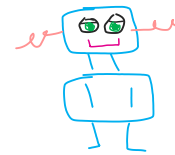


Today's Agenda

- Welcome and course links
 - Cognitive Services
- Bootcamp*
- Mission01 guided labs*
- Bot Framework
- Mission02 guided lab*
- Mission_Final HOL self-guided*

Instructor: Micheleen Harris
Data Scientist @ Microsoft

Welcome to Developing and Deploying Intelligent Chat Bots



<https://aka.ms/botedu>

Main site: aka.ms/botedu

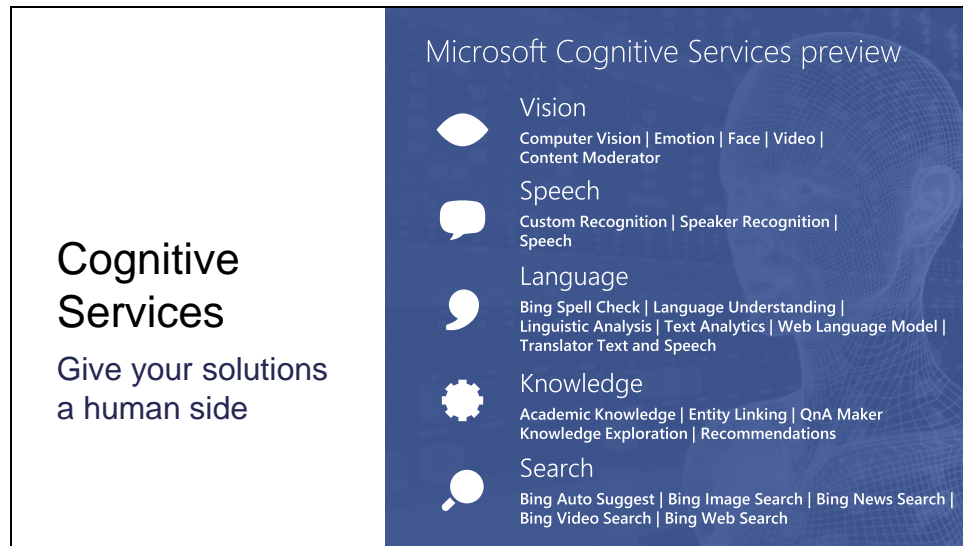
Chat room: aka.ms/botedu-discuss

This link contains additional resources on the bot framework and related topics. [mailto:
michhar@microsoft.com](mailto:michhar@microsoft.com) for questions/comments. [Show sites](#)

What you'll know at the end of this session

1. How to create a model with Cognitive Services Language Understanding and Intelligent Service (LUIS)
2. Be able to list the main options open to you with the Computer Vision API
3. Be able to demo the Bing News Search API and list the options for leveraging it

Some great scenarios for all of the cognitive services can be found on this wiki:
<https://github.com/Azure/bot-education/wiki/Cognitive-Services-Scenarios>



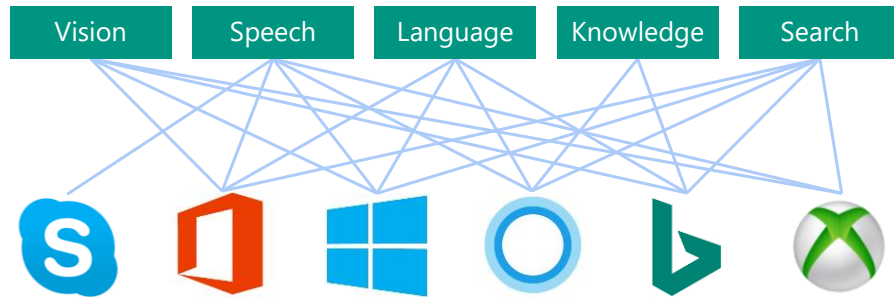
At Microsoft, we've been offering APIs for a very long time across the company. In delivering Microsoft Cognitive Services API, we started with 4 at /build (2015); added 7 more December 2015, and today we have 24 APIs in our collection.

Cognitive Services are available individually or as a part of the Cortana Intelligence Suite, formerly known as Cortana Analytics, which provides a comprehensive collection of services powered by cutting-edge research into machine learning, perception, analytics and social bots.

These APIs are powered by Microsoft Azure.

Developers and businesses can use this suite of services and tools to create apps that learn about our world and interact with people and customers in personalized, intelligent ways.

