



Bot Framework

Eremin Maxim
Moscow Aviation Institute
Microsoft Student Partner
Comparex LLC.

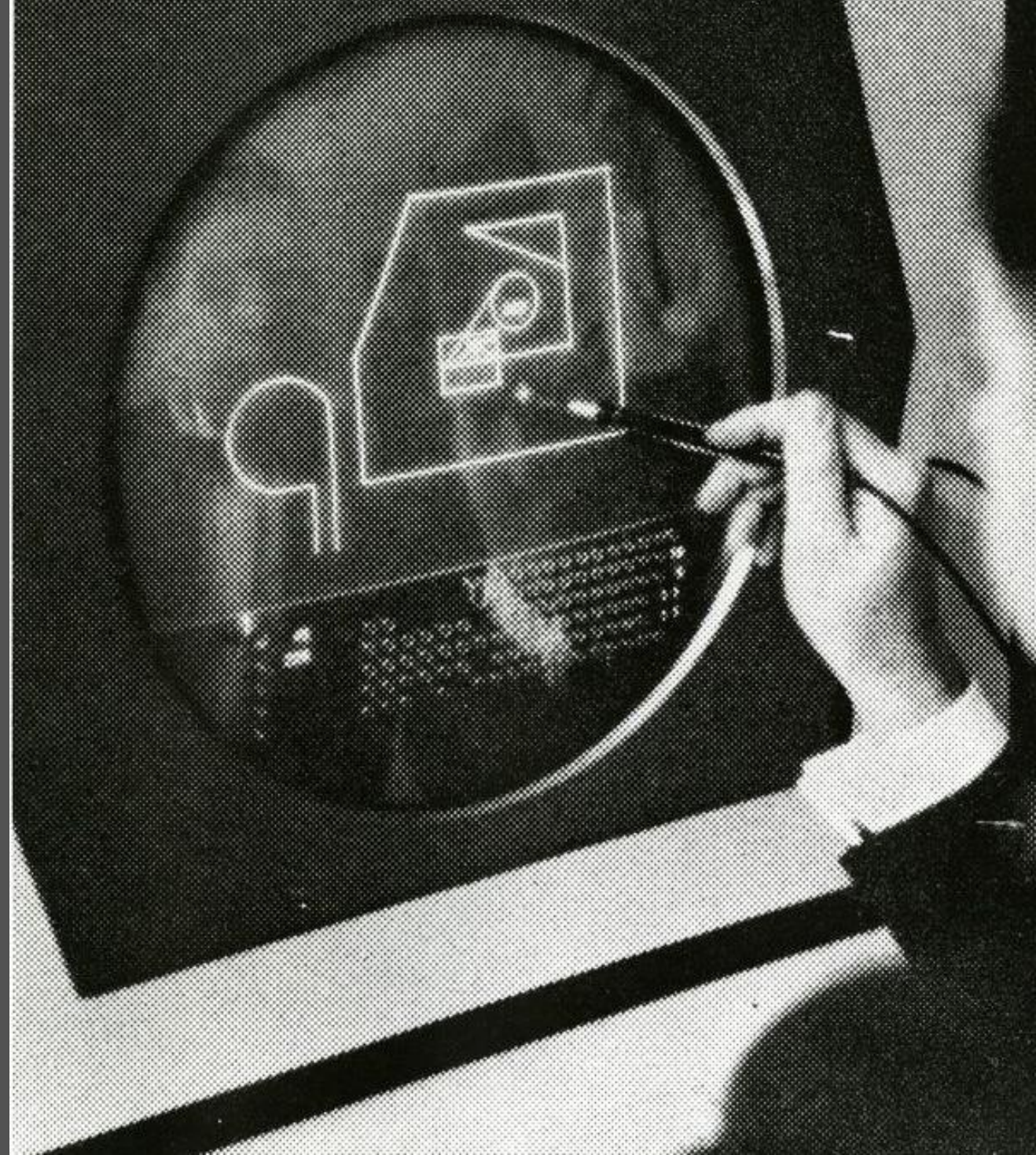


Agenda

- Эволюция интерфейсов
- Введение в Bot Framework
- AdaptiveCards
- LUIS
- QnA
- Демонстрация умного бота

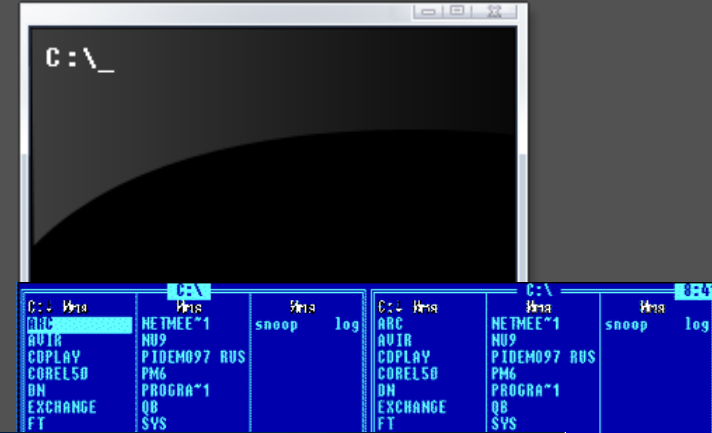


Эволюция интерфейсов: от командной строки до чат-ботов

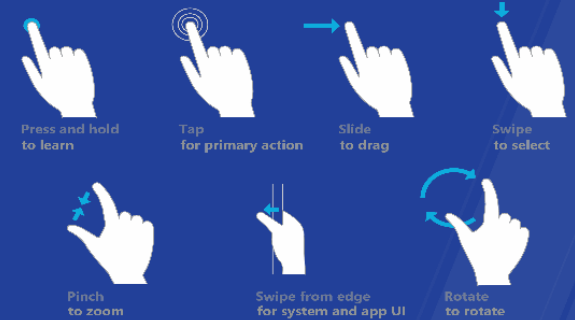


Общение человек-компьютер

- Командная строка
- Оболочки (текстовые и графические)
- Графическая ОС
- Touch-интерфейс

