

# 이제는 챗봇 시대! 인텔전트 챗봇 개발의 모든 것

## The age of Chatbot

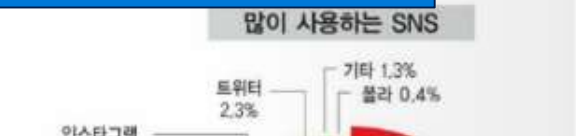
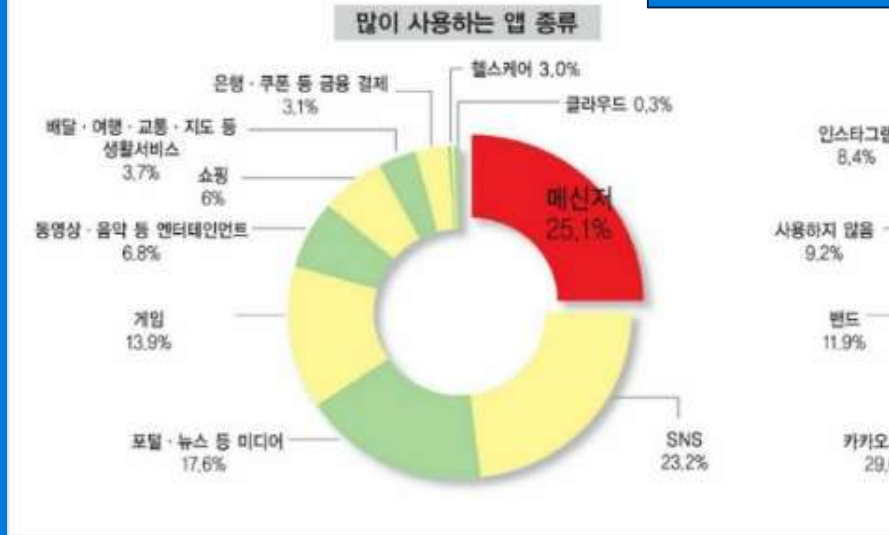
김영욱 부장

Technical Evangelist / DX

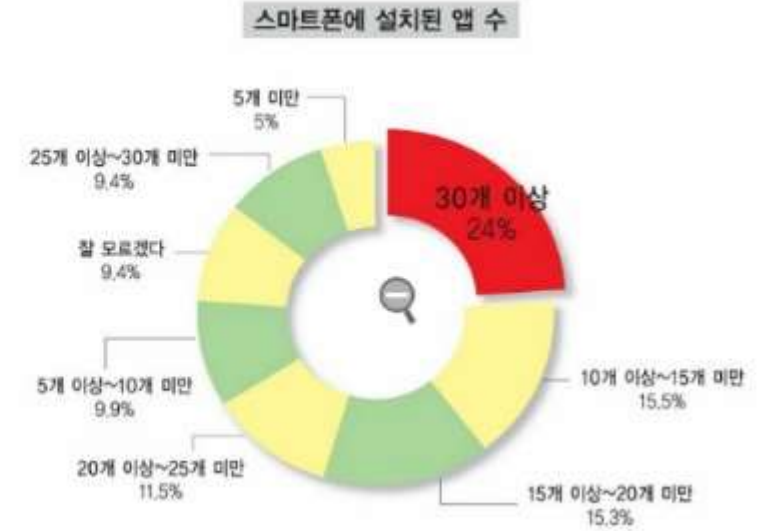
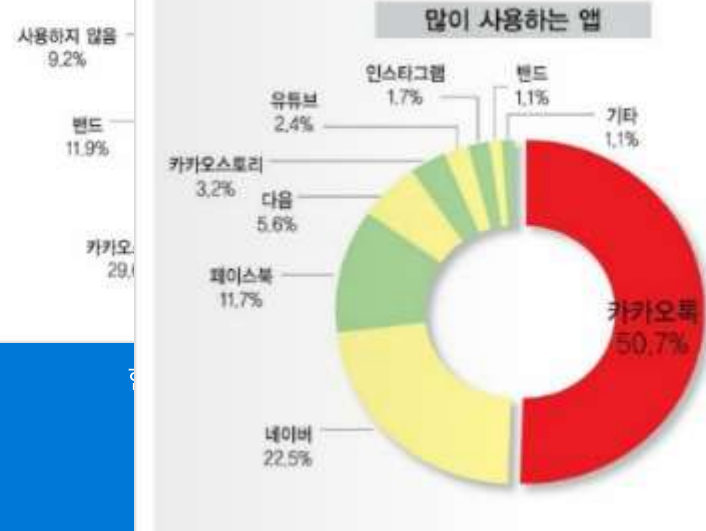
Blog://YoungWook.com

youngwook@outlook.com

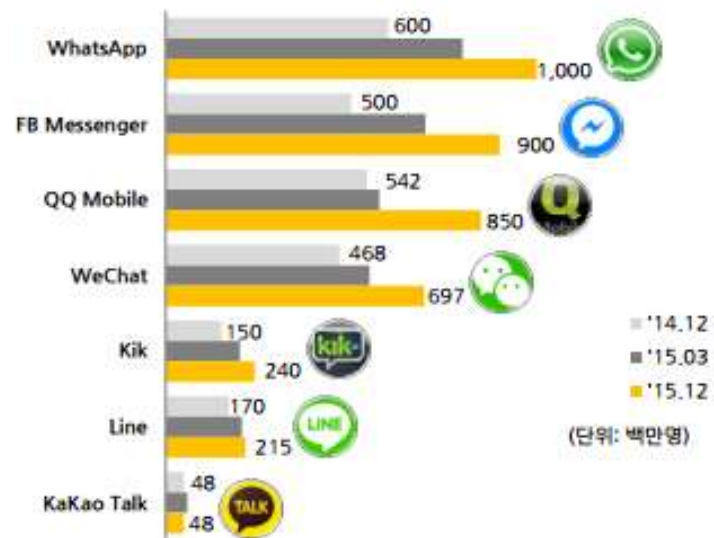
메신저 앱 25.1%



카카오톡 50.7%

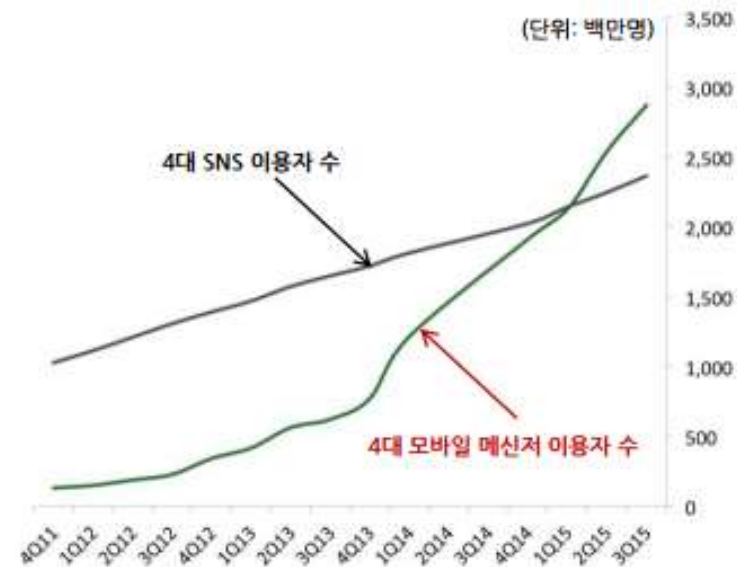


[그림1] 주요 모바일 메신저 이용자 수(MAU)



자료: Digieco 및 각사 발표자료 업데이트

[그림2] 4대 SNS vs. 메신저 이용자 수



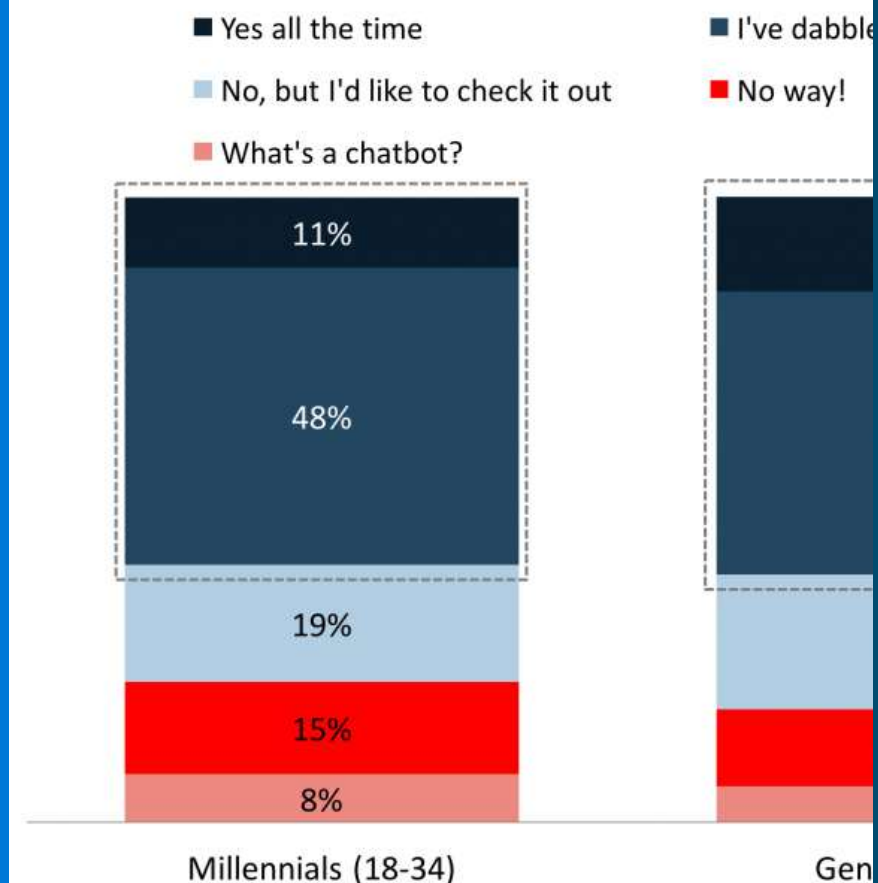
주: 2015년 3분기 기준 (이용자 수 단순 합계)  
자료: Business Insider (2015.11)



대화하는 인공지능, 지금은 챗봇시대!

## US Chatbot Use Among Millennials And Gen Xers

Question: Have you ever interacted with a chatbot over a messaging app?



