

Microsoft Azure IoT and Advanced Cloud Services



Alon Fliess

Chief Software Architect

alonf@codevalue.net

<http://blogs.microsoft.co.il/blogs/alon>



Agenda

- Introduction to cloud computing
 - Microsoft Azure IaaS
 - Microsoft Azure PaaS
 - Azure Web Apps
 - Introduction to the MSA & Serverless
 - Microsoft Azure IoT
- Introduction to C# and Xamarin
- Advanced Azure Analysis Services



About Me

➤ Alon Fliess:

- Chief Software Architect & Co-Founder at CodeValue Ltd.
- More than 25 years of hands-on experience
- Microsoft Regional Director & Microsoft MVP
- Renowned speaker at both international and domestic events





About CodeValue

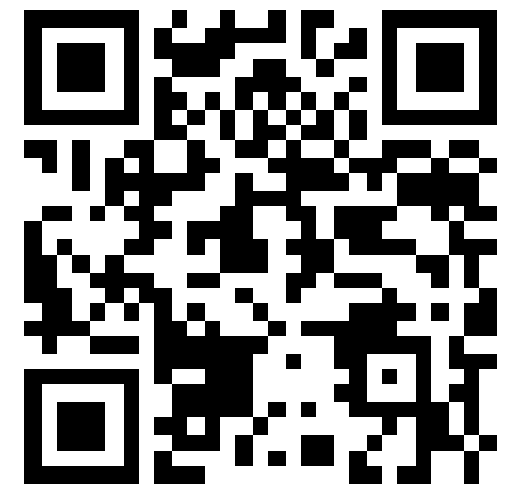
- Awesome software company!
- About 200 employees, most of which technology experts
- High quality software development solutions





Israeli Azure Developer Community

- Curious about Azure? Join us!
- Meet every two months at Microsoft Ra'anana
- Next meeting – December 2015
- <http://www.meetup.com/IsraeliAzureDevelopers>
- <https://www.facebook.com/IsraeliAzureDevelopers>



Cloud Computing





Introduction to Cloud Computing

Cloud Computing - A Game Changing Technology

- Infinite shared resources & services
 - Infrastructure is not a limiting factor
 - No need to equip for peak-load requirements
- Elasticity on demand
 - Anytime, anywhere
- Efficient scalability and high availability
- Suitable pricing models
 - Pay for what you use



Cloud Computing

Evolution of Computing - The Next (Current) Big Thing

- Virtualization and Abstraction
 - Details are abstracted from consumers
 - Reduces complexity
- Not necessarily the Internet, can be on premises
 - Private cloud
- Automation, Monitoring, Deployment
 - Reduce cost, shift risk, shorten time-to-market, focus on business functionality

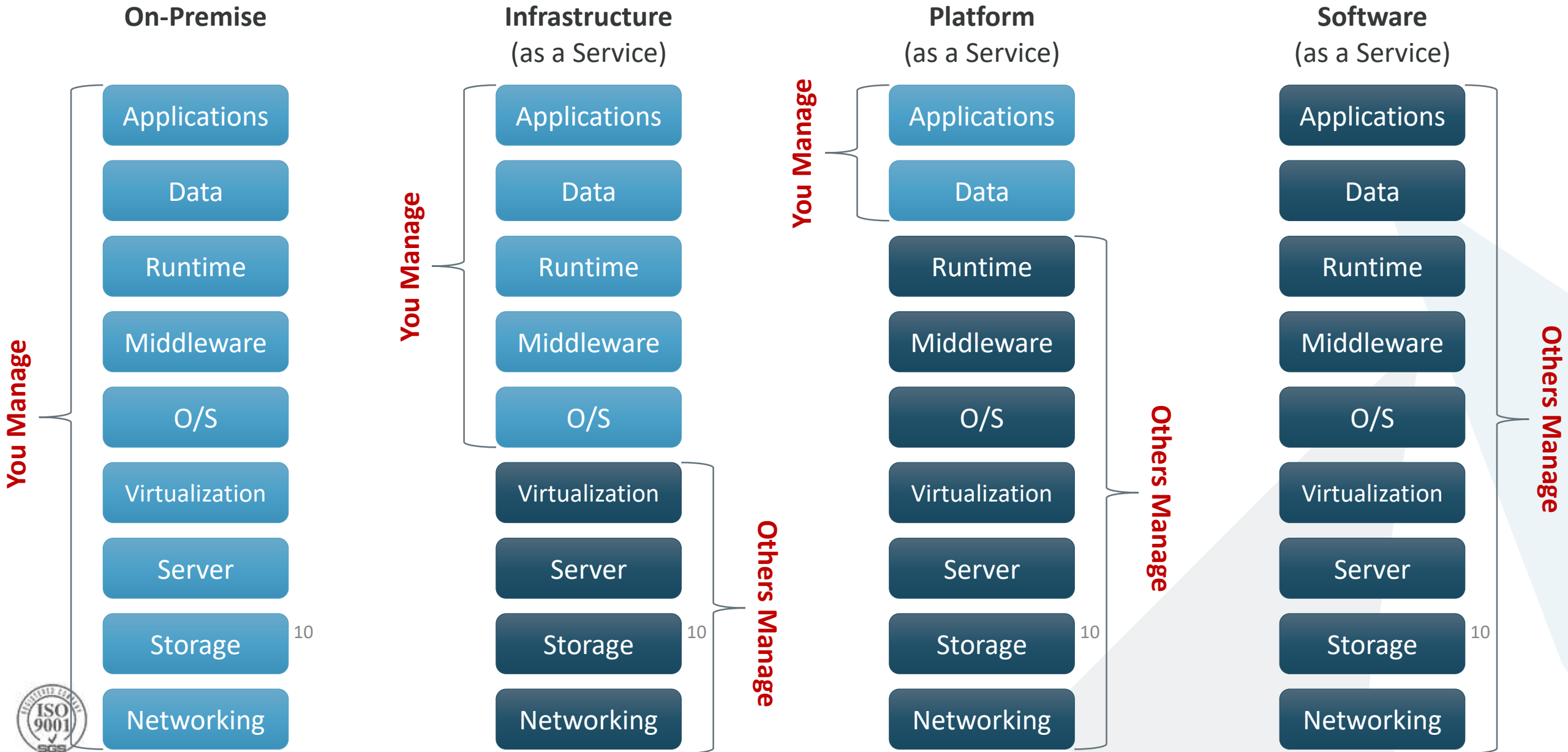


Less Worries

Focus on functionality

- Let others take care of:
 - Resource management
 - Security
 - Environments (staging, production)
 - High availability, scalability, load balancing
 - Fault tolerance
 - OS - installation, licensing, updates, patches
 - Network
 - Maintenance

Why the Cloud? - IaaS, PaaS & SaaS



➤ Introduction to Cloud Architecture

Cloud Computing - A Game Changing Technology

- Infinite shared resources & services
 - Infrastructure is not a limiting factor
 - No need to equip for peak-load requirements
- Elasticity on demand
 - Anytime, anywhere
- Efficient scalability and high availability
- Suitable pricing models
 - Pay for what you use
- **Less worries** about the foundation – **Better application quality!!!**

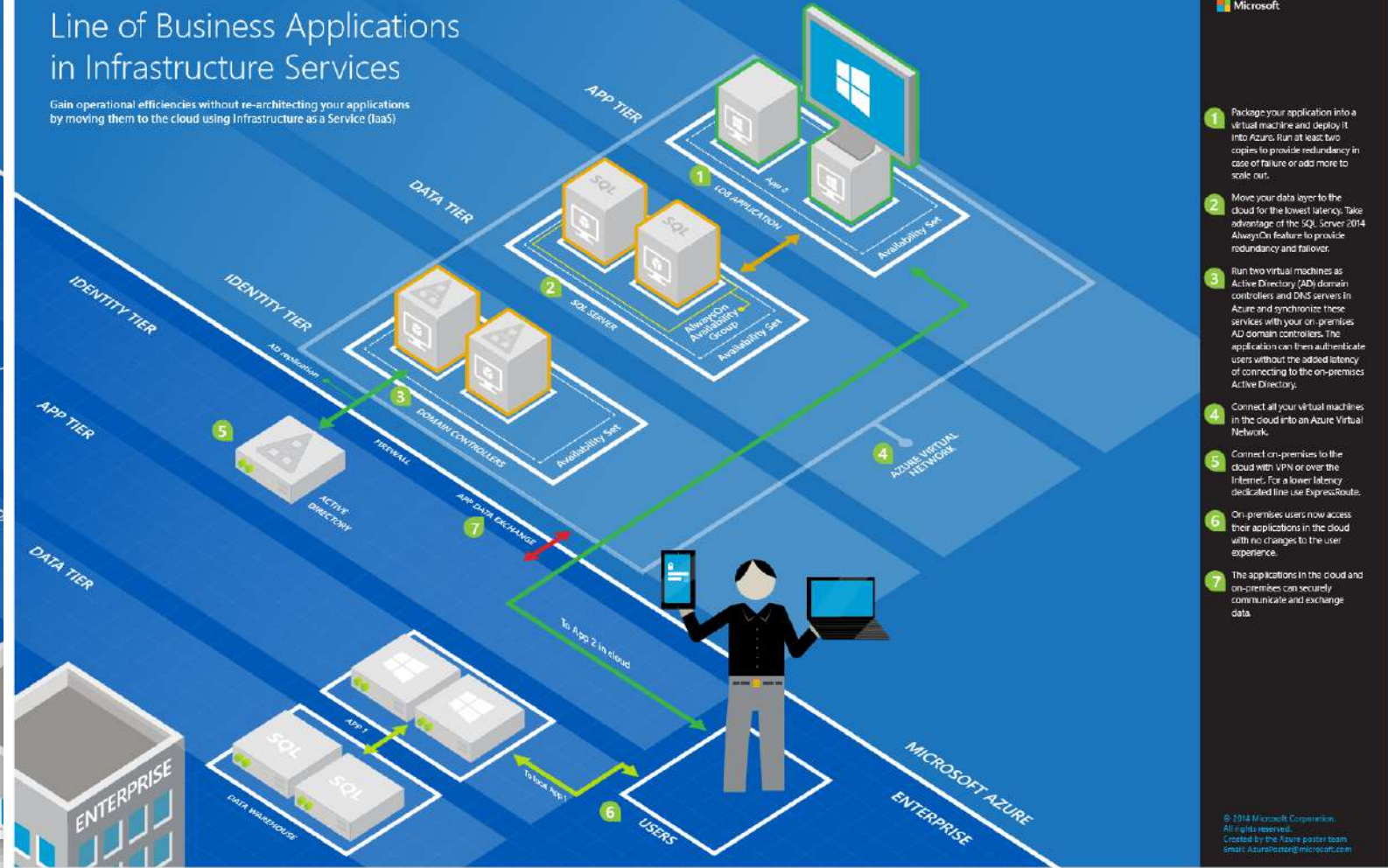




Azure Architecture Blueprints

Line of Business Applications in Infrastructure Services

Gain operational efficiencies without re-architecting your applications by moving them to the cloud using Infrastructure as a Service (IaaS).



© 2014 Microsoft Corporation. All rights reserved. Created by the Azure poster team. Email: Azureposter@microsoft.com

Microsoft Azure



Azure regions

Azure has more global regions than any other cloud provider—offering the scale needed to bring applications closer to users around the world, preserving data residency, and offering comprehensive compliance and resiliency options for customers.

50 regions worldwide **140** available in 140 countries



















Security & Management

-  Security Center
-  Portal
-  Azure Active Directory
-  Azure AD B2C
-  Multi-Factor Authentication
-  Automation
-  Scheduler
-  Key Vault
-  Store/Marketplace
-  VM Image Gallery & VM Depot

Platform Services

Media & CDN

-  Media Services
-  Media Analytics
-  Content Delivery Network








Integration

-  API Management
-  BizTalk Services
-  Logic Apps
-  Service Bus







Compute Services

-  Container Service
-  VM Scale Sets
-  Batch
-  RemoteApp
-  Dev/Test Lab

Application Platform

-  Web Apps
-  Mobile Apps
-  API Apps
-  Cloud Services
-  Service Fabric
-  Notification Hubs
-  Functions

Developer Services

-  Visual Studio
-  Mobile Engagement
-  VS Team Services
-  Xamarin
-  Application Insights
-  HockeyApp











Data

-  SQL Database
-  SQL Data Warehouse
-  DocumentDB
-  SQL Server Stretch Database
-  Redis Cache
-  Storage Tables
-  Azure Search

Intelligence

-  Cognitive Services
-  Bot Framework
-  Cortana

Analytics & IoT

-  HDInsight
-  Machine Learning
-  Stream Analytics
-  Data Catalog
-  Data Lake Analytics Service
-  Data Lake Store
-  IoT Hub
-  Event Hubs
-  Data Factory
-  Power BI Embedded

Hybrid Cloud

-  Azure AD Health Monitoring
-  AD Privileged Identity Management
-  Domain Services
-  Backup
-  Operational Analytics
-  Import/Export
-  Azure Site Recovery
-  StorSimple

Infrastructure Services

Compute

-  Virtual Machines
-  Containers

Storage

-  Blob
-  Queues
-  Files
-  Disks

Networking

-  Virtual Network
-  Load Balancer
-  DNS
-  Express Route
-  Traffic Manager
-  VPN Gateway
-  App Gateway

Datacenter Infrastructure



Smart Home - Microsoft

Securehttps://portal.azure.com/#@codevalue.onmicrosoft.com/dashboard/private/e0698042-da11-4170-bd65-7a7a2d8ee037

Search resources, services and docs

alongf@codevalue.netCODEVALUE LTD.

Microsoft Azure

Create a resource

All services

FAVORITES

Dashboard

Resource groups

All resources

Recent

App Services

SQL databases

Virtual machines (classic)

Virtual machines

Cloud services (classic)

Subscriptions

Azure Active Directory

Monitor

Security Center

Cost Management + Billing

Help + support

Advisor

Smart Home

New dashboard

Upload dashboard

Download dashboard

Edit dashboard

Share

Fullscreen

Clone

Delete

Jerusalem

15:53

WEDNESDAY, MARCH 21, 2018

Pacific Time (US ...)

06:53

WEDNESDAY, MARCH 21, 2018

watertank

AZURE IOT HUB

Active

Resource groups

ALL SUBSCRIPTIONS

Refresh

advabirthday	West Europe
alonfunctionappdemo	North Europe
alonmachineryg	West Europe
alonml	West Europe
AzureIoTHubSupport	North Europe
cloud-shell-storage-westeuropa	West Europe
croptimal	North Europe
deletemesoon	West Europe

See more...

3/21/2018 UTC

WATERTANK

5% TOTAL

MESSAGES

20661 / 400k

DEVICES

2

eventhubfortimeserie...

EVENTHUB

Active

Commands completed, Connected devices and 3 ...

WATERTANK

COMMANDS COMPLETED

0

CONNECTED DEVICES

0.9

watertankvalues

STREAMING JOB

watertank

FUNCTION APP

Running

watertank

TIMESERIESINSIGHTSENVIRO...

Data Conversion Errors, Input Events and 2 more ...

DATA CONVERSION ERR...

0

INPUT EVENTS

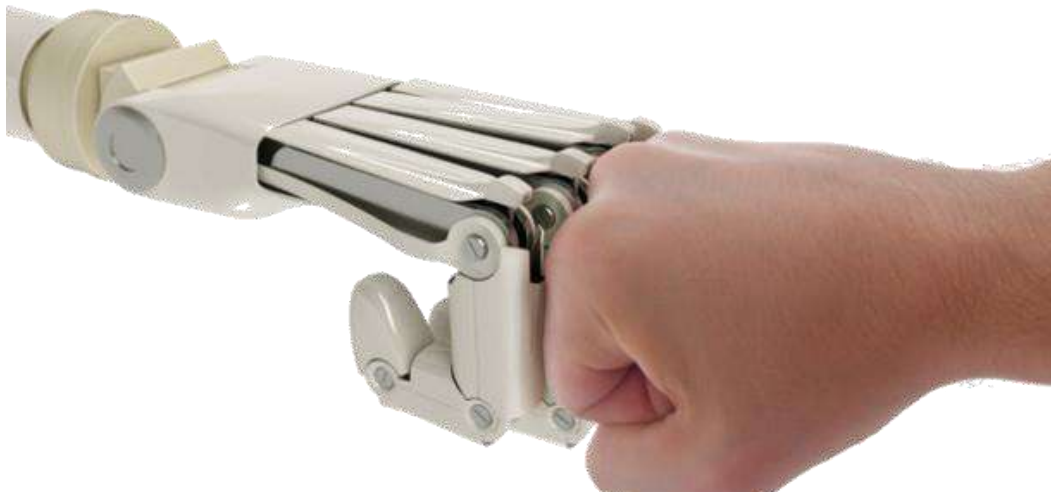
3.52k

watertankeventqueue



Automation is key

- Manual deployment, management and resources-definition can only take you so far
- Automate Azure with:
 - [Azure PowerShell](#)
 - [Azure CLI](#) (Windows, Mac, Linux)
 - Azure Resource Manager (ARM)



Command-Line Syntax Overview

prompt> **azure** **topic** **verb** **options**

account
account location
account affinity-group
vm
vm disk
vm endpoint
vm image
service
service cert
site
config

download
import
list
show
delete
start
restart
shutdown
capture
create
create-from
attach
attach-new
detach
browse
set

username
password
dns-prefix
vm-name
lb-port
target-image-name
source-path
disk-image-name
size-in-gb
thumbprint
value
-json
-v
-vv

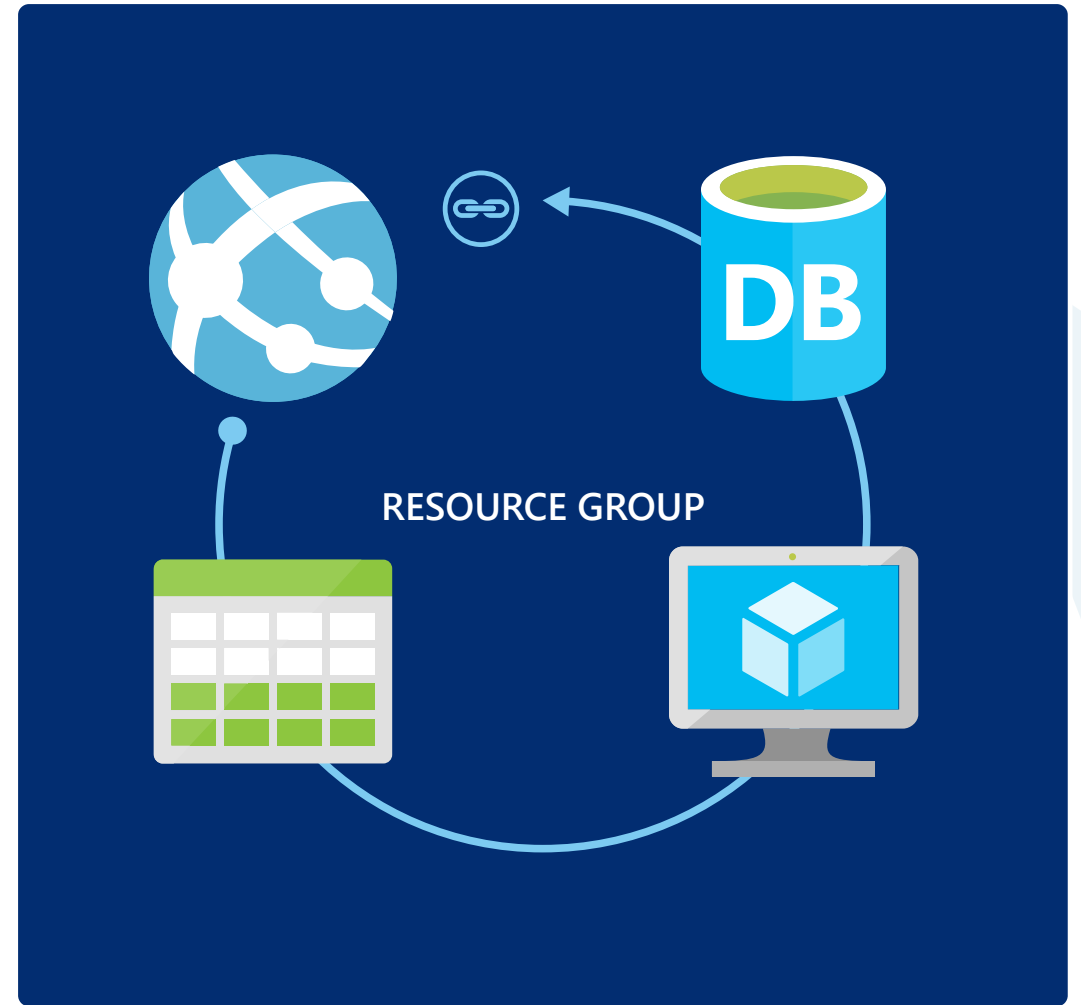
Azure Resource Manager (ARM)

- A holistic view of the entire cloud application resources
- Instead of looking at them as separate entities, they are part of a whole
- Benefits
 - Deploy, manage and monitor the entire solution group
 - Repeatedly deploy your solution
 - Declarative templates
 - Easy resource dependency management (ordered deployment)
 - Native RBAC support to all services in the resource group
 - Simple logical organization with tags
 - Clear billing



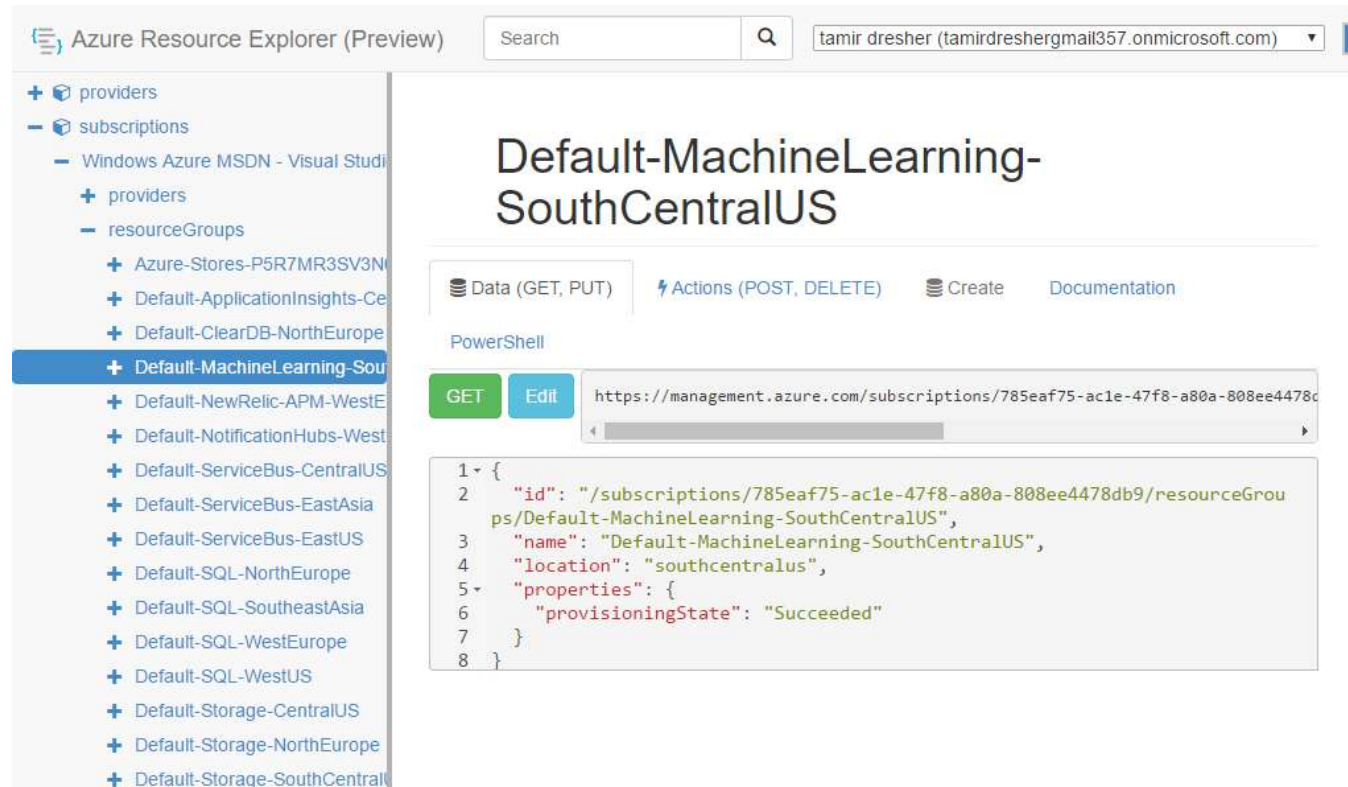
Resource Groups

- Tightly coupled containers of multiple resources
- Every resource exist in one (and only one) resource group
- Resource groups can span regions
- You can deploy, update, delete a resource group as a whole
- Easy to understand billing



Azure Resource Explorer

- resources.azure.com
- Resources management site
- “Explorer like” experience for browsing subscriptions, providers, resources and templates



Azure Resource Explorer (Preview)

