Module 10: Integrating SaaS Services Available on the Azure Platform

Contents:

Module overview

This module introduces multiple SaaS services available in Azure that are available for integration into existing Azure solutions. These services include Cognitive Services, Bot Service, Machine Learning and Media Services.

Objectives

After completing this module, students will be able to:

- Identify when Cognitive Services, Bot Service or Machine Learning is appropriate for their solution.
- Compare the various features available in Media Services and determine the appropriate features for their solution.

Lesson 1: Cognitive Services

This lesson introduces the various APIs available in Azure Cognitive Services and specifically reviews LUIS, Face API and Speech API in detailed examples.

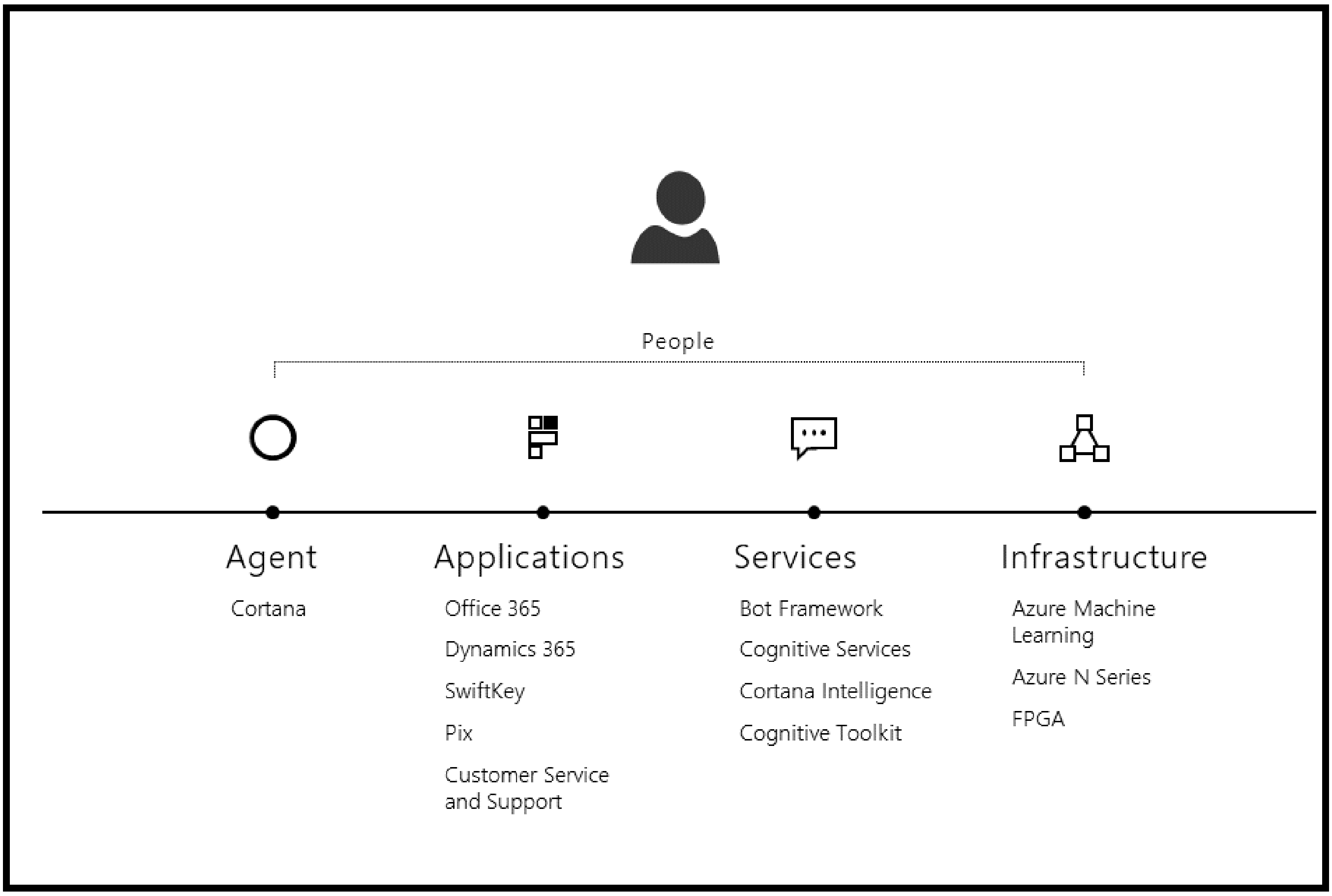
Lesson objectives

After completing this lesson, you will be able to:

- Detail the various APIs available in Cognitive Services.

- Identify when to use the Face API, Speech API or Language Understanding (LUIS) service.

Cognitive Services



Microsoft Cognitive Services are a set of APIs, SDKs and services available to developers to make their applications more intelligent, engaging and discoverable. Microsoft Cognitive Services expands on Microsoft's evolving portfolio of machine learning APIs and enables developers to easily add intelligent features – such as emotion and video detection; facial, speech and vision recognition; and speech and language understanding – into their applications.

