

IoT Virtual Bootcamp

December
12 – 14, 2017



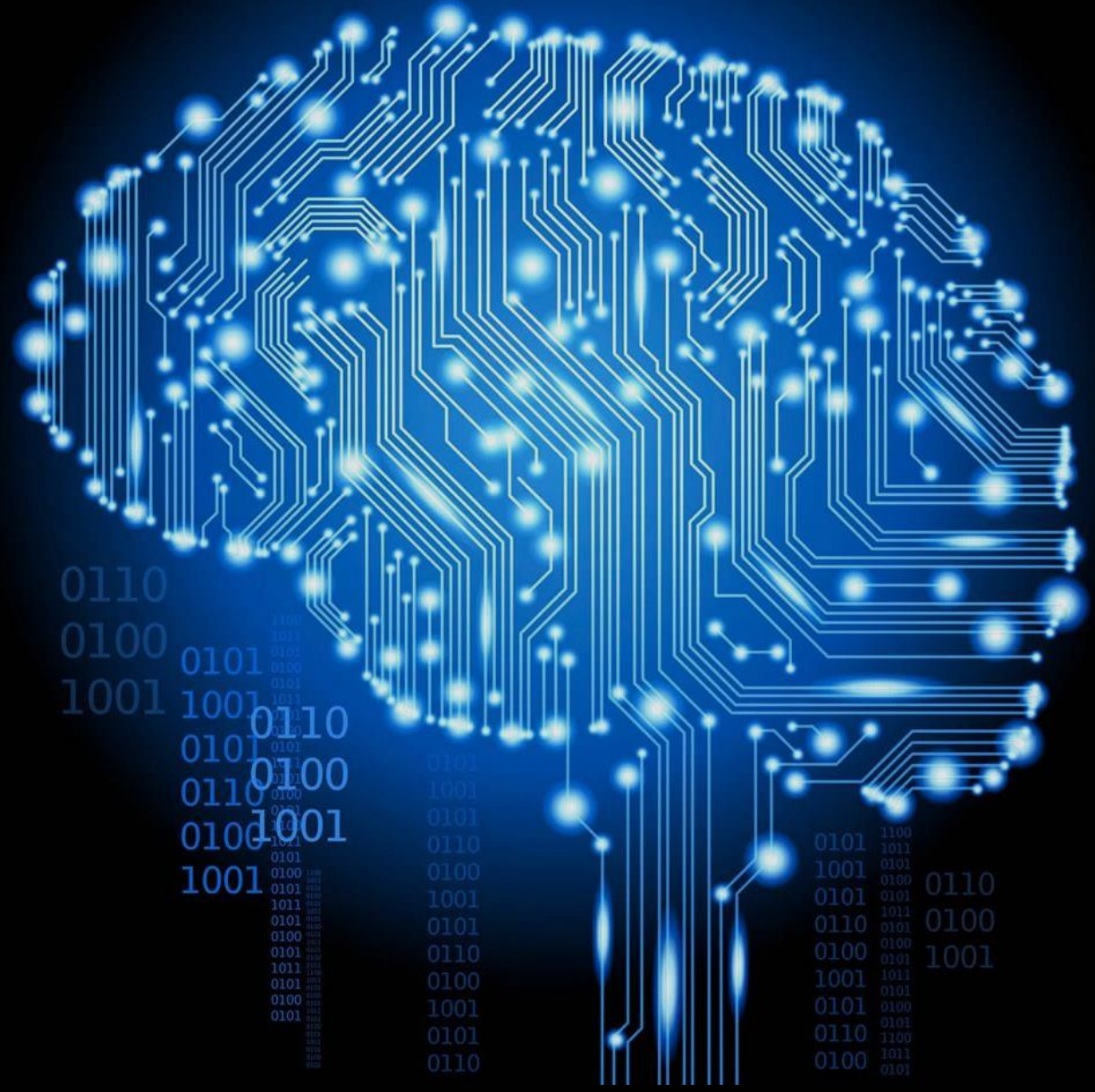
Microsoft Bot Framework & Azure Bot Service

Maarten Struys

"The world and technology are once again transforming."