Search

tamir dresher (tamirdresher@gmail357.onmicrosoft.com)

providers

subscriptions

Windows Azure MSDN - Visual Studio

providers

resourceGroups

Azure-Stores-P5R7MR3SV3N

Default-ApplicationInsights-Ce

Default-ClearDB-NorthEurope

Default-MachineLearning-Sou

Default-NewRelic-APM-WestE

Default-NotificationHubs-West

Default-ServiceBus-CentralUS

Default-ServiceBus-EastAsia

Default-ServiceBus-EastUS

Default-SQL-NorthEurope

Default-SQL-SoutheastAsia

Default-SQL-WestEurope

Default-SQL-WestUS

Default-Storage-CentralUS

Default-Storage-NorthEurope

Default-Storage-SouthCentral

Default-MachineLearning-SouthCentralUS

Data (GET, PUT) Actions (POST, DELETE) Create Documentation

PowerShell

GET Edit

<https://management.azure.com/subscriptions/785eaf75-ac1e-47f8-a80a-808ee4478db9/resourceGroups/Default-MachineLearning-SouthCentralUS>

```
1 {
2   "id": "/subscriptions/785eaf75-ac1e-47f8-a80a-808ee4478db9/resourceGroups/Default-MachineLearning-SouthCentralUS",
3   "name": "Default-MachineLearning-SouthCentralUS",
4   "location": "southcentralus",
5   "properties": {
6     "provisioningState": "Succeeded"
7   }
8 }
```




ARM cmdlets

- Execute and manage deployments
 - `New-AzureResourceGroupDeployment`
- Create individual resources
 - `New-AzureRmResource`
- Invoke specific actions on existing resources
 - `Invoke-AzureRmResourceAction`

```
New-AzureResource -Location "West US" -Properties @{"test"="test"} -ResourceName myTestSiteName -ResourceType  
microsoft.web/sites -ResourceGroupName myResourceGroup -Force
```




ARM Templates

- Declarative JSON files that specifies resource and their dependencies
- Idempotent
- Parametrized
- Source-control friendly

imperative

```
New-AzureVM -VM $myVM  
New-AzureRmStorageAccount -StorageAccountName $acct  
Set-AzureRmVNetConfig -ConfigurationPath -Path
```

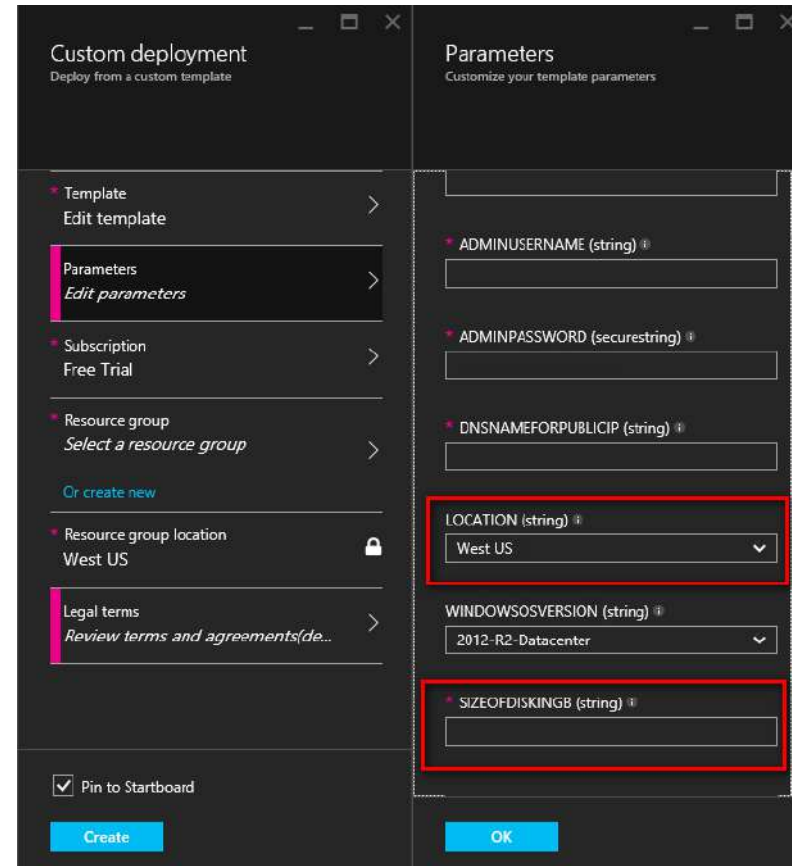
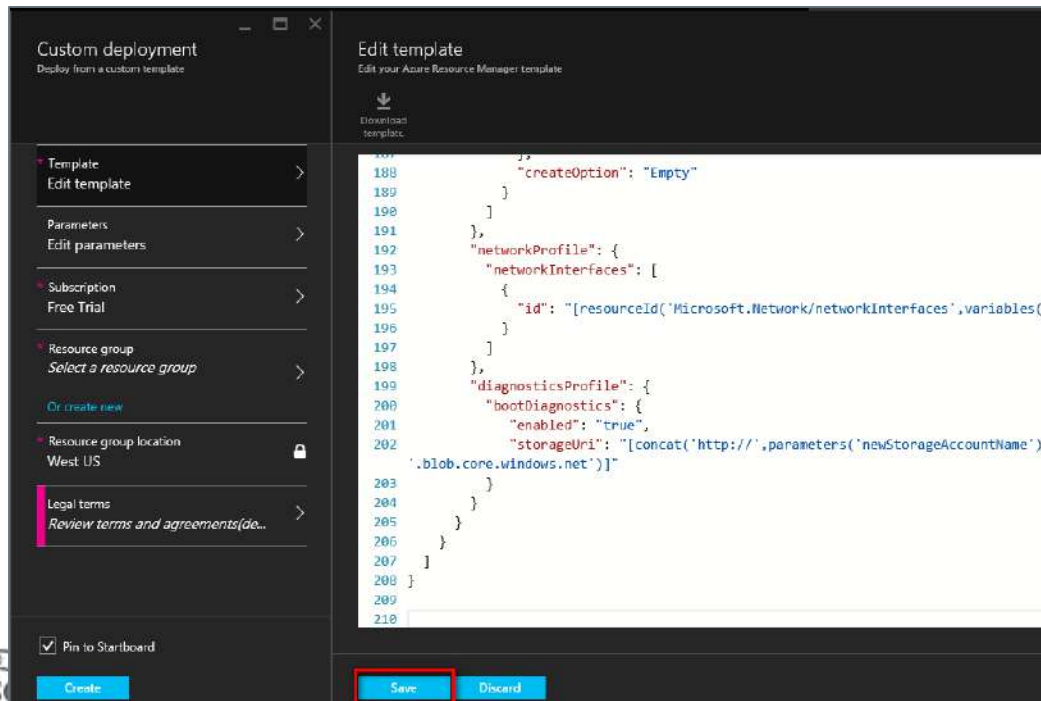
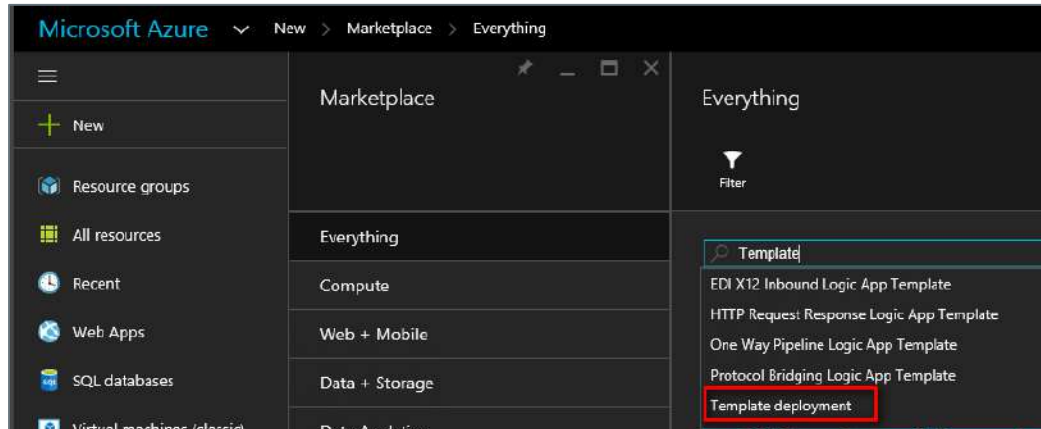


declarative

```
{  
  "$schema": "https://../deploymentTemplate.json#",  
  "contentVersion": "1.0.0.0",  
  "parameters": {},  
  "variables": {},  
  "resources": [],  
  "outputs": {}  
}
```




Deploying Templates

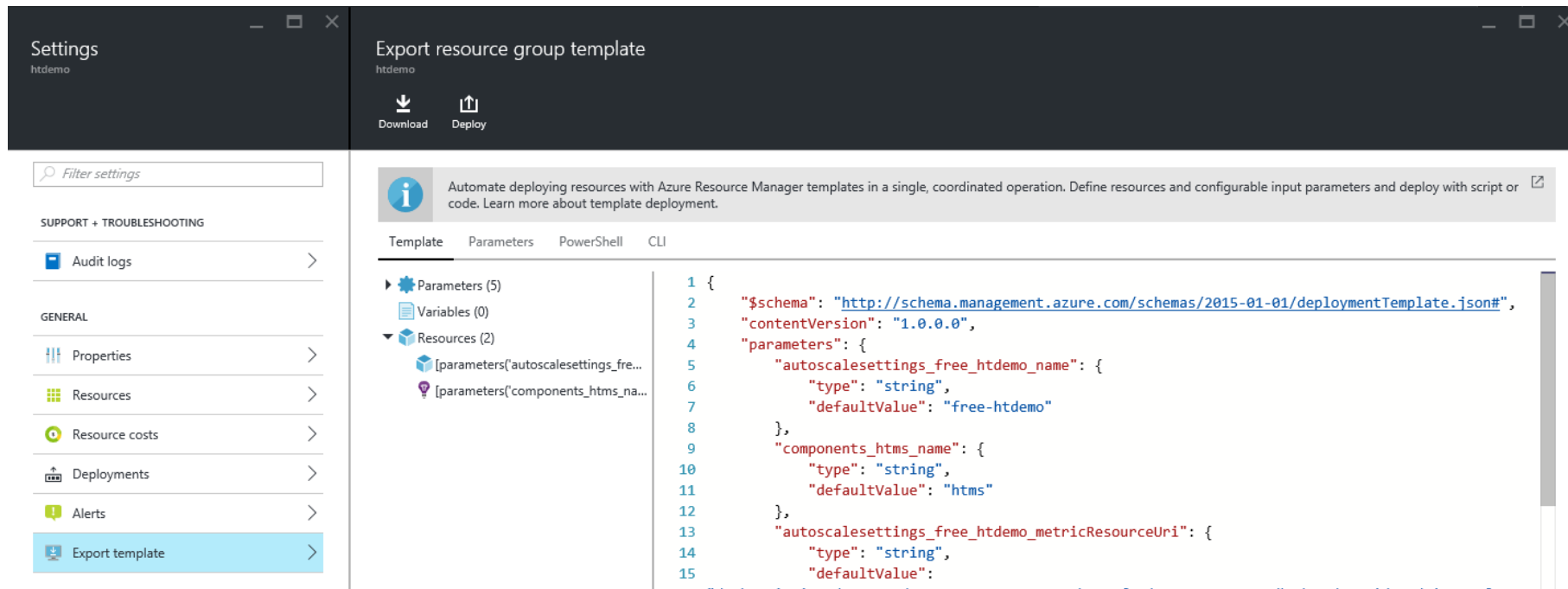


```
New-AzureRmResourceGroupDeployment -DeploymentName  
"Simple-VM" -ResourceGroupName  
RG-AZITCAMP -TemplateFile "C:\GitHub\Templates\101-  
simple-windows-vm\azuredeploy.json
```



Creating\Editing Templates

- Visual Studio
- Visual Studio Code
- <http://armviz.io/>
- [Export Resource Group Templates](#)



Settings
htdemo

Export resource group template
htdemo

Download Deploy

Filter settings

SUPPORT + TROUBLESHOOTING

Audit logs

GENERAL

Properties

Resources

Resource costs

Deployments

Alerts

Export template

Automate deploying resources with Azure Resource Manager templates in a single, coordinated operation. Define resources and configurable input parameters and deploy with script or code. Learn more about template deployment.

Template Parameters PowerShell CLI

Parameters (5)

Variables (0)

Resources (2)

[parameters('autoscalesettings_fre...]

[parameters('components_htms_na...]

```
1 {
2   "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "autoscalesettings_free_htdemo_name": {
6       "type": "string",
7       "defaultValue": "free-htdemo"
8     },
9     "components_htms_name": {
10      "type": "string",
11      "defaultValue": "htms"
12    },
13     "autoscalesettings_free_htdemo_metricResourceUri": {
14      "type": "string",
15      "defaultValue": "https://api.monitor.azure.com/metrics/autoscalesettings_free_htdemo_metricResourceUri"
```


Azure Resource Group Project

The image shows a screenshot of the Visual Studio IDE with three overlapping windows:

- Add New Project:** The 'Recent' tab is selected. Under 'Visual C#', the 'Web' category is expanded, and 'Azure Resource Group' is highlighted. The 'Type: Visual C#' filter is applied.
- Select Azure Template:** A list of templates is shown, including 'DocumentDB', 'Windows Virtual Machine Scale Set', 'Linux Virtual Machine Scale Set', 'Service Fabric Cluster', 'Web app', 'Web app + SQL', 'Windows Virtual Machine', and 'Windows Server Virtual Machines with Load Balancer'. The 'DocumentDB' template is selected, showing its icon and description: 'Allows you to create a new Azure DocumentDB Account. VERSION: 2015-04-08'.
- Solution Explorer:** The 'MyAmazingWebApp' project is selected. The file list shows 'References', 'Scripts' (containing 'Deploy-AzureResourceGroup.ps1'), and 'Templates' (containing 'WebSiteSQLDatabase.json' and 'WebSiteSQLDatabase.parameters.json').



MyAmazingWebApp - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test OzCode ReSharper Analyze Window Help

Debug Any CPU Start

TOTU_Menu.pdf Database.json WebSiteSQLData...arameters.json

Schema: <http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#>

```
1 {
2   "$schema": "http://schema.management.azure.com/sch
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "hostingPlanName": {
6       "type": "string",
7       "minLength": 1
8     },
9     "skuName": {
10      "type": "string",
11      "defaultValue": "F1",
12      "allowedValues": [
13        "F1",
14        "D1",
15        "B1",
16        "B2",
17        "B3",
18        "S1",
19        "S2",
```

JSON Outline

- parameters (10)
- variables (2)
 - webSiteName
 - sqlserverName
- resources (9)
 - SqlServer
 - HostingPlan
 - Website
 - connectionstrings
 - AutoScaleSettings
 - ServerErrorsAlertRule
 - ForbiddenRequestsAlertRule
 - CPUHighAlertRule
 - AutoScaleSettings
 - AppInsightsComponent

How do I deploy project artifacts with an Azure deployment template?

Performance Explorer JSON Outline

Unit Test Sessions Package Manager Console Output Find Results 1 Azure App Service Activity

Ready

↑ Publish

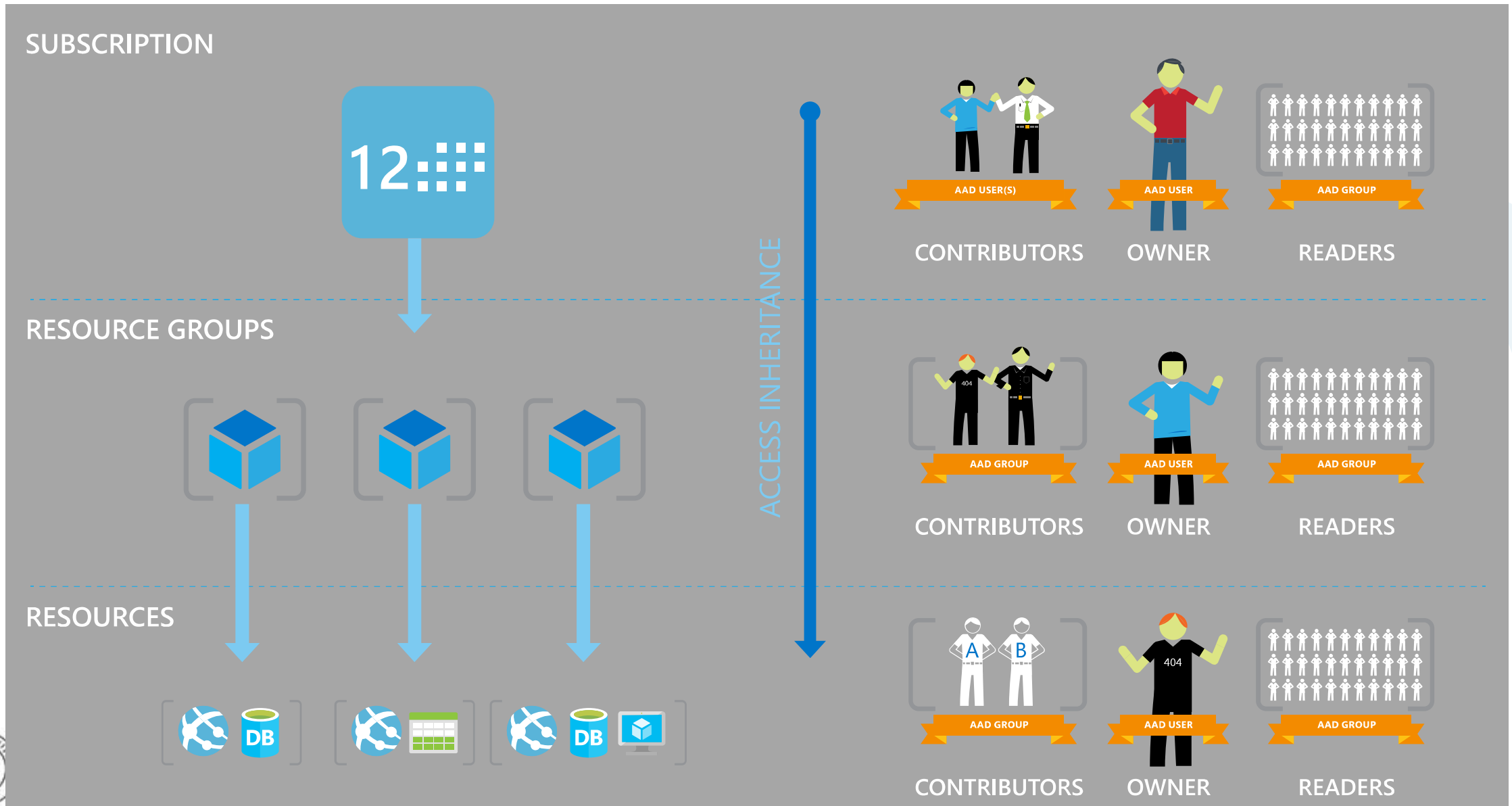
More Azure Deployment Links

- <https://github.com/Azure/azure-quickstart-templates/>
- <https://azure.microsoft.com/en-us/documentation/templates/>
- <http://azure.microsoft.com/en-us/documentation/articles/resource-group-overview/>
- <https://azure.microsoft.com/en-us/documentation/articles/best-practices-resource-manager-design-templates/>
- <https://github.com/Azure/azure-content/blob/master/articles/app-service-web/app-service-deploy-complex-application-predictably.md>

Role Based Access Control

- Allows secure access with granular permissions
- Assignable to users, groups, or service principals
- Built-in roles make it easy to get started
- Key Concepts:
 - Role Definitions – the set of permissions
 - Role Assignments - associate role definitions with an identity
 - Identity == user/group
 - Assignment is per scope (Directory/Subscription/Resource Group/Resource)
 - Inherited – subscription assignments apply to all resources

Role Based Access Control



/subscriptions/{id}/resourceGroups/{name}/providers/.../sites/{site}

subscription level – grants
permissions to all resources
in the sub

resource group level – grants
permissions to all resources
in the group

resource level – grants
permissions to the specific
resource



Built-in Roles

Role name	Permissions
Owner	Full management rights
Contributor	Full management rights except for user management
Reader	View resources and their settings
None	Does not see resources



RBAC in the portal

Settings

Add

Columns

Delete

Refresh

Essentials ^

Subscription name
Windows Azure MSDN - Visual Studio Ultim..

Subscription ID
785eaf75-ac1e-47f8-a80a-808ee4478db9

Last deployment
7/1/2016 (Succeeded)

Location
West Europe

All settings →

Filter items...

NAME	TYPE	LOCATION
myamazingstorage	Storage accou...	West Europe ...

Users
internalcourse-resourcemanagement

Add

Roles

USER	ROLE	ACCESS
exampleapp	Reader	Inherited ..
Subscription admins ⓘ	Owner	Inherited ..
tamir.dresher.reader@outlook.com	Reader	Assigned ..

The background of the slide features two hot air balloons floating in a bright blue sky filled with soft, white clouds. The balloon on the left is smaller and has a blue and white checkered pattern. The balloon on the right is larger and has a warm-toned pattern of orange, yellow, and red. Both balloons have small baskets hanging from them.

Azure Virtual machines

Azure Virtual Machines

- Launch Windows Server and Linux in minutes
- Scale from 1 to 1000s of VM Instances
- Save money with per-minute billing
- Open and extensible





Provisioning VMs

Getting Started



Management Portal

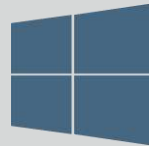


Scripting
(Windows, Linux and Mac)



REST API

Select Image and VM Size



Windows Server



Linux

General Purpose

Basic

Standard

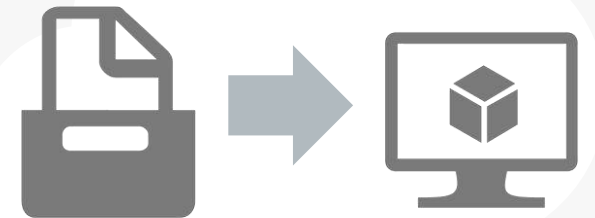
Optimized Compute

Performance Optimized

Network Optimized

New Disk Persisted in Storage

Boot VM from New Disk



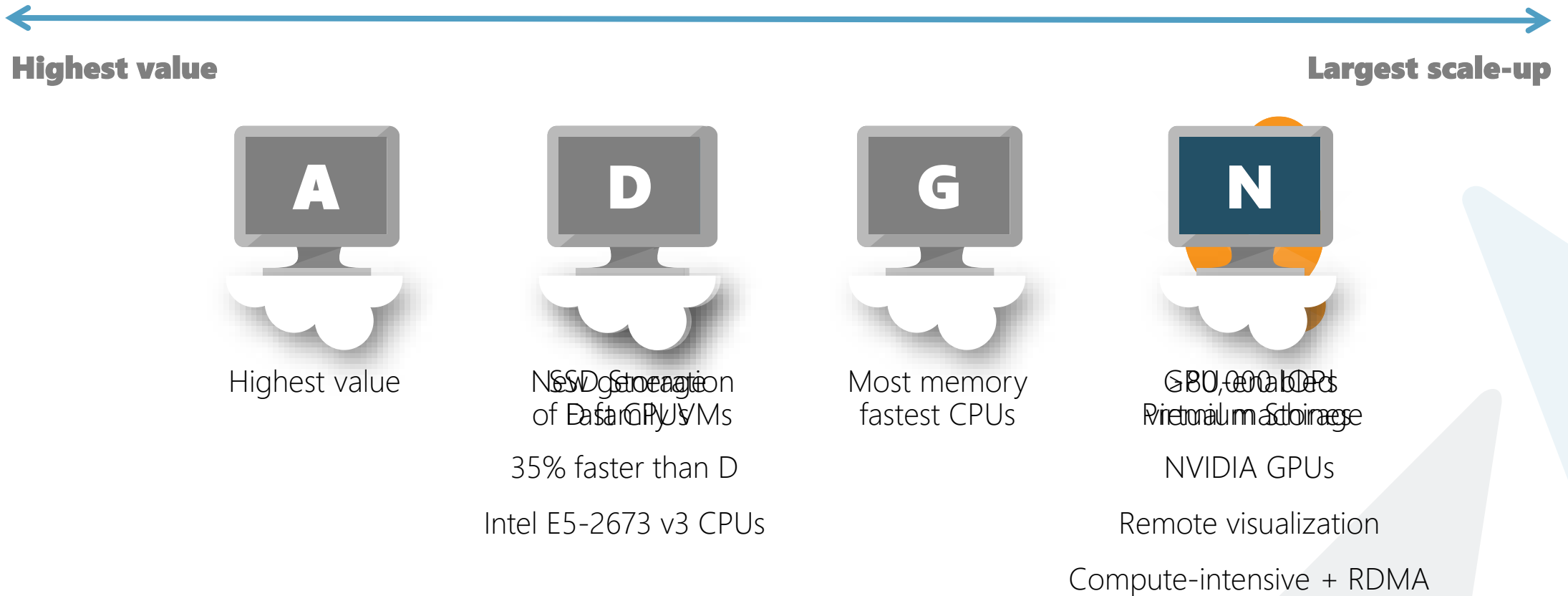
Cloud

► A collection of prebuilt images for various workloads





Scale Up Options




The G family




Optimized for data workloads
Up to 32 CPU cores, 448 GB RAM
6.5 TB local SSD
Latest generation Intel processor
Up to 64 attached disks!!

Dell PowerEdge R930 Rack Server

Starting Price.....\$66,621.00
Instant Savings.....\$20,229.13

Subtotal.....\$46,391.87
As low as **\$1,392.00 /month^**

 [Dell Business Credit | Apply](#)

 [Discount Details](#)
 [Ships in 11 - 15 Business Days](#)
 [Print Summary](#)

Processor Thermal Configuration	2 CPU	edit
PCIe Riser	None	edit
Cooling	None	edit
Memory DIMM Type and Speed	2133MT/s RDIMMs	edit
Memory Configuration Type	Performance Optimized	edit
Memory Capacity	16GB RDIMM, 2133MT/s, Dual Rank, x4 Data Width	edit
RAID Configuration	No RAID for H330/H730P (1-24 HDDs)	edit
RAID Controller	PERC H730P Adapter RAID Controller, 2Gb NV Cache	edit
Hard Drives	800GB Solid State Drive SAS Read Intensive MLC 12Gbps 2.5in Hot-plug Drive	edit
Network Daughter Card	Broadcom 5720 Quad Port 1Gb Network Daughter Card	edit
Additional Network Cards	None	edit

The background of the slide features a photograph of two hot air balloons. One balloon, on the left, has a blue and white checkered pattern. The other, on the right, is primarily orange with a red base. They are floating against a bright blue sky filled with soft, white, wispy clouds. The word "Demo" is centered in a large, bold, blue font.