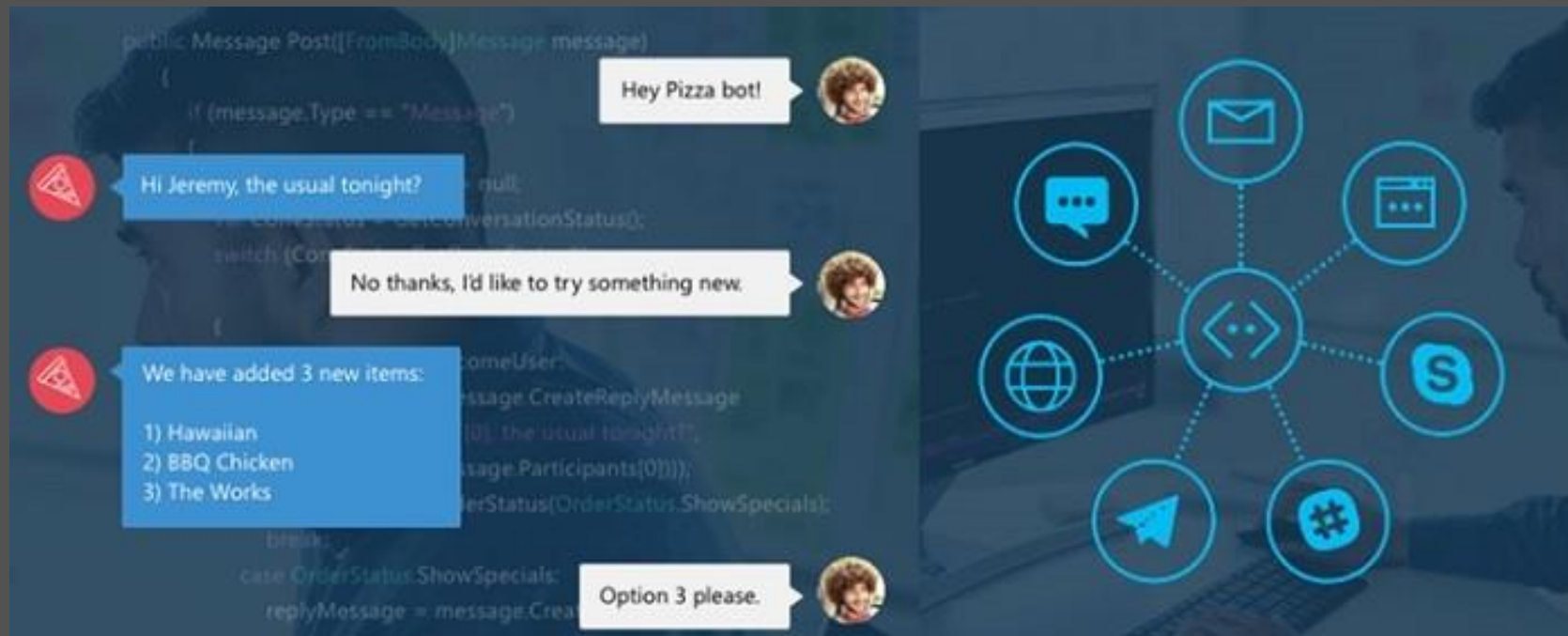


Windows 8 Touch Language



Кто такие чат-боты?

Некая искусственная сущность, с которой можно общаться на человеческом языке посредством мессенджеров.



Какие задачи решают чат-боты

- Прием заказов и продажа товаров
- Бот-ассистент
- Консультация и поддержка клиентов (рекомендательные системы)
- Entertainment
- Информационные службы
- Экспертные системы



Преимущества общения с ботом

- Естественный для человека язык
- Возможность общения с ботом конференцией
- Среда «человек-компьютер» (заказ пиццы с персональным ассистентом)
- Возможность голосового сообщения
- Независимость от канала общения (Skype, Telegram, Facebook Messenger, Slack, SMS, Cortana)

Преимущества в бизнесе

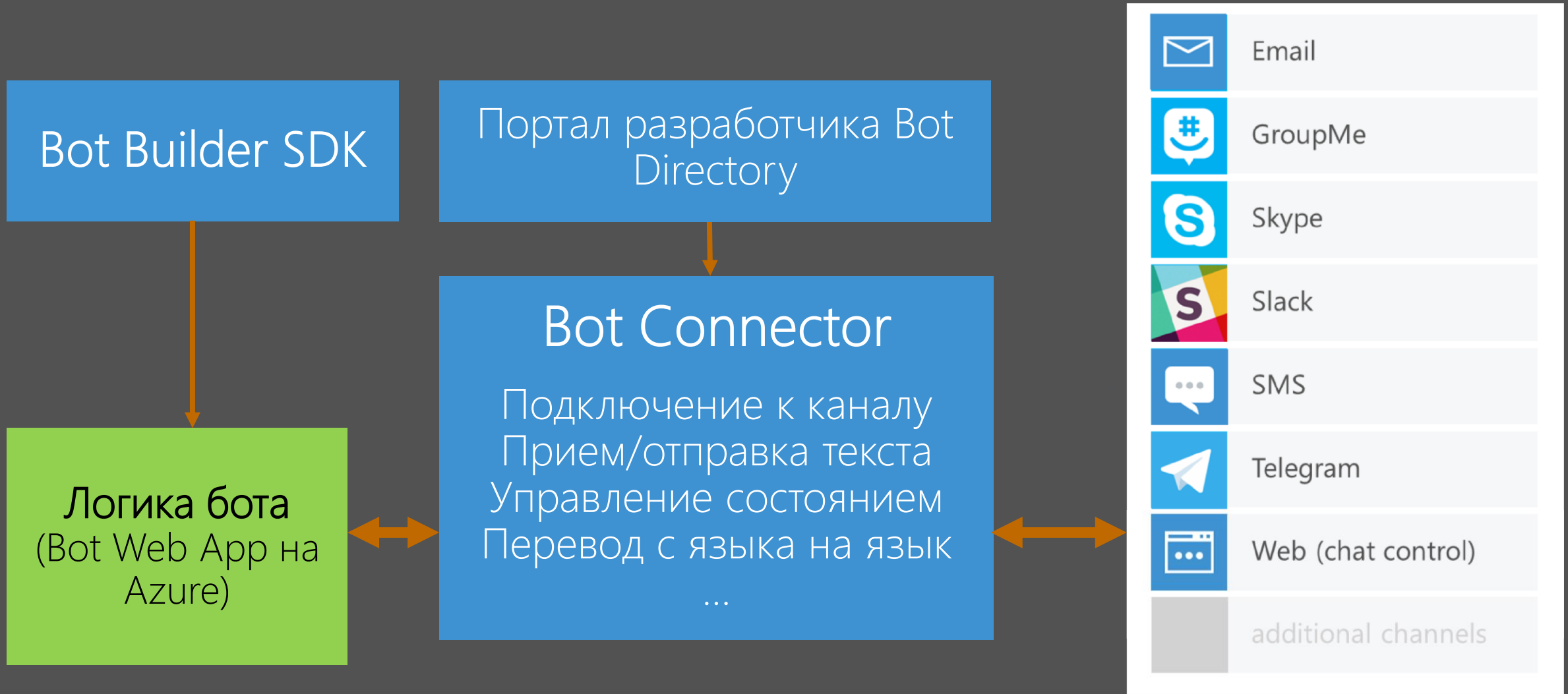
- Увеличение нагрузки на фирму без найма дополнительного штата менеджеров и операторов
- Увеличение количества клиентов, обслуженных фирмой
- Повышение продаж за счет автоматизации приема заказов и консультаций



