orange networks
IT-CONSULTING - IT-DEVELOPMENT

Chat Bot in a Day

Stefan Volkmer

orange networks
IT-CONSULTING - IT-DEVELOPMENT

Stefan Volkmer

Software Architekt und Engineer

Schwerpunkt Azure und AI

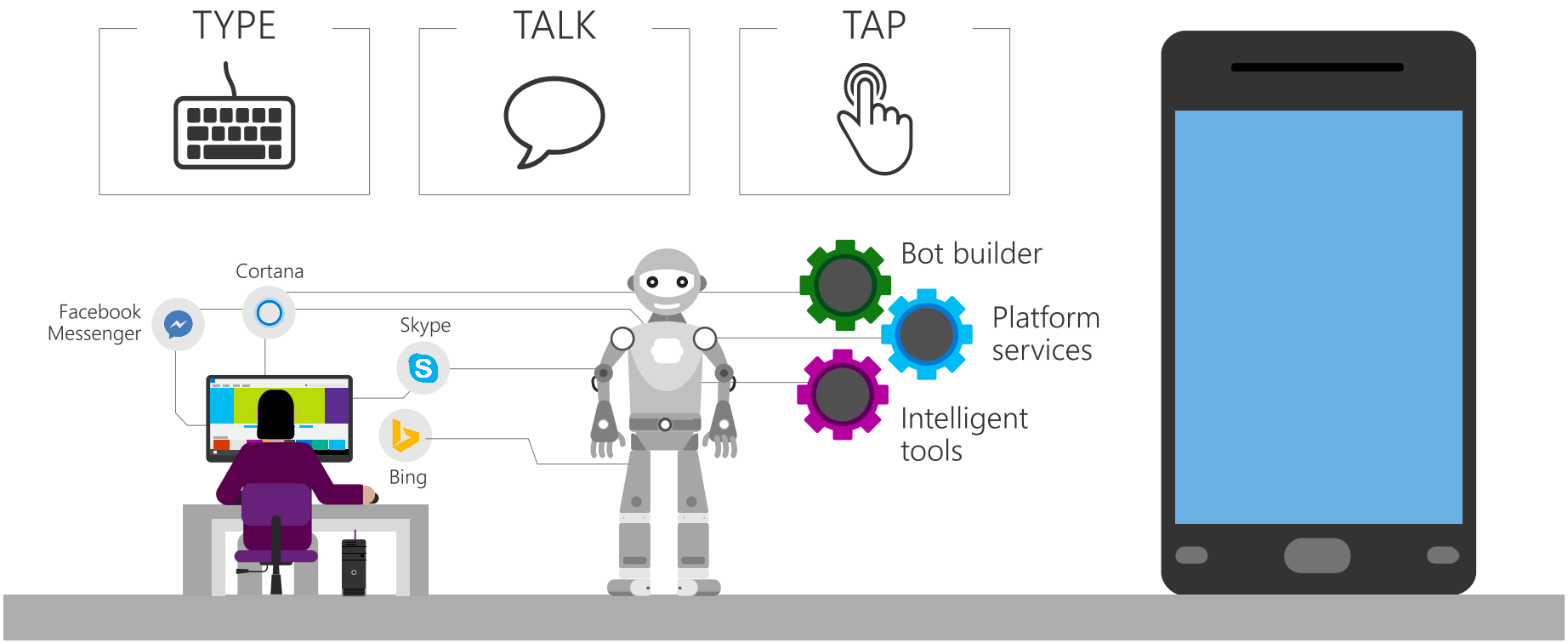
s.volkmer@orangenet.de

orange networks
IT-CONSULTING - IT-DEVELOPMENT

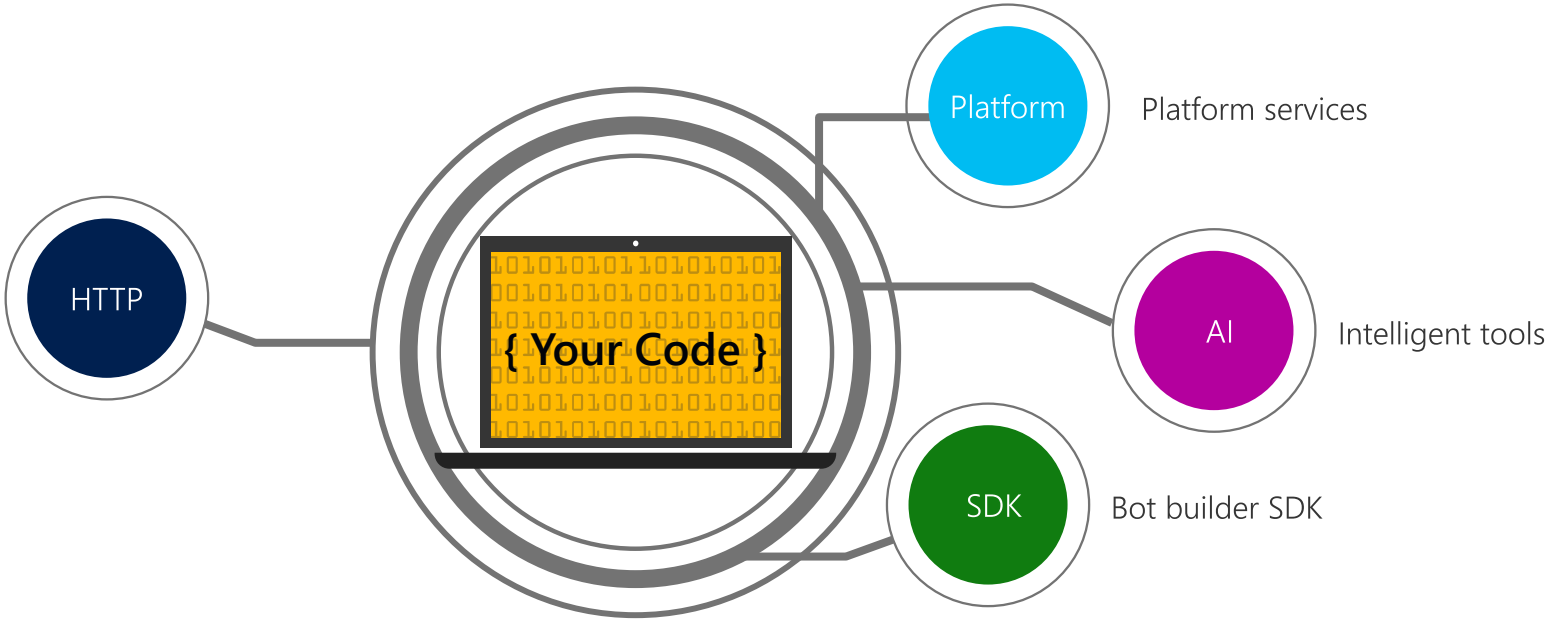
Agenda

- Organizational
- Bot Overview
- Installing the Tools
- Creating your first Bot
- Cognitive Services Overview
- Language Understanding Intelligent Service (LUIS.ai)
- QnA Maker
- Dialog Stack
- Adaptive Cards
- Bot Design Principles

Why a bot?


TYPE
TALK
TAP
Cortana
Facebook Messenger
Skype
Bing
Bot builder
Platform services
Intelligent tools

What is a bot?



REST endpoint
Direct Line Protocol

Conversational and business logic

Canvas aware

Context sensitive

- What Is a Bot?
  - A bot is an app that users interact with the user in a conversational way
  - Bots can communicate conversationally with text, cards, or speech.
  - A bot may be as simple as basic pattern matching with a response, or it may be a sophisticated weaving of artificial intelligence techniques with complex conversational state tracking and integration to existing business services.

- What is a Bot NOT?
  - AI
  - Just a chat (voice or text)

Kinds of bots

1,000+ companies engaging us

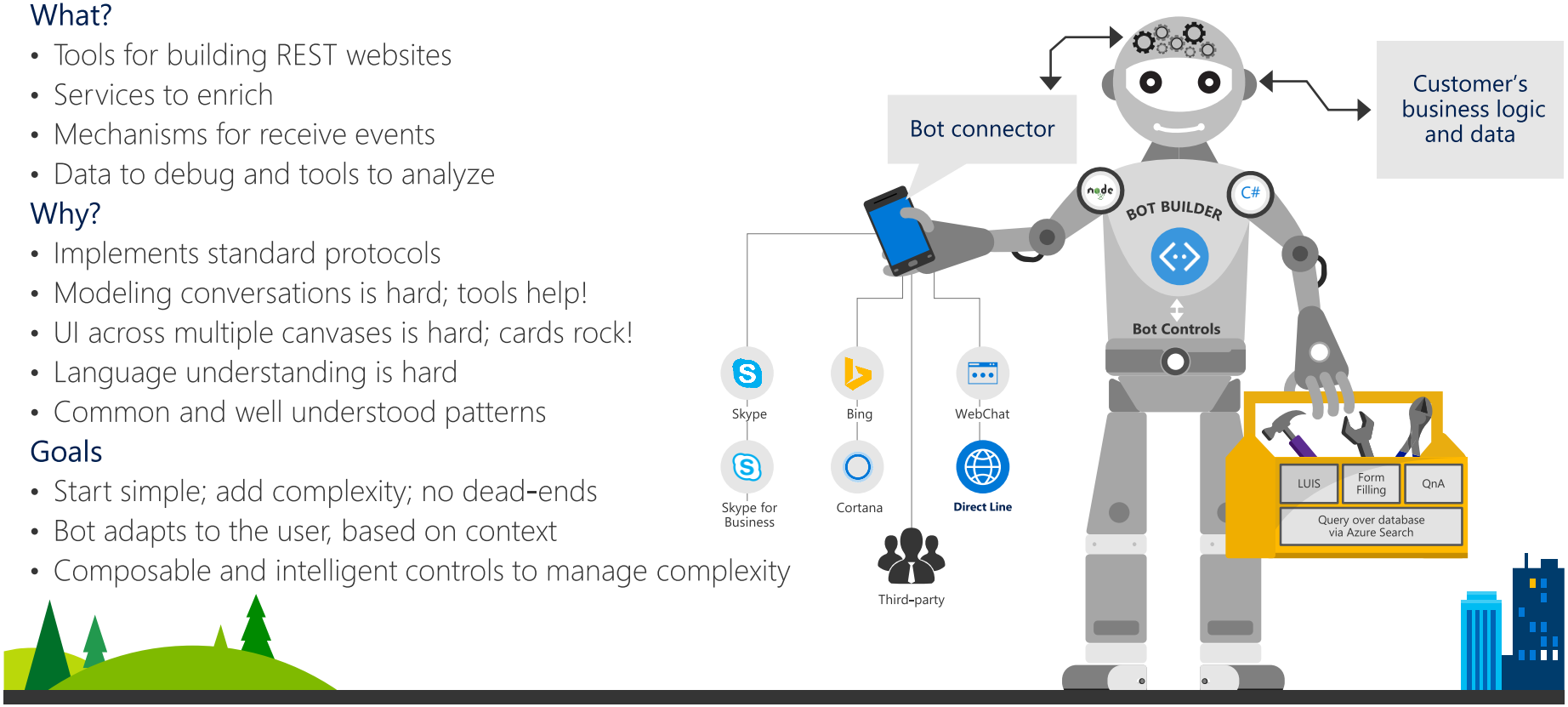
Scenario	Retail	Finance	Insurance	Telecoms	Government	Automotive	Manufacturing	Healthcare	Media	Events
Customer service	✔	✔	✔	✔	✔	✔	✔	✔		✔
Customer retail	✔	✔	✔	✔				✔		
Audio/speech analysis	✔	✔	✔	✔	✔				✔	
Translation		✔	✔							
Surveillance		✔			✔					
Knowledge extraction		✔	✔	✔			✔			
Video/photo analysis		✔			✔				✔	
Product identification	✔						✔	✔		
Digital assistant						✔				
Footfall analysis	✔									✔
HD maps and object detection						✔				

What is the bot framework?

What?
- Tools for building REST websites
- Services to enrich
- Mechanisms for receive events
- Data to debug and tools to analyze

Why?
- Implements standard protocols
- Modeling conversations is hard; tools help!
- UI across multiple canvases is hard; cards rock!
- Language understanding is hard
- Common and well understood patterns

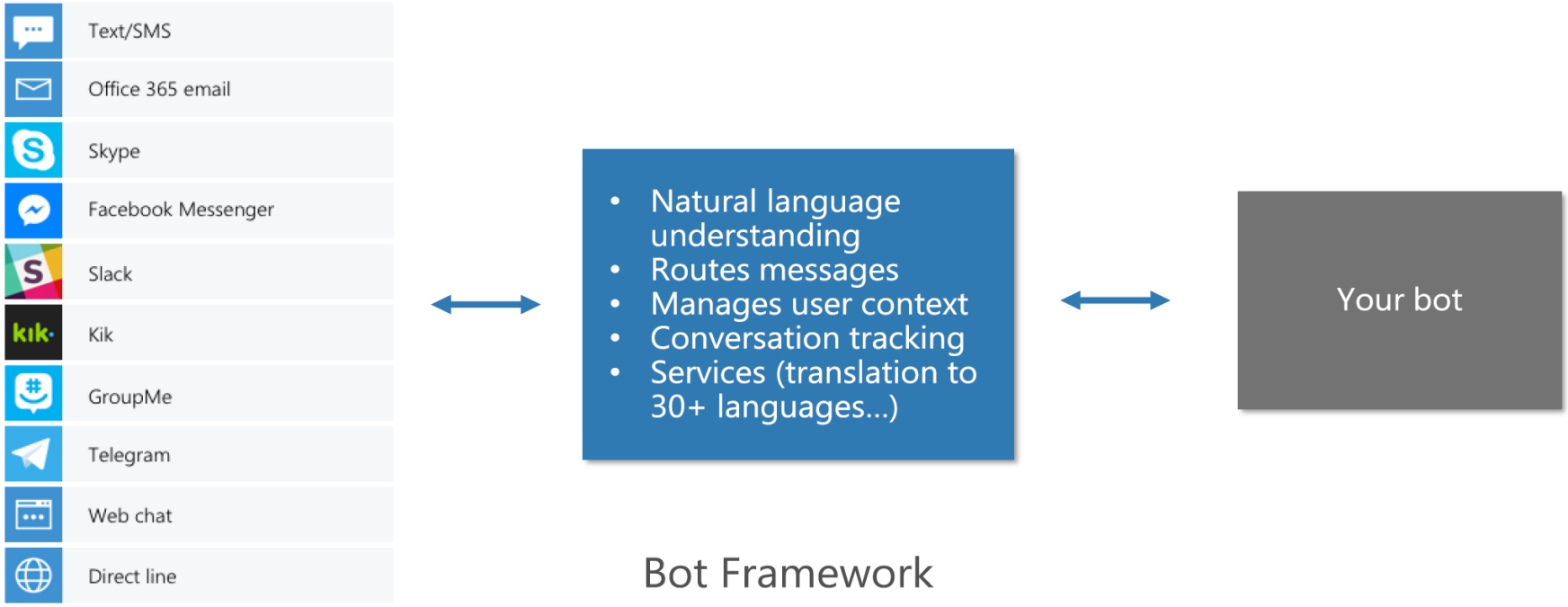
Goals
- Start simple; add complexity; no dead-ends
- Bot adapts to the user, based on context
- Composable and intelligent controls to manage complexity