Cognitive Services power products across Microsoft



Language Understanding Intelligent Service (LUIS)

LUIS Concepts

Intent – aim or goal

Entities – a type or “notion” of person, place or thing

Utterances – the phrase we might use that is added training data

Utterances

If you have unlabeled utterances that your application should handle, they will be available when you edit the application under the "Search" and "Suggest" tabs.

Can link intents to actions and specify requirements for the action

Export LUIS app

Can download your work into a JSON file. This lets you share you application with other developers, or check your LUIS application into your version control.

Pre-Built:

LUIS also provides access to pre-built LUIS applications that use many of the same models found in Microsoft Cortana.

example:

intent – find news on topic and possibly share with another person

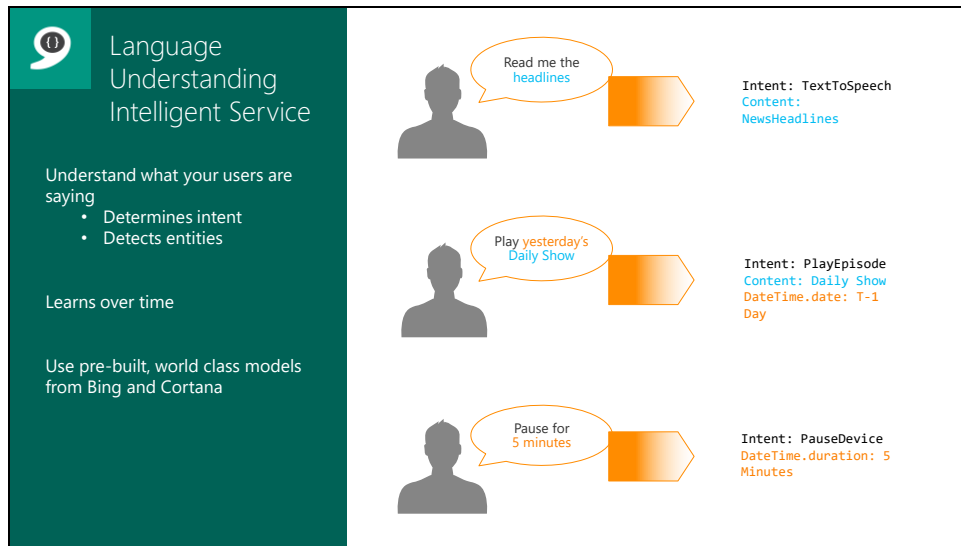
entities - We'd like to be able to say what kind of news we are interested in, and also, for sharing, to say who we'd like to share a story with. In order to capture the notion of a news topic, and a recipient for sharing, let's create two entity types: "Topic" and "Recipient".

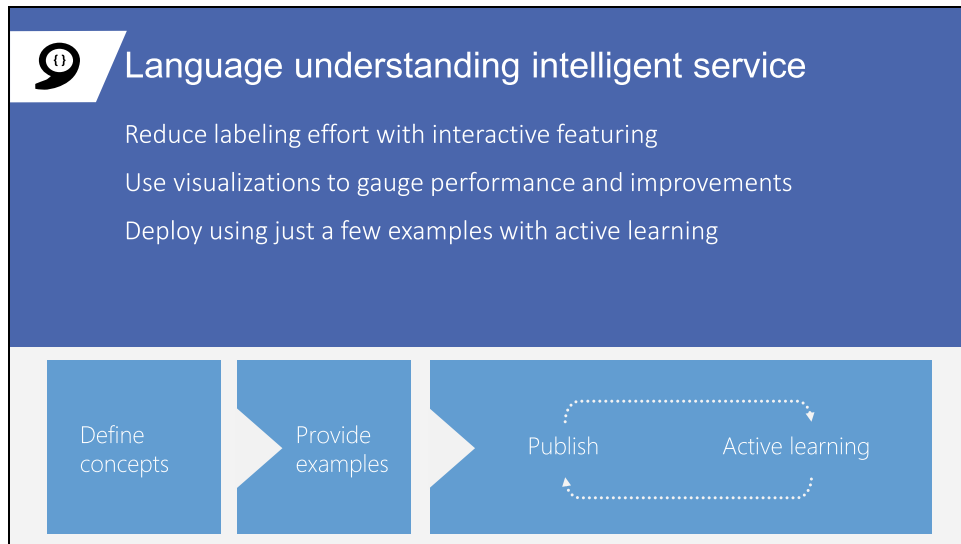
LUIS - Training

When you train a model, LUIS generalizes from the examples you have labeled (utterances), and develops code to recognize the relevant intents and entities in the next usage of this model

Internally, LUIS uses logistic regression classifiers to determine intents, and conditional random fields (CRFs) to determine the entities. The training process results in optimized classifiers and CRFs, referred to as models, that LUIS can use in the future.