Source: BI Intelligence survey (n=1,107), October 2016

## CHATBOT ECOSYSTEM

### Deployment Channels



### Third-party Chatbots



### Enabling Technology



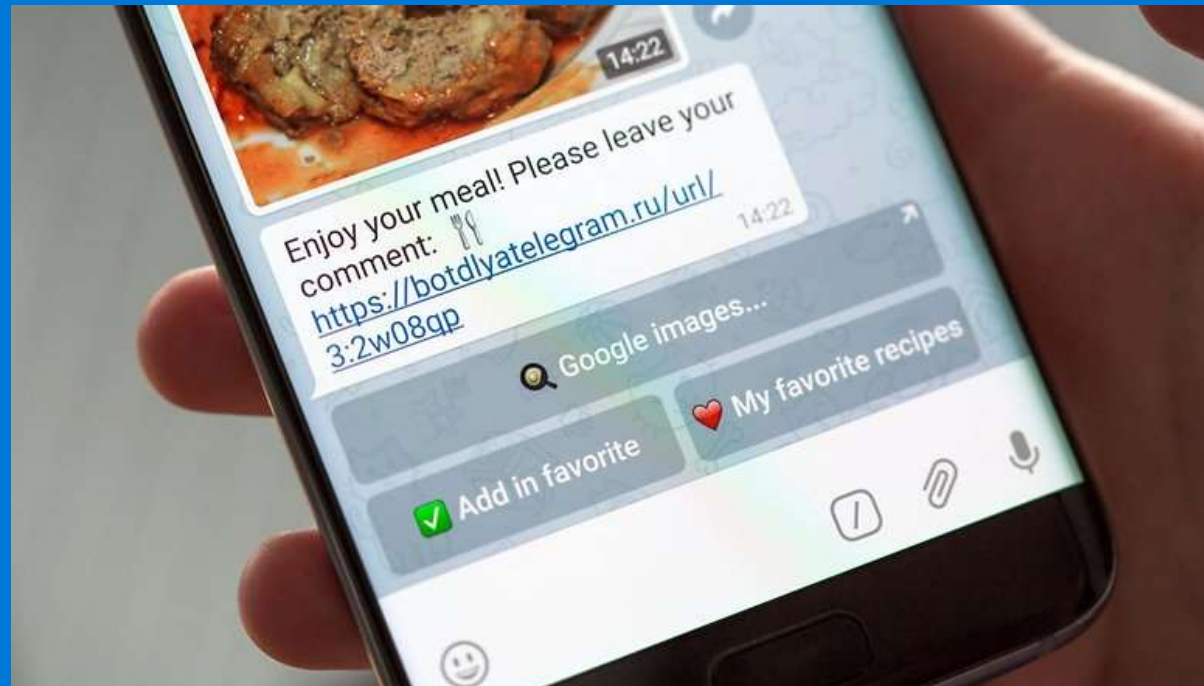
### Native

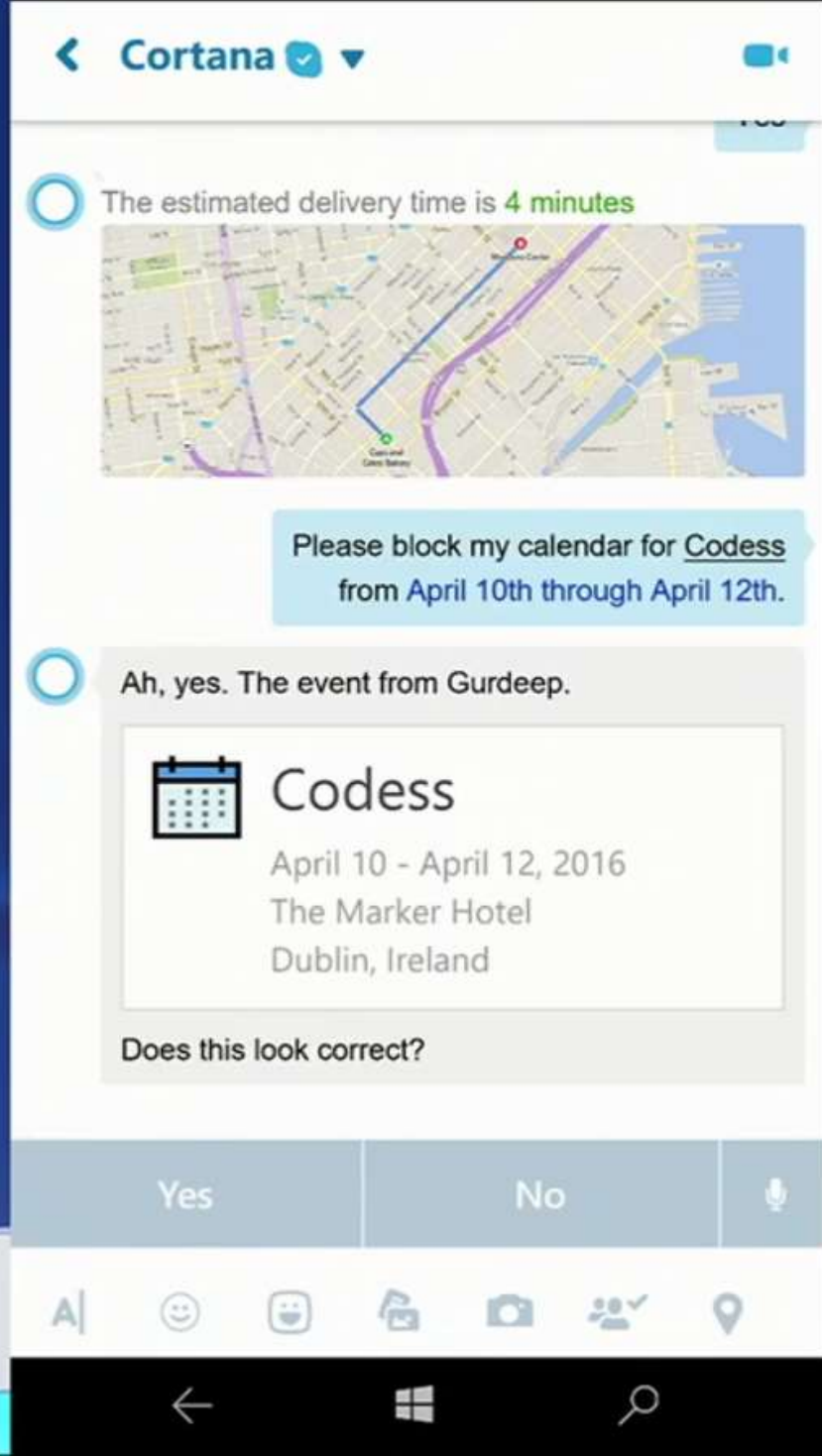


BI INTELLIGENCE



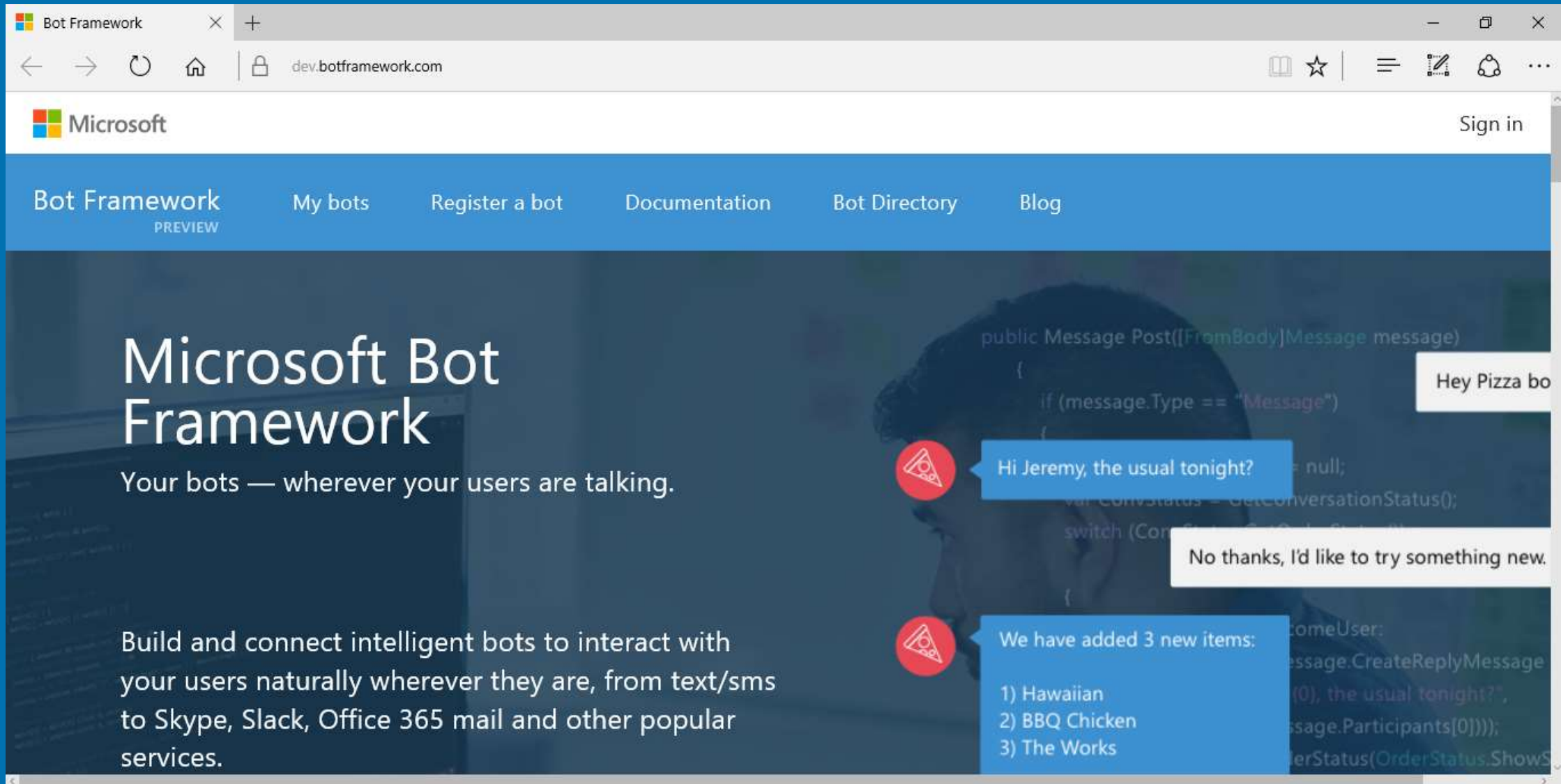
# Microsoft Chatbot.