Введение в Bot Framework



Microsoft Bot Framework



Bot Framework

The logo for JavaScript, featuring the letters "JS" in a bold, black, sans-serif font. The "J" and "S" are connected, with the "S" having a thick, rounded tail. The logo is centered on a solid yellow square background.

JS

The logo for .NET, featuring the text ".NET" in a blue, sans-serif font. The period is a small dot to the left of the "N". The logo is centered on a white square background.

.NET



Bot Framework



Skyscanner

Skyscanner

The world's travel search engine



AzureBot

Microsoft

Be more productive with your own Microsoft Azure subscriptions using



Hipmunk

Hipmunk, Inc.

Hi, hi! I'm a super-powered travel assistant Hipmunk who can dig



Jessie Humani

PullString, Inc.

Jessie Humani's Will you help h



SI x GameOn ...

GameOn Technology

Love the NFL? Add the SI x GameOn NFL bot to get unparalleled



StubHub

StubHub Inc.

Finding tickets and theater as



Baymax

TrueCare24

Your personal healthcare companion. The service is available



Cardea

Powered by RingMD

Meet Cardea, your personal medical aid. She will answer your health



Solitaire

Card Lion

Solitaire is the most popular card game in the world.



UNO

Card Lion

UNO world famous card game bot



Trivia Master

Microsoft

A fiendish quiz hosted by Professor T. Master.



RemitRadar

RemitRadar

Artificial Intelligence remittance assistant chatbot. Helps you to find

Useful links

Template:

<http://aka.ms/bf-bc-vstemplate>

← → ▾ ↑ 📁 > Этот компьютер > Документы > Visual Studio 2017 > Templates > ProjectTemplates > Visual C# >

Emulator:

<https://github.com/Microsoft/BotFramework-Emulator>

Основные понятия



Что нужно знать?

- Основы программирования
- C# / Node.js
- REST API
- Асинхронное программирование
- λ -выражения

REST API

- REST (Representational state transfer) – это стиль архитектуры программного обеспечения для распределенных систем, используется для построения веб-служб.
- Каждая единица информации однозначно определяется глобальным идентификатором – URL

REST API

GET /info/ - список всех объектов

GET /info/{id} - объект

POST /info/ - создать объект

PUT /info/ - создать объект

POST /info/{id} - изменить объект

PUT /info/{id} - изменить объект

DELETE /info/{id} - удалить объект

Асинхронное программирование

- Проблема: Есть какое-то действие, которое требует определенного времени на выполнение, но при этом время ожидания критично для синхронного выполнения.

Например, ожидание ответа на запрос к серверу может выполняться очень долго, блокируя при этом выполнение основного потока.

- Решение: сделать запросы к удаленному серверу асинхронными, не блокируя основного потока программы.

Асинхронное программирование

Task, Task<T>

async

await

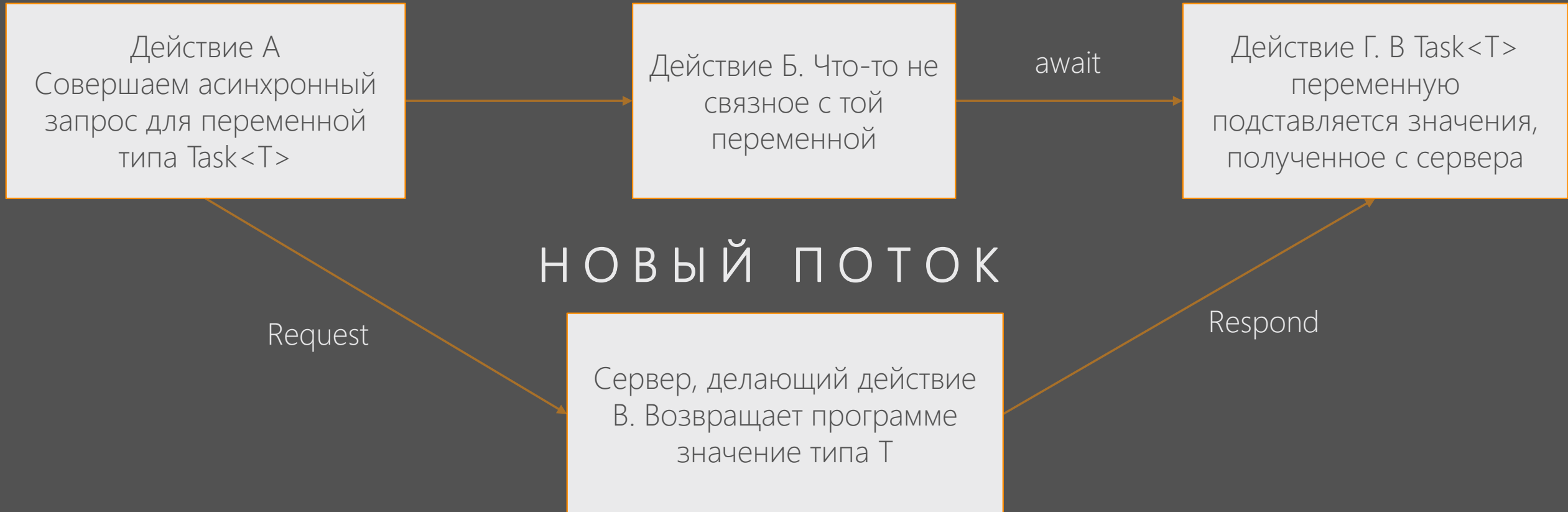
Асинхронное программирование

Логика асинхронных методов:

1. Вызывается `async` метод. Если метод заканчивается до `await`, то результат достаётся синхронно, из метода возвращается *завершённый Task*.
2. Вызывается `async` метод. Выполнение доходит до последнего вызова `await`. Из метода возвращается *завершённый Task*.
3. Работу `async` метода можно прервать до его логического завершения, например по таймауту.

