

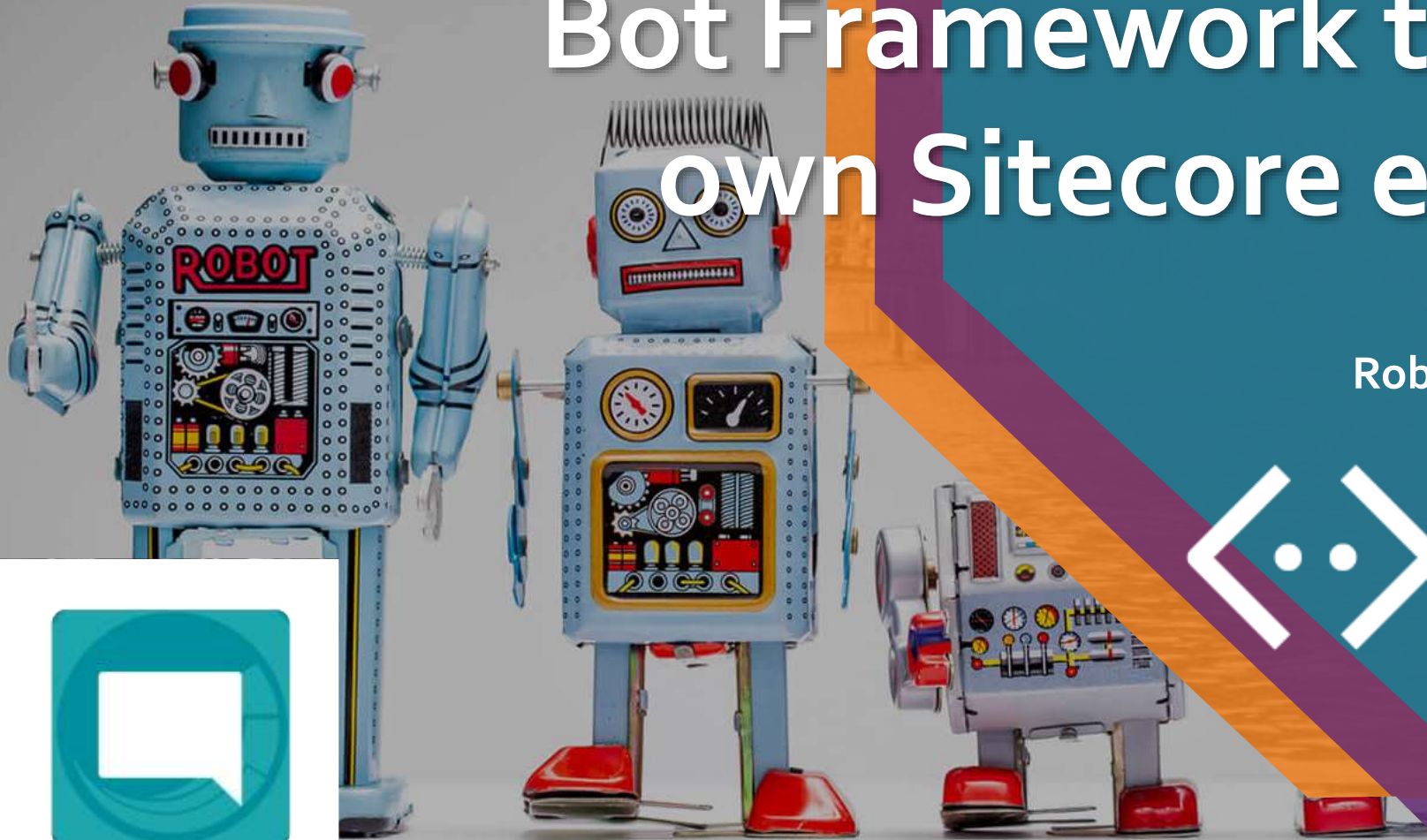
# Leveraging the Microsoft Bot Framework to build your own Sitecore enabled bots

Robbert Hock & Alex van Wolferen

May 18<sup>th</sup> 2017  
Amsterdam / The Netherlands



**SUGCON**  
EUROPE 2017



# Who are we?



**Robbert Hock**

**Freelance Senior Sitecore Specialist  
Kayee**

**@kayeeNL**



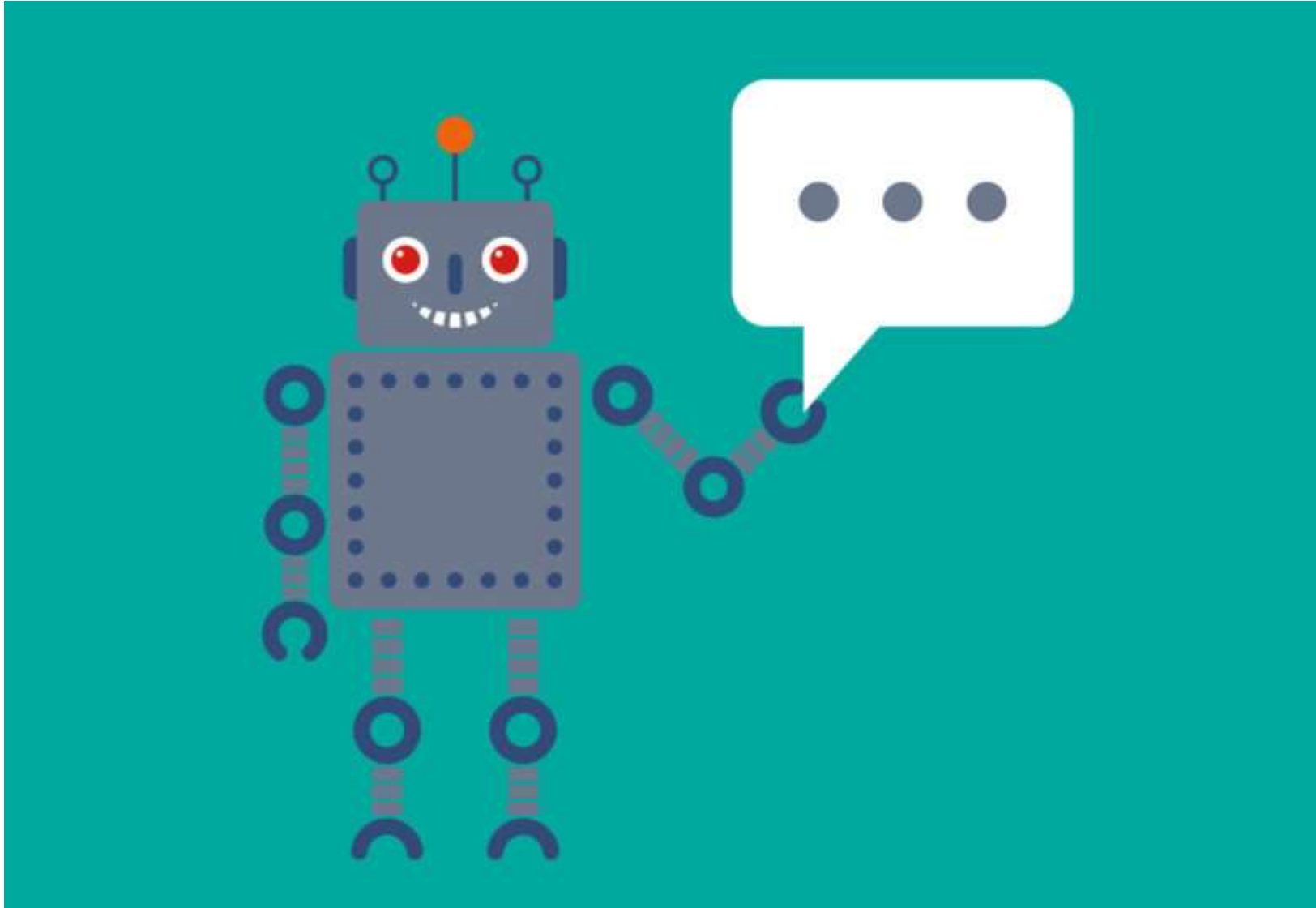
**Alex van Wolferen**

**Senior Sitecore Architect  
Suneco**

**@avwolferen**

- Microsoft Bot Framework
- SUGCON Bot
- LUIS
- Connecting Sitecore with the Bot framework

