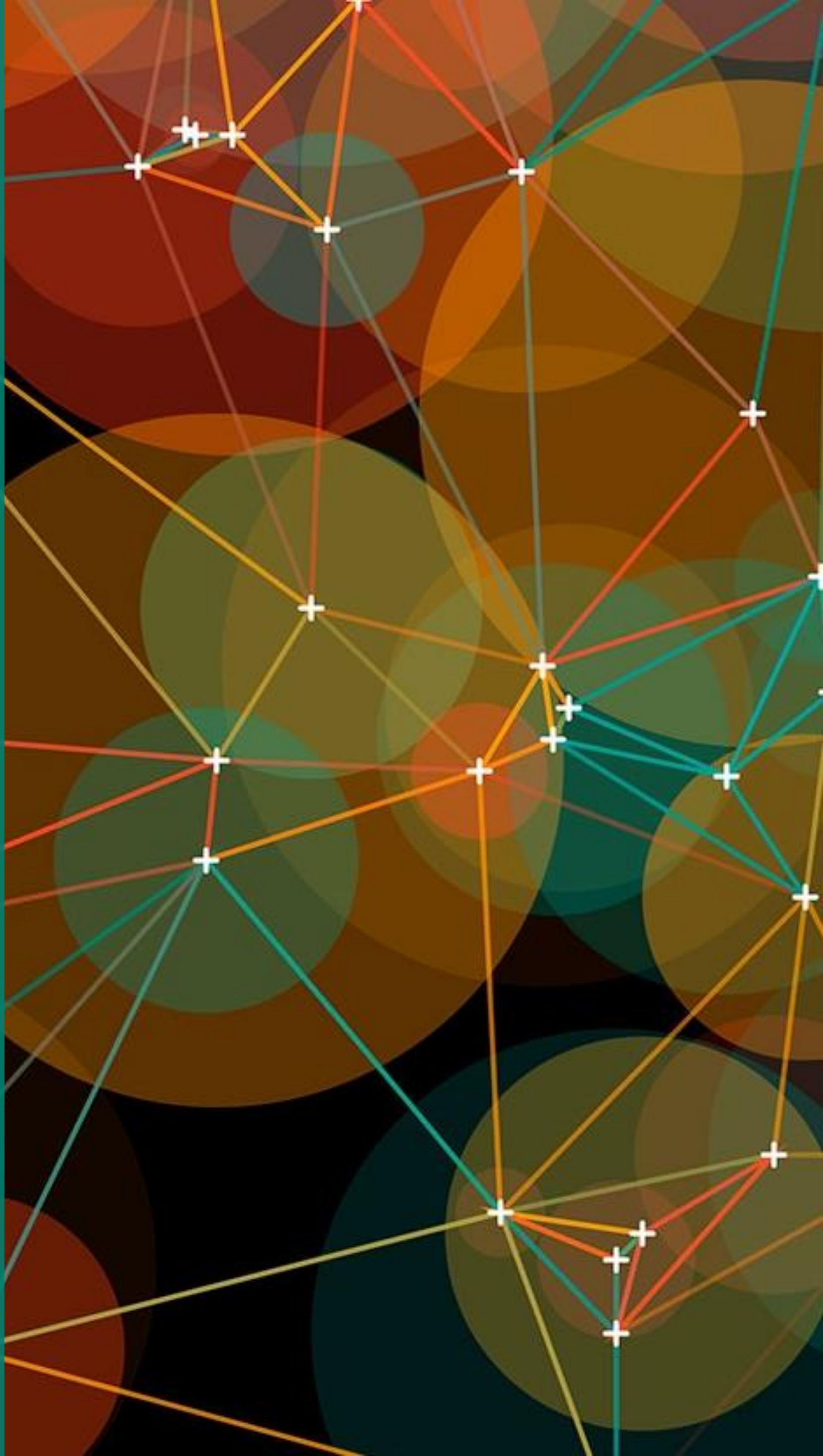
Build your own (ro)bot

Michal Zylinski
Microsoft

michal.zylinski@microsoft.com

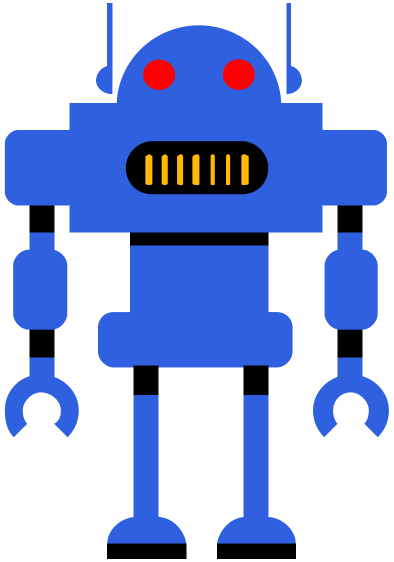


Agenda

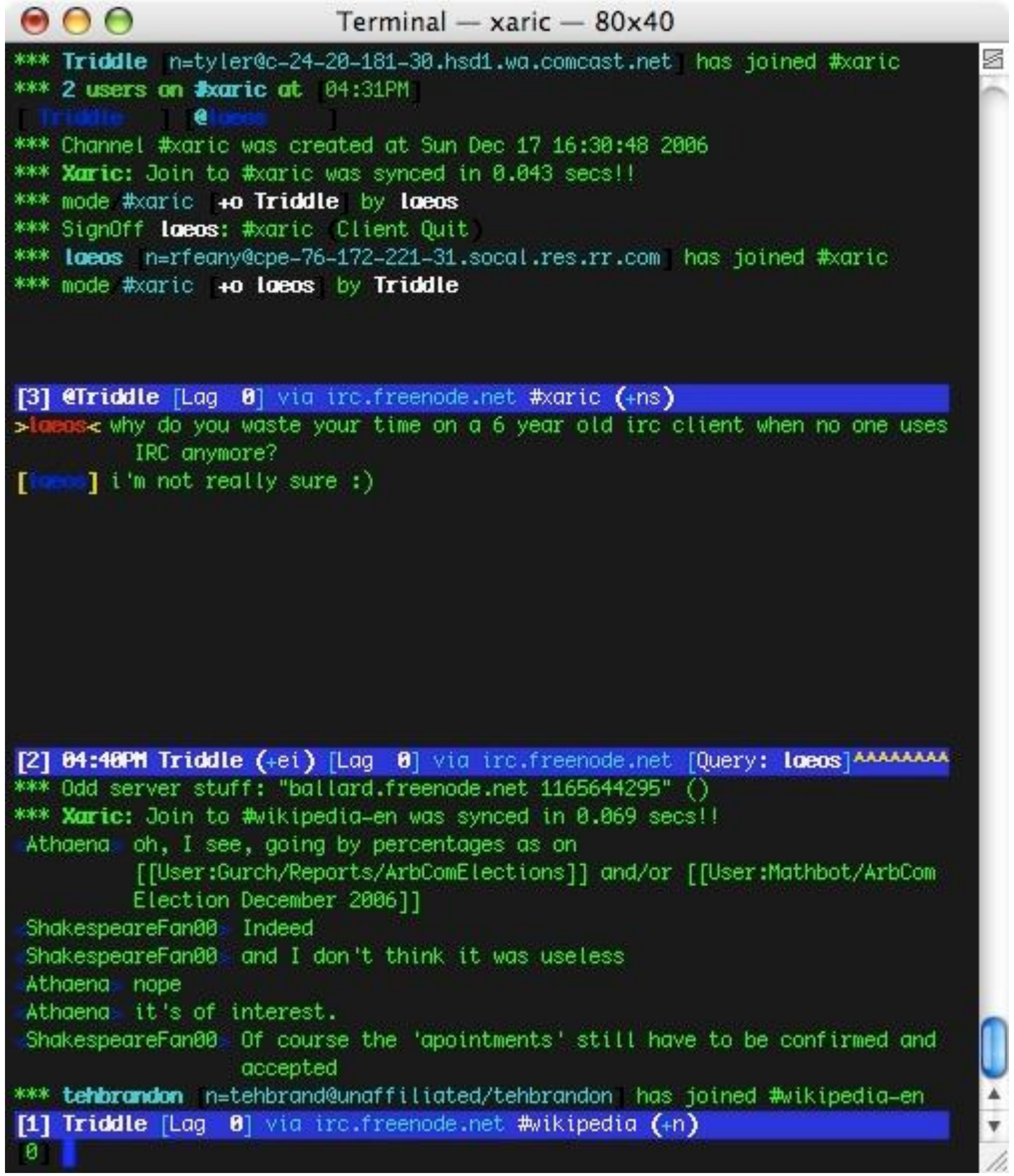
- Motivation
- Design
- Develop
- Deploy
- Test

Motivation

What bots are NOT:



- AI
- Machine Learning
- Self awareness
- A replacement for every UX we use today



```
*** Triddle [n=tyler@c-24-20-181-30.hsd1.wa.comcast.net] has joined #xaric
*** 2 users on #xaric at [04:31PM]
[ Triddle    ] [@laeos      ]
*** Channel #xaric was created at Sun Dec 17 16:30:48 2006
*** Xaric: Join to #xaric was synced in 0.043 secs!!
*** mode #xaric [+o Triddle] by laeos
*** SignOff laeos: #xaric (Client Quit)
*** laeos [n=rfeany@cpe-76-172-221-31.socal.res.rr.com] has joined #xaric
*** mode #xaric [+o laeos] by Triddle



[3] @Triddle [Lag  0] via irc.freenode.net #xaric (+ns)
>laeos< why do you waste your time on a 6 year old irc client when no one uses
         IRC anymore?
[laeos] i'm not really sure :)









[2] 04:40PM Triddle (+ei) [Lag  0] via irc.freenode.net [Query: laeos]^^^^^^^^
*** Odd server stuff: "ballard.freenode.net 1165644295" ()
*** Xaric: Join to #wikipedia-en was synced in 0.069 secs!!
<Athaena> oh, I see, going by percentages as on
          [[User:Gurch/Reports/ArbComElections]] and/or [[User:Mathbot/ArbCom
          Election December 2006]]
<ShakespeareFan00> Indeed
<ShakespeareFan00> and I don't think it was useless
<Athaena> nope
<Athaena> it's of interest.
<ShakespeareFan00> Of course the 'apointments' still have to be confirmed and
                   accepted
*** tehbrandon [n=tehbrand@unaffiliated/tehbrandon] has joined #wikipedia-en
[1] Triddle [Lag  0] via irc.freenode.net #wikipedia (+n)
[0]
```

...and certainly not such thing as well...

Bots are services that people interact with through conversation and messaging

Why Bots?

Meet customers where they are at

Communicate with customers the way they do

Interact through the apps your customers use today

More personal than an app

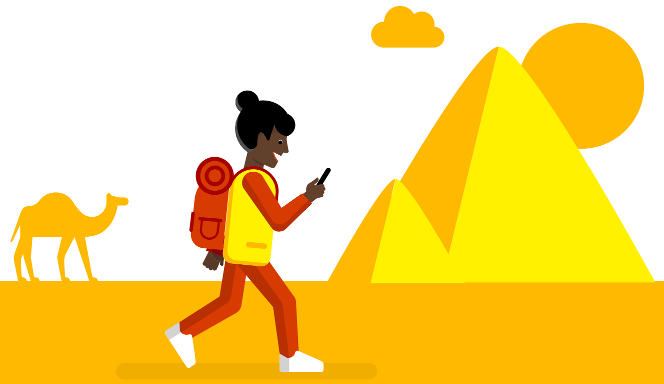
Develop deeper connections

Why now?

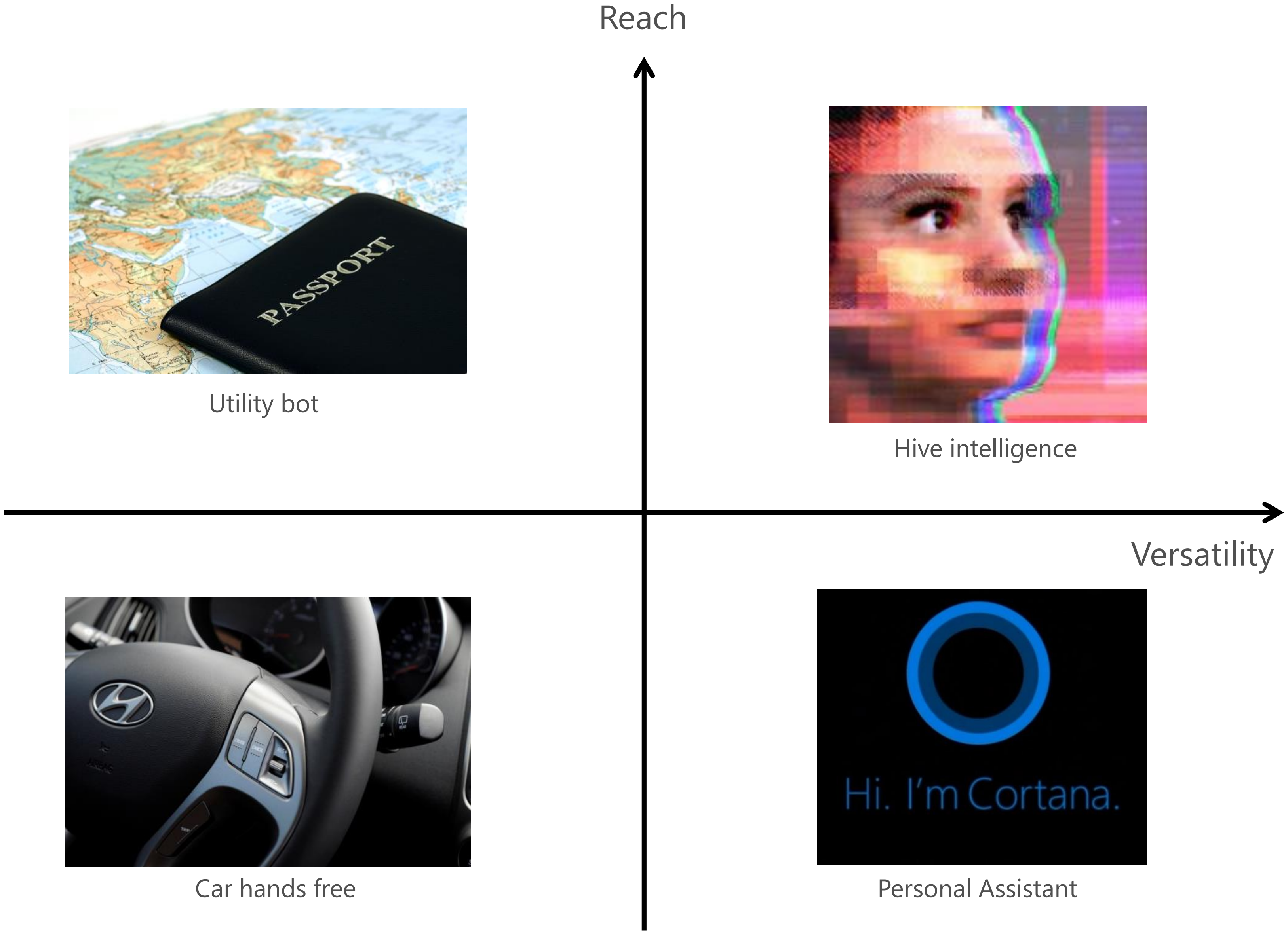
People spend more time messaging

Increase in capability/focus in messaging platforms

Rise in accessible language processing technology



Types of bots



Reach

Utility bot

Hive intelligence

Versatility

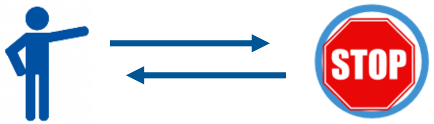
Car hands free

Personal Assistant

Design

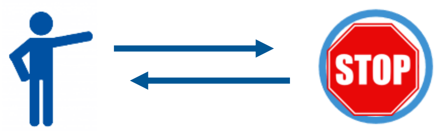
Designing the Interaction

Basic request response
“Info” or „Helper” scenarios

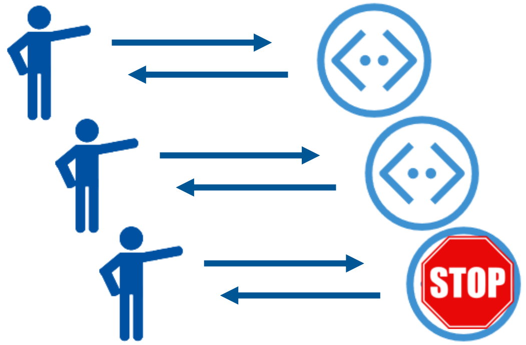


Designing the Interaction

Basic request response
“Info” or „Helper” scenarios

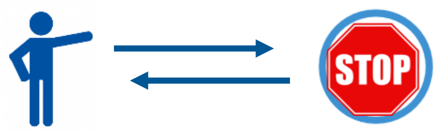


Basic Hierarchy
“Select” scenarios
Minimal state stored

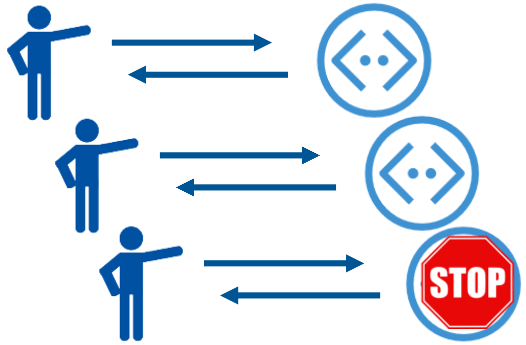


Designing the Interaction

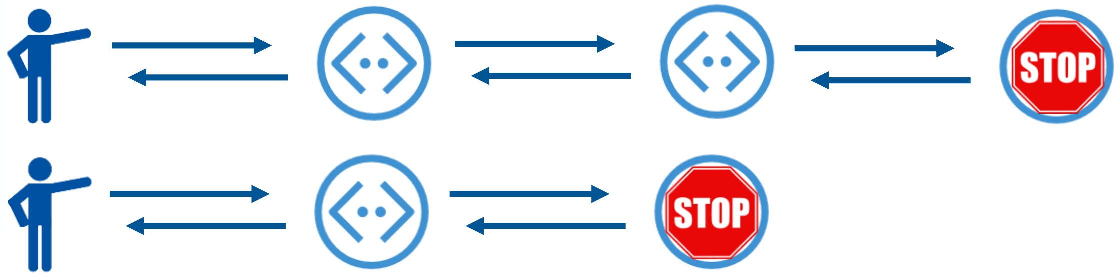
Basic request response
“Info” or „Helper” scenarios