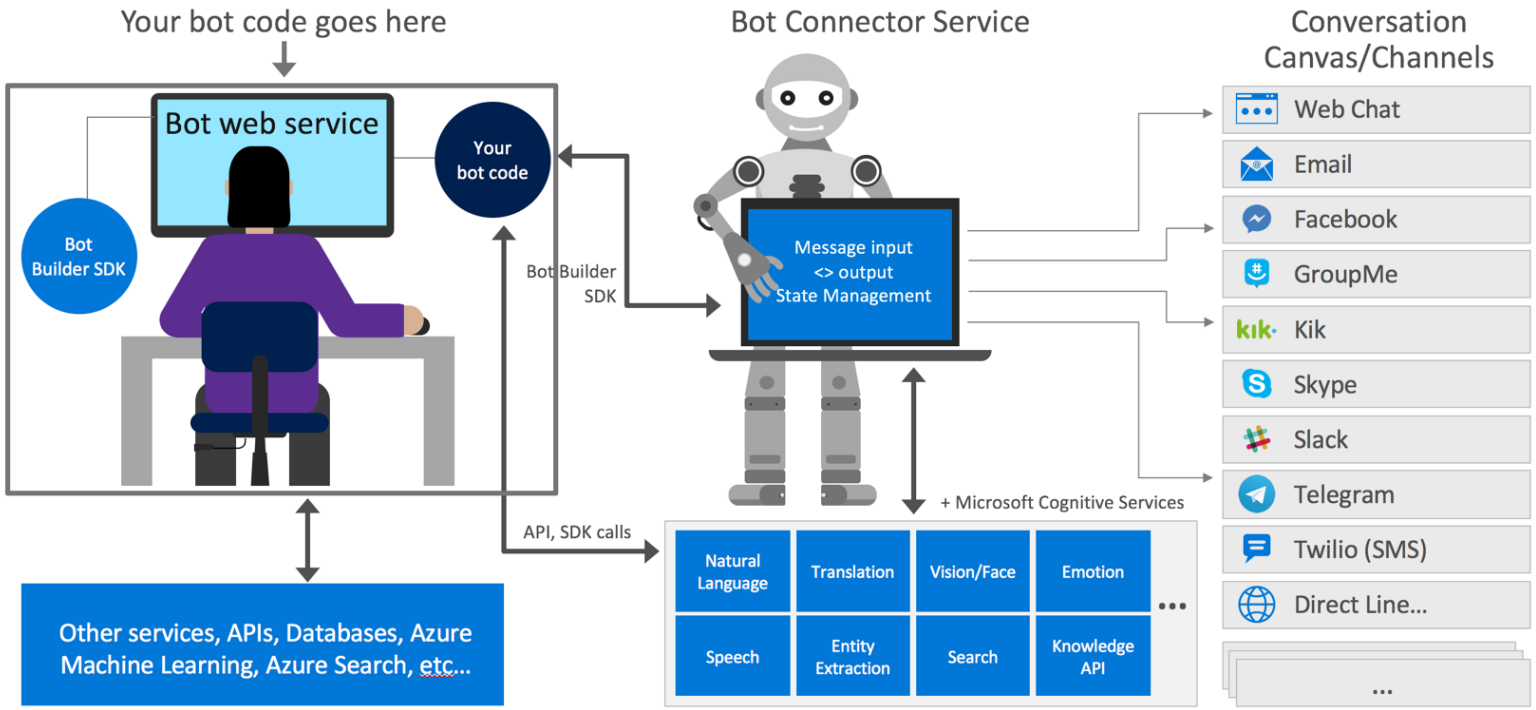
New channels

I M P R O V E M E N T S E V E R Y W H E R E !

Bot Framework Value Proposition



Building Bots


Your bot code goes here
Bot Connector Service
Conversation Canvas/Channels
Bot web service
Your bot code
Bot Builder SDK
Bot Builder SDK
Message input <> output State Management
Web Chat
Email
Facebook
GroupMe
Kik
Skype
Slack
Telegram
Twilio (SMS)
Direct Line...
...
+ Microsoft Cognitive Services
API, SDK calls
Natural Language
Translation
Vision/Face
Emotion
Speech
Entity Extraction
Search
Knowledge API
Other services, APIs, Databases, Azure Machine Learning, Azure Search, etc...

Bot Builder

- Tools and services to build great bots that converse wherever the users are
- Open source SDK on Github for Node.js, .NET and REST
- Support for rich attachments (image, card, video, doc, etc.)
- Support for calling (Skype)
- Online/offline chat Emulator

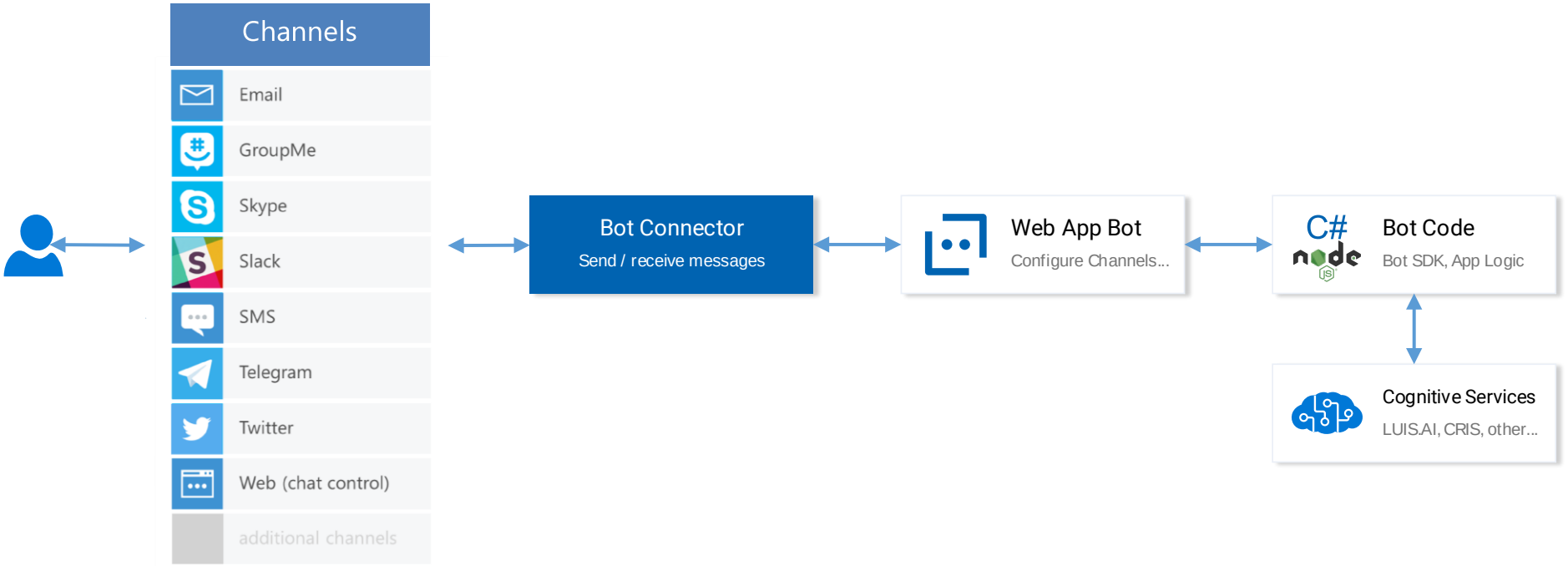
Overview

- Connector Service
  - The Bot Connector service enables the bot to exchange messages with channels that are configured in the Bot Framework Portal, by using industry-standard REST and JSON over HTTPS
- Activity
  - A specific event that occurs between a Bot and Users, such as an actual message, or conversation notification
- Messages
  - A Message is a communication between a Bot and Users, such as a User asking a question, or a Bot responding with a reply

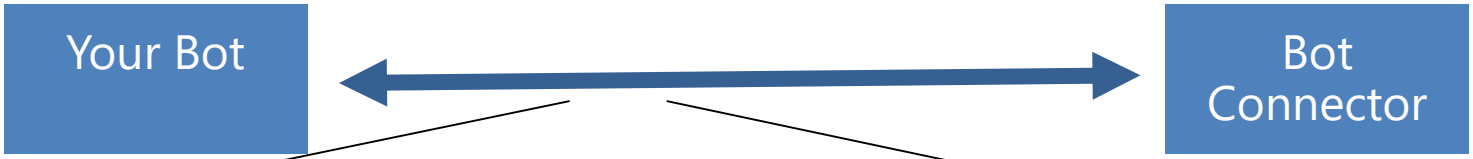
Types of Activities

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

Bot Connector



Bot Connector messages




```
{
  "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 },
  ...
}
```

Azure Bot Service

Accelerate development cycles	Enrich your bots	Boost operational efficiencies

Azure Bot Service

Accelerate development cycles

- Develop your way (node.JS, C#)
- Built in code editor
- Quick start templates
- Integrated chat window

Enrich your bots

- Channel support
- Cognitive Services
- Direct Line support (API)
- Embedded web chat

Boost operational efficiencies

- Powered by Azure Functions
- Continuous deployment
- Scale on demand (AVAILABILITY)
- Reduced dev ops

Installing the Tools

- NodeJs (https://nodejs.org/)
- Visual Studio Code als Code Editor
- Bot Emulator + ngrok (https://docs.microsoft.com/de-de/azure/bot-service/bot-service-debug-emulator?view=azure-bot-service-3.0)
- Github

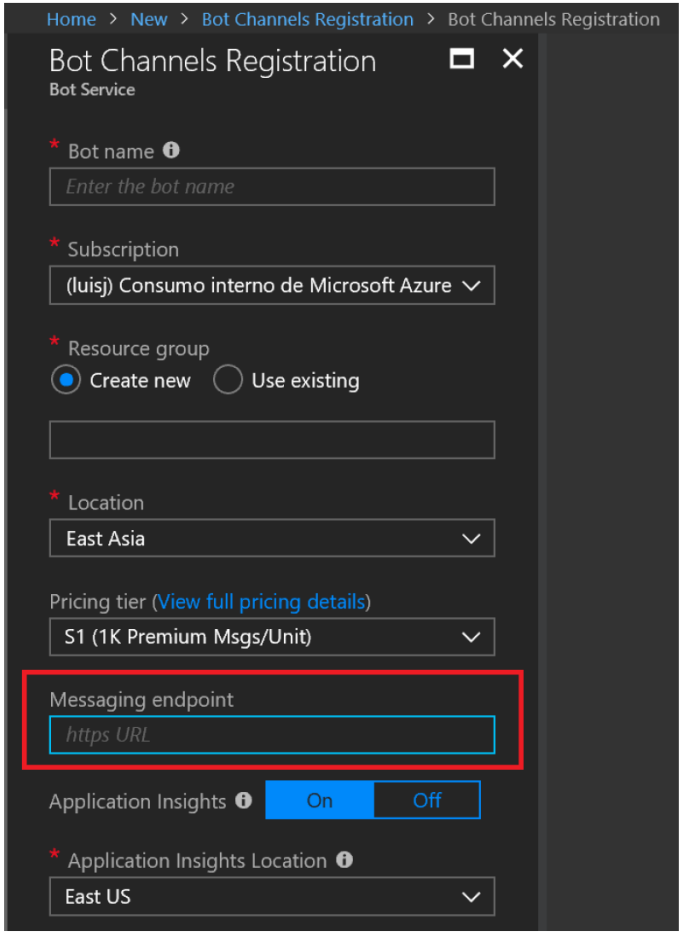
Demo

Creating a Bot with Azure Bot Service

Register a Bot

Register a Bot – Portal

- Enable alternative hosting
- Simple process
- Basic Information
- Endpoint
  - Can be changed later
- Microsoft App Id
- Analytics with Application Insights



Deploy to Azure

Deploy to Azure

- A Bot is a web application
- There are some options to build a Bot and deploy to Azure
- Bot Service can handle automatically CI/CD scenarios
  - The integration uses App Service Kudu Deployment Engine
- Can be directly from Visual Studio
  - Build and publish
- Or from Visual Studio Team Services (VSTS)
  - Configure Build & Release

Demo

How to deploy to Azure

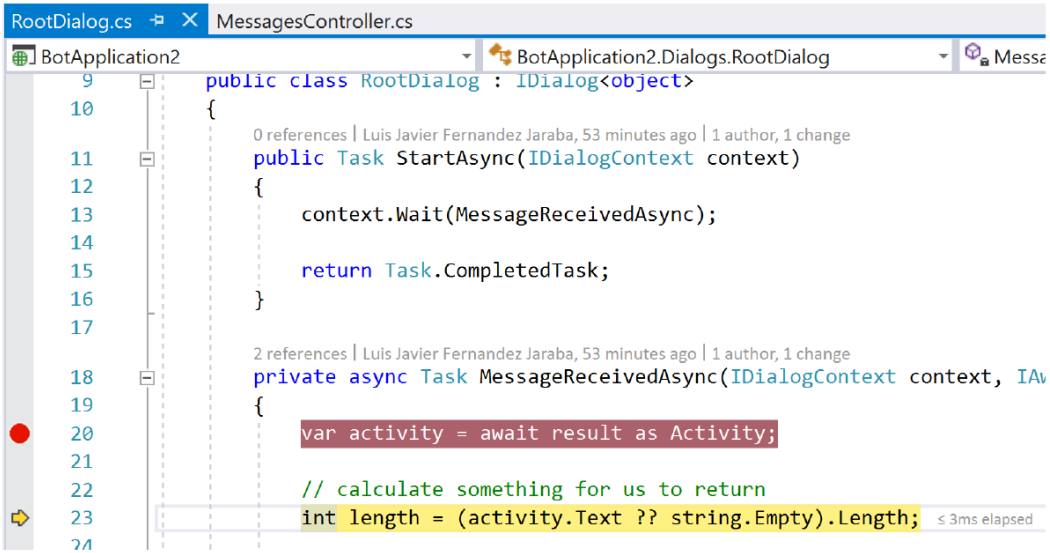
Test and Debug

Using the Bot Framework Emulator

- Is a desktop application that allows bot developers to test and debug their bots, either locally or remotely
  - If you are behind a firewall or other network boundary and want to connect to a bot that is hosted remotely, you must install and configure **ngrok** tunneling software
- Inspect the messages that your bot sends and receives
- You can test your bot using the emulator even if you have not yet registered it with the Bot Framework or configured it to run on any channels.