# Why bots?

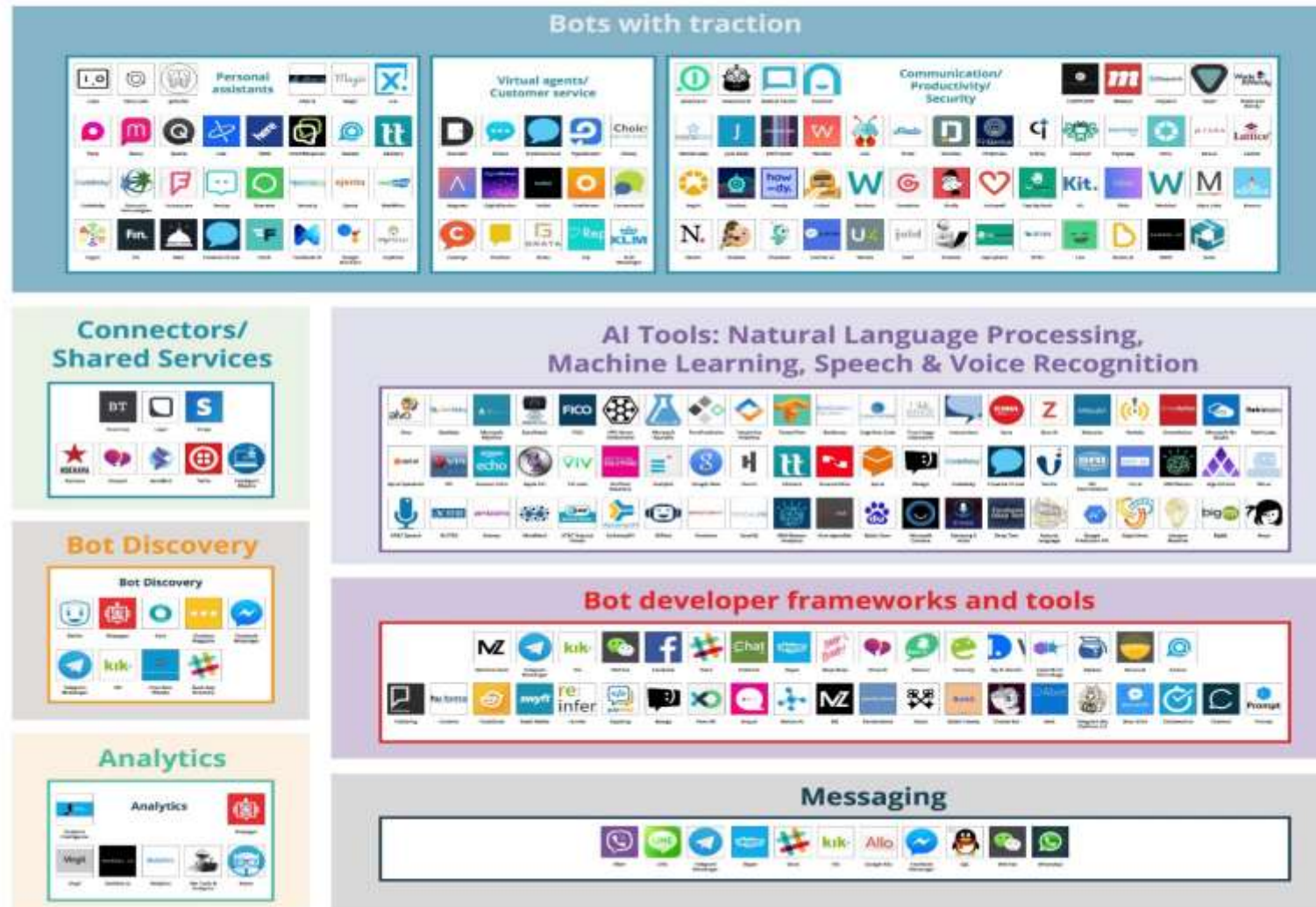




DESIGNED BY  
JON CIFUENTES

## Bots Landscape

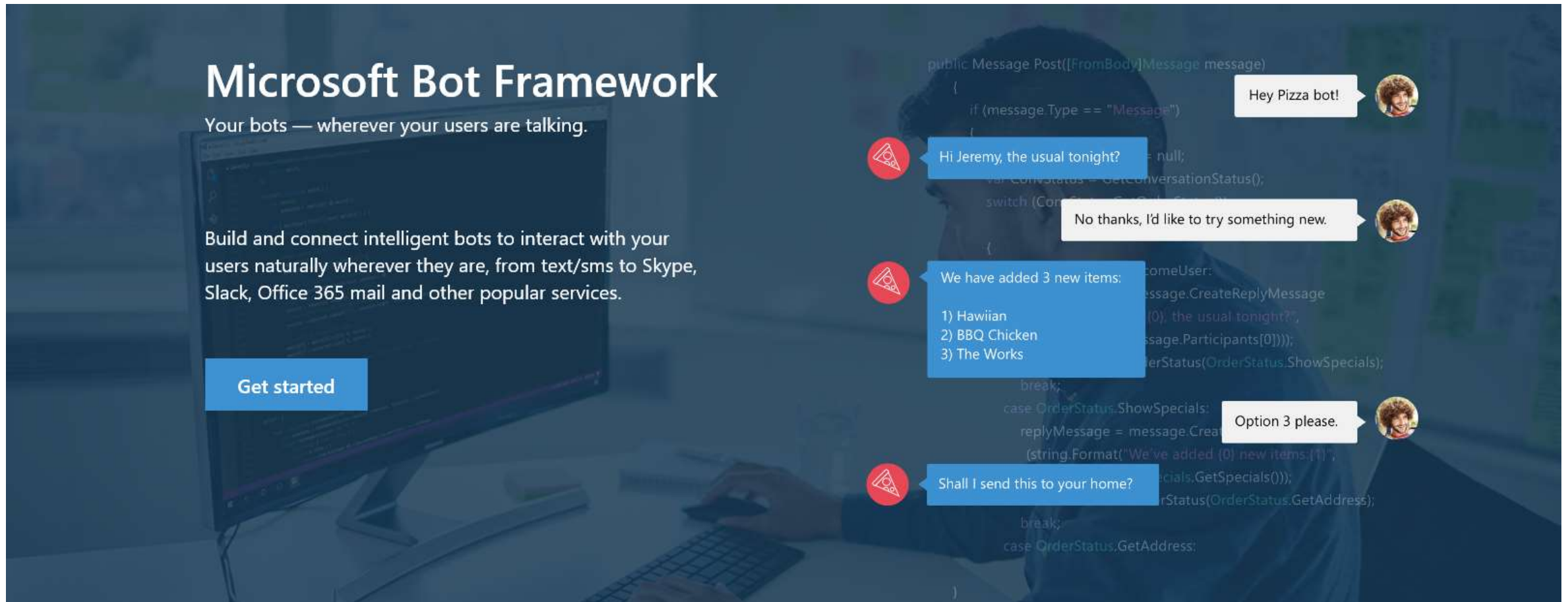
POWERED BY  
**VB | Profiles**





**Microsoft CEO  
Satya Nadella**  
“Bots are the new apps”

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



The image shows the Microsoft Bot Framework landing page. On the left, a computer monitor displays code, and a person's hands are visible typing on a keyboard. The main heading is "Microsoft Bot Framework" with the tagline "Your bots — wherever your users are talking." Below this, it says "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." A blue "Get started" button is present. On the right, a chatbot conversation is shown with a pizza icon as the bot's avatar. The conversation includes a greeting, a list of menu items, and an offer to deliver the pizza.

## 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.

[Get started](#)

```
public Message Post([FromBody]Message message)
{
    if (message.Type == "Message")
    {
        var convStatus = GetConversationStatus();
        switch (ConvStatus)
        {
            case OrderStatus.ShowSpecials:
                break;
            case OrderStatus.ShowSpecials:
                replyMessage = message.CreateReplyMessage(
                    string.Format("We've added (0) new items: {1}",
                        specials.GetSpecials()));
                convStatus(OrderStatus.GetAddress);
                break;
            case OrderStatus.GetAddress:
                break;
        }
    }
}
```

Hey Pizza bot!