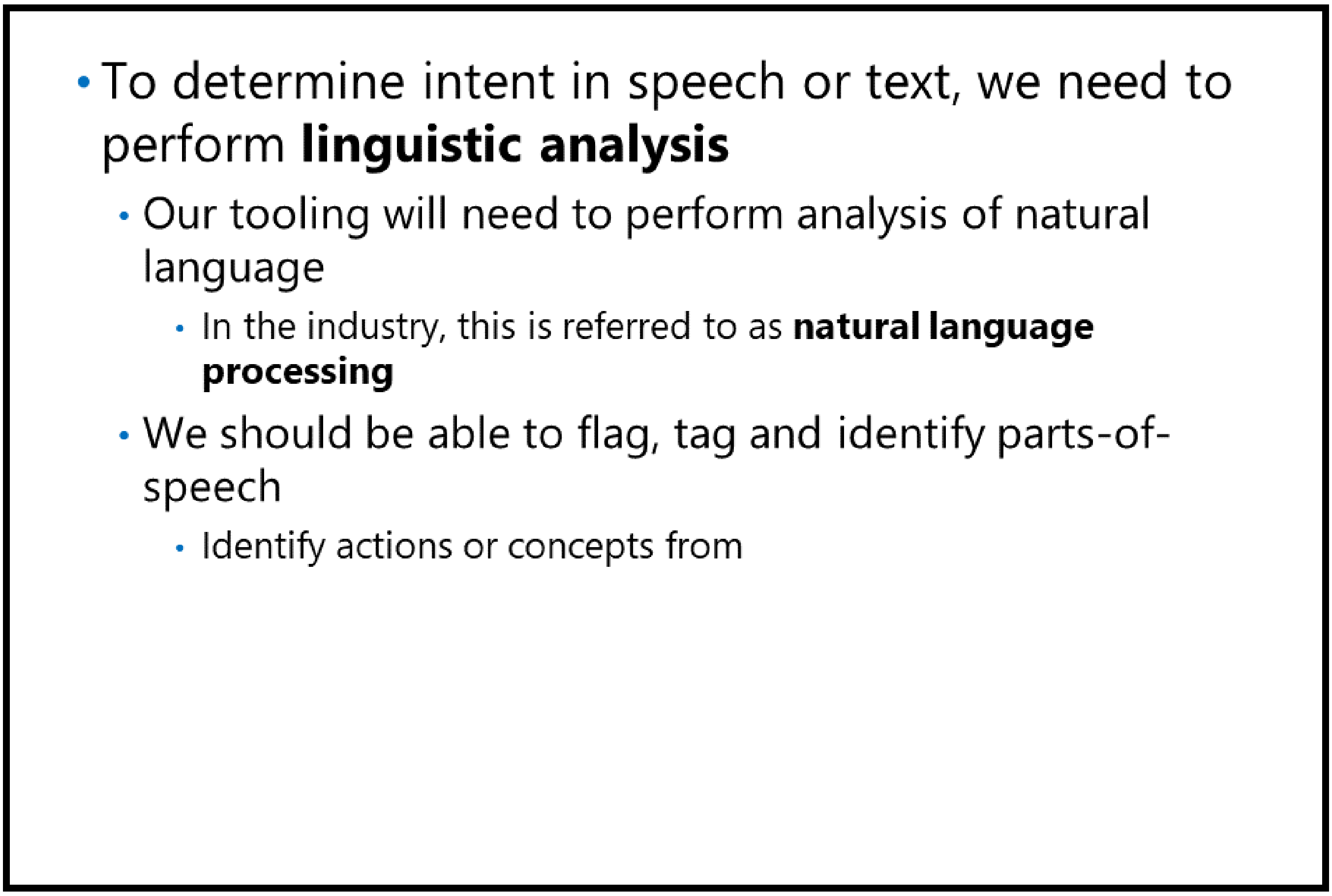
Bing APIs

Cognitive Services, as a suite, also includes various Bing APIs that can be used in your applications:

- **Bing Web Search**
  - Bing Web Search API provides an experience similar to Bing.com/search by returning search results that Bing determines are relevant to a user's query. The results include Web pages and may also include images, videos, and more.
- **Bing Image Search**
  - Bing Image Search API provides an experience similar to Bing.com/images by returning images that Bing determines are relevant to a user's query.
- **Bing Autosuggest**

- Bing Autosuggest API lets you send a partial search query term to Bing and get back a list of suggested queries that other users have searched on. For example, as the user enters each character of their search term, you'd call this API and populate the search box's drop-down list with the suggested query strings.

Intent Detection



- To determine intent in speech or text, we need to perform **linguistic analysis**
  - Our tooling will need to perform analysis of natural language
    - In the industry, this is referred to as **natural language processing**
  - We should be able to flag, tag and identify parts-of-speech
    - Identify actions or concepts from

Language Understanding (LUIS)

Language Understanding (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.

A LUIS app is a domain-specific language model designed by you and tailored to your needs. You can start with a prebuilt domain model, build your own, or blend pieces of a prebuilt domain with your own custom information.

A model starts with a list of general user intentions such as "Book Flight" or "Contact Help Desk." Once the intentions are identified, you supply example phrases called utterances for the intents. Then you label the utterances with any specific details you want LUIS to pull out of the utterance.