Slide 9





LUIS SDKs

LUIS has SDKs in the following languages:

- Node.js
 - Python
 - C#
 - Java for Android
- Additionally, many samples of usage to be found on Cognitive Services site

SDKs

<https://github.com/Microsoft/Cognitive-LUIS-Node.js>

<https://github.com/Microsoft/Cognitive-LUIS-Python>

<https://github.com/Microsoft/Cognitive-LUIS-Windows>

<https://github.com/Microsoft/Cognitive-LUIS-Android>

SDK/Samples site:

<https://www.microsoft.com/cognitive-services/en-us/SDK-Sample>

Computer Vision API

CV API Definitions

Capabilities on images:

- **Analyze** – extraction of visual features (such as tags, colors, faces)
- **Describe** – natural language description
- **OCR** – word recognition
- **Domain specific** – celebrity recognition only currently
- **Thumbnail** – user desired size with regions of interest

<https://www.microsoft.com/cognitive-services/en-us/Computer-Vision-API/documentation> -> details in API Reference



Analyze image



Age: 27
Gender: Male

is Adult Content: False
Categories: people_swimming

Type of image

Clip Art Type

0 Non-clipart

Line Drawing Type

0 Non-Line Drawing

Black & White Image

False

Content of image

Categories

[[{"name": "people_swimming", "score": 0.899609375}]]

Adult Content

False

Adult Score

0.18533889949321747

Faces

[[{"age": 27, "gender": "Male", "faceRectangle": {"left": 472, "top": 258, "width": 199, "height": 199}}]]

Image colors

Dominant Color Background

White

Dominant Color Foreground

Grey

Dominant Colors

White

Accent Color

Illustrative CV Models for captions



Figure 1. An illustrative example of our pipeline.

Key notes:

- sub-regions are used in learned detector
- Training of language model captures commonsense knowledge
- High likelihood sentences are re-ranked by linear weighting

Fang H. *et al.* From Captions to Visual Concepts and Back. 2016

See paper here: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/CVPR15_0866.pdf

CV SDKs and Samples

- CV has SDKs for the following:
 - Swift
 - Android (Java)
 - Windows (C#)
- And samples:
 - Python (Jupyter notebook)
 - C# (image captioning bot)
 - Node.js (image captioning bot)
 - See4Me (Face, Emotion, CV – cross-platform app)

SDKs:

<https://github.com/DanilaVladi/Microsoft-Cognitive-Services-Swift-SDK>

<https://github.com/Microsoft/Cognitive-vision-android>

<https://github.com/Microsoft/Cognitive-vision-windows>

Notebook:

<https://github.com/Microsoft/Cognitive-vision-python>

Bots:

<https://github.com/Microsoft/BotBuilder-Samples/tree/master/Node/intelligence-ImageCaption>

<https://github.com/Microsoft/BotBuilder-Samples/tree/master/CSharp/intelligence-ImageCaption>

See4Me:

<https://github.com/DotNetToscana/See4Me>

Bing News Search API

News Search API Definitions

- **Search** – news articles from a given query
- **Category** – news articles of a given preset category (e.g. science and technology, world, etc.)
- **Trending topics** – trending news articles

News Search Demo

- Cognitive Services site demo
- Another demo, available on GitHub:
<https://github.com/Microsoft/Cognitive-Samples-IntelligentKiosk>

<https://github.com/Microsoft/Cognitive-Samples-IntelligentKiosk>

Developer Resources

Pricing

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

Documentation

<https://www.microsoft.com/cognitive-services> Click on Docs + Help

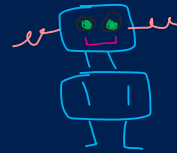
Client SDKs and Example Code

<https://www.microsoft.com/cognitive-services/en-us/SDK-Sample>

Community

<https://stackoverflow.com/questions/tagged/microsoft-cognitive>
<https://cognitive.uservoice.com/>