# MS Bot Framework

<https://dev.botframework.com>



The image is a screenshot of a web browser displaying the Microsoft Bot Framework developer portal. The browser's address bar shows the URL `dev.botframework.com`. The page features a blue header with the Microsoft logo and a navigation menu containing links for Bot Framework, My bots, Register a bot, Documentation, Bot Directory, and Blog. The main content area has a dark background with a large heading "Microsoft Bot Framework" and the tagline "Your bots — wherever your users are talking." Below this, a paragraph describes the framework's capabilities: "Build and connect intelligent bots to interact with your users naturally wherever they are, from text/sms to Skype, Slack, Office 365 mail and other popular services." On the right side, there is a visual representation of a bot conversation. It shows a user's message "Hi Jeremy, the usual tonight?" and a bot's response "We have added 3 new items:" followed by a list: "1) Hawaiian", "2) BBQ Chicken", and "3) The Works". Another user message "No thanks, I'd like to try something new." is also visible. In the background of the conversation, there is a snippet of C# code for a bot's `Post` method.

Bot Framework  
PREVIEW

My bots Register a bot Documentation Bot Directory Blog

## Microsoft Bot Framework

Your bots — wherever your users are talking.

Build and connect intelligent bots to interact with your users naturally wherever they are, from text/sms to Skype, Slack, Office 365 mail and other popular services.

```
public Message Post([FromBody]Message message)
{
    if (message.Type == "Message")
    {
        // ...
    }
}
```

Hey Pizza bo

Hi Jeremy, the usual tonight?

No thanks, I'd like to try something new.

We have added 3 new items:

- 1) Hawaiian
- 2) BBQ Chicken
- 3) The Works

```
comeUser;
message.CreateReplyMessage
(0), the usual tonight?",
message.Participants[0]));
lerStatus(OrderStatus.ShowS
```

# Overview

<http://docs.botframework.com>

Bot Framework Overview

Bot Framework FAQ

Bot Framework Support

Bot Framework Downloads

- ▶ Bot Connector
- ▶ Bot Builder for C#
- ▶ Bot Builder for Node.js
- ▶ Bot Directory

## Bot Framework Overview

The Microsoft Bot Framework provides just what you need to build and connect intelligent bots that interact naturally wherever your users are talking, from text/sms to Skype, Slack, Office 365 mail and other popular services.

### Your Bot Framework Bot

Bot Connector	Bot Builder SDKs	Bot Directory
Connect your bot(s) to text/sms, Office 365 mail, Skype, Slack, Facebook Messenger and more...	Build great dialogs within your Node.js- or C#-based bot	COMING SOON - Try, use, and add published bots to the world's top conversation experiences
<ul style="list-style-type: none"><li>Register, connect, publish and manage your bot through the bot dashboard</li><li>Message routing</li><li>Automatic translation to 30+ languages</li><li>User and state management</li><li>Embeddable web chat control</li><li>Debugging tools</li></ul>	<ul style="list-style-type: none"><li>Open source SDK on Github</li><li>From simple built-in prompts and command dialogs to simple to use yet sophisticated 'FormFlow' dialogs</li><li>Libraries, samples and tools to make a great conversationalist</li><li>Chat emulator</li><li>Leverage related services</li></ul>	<ul style="list-style-type: none"><li>Public directory of bots registered and approved with Bot Framework</li><li>Users can try your bot from the directory via the web chat control</li><li>Users can discover and add your bot to the channels on which it is configured</li></ul>

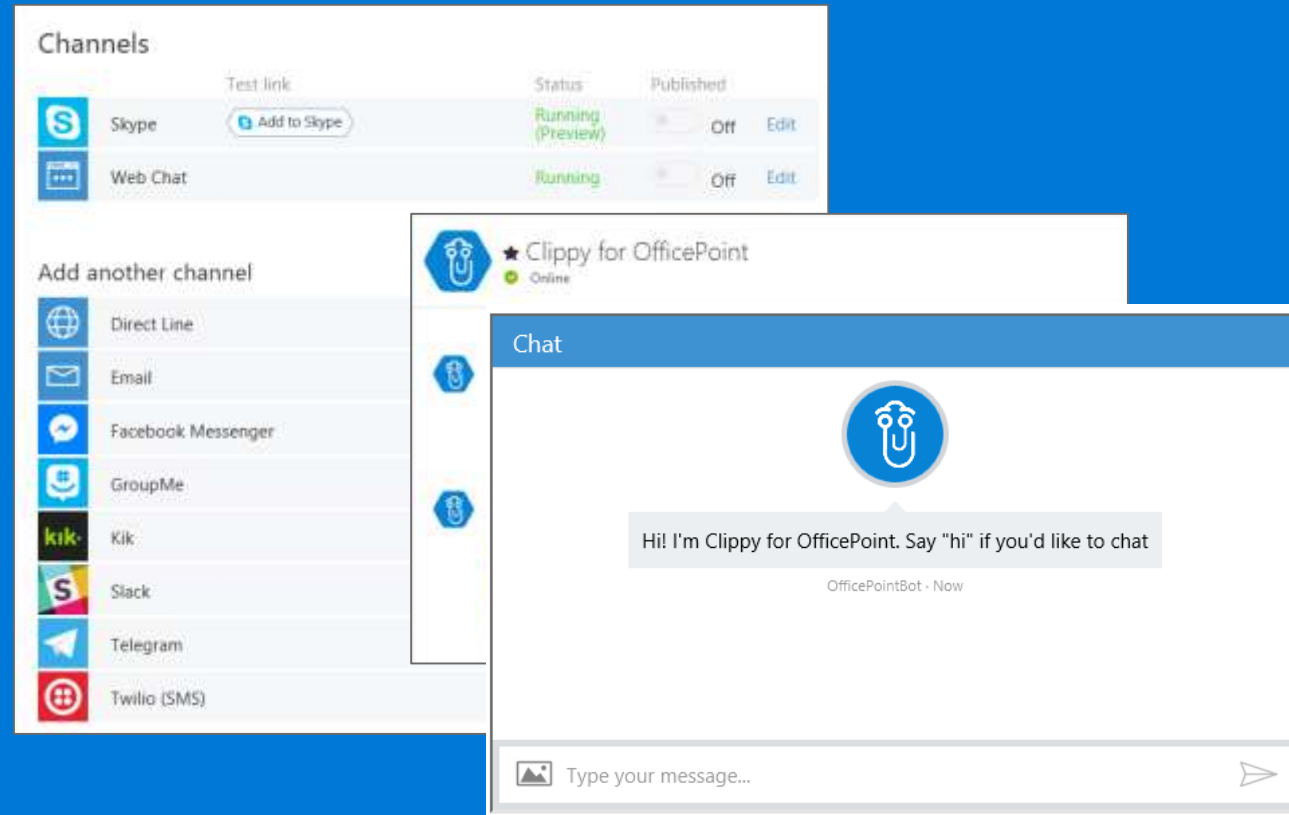
## Options:

- Connector
- Builder
- Directory



# MS Bot Framework

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



# Installing Tools

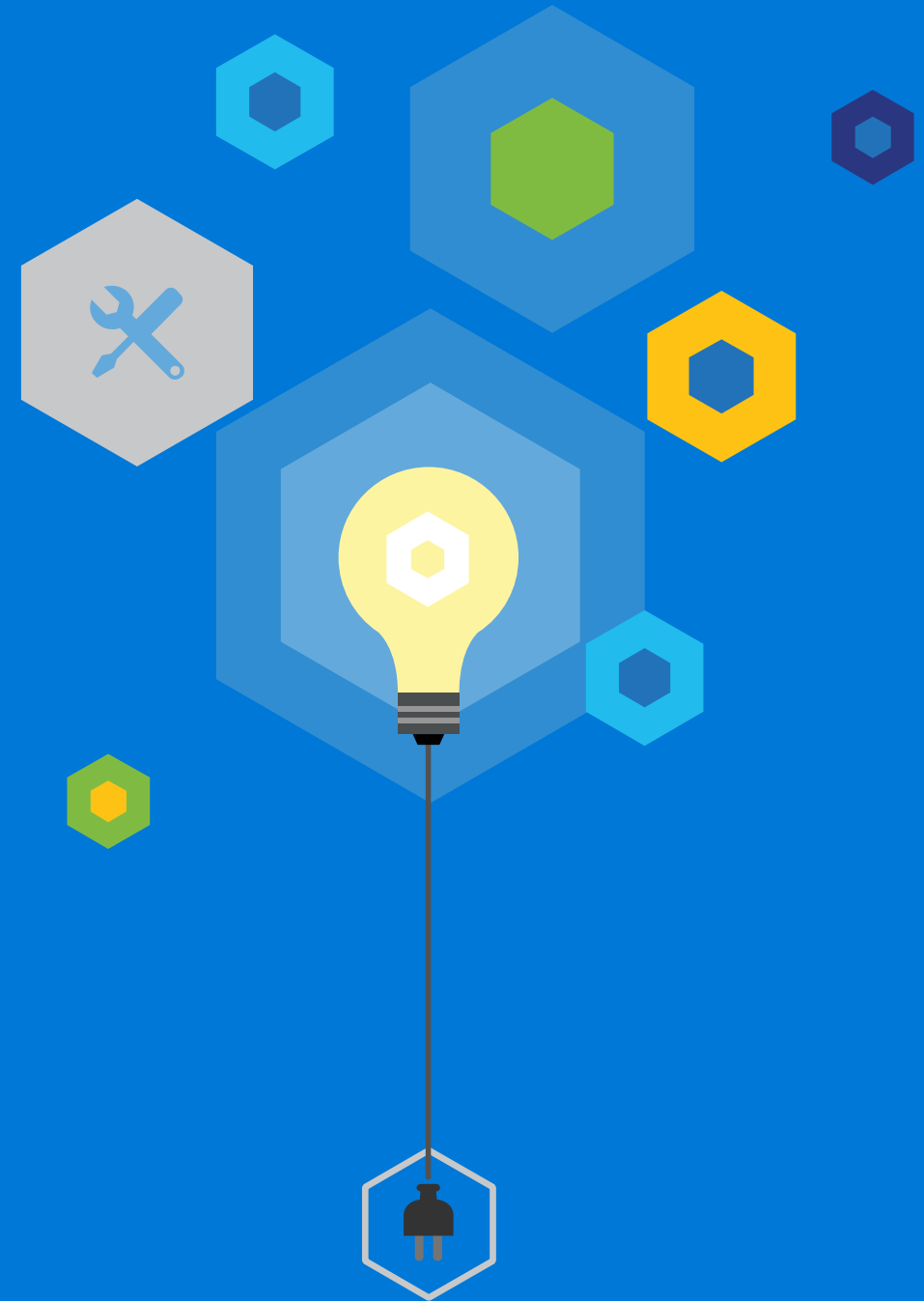
- Microsoft Bot Developer Portal  
<https://dev.botframework.com/>
- Get the Bot Builder SDK  
<https://github.com/Microsoft/BotBuilder>
- Get the Bot Emulator  
<https://download.botframework.com/botconnector/tools/emulator/publish.htm>
- Get the Visual Studio Bot Project Template  
<http://aka.ms/bf-bc-vstemplate>

