

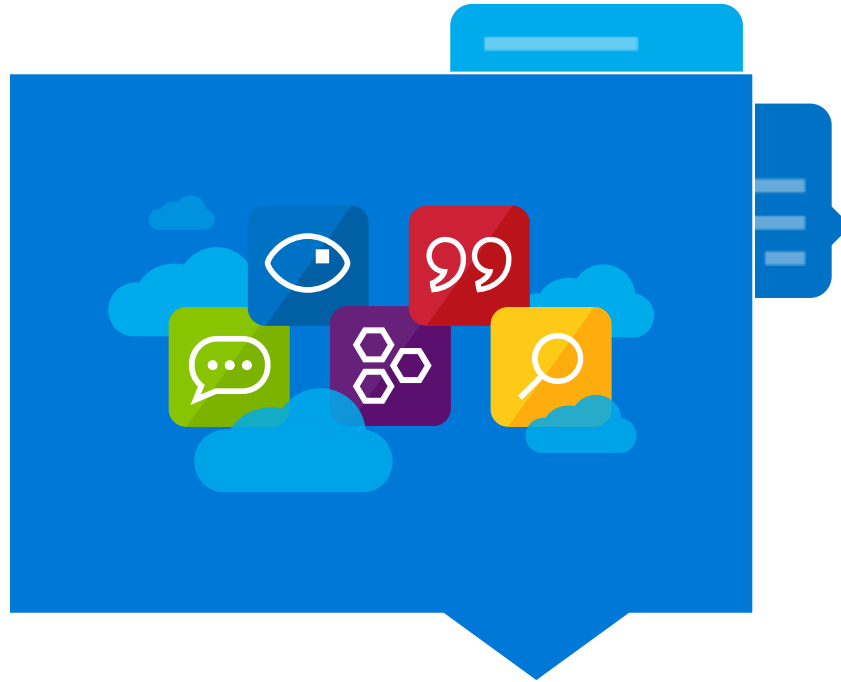
Azure Bot Service

Ashish Talati

Ashish.Talati@microsoft.com

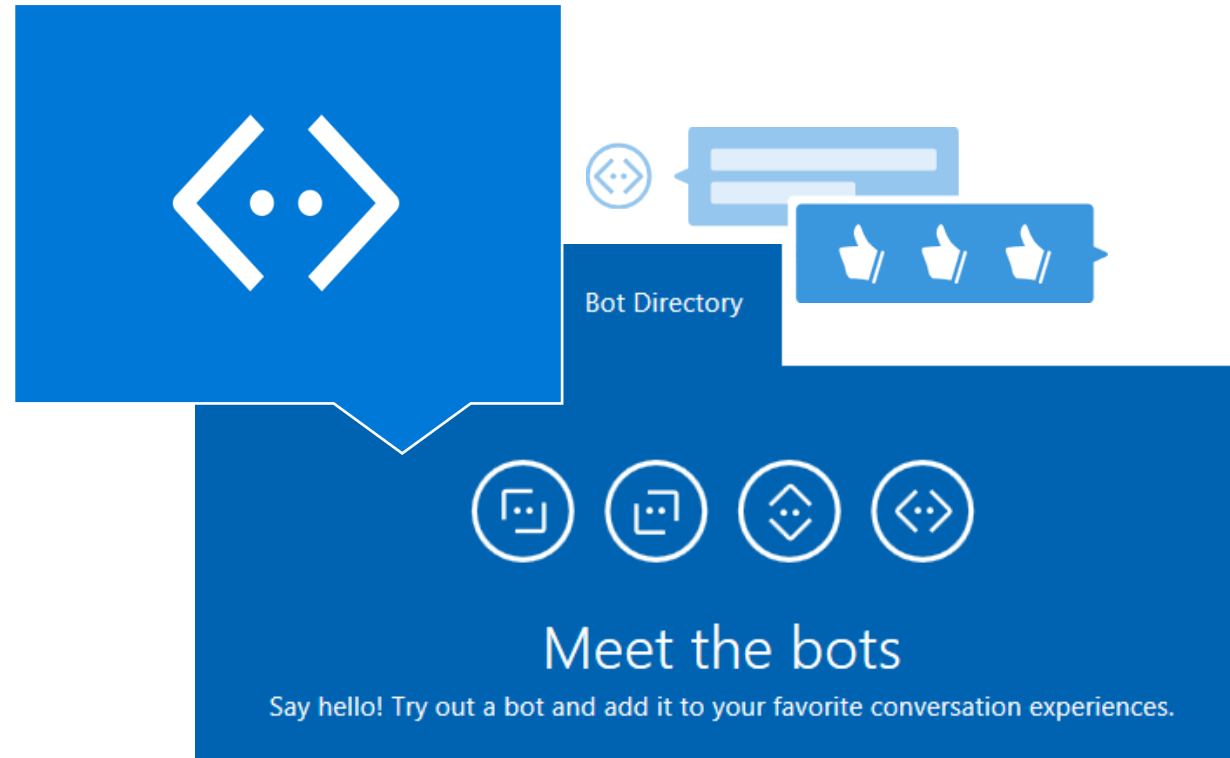
Azure Bot Service

The Azure Bot Service provides tools to connect, deploy, and manage intelligent bots to interact naturally with your users on websites, apps, Cortana, Microsoft Teams, Skype, Slack, Facebook Messenger, and more.



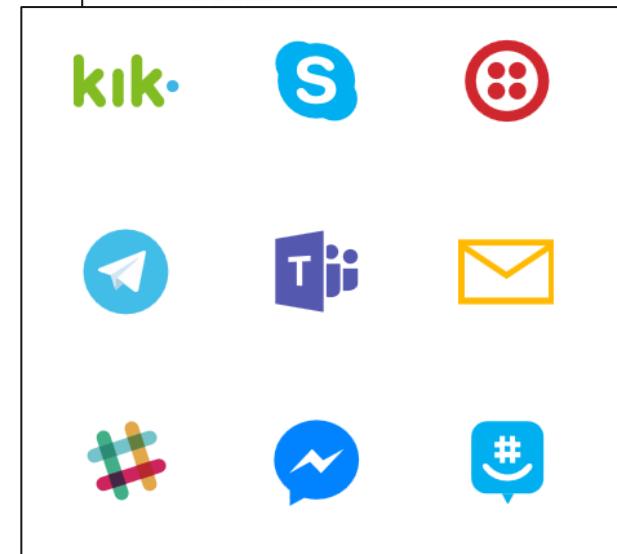