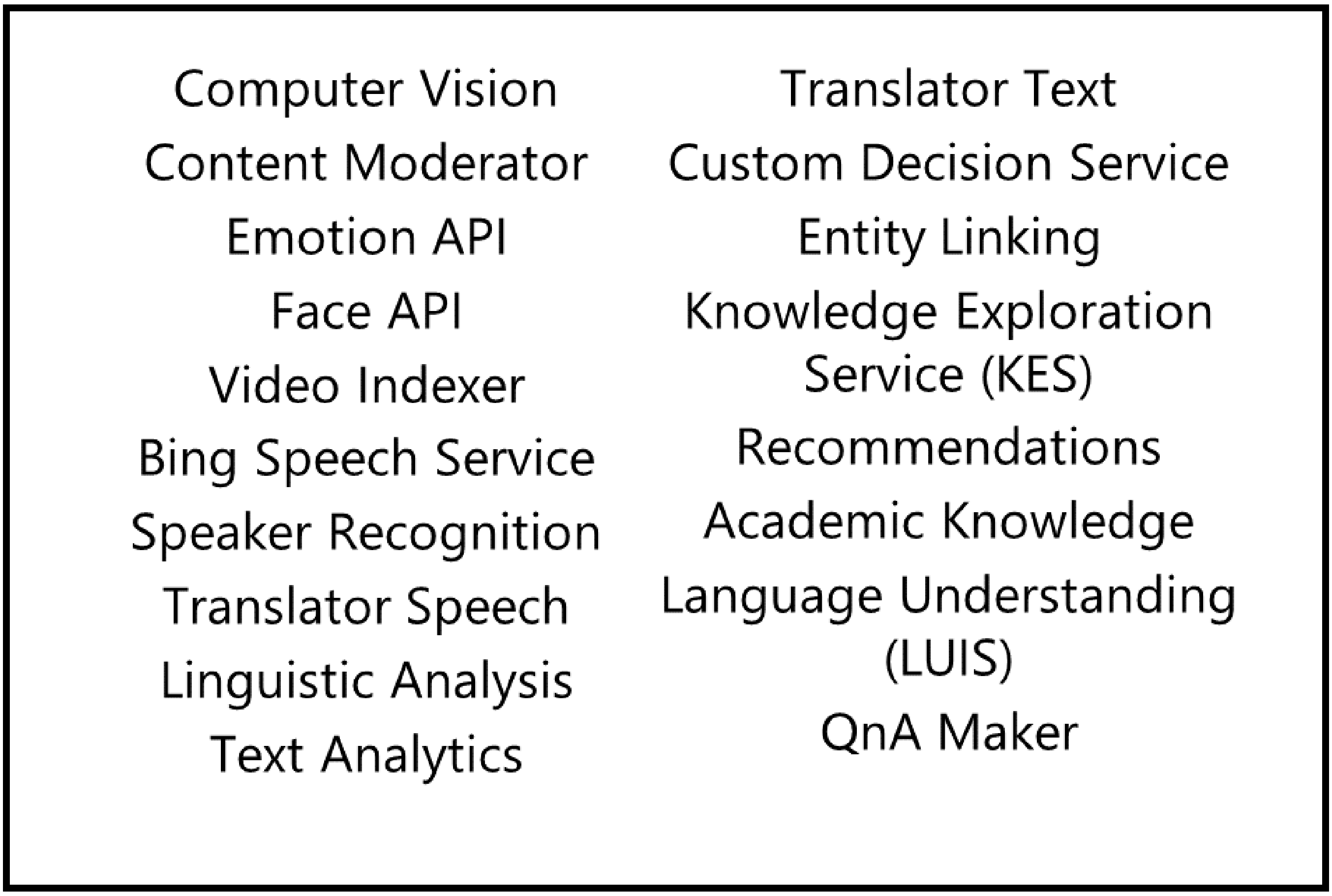
Prebuilt domain models include all these pieces for you and are a great way to start using LUIS quickly.

After the model is designed, trained, and published, it is ready to receive and process utterances. The LUIS app receives the utterance as an HTTP request and responds with extracted user intentions. Your client application sends the utterance and receives LUIS's evaluation as a JSON object. Your client app can then take appropriate action.

Key LUIS Concepts

- **Intents**
    - An intent represents actions the user wants to perform. The intent is a purpose or goal expressed in a user's input, such as booking a flight, paying a bill, or finding a news article. You define and name intents that correspond to these actions. A travel app may define an intent named "BookFlight."
- **Utterances**
    - An utterance is text input from the user that your app needs to understand. It may be a sentence, like "Book a ticket to Paris", or a fragment of a sentence, like "Booking" or "Paris flight." Utterances aren't always well-formed, and there can be many utterance variations for a particular intent.
- **Entities**
    - An entity represents detailed information that is relevant in the utterance. For example, in the utterance "Book a ticket to Paris", "Paris" is a location. By recognizing and labeling the entities that are mentioned in the user’s utterance, LUIS helps you choose the specific action to take to answer a user's request.

Cognitive APIs



Computer Vision
Content Moderator
Emotion API
Face API
Video Indexer
Bing Speech Service
Speaker Recognition
Translator Speech
Linguistic Analysis
Text Analytics

Translator Text
Custom Decision Service
Entity Linking
Knowledge Exploration Service (KES)
Recommendations
Academic Knowledge
Language Understanding (LUIS)
QnA Maker

Cognitive Services, as a suite, includes a wide variety of APIs. Some example APIs include the following APIs listed below.

Text Analytics API

Text Analytics API is a cloud-based service that provides advanced natural language processing over raw text, and includes three main functions: sentiment analysis, key phrase extraction, and language detection.

Speaker Recognition API

Speaker Recognition API is a cloud-based APIs that provide the most advanced algorithms for speaker verification and speaker identification.

Content Moderator API

Content Moderator API tracks, flags, assesses, and filters out offensive and unwanted content that creates risk for applications

Face API

Face API is a cloud-based service that provides the advanced face algorithms with two main functions: face detection with attributes and face recognition. Face API detects up to 64 human faces with high precision face location in an image. And the image can be specified by file in bytes or valid URL. The API returns a face rectangle (left, top, width and height) indicating the face location in the image is returned along with each detected face. Optionally, face detection extracts a series of face related attributes such as pose, gender, age, head pose, facial hair and glasses.

Face recognition is widely used in many scenarios including security, natural user interface, image content analysis and management, mobile apps, and robotics. Four face recognition functions are provided: face verification, finding similar faces, face grouping, and person identification.