Debug a bot with Visual Studio

- Visual Studio has built-in debugging support
- IIS Express + Bot Emulator is an easy way to test and debug locally
- Additionally, you can debug Azure deployed bots



Demo

Debug bots with the Bot Framework Emulator

Lab

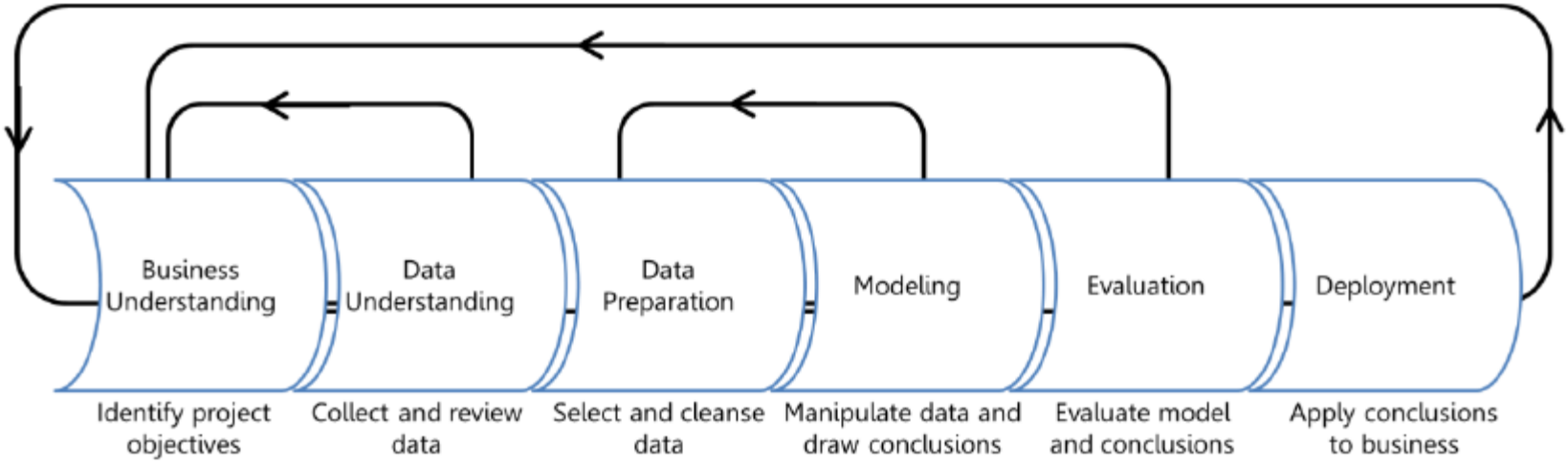
Lab 1 – Creating a simple Bot

orange networks
IT-CONSULTING - IT-DEVELOPMENT

Introducing Microsoft Cognitive Services

Stefan Volkmer

Machine Learning Steps



Why Microsoft Cognitive Services ?

Easy

Roll your own with REST APIs

Simple to add: just a few lines of code required

Flexible

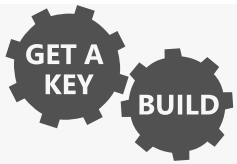
Make the same API code call on iOS, Android, and Windows

Integrate into the language and platform of your choice

Tested

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

Quality documentation, sample code, and community support



Microsoft Cognitive Services

Give your apps a human side

Cognitive Services

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

Demo

Cognitive Services

orange networks
IT-CONSULTING - IT-DEVELOPMENT

Language Understanding Intelligent Service (LUIS.ai)

Introduction & Overview

- Overview
- What is LUIS.AI?
  - Language Understanding Intelligent Service (LUIS) allows your application to understand what a person wants in their own words.
  - LUIS uses machine learning to allow developers to build applications that can receive user input in natural language and extract meaning from it.
  - A client application that converses with the user can pass user input to a LUIS app and receive relevant, detailed information back.

Why LUIS.AI

It is fast and easy

LUIS is designed to enable you to quickly deploy an HTTP endpoint that will take the sentences you send it and interpret them in terms of the intention they convey and the key entities that are present.

It learns and adapts

After your endpoint has processed a few dozen interactions, LUIS begins active learning. LUIS examines all the utterances that have been sent to it, and calls to your attention the ones that it would like you to label.

It offers pre-built domains

Prebuilt domains jump-start your application by providing intents and entities that can be mixed in and modified to create better language understanding with less effort. We include domains for things like Calendar, Music, Devices and many more.

It is a power developer tool

The over all experience of LUIS focuses on boosting developers' productivity though providing a set of powerful tools, offered through a simple user-experience & comprehensive set of APIs.

It is MultiLingual

LUIS is a multi-lingual platform and can be trained to understand over 10 different languages.

Key Concepts

01

Intents - Intents are like verbs in a sentence, it represents actions the user want to perform such as booking a flight, paying a bill

02

Utterance-Is the textual input from the user, that your app needs to interpret. It may be a sentence like "Book me a ticket to Paris"

03

Entities – Entities are nouns, it represents an instance of a class of object that is relevant to a user's intent. For example in the utterance "Book me a ticket to Paris", Paris is an entity of type location

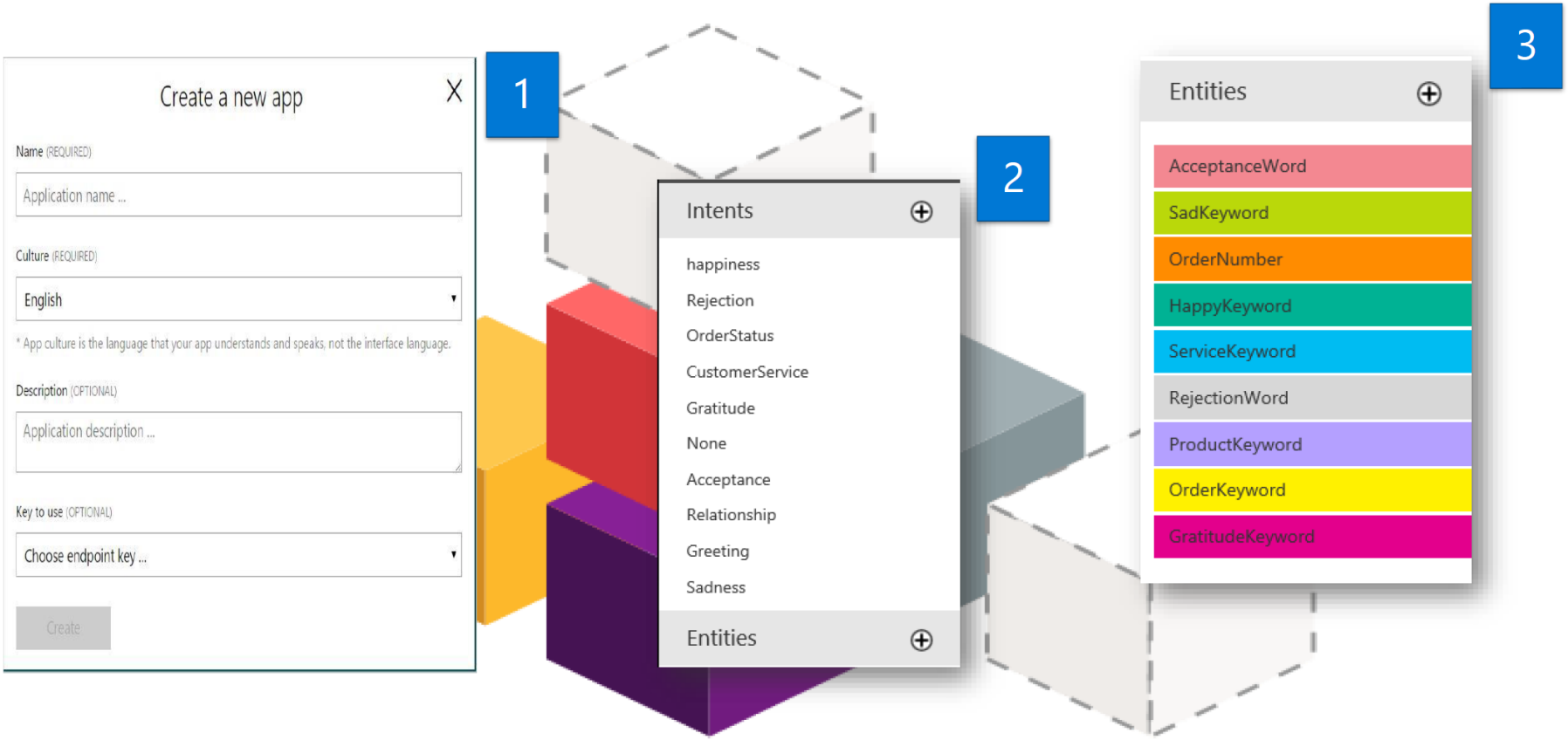
Key Concepts (Label Utterance)

Utterance

book me 2 tickets to Marrakech tomorrow on Air France

Number Location Date Airline

Create your own Luis … App!