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

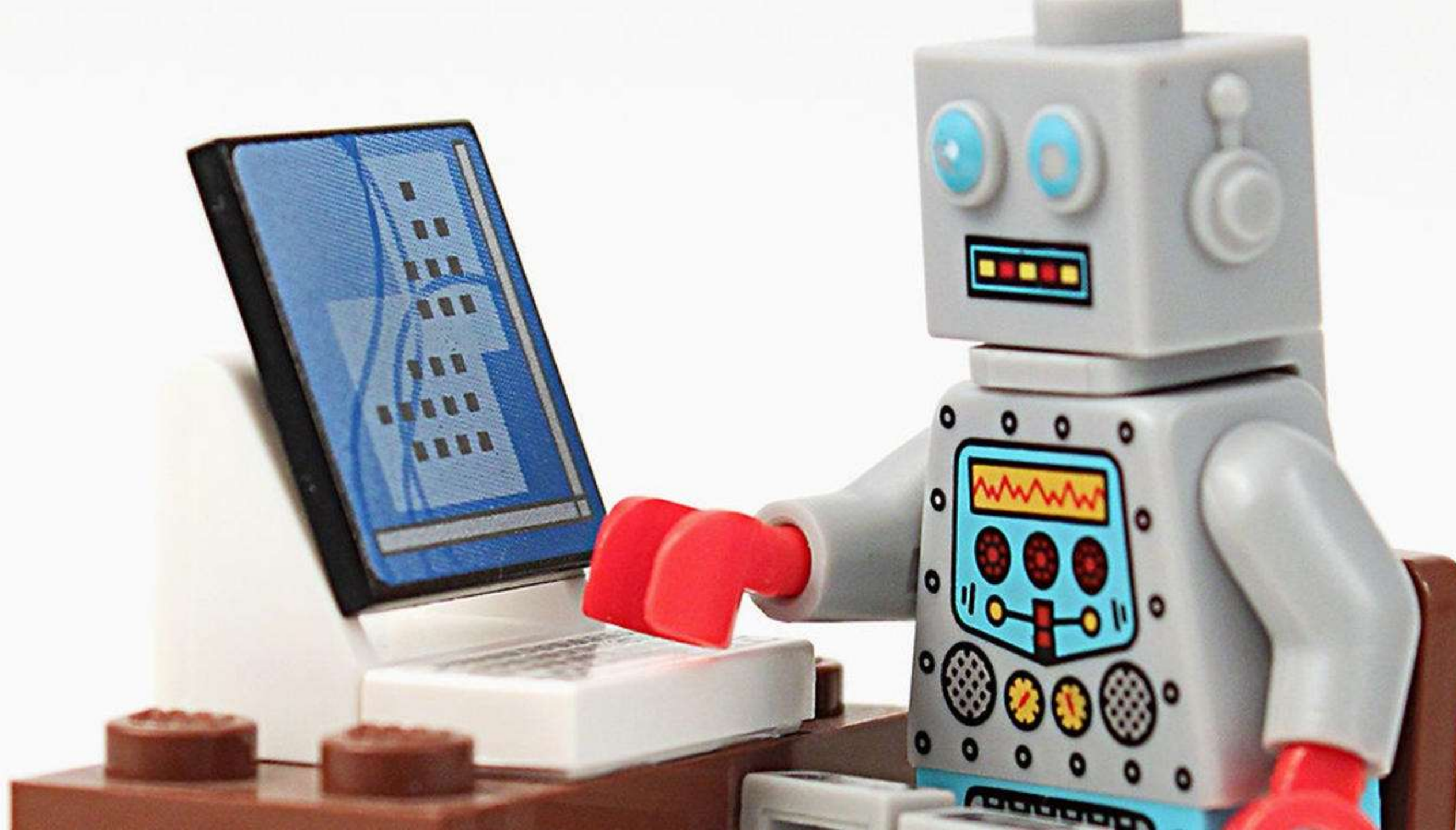
Option 3 please.

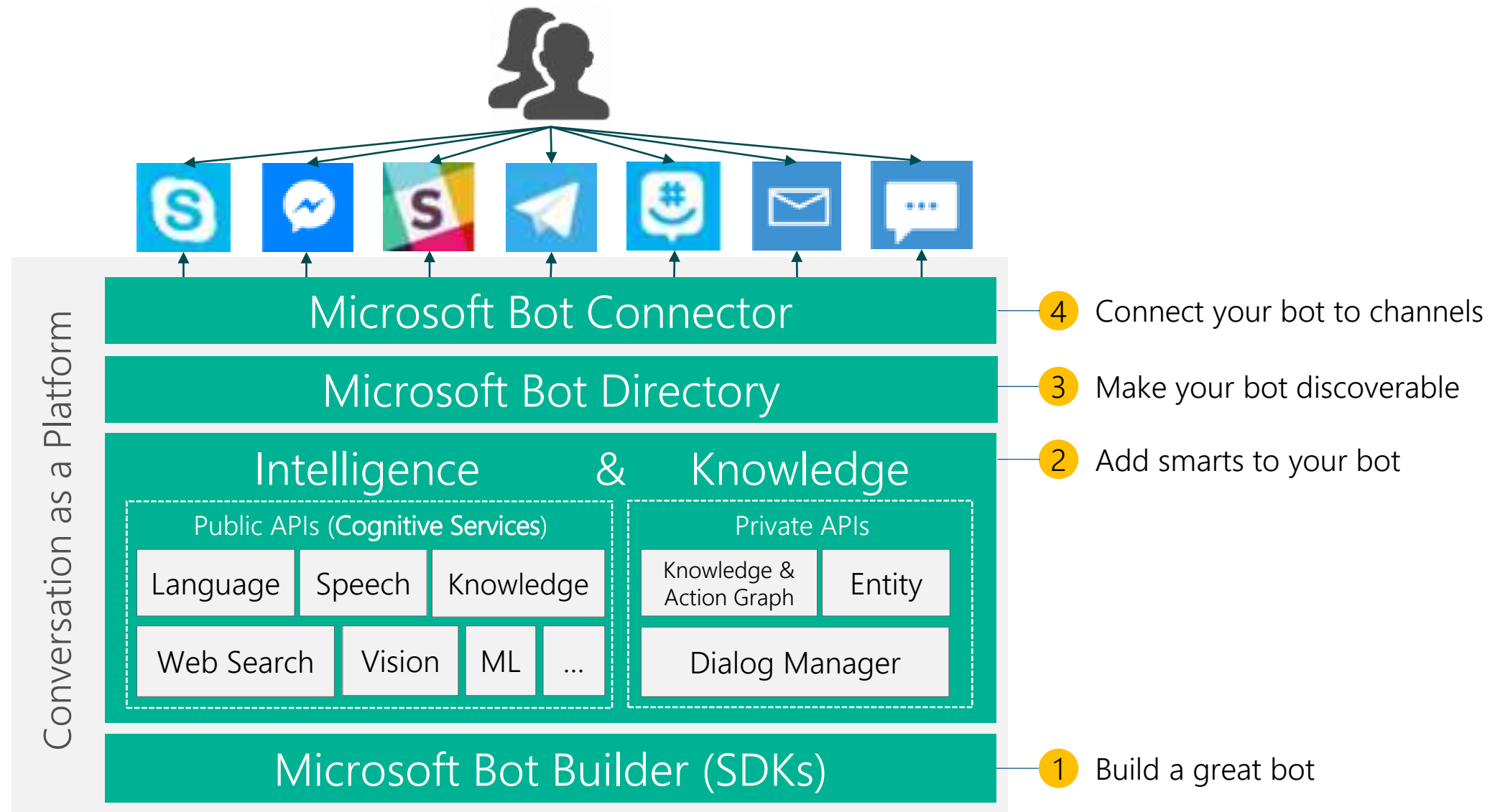
Shall I send this to your home?