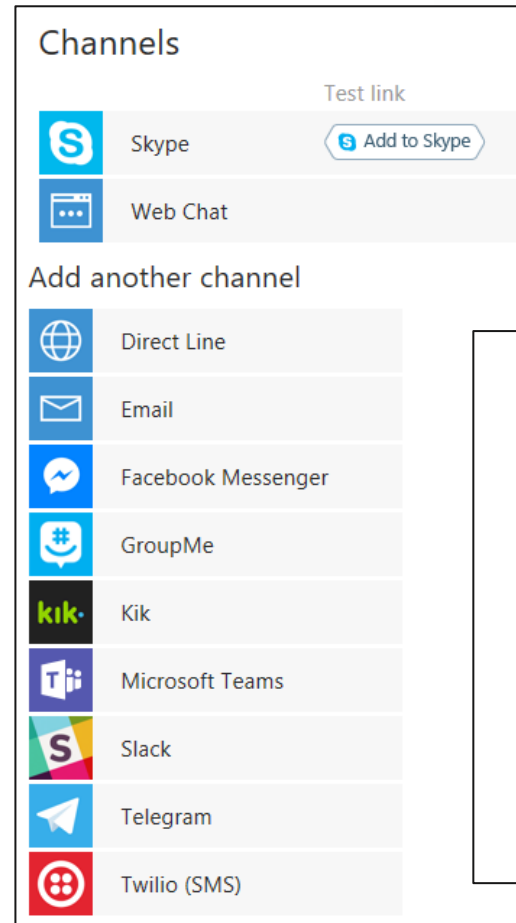
Microsoft Bot Framework

The Microsoft Bot Framework is a comprehensive set of tools and services developers can leverage to build and deploy high-quality, conversational bots in any language.



Supported Channels

Channels are currently available for Skype, web, email, Facebook, GroupMe, Microsoft Teams, Kik, Slack, Telegram, Twilio, and direct line app integration.