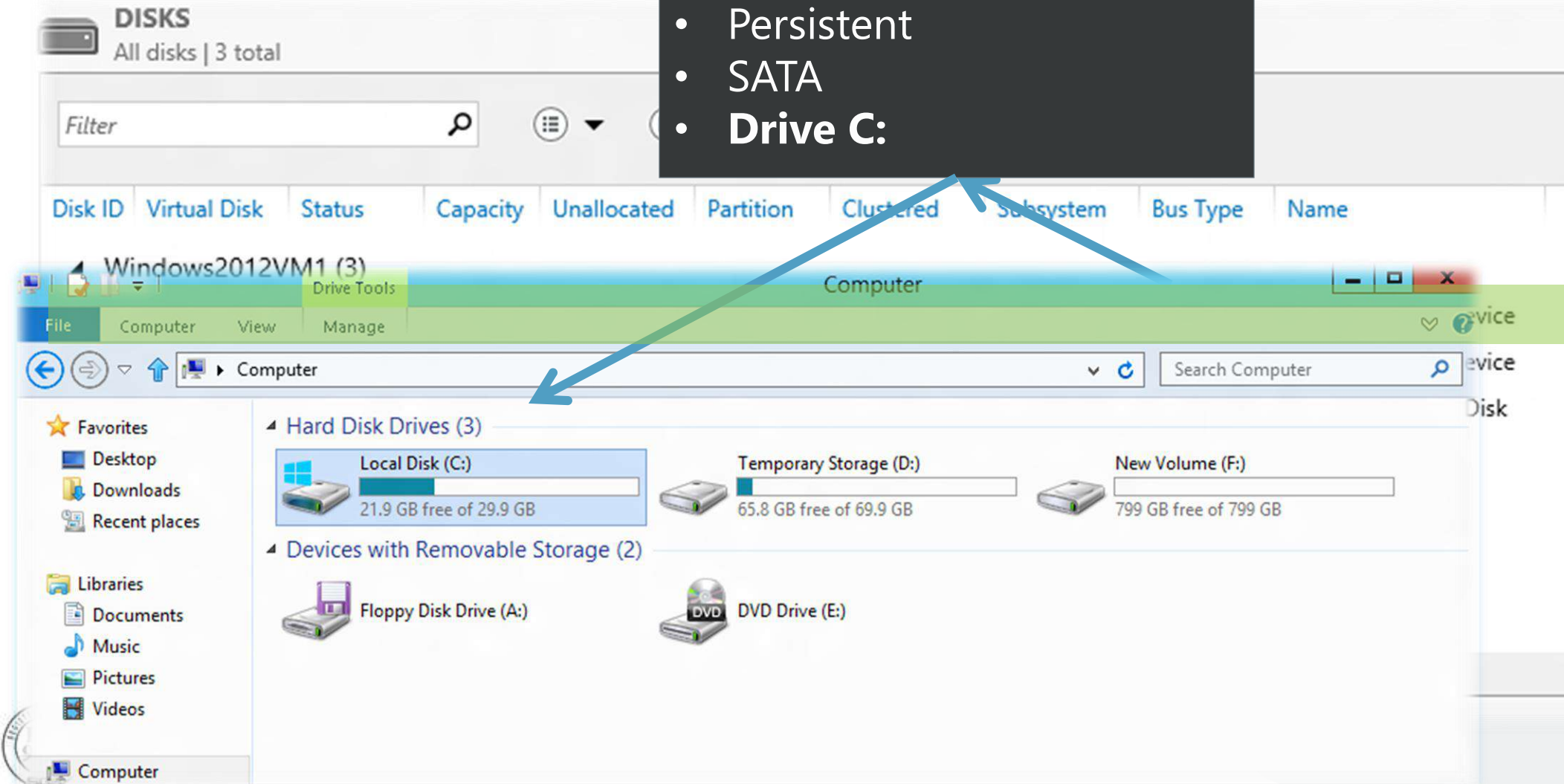
Demo

Creating a VM in the Portal

VM Disk Layout

OS Disk

- Persistent
- SATA
- **Drive C:**





Azure Disks

- Backed by Page BLOBs
 - Stored in an Azure Storage account
- Mounted by a single VM at a time
 - Lease is taken on BLOB itself
- Can be backed by Standard or Premium storage
 - Standard – up to 500 8KB IOPS per disk (60 MB/s)
 - Premium – up to 5000 8KB IOPS per disk (200 MB/s)
- Max disk size (per disk) – 1 TB
- Pay for actual storage only
 - Not paying for capacity itself!
- Premium storage can only be attached to DS & GS machines

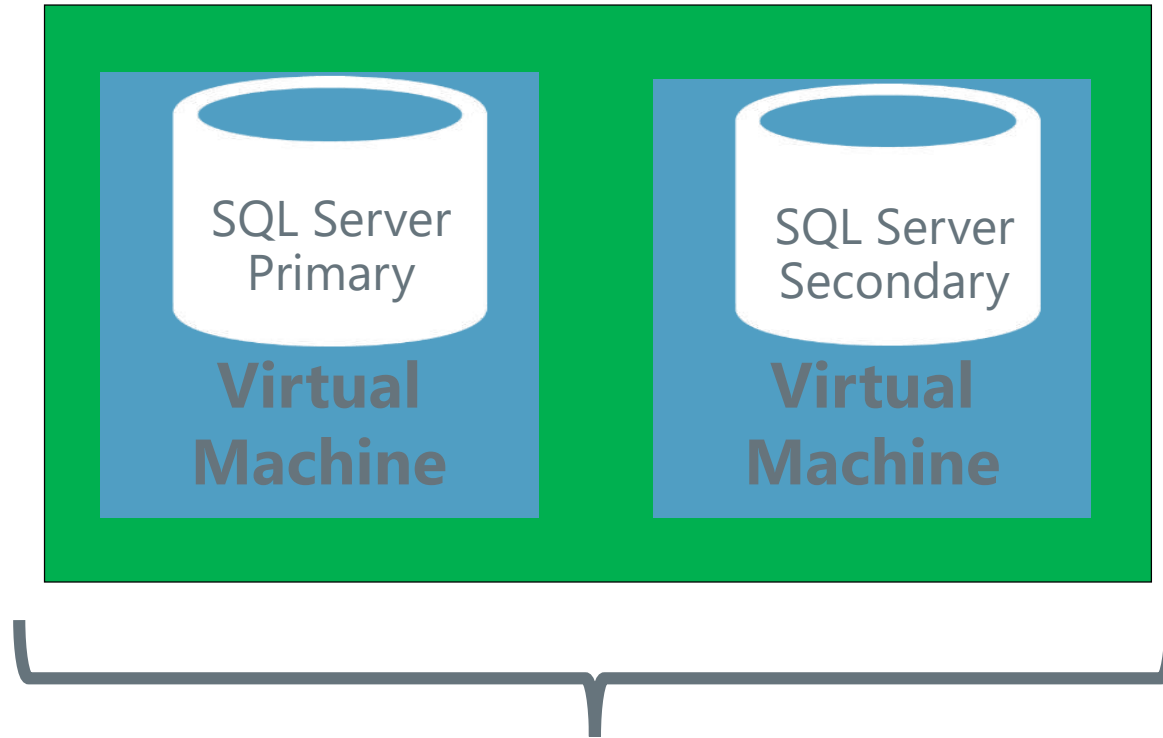


Azure Files

- Shared Network File Storage for Azure
 - File Share as a Service
- Can be mounted by multiple VMs concurrently
- Availability, durability, scalability are managed automatically
- Supports two interfaces: SMB and REST
 - Individual files stored in share are accessible via REST
- Max file share storage (per share) – 5 TB
- Max file size (within share) – 1 TB
- Scale limit – up to 1000 8KB IOPS (up to 60 MB/s per file share)

Availability Sets

Availability set



SLA 99.95

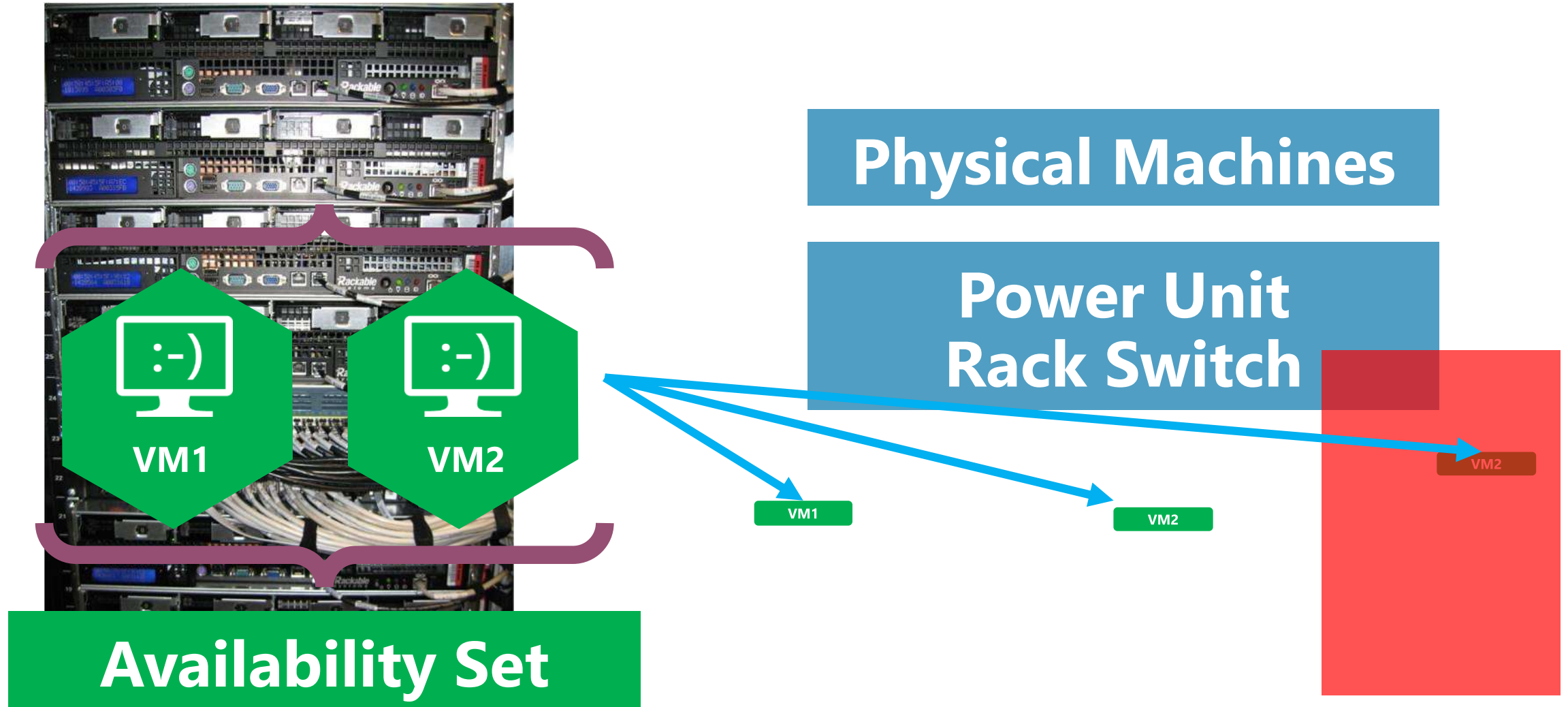
Availability Sets



Physical Machines

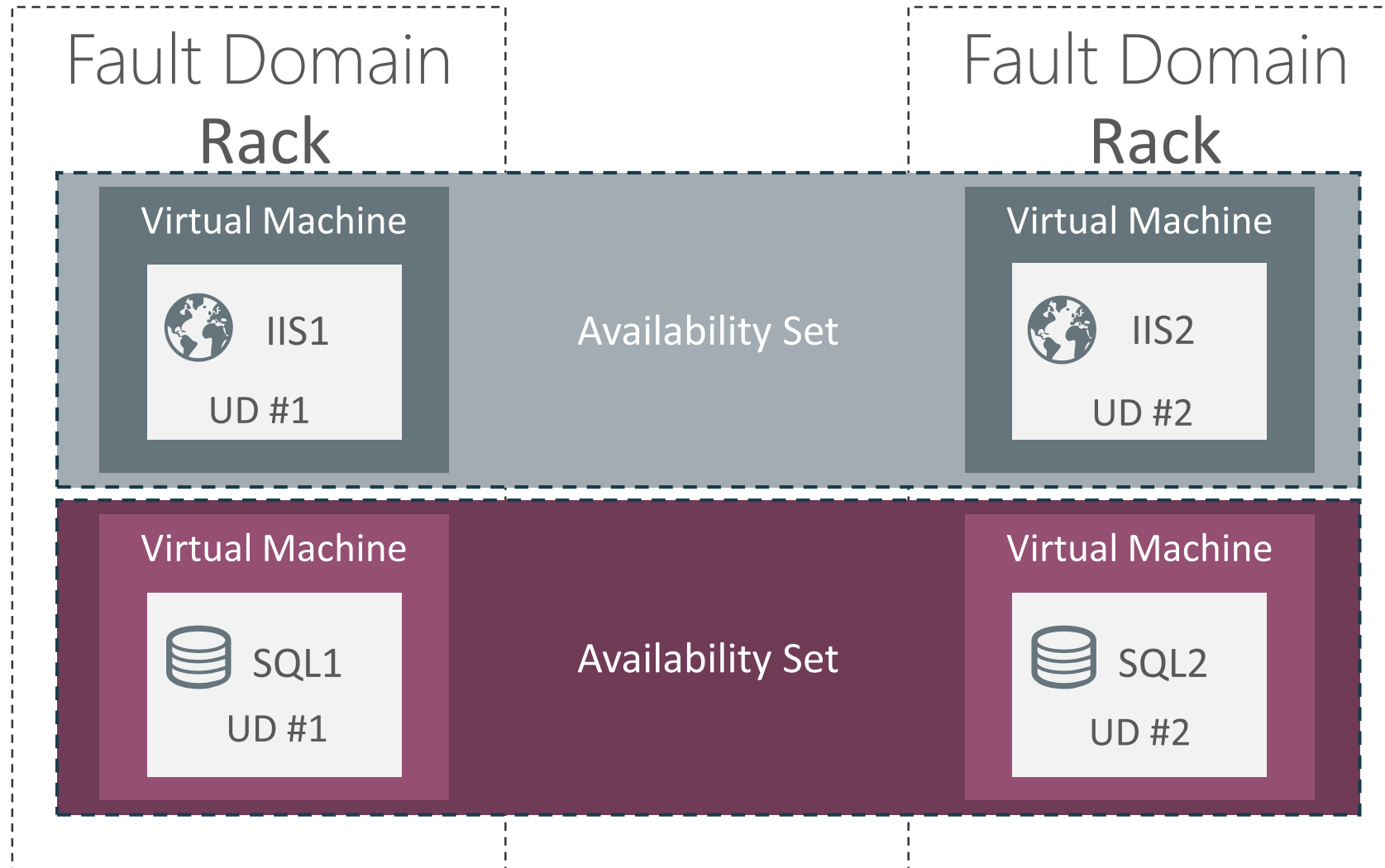
**Power Unit
Rack Switch**

Availability Sets



Virtual Machine Availability Sets

Update Domains are honored by host OS updates



Cloud Scale Compute Patterns

- Pets: Named resources with unique characteristics
- Cattle: Numbered, inherently replaceable, interchangeable



“Future application architectures should use Cattle but Pets with strong configuration management are viable and still needed.” – Gavin McCance (CERN)

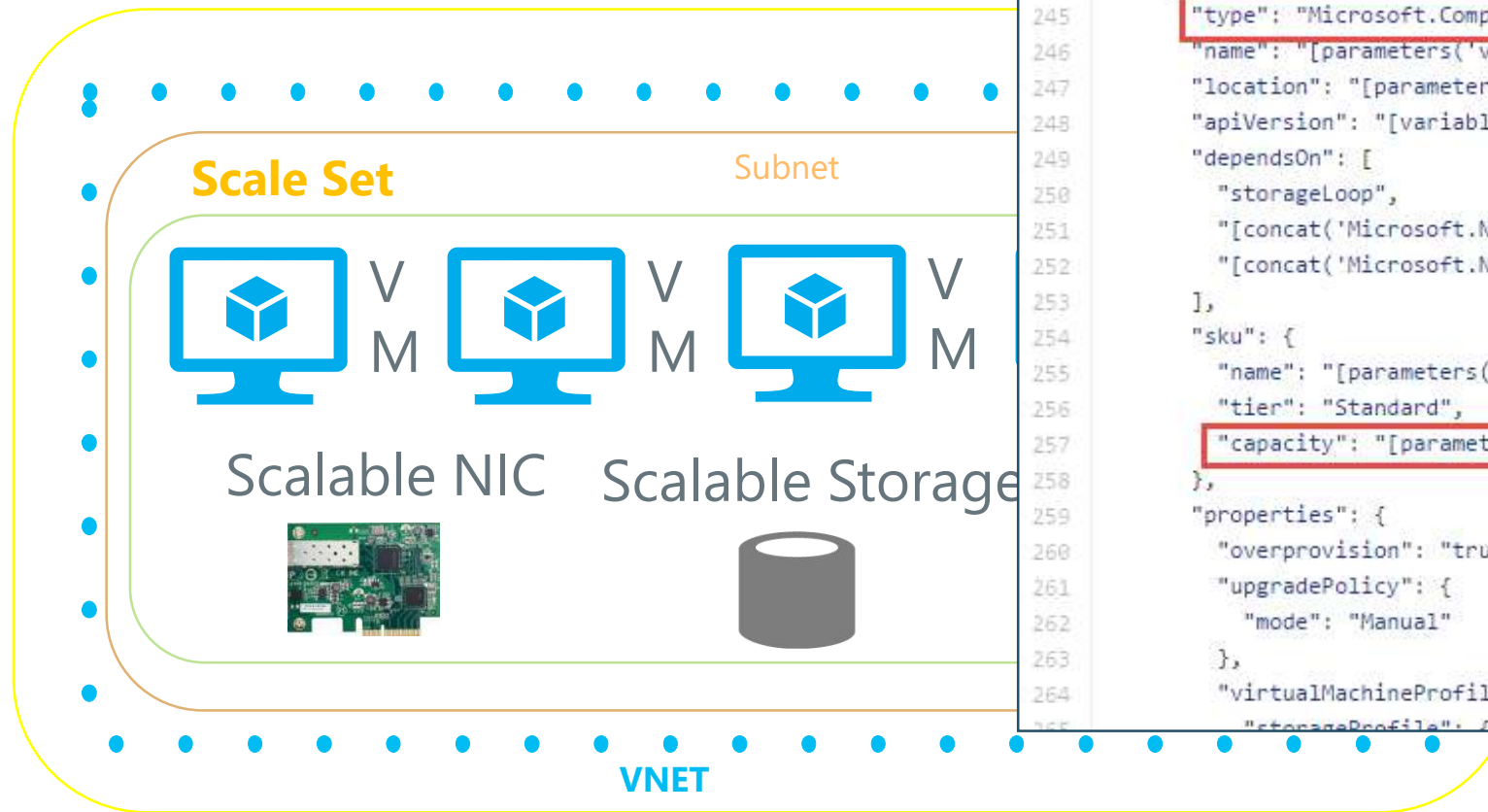
Virtual Machine Scale Sets

- Deploy and manage VMs as a set
- An Azure Compute resource
 - Microsoft.Compute/virtualMachineScaleSets
- Integrated with Azure Autoscale
- Integrated with Azure Load Balancer



VM Scale Sets in ARM

Resource Group



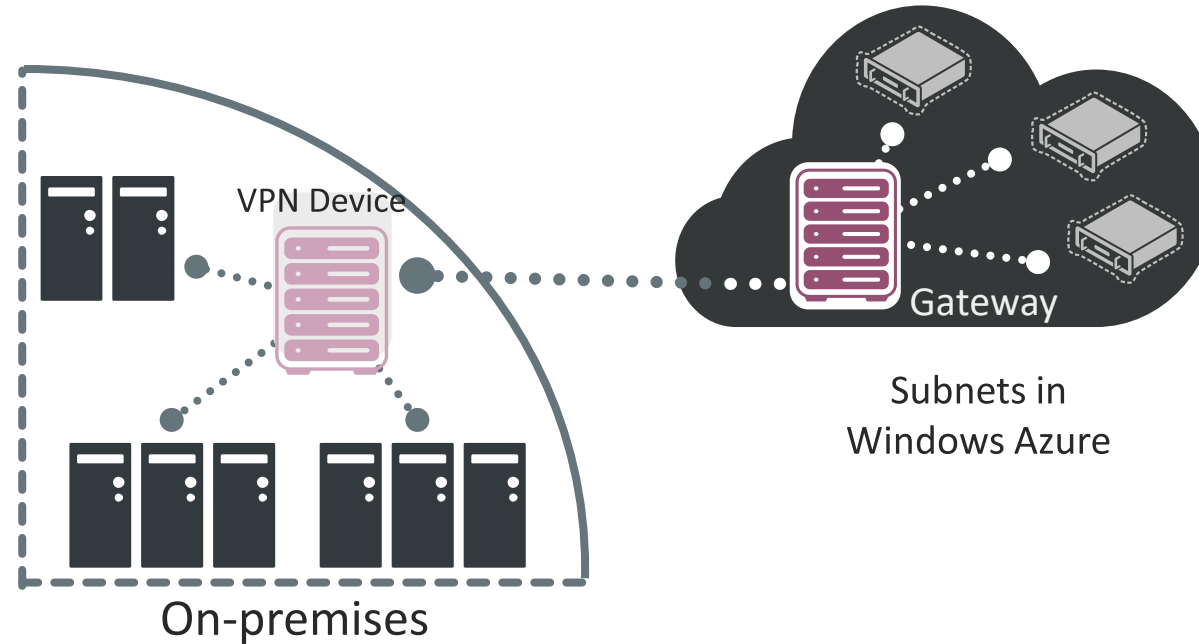
```
244 {  
245   "type": "Microsoft.Compute/virtualMachineScaleSets",  
246   "name": "[parameters('vmssName')]",  
247   "location": "[parameters('resourceLocation')]",  
248   "apiVersion": "[variables('computeApiVersion')]",  
249   "dependsOn": [  
250     "storageLoop",  
251     "[concat('Microsoft.Network/loadBalancers/', variables('loadBalancerName'))]",  
252     "[concat('Microsoft.Network/virtualNetworks/', variables('virtualNetworkName'))]"  
253   ],  
254   "sku": {  
255     "name": "[parameters('vmSku')]",  
256     "tier": "Standard",  
257     "capacity": "[parameters('instanceCount')]"  
258   },  
259   "properties": {  
260     "overprovision": "true",  
261     "upgradePolicy": {  
262       "mode": "Manual"  
263     },  
264     "virtualMachineProfile": {  
265       "storageProfile": {
```


Network Interface Cards (NICs)

- NICs allow VMs to communicate with the world
- Each VM has at least one NIC attached
 - Can have more than one
- A single NIC is associated with a specific subnet
 - Receives a private IP address
 - Can be static or dynamic
- A single NIC can be associated with a public IP address
 - Not mandatory
 - A public IP is a separate entity
 - Can be static or dynamic
- Each NIC is associated with a Network Security Group

Azure Virtual Networks

- A protected private virtual network in cloud
- Extend enterprise networks into Azure
- Cross-premises connectivity



Summary

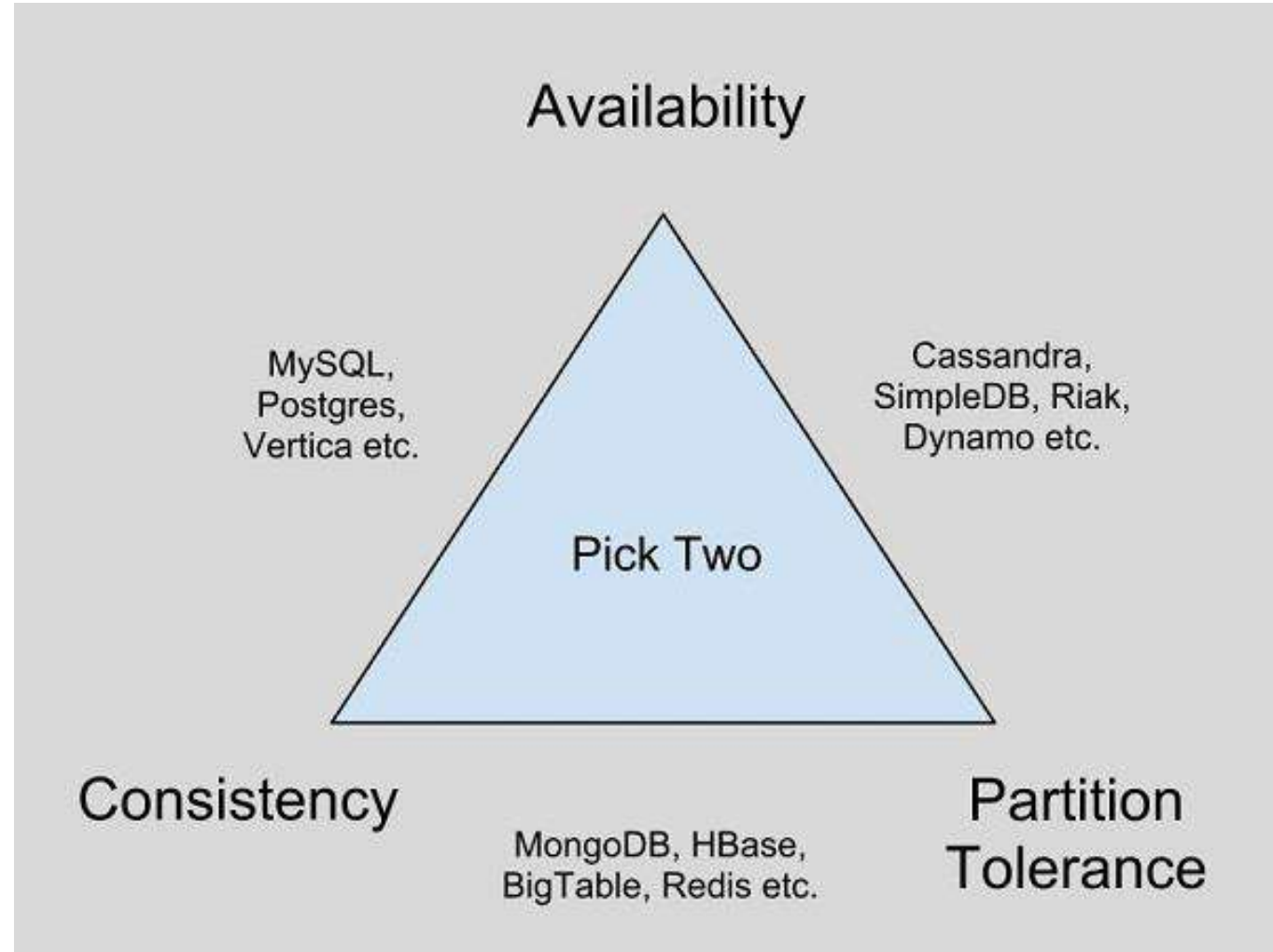
- Azure VMs can be provisioned on various tiers and sizes
- Data disks can be striped for extra performance
- Azure Files can be used for file share as a service
- Temporary disks are, well... temporary!
- Availability sets provide 99.95% uptime SLA
- NICs and Virtual Networks govern VM connectivity
- Network Security Groups can be used to configure network firewalls