Lesson 2: Bot Services

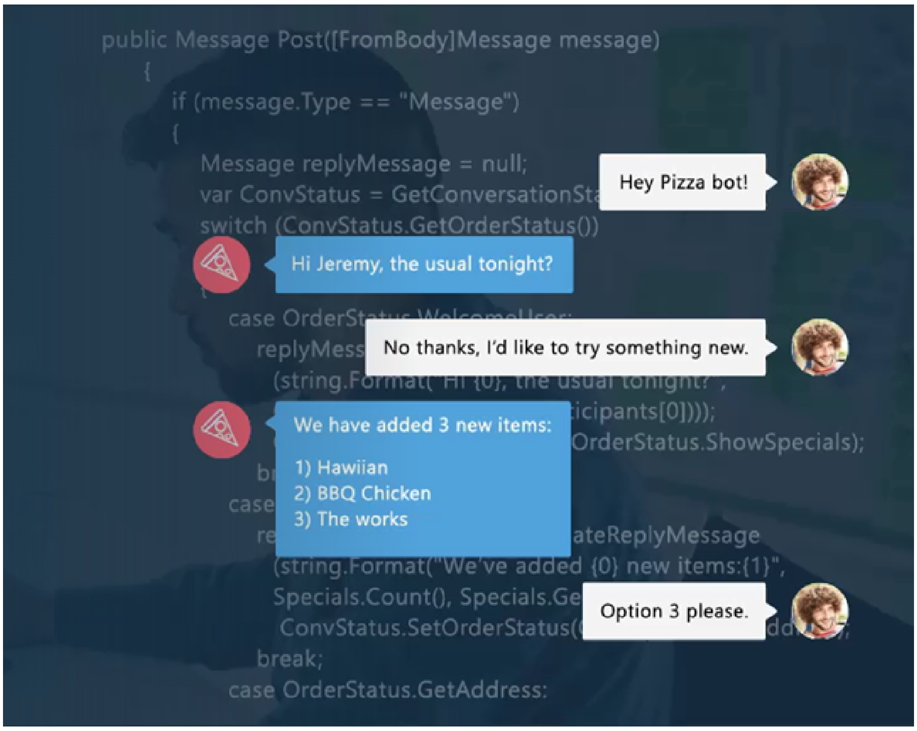
This lesson introduces Bot Services by first detailing it's underlying framework, Bot Framework, and then using QnA Maker to create an example bot.

Lesson objectives

After completing this lesson, you will be able to:

- Describe the relationship to Bot Framework and Azure Bot Services.
- Create a simple bot using QnA Maker.

Bot Services



```
public Message Post([FromBody]Message message)
    {
      if (message.Type == "Message")
      {
        Message replyMessage = null;
        var ConvStatus = GetConversationSt
        switch (ConvStatus.GetOrderStatus())

          case OrderSt
            replyMess
              (string.Format( Hi {0}, the usual tonight? ,
                                            icipants[0])));
                                            OrderStatus.ShowSpecials);
            br
          case
            re                              ateReplyMessage
              (string.Format("We've added {0} new items:{1}",
              Specials.Count(), Specials.Ge
              ConvStatus.SetOrderStatus(
            break;
          case OrderStatus.GetAddress:
```

Hey Pizza bot!
Hi Jeremy, the usual tonight?
No thanks, I'd like to try something new.
We have added 3 new items:
1) Hawiian
2) BBQ Chicken
3) The works
Option 3 please.

Bot Service provides an integrated environment that is purpose-built for bot development, enabling you to build, connect, test, deploy, and manage intelligent bots, all from one place. Bot Service leverages the Bot Builder SDK with support for .NET and Node.js. You can write a bot, connect, test, deploy, and manage it from your web browser with no separate editor or source control required. For simple bots, you may not need to write code at all. Bot Service accelerates bot development with Five bot templates you can choose from when you create a bot. You can further modify your bot directly in the browser using the Azure editor or in an Integrated Development Environment (IDE), such as Visual Studio and Visual Studio Code.

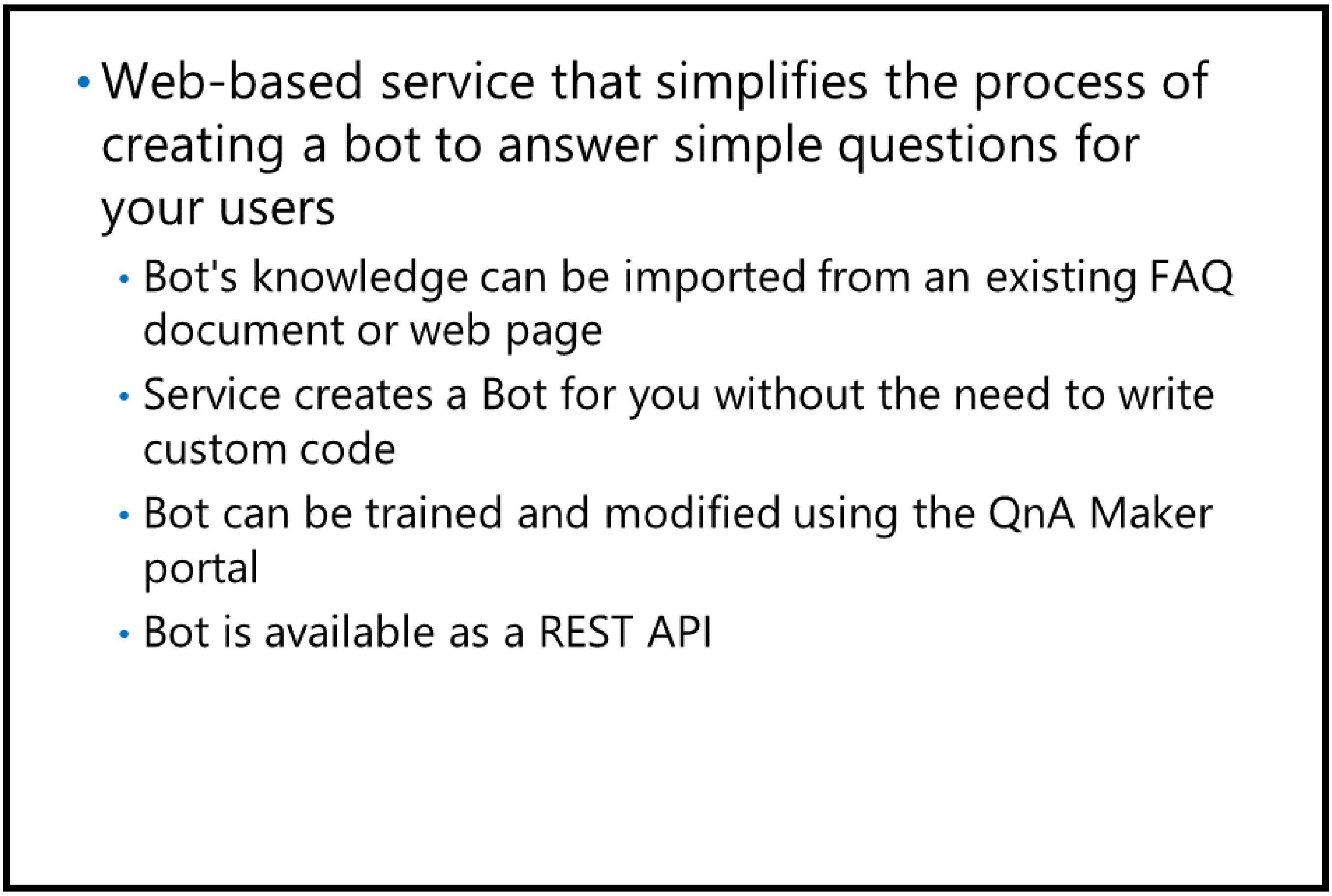
Here are some key features of Bot Service:

- **Multiple language support**
  - Bot Service leverages Bot Builder with support for .NET and Node.js.
- **Bot templates**
  - Bot Service templates allow you to quickly create a bot with the code and features you need. Choose from a Basic bot, a Forms bot for collecting user input, a Language understanding bot that leverages LUIS to understand user intent, a QnA bot to handle FAQs, or a Proactive bot that alerts users of events.
- **Bring your own dependencies**
  - Bots support NuGet and NPM, so you can use your favorite packages in your bot.
- **Flexible development**
  - Code your bot right in the Azure portal or set up continuous integration and deploy your

bot through GitHub, Visual Studio Team Services, and other supported development tools. You can also publish from Visual Studio.

- **Connect to channels**
  - Bot Service supports several popular channels for connecting your bots and the people that use them. Users can start conversations with your bot on any channel that you've configured your bot to work with, including Skype, Facebook, Teams, Slack, SMS, and several others.
- **Tools and services**
  - Test your bot with the Bot Framework Emulator and preview your bot on different channels with the Channel Inspector.
- **Open source**
  - The Bot Builder SDK is open-source and available on GitHub.

QnA Maker



- Web-based service that simplifies the process of creating a bot to answer simple questions for your users
  - Bot's knowledge can be imported from an existing FAQ document or web page
  - Service creates a Bot for you without the need to write custom code
  - Bot can be trained and modified using the QnA Maker portal
  - Bot is available as a REST API

Microsoft QnA Maker is a REST API and web-based service that trains AI to respond to user's questions in a more natural, conversational way. QnA Maker provides a graphical user interface that allows non-developers to

train, manage, and use the service for a wide range of solutions.

QnA Maker extracts a knowledge base from two types of input: FAQ pages and product manuals. The tool supports extraction from FAQ web pages or documents in the question-answer format. The tool can also extra QnA pairs from PDF-format product manuals.

Once extracted, the QnA Maker service creates a knowledge base and bot using the knowledge base. The bot can then be used, via a REST API, in any existing web application or website to answer questions for users. Over time, the knowledge base can be updated, retrained, and republished to meet the morphing needs to a user-facing web application.

Lesson 3: Machine Learning

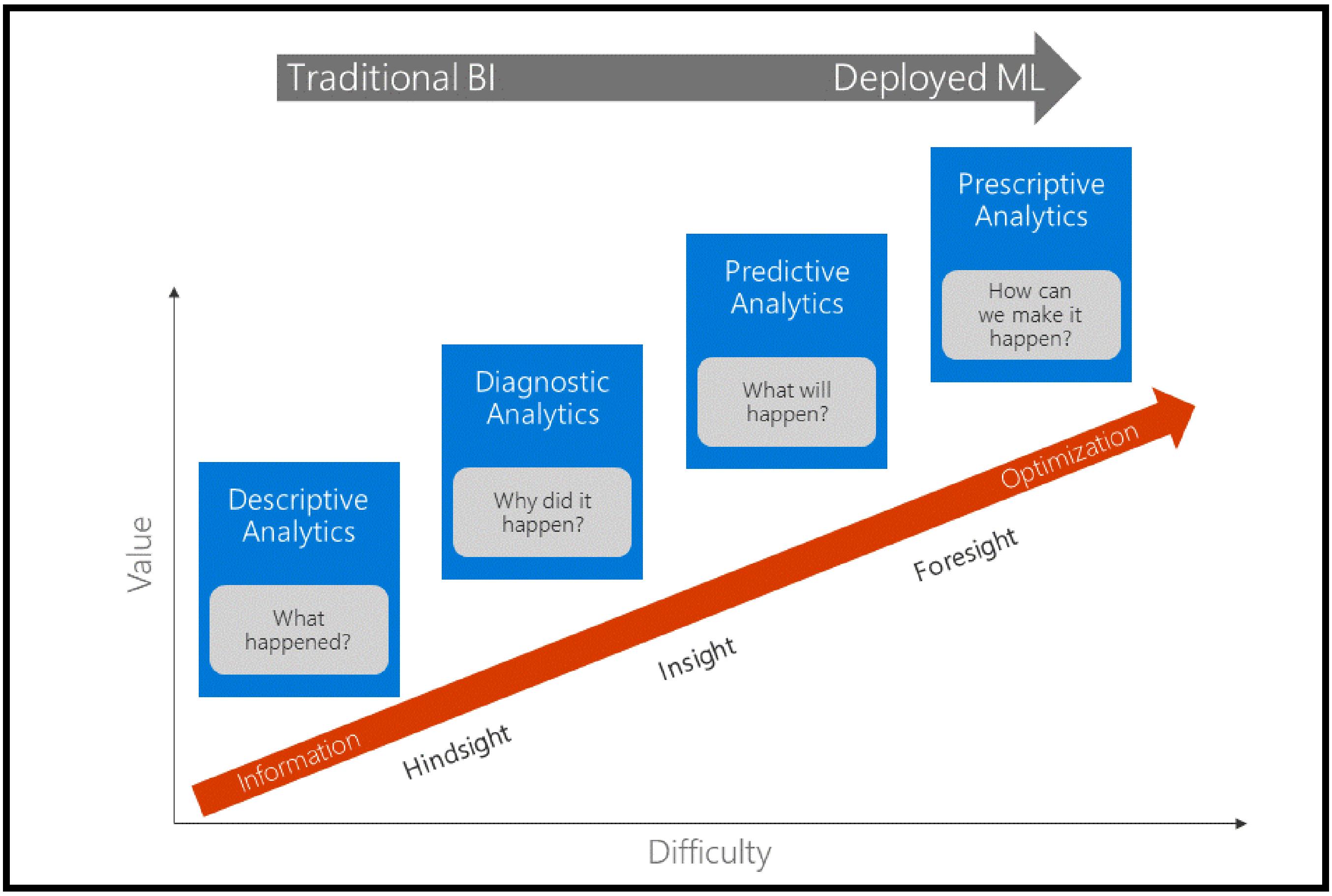
This lesson briefly introduces Azure Machine Learning in the context of traditional BI and prescriptive analytics.