"Computers may win at games, but imagine what's possible when human and machine work together to solve society's greatest challenges like beating disease, ignorance, and poverty."

SATYA NADELLA, CEO MICROSOFT



1990s: **Internet**

- Search
- User “visits” websites

The future: **Conversations**

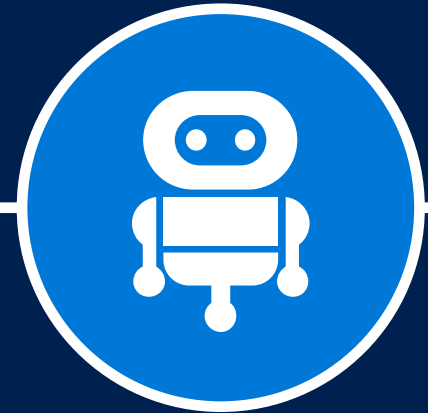
- Natural language between people and technology
- Conversational canvas
- Bots and agents

1980s: **PC**

- Desktop

2000s: **Mobile**

- Social
- User download apps from App Stores



Conversations as a platform



Developers

The new canvas

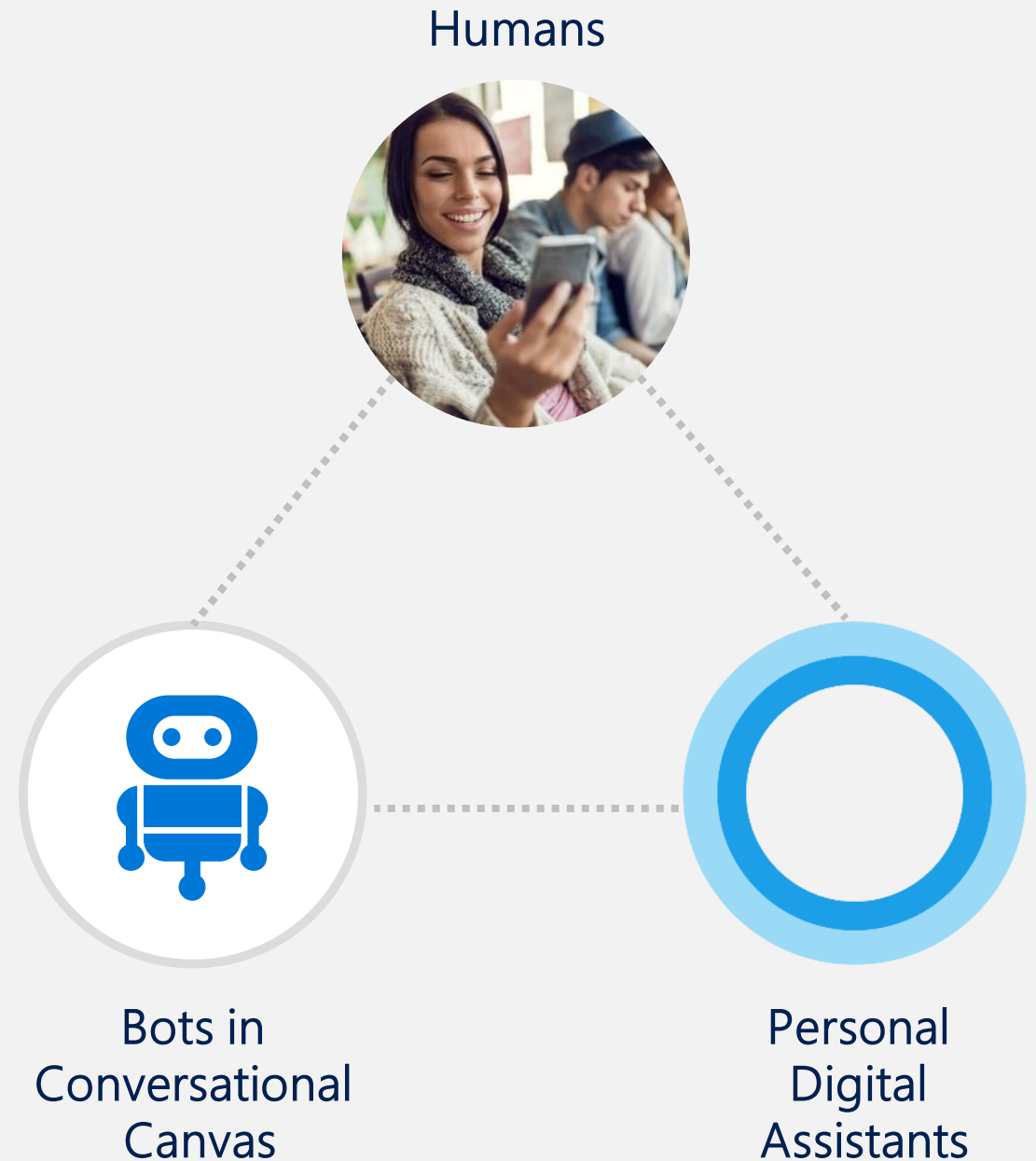
Businesses

The new opportunity

Individuals

More personal way to
discover, access, and interact

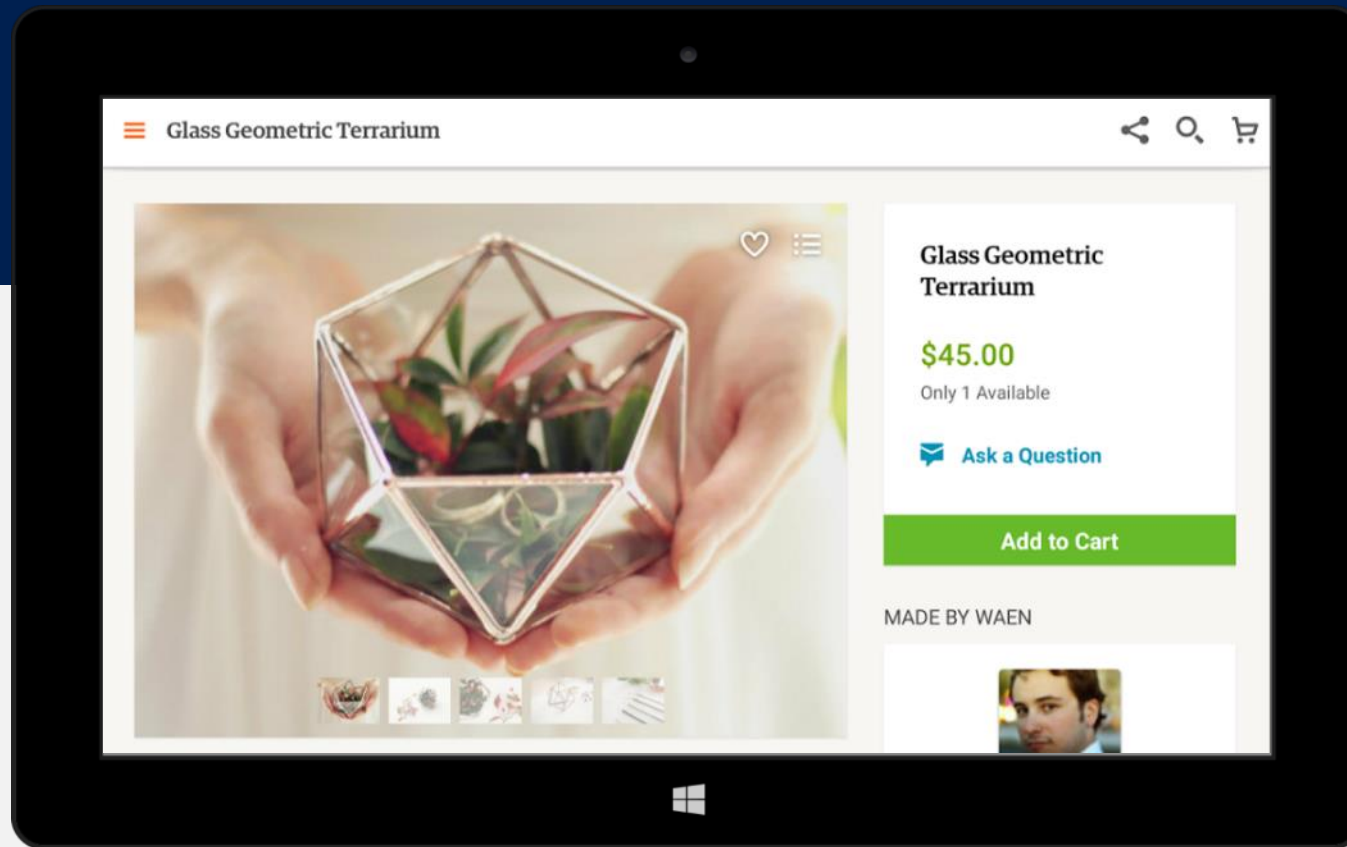
Conversations as a platform