# Setting Up the Solution

- Download and copy the Bot Project Template
- Create a new Solution with the following:
  - **Bot Application**
- Review the configuration
- Review NuGet dependencies
- Run the Solution to validate

# DEMO

## Setting Up the Solution

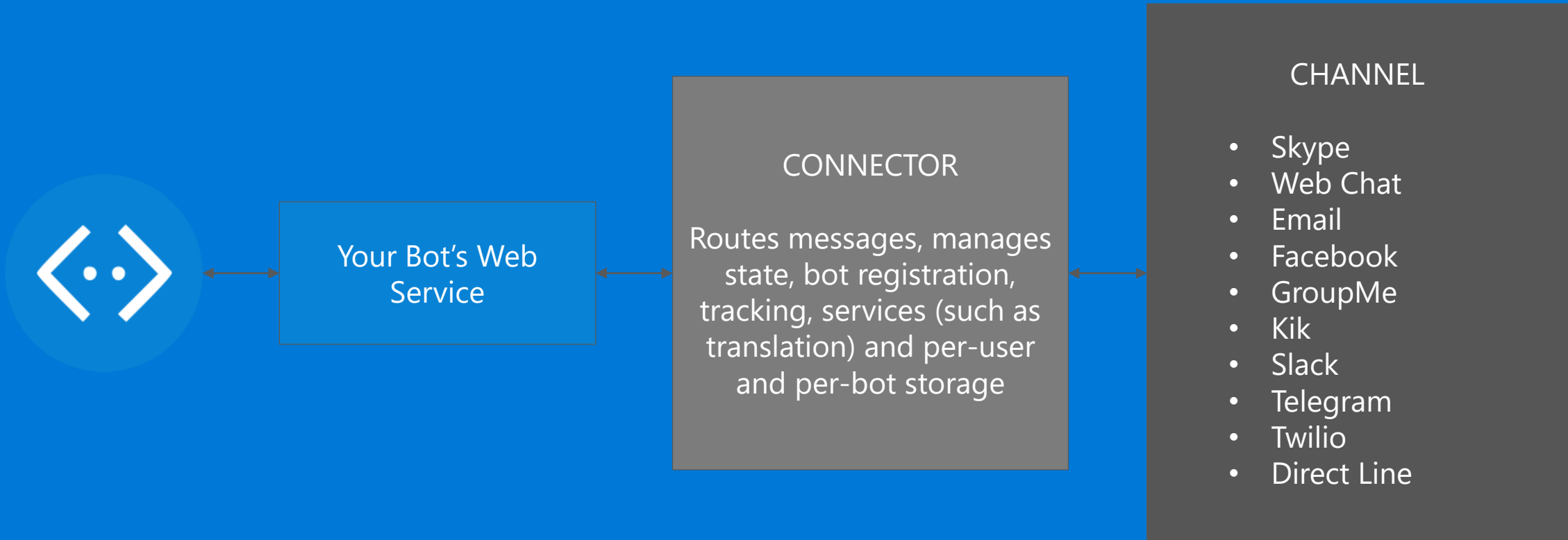




# Connector, Activities & Messages

- The Connector (or Connector Service) handles all communication, conversations, state, and authorization for all activities between a Bot and Users.
- An Activity is a specific event that occurs between a Bot and Users, such as an actual message, or conversation notification.
- A Message is an overt (typically visible) communication between a Bot and Users, such as a User asking a question, or a Bot responding with a reply.


# Connector Service Flow.



# Bot Connector messages

Your bot

Bot  
Connector



```
{
  "type": "Message",
  "id": "68YrxgtB53Y",
  "conversationId": "DphPaFQrDuZDKyCez4AFGcT4vy5aQDje1lLGIjB8v18MFtb",
  "language": "en",
  "text": "You can say \"/order\" to order!",
  "attachments": [ ],
  "from": {
    "name": "+12065551212",
    "channelId": "sms",
    "address": "+12065551212",
    "id": "Ro52hKN287",
    "isBot": false
  },
  "channelData": { SMS data here },
  "botUserData": { your data here },
  ...
}
```

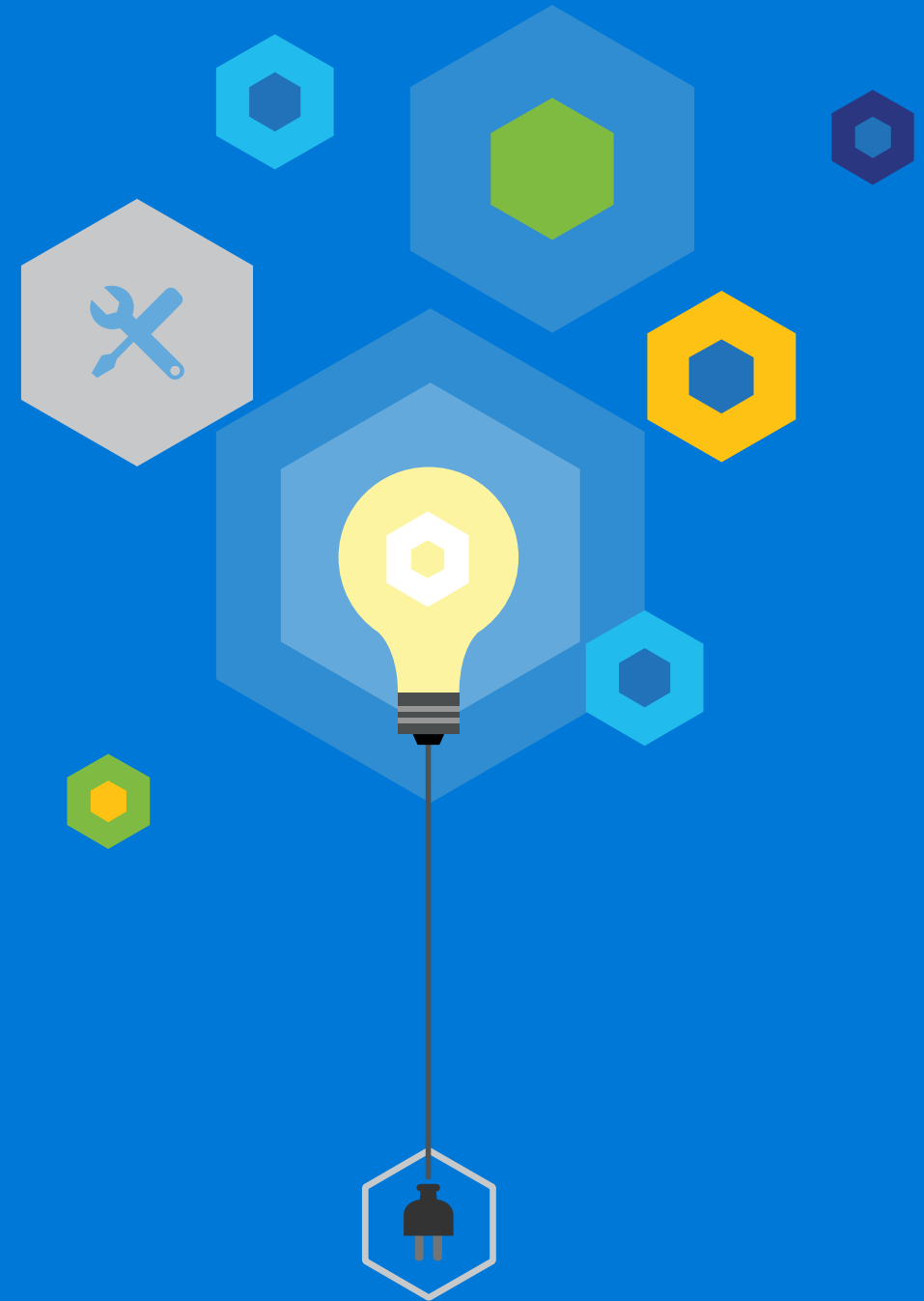
# Types of Activities

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



# DEMO

## Connector Activities & Messages



# 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

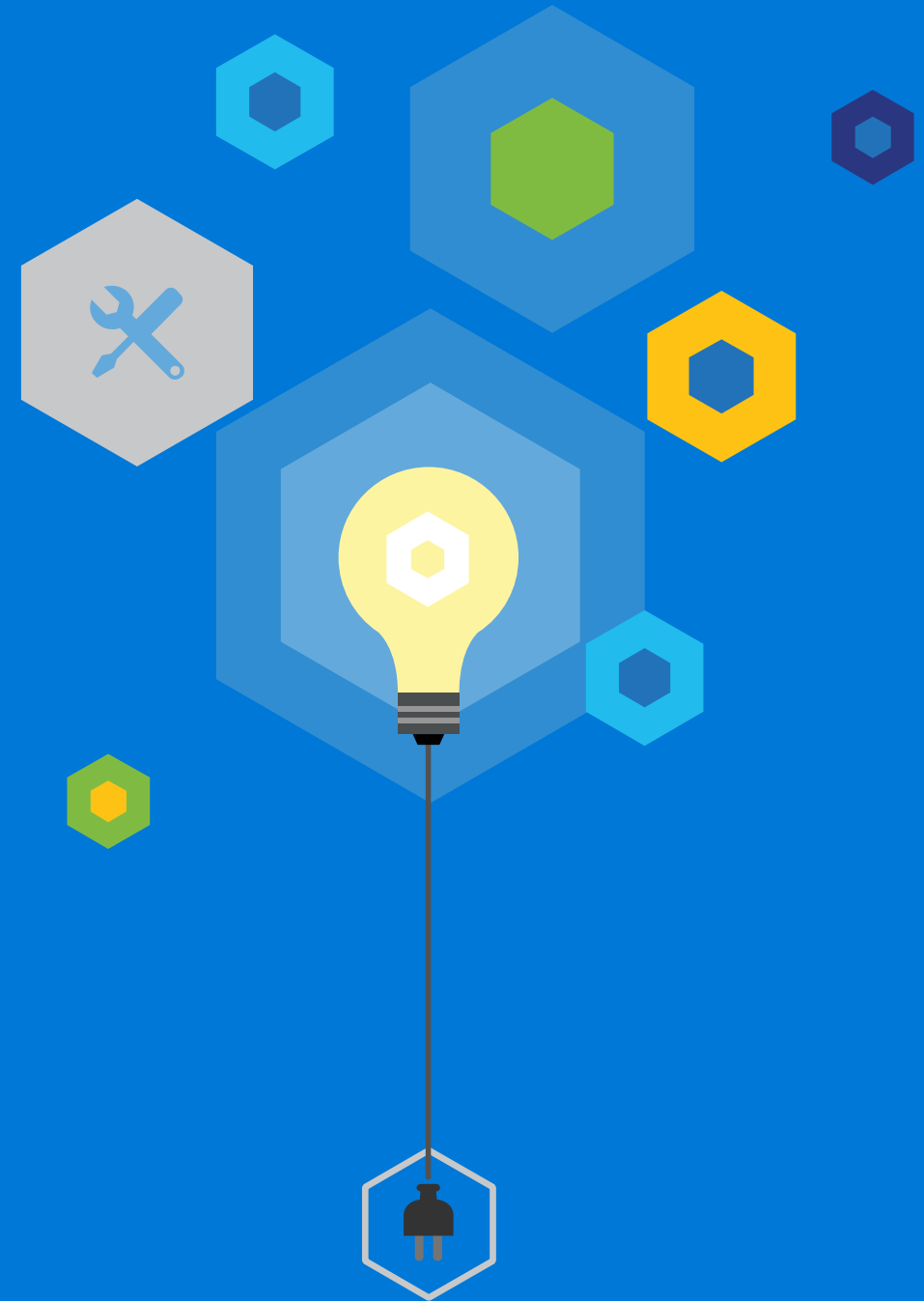
# Using Dialogs

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.