Lesson objectives

After completing this lesson, you will be able to:

- Describe Azure Machine Learning.

Machine Learning



Machine learning is a data science technique that allows computers to use existing data to forecast future behaviors, outcomes, and trends. Using machine learning, computers learn without being explicitly programmed.

Forecasts or predictions from machine learning can make apps and devices smarter. When you shop online, machine learning helps recommend other products you might like based on what you've purchased. When your credit card is swiped, machine learning compares the transaction to a database of transactions and helps detect fraud. When your robot vacuum cleaner vacuums a room, machine learning helps it decide whether the job is done.

Azure Machine Learning

Azure Machine learning is an end-to-end data science and analytics solution that's integrated into Azure. It allows users to develop experiments as well as deploy data and models via the cloud. Its composed of the Azure Machine Learning Workbench, Experimentation service, Model Management Service, Libraries for Apache Spark, and the Visual Studio Code Tools for AI.

Azure Machine learning fully support various open source technologies, such as scikit-learn, TensorFlow, and more. A massive library of open source Python packages is accessible. Also, you can execute experiments in managed environments such as spark clusters or docker containers. Azure Machine Learning built on top of open source technologies. These technologies are Jupyter Notebook, Conda, Python, Docker, Apache Spark, and Kubernetes. It also includes open source technologies from Microsoft itself, such as Cognitive Toolkit.

Azure Machine Learning Workbench is a desktop application supported on both windows and macOS that includes command-line tools. It allows users to help manage learning solutions via data ingestion and preparation, model development, experiment management, and even model deployment in different target environments.

The Azure Machine Learning Experimentation Service helps handle the implementation of machine learning experiments. It also provides project management, roaming, sharing, and git integration to support the Workbench. Azure Machine Learning Experimentation Services allows the implementation of services across a range of environment options such as Local native, Local Docker container, or Scale out Spark cluster in Azure. The Experimentation Service also creates Virtual environments for scripts to provide an isolated space with reproducible results. It documents run history information and visually displays the information so you can quickly select the best model from your experiments.

Azure Machine Learning Model Management Service provides users the ability to deploy predictive models into a range of environments. Information on models, such as the version and lineage, is notated from training runs throughout the deployment. The models themselves are registered, managed, and stored in the cloud.

The Microsoft Machine Learning Library for Apache Spark or MMLSpark is an open-source Spark Package providing data science and Deep Learning tools for Apache Spark. MMLSpark allows users to create robust, analytical, and highly scalable predictive models for large image and text datasets.

Visual Studio Code Tools for AI itself is an extension used with Visual Studio code that allows you to test, build, and deploy AI and Deep Learning solutions. It contains various integration points from Azure Machine learning. Such examples include visualization of run history that displays the performance of training runs, a gallery view allowing you to bootstrap and browse new projects within the Microsoft Cognitive Toolkit and other deep-learning frameworks, as well as an explorer view, to allow users to select targets for your scripts to execute.

Lesson 4: Media Processing

This lesson introduces the Cognitive Services Computer Vision API and Azure Media Services as components that can assist with the processing of still-image and video media.