Асинхронное программирование

ОСНОВНОЙ ПОТОК



Лямбда-выражения

Лямбда-выражение — это анонимная функция, с помощью которой можно писать локальные функции, которые можно передавать в качестве аргументов или возвращать в качестве значений из вызовов функций

Если проще — это методы без названий

Нужны для того, чтобы сделать мелкие действия без создания громоздких методов, засоряющих код

Лямбда-выражения

Лямбда-оператор - \Rightarrow

Слева — входные параметры

Справа — блок выражений или операторов

Параметр

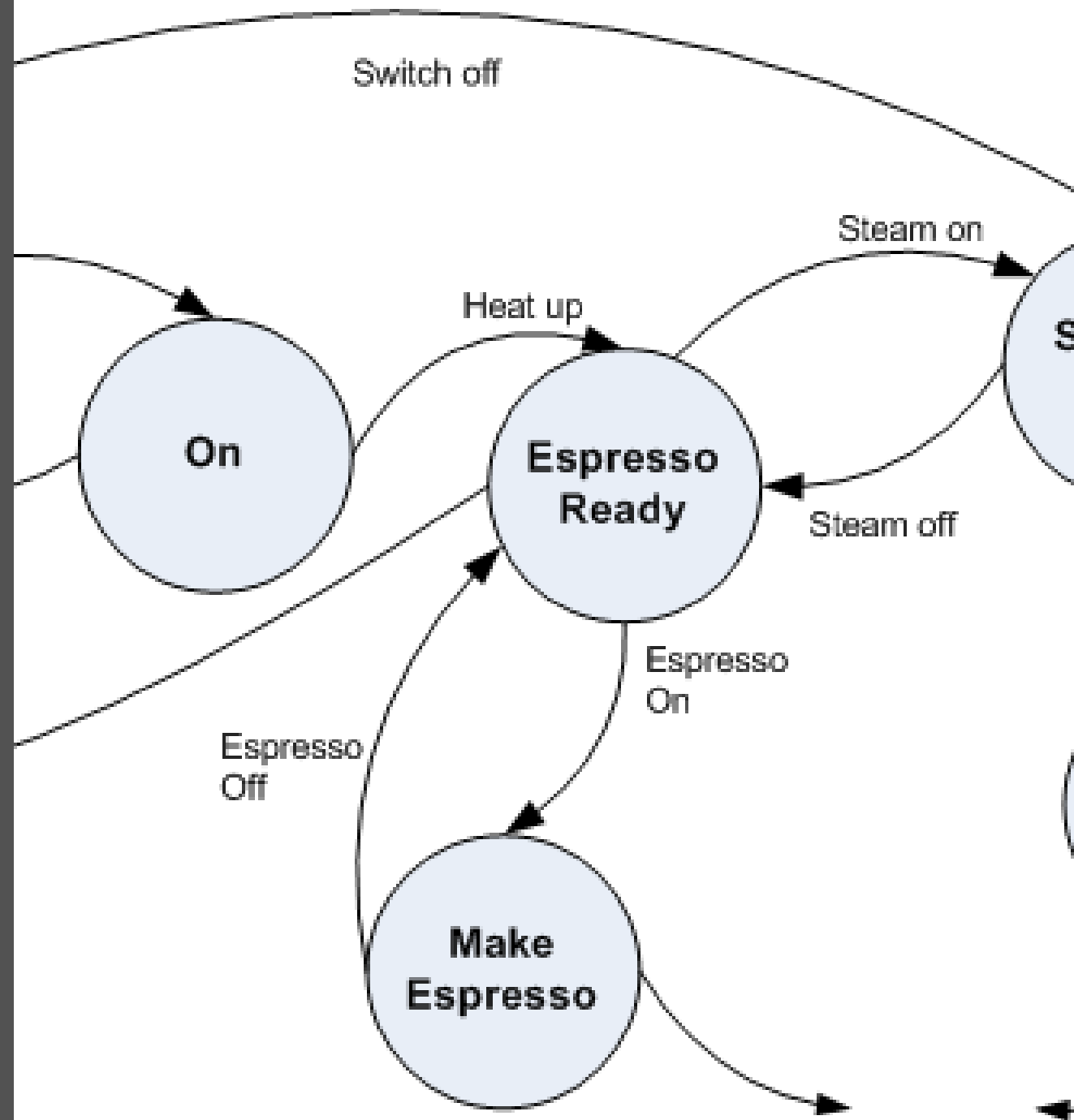
$x \Rightarrow x * x$

Выражение

Лямбда-выражения

```
await Conversation  
  .SendAsync(activity, () => new Dialog());
```


Bot State



Bot State

- Состояние бота на момент ответа
- Сохраняем каждый раз состояние, если наши данные, полученные от бота или переданные ему, изменяются.
- Загружаем состояние.

Bot State

```
var State = activity.GetStateClient();  
var UserData = await State.BotState.  
GetUserDataAsync(activity.ChannelId, activity.From.Id);  
var x = UserData.GetProperty<PropertyType>("property");  
var reply = await Reply(activity.Text);  
Activity replyActivity = activity.CreateReply(reply);  
UserData.SetProperty<PropertyType>("property", Prop);
```

IDialog<object>

Chat



Congratulations! The number to guess was 6. You needed 6 attempts. Would you like to play again?

Yes

No

No



Goodbye!