How your brands communicate with customers today

Website

App



Transforming engagement



Productivity task completion

Enhance productivity through task automation and through automated workflows



Reach customers anywhere, on any platform or device



Engage with customers and users in a natural way, conversationally and in context



Expose your products and services via messaging platforms, where users are already highly engaged



Improve customer experience and reduce the need for human assistance

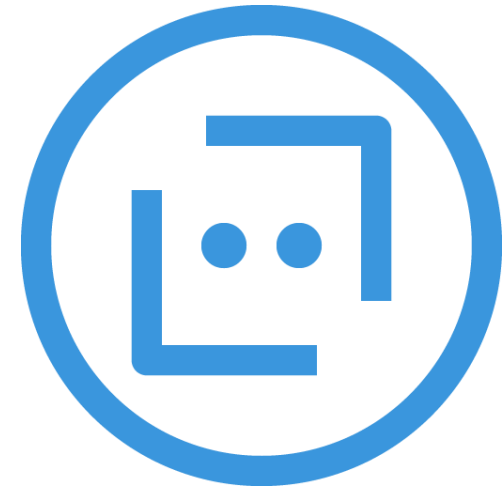
Intelligence platform



Cognitive Services



Microsoft Bot
Framework



Azure Bot Service

Your Bot Framework Bot



Bot Builder

Tools and services to build great bots that converse wherever your users are.

- Open source SDK on Github for Node.js, .NET and REST
- From simple built-in prompts and command dialogs to simple to use yet sophisticated 'FormFlow' dialogs
- Support for rich attachments (image, card, video, doc, etc.); support for calling (Skype)
- Online/offline chat Emulator
- Add bot smarts with Cognitive Services for language understanding and more



Developer Portal

Connect your bots to text/sms, Skype, Slack, Facebook Messenger, Office 365 mail and other channels.

- Register, connect, publish and manage your bot through your bot's dashboard
- Automatic card normalization across channels
- Skype channel auto-configured
- Embeddable Web chat control
- Host your bot in your app via the Direct Line API
- Fast, scalable message routing
- Bot Directory for discovery and trial