Basic Hierarchy
“Select” scenarios
Minimal state stored



Nested waterfalls
Configuration/multi-step
Requires state + reset

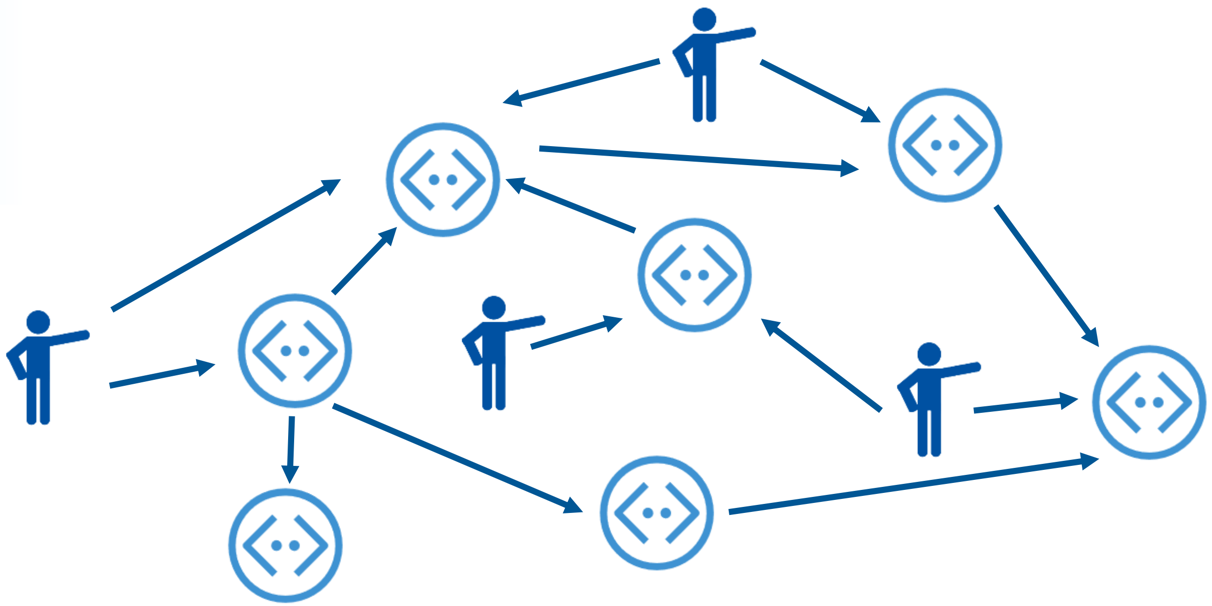


Designing the Interaction

Arbitrary state machine

“Free form” scenarios

Requires navigation for the user



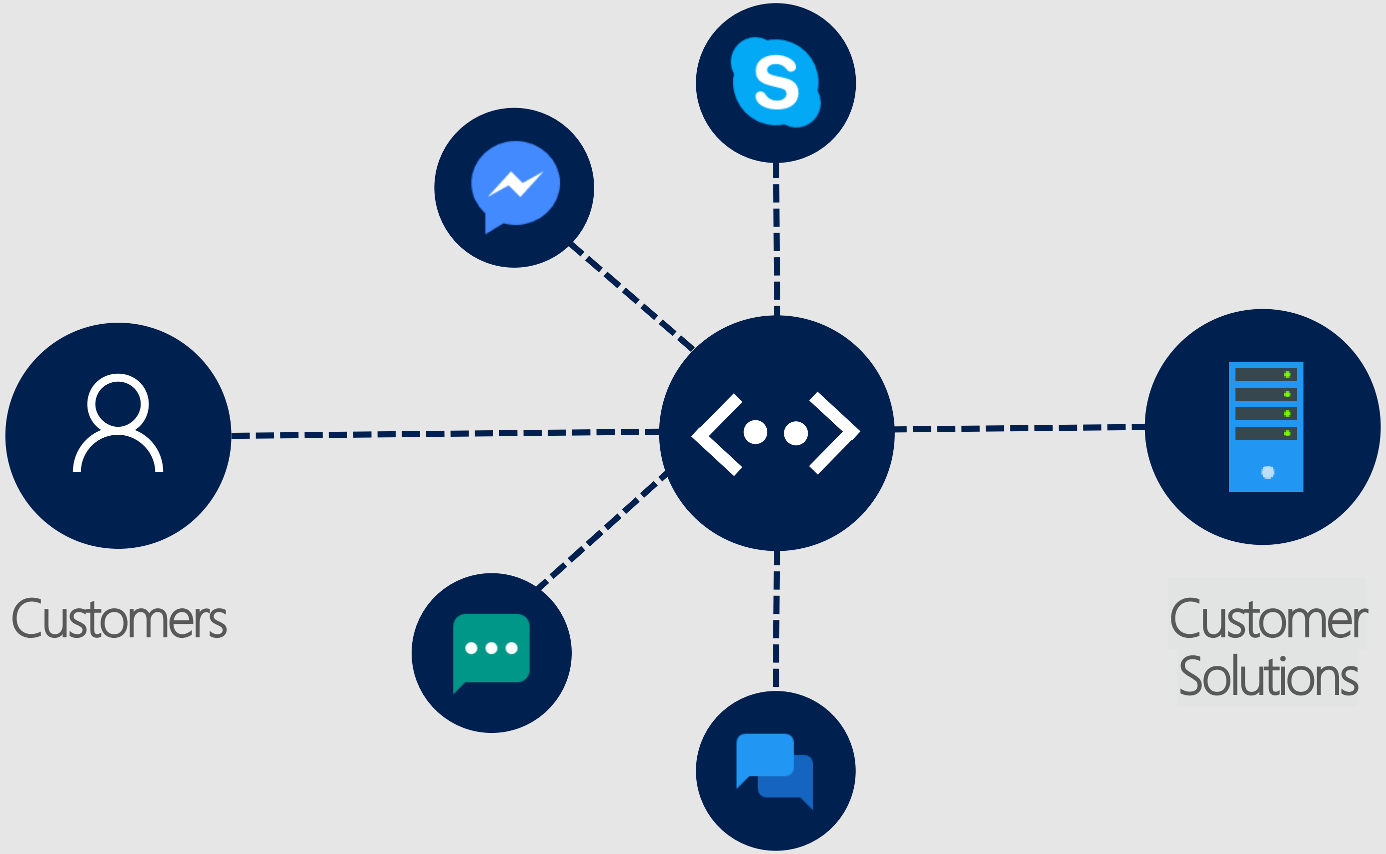
Other considerations

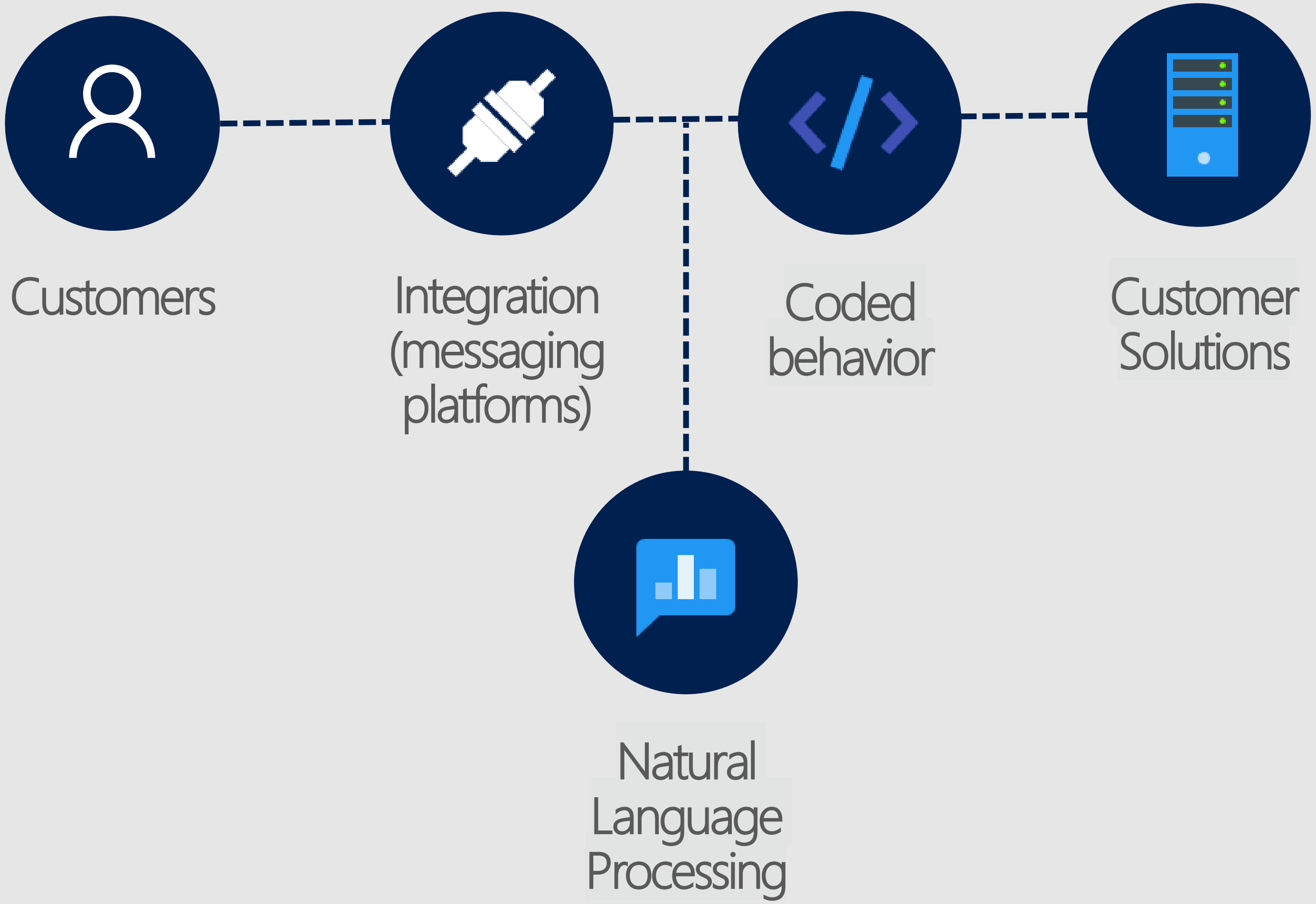
✓Start with customer problem

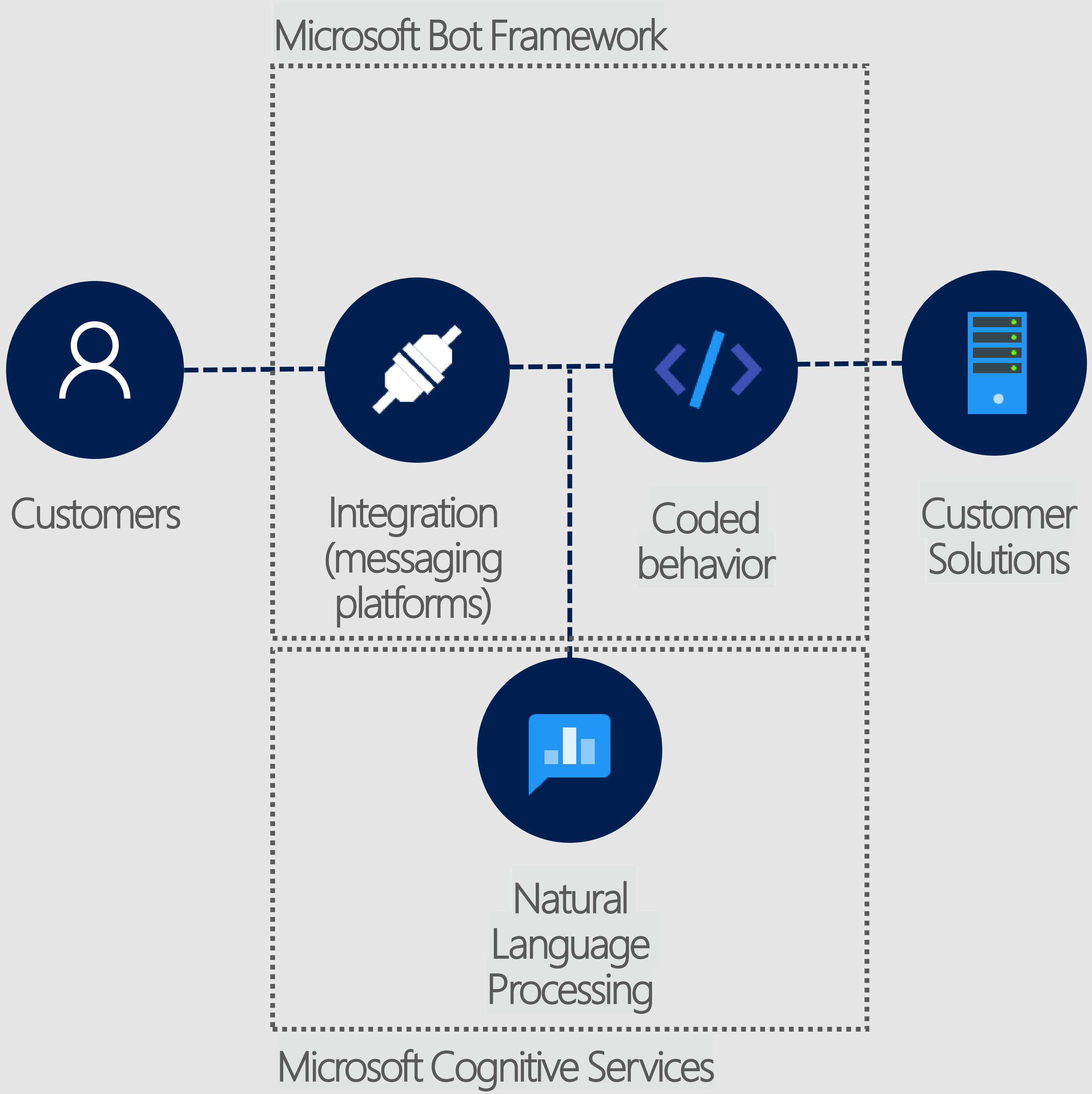
✓ A bot should be conversational. Who is writing the bot’s dialogue?

✓Bots talk. But it is backend that makes the difference.

Develop


Customers
Customer Solutions


Customers
Integration (messaging platforms)
Coded behavior
Customer Solutions
Natural Language Processing


Microsoft Bot Framework
Customers
Integration (messaging platforms)
Coded behavior