Hello



Hi Welcome! - Guess a number between 1 and 10

IDialog<object>

- Обертка сохранения состояний.
- Автоматически сохраняем состояния и запоминаем момент диалога на уровне данных.
- Весь диалог происходит ОБЯЗАТЕЛЬНО в отдельном классе, наследованном от IDialog<T>

IDialog<object>

```
[Serializable]
public class RootDialog : IDialog<object>
{
    public Task StartAsync(IDialogContext context)
    {
        context.Wait(MessageReceivedAsync);
        return Task.CompletedTask;
    }
    private async Task MessageReceivedAsync(IDialogContext context, IAwaitable<IActivity>
result)
    {
        Activity input = await result as Activity;
        context.Wait(MessageReceivedAsync);
    }
}
```

IDialog<object> | context

```
Context.Call(IDialog<R> child, ResumeAfter<R> resume);  
Context.Done(this);  
Context.Forward(IDialog<R> child, ResumeAfter<R> resume, T  
item, CancellationToken);  
Context.PostAsync(T item);  
Context.Wait(...);
```


Adaptive Cards



Adaptive Cards

- Возможность отправлять пользователям сообщения с активным наполнением.
- Кнопки, аудио, видео, размеченные сообщения с медиа-контентом
- Возможность вставлять превью ссылок на сторонние ресурсы

Adaptive Cards



Adaptive Cards

- Hero Card
- Receipt Card
- Signing Card
- Animation Card
- Video Card
- Audio Card

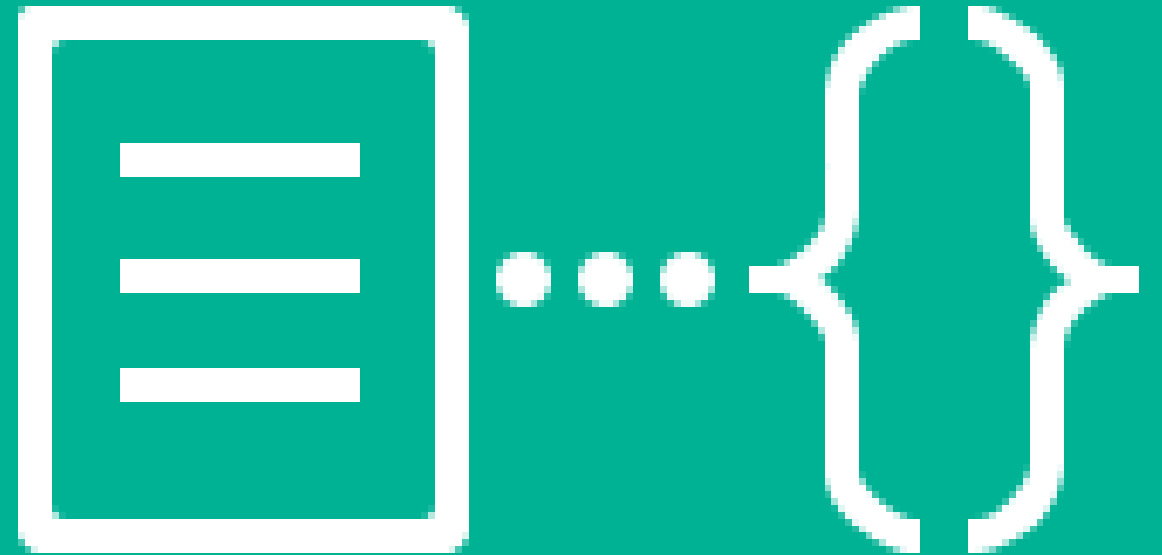
Adaptive Cards

```
Public async Task MessageReceivedAsync(IDialogContext context,  
                                       Iawaitable<IActivity> result)  
{  
    IMessageActivity message = context.MakeMessage();  
    Attachment attachment = new HeroCard().ToAttachment()  
    await context.PostAsync(message);  
}
```

Adaptive Cards

```
HeroCard = new HeroCard
{
    Title = "HeroCard Sample",
    Subtitle = "Good capmle",
    Text = "Very good text",
    Images = new List<CardImage> {
        new CardImage("url")
    },
    Buttons = new List<CardAction> {
        new CardAction(ActionTypes.MessageBack, "GetCard", null, null,
            "Card", "Card")
    }
}
```