# DEMO

Using State, Authentication &  
Debugs





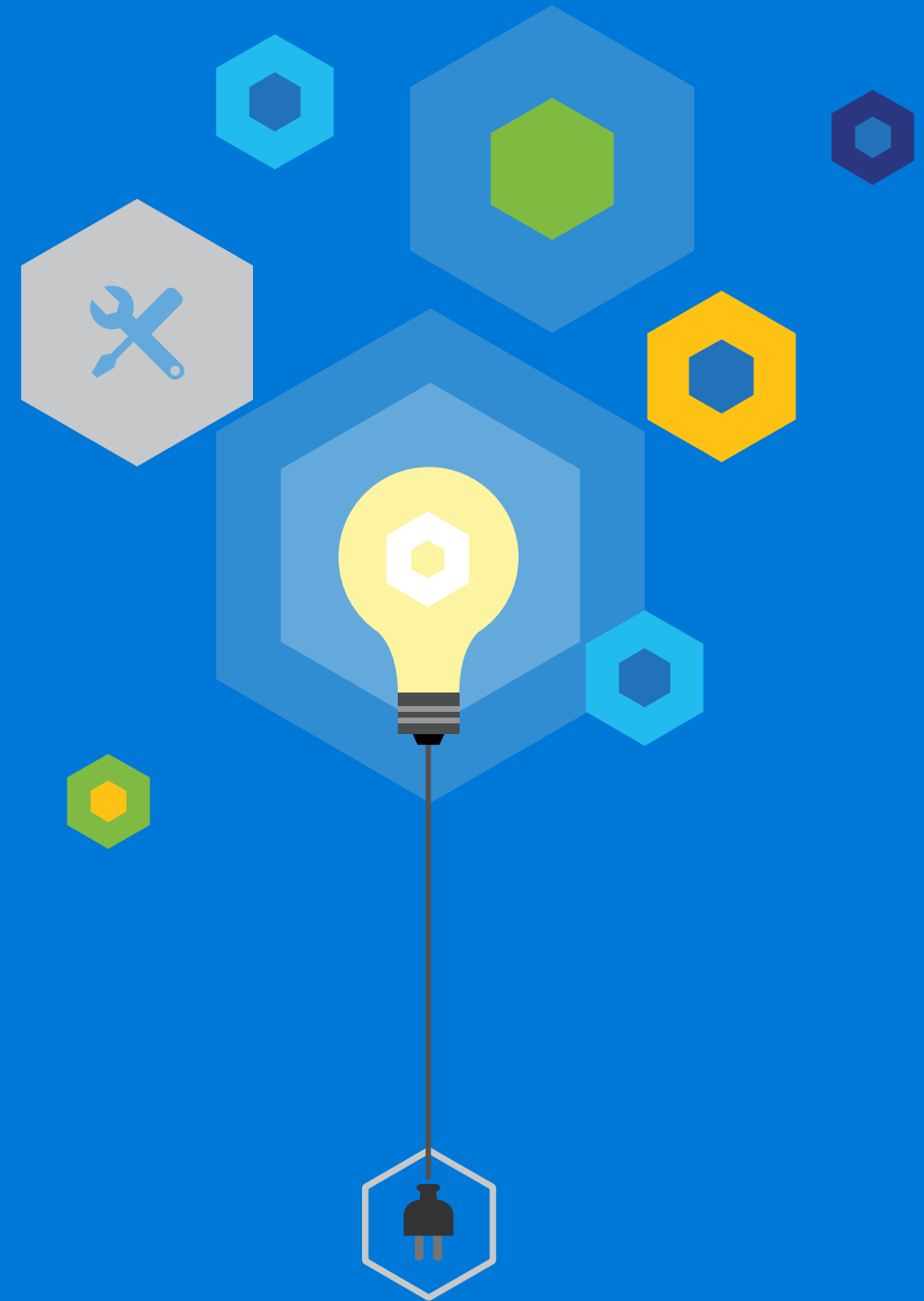
# Using Forms with FormFlow

Although Dialogs are the basic building block of a conversation, it's difficult to create a “guided” conversation. FormFlow creates Dialogs and guides a User through filling in a “form” while providing help and guidance along the way.

- Super smart based on enumerations and properties
- Provide content for various scenarios, such as not understanding, clarification, and confirmations
- Easily provide optional conversation paths based on previous choices

# DEMO

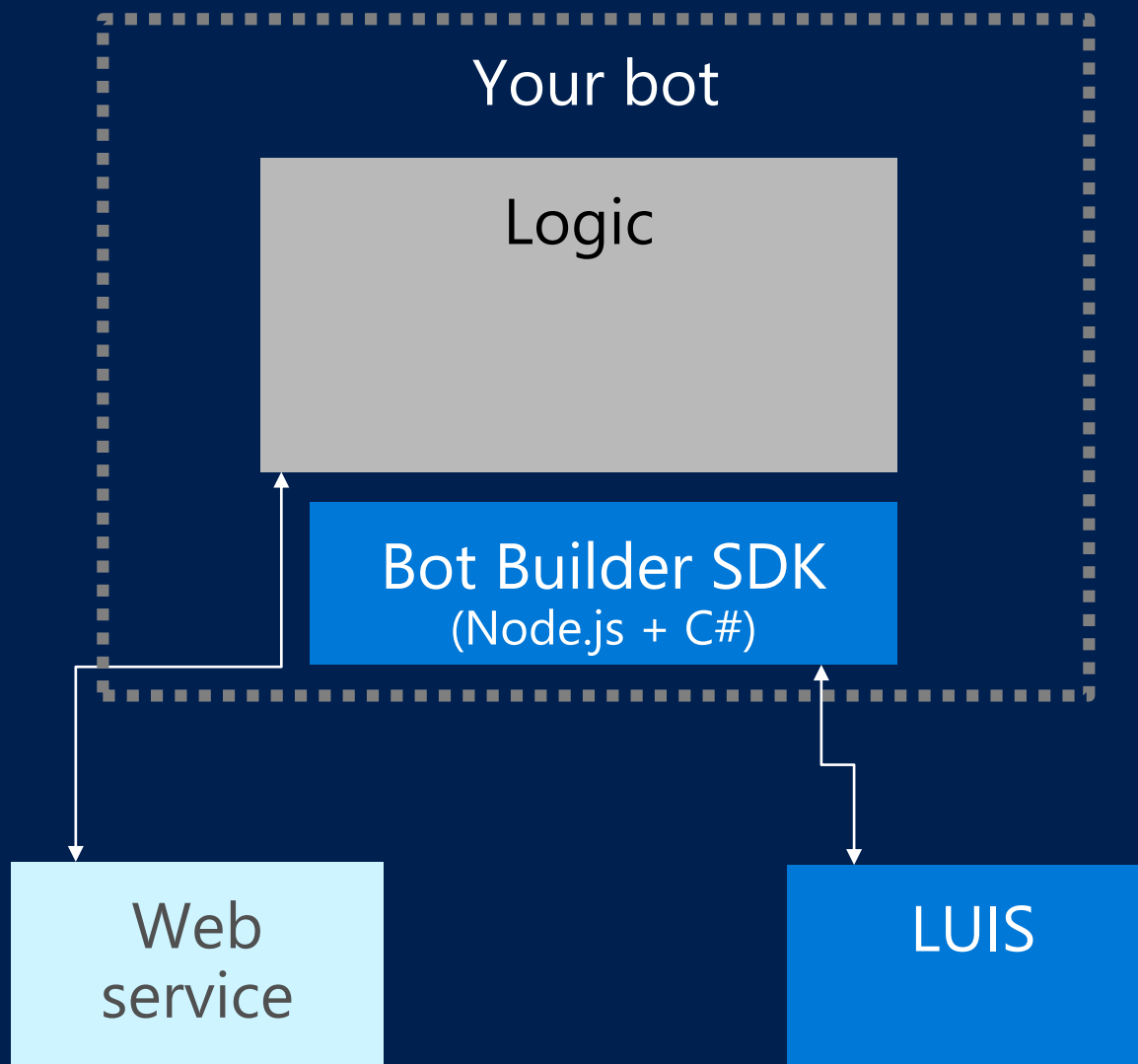
Using Forms with FormFlow



# Integrating Language Understanding Intelligence Services

LUIS is part of Microsoft Cognitive Services offering and can be used for any device, on any platform, and any application scenario.