Cognitive Services

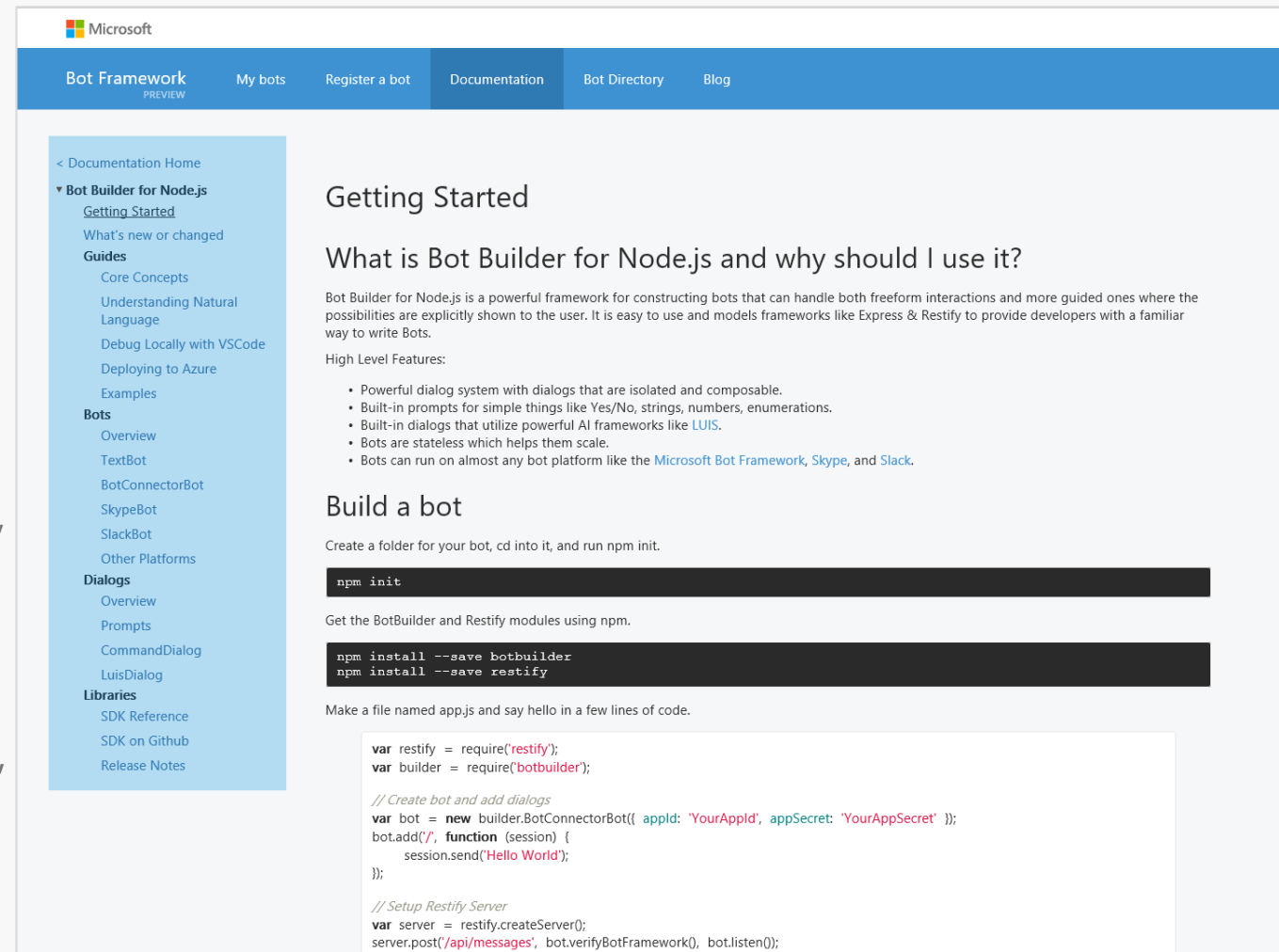
23 purpose-built ML and AI services to add predictable, configurable intelligence into any software

- Parse natural language (LUIS)
- Real-time transcription and translation services for voice
- Vision and face detection algorithms
- Sentence diagramming
- Web language (hashtag) parsing
- Many more

Bot Builder SDK

Node.js, .NET and REST

- Dialogs to model conversation
 - Dialogs are reusable
 - Types of Dialogs include:
 - Built-in prompts
 - Yes/No, String, Number, Choices
 - FormFlow and form slot filling (branching, disambiguation, multi-turn)
 - Conversations are scalable to multiple machines
- Rich interactions
 - Support for rich attachments (image, card, video, doc, etc.); support for calling (Skype)
 - Service extensions for language understanding (LUIS) and translation
- Online/offline Chat Emulator
- Samples



The screenshot shows the Microsoft Bot Framework documentation page for Node.js. The page has a blue header with navigation links: Bot Framework (PREVIEW), My bots, Register a bot, Documentation, Bot Directory, and Blog. The left sidebar contains a table of contents for the Node.js documentation, including sections like Getting Started, What's new or changed, Guides, Bots, Dialogs, and Libraries. The main content area is titled "Getting Started" and includes a subheading "What is Bot Builder for Node.js and why should I use it?". Below this, there is a paragraph explaining the framework's capabilities and a list of high-level features. The next section is "Build a bot", which provides instructions on how to create a folder, run npm init, and install the botbuilder and restify modules. It also includes a code block showing the setup for a simple bot that responds with "Hello World".