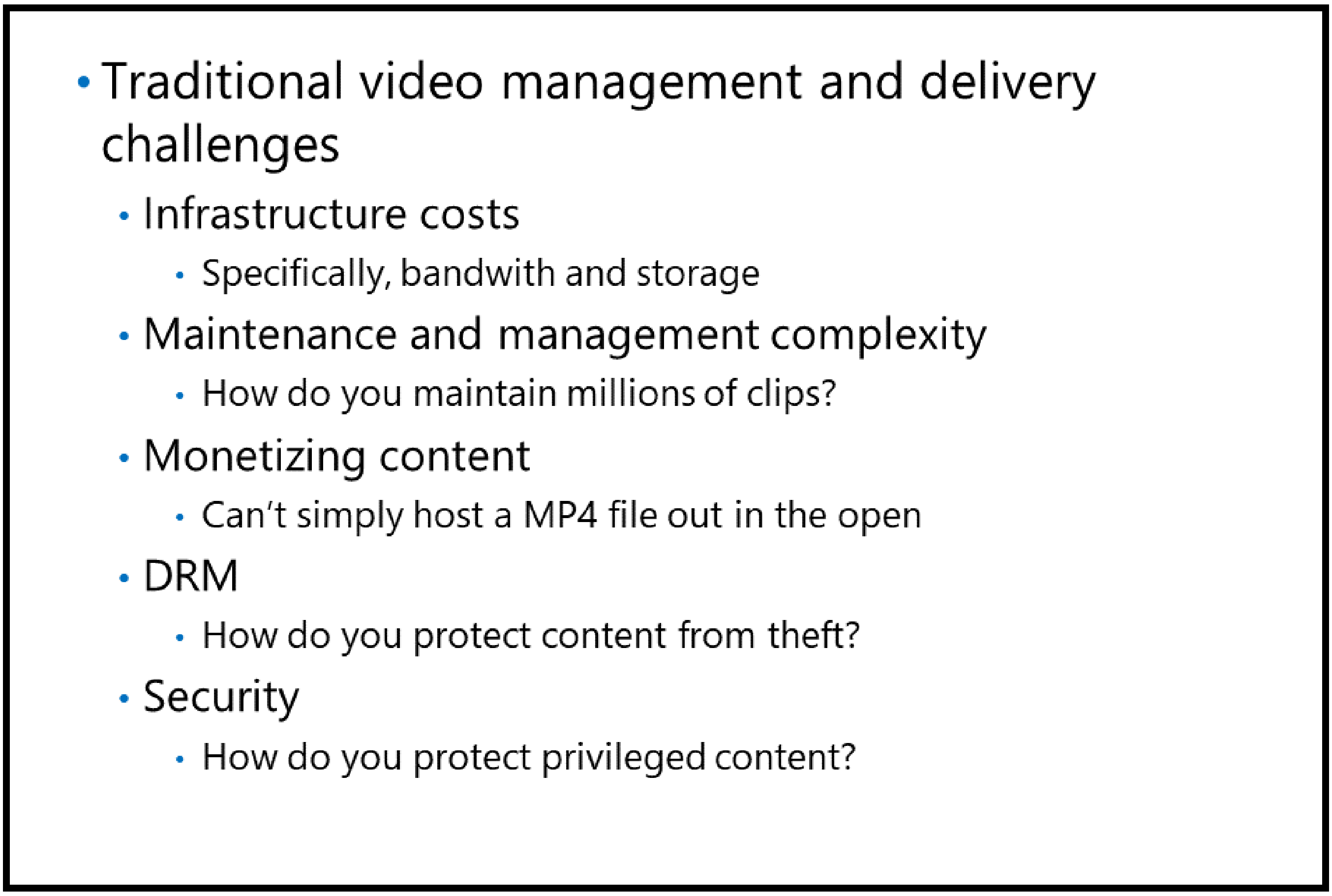
Lesson objectives

After completing this lesson, you will be able to:

- Describe Azure Media Services.

- Discuss Media Services workflows including live streaming, dynamic packaging and static conversion.
- Detail uses of the Computer Vision API.