Azure Storage Options

Relational					Files	Search
<ul style="list-style-type: none">• Azure SQL Server• SQL Server• Postgres• MySQL• Oracle• SQLite	<ul style="list-style-type: none">• Azure Blob Storage• Azure Table Storage• CosmosDB• Redis• Memcached• Riak	<ul style="list-style-type: none">• Cassandra• HBase	<ul style="list-style-type: none">• Cosmos DB (previously Document DB)• MongoDB• RavenDB• CouchDB	<ul style="list-style-type: none">• Cosmos DB• Neo4J	<ul style="list-style-type: none">• Azure Blob• Azure File Storage (SMB)	<ul style="list-style-type: none">• Elasticsearch• Azure Search

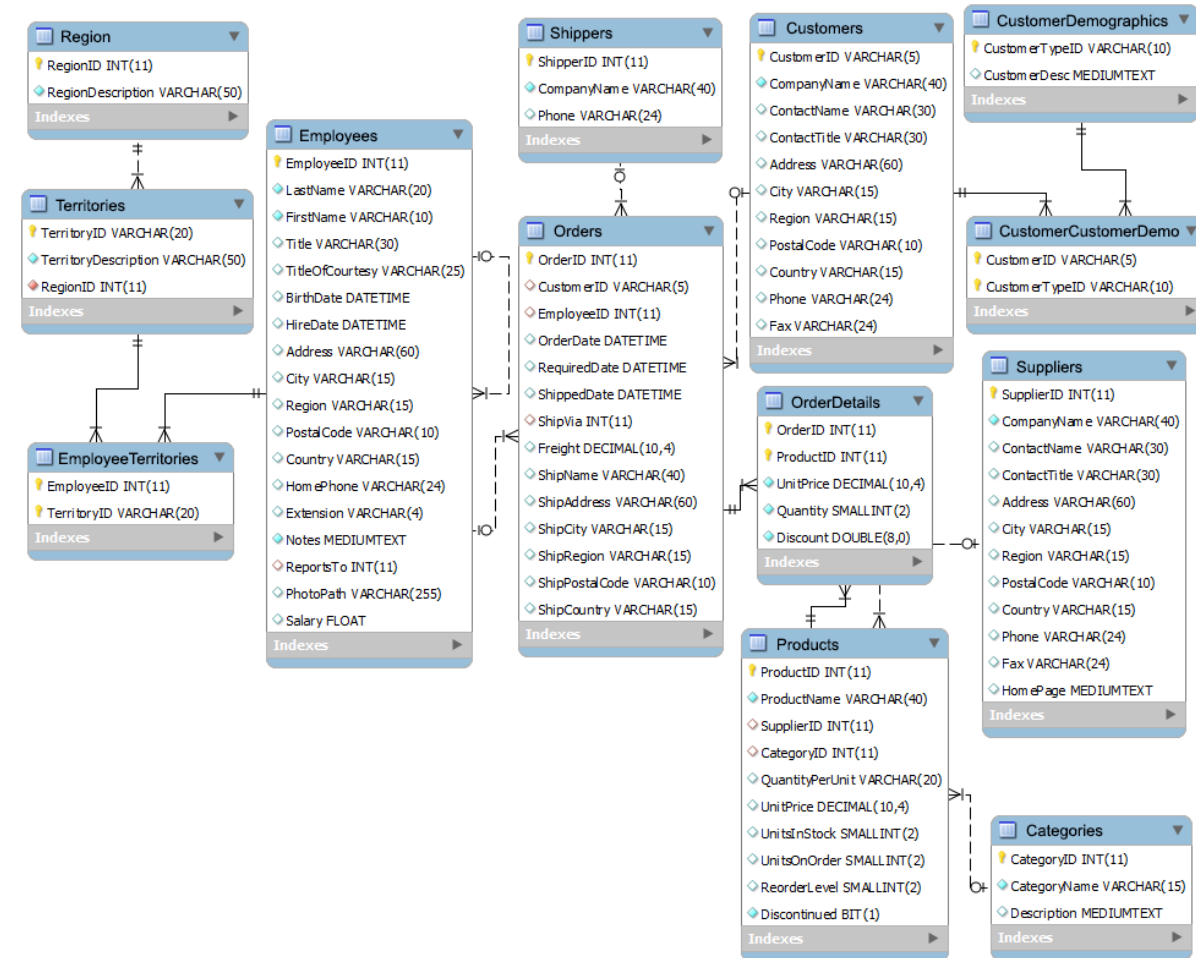
+ **Azure Queues**, which are part of the [Azure storage](#) infrastructure

CAP Theorem

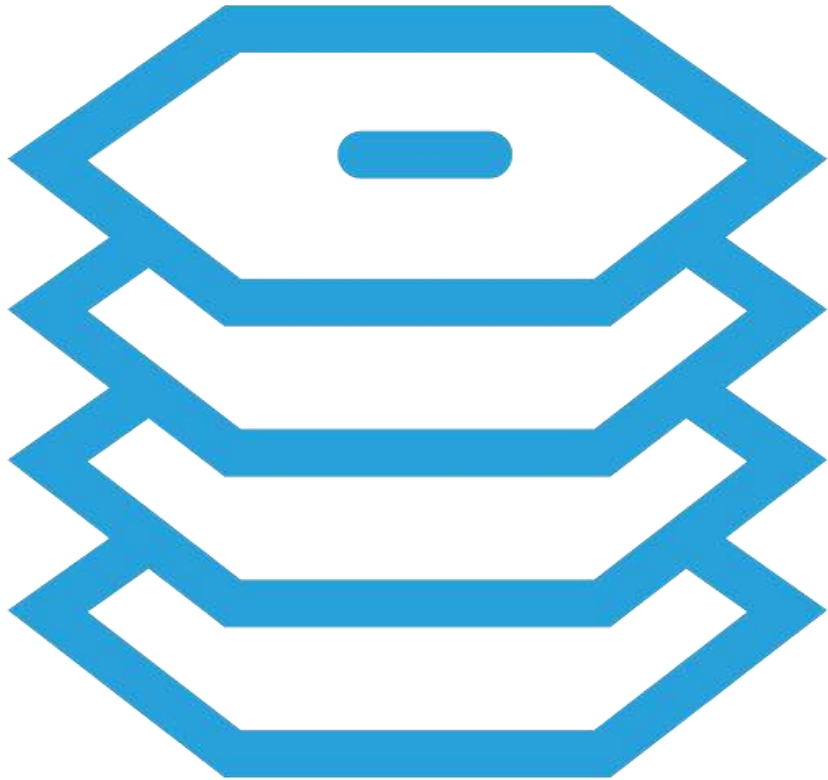


- Relational model exist for many years
- Tables, columns, relationships and constraints
- Azure PaaS solutions:

- MS SQL Server
- PostgreSQL
- MySQL



Azure SQL - Basics



SQL Database

SQL Server database technology as a service

Fully Managed

Enterprise-ready with automatic support for HA

Designed to scale out elastically with demand

Ideal for simple and complex applications

Feature comparison with SQL Server –

<https://docs.microsoft.com/en-us/azure/sql-database/sql-database-features>

Server Provisioning

Server Defined

Service head that contains databases

Connect via automatically generated FQDN (xxx.database.windows.net)

Initially contains only a **master** database

Provision Servers Interactively

Log on to Windows Azure Management Portal

Create a SQL Database server

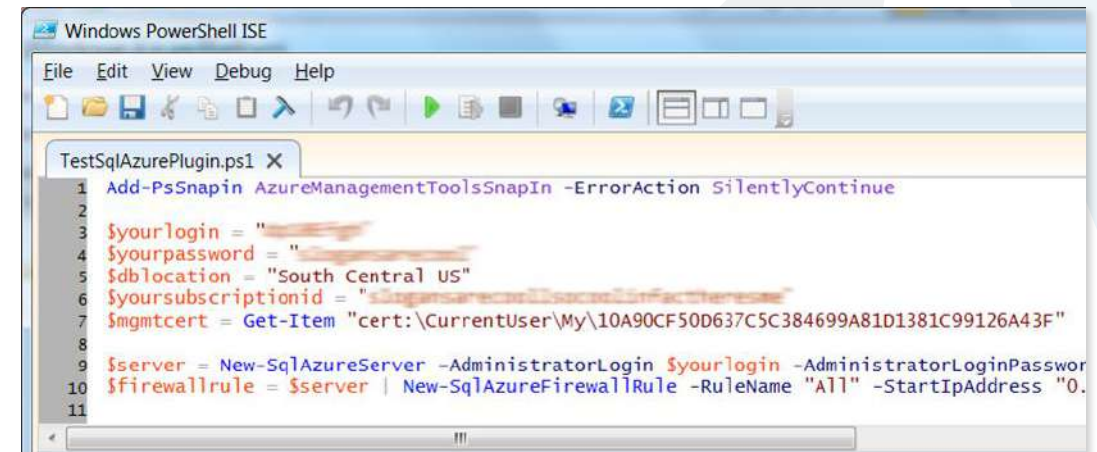
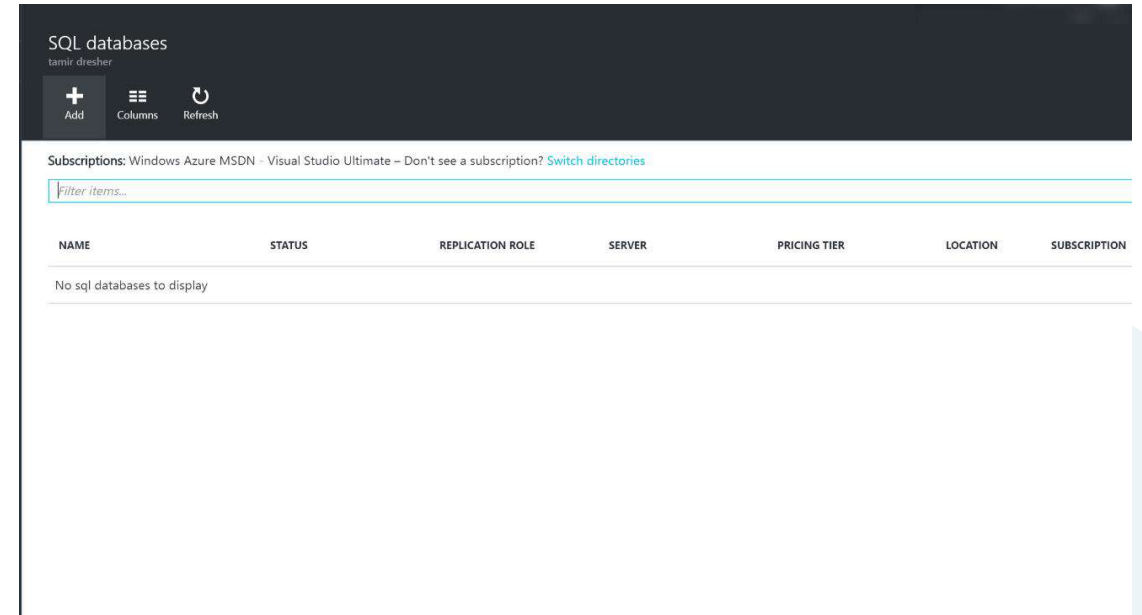
Specify admin login credentials

Add firewall rules and enable service access

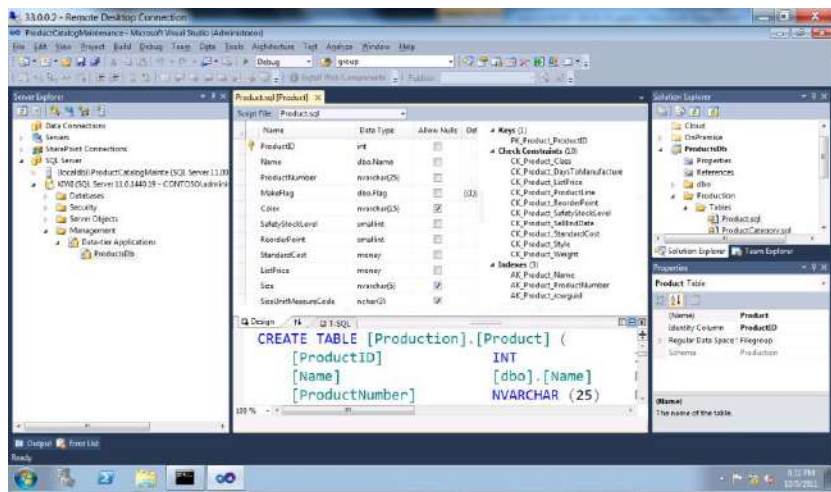
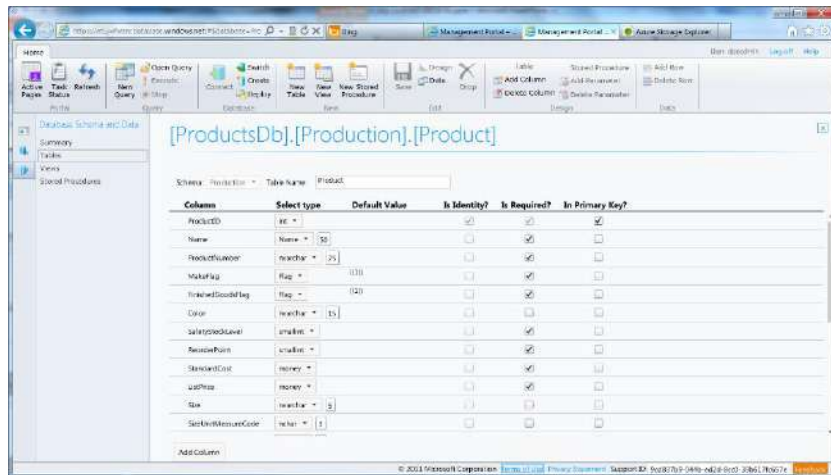
Automate Server Provisioning

Use Windows Azure Platform PowerShell cmdlets
(or use REST API directly)

wappowershell.codeplex.com



Enhanced Tooling



SQL Database Management Portal

Web designers for tables, views, stored procs

Interactive query editing and execution

SQL Server Data Tools (SSDT)

Visual Studio IDE for database development

Includes modern designers and projects with declarative, model-driven development

Develop and test in both connected and disconnected states

Platform targeting for both SQL Server (2005 and above) and SQL Database

Get it free with Web PI, with SQL Server 2012 and with Visual Studio 11



SQL Database editions

Service Tier	Performance Level	Common App Pattern	Performance			Business Continuity	
			Max DB Size	Transaction Perf. Objective	DTU	PITR	DR / GEO-Rep
Basic	Basic	Small DB, SQL opp	2 GB	Reliability / Hr.	5	7 Days	DB Copy + Manual Export
Standard	S0 S1 S2	Wrkgp/cloud app, multiple concurrent operations	250 GB	Reliability / Min.	10 20 50	14 Days	DB Copy + Manual Export
Premium	P1 P2 P3	Mission Critical, High volume, Many concurrent Users	500 GB	Reliability / sec.	100 200 800	35 Days	Active Geo-replication

<http://dtucalculator.azurewebsites.net/>





Backup and Restore

- Auto backups, transactional logs every 5 min
- Backups in Azure Storage and geo-replicated
- Creates a side-by-side copy, non-disruptive
- Backups retention policy: 7, 14 or 35 days
- Automated export of logical backups for long-term backup protection
- Geo-restore – Available in all tiers: Basic, Standard and Premium

Azure Storage Account Services

Blobs

Highly scalable,
REST based cloud
object store

Block Blobs: Sequential
file I/O

Page Blobs: Random-
write pattern data

Tables

Massive auto-scaling
NoSQL store

Dynamic scaling based
on load

Scale to PBs of table
data

Fast key/value lookups

Queues

Reliable queues at
scale for cloud
services

Decouple and scale
components

Message visibility timeout
and update message to
protect against unreliable
dequeuers

Disks

Persistent disks for
Azure IaaS VMs

Built on page blobs

Premium Storage Disks:
SSD based, high IOPS,
low latency

Files

Fully Managed File
Shares in the Cloud

Map to file share,
standard file system
semantics

"Lift and shift" legacy
apps

Code against (REST API)

Use on Windows & Linux VMs

Real Worlds Examples:

XBOX – Cloud Game Save, Halo 4, Music, Kinect data collection

OneDrive

Bing – stores raw data from Twitter and Facebook to digest later

Skype – Video Messaging



Azure Storage Account types

➤ General-purpose Storage Accounts

- Tables, Queues, Files, Blobs and Azure virtual machine
- performance tiers:
 - **Standard storage performance tier** allows you to store Tables, Queues, Files, Blobs and Azure virtual machine disks.
 - **Premium storage performance tier** provides [High-Performance Storage for Azure Virtual Machine Workloads](#)

➤ Blob Storage Accounts

- specialized storage for unstructured data as blobs (objects)
 - Only block and append blobs
- Access tiers:
 - **Hot access tier** indicates that the objects in the storage account will be more frequently accessed.
 - **Cool access tier** indicates that the objects in the storage account will be less frequently accessed.



+ New

Resource groups

All resources

Recent

App Services

Virtual machines (classic)

Virtual machines

SQL databases

Cloud services (classic)

Subscriptions

Application Insights

Browse >

New

Search the marketplace

MARKETPLACE

[See all](#)

Virtual Machines >

Web + Mobile >

Data + Storage >

Data + Analytics >

Internet of Things >

Networking >

Media + CDN >

Hybrid Integration >

Security + Identity >

Developer Services >

Management >

Intelligence >

Container Apps >

RECENT

Data + Storage



business-class apps.



Data Lake Store

Hyper-scale repository for big data analytic workloads



SQL Data Warehouse

Fully elastic, managed, and parallelized relational database. Analyze and scale in seconds.



Azure DocumentDB

Scalable and managed NoSQL document database service for modern cloud applications.



Storage account

Use Blobs, Tables, Queues, and Files for reliable, economical cloud storage.



Redis Cache

Distributed, in-memory Redis Cache service for modern cloud applications



Azure Search

Search-as-a-service solution

Create storage account

The cost of your storage account depends on the usage and the options you choose below.

[Learn more](#)

* Name ⓘ

.core.windows.net

Deployment model ⓘ

Resource manager

Classic

Performance ⓘ

Standard

Premium

Replication ⓘ

Locally-redundant storage (LRS) ▼

Subscription

Windows Azure MSDN - Visual Studio Ultim: ▼

* Resource group

+ New ▼

New resource group name

☐ Pin to dashboard

Create



Storage account

Settings Delete

Essentials ^

Resource group
Default-Web-WestUS

Status
Primary: Available

Location
East US

Subscription name
Windows Azure MSDN - Visual Studio Ultim..

Subscription ID
785eaf75-ac1e-47f8-a80a-808ee4478db9

Performance
Standard

Replication
Locally-redundant storage (LRS)

Services

Blobs

Files

Tables

Queues

Monitoring

Settings tamirtest1

SUPPORT + TROUBLESHOOTING

Audit logs

New support request

GENERAL

Properties

Access keys

Configuration

Encryption

MONITORING

Alert rules

Access keys tamirtest1

Regenera... Key1

Regenera... Key2

STORAGE ACCOUNT NAME

Access keys

KEY1

KEY2

Connection strings

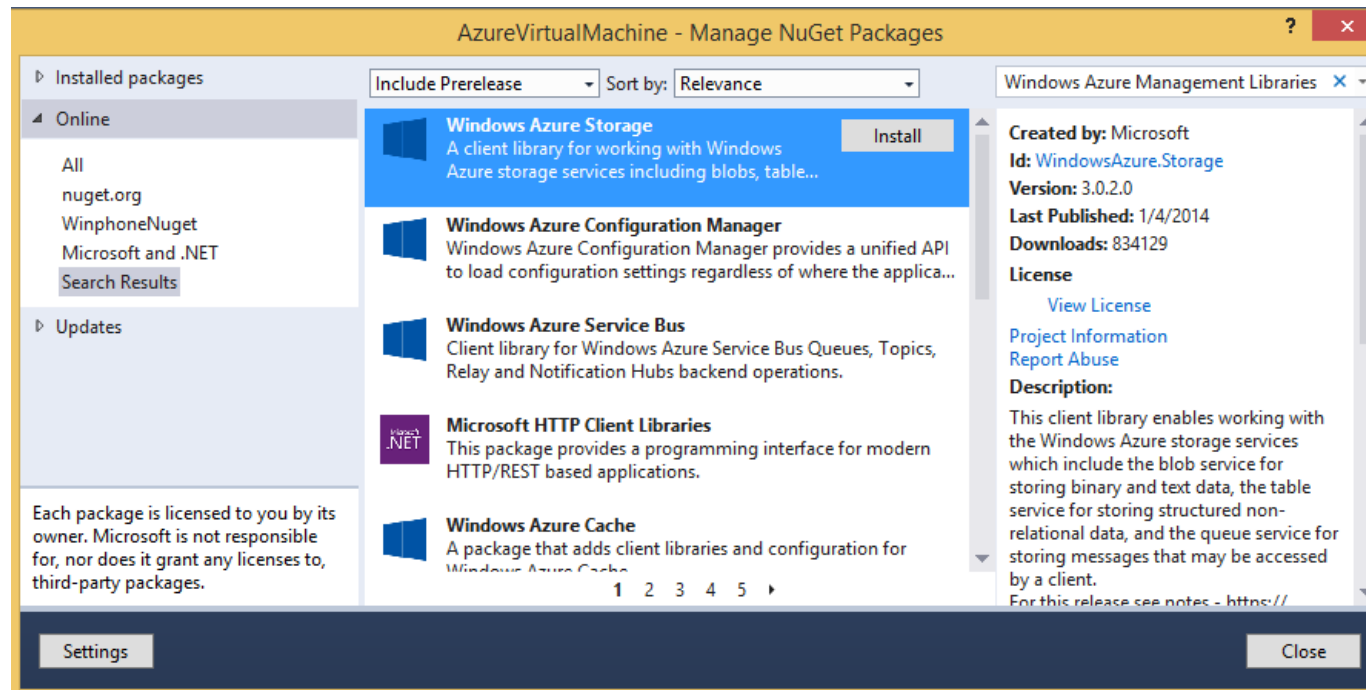
KEY1

KEY2



Storage API

- REST
- Client API from SDK: WindowsAzure.Storage namespace
 - A wrapper around the REST API
 - Hides many of the complexities of the service + Auto retries