The Connector

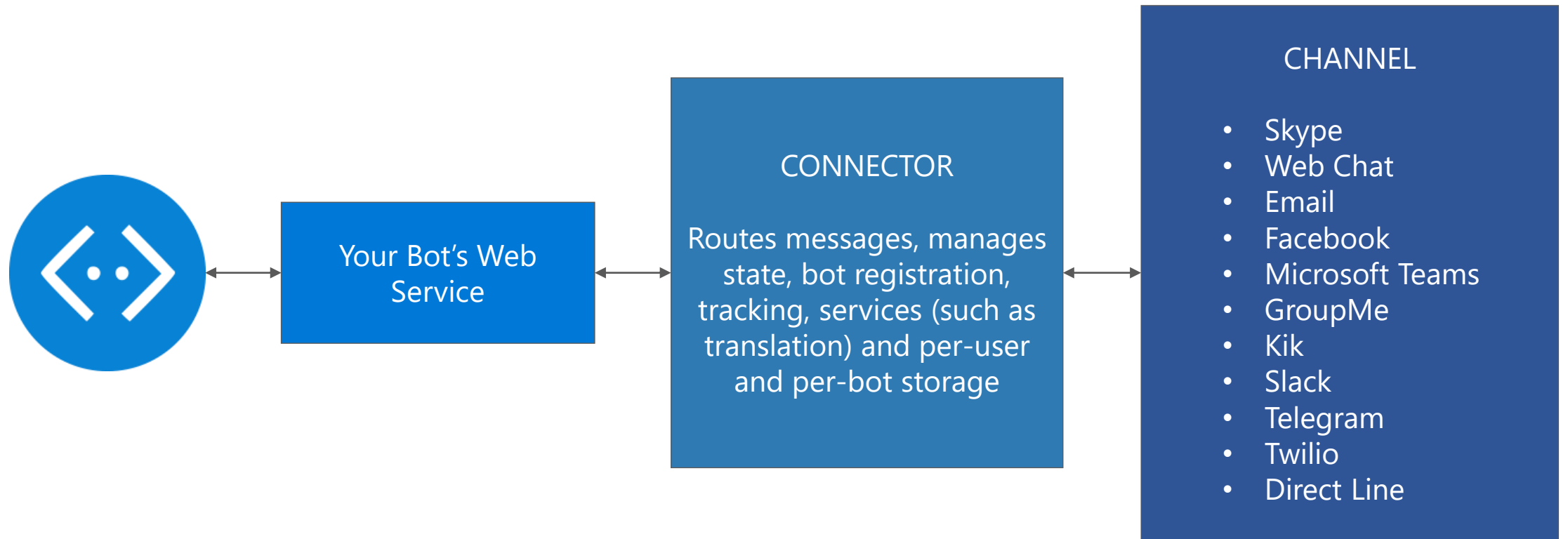
The Connector (or Connector Service) handles all communication, conversations, state, and authorization activities between a bot and users.

```
var connector = new
ConnectorClient(incomingMessage.ServiceUrl);

var replyMessage =
incomingMessage.CreateReply("How can I
help?");

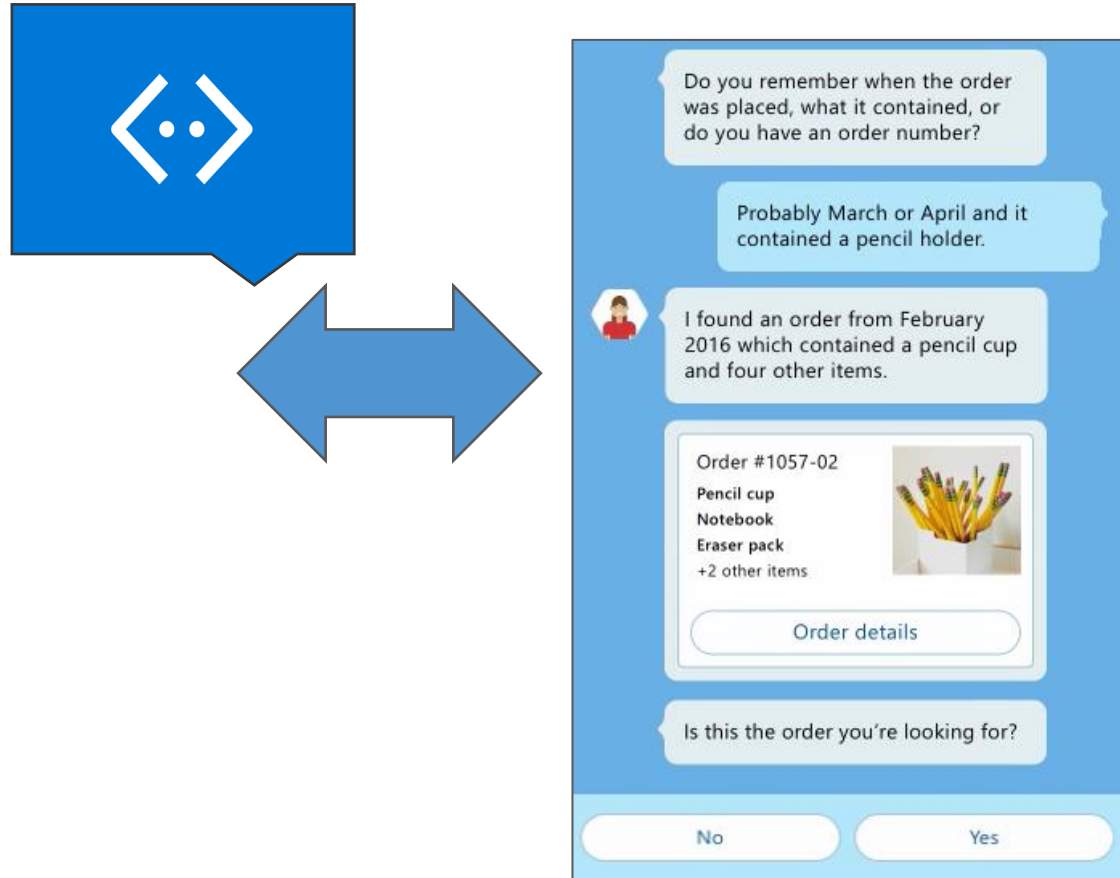
await
connector.Conversations.ReplyToActivityAsync(
replyMessage);
```