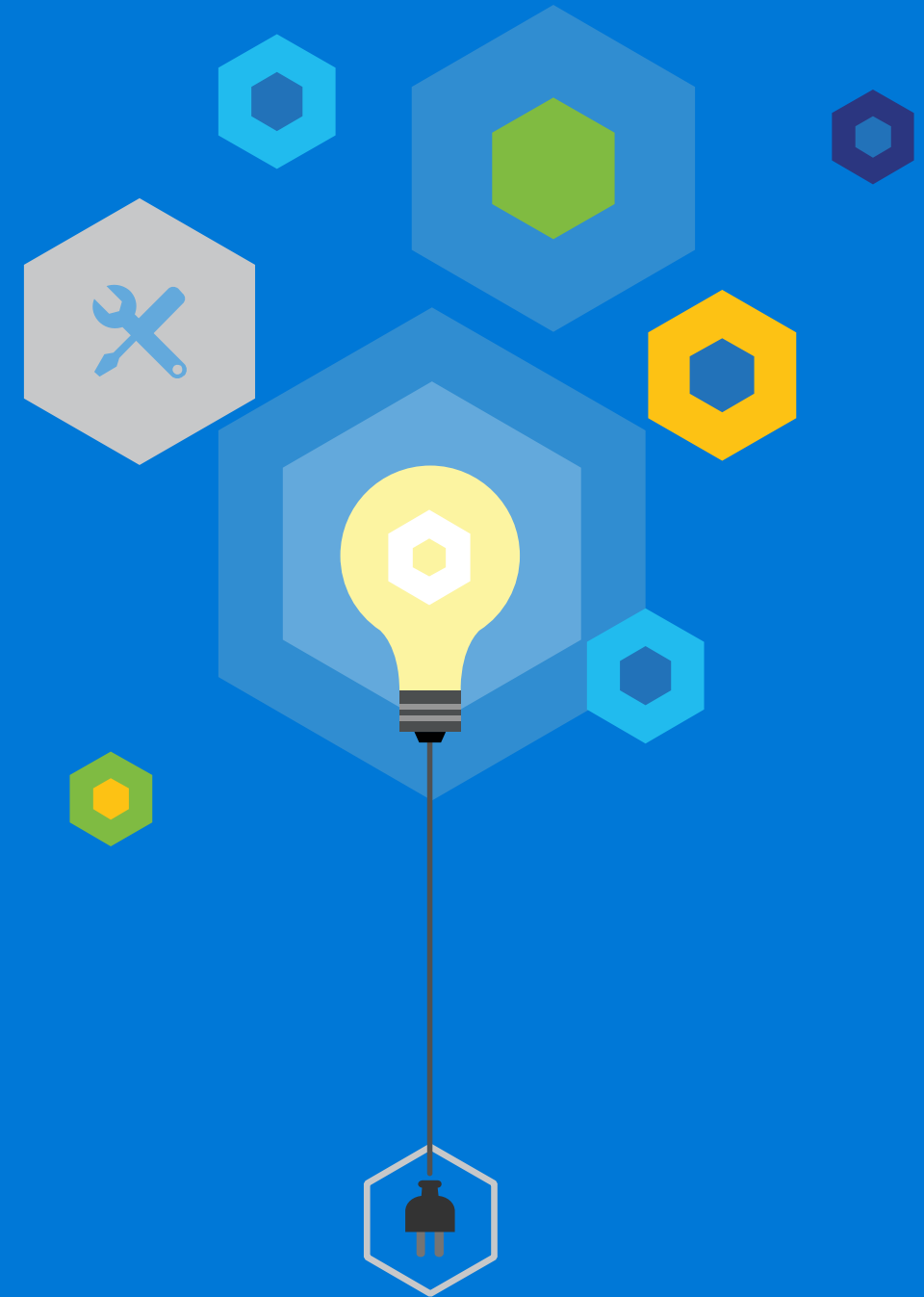
- Essentially what Cortana uses on the backend for language and semantic interpretations
- Provides “built in” logic that can be leveraged “out of the box”
- Natural, adaptable, conversational intelligence
- Model-based via intents and entities





# DEMO

Integration Language Understanding  
Intelligence Services



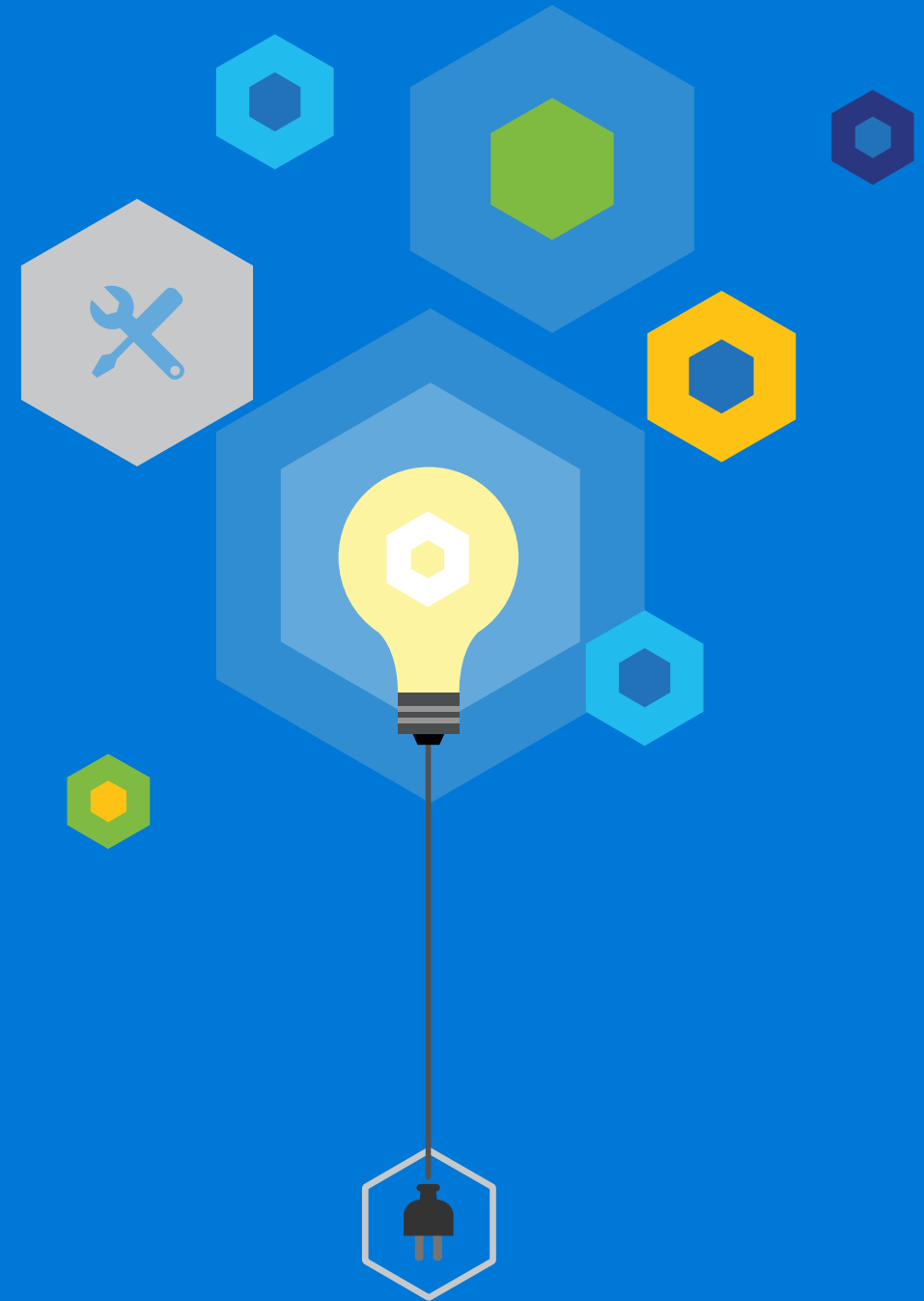
# Putting it All Together

- Add some more features to demonstrate LUIS integration
- Walkthrough an example of using Attachments, Cards and Actions
- Publishing your Bot
- Using your Bot in a web chat and Skype

Putting it All Together

# DEMO

Putting It All Together



# Azure Function

The screenshot displays the Azure Functions portal interface for a function named 'myBlob (Azure Storage Blob)'. The top navigation bar includes 'Develop', 'Integrate', and 'Monitor' tabs. The main workspace is divided into two sections: configuration on the left and code on the right.

**Configuration Section:**

- Trigger:** myBlob (Azure Storage Blob)
- Input:** + New input
- Output:** (empty)
- Name the blob for use in your code:** myBlob
- What path should the trigger monitor?:** ocmé
- Behavior:** trigger
- Which storage:** (empty)

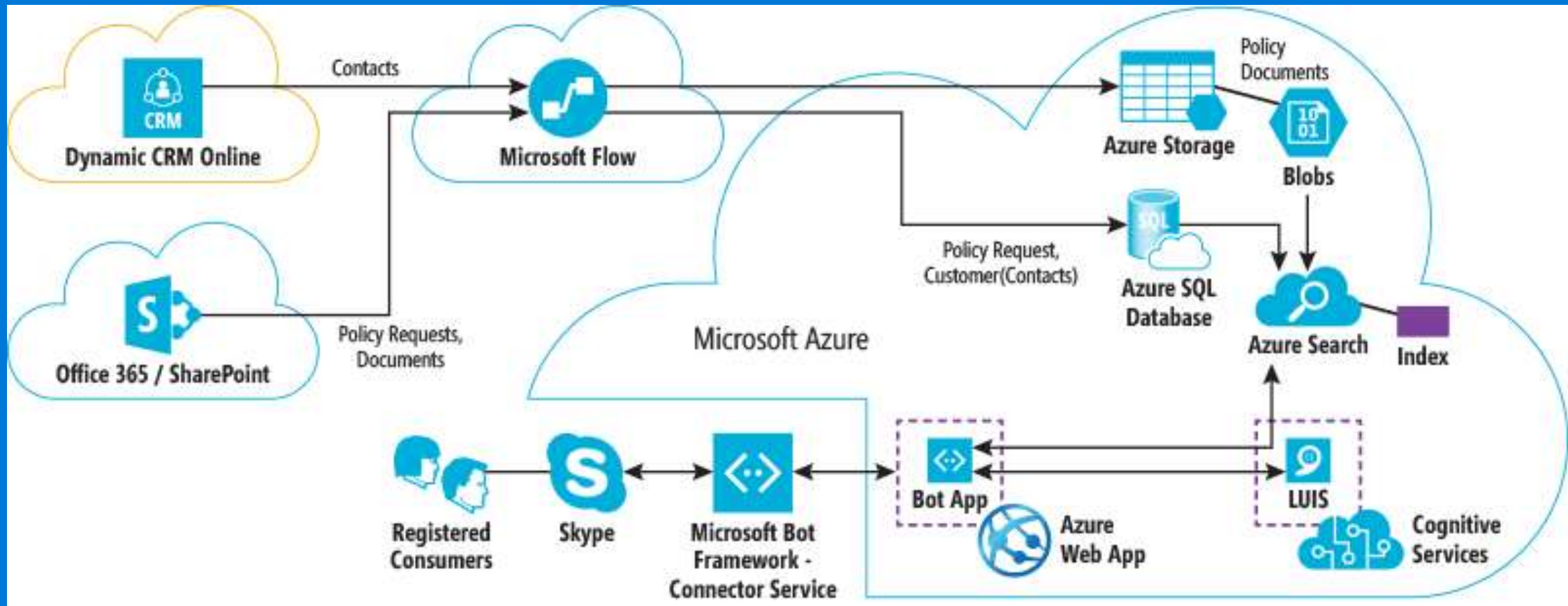
**Code Section:**

```
1 #r "Microsoft.WindowsAzure.Storage"
2 #r "System.Runtime"
3 #r "System.Threading.Tasks"
4 #r "System.IO"
5
6 using System;
7 using System.Threading.Tasks;
8 using Microsoft.WindowsAzure.Storage.Blob;
9 using Microsoft.ProjectOxford.Vision;
10
11 public static async Task Run(ICloudBlob myBlob, TraceWriter log, IAsyncCollector<object> document)
12 {
13     var visionClient = new VisionServiceClient("e6dcf6fa3e4942ac81042bfd1d8af235");
14     var result = await visionClient.RecognizeTextAsync(myBlob.Uri.ToString(), "en");
15
16     var words = from r in result.Regions
17                 from l in r.Lines
18                 from w in l.Words
19                 select w.Text;
20 }
```