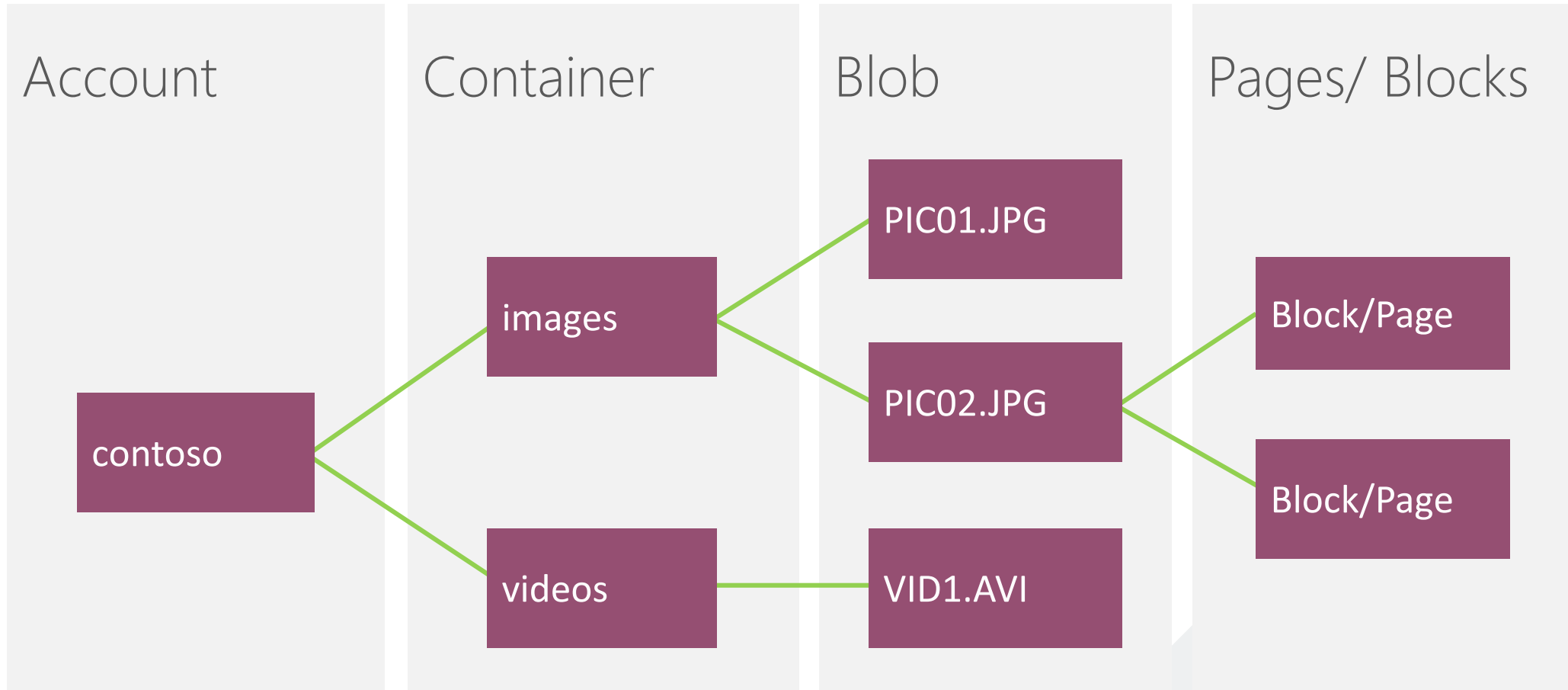


Cloud Storage - Azure BLOB Storage

- BLOB – Binary Large Object
- Storage for any type of entity such as binary files and text documents
- Distributed File Service (DFS)
 - Scalability and High availability
- BLOB file is distributed between multiple server and replicated at least 3 times
- [Get Started with Storage Account](#)
- [Get Started with Blob Storage](#)

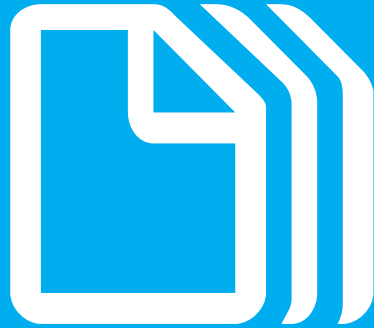
Azure Blob Storage Concepts

`http://<account>.blob.core.windows.net/<container>/<blobname>`

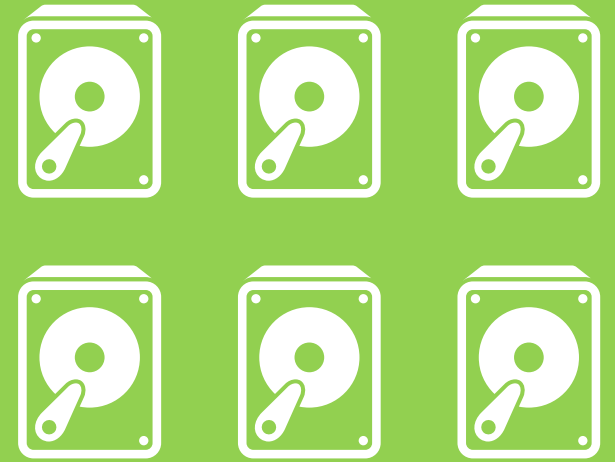


Blob Operations

PutBlob
GetBlob
DeleteBlob
CopyBlob
SnapshotBlob
LeaseBlob



REST



Windows Azure Storage



Page Blob in code

► Creating

```
using Microsoft.WindowsAzure.StorageClient ;

CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
    $"DefaultEndpointsProtocol=https;AccountName={accountName};AccountKey={accountKey}");
CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();
CloudBlobContainer container = blobClient.GetContainerReference(containerName);
container.CreateIfNotExist();
CloudPageBlob pageBlob = container.GetPageBlobReference(blobName);
pageBlob.Create(blobSize);
```

► Writing

```
pageBlob.WritePages(dataStream, startingOffset) ;
```




Page Blob in code

► Reading

```
BlobStream blobStream = pageBlob.OpenRead();  
byte[] buffer = new byte[rangeSize] ;  
blobStream.Seek(blobOffset, SeekOrigin.Begin) ;  
int numBytesRead = blobStream.Read(buffer, bufferOffset, rangeSize);
```

► Clear Pages

```
pageBlob.ClearPages(startOffset, length)
```




Page Blob in code

► Reading

```
BlobStream blobStream = pageBlob.OpenRead();  
byte[] buffer = new byte[rangeSize] ;  
blobStream.Seek(blobOffset, SeekOrigin.Begin) ;  
int numBytesRead = blobStream.Read(buffer, bufferOffset, rangeSize);
```

► Clear Pages

```
pageBlob.ClearPages(startOffset, length)
```


Concurrency

- Optimistic concurrency – Timestamps/ETags
 - Timestamp based – If-Modified-Since and If-UnModified-Since
 - ETag based – If-Match and If-None-Match (can force update with *)
 - Conditional update with supplied Timestamp or ETag will fail if conditions not met
- Pessimistic Concurrency - Leases
 - Lease Blob for exclusive write and delete access
 - 15-60s lease duration (can be renewed) or infinite lease (locks)
 - Can change lease id to acquire ownership in a chain/workflow
 - Can also acquire on containers to prevent container deletion
- Last Writer wins
 - <https://azure.microsoft.com/en-us/blog/managing-concurrency-in-microsoft-azure-storage-2/>
 - <https://msdn.microsoft.com/en-us/library/dd179371.aspx>

Optimistic Concurrency

```
string originalETag = blockBlob.Properties.ETag;
try
{
    blockBlob.UploadText(helloText,
        accessCondition: AccessCondition.GenerateIfMatchCondition(originalETag));
}
catch (StorageException ex)
{
    if (ex.RequestInformation.HttpStatusCode == (int)HttpStatusCode.PreconditionFailed)
    {
        Console.WriteLine("Blob's original etag no longer matches");
    }
    else
        throw;
}
```


Passimistic Concurrency

```
string lease = blockBlob.AcquireLease(TimeSpan.FromSeconds(15), null);

// Update blob using lease. This operation will succeed
var accessCondition = AccessCondition.GenerateLeaseCondition(lease);
blockBlob.UploadText("update", accessCondition: accessCondition);

try
{
    // Below operation will fail as no valid lease provided
    blockBlob.UploadText("Update without lease, will fail");
}
catch (StorageException ex)
{
    if (ex.RequestInformation.HttpStatusCode == (int)HttpStatusCode.PreconditionFailed)
        Console.WriteLine("Blob's lease does not match");
    else
        throw;
}
```




Transient Faults

- “ **transient fault** is a fault that is no longer present if power is disconnected for a short time and then restored.”
(http://en.wikipedia.org/wiki/Transient_fault#Transient_fault)
- Many faults in connectivity to cloud are transient by nature
- Commonly occur when connecting to service or database

Transient Faults handling

➤ Retry Logic

- Linear – every fixed amount of time
- Exponential – if the server is heavy-used (throttling) we don't want to flood it

immediate....1 sec....5 seconds....etc.

➤ Idempotency

- operations in [mathematics](#) and [computer science](#), that can be applied multiple times without changing the result beyond the initial application (wikipedia)
- Same messages could be sent more than once or out of sequence
- Design for idempotency

Retry Policy Application

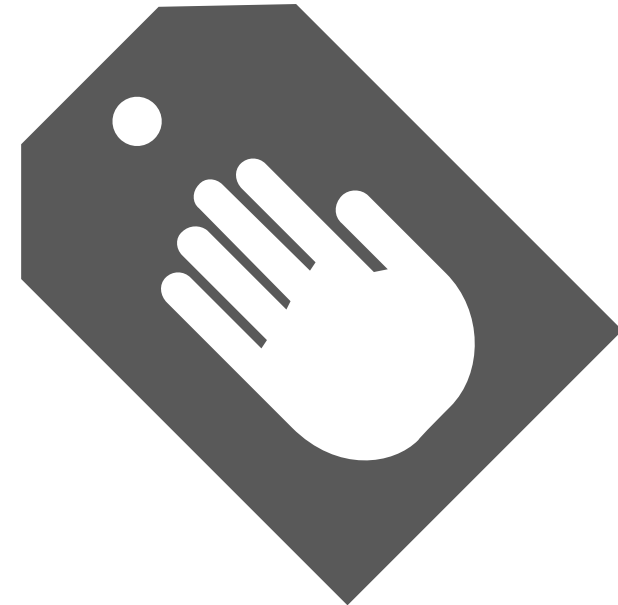
- Microsoft.WindowsAzure.Storage.RetryPolicies.IRetryPolicy Interface
 - [ExponentialRetry](#)
 - [LinearRetry](#)
 - [NoRetry](#)
- Default is exponential – if you don't want any retry logic then you must override

Cloud Storage - Table Storage

- Not RDBMS
 - No relationships between entities
 - NoSql
- Entity can have up to 255 properties - Up to 1MB per entity
- Mandatory Properties for every entity
 - PartitionKey & RowKey (only indexed properties)
 - Uniquely identifies an entity
 - Same RowKey can be used in different PartitionKey
 - Defines the sort order
 - Timestamp - Optimistic Concurrency
- Strongly consistent
- [Get Started with Table Storage](#)

Shared Access Signatures

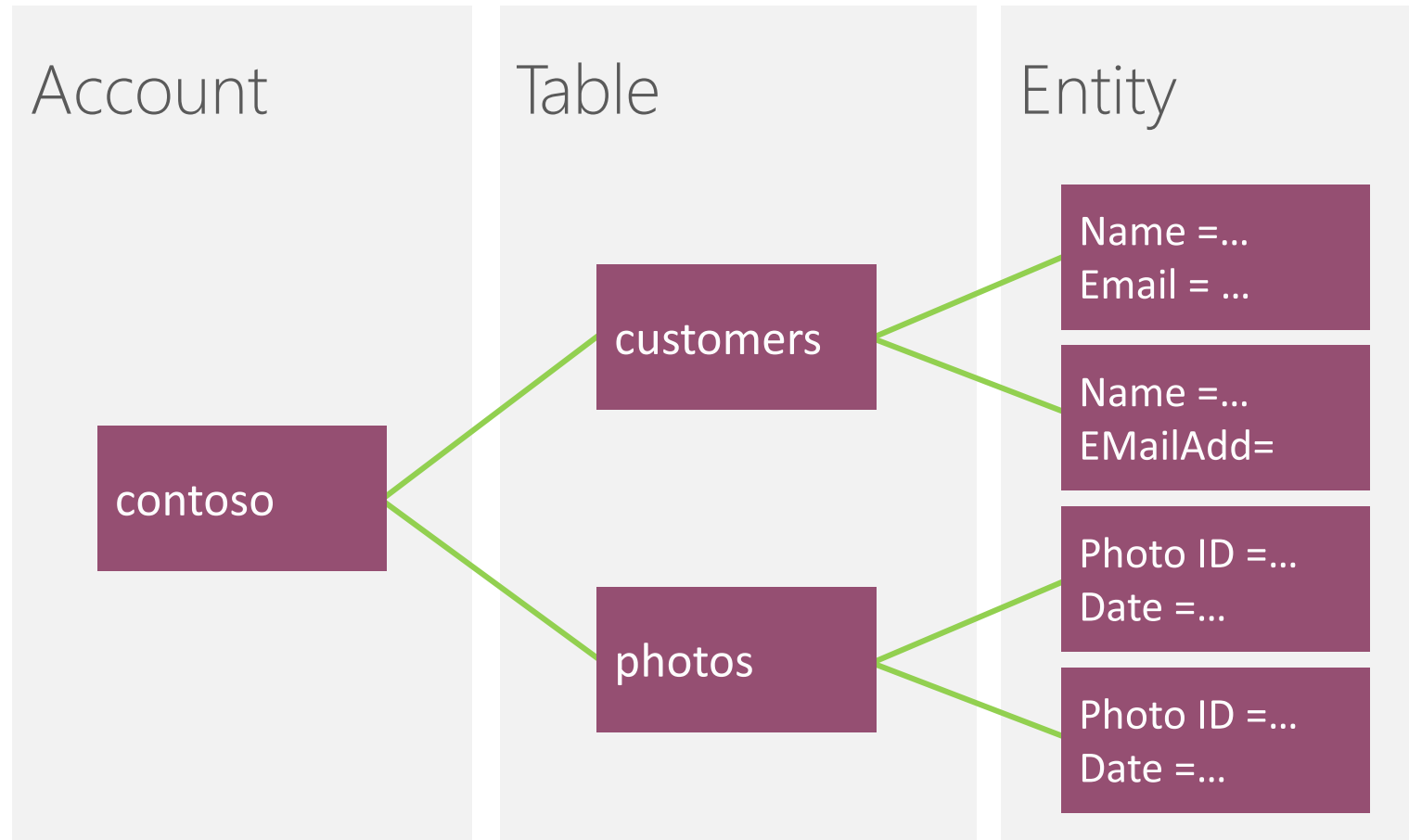
- Fine grain access rights to blobs and containers
- Sign URL with storage key – permit elevated rights
- Revocation
 - Use short time periods and re-issue
 - Use container level policy that can be deleted
- Two broad approaches
 - Ad-hoc
 - Policy based



Tables

Azure Storage

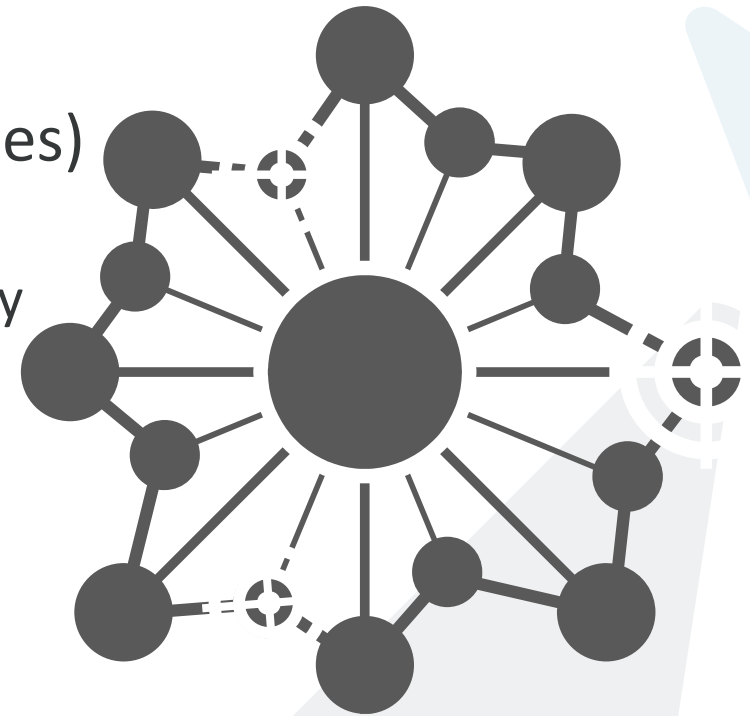
Table Storage Concepts





Entity Properties

- Entity can have up to 255 properties
 - Up to 1MB per entity
- Mandatory Properties for every entity
 - PartitionKey & RowKey (only indexed properties)
 - Uniquely identifies an entity
 - Same RowKey can be used in different PartitionKey
 - Defines the sort order
 - Timestamp
 - Optimistic Concurrency
 - Exposed as an HTTP Etag





Sample – Inserting an Entity into a Table

```
// You will need the following using statements
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Table;

// Create the table client.
CloudTableClient tableClient = storageAccount.CreateCloudTableClient();
CloudTable peopleTable = tableClient.GetTableReference("people");
peopleTable.CreateIfNotExists();

// Create a new customer entity.
CustomerEntity customer1 = new CustomerEntity("Harp", "Walter");
customer1.Email = "Walter@contoso.com";
customer1.PhoneNumber = "425-555-0101";

// Create an operation to add the new customer to the people table.
TableOperation insertCustomer1 = TableOperation.Insert(customer1);

// Submit the operation to the table service.
peopleTable.Execute(insertCustomer1);
```




Table Object Model

- *ITableEntity* interface –PartitionKey, RowKey, Timestamp, and Etag properties
 - Implemented by *TableEntity* and *DynamicTableEntity*

```
// This class defines one additional property of integer type,  
// since it derives from TableEntity it will be automatically  
// serialized and deserialized.  
public class SampleEntity : TableEntity  
{  
    public int SampleProperty { get; set; }  
}
```


Querying

- Retrieve(PartitionKey, RowKey) – retrieve single entity that satisfy the arguments
- TableQuery - lightweight object that represents a query for a given set of entities
- IQueryable (not efficient)

```
IQueryable<Footwear> query = table.CreateQuery<Footwear>()  
    .Where(f => f.Gender == "Male" && (f.Size > 4 && f.Size < 7));  
  
IEnumerable<Footwear> shoes = query.ToList();
```

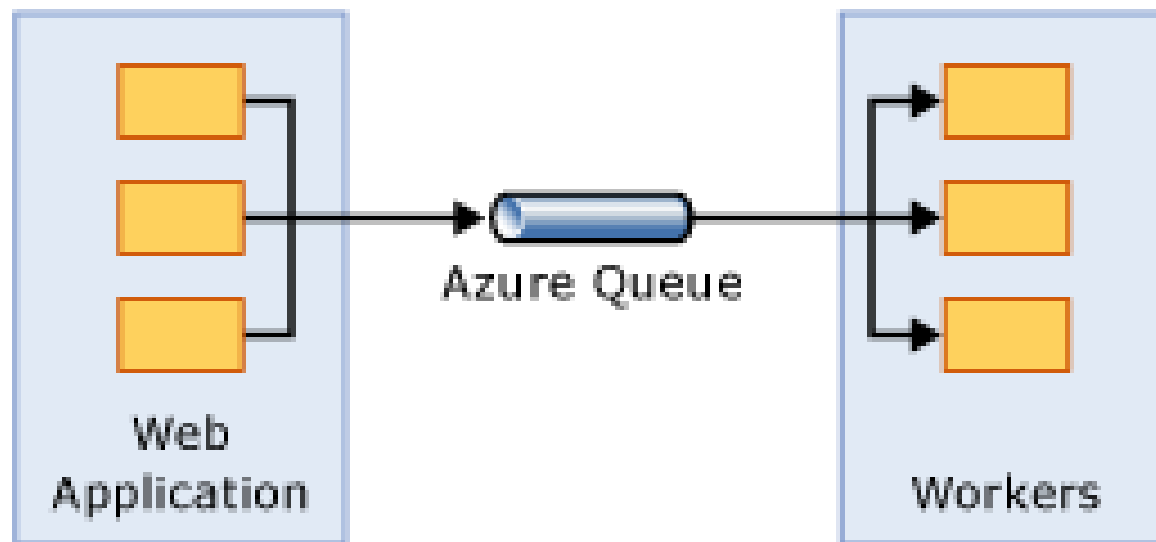
full table scan will be performed.
because no Partition key was specified, the query will be sent
to every Partition Server.

queues

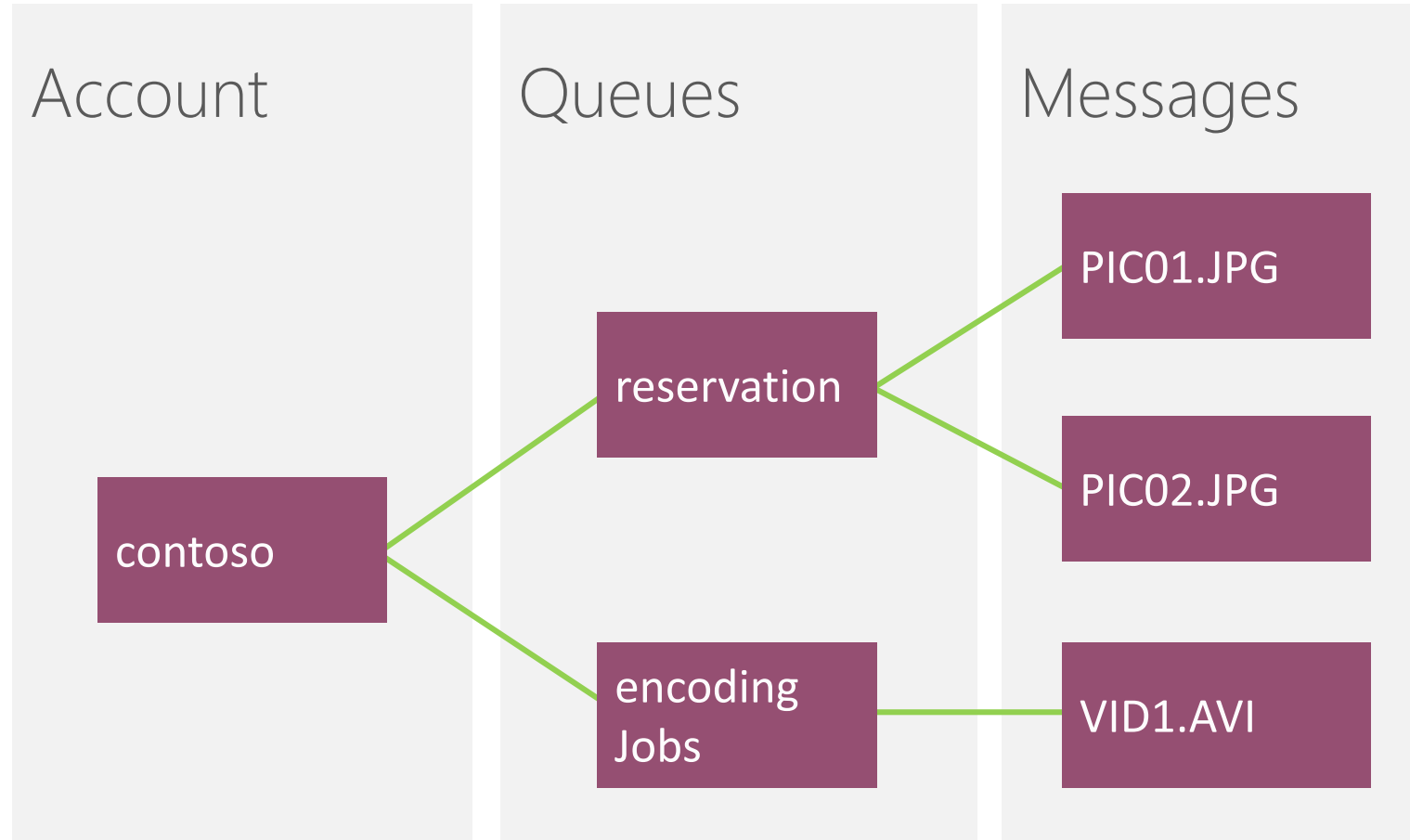
Azure Storage

Windows Azure Queues

- Queue – First In First Out (FIFO) – Not guaranteed
- Queue are performance efficient, highly available and provide reliable message delivery
 - Simple, asynchronous work dispatch
 - Programming semantics ensure that a message can be processed at least once
- Decouple Producers and Consumers



Storage Queue Concepts





Queue Operations

➤ Queue

- Create Queue
- Delete Queue
- List Queues
- Get/Set Queue Metadata

➤ Messages

- Add Message (i.e. Enqueue Message)
- Get Message(s) (i.e. Dequeue Message)
- Peek Message(s)
- Delete Message

Queue API

```
// Retrieve storage account from connection string
CloudStorageAccount storageAccount = CloudStorageAccount.Parse(
    CloudConfigurationManager.GetSetting("StorageConnectionString"));

// Create the queue client
CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient();

// Retrieve a reference to a queue
CloudQueue queue = queueClient.GetQueueReference("myqueue");

// Create the queue if it doesn't already exist
queue.CreateIfNotExists();

// Create a message and add it to the queue.
CloudQueueMessage message = new CloudQueueMessage("Hello, World");
queue.AddMessage(message);
```


Queue API

```
// Peek at the next message
```

```
CloudQueueMessage peekedMessage = queue.PeekMessage();
```

```
// Get the next message
```

```
CloudQueueMessage retrievedMessage = queue.GetMessage();
```

```
//Process the message in less than 30 seconds, and then delete the message
```

```
queue.DeleteMessage(retrievedMessage);
```

```
// Get the message from the queue and update the message contents.
```

```
CloudQueueMessage message = queue.GetMessage();
```

```
message.SetMessageContent("Updated contents.") ;
```

```
queue.UpdateMessage(message,
```


```
    TimeSpan.FromSeconds(0.0), // Make it visible immediately.
```

```
    MessageUpdateFields.Content | MessageUpdateFields.Visibility);
```





Message Visibility

- By default, after dequeuing, messages are invisible for 30 seconds
- While invisible, no other consumer can dequeue the message
- You can set the visibility-timeout when getting the message from the queue

```
 GetMessageAsync Task<CloudQueueMessage>  
(TimeSpan? visibilityTimeout, QueueRequestOptions options, OperationContext operationContext):  
Task<CloudQueueMessage>
```

- You can extend the visibility-timeout by executing the UpdateMessageAsync method

```
 UpdateMessageAsync Task  
(CloudQueueMessage message, TimeSpan visibilityTimeout, MessageUpdateFields updateFields):Task
```

- Call the DeleteMessageAsync method to remove the message from the queue
- Use the DequeueCount property to validate the amount of times the message was dequeued



Poison Messages


















- Message can cause the consumer to crash
- find “poison” messages when dequeuing by examining the [DequeueCount](#) property of the message.
- If [DequeueCount](#) is above a given threshold it is a potential “poison” message
- Two options
 1. Delete the message
 2. Store in Poison Queue/Table

Cosmos DB



Cosmos DB

Microsoft's globally distributed, multi-model database service for mission-critical applications.