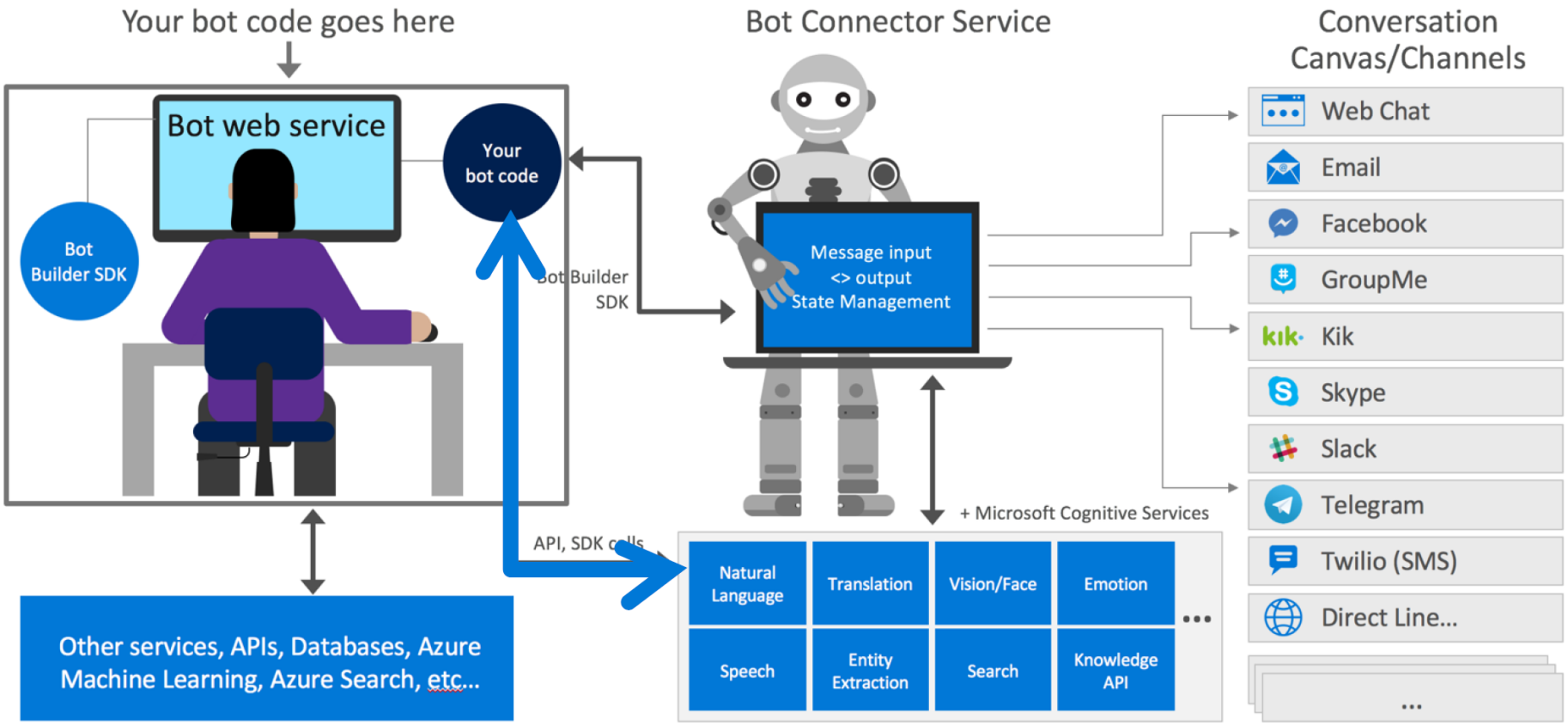
Demo

Luis.ai

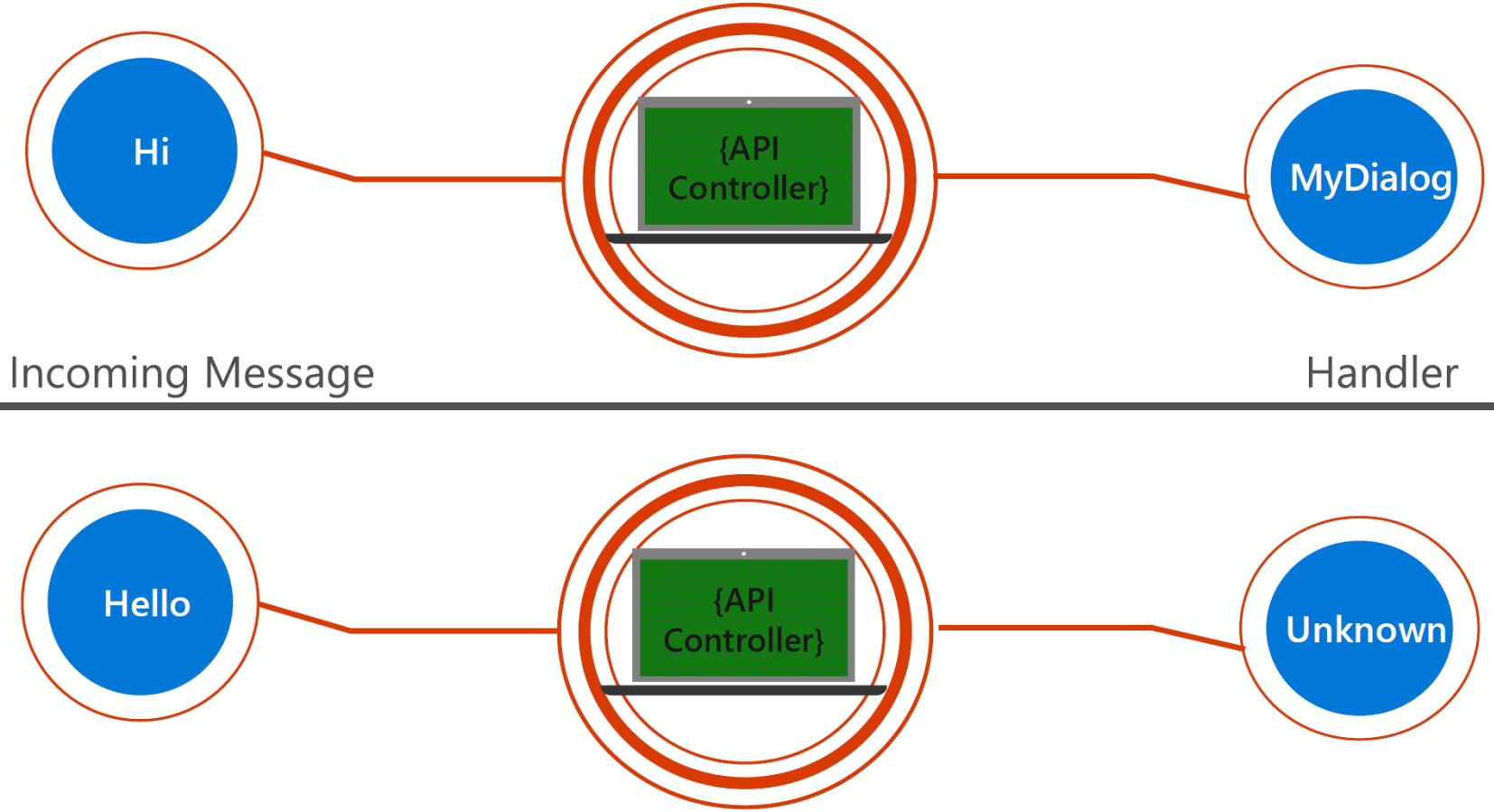
orange networks
IT-CONSULTING - IT-DEVELOPMENT

Integrating LUIS with a Bot

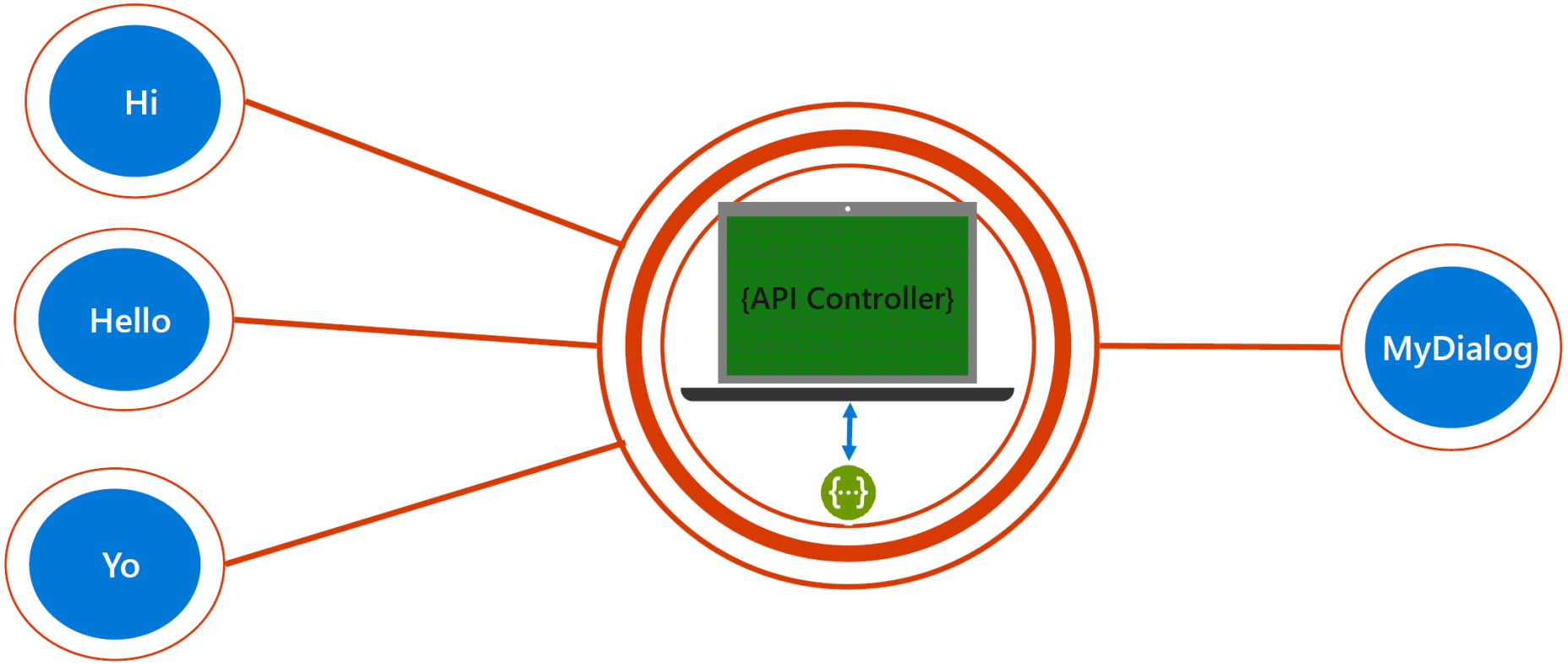
Bot Framework



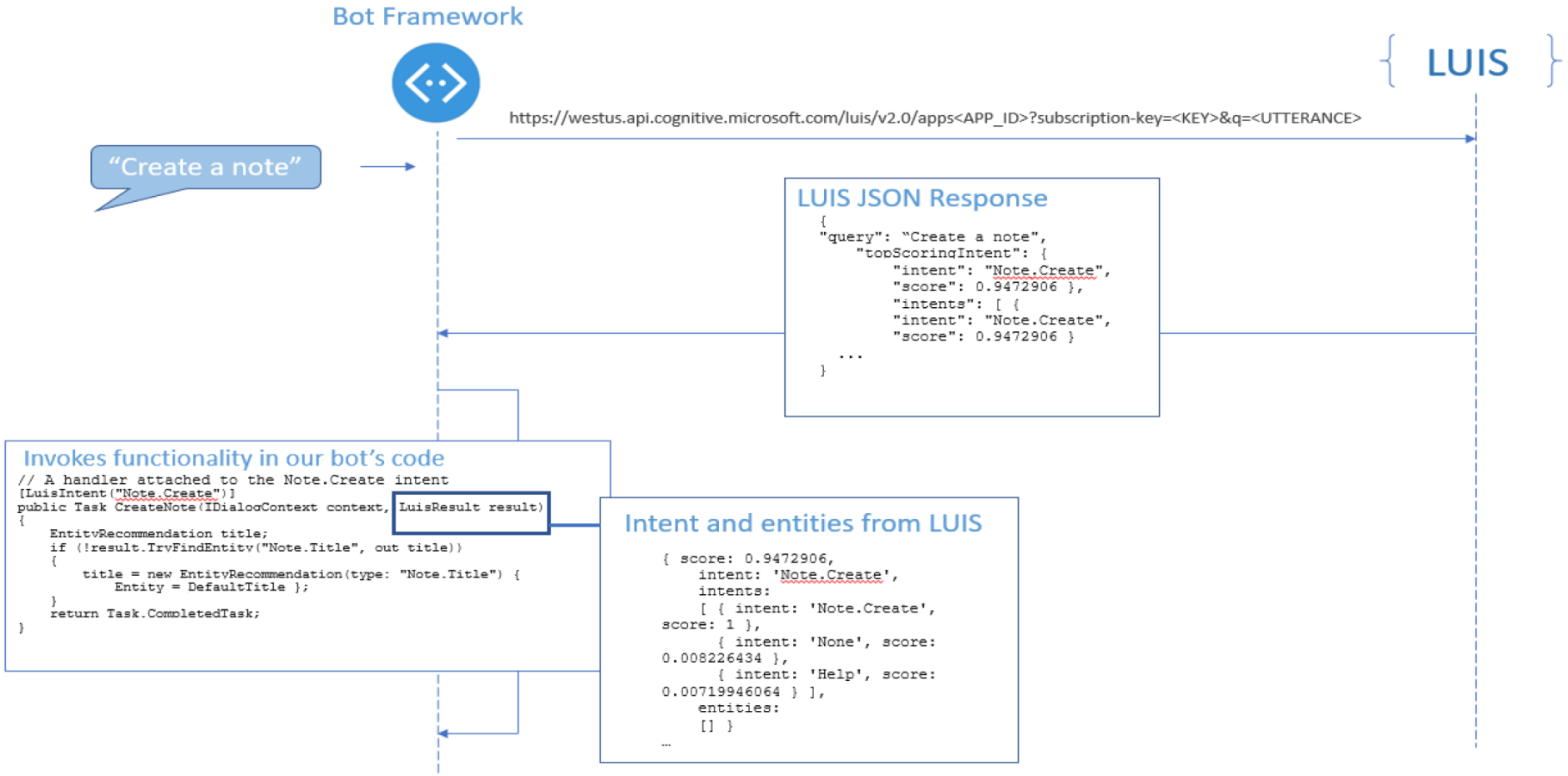
A world without LUIS



A world with LUIS



How LUIS passes intents and entities to your bot



LUIS Dialog Development

- Add Recognizer Plugin from Botbuilder Node SDK (LuisRecognizer)

```
// You can provide your own model by specifing the 'LUIS_MODEL_URL' environment variable
// This Url can be obtained by uploading or creating your model from the LUIS portal: https://www.luis.ai/
var recognizer = new builder.LuisRecognizer(process.env.LUIS_MODEL_URL);
bot.recognizer(recognizer);
```

LUIS Dialog Development

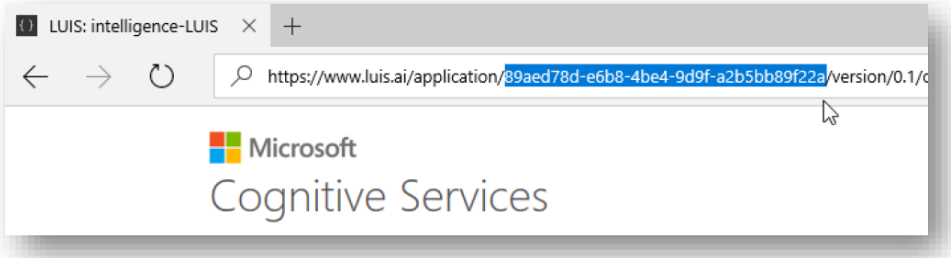
- Add triggerAction option matches to trigger the LUIS.ai Intent

```
bot.dialog('ShowHotelsReviews', function (session, args) {
    // retrieve hotel name from matched entities
    var hotelEntity = builder.EntityRecognizer.findEntity(args.intent.entities, 'Hotel');
    if (hotelEntity) {
        session.send('Looking for reviews of \'%s\'...', hotelEntity.entity);
        Store.searchHotelReviews(hotelEntity.entity)
            .then(function (reviews) {
                var message = new builder.Message()
                    .attachmentLayout(builder.AttachmentLayout.carousel)
                    .attachments(reviews.map(reviewAsAttachment));
                session.endDialog(message);
            });
    }
}).triggerAction({
    matches: 'ShowHotelsReviews'
});
```

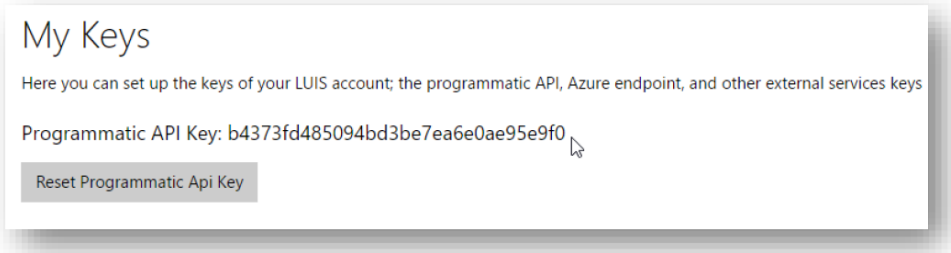
LUIS Dialog Development

LuisModelAttribute Parameters

- modelID:



- subscriptionKey:The LUIS subscription key.