LUIS

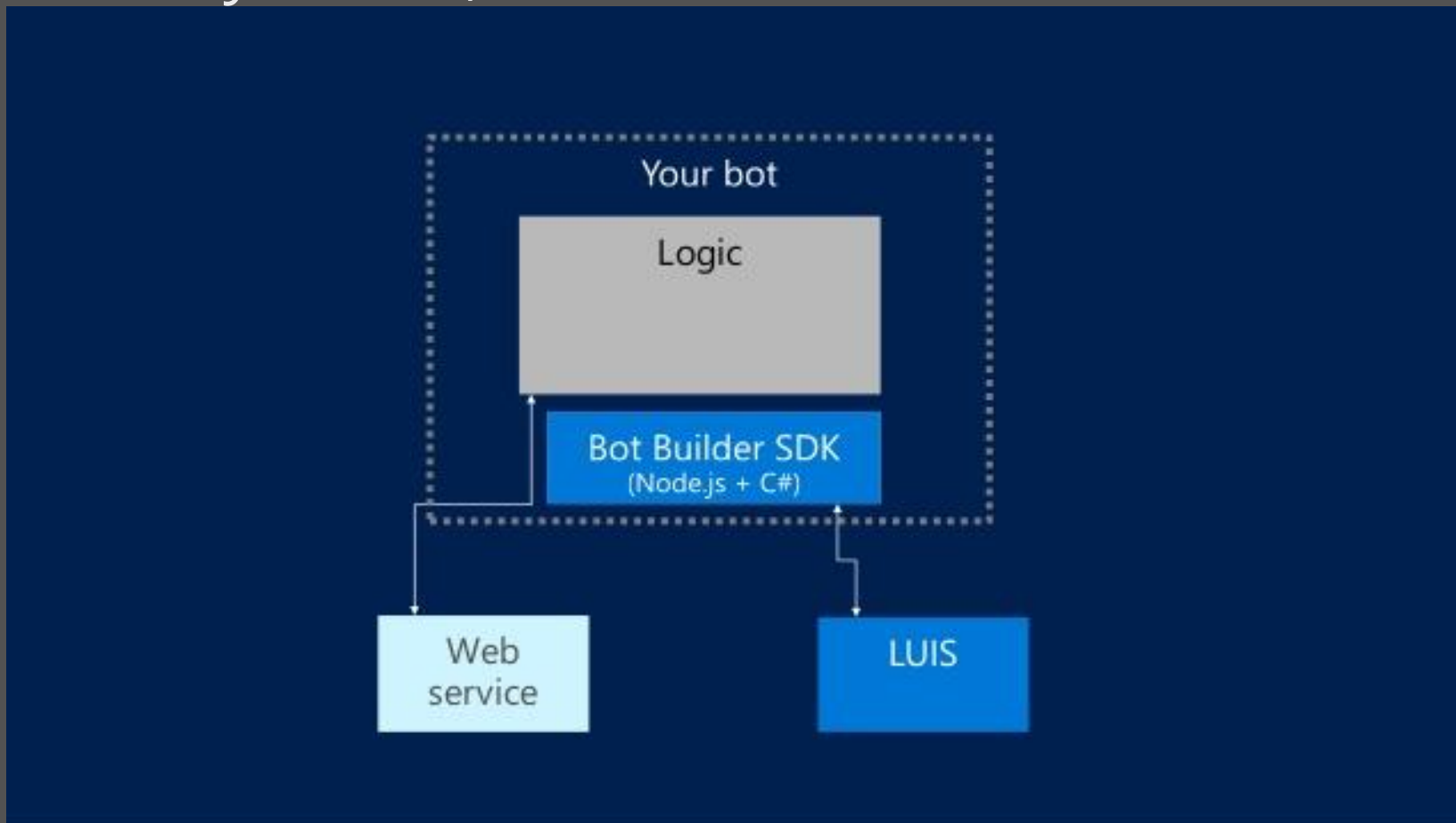


LUIS

LUIS (Language Understanding Intelligent Service) – это когнитивный сервис, который позволяет выделять намерения и некоторые ключевые сущности из коротких предложений на естественном языке.

Зачем нужен LUIS?

Научите вашего бота по-настоящему естественному общению.



LUIS

- LUIS – Language Understanding Intelligence Service
- Служба машинного обучения для распознавания естественного языка
- API для обучения бота простым (а может даже и сложным) наборам возможных путей диалога

Плюсы:

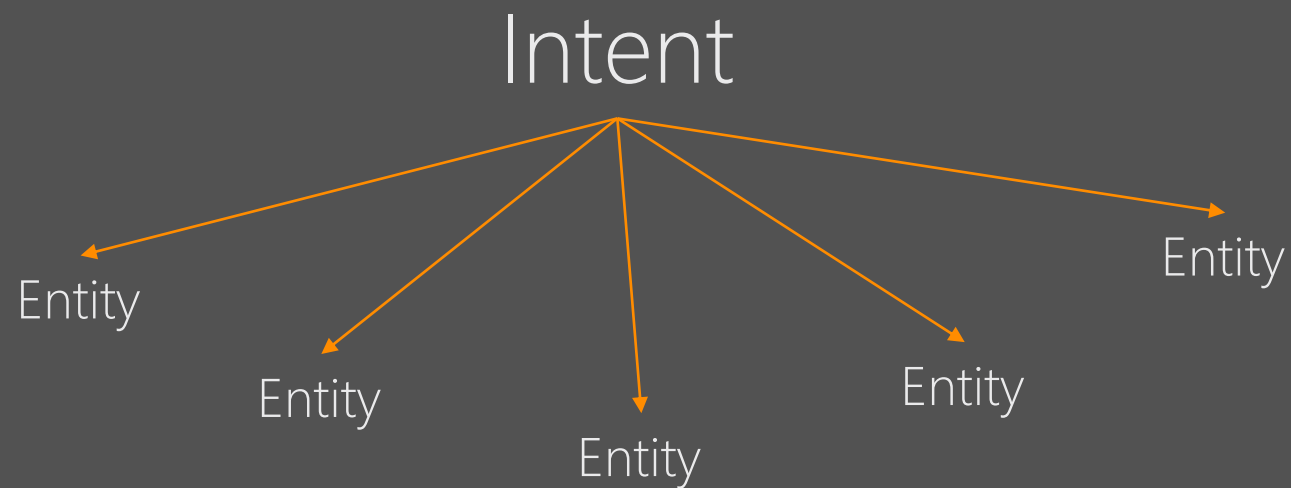
- Распознавание языка
- Есть готовые наборы словарей
- Масштабирование
- Все хранится в облаке

Минусы:

- Нет русского языка
- Всегда требуется сеть

LUIS

- Intent – намерение.
- Entity – сущность.



Intents and Entities

Intents ?

Create new intent

Add prebuilt intent

Search intents

Name
Hello
None

Utt Hello

Delete Intent

0

Type about 5 examples of what a user might say to trigger this task and hit Enter.

0

Search for an utterance

Reassi...

Delete utterance(s)

Filters: ☐ Errors ☒ Entity ☒ Entities view ☒ Fuzzy search

<input type="checkbox"/> Utterance	Labeled intent ?	
<input type="checkbox"/> what ' s up	Hello -1	...
<input type="checkbox"/> hey buddy	Hello -1	...
<input type="checkbox"/> yo	Hello -1	...
<input type="checkbox"/> hey	Hello -1	...

LUIS B Bot Framework

```
[Serializable]
[LuisModel("", "")]
Public class LuisDialog : LuisDialog<object>
{
    [LuisIntent("")]
    public async Task ProcessNone(IDialogContext context, LuisResult result)
    {
        await context.PostAsync("I do not understand");
        context.Wait(MessageReceived);
    }
}
```