					
DocumentDB API					
MongoDB API					
Graph API					
Table API					



CosmosDB Packages

➤ Nuget:

- WindowsAzure.Storage-PremiumTable
- Microsoft.Azure.DocumentDB
- Microsoft.Azure.Graphs

➤ Emulator

- <https://docs.microsoft.com/en-us/azure/cosmos-db/local-emulator>

Tunable Consistency – 5 levels



Consistency Level	Guarantees
Strong	Linearizability
Bounded Staleness	Consistent Prefix. Reads lag behind writes by k prefixes or t interval
Session	Consistent Prefix. Monotonic reads, monotonic writes, read-your-writes, write-follows-reads
Consistent Prefix	Updates returned are some prefix of all the updates, with no gaps
Eventual	Out of order reads

Tunable Consistency – 5 levels

- **Strong** - All writes are visible to all readers. Writes synchronously committed by a majority quorum of replicas and reads are acknowledged by the majority read quorum.
- **Bounded Stateless** - Guaranteed ordering of writes, reads adhere to minimum freshness. Writes are propagated asynchronously, reads are acknowledged by majority quorum lagging by at most K prefixes.
- **Session** - Read your own writes. Writes are propagated asynchronously while reads for a session are issued against the replica that can serve the requested version.
- **Consistent Prefix** - Updates returned are some prefix of all the updates, with no gaps
- **Eventual** - Reads eventually converge with writes. Writes are propagated asynchronously while reads can be acknowledged by any replica. Readers may view older data than previously observed.

Tunable Consistency

	<i>Writes</i>	<i>Reads</i>
Strong	sync quorum writes	quorum reads
Bounded	async replication	quorum reads
Session*	async replication	session bound replica
Eventual	async replication	any replica

* Ideal consistency and performance tradeoff for many application scenarios. High performance writes and reads with predictable consistency.

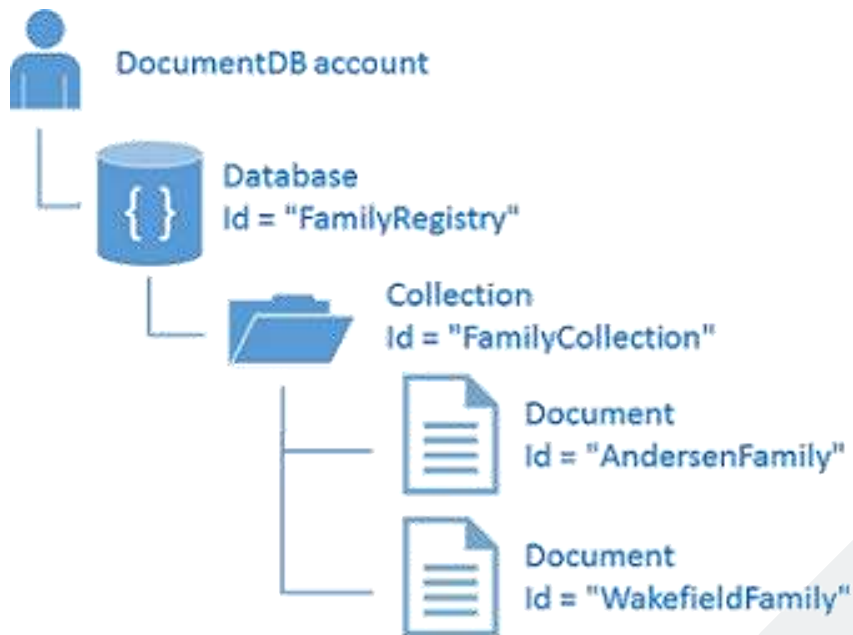
The background of the slide features two hot air balloons floating in a bright blue sky filled with soft, white clouds. The balloon on the left is white with blue and yellow checkered patterns. The balloon on the right is primarily orange with yellow and red accents. Both balloons have small baskets hanging from them.

Document database

CosomosDB

CosmosDB - Document

- Fully managed, scalable JSON document database service
- Supports SQL as querying language
- Joins with nested documents is supported
- Everything is indexed



Creating DocumentDB Database

```
using (client = new DocumentClient(new Uri(endpointUrl), authorizationKey))
{
    Database database = client.CreateDatabaseQuery().Where(c => c.Id == id)
        .ToArray()
        .FirstOrDefault();

    if (database == null)
    {
        database = await client.CreateDatabaseAsync(new Database { Id = id });
    }
}
```




Creating DocumentDB Collection

```
DocumentCollection collection =
client.CreateDocumentCollectionQuery(UriFactory.CreateDatabaseUri(databaseId))
    .Where(c => c.Id == collectionId)
    .ToArray()
    .SingleOrDefault();

if (collection == null)
{
    DocumentCollection collectionDefinition = new DocumentCollection();
    collectionDefinition.Id = collectionId;
    collectionDefinition.IndexingPolicy = new IndexingPolicy(
        new RangeIndex(DataType.String) { Precision = -1 });
    collectionDefinition.PartitionKey.Paths.Add("/LastName");

    collection = await client.CreateDocumentCollectionAsync(
        UriFactory.CreateDatabaseUri(databaseId),
            collectionDefinition,
            new RequestOptions { OfferThroughput = 400 }));
}
```




Creating DocumentDB Document

```
Family AndersonFamily = new Family() {...}  
string collectionLink = collection.SelfLink;  
await client.CreateDocumentAsync(collectionLink, AndersonFamily);
```

```
Document doc1 = await client.CreateDocumentAsync(col1.DocumentsLink,  
    new { id = "doc1", partitionKey = "partitionKey1" });
```


Demo: Interacting with DocumentDB

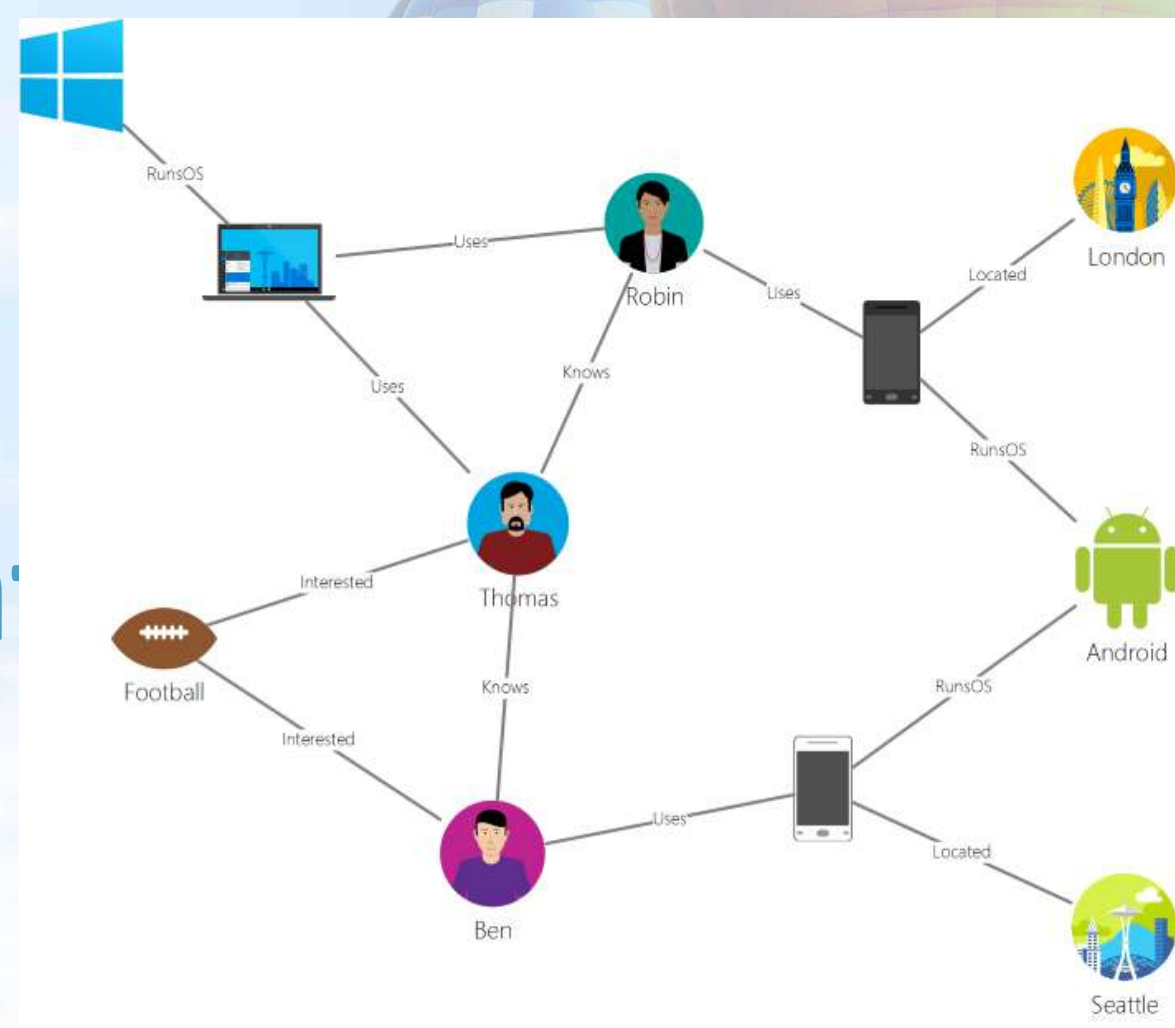
```
var query = client.CreateDocumentQuery<Family>(
    collectionLink,
    new FeedOptions { MaxItemCount = 1, EnableCrossPartitionQuery = true })
    .Where(d => d.LastName == "Andersen")
    .Select(f => new { Name = f.LastName })
    .AsDocumentQuery();

foreach (var item in query.ToList())
{
    Console.WriteLine("The {0} family live in {1}", item.Name, item.City);
}
```

```
var query = client.CreateDocumentQuery<Family>(collectionLink, new SqlQuerySpec()
{
    QueryText = "SELECT * FROM Families f WHERE (f.id = @id)",
    Parameters = new SqlParameterCollection() { new SqlParameter("@id", "AndersenFamily")
    }
}, DefaultOptions);
```

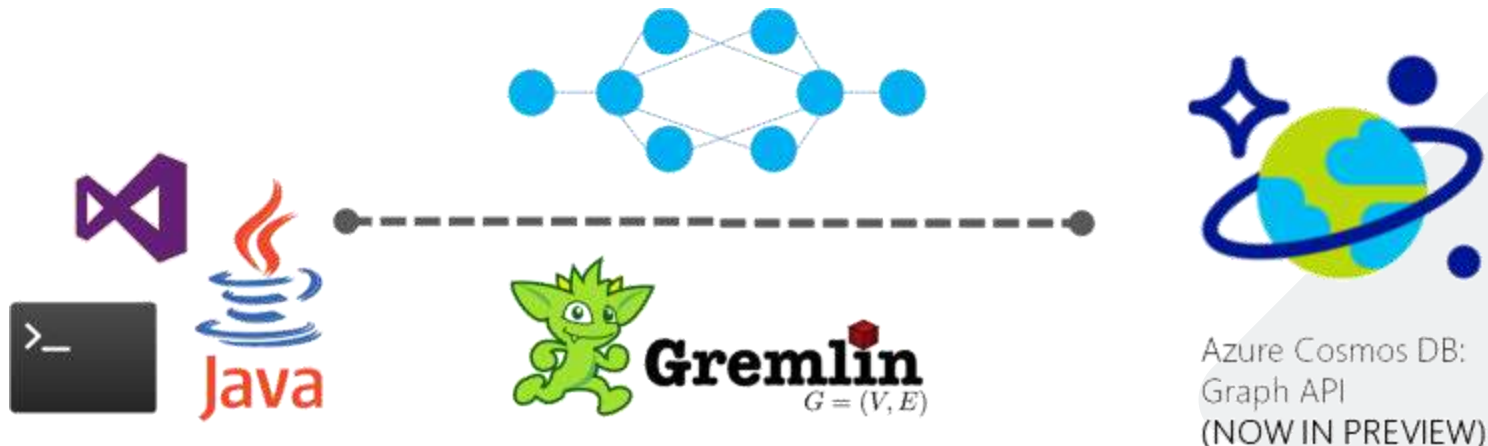

Graph data

CosmosDB



Gremlin

- Apache Tinkerpop's graph traversal language, Gremlin is a Graph API for creating graph entities, and performing graph query operations.
 - <http://gremlindocs.spmallette.documentup.com/>
 - <https://docs.microsoft.com/en-us/azure/cosmos-db/graph-introduction>




```
g.addV('person')  
  .property('id', 'thomas')  
  .property('firstName', 'Thomas')  
  .property('age', 44)
```

```
g.addV('person')  
  .property('id', 'marry')  
  .property('firstName', 'Marry')  
  .property('age', 35)
```

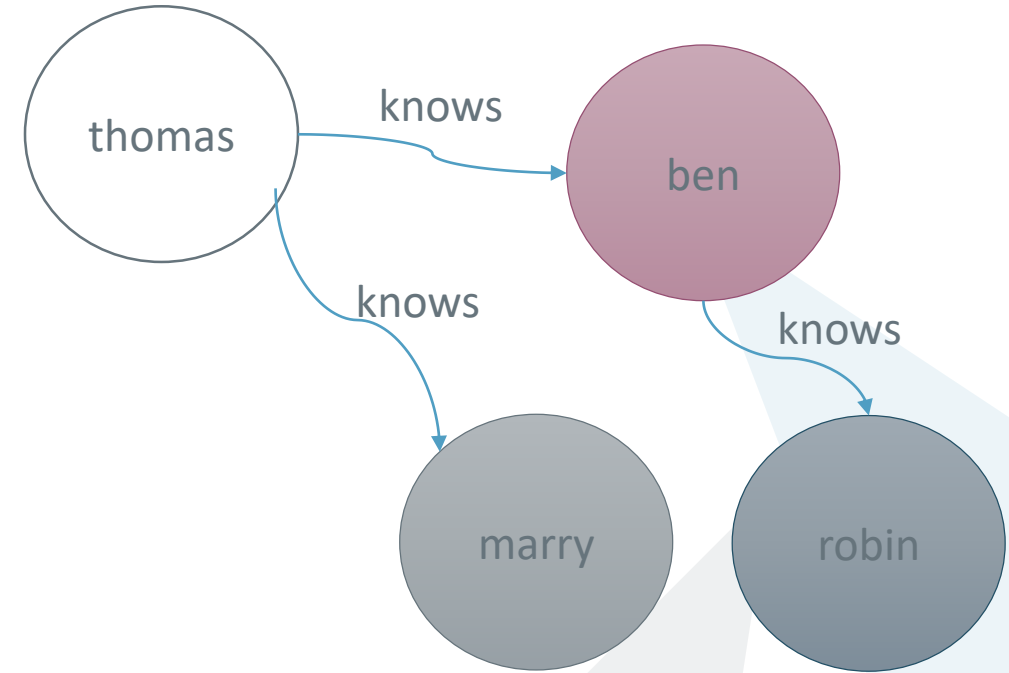
```
g.addV('person')  
  .property('id', 'ben')  
  .property('firstName', 'Ben')  
  .property('age', 36)
```

```
g.addV('person')  
  .property('id', 'robin')  
  .property('firstName', 'Robin')  
  .property('age', 35)
```

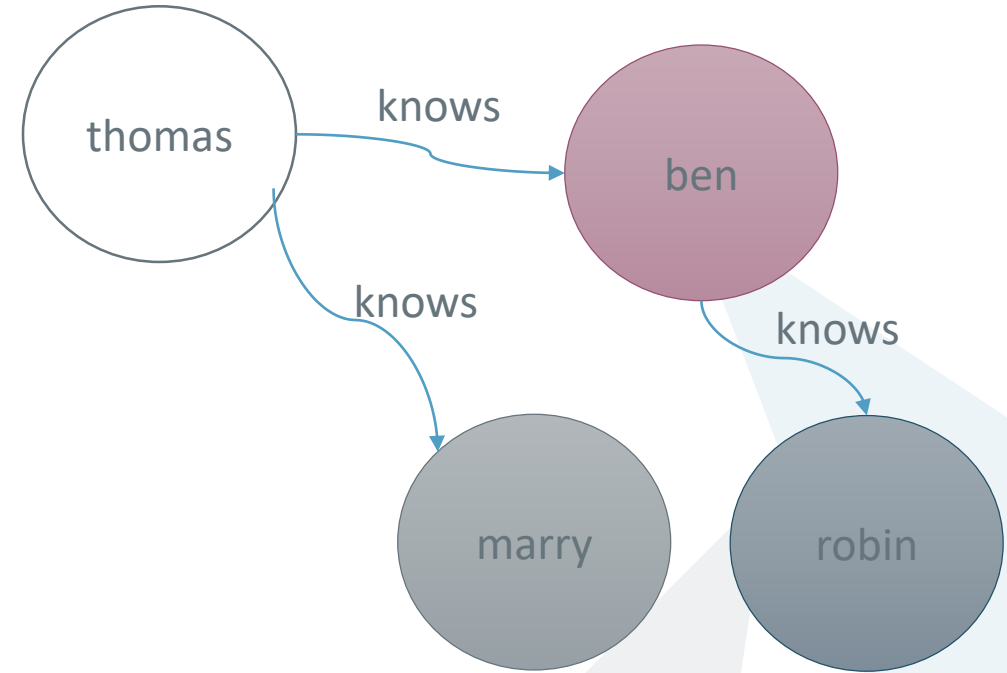
```
g.V('thomas')  
  .addE('knows')  
  .to(g.V('mary'))
```

```
g.V('thomas')  
  .addE('knows')  
  .to(g.V('ben'))
```

```
g.V('ben')  
  .addE('knows')  
  .to(g.V('robin'))
```




```
g.V('thomas')  
  .outE('knows')  
  .inV().hasLabel('person')  
  .outE('knows')  
  .inV().hasLabel('person')
```





Summary

- **Choosing the correct persistence store is crucial for good architecture**
- **Realtional**
- **Storage Account**
 - BLOBs
 - Tables
 - Queues
 - Files
 - Disks
- **Shared Access Signatures (SAS) and Policies**
- **Partitioning and Consistency**
- **CosmosDB**
 - Document
 - Graph
 - Premium Tables

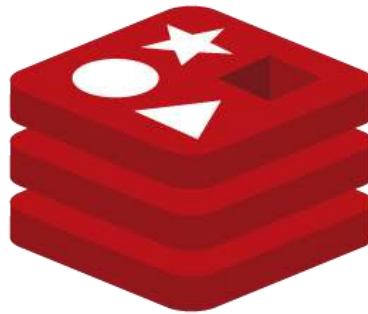


Other Data Services

- **Azure Search** - Full text search and text analysis, sophisticated data indexing



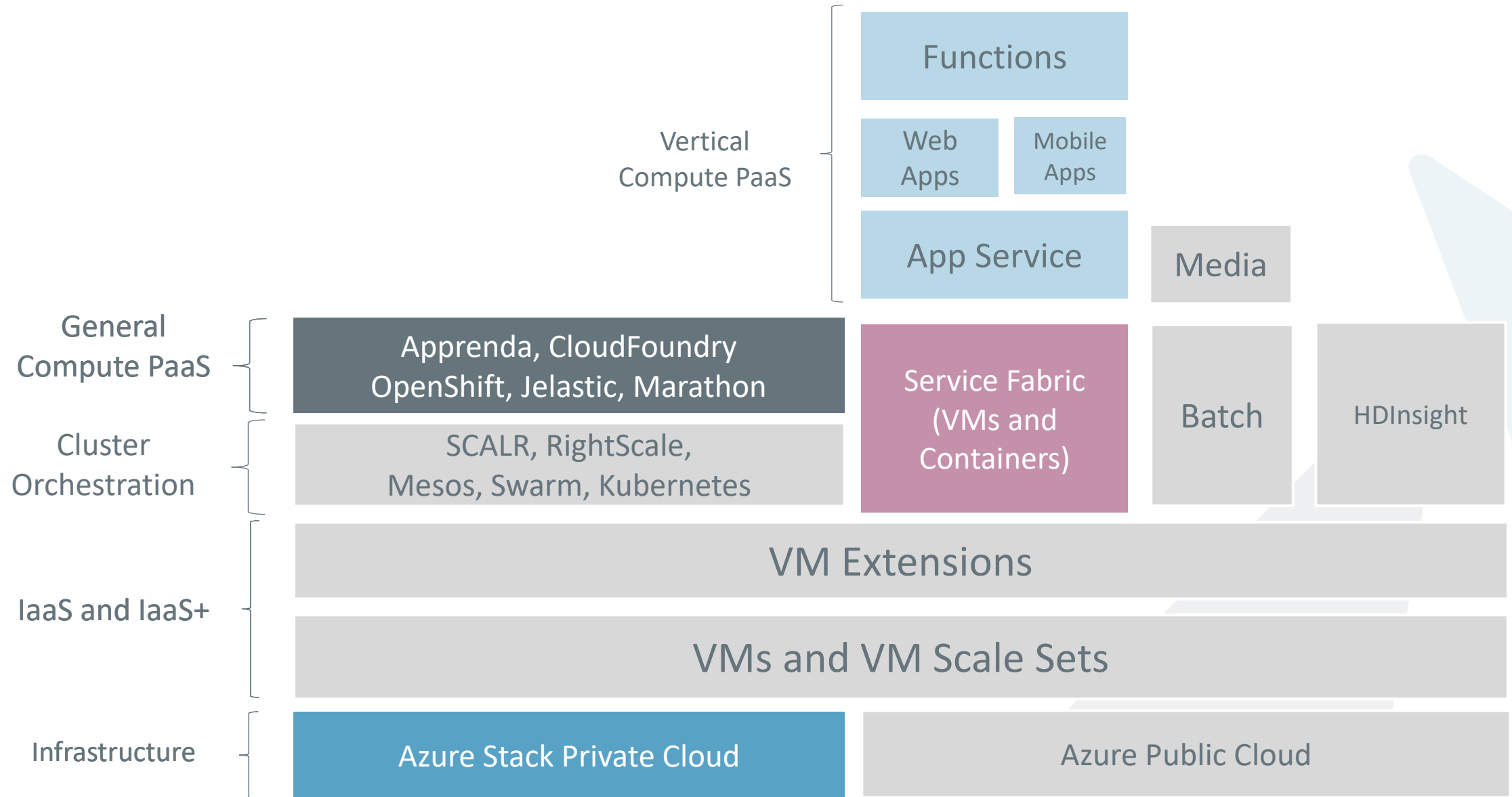
- **Azure Redis Cache** - gives you access to a secure, dedicated Redis cache, managed by Microsoft and accessible from any application within Azure.