Media Services

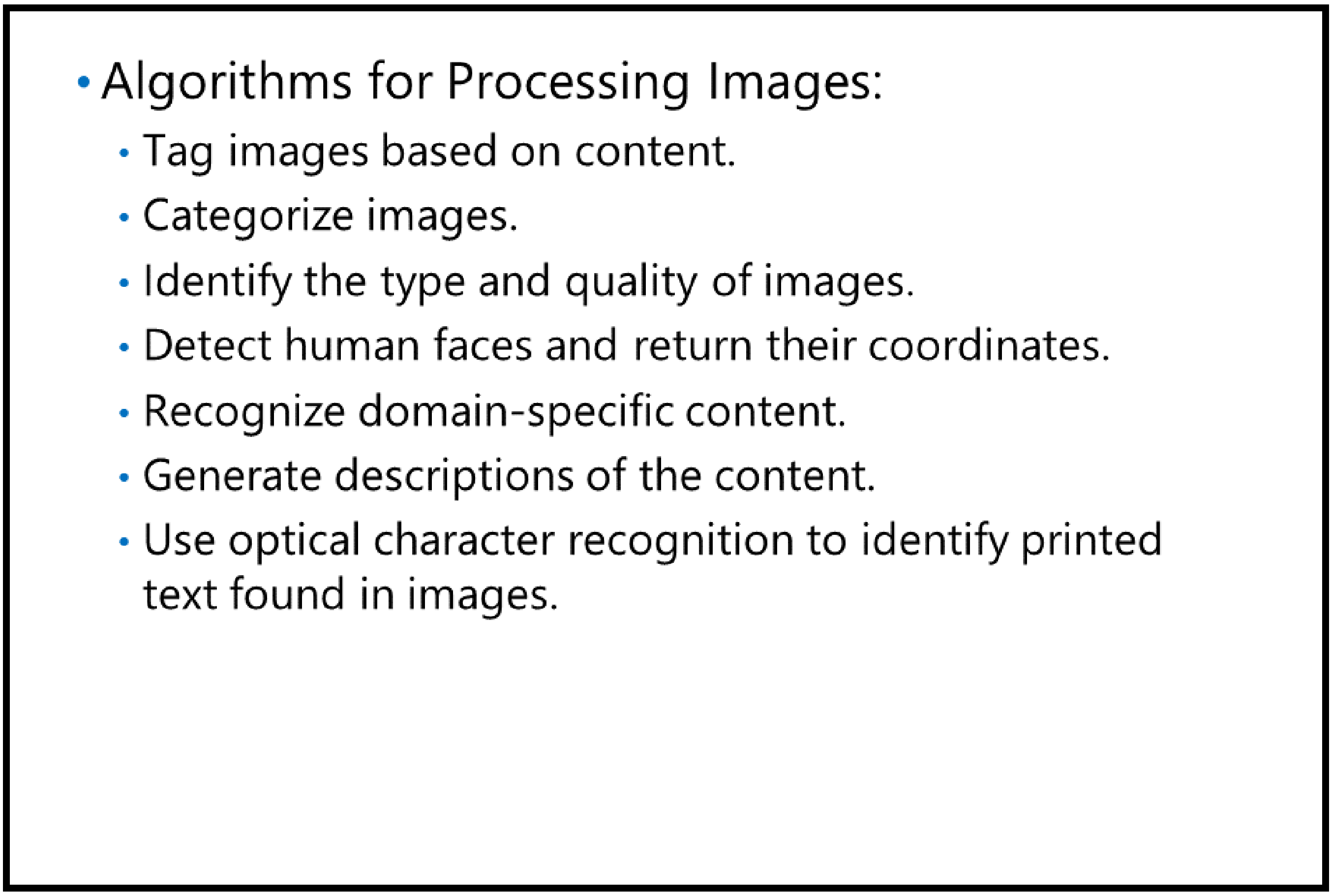


- Traditional video management and delivery challenges
  - Infrastructure costs
    - Specifically, bandwith and storage
  - Maintenance and management complexity
    - How do you maintain millions of clips?
  - Monetizing content
    - Can't simply host a MP4 file out in the open
  - DRM
    - How do you protect content from theft?
  - Security
    - How do you protect privileged content?

Microsoft Azure Media Services is an extensible cloud-based platform that enables developers to build scalable media management and delivery applications. Media Services is based on REST APIs that enable you to securely upload, store, encode, and package video or audio content for both on-demand and live streaming delivery to various clients (for example, TV, PC, and mobile devices).

You can build end-to-end workflows using entirely Media Services. You can also choose to use third-party components for some parts of your workflow. For example, encode using a third-party encoder. Then, upload, protect, package, deliver using Media Services. You can choose to stream your content live or deliver content on-demand.

Computer Vision API



- Algorithms for Processing Images:
  - Tag images based on content.
  - Categorize images.
  - Identify the type and quality of images.
  - Detect human faces and return their coordinates.
  - Recognize domain-specific content.
  - Generate descriptions of the content.
  - Use optical character recognition to identify printed text found in images.

The cloud-based Computer Vision API provides developers with access to advanced algorithms for processing images and returning information. By uploading an image or specifying an image URL, Microsoft Computer Vision algorithms can analyze visual content in different ways based on inputs and user choices. With the Computer Vision API users can analyze images to:+

- Tag images based on content.
- Categorize images.
- Identify the type and quality of images.
- Detect human faces and return their coordinates.
- Recognize domain-specific content.
- Generate descriptions of the content.
- Use optical character recognition to identify printed text found in images.
- Recognize handwritten text.
- Distinguish color schemes.
- Flag adult content.
- Crop photos to be used as thumbnails.

Image Tagging

Computer Vision API returns tags based on more than 2000 recognizable objects, living beings, scenery, and actions. When tags are ambiguous or not common knowledge, the API response provides 'hints' to clarify the meaning of the tag in context of a known setting. Tags are not organized as a taxonomy and no inheritance hierarchies exist. A collection of content tags forms the foundation for an image 'description' displayed as human readable language formatted in complete sentences.

After uploading an image or specifying an image URL, Computer Vision API's algorithms output tags based on the objects, living beings, and actions identified in the image. Tagging is not limited to the main subject, such as a person in the foreground, but also includes the setting (indoor or outdoor), furniture, tools, plants, animals, accessories, gadgets etc.

Description Generation

Computer Vision API's algorithms analyze the content in an image. This analysis forms the foundation for a 'description' displayed as human-readable language in complete sentences. The description summarizes what is found in the image. Computer Vision API's algorithms generate various descriptions based on the objects identified in the image. The descriptions are each evaluated and a confidence score generated. A list is then returned ordered from highest confidence score to lowest.

Color Schemes

The Computer Vision algorithm extracts colors from an image. The colors are analyzed in three different contexts: foreground, background, and whole. They are grouped into twelve 12 dominant accent colors. Those accent colors are black, blue, brown, gray, green, orange, pink, purple, red, teal, white, and yellow. Depending on the colors in an image, simple black and white or accent colors may be returned in hexadecimal color codes.

The Computer Vision API can also determine color extracted from an image designed to represent the most eye-popping color to users via a mix of dominant colors and saturation and identify this color as the **Accent Color**.

Optical Character Recognition (OCR)

OCR technology detects text content in an image and extracts the identified text into a machine-readable character stream. You can use the result for search and numerous other purposes like medical records, security, and banking. It automatically detects the language. OCR saves time and provides convenience for users by allowing them to take photos of text instead of transcribing the text. If needed, OCR corrects the rotation of the recognized text, in degrees, around the horizontal image axis.

Lab: Deploying Service Instances as Components of Overall Azure Solutions

Scenario

A local ESL (English-as-a-second-language) outreach group wants to add a quick translation tool to their website. To accomplish this, they would like to use the Translation API in Cognitive Services and wrap it with a simple proxy so that they can use it on their website and potentially a future mobile application.

Objectives

- Deploy ARM Template with Function App and Cognitive Service

Lab setup

Estimated Time: 60 minutes

Virtual machine: **20535A-SEA-ARCH**

User name: **Admin**

Password: **Pa55w.rd**

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

Exercise 1: Deploy Function App and Cognitive Service using ARM Template

Exercise 2: Cleanup Subscription

Review Question(s)

Module review and takeaways

Review Question(s)