Connector Service Flow



Activities

An Activity is a specific event that occurs between a bot and users, such as an actual message, or conversation notification.



Types of Activities

Activity Type	Description
Message	Sent when general content is passed to or from a user and a bot
Conversation Update	Sent when the conversation's properties change, for example the topic name, or when user joins or leaves the group
Contact Relation Update	Sent when bot added or removed to contact list
Delete User Data	Send when user is removed from a conversation
Typing	Sent when a user is typing
Ping	Send when a keep-alive is needed

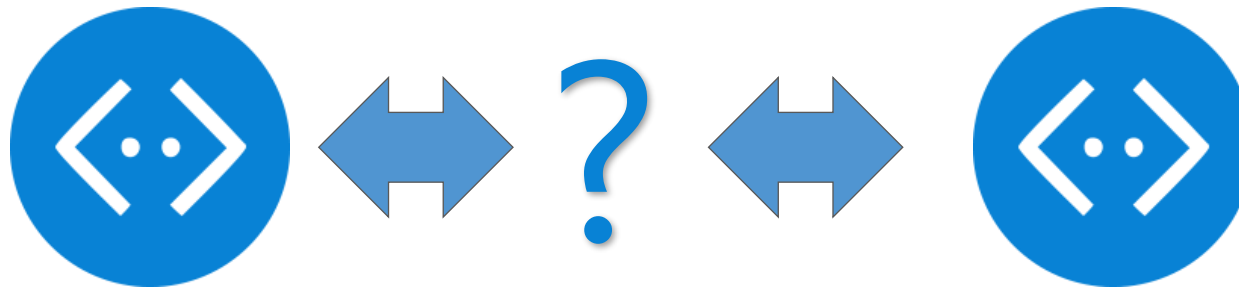
Messages

A Message is an overt communication between a bot and users, such as a user asking a question, or a bot responding with a reply.