QnA



Here enters the Bot
Framework

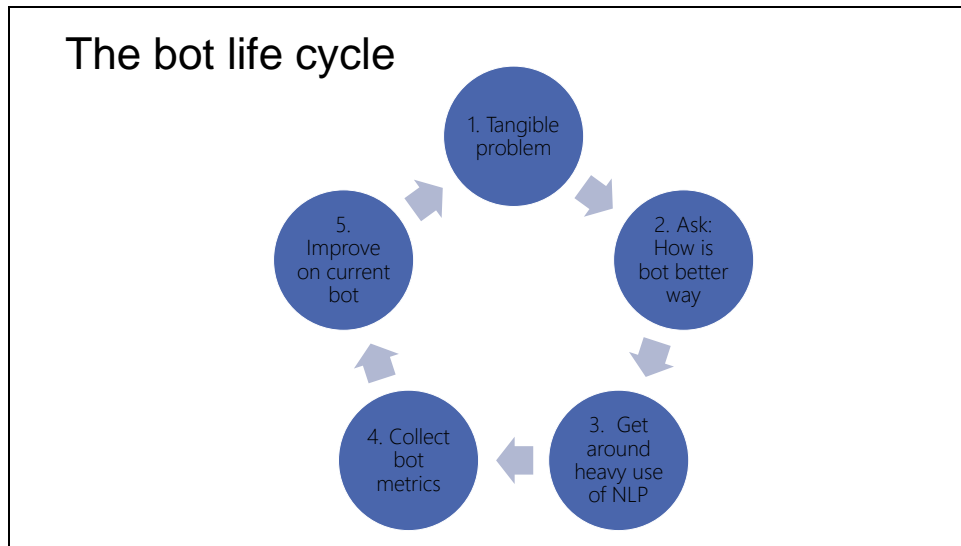


Conversation as a Platform

Human language is the new UI

Bots are the new apps;
digital assistants are meta apps

Intelligence infused into all interactions



- Start by asking what problem are we trying to solve. Refine until it looks like a tangible problem and not “magic”
- Ask how a bot will be a better experience. User experience is EVERYTHING
- Avoid too much natural language. Careful with unrealistic expectations. Natural language recognition is limited. Menus work great. Commands work great. Buttons, etc.
- You can only analyze and improve your bot if you’re collecting metrics for it
- Iterate, improve

Notes on creating an awesome
experience

Creating an awesome experience: Aims

Be valuable
Simple
Straightforward
Engaging
Welcoming
User is in control
Feedback
Integrity
Respect
Considerate
Avoid pushiness
Personality
Phone app?
Avoid typing

From: <https://docs.botframework.com/en-us/directory/best-practices>

This may have been in your mind before this tutorial. Ethical and societal considerations as well in an article by Satya Nadella: <https://www.linkedin.com/pulse/partnership-future-how-humans-ai-can-work-together-solve-nadella>

Creating an awesome experience: the AI

Key considerations around AI and design today:

Outcomes are determined as much by the human element as by the software element.

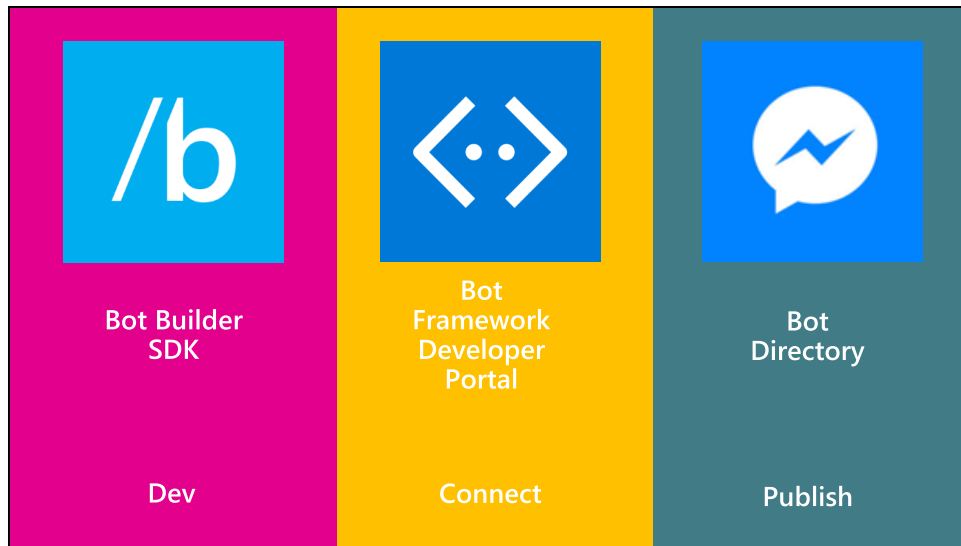
The quality of the user experience determines both the usefulness of the product and its rate of adoption, and this is why [it is believed] design is the next frontier of AI.