- domain: Domain where LUIS model is located.

https://docs.microsoft.com/en-us/bot-framework/dotnet/bot-builder-dotnet-luis-dialogs#regions-and-keys

Retrieves Entity in args Parameter

- To retrieve an entity value from LUIS EntityRecognizer.findEntity method.
- Entity is part of the args Parameter

```
bot.dialog('ShowHotelsReviews', function (session, args) {
    // retrieve hotel name from matched entities
    var hotelEntity = builder.EntityRecognizer.findEntity(args.intent.entities, 'Hotel');
    if (hotelEntity) {
        session.send('Looking for reviews of \'%s\'...', hotelEntity.entity);
        Store.searchHotelReviews(hotelEntity.entity)
            .then(function (reviews) {
                var message = new builder.Message()
                    .attachmentLayout(builder.AttachmentLayout.carousel)
                    .attachments(reviews.map(reviewAsAttachment));
                session.endDialog(message);
            });
    }
}).triggerAction({
    matches: 'ShowHotelsReviews'
});
```

Demo

Luis Bot with NodeJs

Lab

Lab 2 – Developing a smarter Bot

orange networks
IT-CONSULTING - IT-DEVELOPMENT

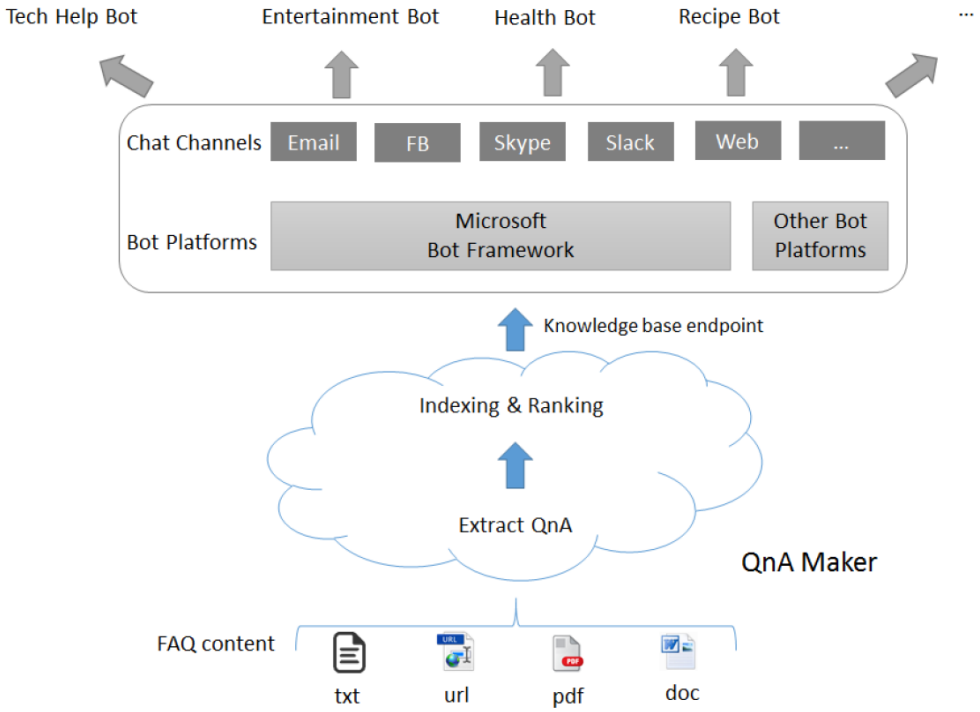
QnA Maker

What is QnA Maker?

- It's a free, easy-to-use, RESP API and Web service that trains AI.
- Allows the bot to provide answers to user's questions in a more natural, conversational way.
- Compatible across development platforms, hosting services, and channels.
- QnA Maker is the only question and answer service with a graphical user interface; meaning you don't need to be a developer to train, manage, and use it for a wide range of solutions.

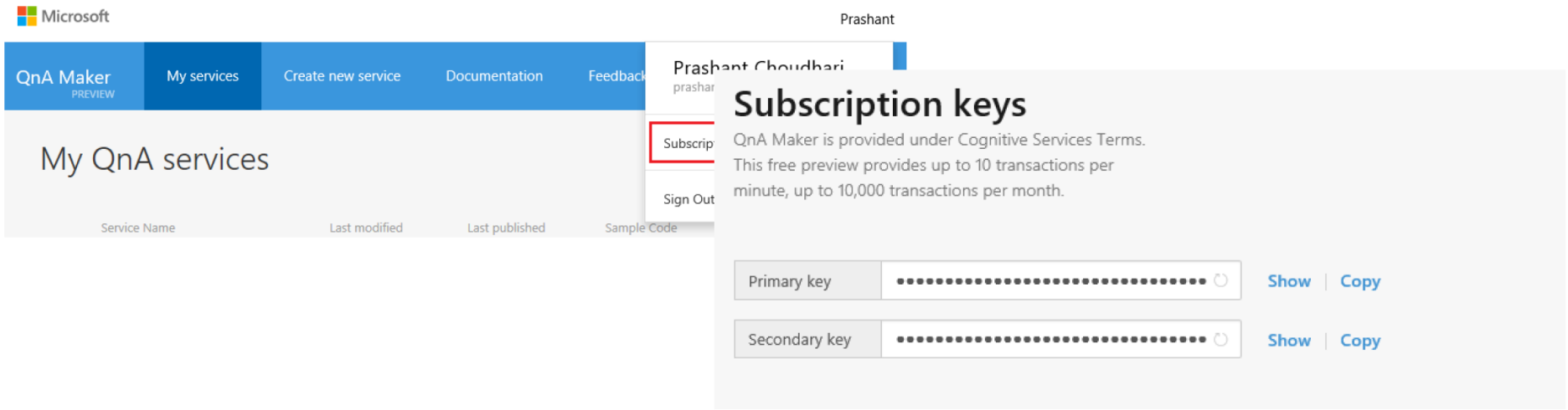
What is QnA Maker?

- With optimized Machine Learning logic and the ability to integrate industry-leading language processing with ease, QnA Maker distills masses of information into distinct, helpful answers.



Authentication & Subscription keys

- It is required a Microsoft account to sign in to the portal.
- You will receive a unique pair of keys.
  - The second one is just a spare.
  - Do not share the secret keys with anyone.
- These subscription keys are used to track your usage of the service and need to be part of every request.



Pricing

- Since currently the QnA Maker is a free to use tool.
- We have the following restriction of usage per Subscription key.
  - 10.000 transactions per month.
  - 10 per minute.
- Beyond this your requests will be throttled.

Create your knowledge base

- Currently the tool can auto-extract question and answer pairs from two types of input:

FAQ Pages

- We support extraction from following types of FAQ URLs:
  - Plain FAQ pages: Questions are immediately followed by answers.
  - FAQ pages with section links: Questions are aggregated together and linked to answers on the same page.
  - FAQ pages with linked answers: Questions are aggregated together and linked to answers on a different page.
- We also support extraction from offline doc types (.docx, .doc, .pdf, .xlsx and .tsv)
- Auto-extraction works best on FAQ pages with clear Q-A structure and semantics such as question ending with "?" and question contains interrogative words such as "why", "what", "how", etc.

Product manuals

- We support extraction of Q-As from PDF format product manuals.
- QnA Maker extracts the headings and sub-headings as questions and the proceeding content as the answer.
- Auto-extraction works best of manuals with a table of contents and/or an index page, and clear structure with hierarchical headings.

Update your knowledge base

- There are several ways you might want to update your knowledge base:

Editorial QnA updates:

- Edit the QnA table directly.
- You can:
  - Add/delete a QnA Pair.
  - Add an alteration of an existing QnA Pair.
- This is ideal for quick editorial fixes to your knowledge base.

Update the sources

- You could add/delete/modify existing sources.

Download and upload

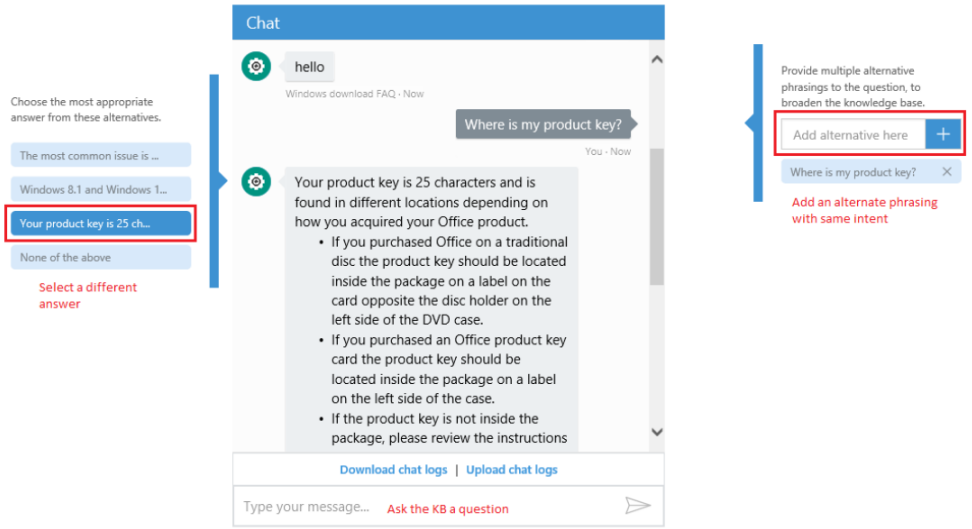
- You could replace your entire knowledge base at one go.
- Ideal for bulk updates to your knowledge base.
- Upload KB expects file format of tab separated columns of Question, Answer and Source.
- You could download the entire KB, make changes, and then upload the knowledge base.

Train your knowledge base

- The train feature lets you evaluate the correctness of the responses and correct them and re-train the KB.
- There are two ways you can improve the relevance of the responses.

Chat with your KB

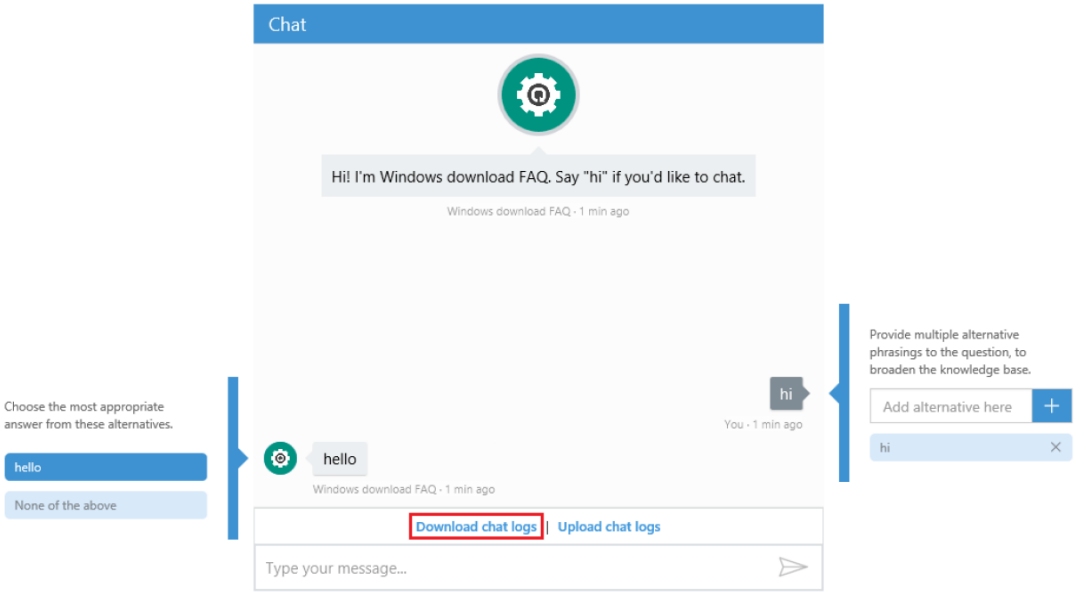
- While chatting with your KB, you can see the relevance of the responses.
- You can add a variation to an existing question as well as choose a different answer for a question.



Train your knowledge base

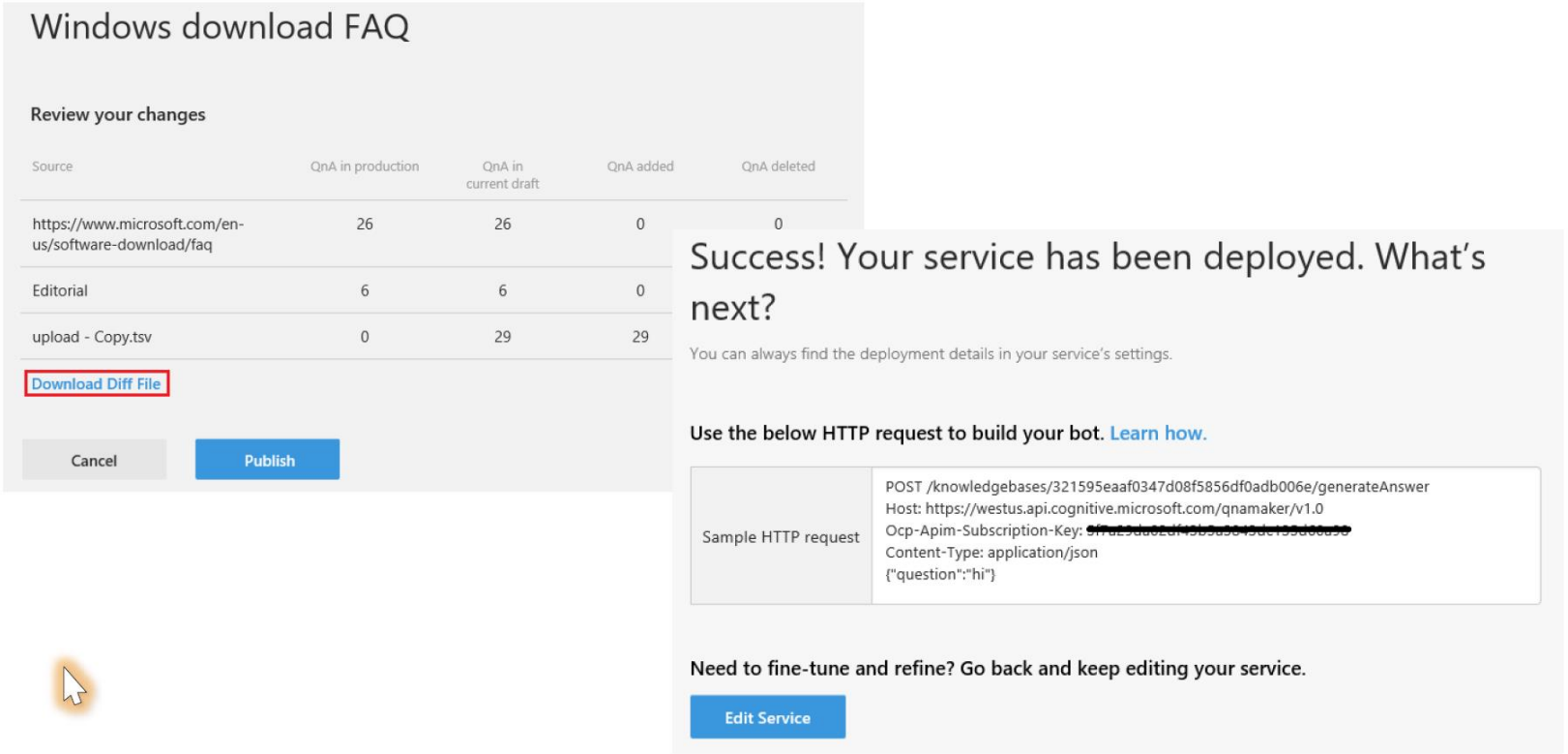
Replay live chat logs

- A very useful feature is to see what responses the service returns for live traffic, and the train it appropriately.
- You can download the live chat traffic hitting your published end-point.
  - This will download all the questions hitting your end-point in descending order of frequency.
- Looking at the chat logs, you can decide which questions you want to test and train your KB base on.