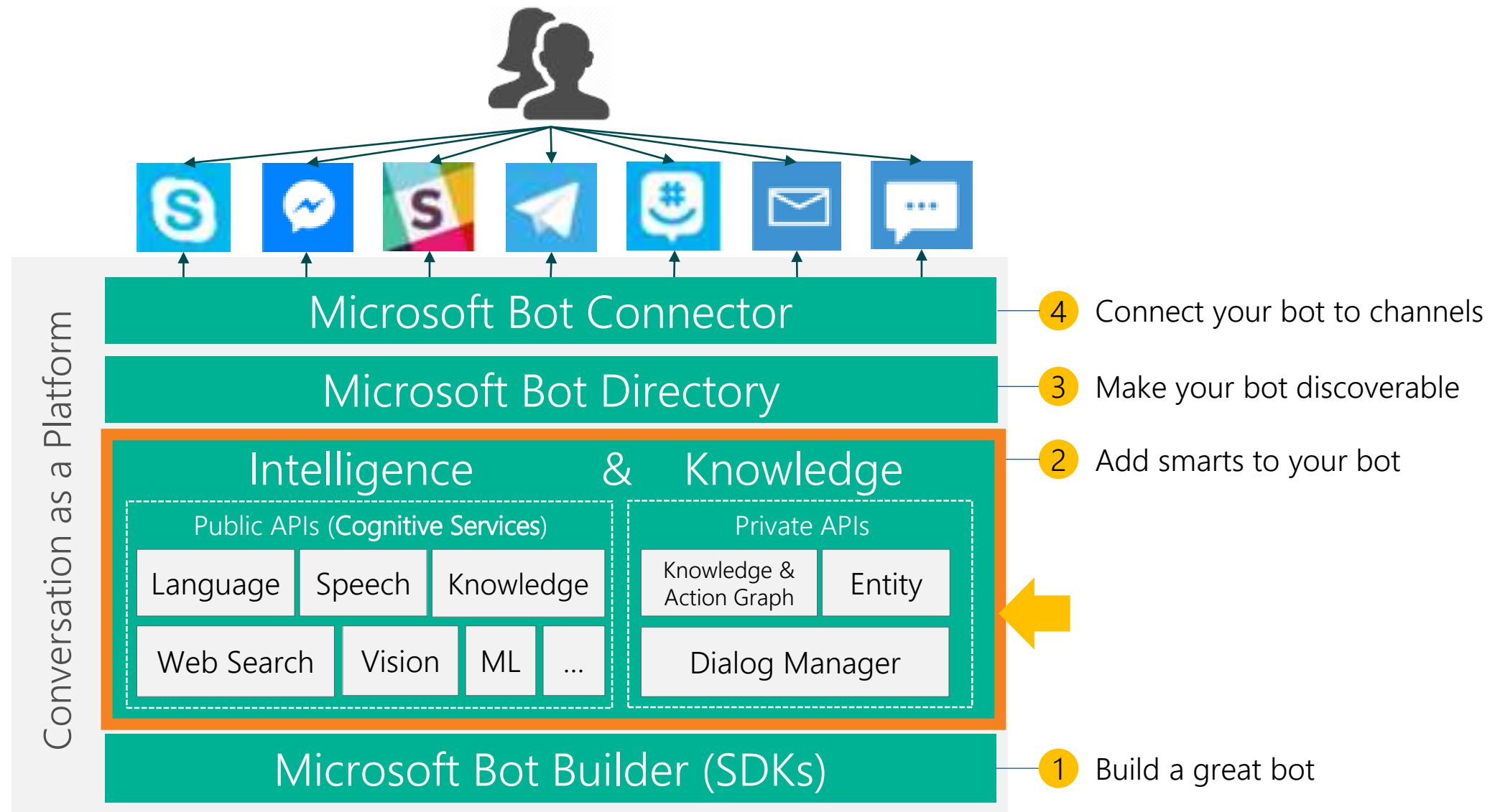
# We have created a bot for you!

We have added a few hundred if-statements...









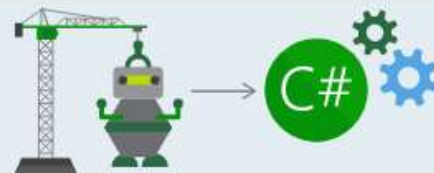
## Bot Framework Documentation

Overview

Developing Bots

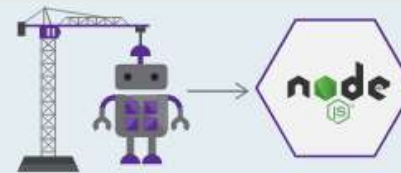
Intelligence

Resources



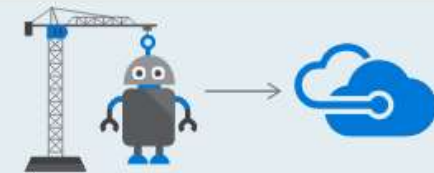
### Build bots with C#

Learn how to build bots with the Bot Builder SDK for .NET.



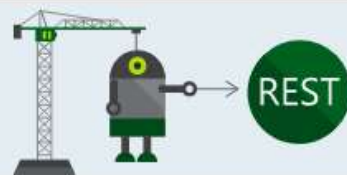
### Build bots with Node.js

Learn how to build bots with the Bot Builder SDK for Node.js.



### Build bots with Azure Bot Service

Learn how to build bots with the Azure Bot Service.



### Build bots with REST

Learn how to build bots with the Bot Framework REST APIs.



### Deploy bots

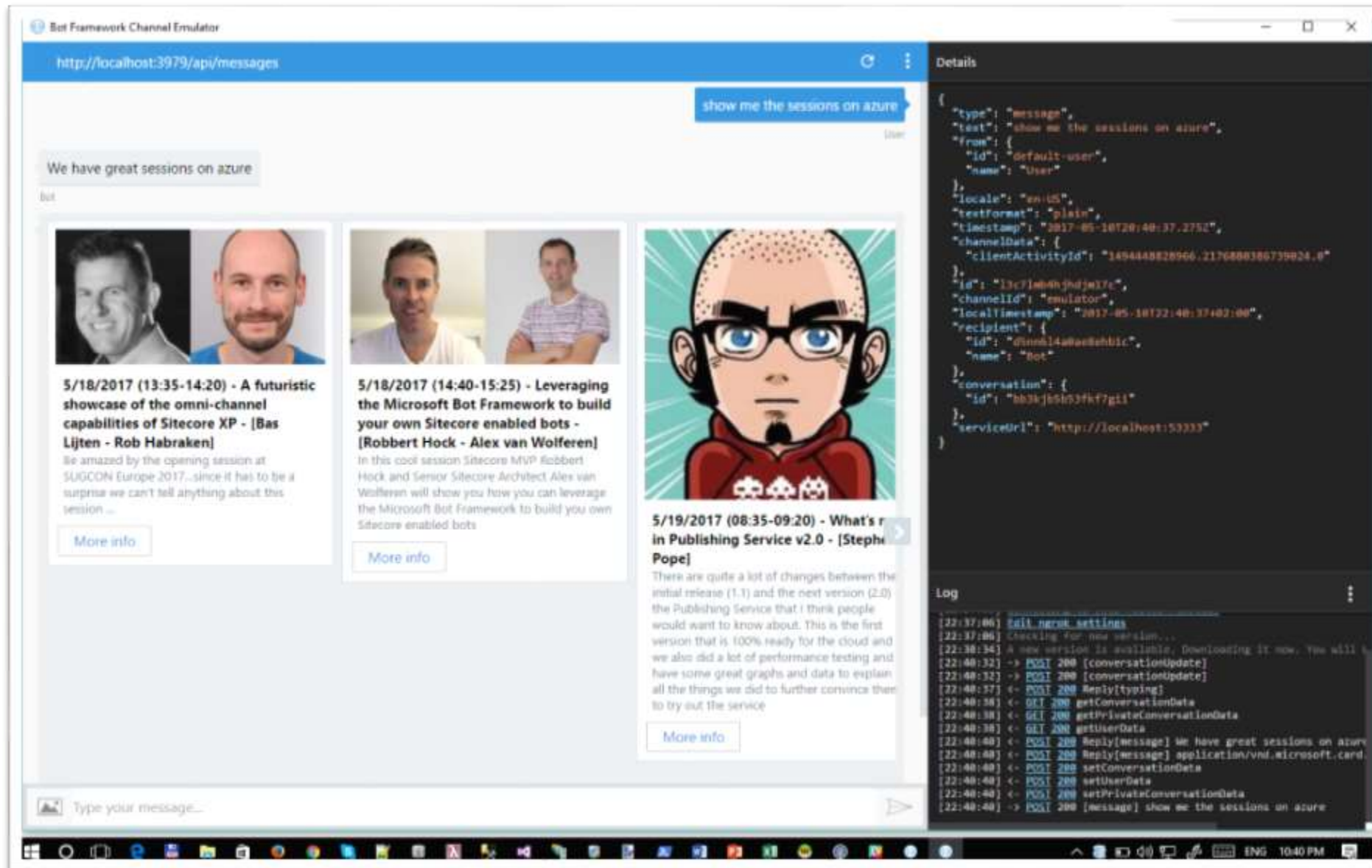
Learn about the different ways you can deploy bots.