What does this mean?

Credit for paraphrased quotes Manoj Saxena: <https://www.fastcodesign.com/3068005/whats-still-missing-from-the-ai-revolution>

What is the Bot Framework

A development tool and much more...

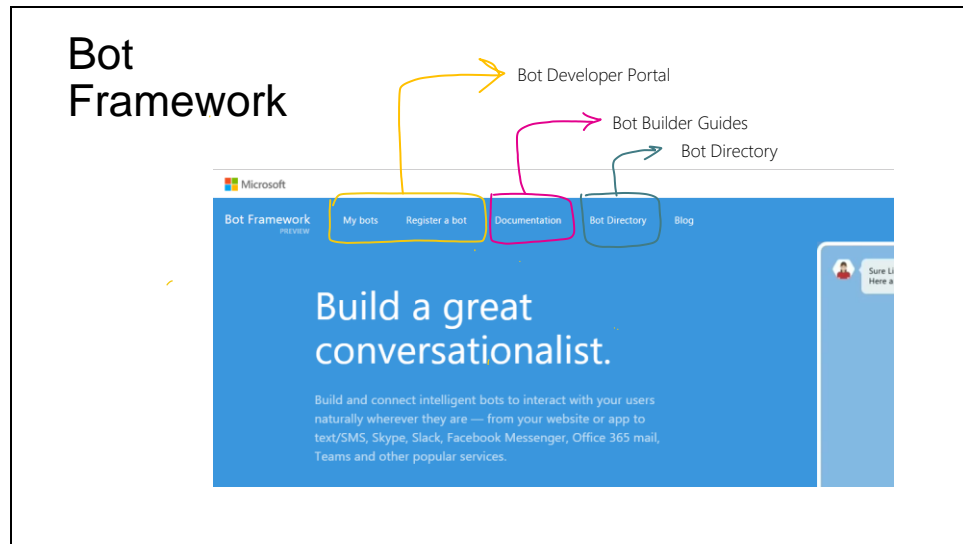


Bot Builder is itself a framework for building conversational applications (“Bots”).

The Bot Builder SDK is [an open source SDK hosted on GitHub](#) that provides everything you need to build great dialogs within your Node.js-, .NET- or REST API-based bot.

The Bot Framework Developer Portal lets you connect your bot(s) seamlessly text/sms to Skype, Slack, Facebook Messenger, Kik, Office 365 mail and other popular services. Register, configure and publish.

The Bot Directory is a public directory of all reviewed bots registered through the Developer Portal.