Пример запроса

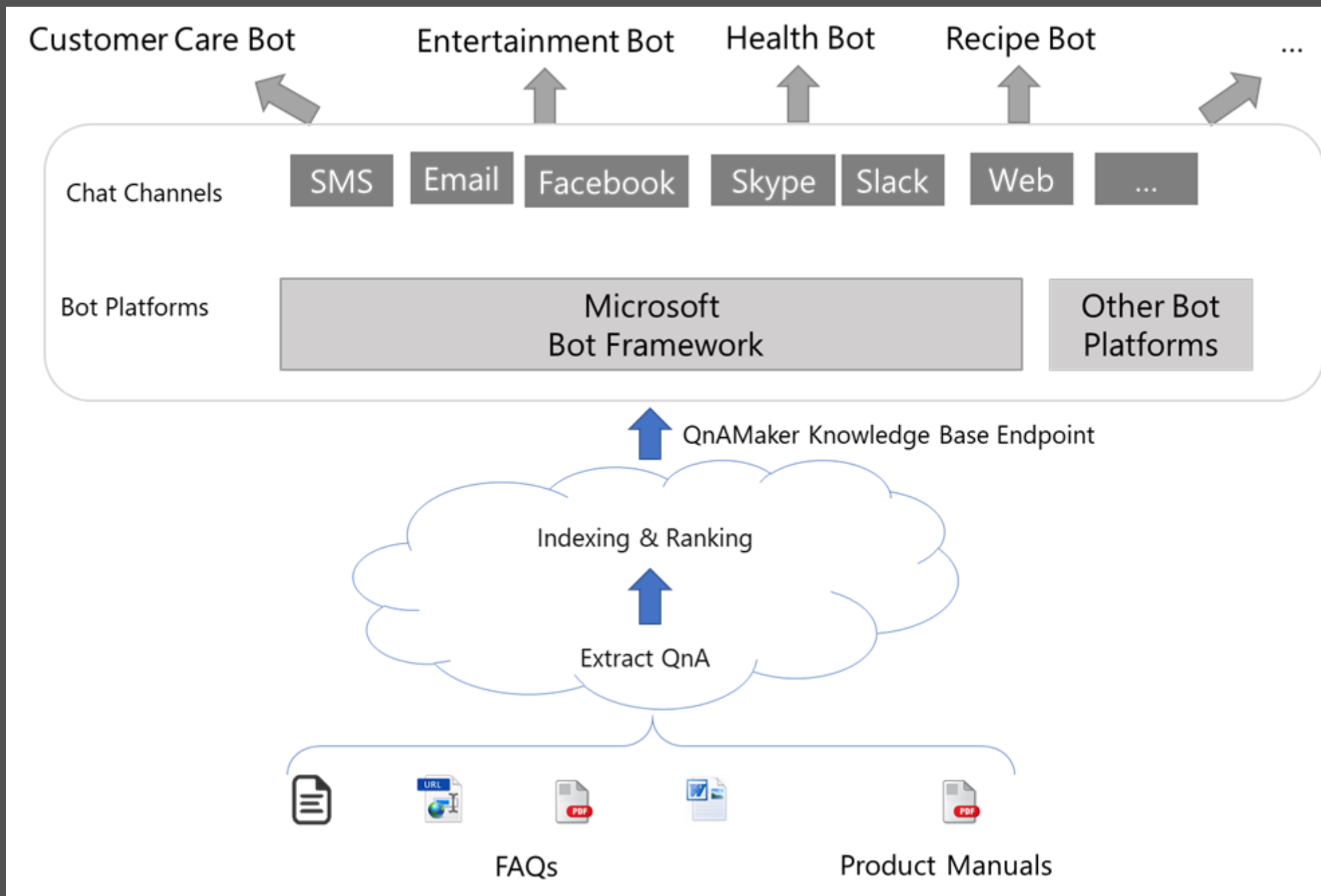
```
{
  "query": "hello",
  "topScoringIntent": {
    "intent": "Hello",
    "score": 0.9909954
  },
  "intents": [
    {
      "intent": "Hello",
      "score": 0.9909954
    },
    {
      "intent": "None",
      "score": 0.0339101851
    },
    {
      "intent": "How_are_you",
      "score": 0.012524181
    }
  ],
  "entities": []
}
```


QnA



QnA Maker

QnA Maker — это служба для отправки вопросов и получения ответов из частично структурированного содержимого, например документов с вопросами и ответами, URL-адресов и руководств по продуктам.

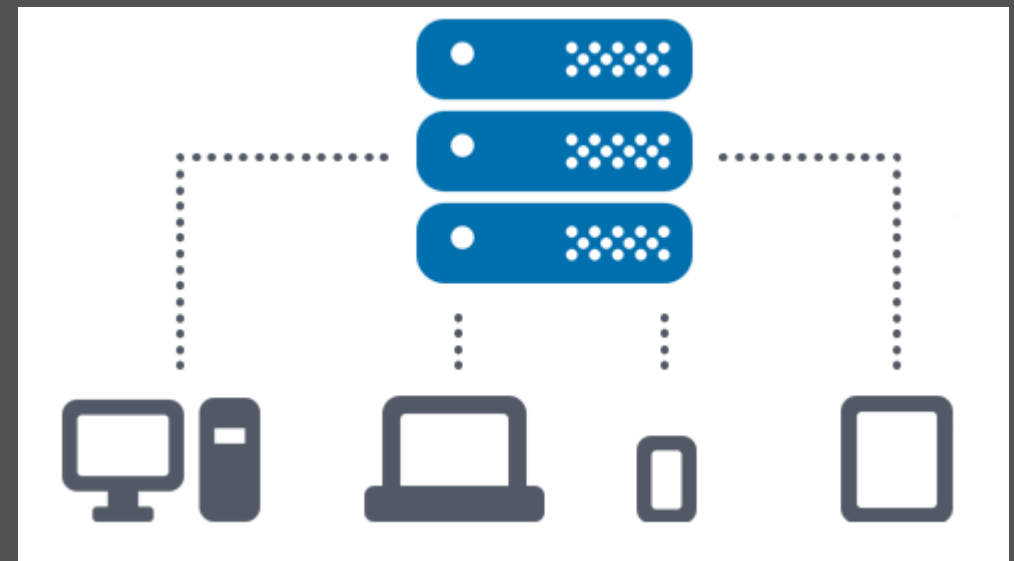
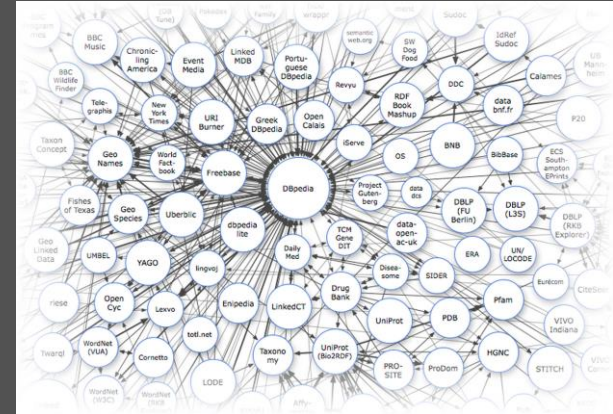