Message Property	Description
Text	A text payload in markdown syntax which will be rendered as appropriate on each channel.
TextFormat	Plain, markdown, XML (Skype only)
Locale	Two-digit culture/region locale

Intents and Prompts

Mechanisms leveraged by the Microsoft Bot Framework to determine what a user really wants to do are typically referred to as intents and prompts.



Managing State

Not all bots require state management outside of what the Connector provides inherently, however if you need to manage state, there are various methods.

- Custom storage methods, such as a remote database
- Class-specific logic at runtime
- Implicitly during a Dialog or Form process
- The Bot Framework State Client

Using Dialogs

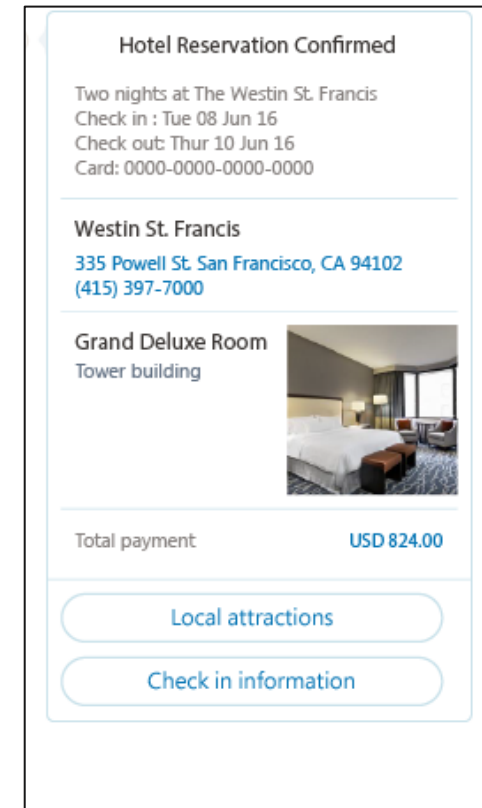
Dialogs are a way of wrapping an entire “experience” into an easily managed interaction based on a “chained” and “conversational” paradigm.

- Send information to a User
- Prompt a User for more information or confirmation
- Provide conditional logic
- Provide “as you need it” content
- Dialogs can contain or forward to other Dialogs

Attachments, Cards, and Actions

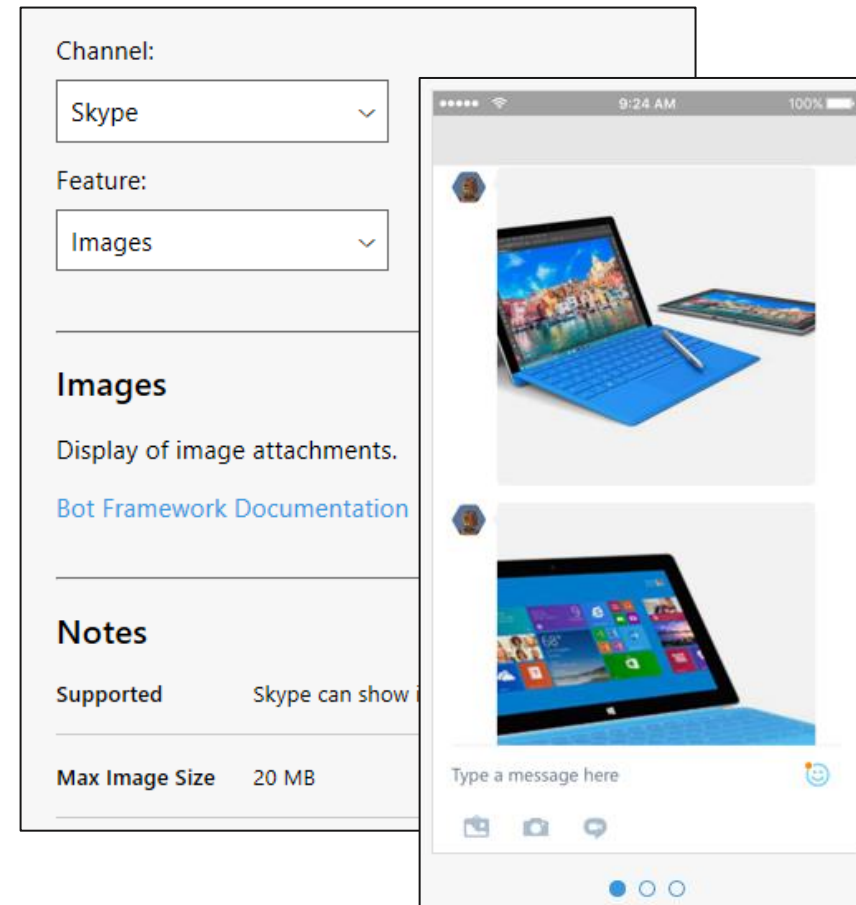
Bot conversations can contain enhanced content in the form of Attachments, Cards, and Actions