### Reference

Explore the Bot Framework reference documentation.

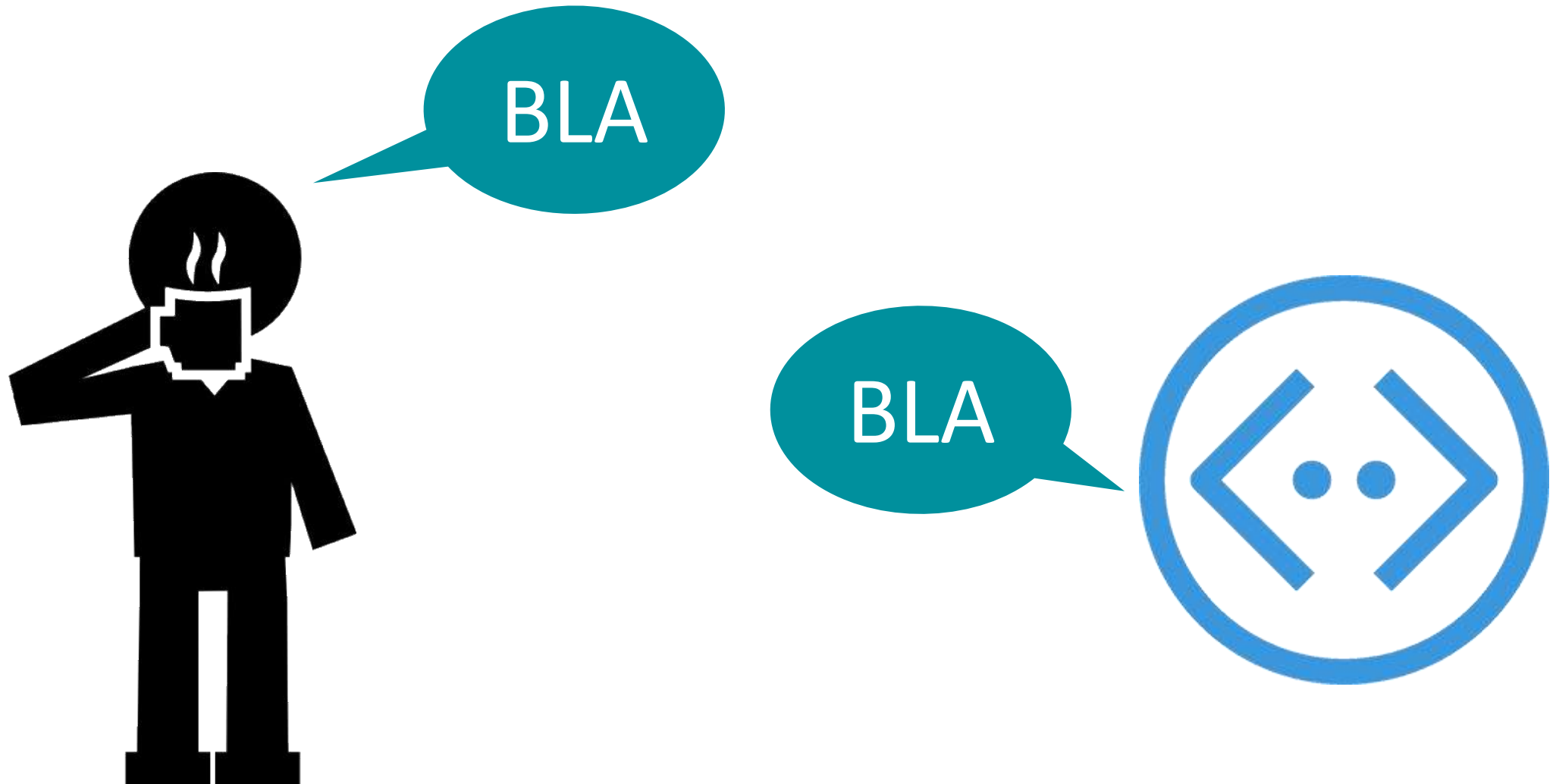






- Microsoft Bot Builder SDK
- Messages API that handles activities
  - **Message**
  - Contact Relation Update
  - Conversation Update
  - Typing
  - Ping
  - End of Conversation
  - Trigger
  - Event
  - Invoke
  - Delete User Data

- Dialogs are the key of success to Bots
- Break up conversations into smaller pieces
- Follow the Single Responsibility Principle
- Must implement the IDialog interface
  - Task StartAsync(IDialogContext context)
- Located in the Microsoft.Bot.Builder.Dialogs namespace
- Must be serializable



# SUGCON Bot

# Introducing the SUGCON Europe 2017 bot

SUGCON  
EUROPE 2017

## WebChat

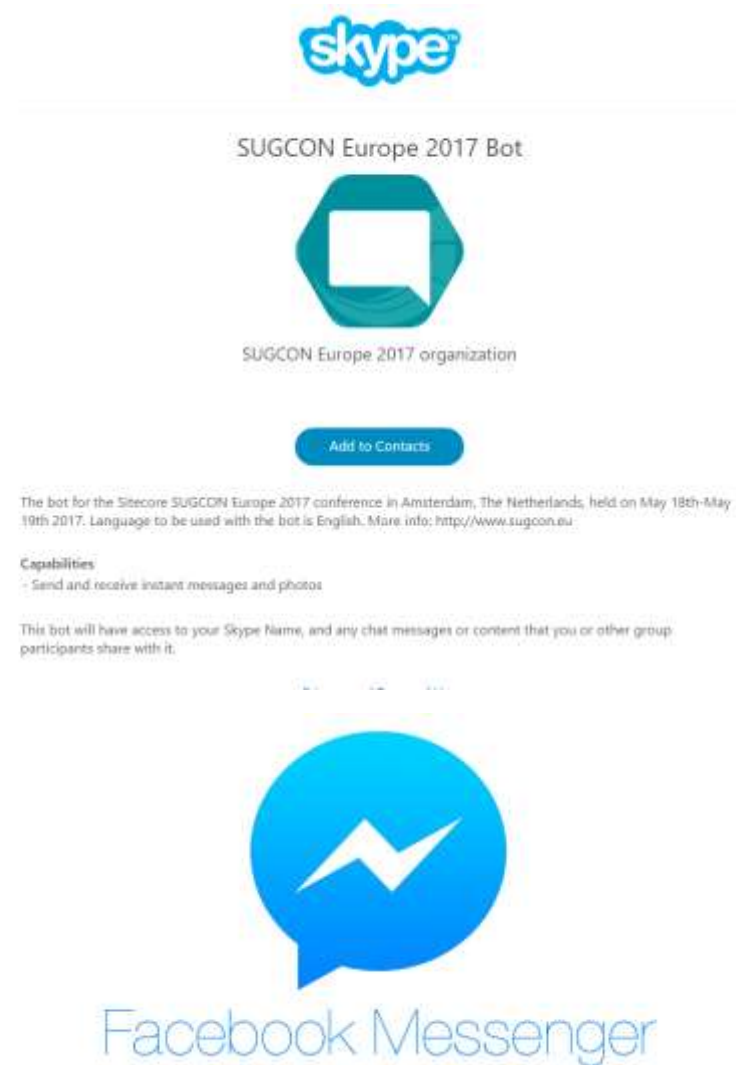
<http://www.sugcon.eu/bot/>

## Skype

<https://join.skype.com/bot/12of1168-77bd-48f3-8baa-c78d81aba458>

## Facebook Messenger






<https://www.messenger.com/t/1208427272620146>







[microsoft.com/cognitive](https://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



SUGCON

Event bot

Version: 0.1

Settings

Dashboard

Intents

Entities

Features

Train & Test

Publish App

← Back to App list

## Overview

Facts & statistics about the app's data and the received endpoint requests at any period of time ... [Learn more](#)

Dashboard Suggested utterances

App Id: 07b06373-9949-4b5e-b7a5-a3397fefb754

You don't have to create all your entities from scratch. Check out a list of prebuilt entities for common knowledge concepts (e.g. date & time, numbers, temperature, percentage, etc.). Consider using these ready-made entities in your model rather than creating them by hand.