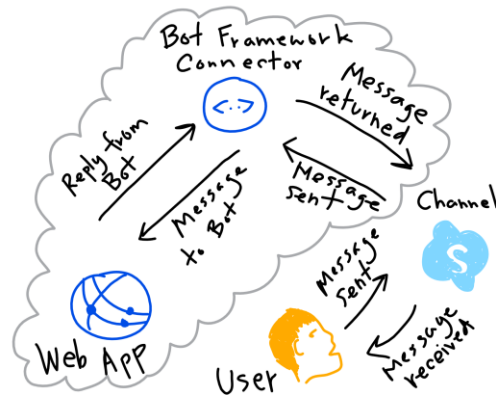
Main page: <https://botframework.com>

What is the Bot Framework

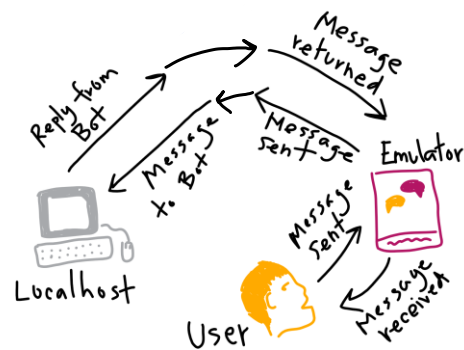
It's also a messaging platform

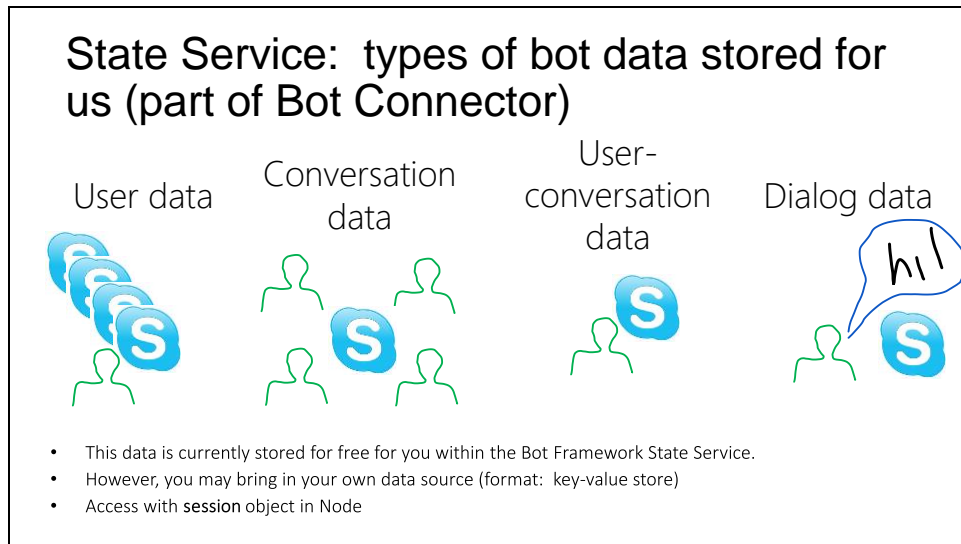
Let's get a picture of the way it works

What is the Bot Framework: as a cloud solution



What is the Bot Framework: as a development solution





User data – globally available for user across all conversations

conversation data – stores globally for a single conversation (many users could be involved)

User-conversation data – stores globally conversation data for a user (But private to just that user)

Dialog data as well – persists for a single dialog (helpful for temp data in a waterfall set of steps)

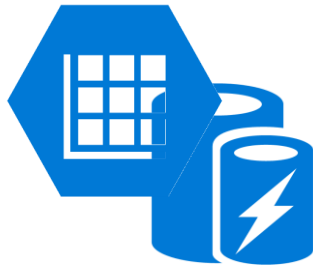
If I have a bot that plays Blackjack with me, my stats would be stored in user data (would follow me around from game to game), the deck information and stats in the conversation data (i.e. other players could use the same deck), and my hand in a game would be in user-conversation data (my immediate game's data).

The dialog data persistence ensures that the dialogs state is properly maintained between each turn of the conversation. Dialog data also ensures that a conversation can be picked up later and even on a different machine.

Anything can be stored in these data stores or bags, however it should be limited to data types that are serializable like primitives.

Storage behind the scenes

Bring your own:
Key-Value based storage



Bot Framework state service:
Azure Table Storage

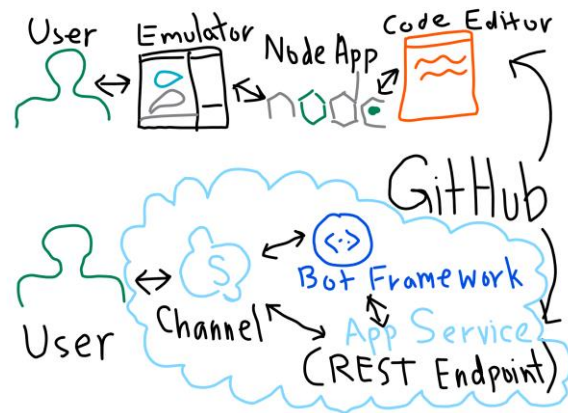


The State REST API has wrappers built around Azure Table Storage. NB, you can bring your own storage in the form of a key-value store like Redis Cache, Table Storage etc. The Bot Framework manages this default storage for you so you can maintain a stateless bot experience and if you bring your own, you'll need to maintain that store and make sure it scales.

Adding your own state example

<https://github.com/Microsoft/BotBuilder-Samples/tree/master/Node/core-CustomState>

Dev process



Q and A

Developing and Deploying Intelligent Chat Bots