**Logs Section:**

2016-03-29T16:53:18 Welcome, you are now connected to log-streaming service.  
2016-03-29T16:54:18 No new trace in the past 1 min(s).  
2016-03-29T16:54:23.661 Function started (Id=821a788a-7c72-434a-87f5-e70738fb1fc5)  
2016-03-29T16:54:23.786 Function started (Id=8423dc01-533d-43d8-a61e-3babe1d85660)  
2016-03-29T16:54:23.879 Function started (Id=f2262b30-b1e6-4c87-8be3-a735898ab602)  
2016-03-29T16:54:24.696 Recognized words: Ella Mashkouski  
2016-03-29T16:54:24.712 Recognized words: Menu p Home Current Statement Posted Transactions DATE v MAR 3 Doing business as: Statements & Activity AMERICAN  
2016-03-29T16:54:24.884 Function completed (Success, Id=821a788a-7c72-434a-87f5-e70738fb1fc5)  
2016-03-29T16:54:25.182 Function completed (Success, Id=8423dc01-533d-43d8-a61e-3babe1d85660)  
2016-03-29T16:54:27.092 Recognized words: Microsoft Completion eLesson: Unconscious Bias Closing Remarks/Completion 1 Completion Contents CREDIT RECEIVED  
2016-03-29T16:54:27.216 Function completed (Success, Id=f2262b30-b1e6-4c87-8be3-a735898ab602)

# Azure Search



# Tips & Tricks

- Use an empty string for your Microsoft App Id and Password in development, but remember to change these to your actual values before you publishing into production.
- Get used to understanding Lambda and “Fluid” concepts to make it easier to work with Dialogs, Forms and especially the concept of the Bot process “Chain”.
- This session demonstrates storing and managing Bot State is a number of ways. Use what makes sense in your environment by uses these concepts purely as examples.
- If the Microsoft Bot Emulator continues to give you authorization issues, try changing the port for your Bot Service in the Web tab of the project and redeploy.

# GitHub!

The screenshot shows the GitHub web interface for the Microsoft/BotBuilder repository. At the top, the navigation bar includes links for Personal, Open source, Business, and Explore, along with Pricing, Blog, and Support. A search bar and Sign in/Sign up buttons are also present. The repository header shows 'Microsoft / BotBuilder' with 198 watchers, 2,131 stars, and 374 forks. Below this, tabs for Code, Issues (72), Pull requests (1), Pulse, and Graphs are visible. The main description states that the Microsoft Bot Builder SDK is part of the Microsoft Bot Framework, which provides tools for building intelligent bots. It includes a link to <http://botframework.com>. A progress bar shows repository statistics: 706 commits, 4 branches, 29 releases, and 26 contributors. Below the progress bar, there are buttons for 'Branch: master', 'New pull request', 'Find file', and 'Clone or download'. A recent commit by Stevenic is shown, merging pull request #409 from mulyoved/master, with the latest commit c2dcafc made 9 days ago. At the bottom, a file named 'CSharp' is listed with a note to 'Update version to 1.2.3.0' made 12 days ago.

GitHub - Microsoft/BotBuilder

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

Microsoft / BotBuilder

Watch 198 Star 2,131 Fork 374

Code Issues 72 Pull requests 1 Pulse Graphs

The Microsoft Bot Builder SDK is one of three main components of the Microsoft Bot Framework. The Microsoft Bot Framework provides just what you need to build and connect intelligent bots that interact naturally wherever your users are talking, from text/SMS to Skype, Slack, Office 365 mail and other popular services. <http://botframework.com>

706 commits 4 branches 29 releases 26 contributors

Branch: master New pull request Find file Clone or download

Stevenic Merge pull request #409 from mulyoved/master Latest commit c2dcafc 9 days ago

CSharp Update version to 1.2.3.0 12 days ago

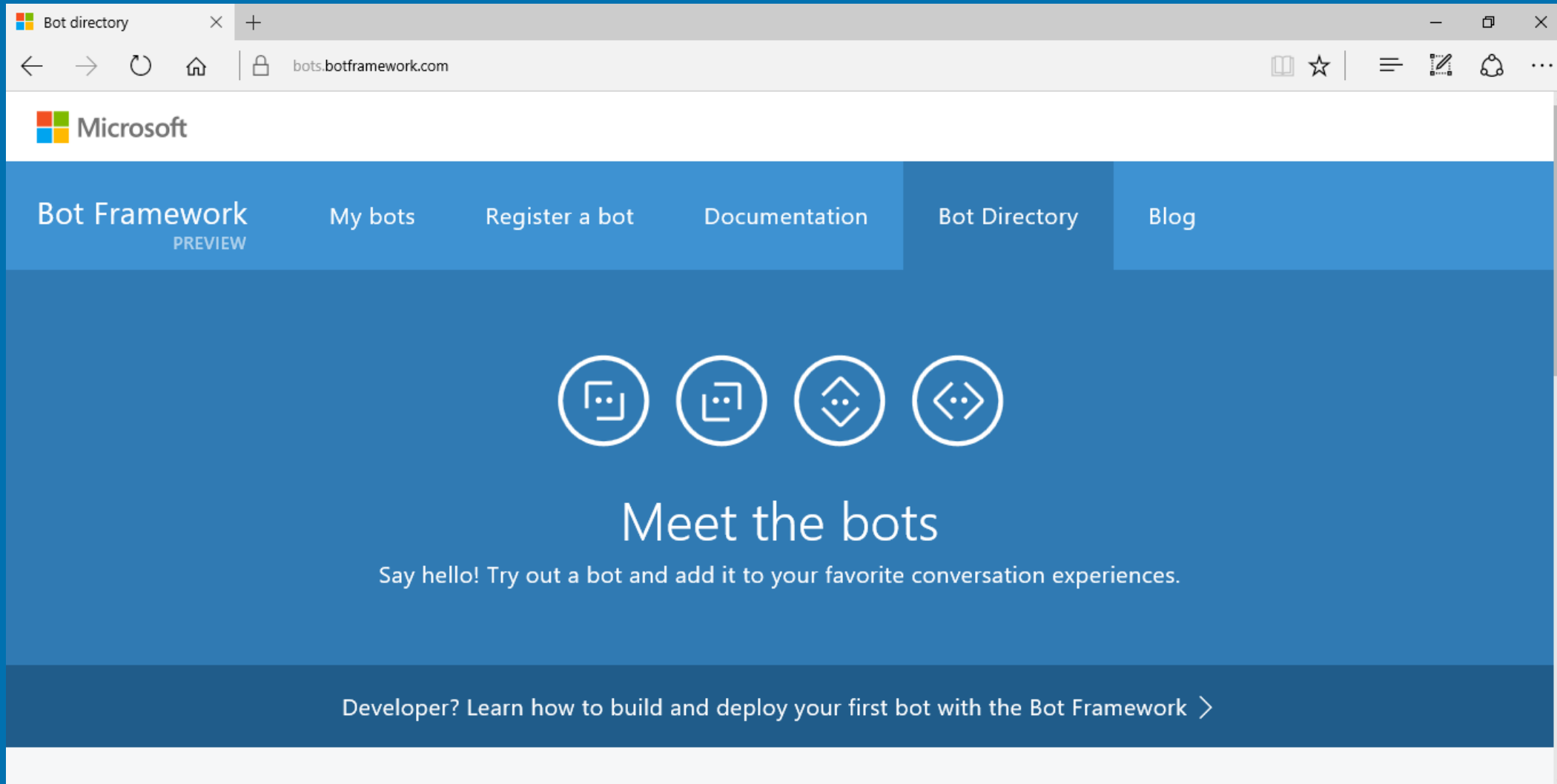
<https://github.com/Microsoft/botbuilder>



감사합니다

Thank you~!

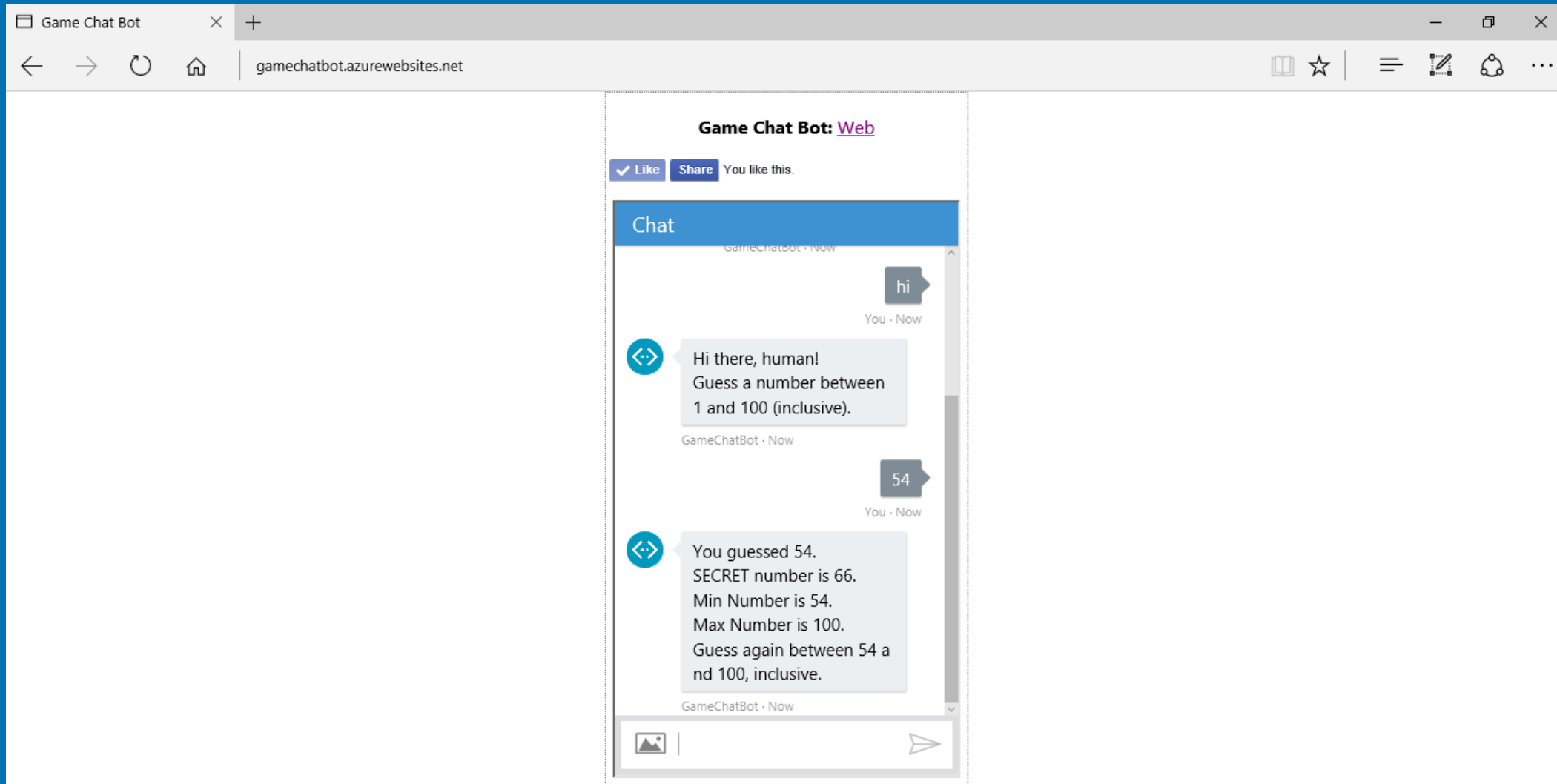
# Bot Directory



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

Demo

# Game Chat Bot



<http://gamechatbot.azurewebsites.net/>



40 OLD  
TOWN

BIRKSEN

(020)  
76226466

40 OLD  
TOWN

BIRKSEN

(020)  
76226466

flowers by



# Variety of Creative Apps



How-Old.net  
HOW OLD DO I LOOK? #HowOldNet

4

Sorry if we didn't quite get the age and gender right - we are still working on this feature

Try Another Photo!