Add prebuilt entities

Next tip

App status

Last train: May 7, 2017, 10:41:30 PM

Last published: May 7, 2017, 10:09:56 PM

Intent Count

12 / 80

Entity Count

9 / 30

List Entity Count

0 / 50

Labeled Utterances Count

396

LUIS.ai



# Microsoft Bot Framework

- Create a conversational Bot that integrates with the whole Experience Platform from a visitor perspective
- Challenges?
  - Create demo site with reasonable interaction needs
  - 8.2 u3 didn't have xConnect
  - Way too time consuming to create required stack to replace xConnect
  - Create several API's



- Create conversational Bot that communicates with Sitecore from a operational perspective
- Challenges?
  - Create a authentication mechanism between the Bot and Sitecore
  - Create several API's
    - Authentication, UserManagement, Indexing, Publishing, XDB, Logging

- Create a multi dialog conversational Bot (CaaP)
- Authenticate Bot user with Sitecore
- Create several API's for access to Sitecore functionality
- Make use of LUIS for understanding what the user wants and to break up the intents
- Make use of FormFlow for conversation guidance
- Make use of Application Insights REST Api

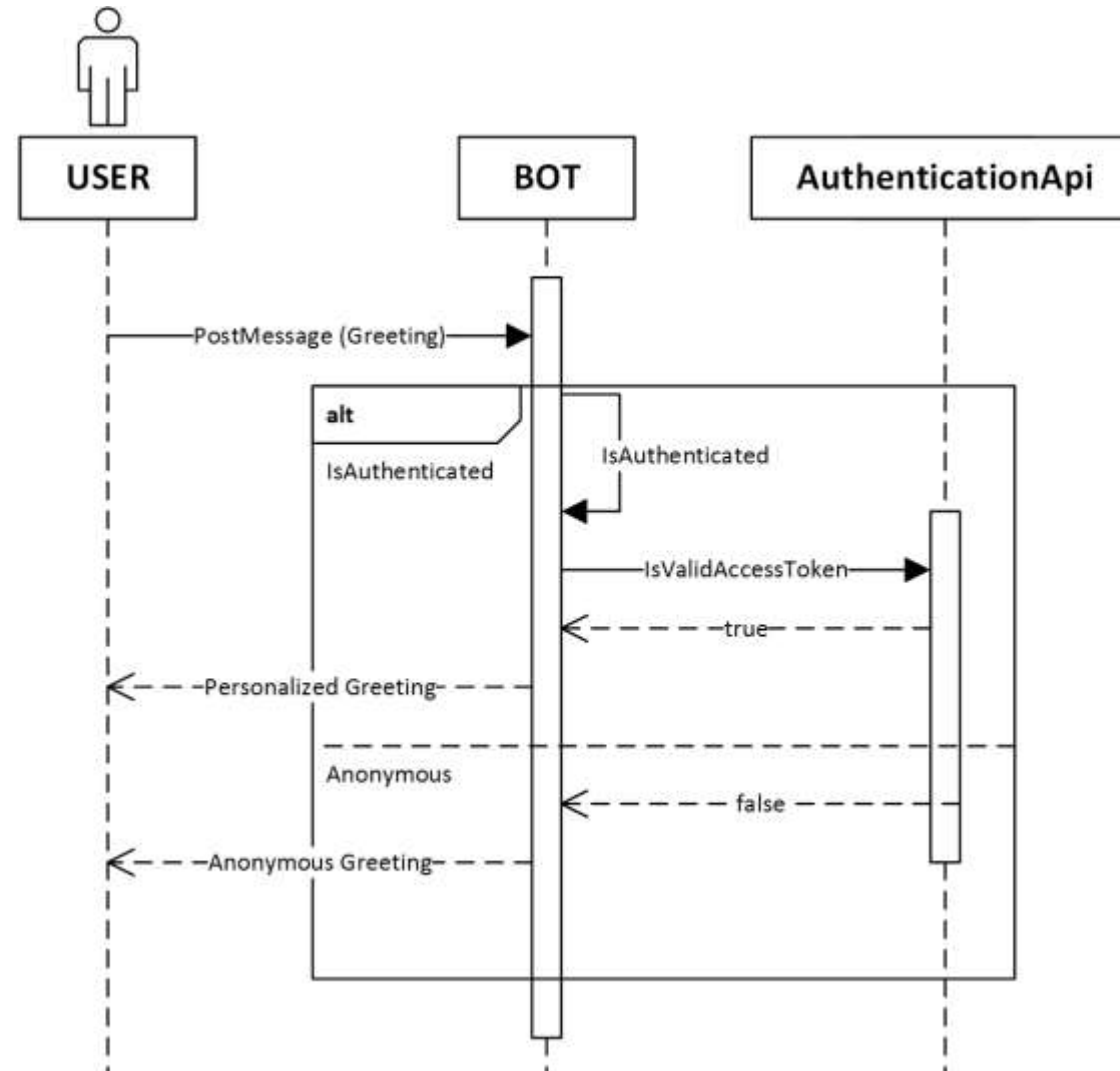
- LUIS Model
  - 19 intents
  - 6 entities
  - 105 labeled utterances
- Multi Dialog
  - Authentication
  - User Management
  - xDB
  - Publishing
  - Indexing
  - Application Insights

# Authentication with Sitecore

- User in conversation needs to be authenticated
- IDialogContext (IBotData)
  - Get/SetUserData
  - Get/SetConversationData
  - **Get/SetPrivateConversationData**
  - DeleteStateForUser