Customer Solutions
Natural Language Processing
Microsoft Cognitive Services

Bot Framework Components

Your Bot

Developer Portal

Connect your bot(s) to text/sms, Office 365 mail, Skype, Slack, and other services.

- Register, connect, publish and manage your bot through the bot dashboard
- Automatic card normalization across channels
- Skype channel auto-configured
- Embeddable web chat control
- Host your bot in your app using DirectLine API
- Fast, secure message routing
- Diagnostic tools

Bot Builder SDKs

Build great dialogs within your Node.js- or C#-based bot

- Open source SDK on Github
- 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 emulator

Bot Directory

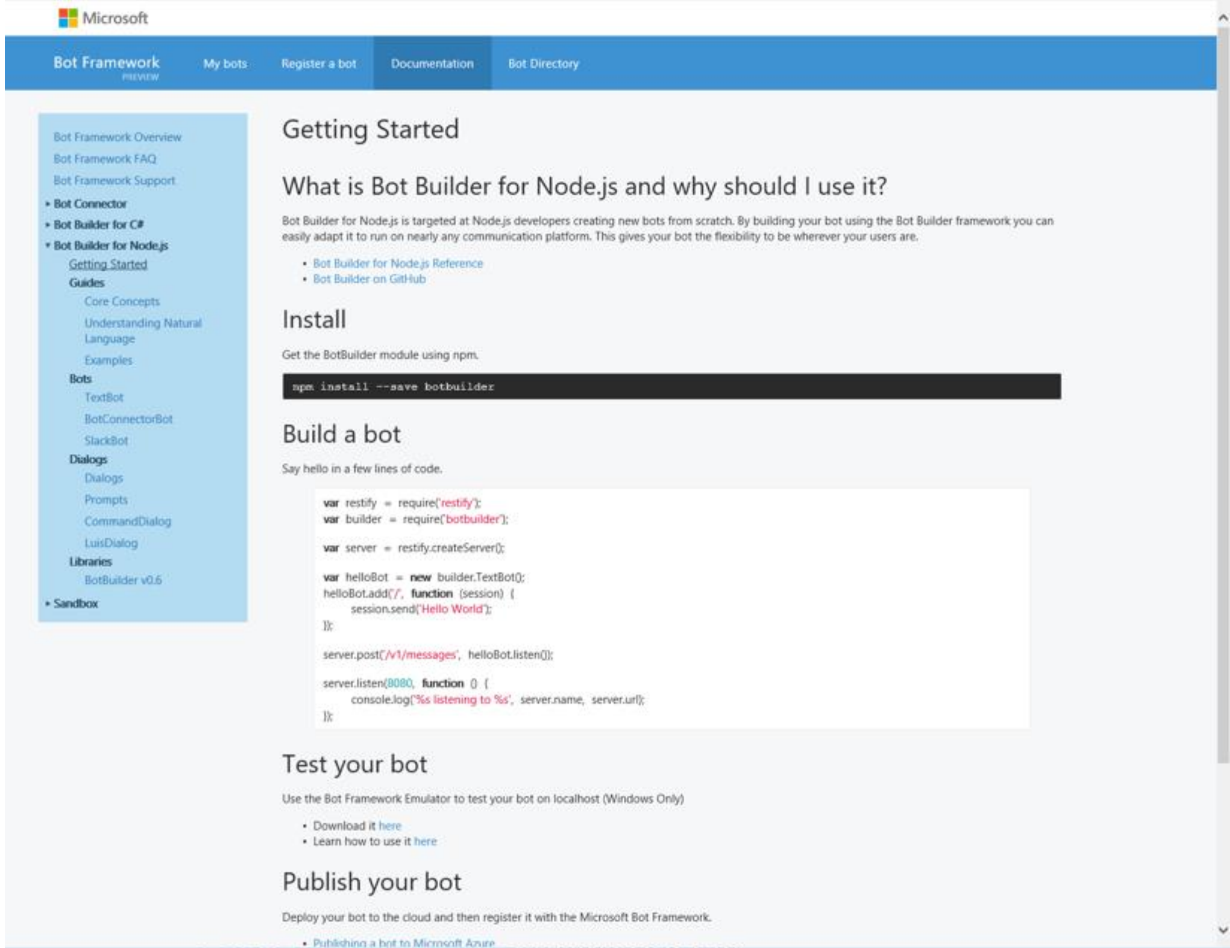
Try, use, and add published bots to the world's top conversation experiences