Microsoft

Bot Framework PREVIEW My bots Register a bot Documentation Bot Directory Blog

< Documentation Home

▼ Bot Builder for Node.js

- Getting Started
- What's new or changed
- Guides
 - Core Concepts
 - Understanding Natural Language
 - Debug Locally with VSCode
 - Deploying to Azure
 - Examples
- Bots
 - Overview
 - TextBot
 - BotConnectorBot
 - SkypeBot
 - SlackBot
 - Other Platforms
- Dialogs
 - Overview
 - Prompts
 - CommandDialog
 - LuisDialog
- Libraries
 - SDK Reference
 - SDK on Github
 - Release Notes

Getting Started

What is Bot Builder for Node.js and why should I use it?

Bot Builder for Node.js is a powerful framework for constructing bots that can handle both freeform interactions and more guided ones where the possibilities are explicitly shown to the user. It is easy to use and models frameworks like Express & Restify to provide developers with a familiar way to write Bots.

High Level Features:

- Powerful dialog system with dialogs that are isolated and composable.
- Built-in prompts for simple things like Yes/No, strings, numbers, enumerations.
- Built-in dialogs that utilize powerful AI frameworks like [LUIS](#).
- Bots are stateless which helps them scale.
- Bots can run on almost any bot platform like the [Microsoft Bot Framework](#), [Skype](#), and [Slack](#).

Build a bot

Create a folder for your bot, cd into it, and run npm init.

```
npm init
```

Get the BotBuilder and Restify modules using npm.

```
npm install --save botbuilder
npm install --save restify
```

Make a file named app.js and say hello in a few lines of code.

```
var restify = require('restify');
var builder = require('botbuilder');

// Create bot and add dialogs
var bot = new builder.BotConnectorBot({ appId: 'YourAppId', appSecret: 'YourAppSecret' });
bot.add('/', function (session) {
    session.send('Hello World');
});

// Setup Restify Server
var server = restify.createServer();
server.post('/api/messages', bot.verifyBotFramework(), bot.listen());
```

Developer Portal

- Register your bot
- Connect to channels
- Test
- Publish
- Manage
- Measure

The screenshot shows the Microsoft Bot Framework Developer Portal interface for a bot named "Liberty Sandwich Bot". The top navigation bar includes links for "Bot Framework", "My bots", "Register a bot", "Documentation", "Bot Directory", and "Blog", along with a search bar and a user profile icon. A success message at the top states: "You've successfully connected your bot to Skype and Web chat. Click **Edit** on a channel to fine tune its specific settings."

The main content area is divided into several sections:

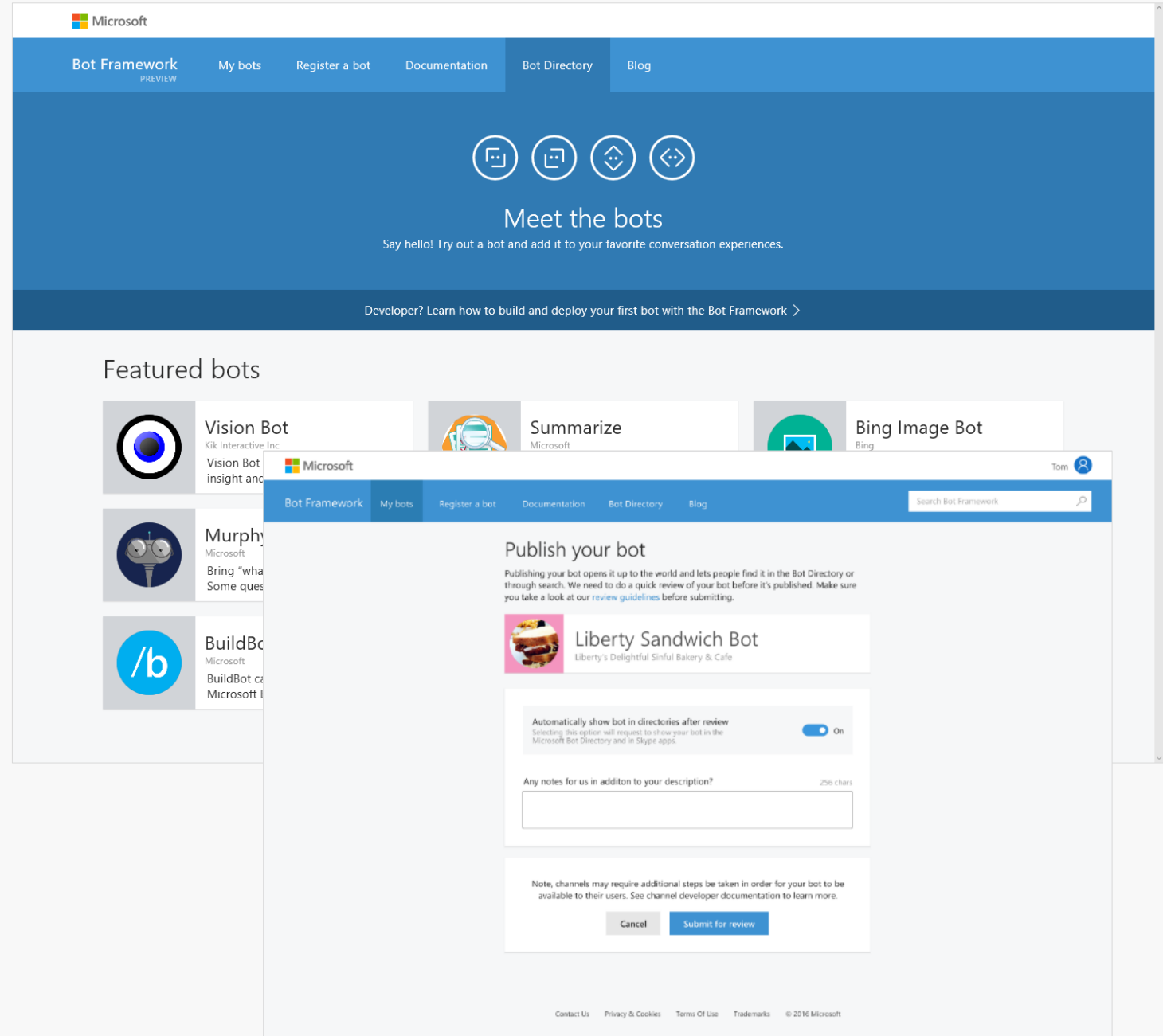
- Details:** Displays bot information including Bot ID (LibertySandwichBot), App ID (ae9284bf-06a6-4f4b-a8b6-b779b390a415), Password (masked), and Messaging endpoint (https://fc-adbot-scratch.azurewebsites.net).
- Channels configuration:** A table showing the status of connected channels. Skype is "Running (Preview)" and "On". Web chat is "Disabled" and "Off". Other channels like Direct line, Email Office 365, Facebook Messenger, GroupMe, Slack, and SMS are listed with "Add" buttons.
- Test connection to Bot framework:** A section with a text input field and a "Type your message" button.
- Analytics:** A section prompting the user to provide their Azure App Insights key to receive analytics.
- Cognitive Services:** A section explaining that Cognitive Services enable bots to see, hear, interpret, and interact in more intelligent ways.
- Chat:** A section for testing the bot via the web channel, showing a chat history with the bot's introduction and a contact details request.

A notification bubble on the right states: "Skype and Web chat channels have been auto configured. Click 'Edit' to fine tune channel specific capabilities and settings."