- Media attachments:
Image, audio, video, file
- Rich card attachments:
Hero, thumbnail, receipt, sign-in
- Button and card actions:
Open URL, call, play audio/video, show image, download file, sign-in



Channel Inspector

The Channel Inspector is a web-based, interactive tool you can use to evaluate supported features for specific chat clients, making it easy to understand the style your users will see and limitations in presentation, based on target chat platforms.



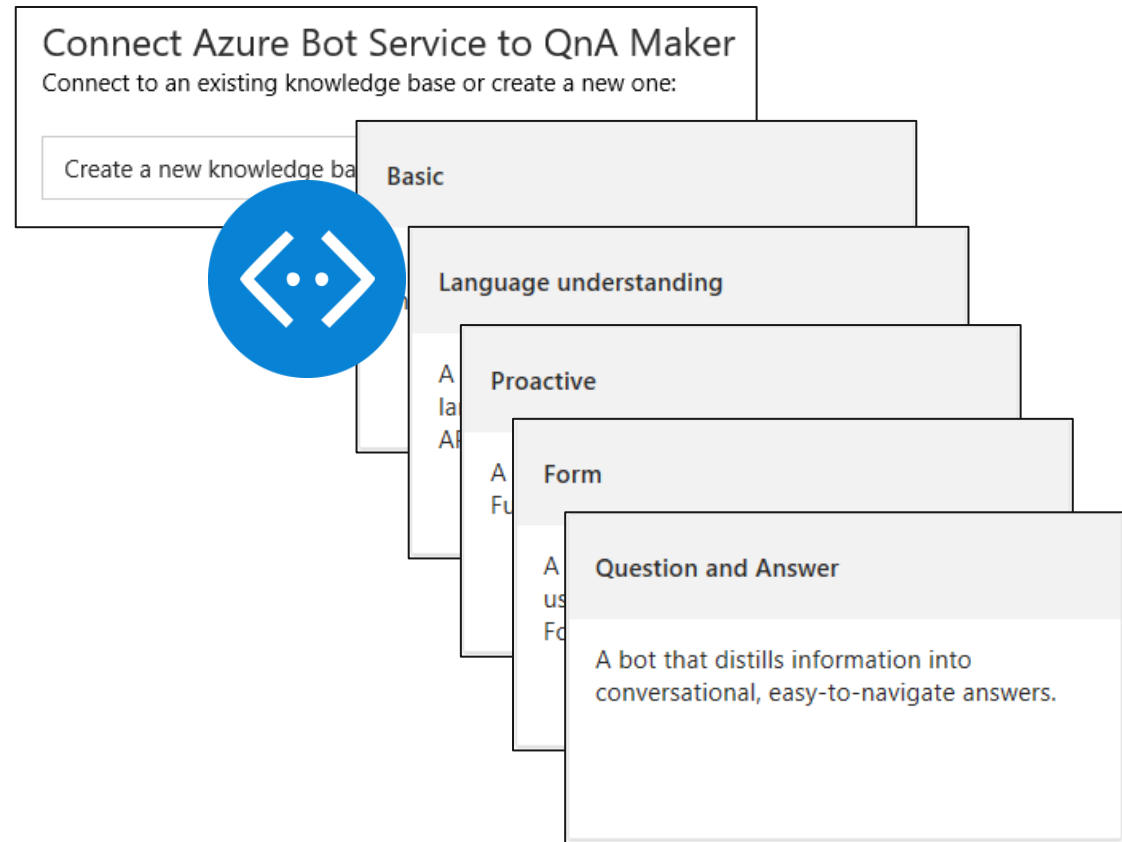
The Bot Framework Emulator

Emulate bot conversations easily on any device and any supported channel scenario.