- Public directory of bots registered and approved with Bot Framework
- Users can try your bot from the directory via the web chat control
- Users can discover and add your bot to the channels on which it is configured

Authoring the code

C# and Node.js

- Dialogs to model a conversation
  - Dialogs are reusable
  - Conversations are scalable to multiple machines
  - Dialog state is persisted in Bot Connector (per-user, per-session, per- user+session)
- Types of Dialogs:
  - Built-in prompts
    - Yes/No, String, Number, Choices
  - Form slot filling (branching, disambiguation, multi-turn)
  - Profile (e.g., home address)



Getting Started

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

Bot Builder for Node.js is targeted at Node.js developers creating new bots from scratch. By building your bot using the Bot Builder framework you can easily adapt it to run on nearly any communication platform. This gives your bot the flexibility to be wherever your users are.

- Bot Builder for Node.js Reference
- Bot Builder on GitHub

Install

Get the BotBuilder module using npm.

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

Build a bot

Say hello in a few lines of code.

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

var server = restify.createServer();

var helloBot = new builder.TextBot();
helloBot.add('/', function (session) {
    session.send('Hello World');
});

server.post('/v1/messages', helloBot.listen());

server.listen(8080, function () {
    console.log('%s listening to %s', server.name, server.url);
});
```

Test your bot

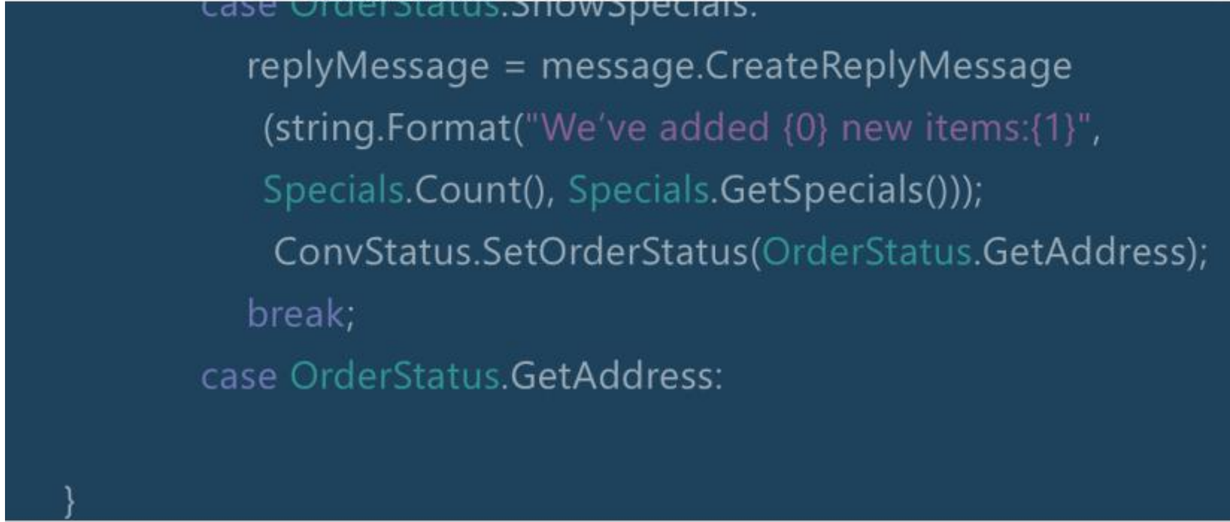
Use the Bot Framework Emulator to test your bot on localhost (Windows Only)

- Download it here
- Learn how to use it here

Publish your bot

Deploy your bot to the cloud and then register it with the Microsoft Bot Framework.

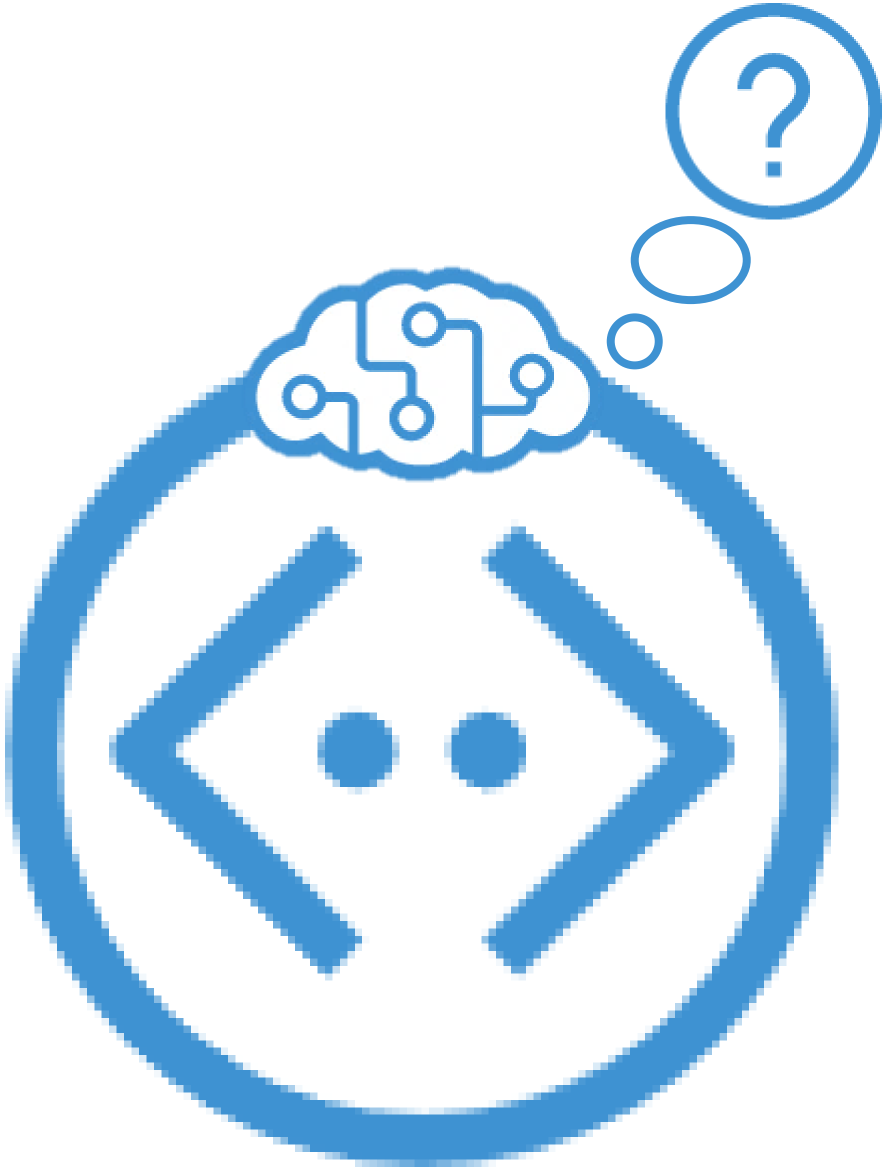
- Publishing a bot to Microsoft Azure



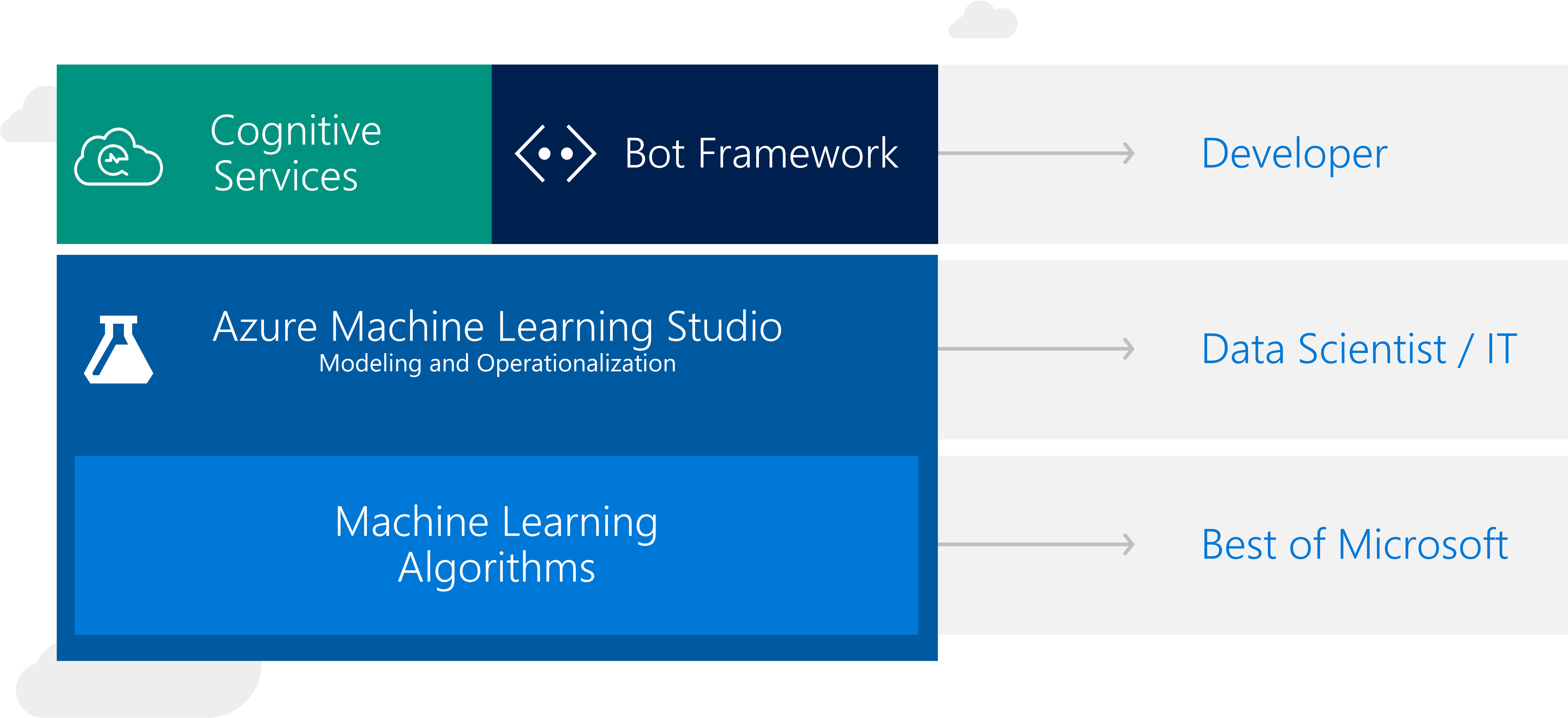
```
case OrderStatus.ShowSpecials:
    replyMessage = message.CreateReplyMessage
     (string.Format("We've added {0} new items:{1}",
     Specials.Count(), Specials.GetSpecials()));
      ConvStatus.SetOrderStatus(OrderStatus.GetAddress);
    break;
case OrderStatus.GetAddress:


}
```

Demo

My first (and second) bot



Machine Learning services in the cloud



Cognitive Services

Bot Framework

→ Developer

Azure Machine Learning Studio

Modeling and Operationalization

→ Data Scientist / IT

Machine Learning Algorithms

→ Best of Microsoft

Microsoft Cognitive Services

Give your apps a human side

Vision

From faces to feelings, allow your apps to understand images and video

Speech

Hear and speak to your users by filtering noise, identifying speakers, and understanding intent

Language

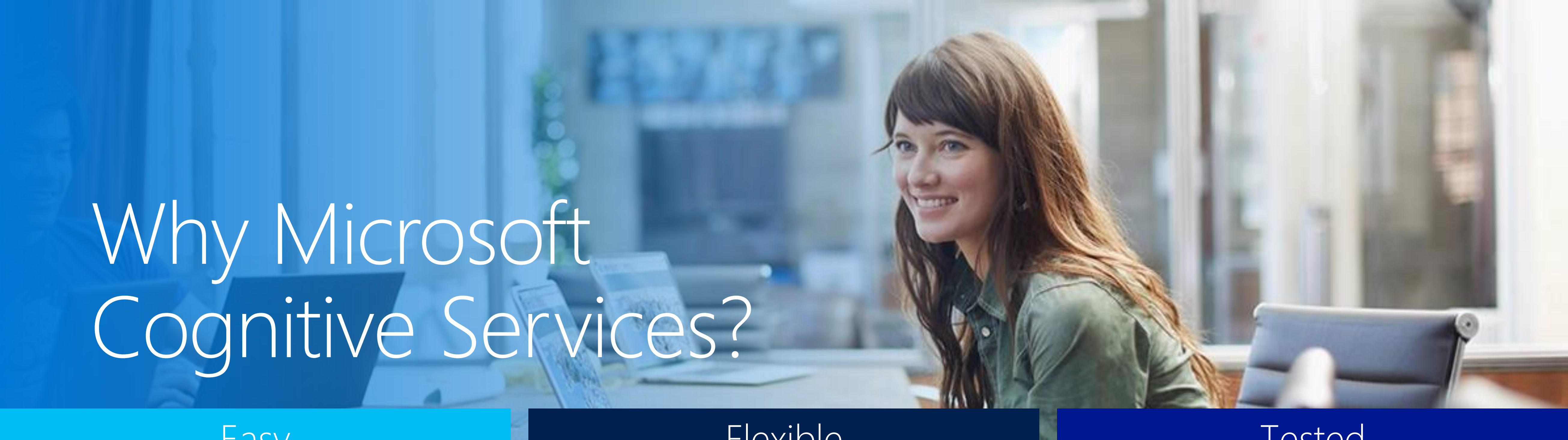
Process text and learn how to recognize what users want

Knowledge

Tap into rich knowledge amassed from the web, academia, or your own data

Search

Access billions of web pages, images, videos, and news with the power of Bing APIs



Why Microsoft Cognitive Services?

Easy

Roll your own with REST APIs

Simple to add: just a few lines of code required

GET A KEY

BUILD

Flexible

Integrate into the language and platform of your choice

Breadth of offerings helps you find the right API for your app

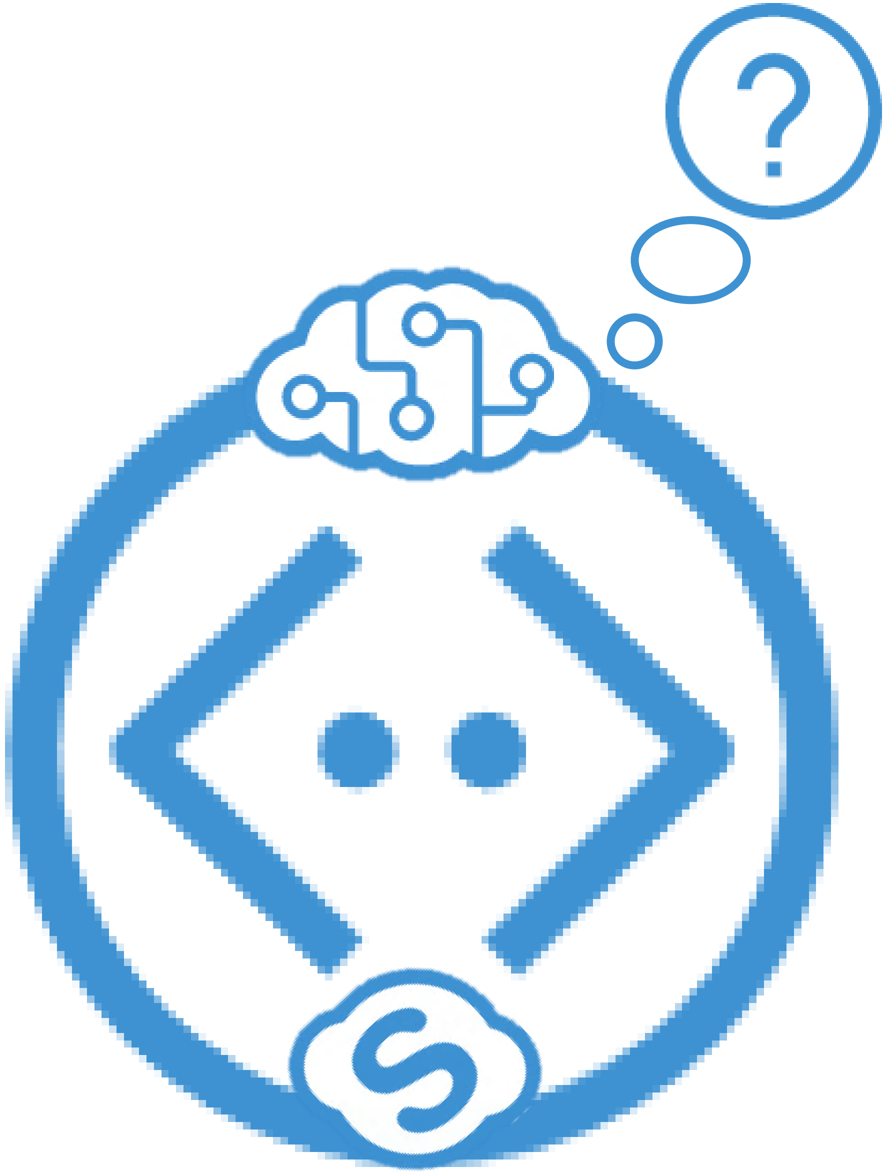
Tested

Built by experts in their field from Microsoft Research, Bing, and Azure Machine Learning

