individuals who may be operating a vehicle.

As users and developers of AI solutions, you must define and follow criteria that reflect your AI principles. For example, with fairness, it is important to consider the potential harms of allocation, representation, or association that could result from unfair bias in data and models. An AI application being used to screen job candidates could associate gender with certain hobbies, such as sports activities, based on biases in training data and word-embedding models, which may result in unfair selection. It is important to assess and monitor an AI solution's fairness and mitigate observed unfairness using tools like the Fair Learn toolkit.

It is also important to understand the context in which the solution operates. For example, with bots that have humanlike personas it is especially important that they interact respectfully and safely with users and have built-in safeguards to handle misuse and abuse. A bot that helps consumers book entertainment is likely to support

tone and terms that are not appropriate for a healthcare-focused bot.

In order to build trust, it is critical that people understand what a solution can do and what information that solution collects. Some critical questions that developers should be asking themselves include:

- Should users be aware that a bot lies behind the dialog in which they are engaged?
- How does the system handle failures, for example when there is an error in speech-to-text transcription?
- What data does it collect, and how do users control that data collection?
- Is the system susceptible to new forms of attack?

Finally, for people to be accountable and effectively use and make decisions based on the results of an AI solution, they must understand how the solution works and be able to explain the results. This is critical in situations where the impact is high, for example in a healthcare solution that may impact the treatments that a patient receives. Historically, developers had to trade off accuracy and transparency, but newer techniques, including model selection and model interpretability tools, combined with greater rigor in data and system documentation can eliminate that trade-off. It is important to design AI solutions for both accuracy and transparency.

The objective of considering responsible AI principles is to build trust in the solution and ultimately the people, service, and company the solution represents.

## About the Authors

**Elaine Chang** is a leader of product development and customer success for conversational AI at Microsoft, where she focuses on solutions including Virtual Assistant Solution Accelerator and Skills. She has been one of the key product leaders for Microsoft Bot Framework and has led Azure Bot Service to general availability and enterprise compliance.

Elaine is a featured speaker at Microsoft Build Conference, Microsoft Ignite Conference, Microsoft MVP Summit, Microsoft AI Innovate, and more. Elaine is also a strategic innovator, a certified professional coach, and a business leader who advocates driving innovation through diversity and inclusion.

**Darren Jefford** has over 20 years of engineering and architect experience across a variety of industries. While at Microsoft, he has worked in high-impact, customer-facing roles to architect and deliver highly complex solutions using a broad range of technologies. In recent years, he has led some of the first conversational AI projects for a variety of organizations.

Darren is currently a principal architect in the Bot Framework team at Microsoft, where he leads the Virtual Assistant team to enable complex conversational experiences with key customers and the broader developer ecosystem.

Darren is a regular speaker at Microsoft events and is also the author of two books focusing on Visual Studio and BizTalk Server.