Publish your knowledge base

- Before the final publish, you can preview the changes that will affect the KB on final publish.

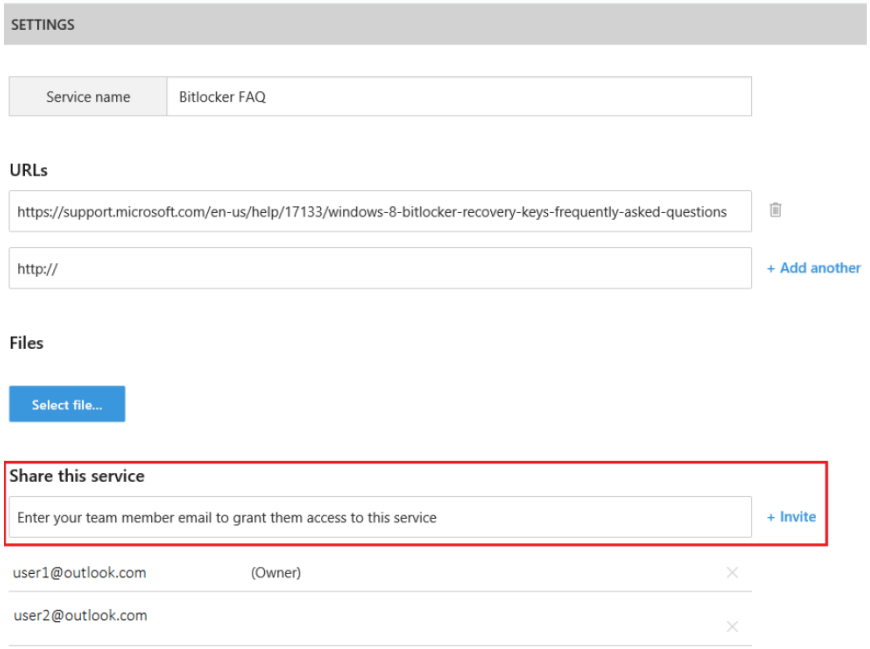


Share your knowledge base

- You could add other users to your KB, so they can contribute to it as well.

Add users to your KB

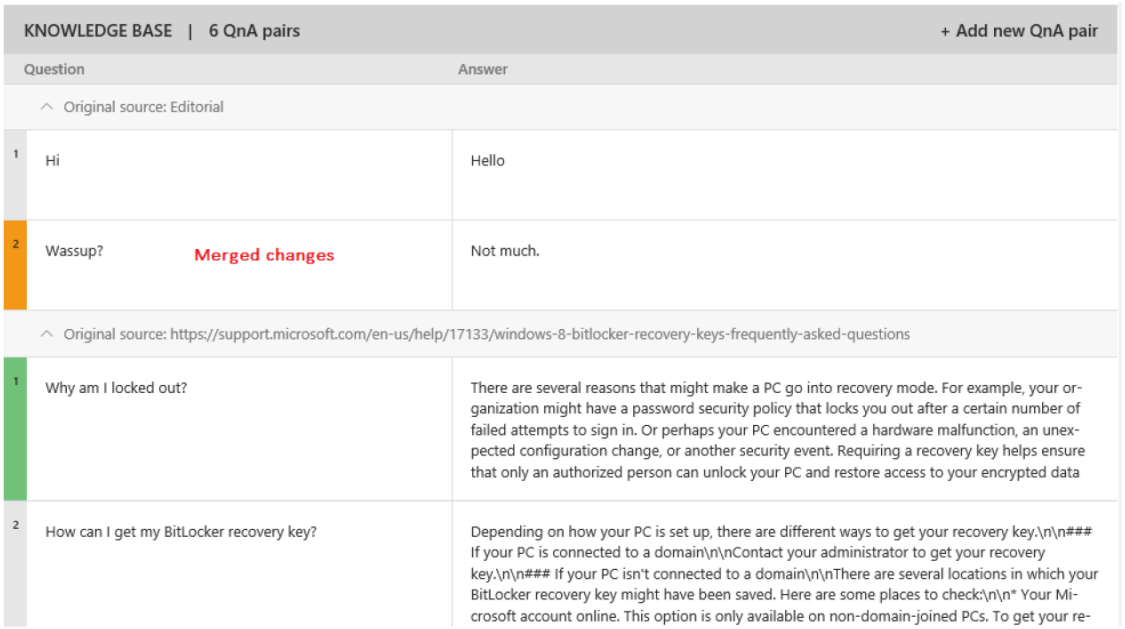
- Go to the settings tab and add users



Share your knowledge base

Merge changes

- In case you have unsaved changes, and another user updates the KB, you will be asked to refresh before you can save your changes.
- When you refresh, you will be able to see the merged changes in orange.



Demo

qnamaker.ai

Demo

Bot + QnAMaker

Lab

Lab 3 – Connect to QnA Maker

orange networks
IT-CONSULTING - IT-DEVELOPMENT

Dialog Stack

Maintaining Conversation State with Dialogs

Conversations rely heavily on the context.

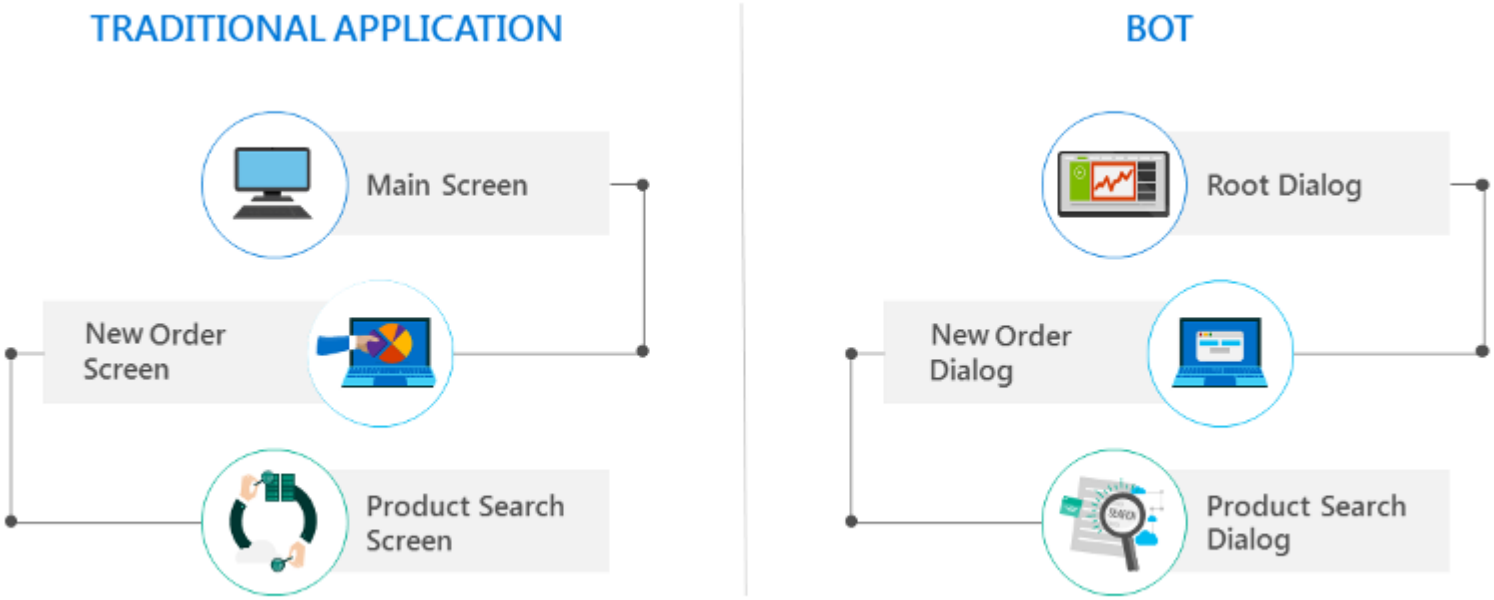
Dialogs and **State** work together to provide context to conversations with chatbots

Maintaining Conversation State with Dialogs

- What Are Dialogs
  - Way of Wrapping an entire “experience” into easily managed interaction based on a “chained” and “conversational” paradigm
  - Dialogs make the UI of a Bot
  - Dialogs may or may not have graphical interfaces
  - They may contain buttons, text, and other elements, or be entirely speech-based
  - Dialogs also contain actions to perform tasks such as invoking other dialogs or processing user input.

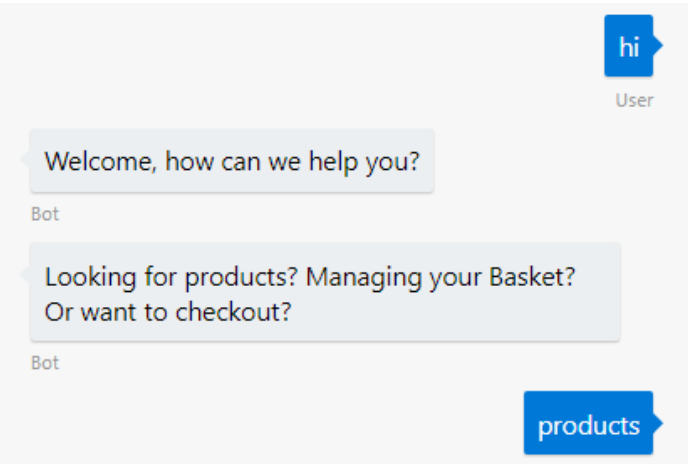
Dialog Stack

- Dialogs and the flow of conversation behind the scenes are managed via the DialogStack.
- The DialogStack keeps track in which Dialog a user currently is.