Quality documentation, sample code, and community support

Demo

Cognitive services



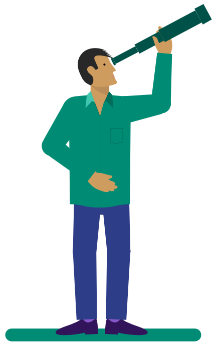
Actively Learning with LUIS

LUIS – Language Understanding Intelligent Service

Enables natural language understanding

Augments more complex dialog flows

Balance between guiding and open ended



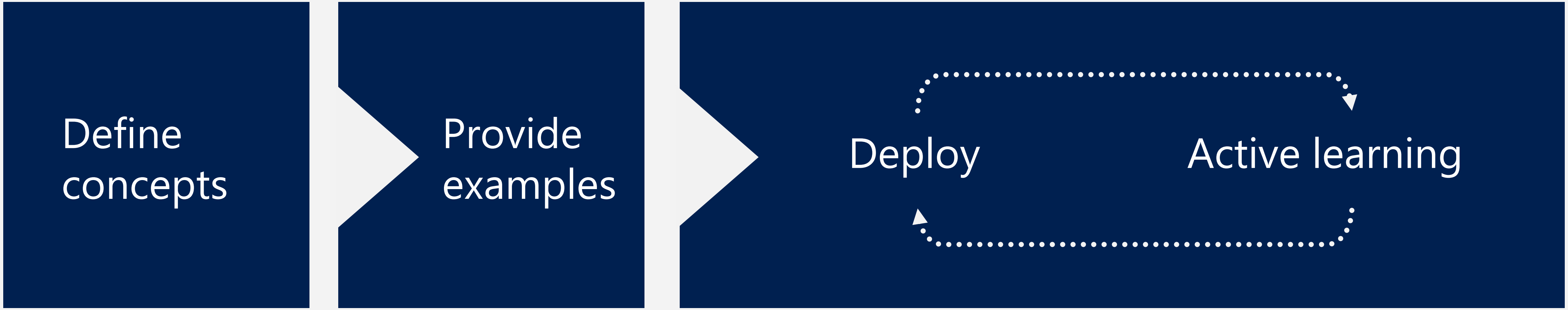
Language understanding intelligent service

Reduce labeling effort with interactive featuring

Use visualizations to gauge performance and improvements

Leverage speech recognition with seamless integration

Deploy using just a few examples with active learning



Language understanding (LUIS)

Define entities and intents

Entities–DepartureCity, ArrivalCity, DepartureDate, ReturnDate

Intent–book a flight

Map some utterances to an intent

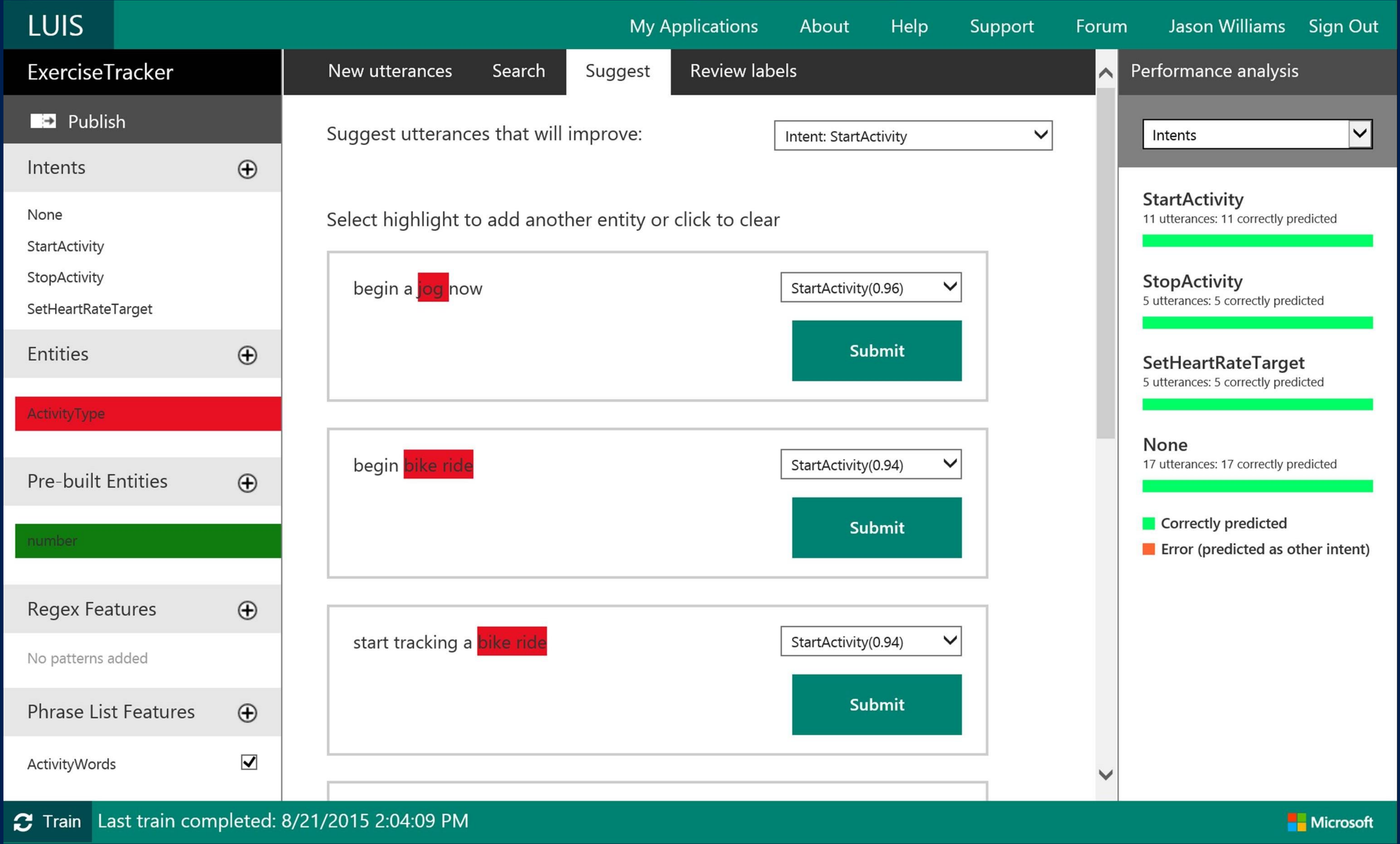
Examples: "I want to go to Paris from Sept 25 to Sept 29, 2016", "Book me a flight from DTW to CDG leaving on 9/25/2016 and returning 9/28/2016", etc.

Help your model improve over time based on real feedback

See what real users are sending to your model, and map those utterances to intents (or create new intents based on what your users are asking).

Language understanding models