- Supports Mac, Linux and Windows
- View all Bot Framework card types
- Save multiple profiles for working offline
- Send requests and receive responses to/from your bot endpoint on localhost
- Inspect responses, and emulate specific users and conversations

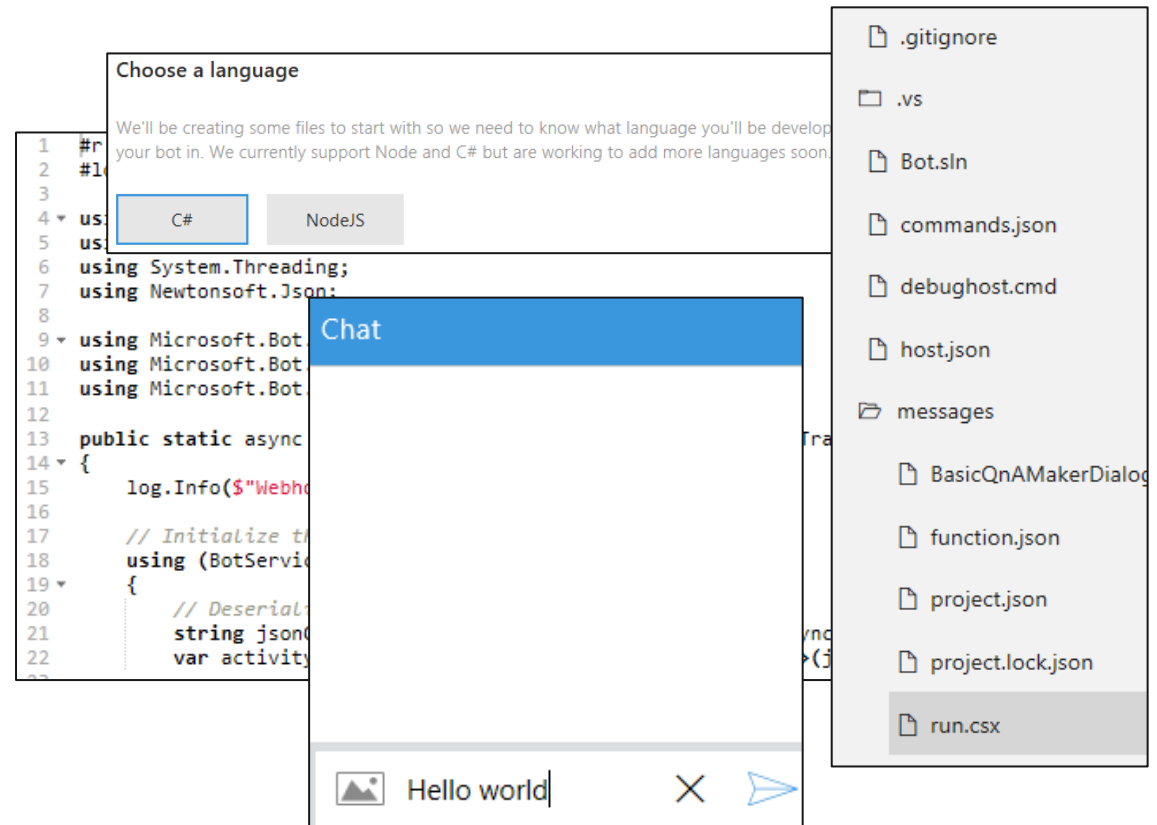
Azure Bot Services

The Azure Bot Service, powered by the Microsoft Bot Framework, provides an integrated environment for managing bot logic and integration, including the ability to edit services without the need for a local editor or source control.