TwinsOrNot.net  
Are you twins? #TwinsOrNotRobot

100%  
Perfect match

Help us improve or start over

You can also click a pic to compare to someone new

Powered by Project Oxford

365 1.85K

Privacy & Cookies | Terms of Use | Trademark | © 2015 Microsoft

Microsoft



MyMoustache.net  
15,381 faces analyzed and counting #MyMoustacheRobot

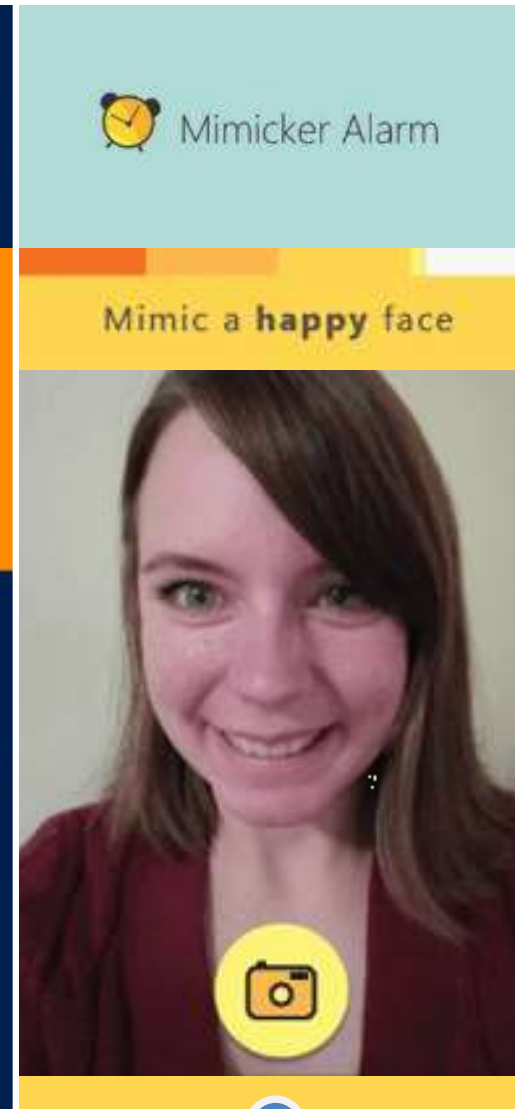
De-stache Me! Try Again!

Sorry if we didn't quite get the moustache quite right - we are still improving this feature

It doesn't look like you have a moustache! #NOSTACHE

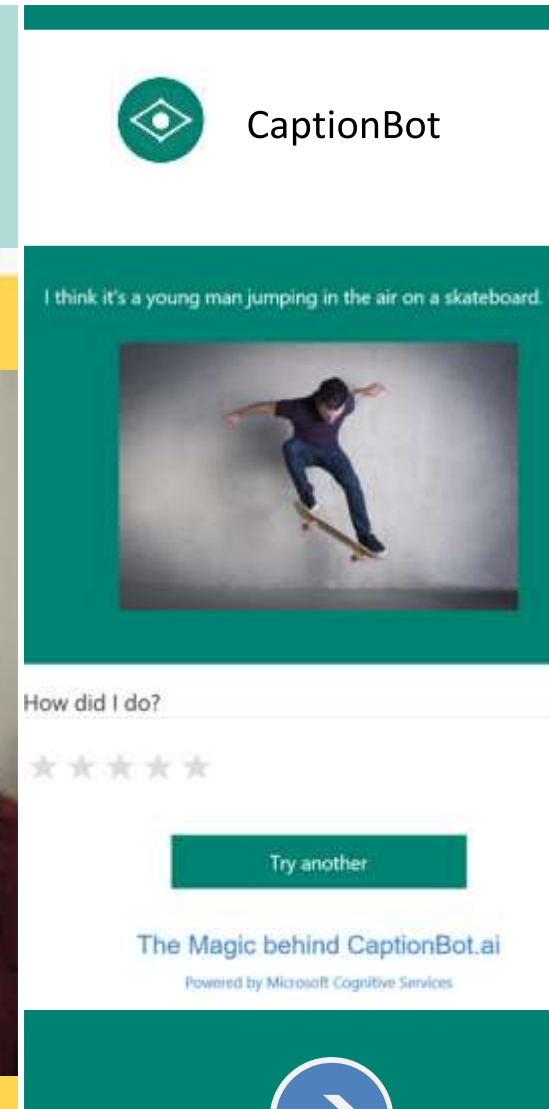
WAVE ME! GETTING THERE! ULTIMATE ME!

You  
Everyone Else (16086)



Mimicker Alarm

Mimic a happy face



CaptionBot

I think it's a young man jumping in the air on a skateboard.

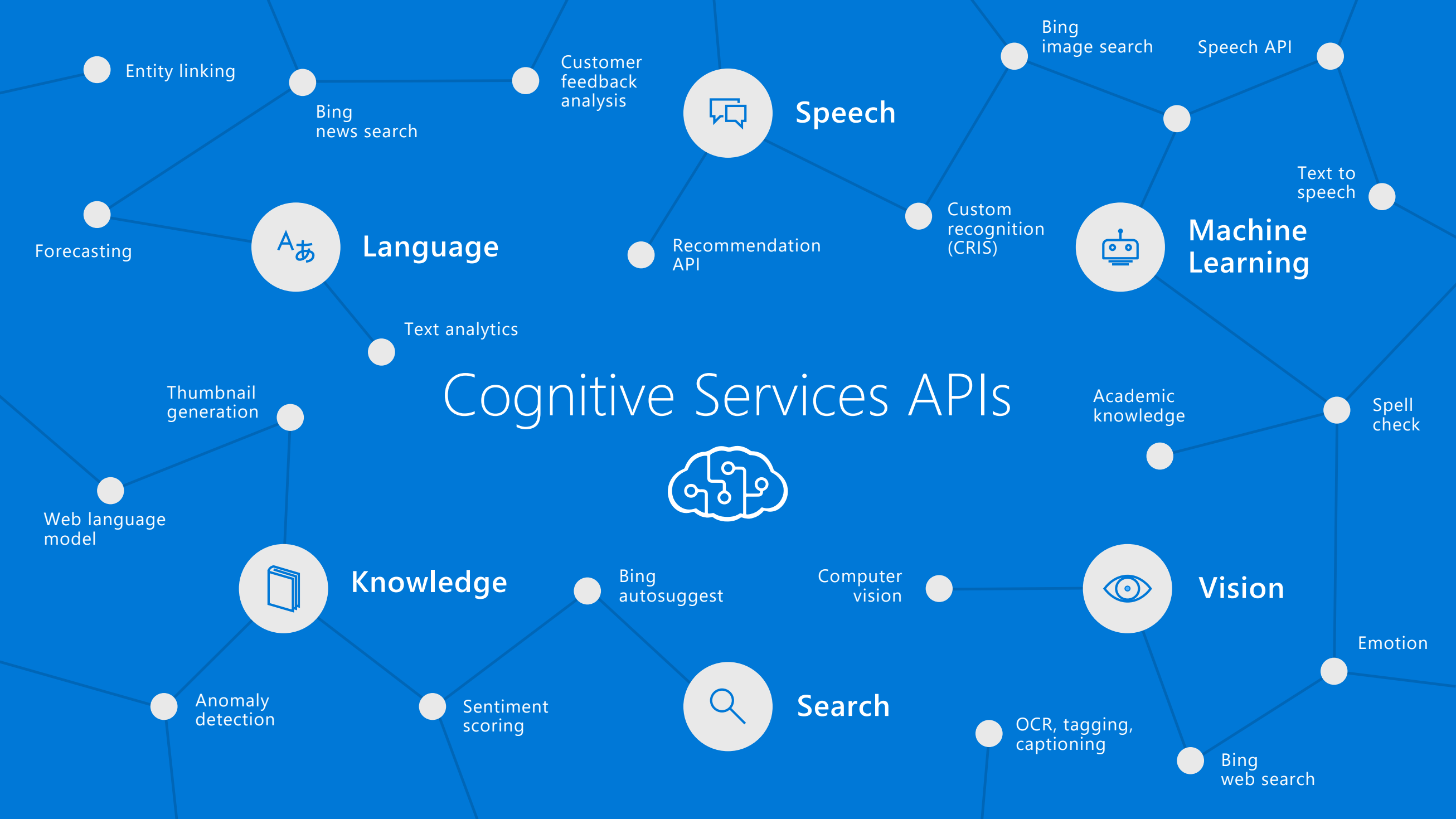
How did I do?

★★★★★

Try another

The Magic behind CaptionBot.ai  
Powered by Microsoft Cognitive Services












# Cognitive Services

microsoft.com/cognitive

 Vision	 Speech	 Language	 Knowledge	 Search
Computer Vision	Custom Recognition	Bing Spell Check	Academic Knowledge	Bing Web Search
Emotion	Speaker Recognition	Linguistic Analysis	Entity Linking	Bing Image Search
Face	Speech	Language Understanding	Knowledge Exploration	Bing Video Search
Video	Translator	Text Analytics	Recommendations	Bing News Search
		WebLM		Bing Autosuggest

<https://www.microsoft.com/cognitive-services/>