Особенности

- Создать бота для ответов на вопросы
- SaaS
- Масштабирование

Основные процессы QnA Maker

1. Создание базы знаний (извлечение)
2. Создание конечной точки опубликованной модели (сопоставление)





1. Create a QnAMaker resource in Azure



2. Log on to the qnamaker portal

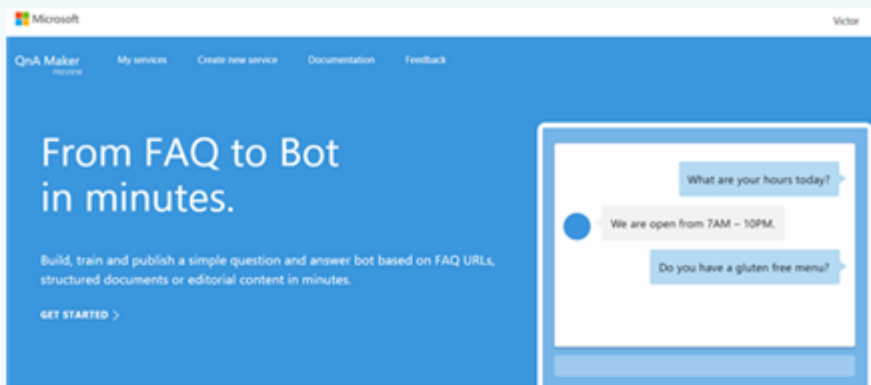
5. Manage KB in portal or via APIs

3. Create a Knowledge Base

4. Use QnA endpoint in Bot

<https://userxyz.azurewebsites.net/generateanswer...>

<https://qnamaker.ai>



QnA Maker Subscription



App Service (QnA runtime + QnA ranking)

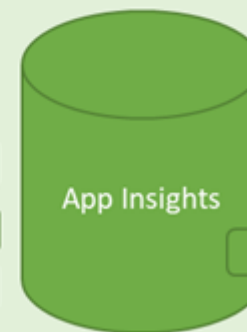


Azure Search

Metadata

QnA

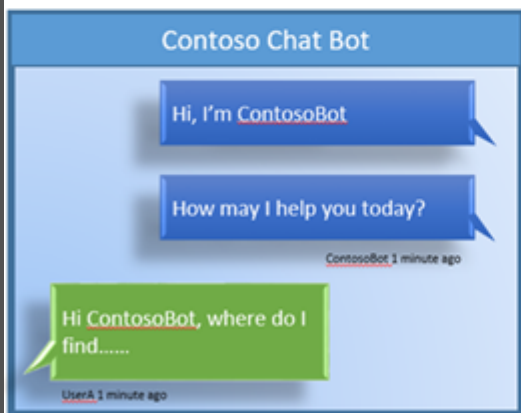
Synonyms



App Insights

Chat log

User Subscription



Bot Emulator



Bot Emulator

Microsoft Bot Framework Emulator

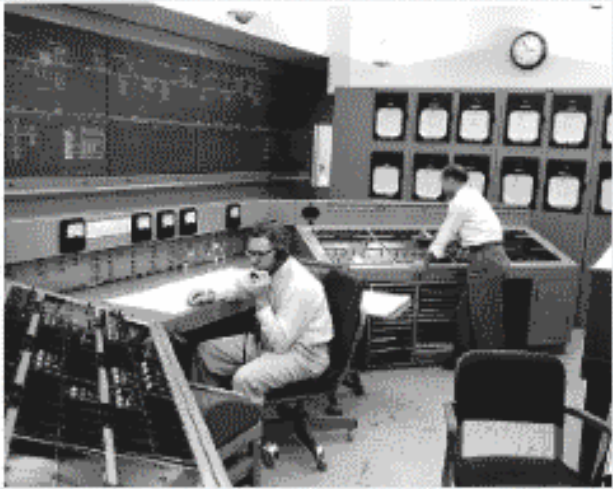
http://localhost:3978/api/messages

Bot

computers

you

Here's what I found:



Bot at 8:10:56 PM

Type your message...

Details

```
{
  "type": "message",
  "text": "Here's what I found:",
  "attachments": [
    {
      "contentType": "image/jpg",
      "contentUrl":
        "https://farm4.staticflickr.com/3685/9467782468_e0955d9444.jpg"
    }
  ],
  "from": {
    "id": "0km8cflmcbnb28de6",
    "name": "Bot"
  },
}
```

Log

```
[20:06:42] Emulator listening on http://[::]
[20:06:42] Checking for new version...
[20:06:42] Application is up to date.
[20:06:42] ngrok listening on https://83edba
[20:06:42] ngrok traffic inspector: http://1
[20:06:53] -> POST 202 [conversationUpdate]
[20:06:54] <- POST 200 Reply[message] ### F1
[20:07:01] -> POST 202 [message] hello
[20:07:03] <- POST 200 Reply[message] image/
[20:07:14] -> POST 202 [message] robots
[20:07:15] <- POST 200 Reply[message] I didn
[20:07:19] -> POST 202 [message] bot
[20:07:20] <- POST 200 Reply[message] I didn
[20:07:28] -> POST 202 [message] computers
```


Bot Emulator

<https://github.com/Microsoft/BotFramework-Emulator/releases>

Bot Template

<http://aka.ms/bf-bc-vstemplate>

Этот компьютер > Документы > Visual Studio 2017 > Templates > ProjectTemplates > Visual C# >				
Имя	^	Дата изменения	Тип	Размер
Бот	Bot Application.zip	19.10.2017 11:46	WinRAR ZIP archive	25 КБ

More information

<http://bit.ly/BotFramework>

<https://github.com/MaximDmitrievich/MAIBot>

Useful links

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

Building Bots with Microsoft Bot Framework by Kishore Gaddam

<https://stackoverflow.com/questions/tagged/botframework>

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

<https://docs.microsoft.com/en-us/azure/bot-service/?view=azure-bot-service-3.0>

Contact Information:

E-mail:
mdereimin@gmail.com

Telegram: @MaximEremin