PaaS - Azure App Service



Service hosting & management layers





Agenda

- Web Apps
 - WebJobs
- App Service Plans
- Deployment Slots
- Mobile Apps
- API Apps
- Logic Apps
- Azure Functions



Azure App Services

- Building and hosting web applications without managing infrastructure
- Offers auto-scaling and high availability
- Supports both Windows and Linux
- Enables automated deployments from GitHub, VSTS, or any Git repo
- Service App suite includes
 - **Web Apps** - websites and web applications
 - **API Apps** - RESTful APIs
 - **Mobile Apps** - mobile app back ends
 - **Logic Apps** - automating business processes and integrating systems and data across clouds without writing code
- **Azure Functions** are also based on the App Services infrastructure



Azure web APPS

Azure App Services



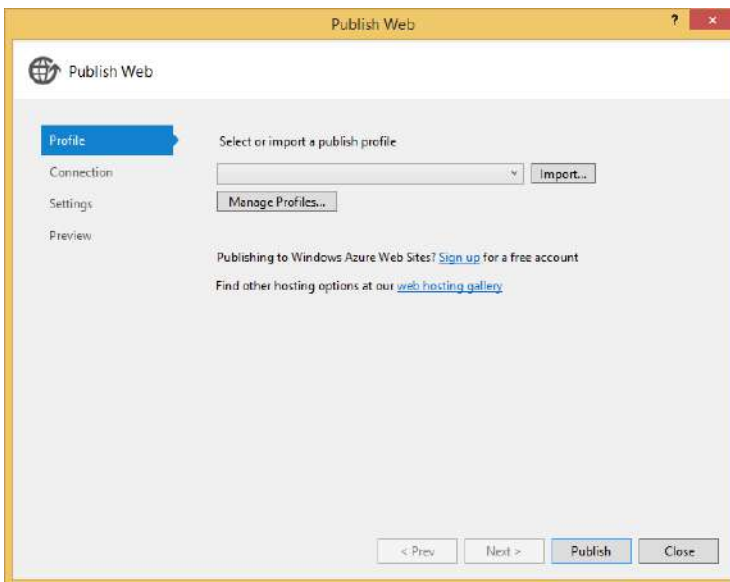
Azure Web Sites

- Provision a Web Application Fast
- You can use IDE, PowerShell, Portal
- Deploy Easily via a Source Control

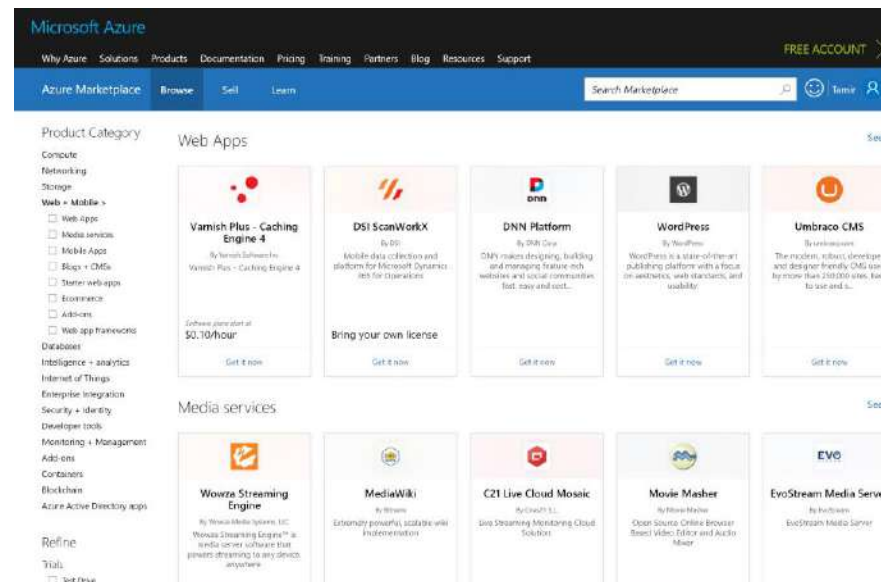


Creating a Web Site

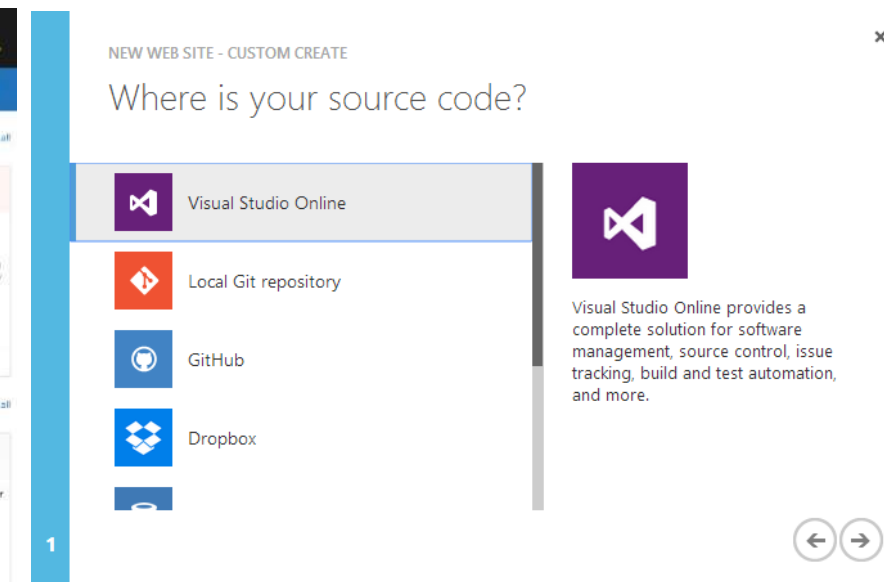
Publish from VS



From Gallery



Sync with Source Control

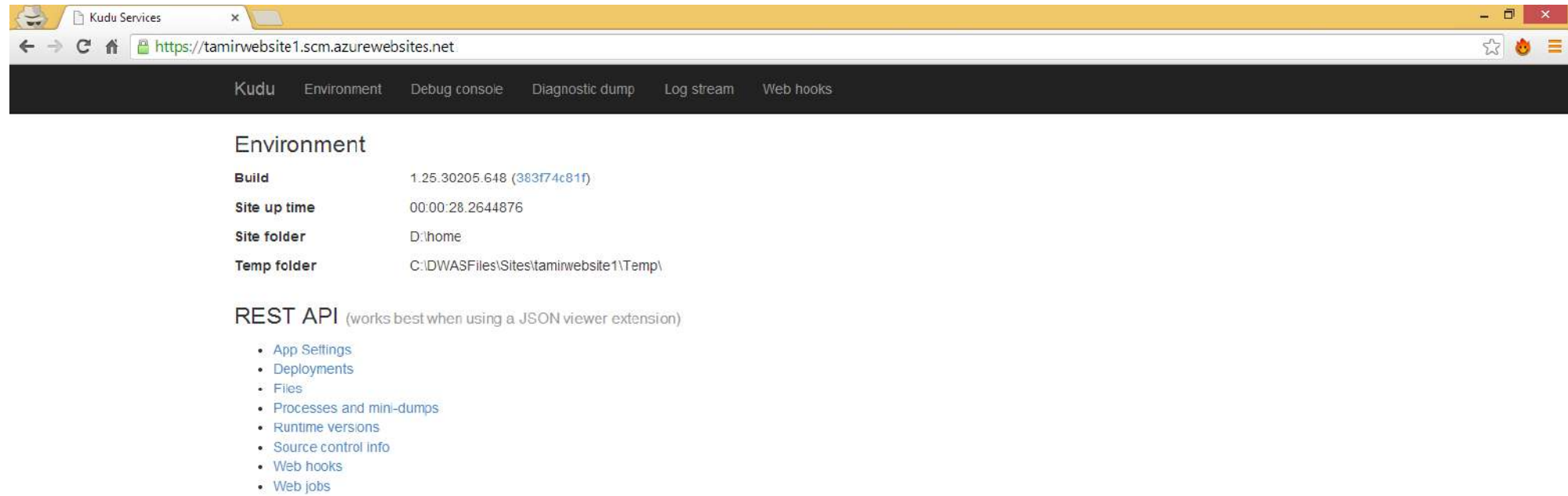


The background of the slide features a photograph of two hot air balloons. One balloon, on the left, has a blue and white checkered pattern. The other, on the right, is primarily orange with a red base. They are floating against a bright blue sky filled with soft, white clouds. The word "DEMO" is centered in the middle of the slide in a large, bold, blue font.

DEMO

Publishing from VS

- Every Azure web site has an associated Kudu service site
- Kudu is the engine behind [git deployments in Azure Web Sites](#)
- If your web site has URL
`http://mysite.azurewebsites.net/`
then the root URL of the Kudu service is
`https://mysite.scm.azurewebsites.net/`.
- Gives monitoring utils for the deployment



The screenshot shows a web browser window with the address bar displaying `https://tamirwebsite1.scm.azurewebsites.net`. The browser tab is labeled "Kudu Services". Below the address bar is a dark navigation bar with links: "Kudu", "Environment", "Debug console", "Diagnostic dump", "Log stream", and "Web hooks". The main content area is titled "Environment" and displays the following information:

Build	1.25.30205.648 (383f74c81f)
Site up time	00:00:28.2644876
Site folder	D:\home
Temp folder	C:\DWASFiles\Sites\tamirwebsite1\Temp\

Below the environment information is a section titled "REST API" with a note "(works best when using a JSON viewer extension)". It contains a list of links:

- [App Settings](#)
- [Deployments](#)
- [Files](#)
- [Processes and mini-dumps](#)
- [Runtime versions](#)
- [Source control info](#)
- [Web hooks](#)
- [Web jobs](#)

WebJobs

- Windows Azure Web App enables you to run custom jobs (running executables or scripts) on your web site
- The WebJobs SDK has a binding and trigger system which works with Windows Azure Storage Blobs, Queues and Tables.
- The trigger system calls a function in your code whenever any new data is received in a queue or blob.
- [You can create your own binders and triggers](#)
- Most of WebJobs functionality is now provided by Azure Functions



App Service Plan

- Represents a set of feature and capacity that can be shared across multiple apps in Azure App Service.
- This is the physical resources representative
- 5 pricing tiers – Free, Shared, Basic, Standard, Premium
- Apps can share the Service Plan if they are in the same subscription and same location
- A good usage for example is to share resources for each environment (DEV, TEST, PROD)



Mobile Apps

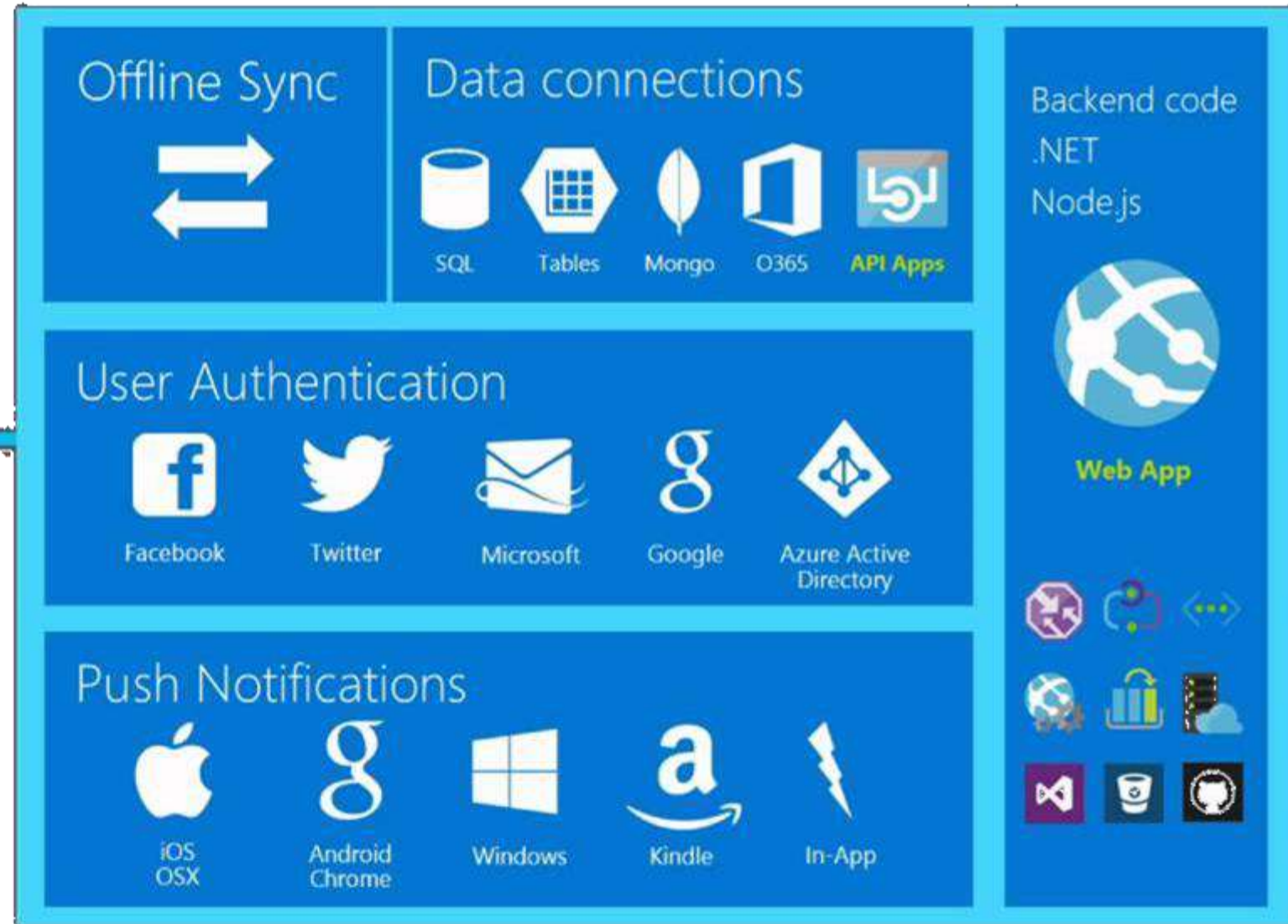
Azure App Services



What is Mobile Apps?



REST API





Structured Storage

- Powered by SQL Database
- Same DB – Multiple Mobile Services
- Data management in
 - Windows Azure Portal
 - SQL Portal
 - SQL Management Studio
 - REST API
 - CLI Tools
- JSON to SQL Type Mappings

Base REST API Endpoint URL

<https://Mobileservice.azure-mobile.net/tables/>*

Data Operations and their REST Equivalents

Action	HTTP Verb	URL Suffix
Create	POST	/TodoItem
Read	GET	/TodoItem?\$filter=id%3D42
Update	PATCH	/TodoItem/id
Delete	DELETE	/TodoItem/id



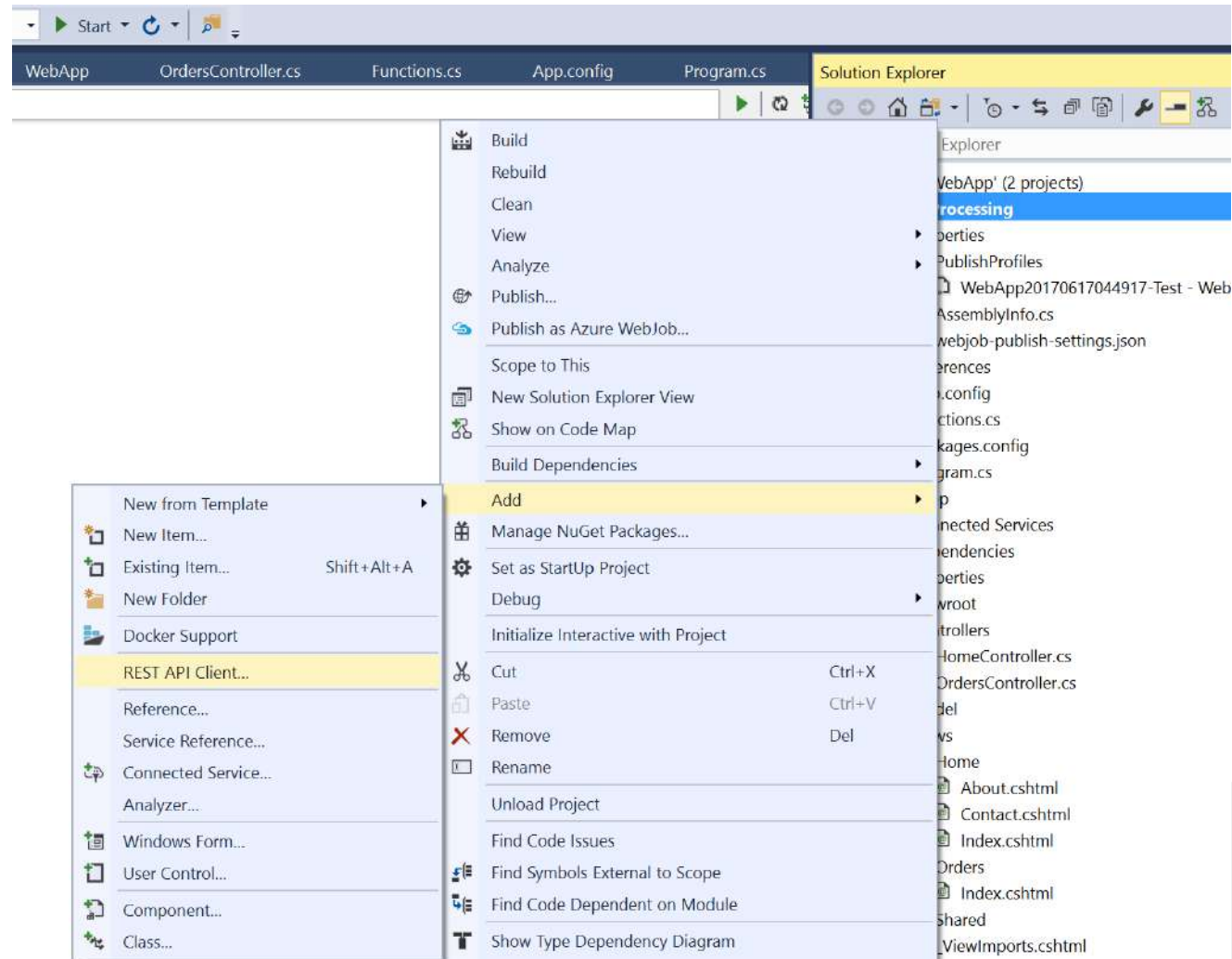
API Apps

Azure App Services

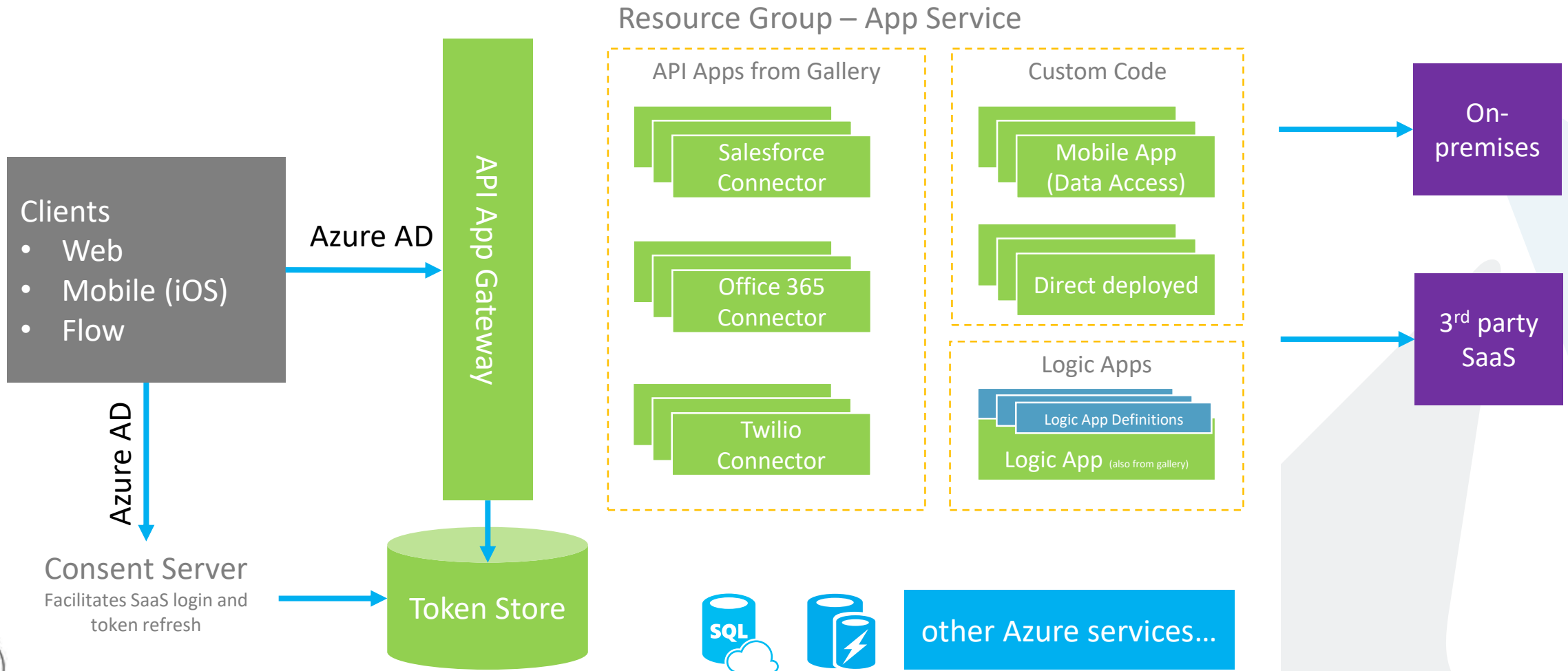
API Apps

- Basically Web Apps for Web Api's
- Simple access control
- Swagger metadata
- Logic App Integration
- Marketplace support for connectors
- VS tooling and support (for client side as well)

Creating REST Client from Swagger



API Apps Architecture Example



The background of the slide features two hot air balloons floating in a bright blue sky filled with soft, white clouds. The balloon on the left is white with blue and yellow checkered patterns. The balloon on the right is primarily orange with yellow and red accents. Both balloons have small baskets hanging from them.

Logic Apps

Azure App Services

Logic Apps

- Visually create business process and workflows based on Triggers and Actions
- Deliver integration capabilities in Web, Mobile, and API Apps
- Integrate with your SaaS and enterprise applications
- Automate EAI/B2B and business processes
- Connect to on-premises data



Logic App Designer

Microsoft Azure MyLogicApp - Logic App Designer

Search (Ctrl+/) Save Discard Run Designer Code view Templates Connectors Help


MyLogicApp - Logic App Designer
Logic app

Overview
Activity log
Access control (IAM)
Tags

DEVELOPMENT TOOLS

Logic App Designer
Logic App Code View
Versions

To use this template:

Office 365 Outlook
Office 365 Outlook 
[Switch account](#) [Update](#)

Office 365 Users
Sign in required [Sign in](#)

Continue



Built-in API Connectors

Connectors

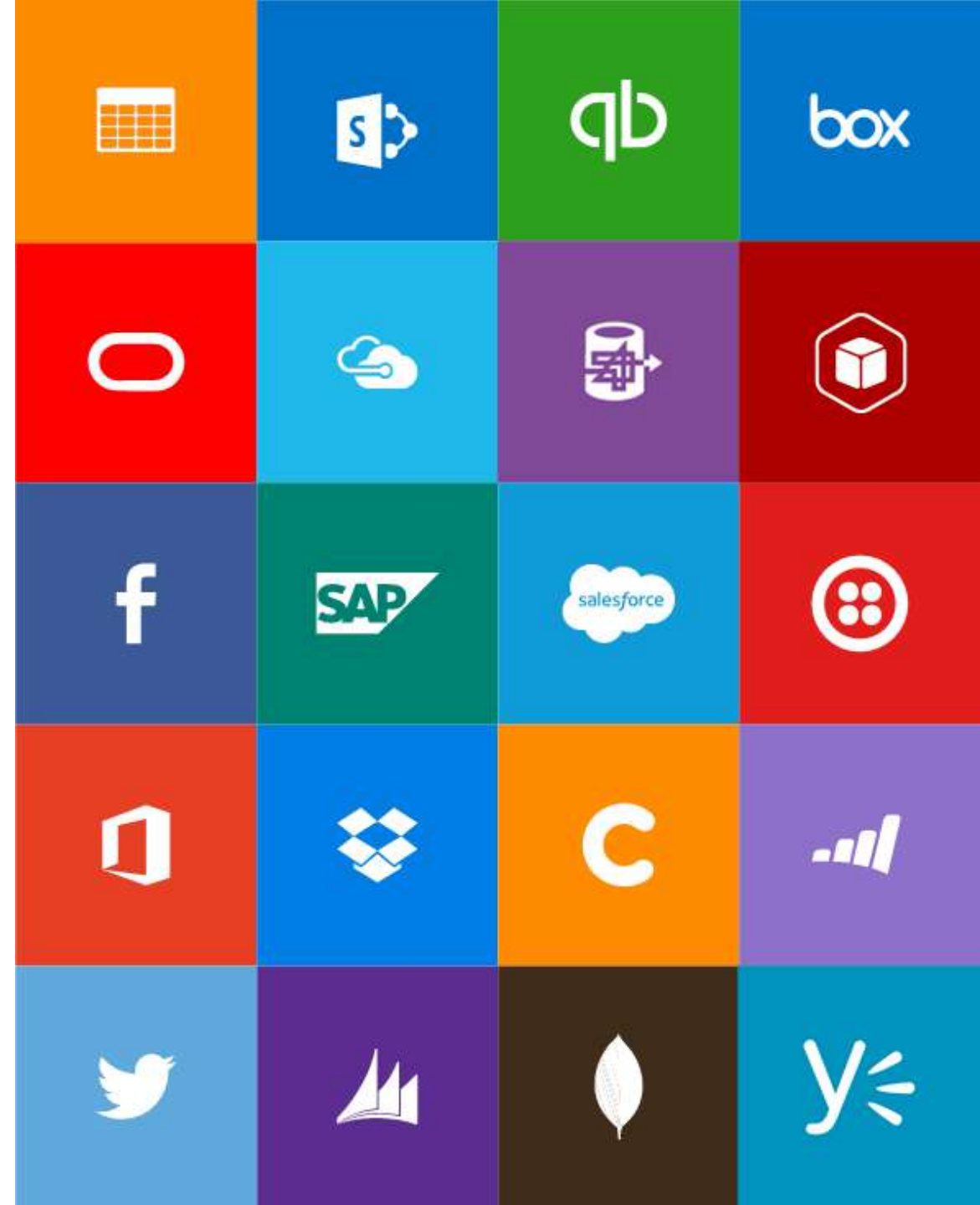
- Box
- Chatter
- Delay
- Dropbox
- Azure HD Insight
- Marketo
- Azure Media Services
- OneDrive
- SharePoint
- SQL Server
- Office 365
- Oracle
- QuickBooks
- SalesForce
- Sugar CRM
- SAP
- Azure Service Bus
- Azure Storage
- Timer / Recurrence
- Twilio
- Twitter
- IBM DB2
- Informix
- Websphere MQ
- Azure Web Jobs
- Yammer
- Dynamics CRM
- Dynamics AX
- Hybrid Connectivity

Protocols

- HTTP, HTTPS
- File
- Flat File
- FTP, SFTP
- POP3/IMAP
- SMTP
- SOAP + WCF

BizTalk Services

- Batching / Debatching
- Validate
- Extract (XPath)
- Transform (+Mapper)
- Convert (XML-JSON)
- Convert (XML-FF)
- X12
- EDIFACT
- AS2
- TPMOM
- Rules Engine





Azure Functions

- Azure Functions is an event driven, compute-on-demand experience
- Azure Functions scale based on demand and you pay only for the resources you consume.
- Function can be written in C# or nodeJS
- The runtime, otherwise known as the script host, is the underlying WebJobs SDK host which listens for events, gathers and sends data, and ultimately runs your code.