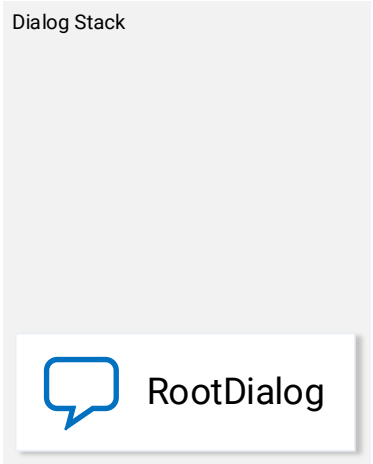


Dialog Stack - Details

Chat

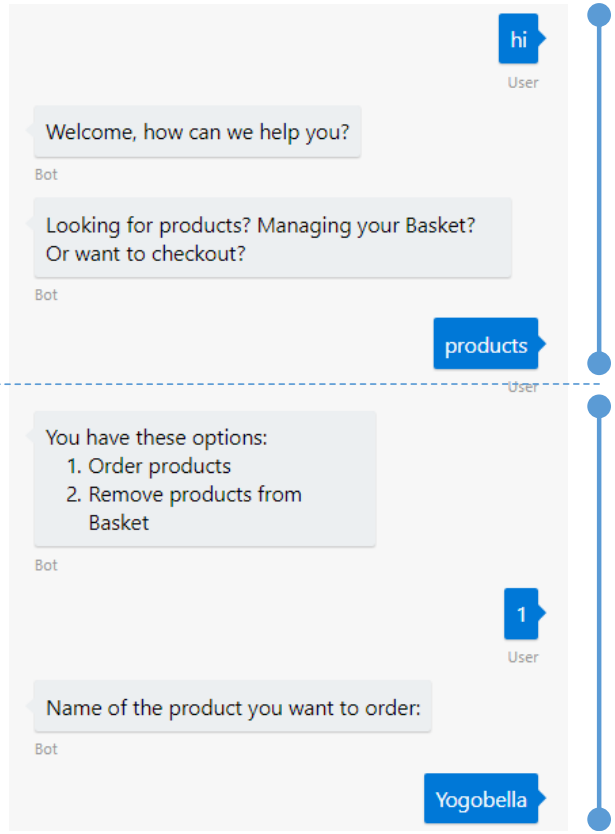


RootDialog



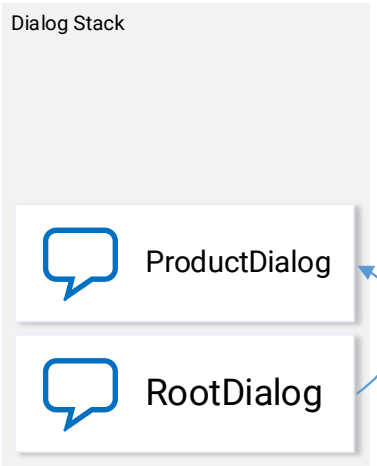
Dialog Stack - Details

Chat



RootDialog

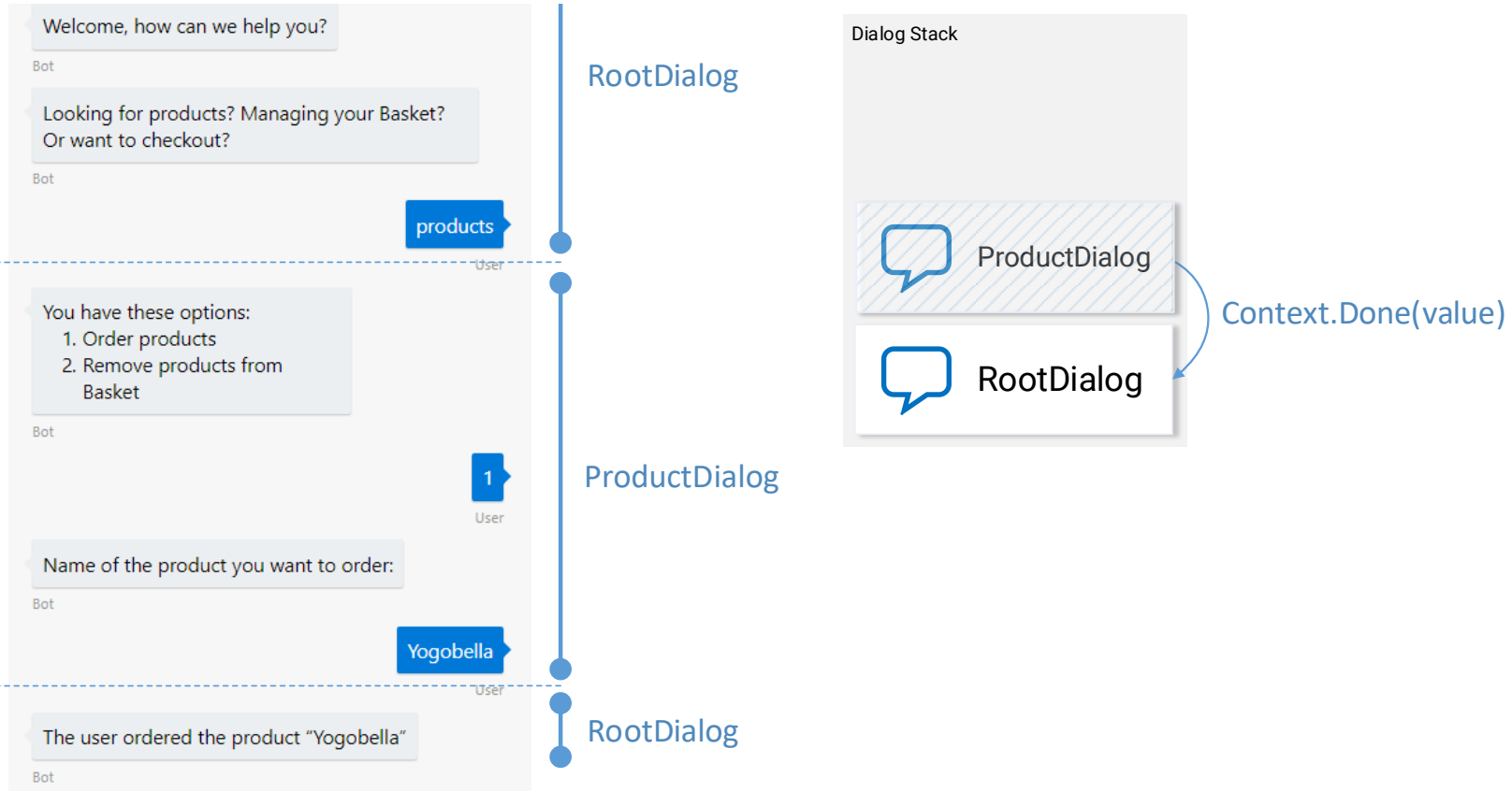
ProductDialog



Context.Forward(ProductDialog)

Dialog Stack - Details

Chat



Dialog

- session.beginDialog('/products') -> forwards the call to the dialog '/products'
- session.endDialog() -> close a dialog and remove it from the stack
- session.endDialogWithResult({...}) -> removes the dialog from the stack, previous dialog takes control and result is parsed to the method in sequence.
- Every method must be ended with context.wait(), context.fail(), context.done() or redirection (Forward, Call)

Lab

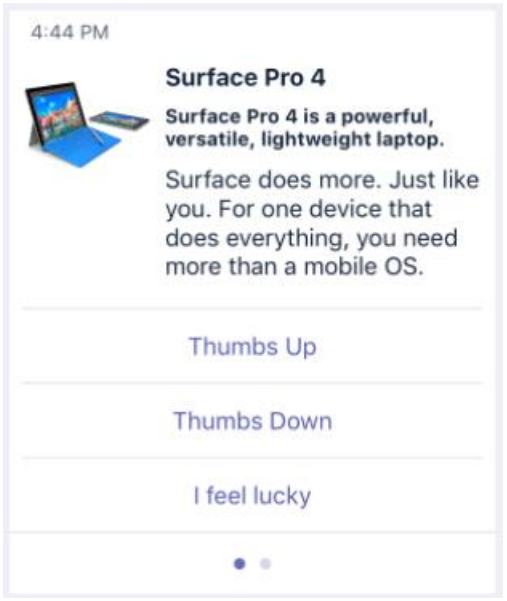
Lab X – Dialogs

orange networks
IT-CONSULTING - IT-DEVELOPMENT

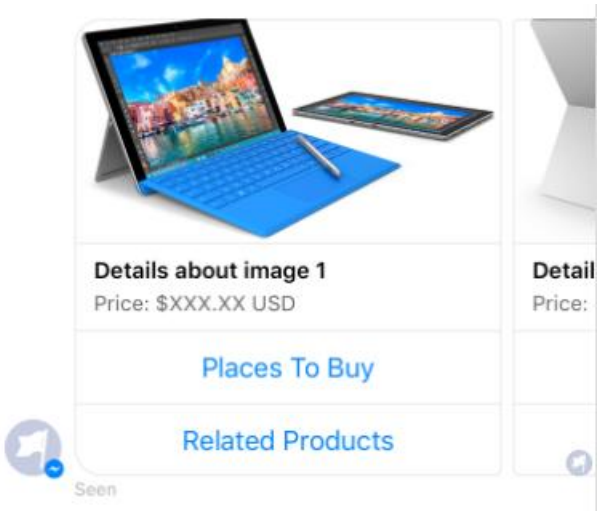
Attachments & Adaptive Cards

Attachments

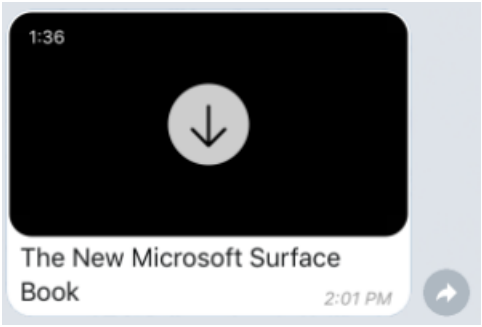
- Text Messages are not always the best medium to transfer information.
- Information exchange can be aided with Images, Videos and Audio Files.
- Bot Framework supports those media using Attachments.



Thumbnail Card



Carousel Card

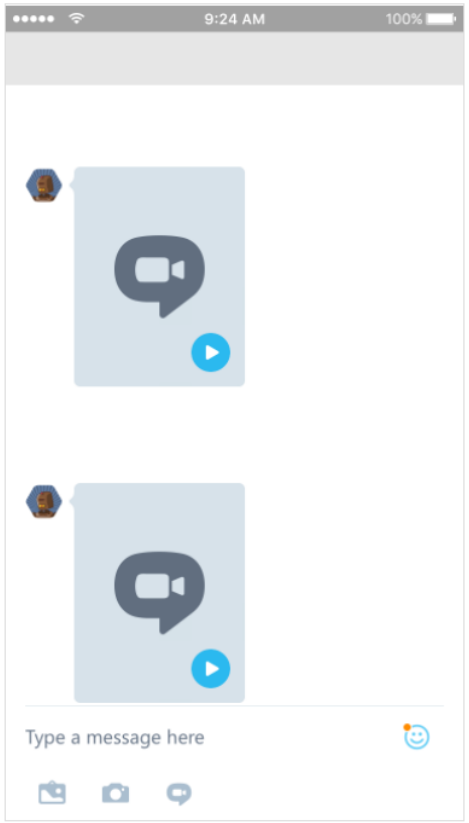


Video Card

Attachments

Not all media types are supported by all channels.

Channel Inspector
Shows what media attachment are supported and how they look in each channel.



Channel Inspector

Channel: Skype

Feature: Videos

Videos

Playing of video attachments.

Bot Framework Documentation

Notes

Appearance	Native Control.
Max Video Size	100 MB
Thumbnail Image	Should be set in thumbnailUrl

See something wrong? Send Feedback

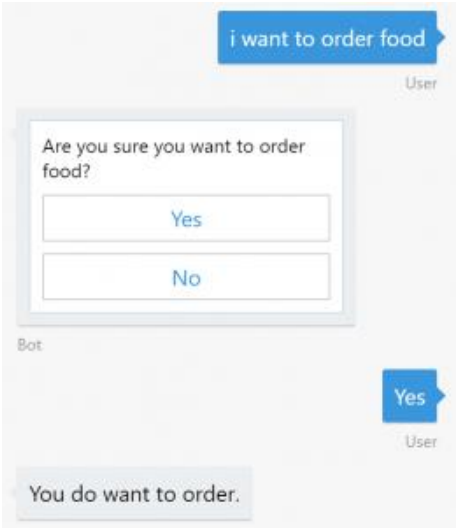
Rich Card Types

HeroCard	A card that typically contains a single large image, one or more buttons, and text.
AudioCard	A card that can play an audio file.
AnimationCard	A card that can play animated GIFs or short videos.
ThumbnailCard	A card that typically contains a single thumbnail image, one or more buttons, and text.
ReceiptCard	A card that enables a bot to provide a receipt to the user. It typically contains the list of items to include on the receipt, tax and total information, and other text.
SignInCard	A card that enables a bot to request that a user sign-in. It typically contains text and one or more buttons that the user can click to initiate the sign-in process.
VideoCard	A card that can play videos.
AdaptiveCard	A card that can contain any combination of text, speech, images, buttons, and input fields.