Bot Directory

Public Directory of Bot Framework Bots

- Users can discover, try, and add bots to the conversation experiences on which the bot is configured (no app required)
- Bots are public at developer discretion; bots must be submitted for review in order to appear in the directory
- Searchable



Benefits

For developers

- Easiest way to reach the broadest set of users where they already are conversing
- Excellent integration with companion services to add intelligence to your bot, such as LUIS for language understanding as well as many other Cognitive Services

For end users

- Users can choose from a variety of conversation channels
- Users have trust and control of their data

For businesses

- Broad access to your customers, new ways to delight them, serve them, do business with them – in your own web site or app or in any number of popular messaging experiences
- Reduced cost of development

Demo

Building a simple bot using the Microsoft Bot Framework



Azure Bot Service Preview



Accelerate development
cycles

Enrich your bots

Boost operational
efficiencies



Azure Bot Service Preview

Accelerate development cycles

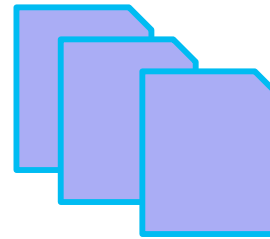
nodeJS

C#

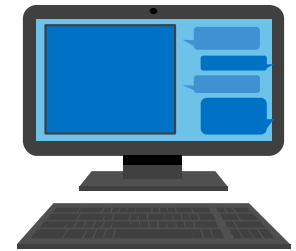
Develop
your way



Built in
code editor



Quick start
templates

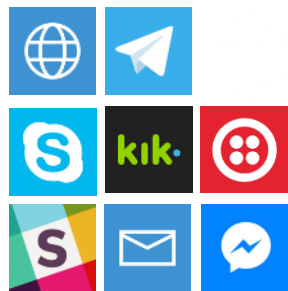


Integrated
chat window

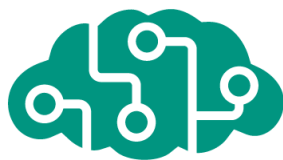


Azure Bot Service Preview

Enrich your bots



Channel
support



Cognitive
Services



Direct
Line support



Embedded
web chat

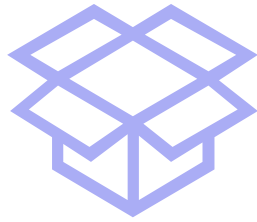


Azure Bot Service Preview

Boost operational efficiency



Powered by
Azure Functions



Continuous
deployment



Scale
on demand



Reduced
dev ops



azuretestbot2

Develop

Channels

Settings

Publish

Chat

Configure continuous integration - manage your code in your repo of choice and edit locally.

save

azuretestbot2

EmptyBot

EchoDialog.csx

function.json

PostDialogExtensions

project.json

project.lock.json

run.csx

Utils.csx

host.json

```
1 #load "Utils.csx"
2
3 using System;
4 using System.Threading.Tasks;
5 using Microsoft.Bot.Builder.Dialogs;
6 using Microsoft.Bot.Connector;
7
8 [Serializable]
9 public class EchoDialog : IDialog<object>
10 {
11     protected int count = 1;
12
13     public Task StartAsync(IDialogContext context)
14     {
15         try
16         {
17
```

Logging clear

2016-11-09T14:17:41 Welcome, you are now connected to log-streaming service.
2016-11-09T14:17:53.916 Function started (Id=e983d667-03a1-4482-8014-ee7706456c09)
2016-11-09T14:17:53.916 Webhook was triggered!
2016-11-09T14:17:55.327 Function completed (Success, Id=e983d667-03a1-4482-8014-ee7706456c09)

Welcome!

azuretestbot2

How are you?

you

1: You really said How are you?

azuretestbot2

Type your message...



Demo

Building a simple Q&A bot using the Azure Bot Service

Next steps

Sign up for a free Azure Trial

<https://azure.microsoft.com/en-us/free/>

Create your first bot

<https://azure.microsoft.com/services/bot-service/>

Microsoft Bot Framework

<https://dev.botframework.com/>