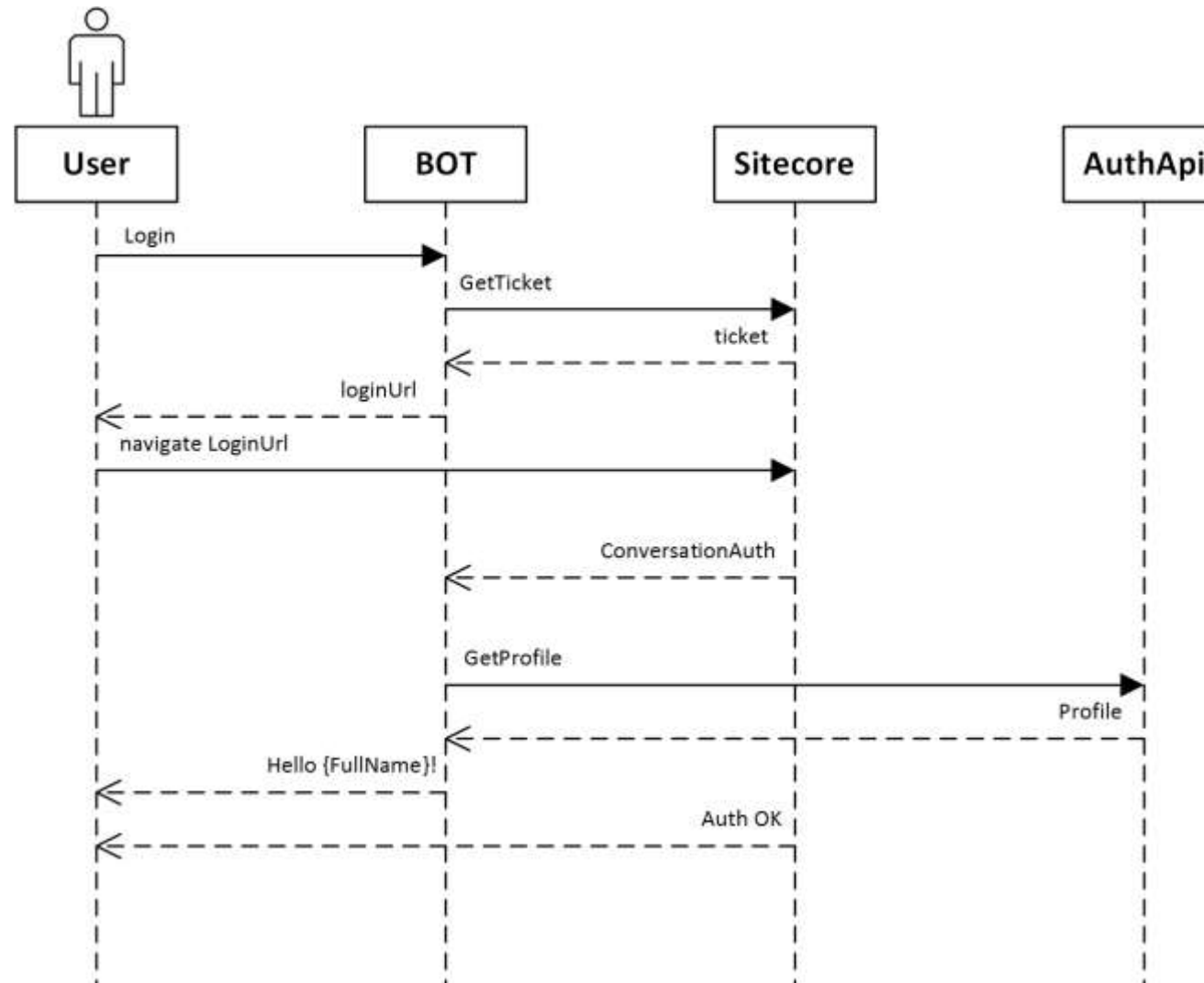
# Authentication with Sitecore from Bot



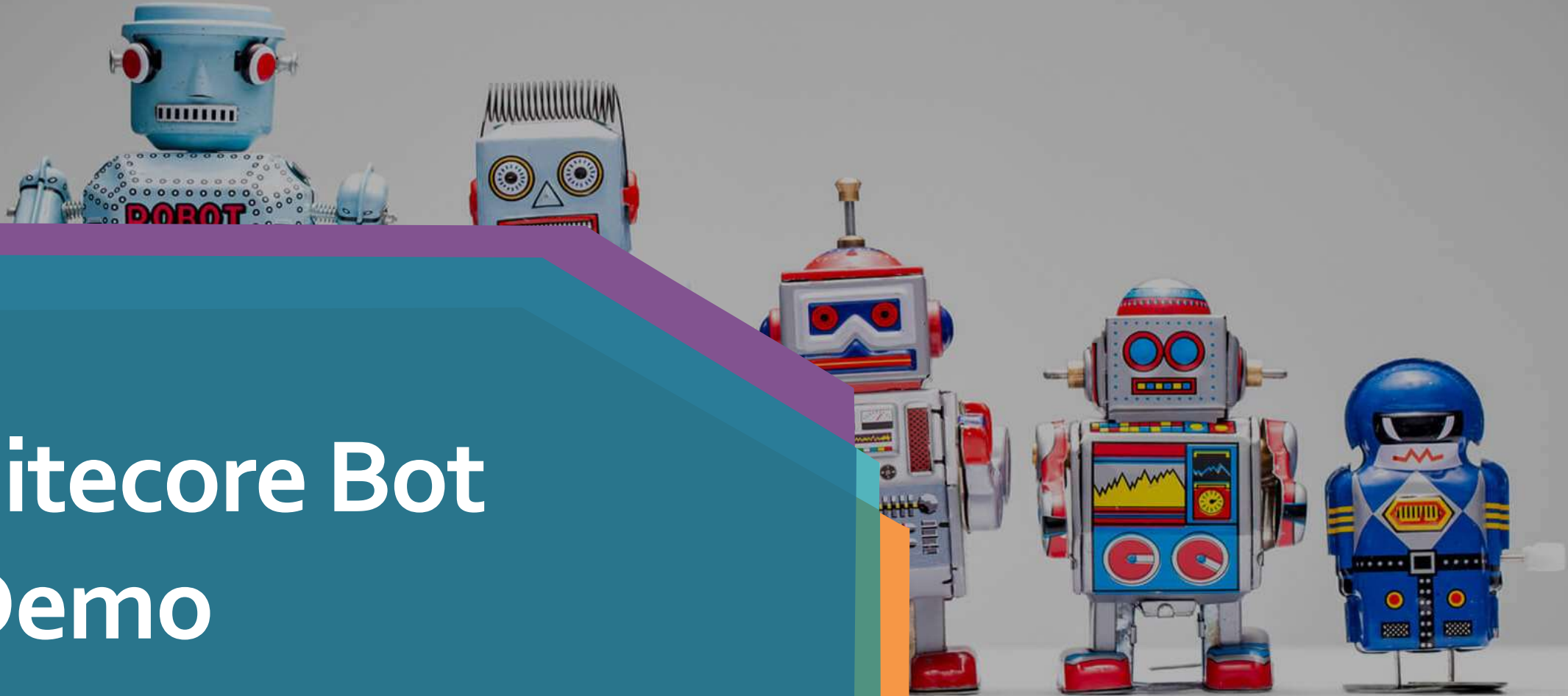
1. AccessToken InValid
2. Forward conversation to AuthenticationDialog
3. Reply message and call Login
4. Trade conversationreference for a login ticket with API build in Sitecore environment
5. Redirect the user from conversation to Bot Authentication form in Sitecore environment
6. Authenticate the user
7. Perform a callback from Sitecore environment to dedidated API in Bot
8. Authenticate the user in the conversation (SetPrivateConversationData)
9. User is authenticated for a certain amount of time; User can logout;

Keep calm and Bot on!

# Authentication with Sitecore from Bot



# Sitecore Bot Demo



# FormFlow

Natural guidance and help...  
More options with less effort!

- Provides clear guidance and help
- Understands both numbers and textual entries
- Gives feedback on what is understood and not
- Asks clarifying questions when needed
- Allows navigating between steps
- Messages during the process of filling in a form.
- Custom prompts per field.
- Templates to use when automatically generating prompts or help.
- Terms to match on.
- Whether to show choices and numbers or not.
- Fields that are optional.
- Conditional fields.
- Value validation
- Binds to a model