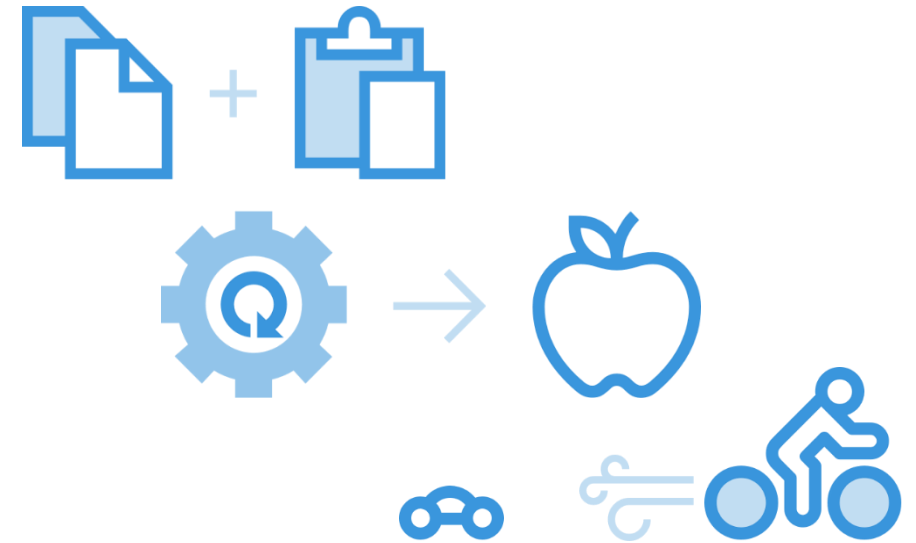
Azure Bot Services

Bots can be created and managed within a browser directly, in both C# and Node.js, using the Azure editor, effectively removing any need for a local editor. You can quickly test your bot directly in the Azure service environment.



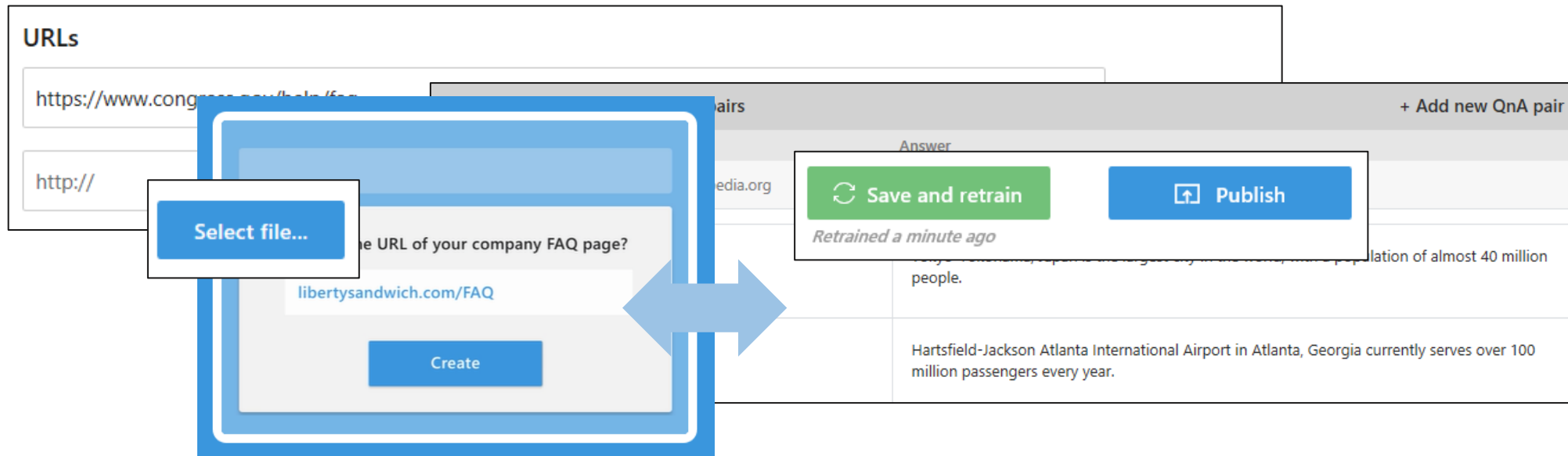
Microsoft QnA Maker

Microsoft QnA Maker is a free REST API and web-based service that enables marketing, support, operations, and developers to train an artificial intelligence engine to respond to user's questions in a natural, conversational, question and answer-based format.



Microsoft QnA Maker

There are several ways to update a QnA knowledge base, including simple FAQ reference via a URL or direct management of QnA question and answer pairs.



Hands-On Lab

Azure Bot Service HOL.html



© 2016 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.