```
{
  “entities”: [
    {
      “entity”: “flight_delays”,
      “type”: “Topic”
    }
  ],
  “intents”: [
    {
      “intent”: “FindNews”,
      “score”: 0.99853384
    },
    {
      “intent”: “None”,
      “score”: 0.07289317
    },
    {
      “intent”: “ReadNews”,
      “score”: 0.0167122427
    },
    {
      “intent”: “ShareNews”,
      “score”: 1.0919299E-06
    }
  ]
}
```



Suggest utterances that will improve: Intent: StartActivity

Select highlight to add another entity or click to clear

begin a jog now — StartActivity(0.96)

begin bike ride — StartActivity(0.94)

start tracking a bike ride — StartActivity(0.94)

Intents

StartActivity
11 utterances: 11 correctly predicted

StopActivity
5 utterances: 5 correctly predicted

SetHeartRateTarget
5 utterances: 5 correctly predicted

None
17 utterances: 17 correctly predicted

🟩 Correctly predicted
🟧 Error (predicted as other intent)

Last train completed: 8/21/2015 2:04:09 PM

Demo

LUIS

Deploy

Hosting Your Bot

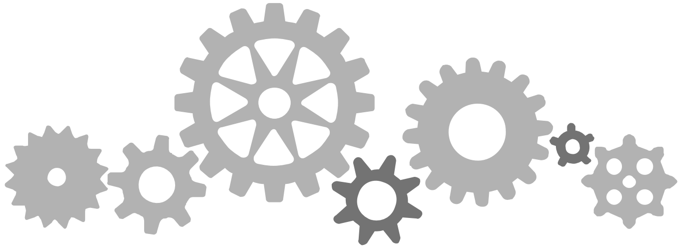
All you need is compute!

You may also use your own hosting

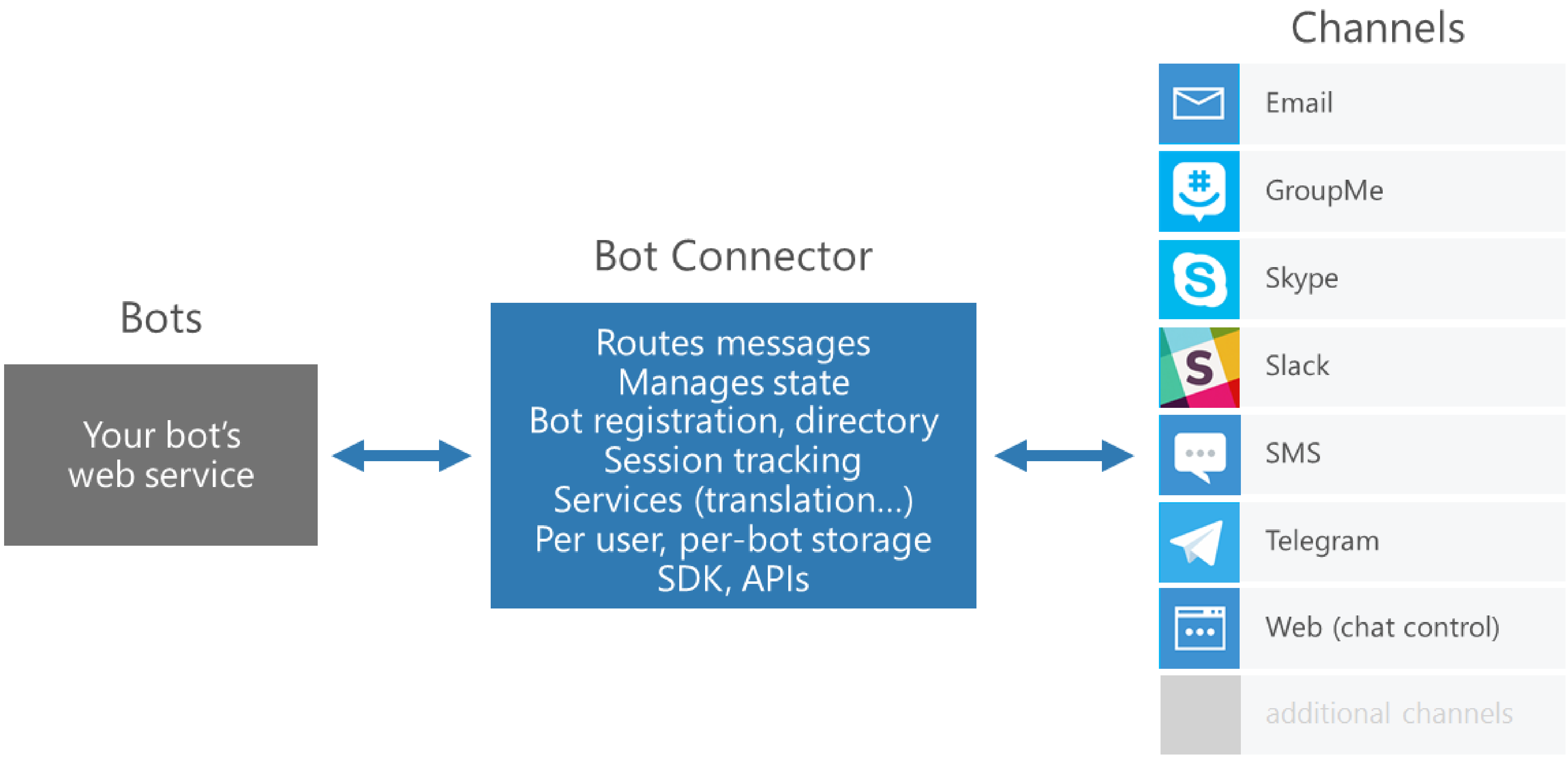
- (needs internet routable URL)

Azure fits as well ☺:

- Azure App Services
- Azure Virtual Machines
- Azure Functions



Bot Connector Basics



Bots

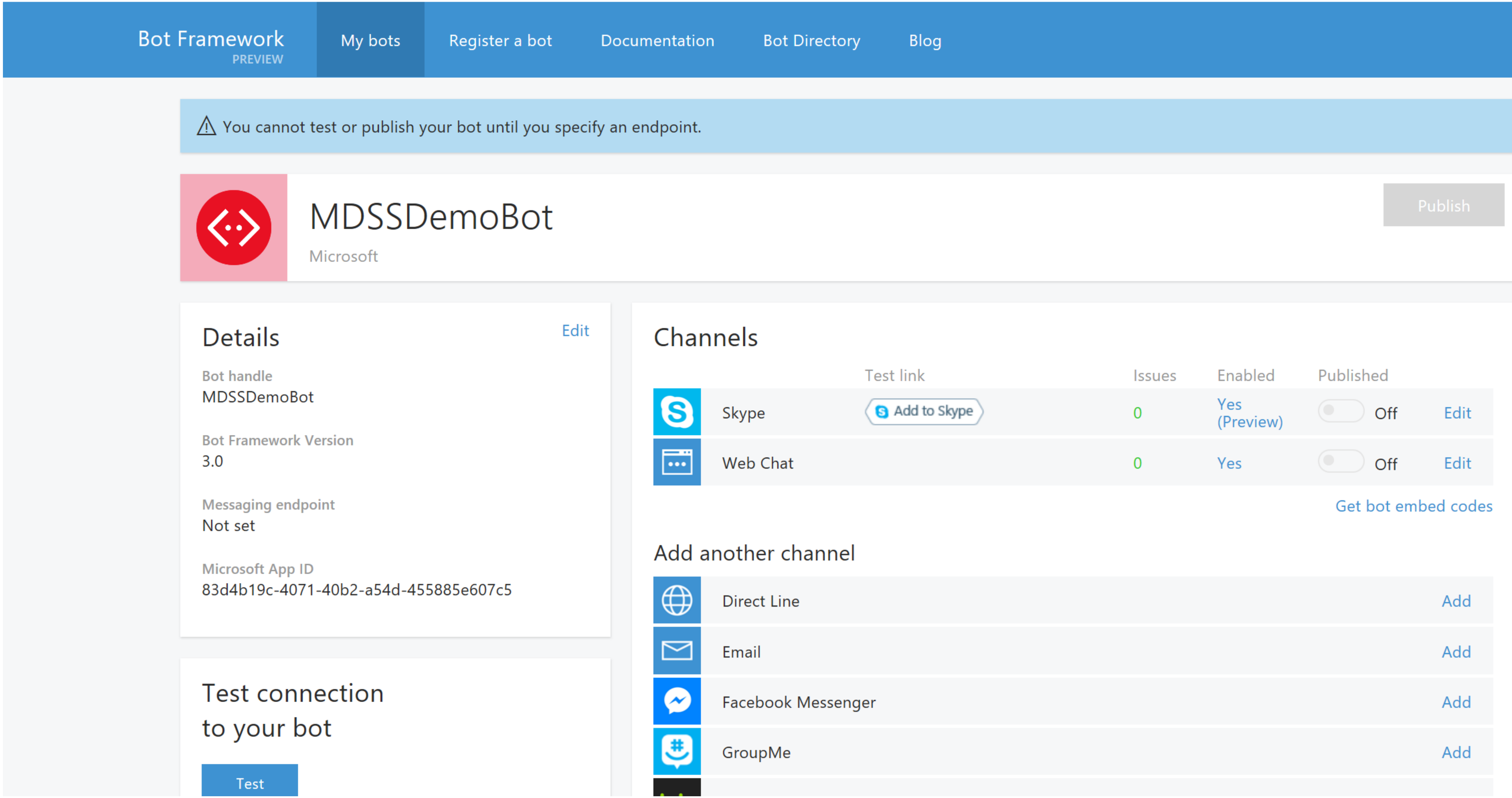
Your bot’s web service

Bot Connector

Routes messages
Manages state
Bot registration, directory
Session tracking
Services (translation…)
Per user, per-bot storage
SDK, APIs

Channels

- Email
- GroupMe
- Skype
- Slack
- SMS
- Telegram
- Web (chat control)
- additional channels



⚠ You cannot test or publish your bot until you specify an endpoint.

MDSSDemoBot

Microsoft

Details

Bot handle
MDSSDemoBot

Bot Framework Version
3.0

Messaging endpoint
Not set

Microsoft App ID
83d4b19c-4071-40b2-a54d-455885e607c5

Test connection to your bot

Channels

	Test link	Issues	Enabled	Published	
Skype	Add to Skype	0	Yes (Preview)	Off	Edit
Web Chat		0	Yes	Off	Edit

Add another channel

Direct Line	Add
Email	Add
Facebook Messenger	Add
GroupMe	Add

Test

It is not over yet…

Define your success criteria i.e.:

- increasing customer engagement
- approaching new customers
- learning and gathering feedback

Define your metrics i.e:

- Average time of conversation
- Conversation/closed case ratio

Measure!!!

Demo

Example bots

What's Next?

Find realistic scenario

Write your first bot!

Register your bot in the bot directory

Rinse and repeat

Bot Framework Resources

Bot Framework Home Page

https://dev.botframework.com/

Bot Builder SDK on GitHub

https://github.com/Microsoft/BotBuilder

Bot Framework Blog

https://blog.botframework.com/

LUIS

http://luis.ai

AzureBot

https://github.com/Microsoft/AzureBot

Microsoft