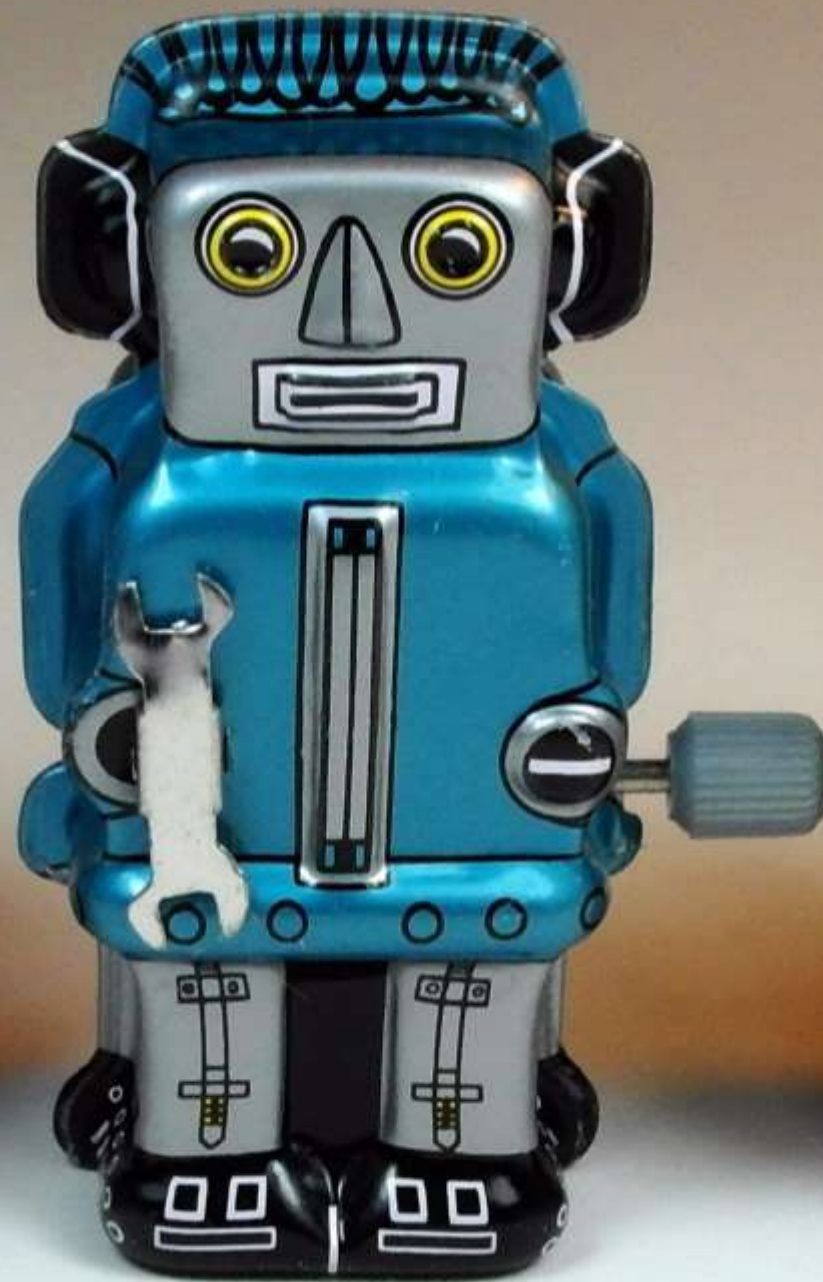
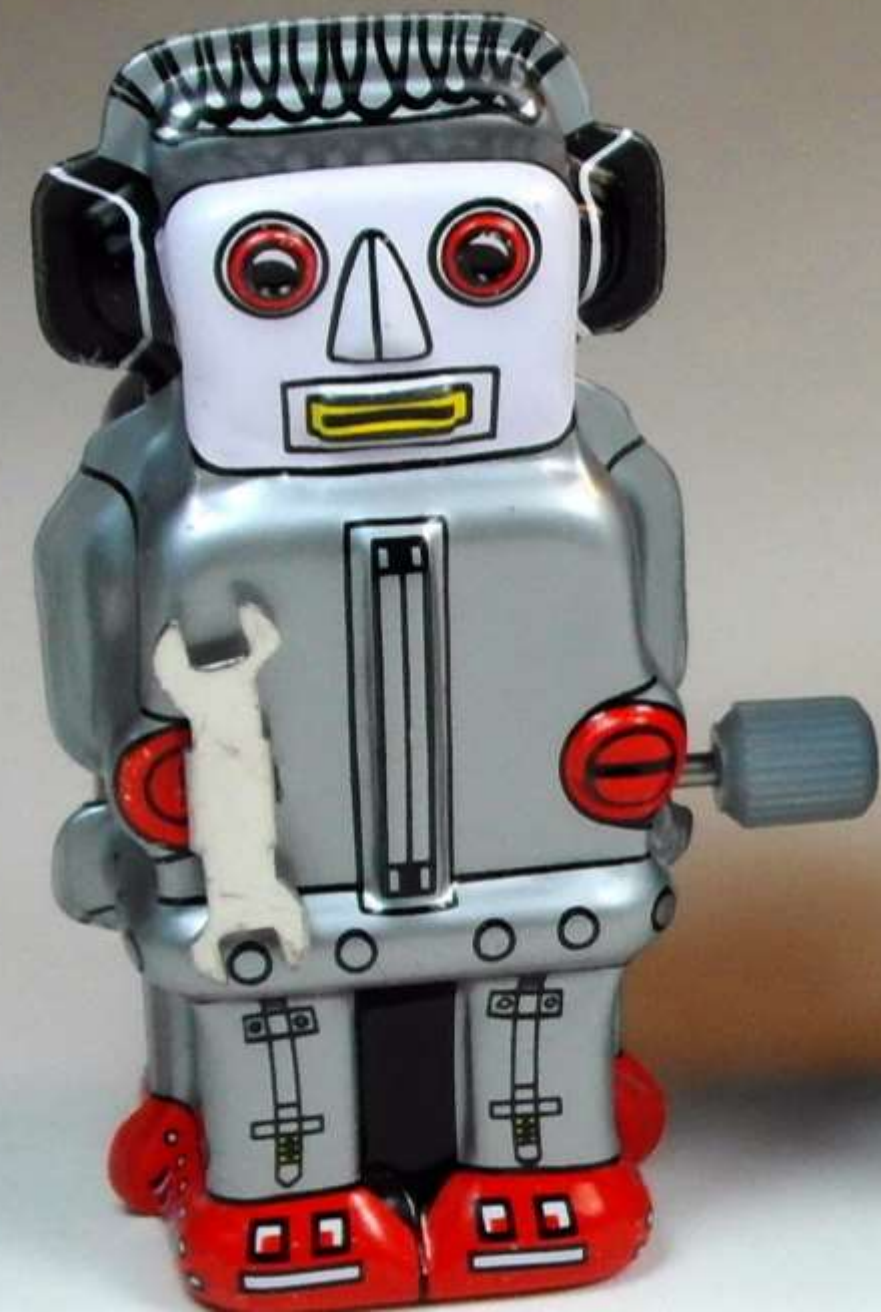


```
[Serializable]
public class CreateUser
{
    [Pattern(@"^([\w-]+(?:\.[\w-]+)*)@((?![\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$")]
    public string EmailAddress;
    public string FullName;
    [Pattern(@"^[_a-zA-Z0-9]+$")]
    public string Username;
    public List<UserRole> UserRoles;
    public bool? AdministratorRoleForUser;
    public bool? EmailSendWithPasswordToTheUser;


    public static IForm<CreateUser> BuildForm()
    {
        var form = new FormBuilder<CreateUser>()
            .Message("Let's get some more details for the user!");

        form.Configuration.DefaultPrompt.ChoiceStyle = ChoiceStyleOptions.Buttons;

        return form.Build();
    }
}
```



## New – Cortana, Bing and Skype for Business Channels



Custom Skills

---

Memory / Profile

---

Speech




Business Bots

---

Top Bots

---

Bots in Bing Search



LOB Bots

---

IT Managed

---

Integrated Auth






## Adaptive Cards



OPEN FRAMEWORK  
MULTIPLE CANVASES

Card created: Publish Adaptive Card schema

 **Miguel Garcia**  
Created Monday, February 13, 2017 10:08:39 ...

Now that we have define the main rules and features of the format, we need to produce a schema and ...

Board: Adaptive Card  
List: Backlog  
Assigned to: David Claux  
Due date: Not set

Set due da... Comment View

Card created: Publish Adaptive Card schema


 **Miguel Garcia**  
Created Monday, February 13, 2017 10:08:39 PM

Now that we have define the main rules and features of the format, we need to produce a schema and publish it to GitHub. The schema will be the starting point of our reference documentation.

Board: Adaptive Card  
List: Backlog  
Assigned to: David Claux  
Due date: Not set

Set due date Comment View

Card created: Publish Adaptive Card schema


 **Miguel Garcia**  
Created Monday, February 13, 2017 10:...

Now that we have define the main rules and features of the format, we need to produce a ...

Board: Adaptive Card  
List: Backlog  
Assigned to: David Claux  
Due date: Not set

Set due date  
Comment  
View

Card Created: Publish...

 **Miguel Garcia**  
Created Monday, February 13, 2017

Now that we have define the main rules and features of the format, we need to produce a schema and publish it to...

Board: Adaptive Card  
List: Backlog  
Assigned to: David Claux  
Due date: Not set

Card Created: Publish...

 **Miguel Garcia**  
Created Monday, February 13, 2017

Now that we have define the main rules and features of the format, we need to produce a schema and publish it to...

Board: Adaptive Card  
List: Backlog  
Assigned to: David Claux  
Due date: Not set

Notification

Microsoft Teams

Skype

Android

iOS

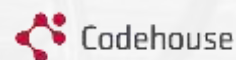
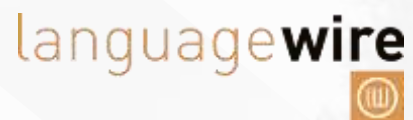


Mark Stiles

Sitecore MVP & Principal  
Developer - @Velir

@maaakstiles





# Thank you!

Coontact us at:

@kayeeNL

@avwolferen