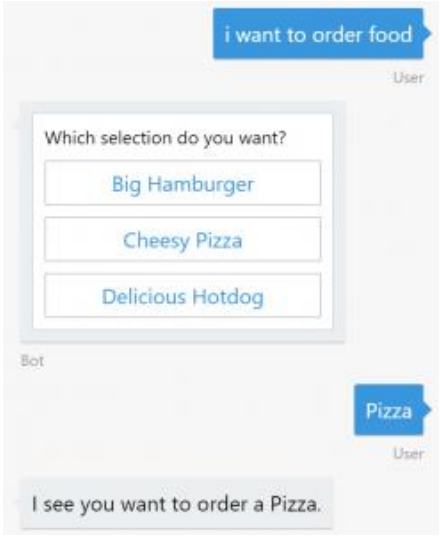
Prompt Dialogs

Simple way to provide the user with different type of dialogs

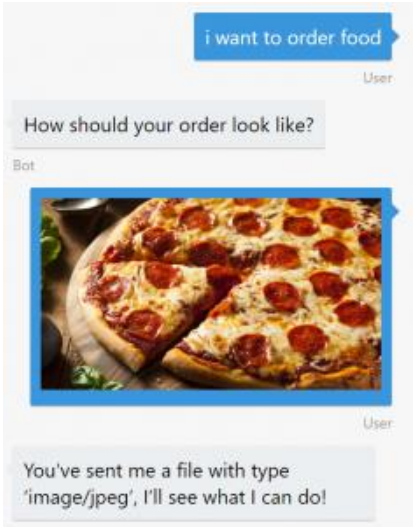
Confirm



Choice

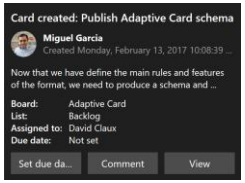


Attachment



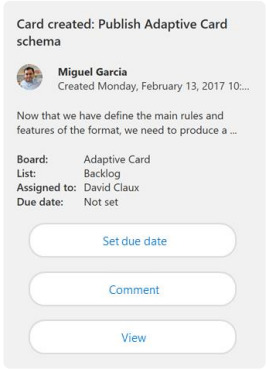
Adaptive cards

OPEN FRAMEWORK
MULTIPLE CANVASES

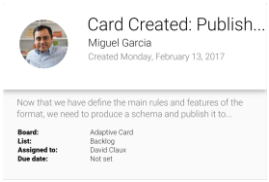


Notification

Microsoft Teams



Skype



Android

iOS

Payments

Works across most major platforms

Fast and simple payment experience

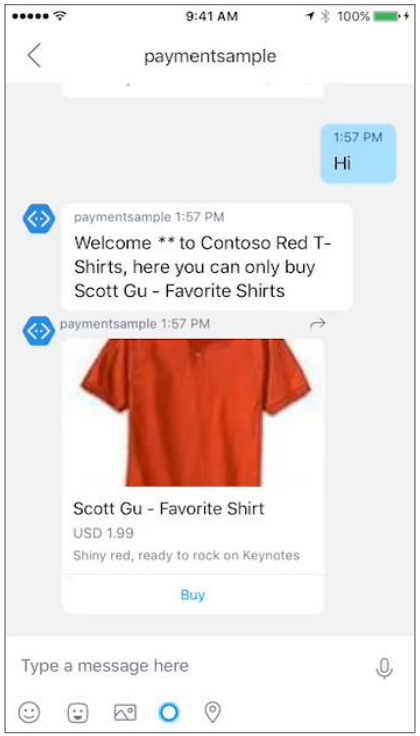
Bring commerce capabilities to your bot

Stripe is currently supported

Visit Microsoft seller center

Obtain merchant ID

Start building

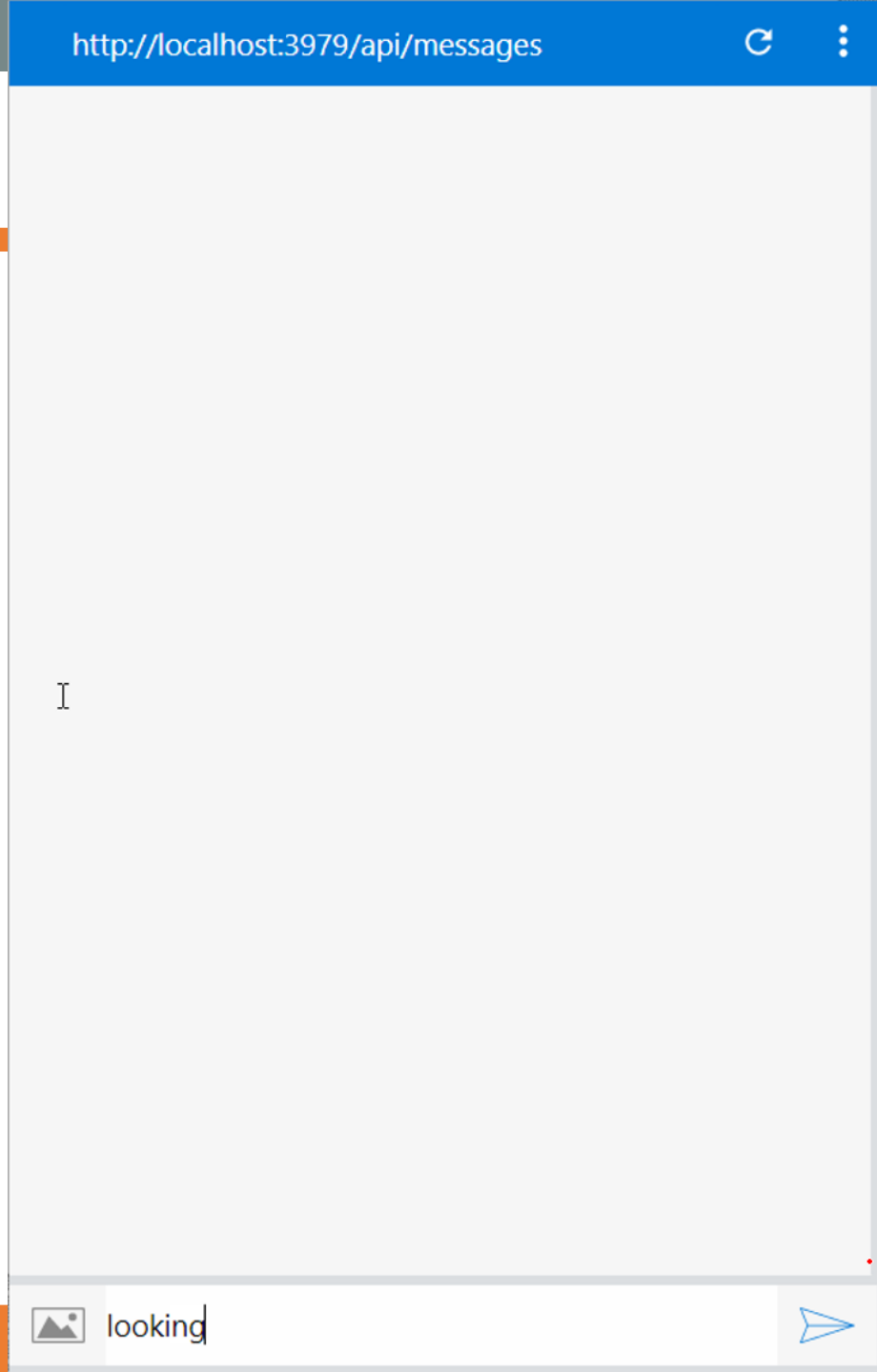


Learn more on Microsoft Seller Center!

Read the integration guide for the bot framework

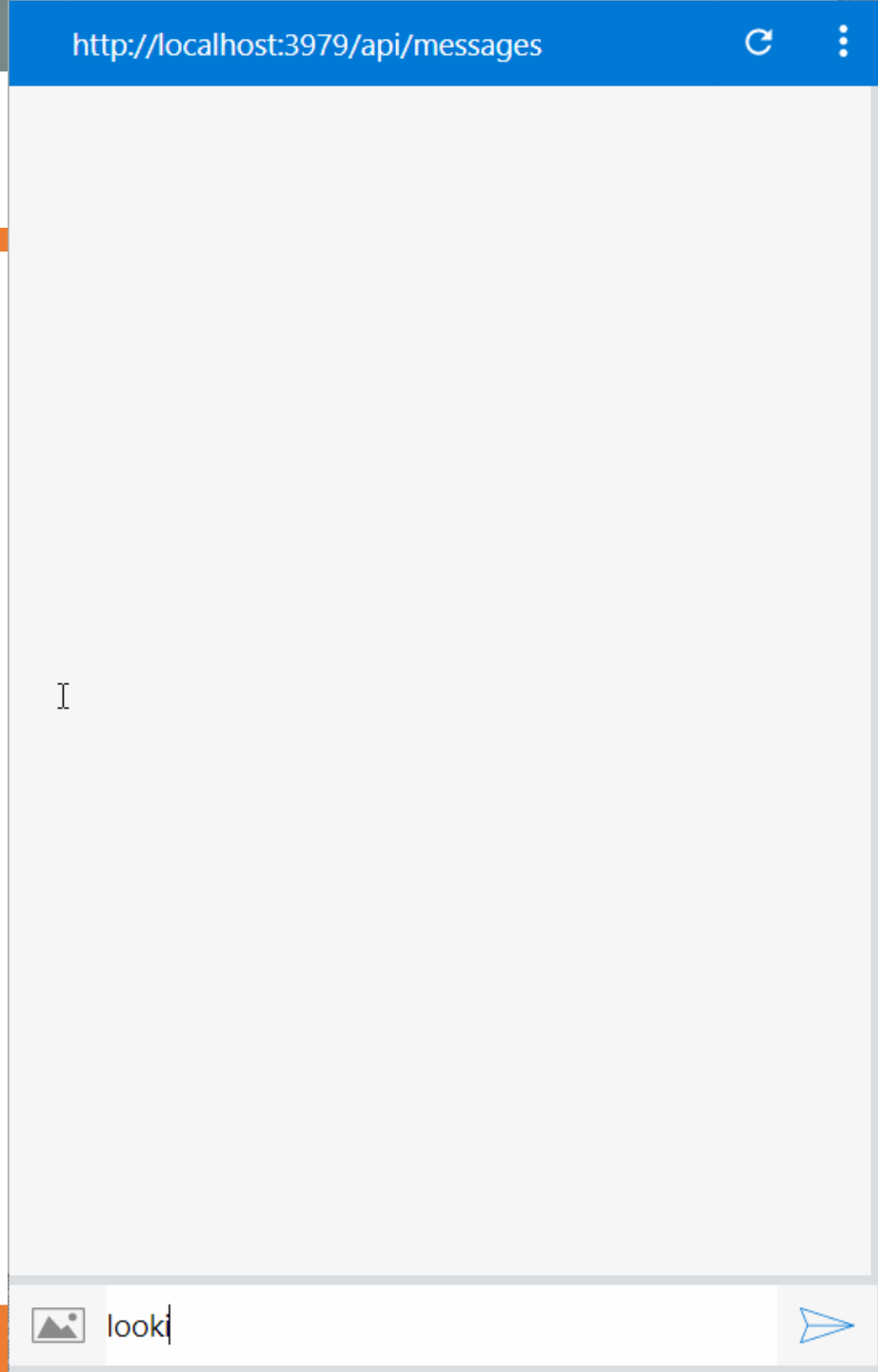
Adaptive Cards

- Adaptve Cards enable:
- Rich presentation experience,
- Improve the Bot UX, and
- SDKs for:
  - JavaScript
  - Android
  - iOS
  - UWP
  - .NET

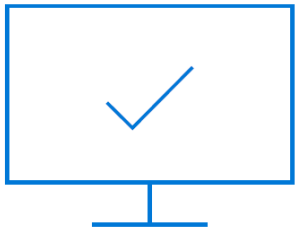


Adaptive Cards

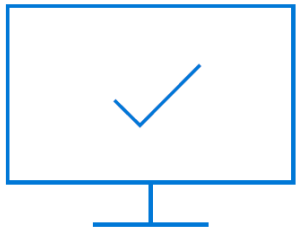
- Adaptve Cards enable:
- Rich presentation experience,
- Improve the Bot UX, and
- SDKs for:
  - JavaScript
  - Android
  - iOS
  - UWP
  - .NET



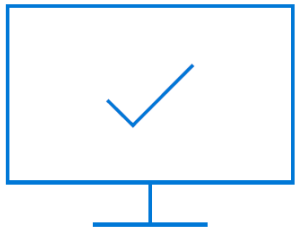
Adaptive Cards



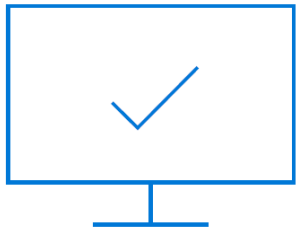
Rich schema for to layout content



Actions to open URLs, show other cards, etc



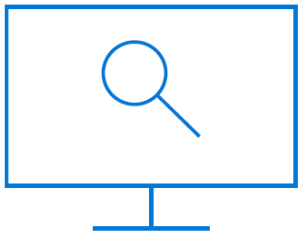
Input to gather information from users



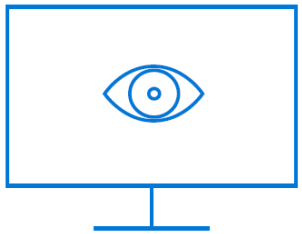
Speech enabled from day-one

Tooling

Interactive Visualizer



Schema Explorer



VS Code Previewer

Create and send an Adaptive Card from your bot

Adaptive Cards offer much more than just customizable text:

- Add Images to your card
- Organize your content with Containers and Columns
- Add multiple types of Actions
- Collect Input from your users
- Have one card show another card

Create and send an Adaptive Card from your bot

Property	Type	Required	Description
type	`"AdaptiveCard"`	Yes	Must be AdaptiveCard
actions	`Action[]`	No	The Actions to show in the card's action bar
body	`array[]`	No	The Card Elements to show in the primary card region
version	`string`	Yes	version of schema that this card was authored
minVersion	`string`	No	if a client doesn't support the minVersion the card should be rejected and return the fallbackText. If it does, then the elements that are not supported are safe to ignore
fallbackText	`string`	No	if a client is not able to show the card, show fallbackText to the user. This can be in markdown format.
speak	`string`	No	Specifies what should be spoken for this entire Item. This is simple text or SSML fragment

Create an Adaptive Card in NodeJs

```
// Display Welcome card with Hotels and Flights search options
var card = {
    'contentType': 'application/vnd.microsoft.card.adaptive',
    'content': {
        '$schema': 'http://adaptivecards.io/schemas/adaptive-card.json',
        'type': 'AdaptiveCard',
        'version': '1.0',
        'body': [
            {
                'type': 'Container',
                'speak': '<s>Hello!</s><s>Are you looking for a flight or a hotel?</s>',
                'items': [
                    {
                        'type': 'ColumnSet',
                        'columns': [
                            {
                                'type': 'Column',
                                'size': 'auto',
                                'items': [
                                    {
                                        'type': 'Image',
                                        'url': 'https://placeholdit.imgix.net/~text?txtsize=65&txt=Adaptive+Cards&w=300&h=300
                                        'size': 'medium',
                                        'style': 'person'
                                    }
```

Send an Adaptive Card From Your Bot

```
var msg = new builder.Message(session)
    .addAttachment(card);
session.send(msg);
```

Demo

http://adaptivecards.io/

Lab

Lab 4 – Utilize Adaptive Cards with Bots

orange networks
IT-CONSULTING - IT-DEVELOPMENT

Bot Design Principles

- Provides business value
  - Bots must generate productivity and / or revenue like any other business endeavor!
- Well thought out and designed
- Has a good user experience
- Does not try to do everything or re-invent the wheel

Bad Bot Traits

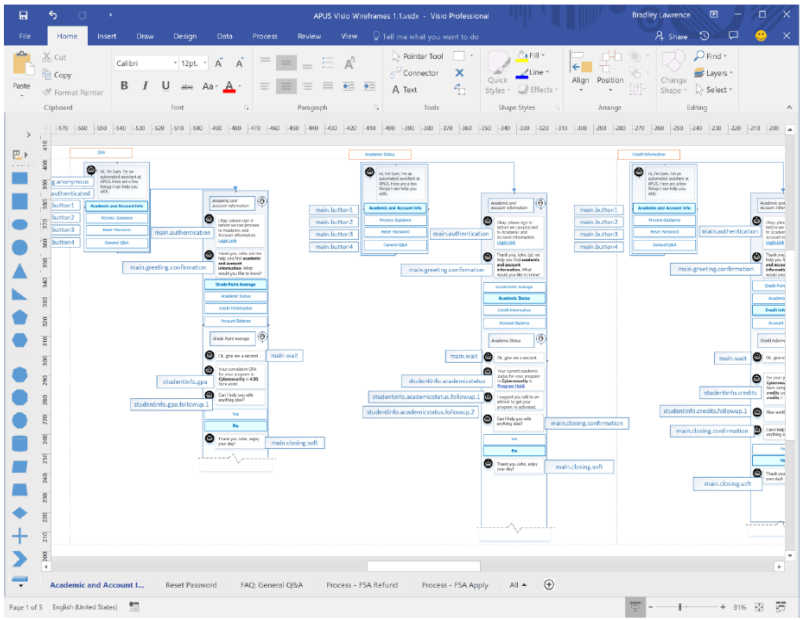
- Unclear business value
- Inconsistent and inflexible design
- Poor user experience
- Attempts to do too much
- Simply duplicates functionality (i.e. search bot anti-pattern)
- Boring design and personality

Good Bots Are Planned

- Understand the opportunity and quantify the ROI
- Provide adequate time for up-front envisioning
- Schedule time for re-training after launch and during the bot's lifecycle
- Create quick MVPs to prove the concept and generate executive and user excitement

Envisioning Sessions

- Develop user stories
- Design conversation layouts
- Key deliverables:
  - Project scope
  - Full conversation maps
  - Fully understood integration points
  - Initial LUIS intent list if applicable



Conversation Design Guidelines

- Provide examples of what the user can ask
- Utilize suggested action buttons
- Never leave the user without a clear path
- Always have a welcome message
- Allow users to ask questions at any time even if you need to cancel the current flow (they will try anyways…)

User Experience Guidelines

- Know which channels users will use
- Don’t get too wordy or send too much information at once
- Be consistent in branding and personality
- Provide access to quick help or ability to send feedback
- Ensure mobile user can use buttons as opposed to typing as much as possible

orange networks
IT-CONSULTING - IT-DEVELOPMENT

Viel Spaß beim Hackathon ☺