Azure Functions architecture

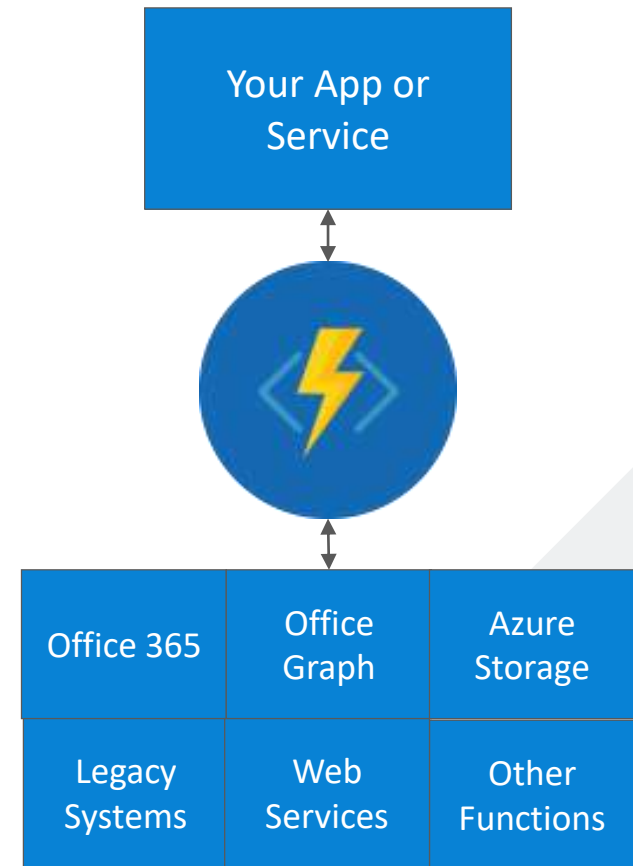
- Azure Functions is built around the WebJobs SDK runtime. The WebJobs SDK makes it easy to react to events and work with data in a consistent abstracted fashion.





Common Scenarios

- Timer-based processing
- Azure service event processing
- SaaS event processing
- Serverless web application architectures
- Serverless mobile backends
- Real-time stream processing
- Real-time bot messaging



Function App Templates

Function App templates are categorized into general areas of Timer, Data Processing, and Webhook & API

- BlobTrigger
- EventHubTrigger
- Generic webhook
- GitHub webhook
- HTTPTrigger
- QueueTrigger
- ServiceBusQueueTrigger
- ServiceBusTopicTrigger
- TimerTrigger
- Blank & Experimental

Choose a template

Language: Scenario:

BlobTrigger - C# A C# function that will be run whenever a blob is added to a specified container	BlobTrigger - Node A Node.js function that will be run whenever a blob is added to a specified container	Empty - C# An empty C# function without triggers, inputs, or outputs	Empty - Node An empty Node.js function without triggers, inputs, or outputs
EventHubTrigger - Node A Node.js function that will be run whenever an event hub receives a new event	Generic Webhook - C# A C# function that will be run whenever it receives a webhook request	Generic Webhook - Node A Node.js function that will be run whenever it receives a webhook request	GitHub WebHook - C# A C# function that will be run whenever it receives a GitHub webhook request

Azure Functions folder structure

► nodeJS

```
mynodefunction
| - function.json
| - index.js
| - node_modules
| | - ... packages ...
| - package.json
```

► C#

```
mycsharpfunction
| - function.json
| - run.csx
```




function.json

```
{
  "disabled": false,
  "bindings": [
    // ... bindings here
    {
      "type": "bindingType",
      "direction": "in",
      "name": "myParamName",
      // ... more depending on binding
    }
  ]
}
```

- Type - Binding type. For example, queueTrigger.
- direction - 'in', 'out'
Indicates whether the binding is for receiving data into the function or sending data from the function.
- name - The name that will be used for the bound data in the function.
For C# this will be an argument name; for JavaScript it will be the key in a key/value list.

logging

- To log output to your streaming logs in C#, you can include a `TraceWriter` typed argument. We recommend that you name it **log** or **logger**. It's recommend to avoid using `Console.WriteLine` in Azure Functions.

```
public static void Run(string myBlob, TraceWriter log)
{
    log.Verbose($"C# Blob trigger function processed: {myBlob}");
}
```

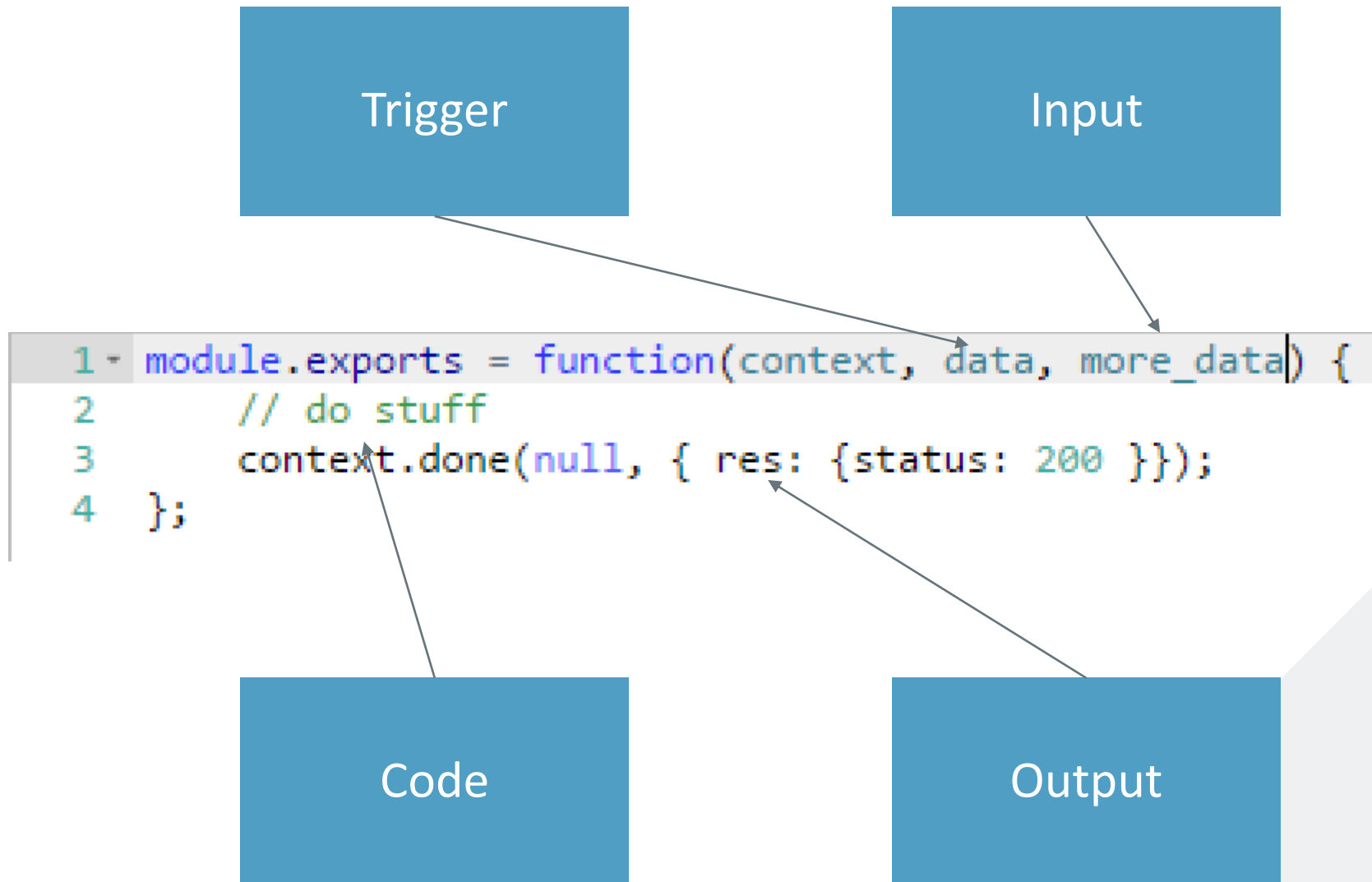

- A lightweight C# script
- Only the .NET Framework 4.6 is supported
- If you need to reference a private assembly, you can upload the assembly file into a bin folder relative to your function and reference it by using the file name
 - `#r "AssemblyName"`
- Supports Nuget by adding the `packages.json`
 - When you upload a *project.json* file, the runtime gets the packages and automatically adds references to the package assemblies
- Other *.csx files can be reused by adding `#load "myfile.csx"`



Dynamic tier pricing

- Pay per execution model - two meters, three units
 - Number of executions
 - Duration of execution x reserved memory

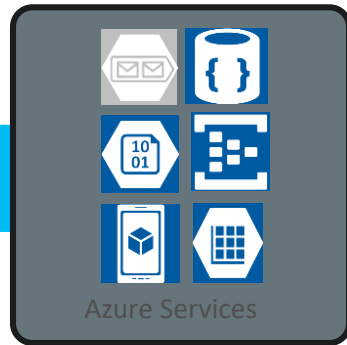
Functions programming concepts



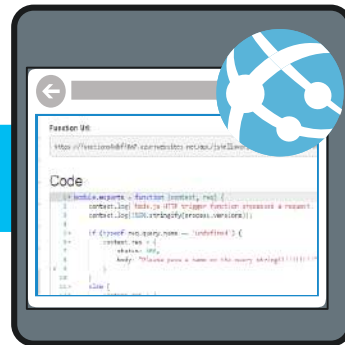
1) Trigger



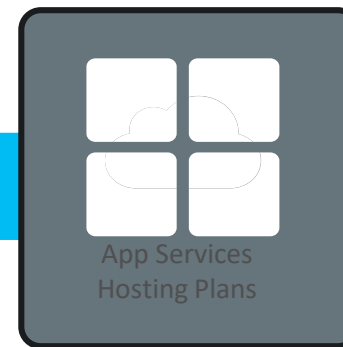
2) Input Binding



3) Develop



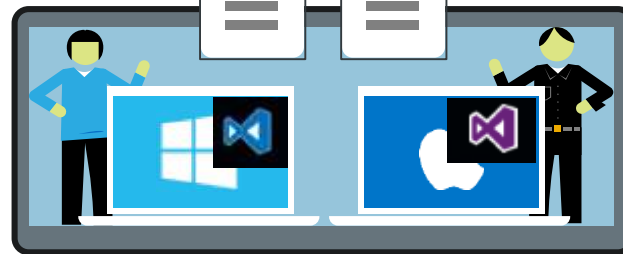
4) Execute



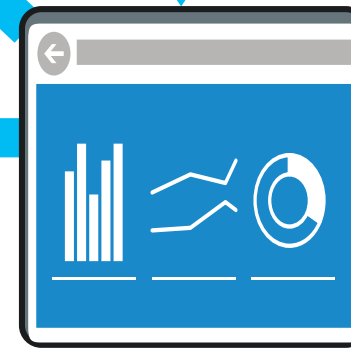
5) Output Binding



7) Develop Locally



6) Monitor and Improve



Benefits of “serverless”

- “Pinnacle of PaaS compute”
- Not just hardware “servers”, but software servers are also **managed for you**
- Focus on **business logic**, not solving technical problems not **core to business**
- Lower effort to get started makes it easier to experiment (bots, etc.)



Signs that a serverless pattern might be useful for a given scenario

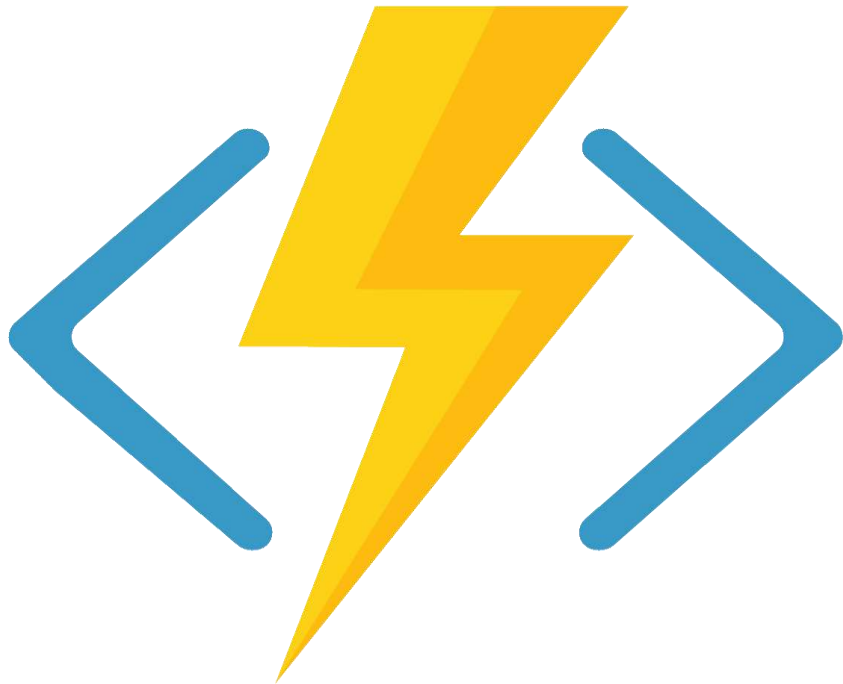
1. Stateless → Scale
2. Too complicated to deploy a traditional backend
3. Workload is sporadic (very low & high scale)
4. (Human) Operational costs need to stay low
5. Lots of different services involved

Suggestions for getting started

1. For existing services, start small. Replace 1 API or background processing item
2. Integration is a great place to introduce serverless, because it is often a new layer on top of old layers
3. For new services, establish a pattern early and stick with it. Lack of tooling/established patterns mean you pay an early adopter tax. Build automation asap

Get started and reach out!

- Try Azure - <https://azure.microsoft.com/en-us/free/>
- Try Functions – <https://functions.azure.com>
- Try App Service – <https://tryappservice.azure.com>



Advanced Messaging

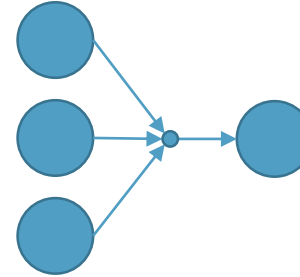


Communication Patterns



Synchronous
Request-Reply

- A.K.A RPC – Remote Procedure Call
- Client synchronously wait for the server response
- Connection remains open -> Increase load on server
- Sent message is not durable



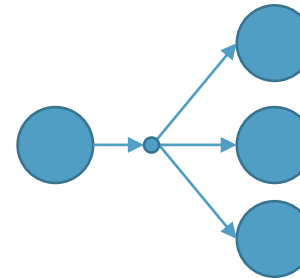
Fan in

- Server receives messages asynchronously from multiple producers
- Decoupling of client and server



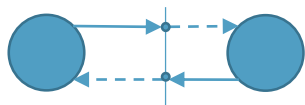
Fire and Forget

- After the server acknowledge, client continues without waiting for response (not even for operation completion)
- Sent message is not durable



Fan out

- A.K.A Publish-Subscribe (PubSub)
- The producer broadcasts a message
- Decoupling of client and server

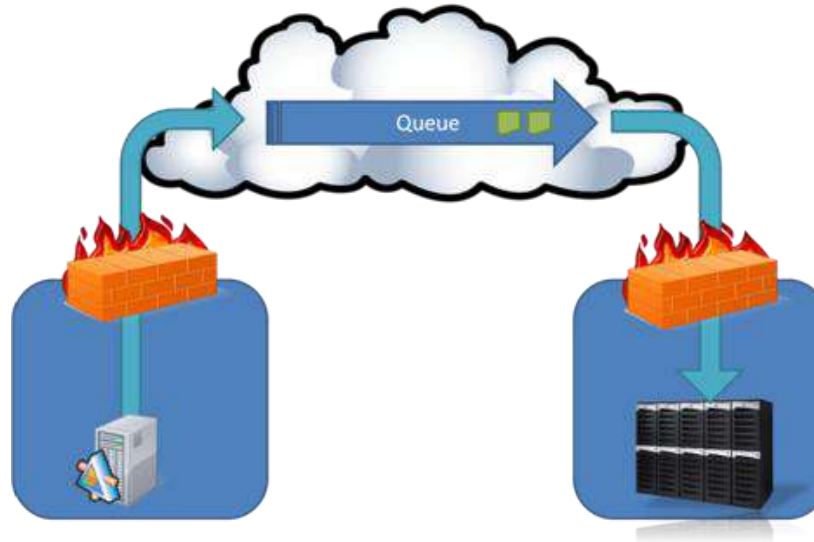


Asynchronous
Request-Reply

- Decoupling of client and server
- The server asynchronously process the message and post a response
- The client asynchronously process the response

Service Bus/Queue Scenarios

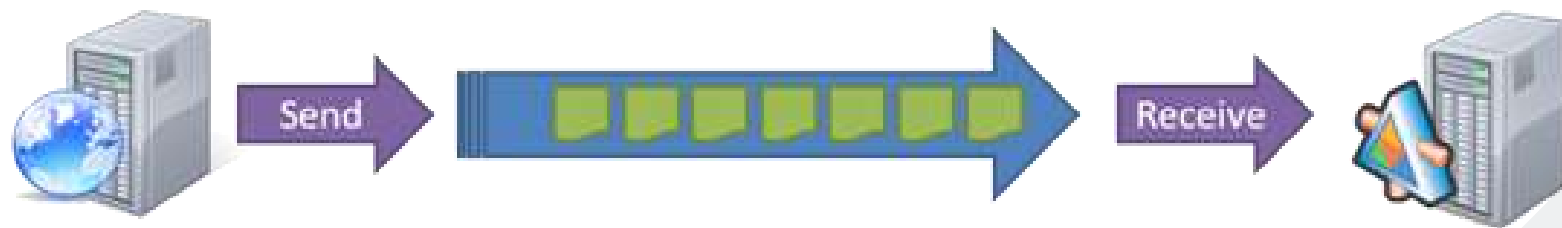
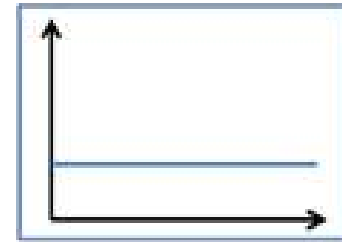
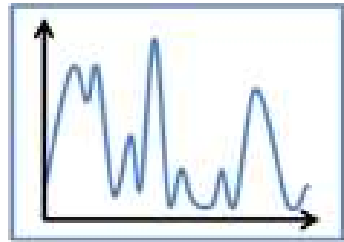
➤ Secure Network Traversal



<http://www.cloudcasts.net/devguide>

Service Bus/Queue Scenarios

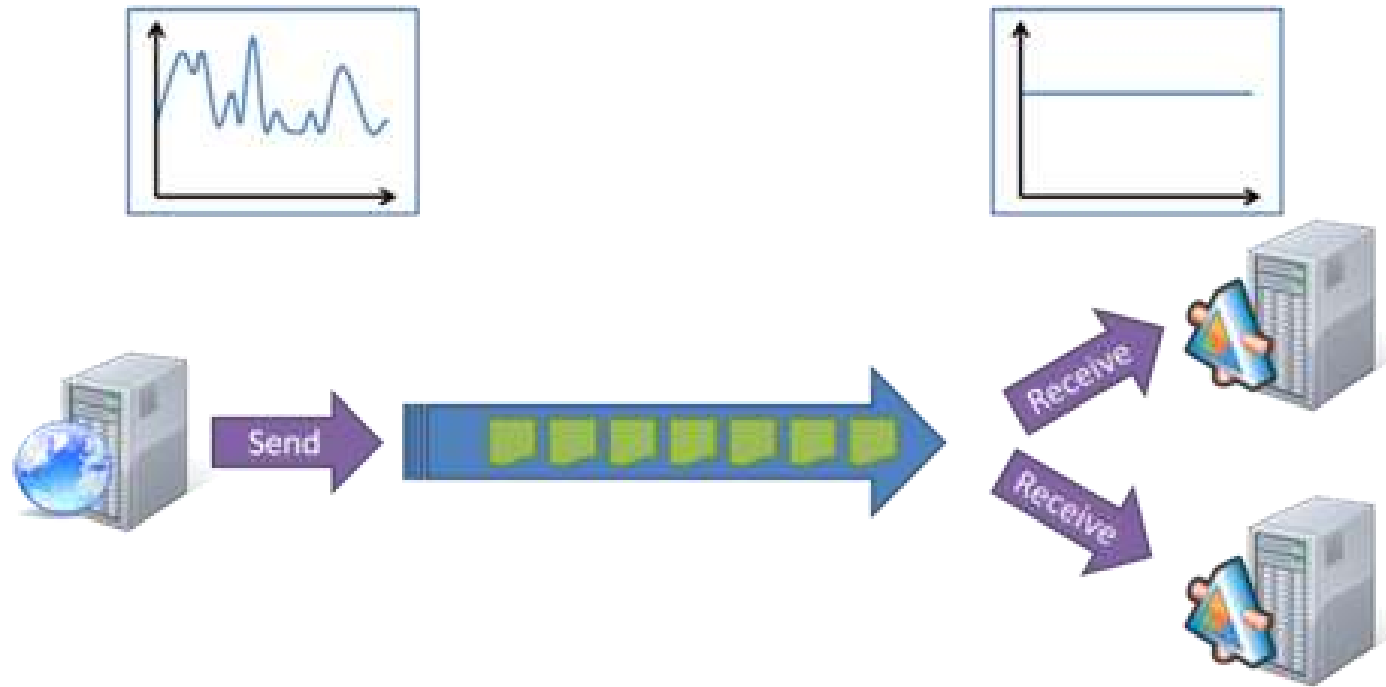
► Load Leveling



<http://www.cloudcasts.net/devguide>

Service Bus/Queue Scenarios

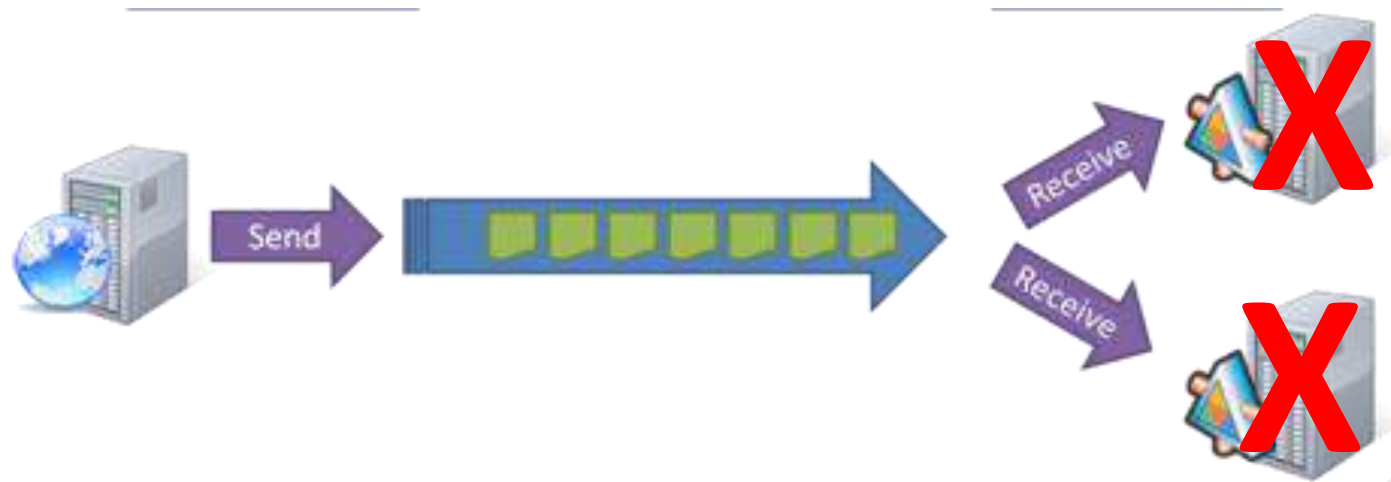
► Load Balancing



<http://www.cloudcasts.net/devguide>

Service Bus/Queue Scenarios

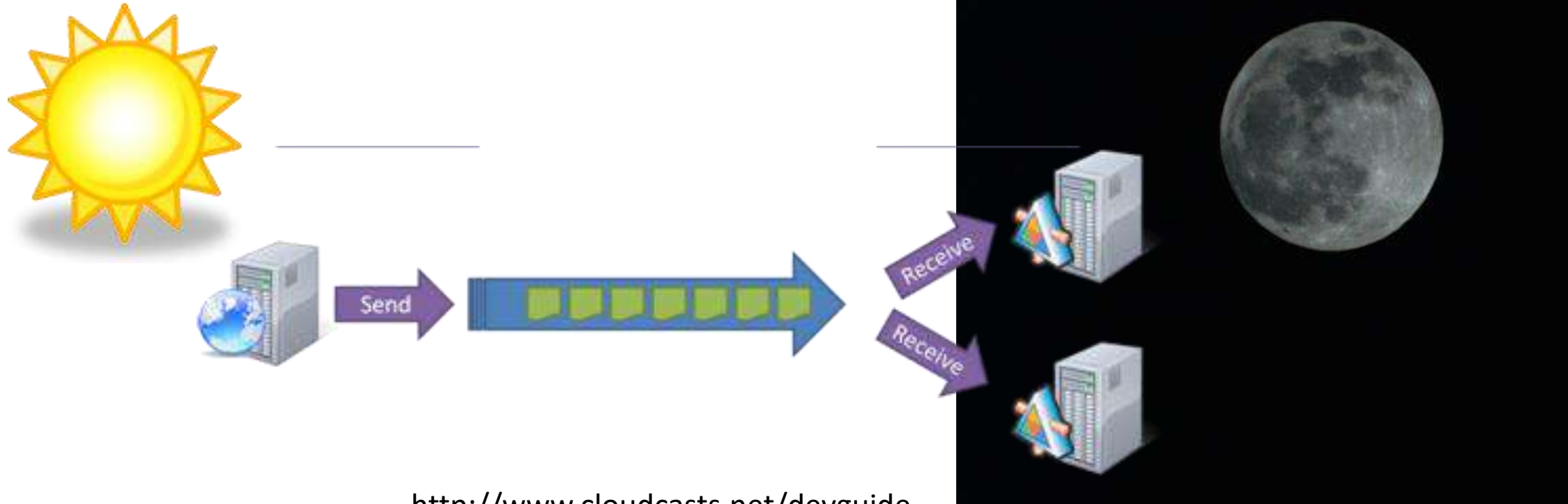
► Resilience against Service Failure



<http://www.cloudcasts.net/devguide>

Service Bus/Queue Scenarios

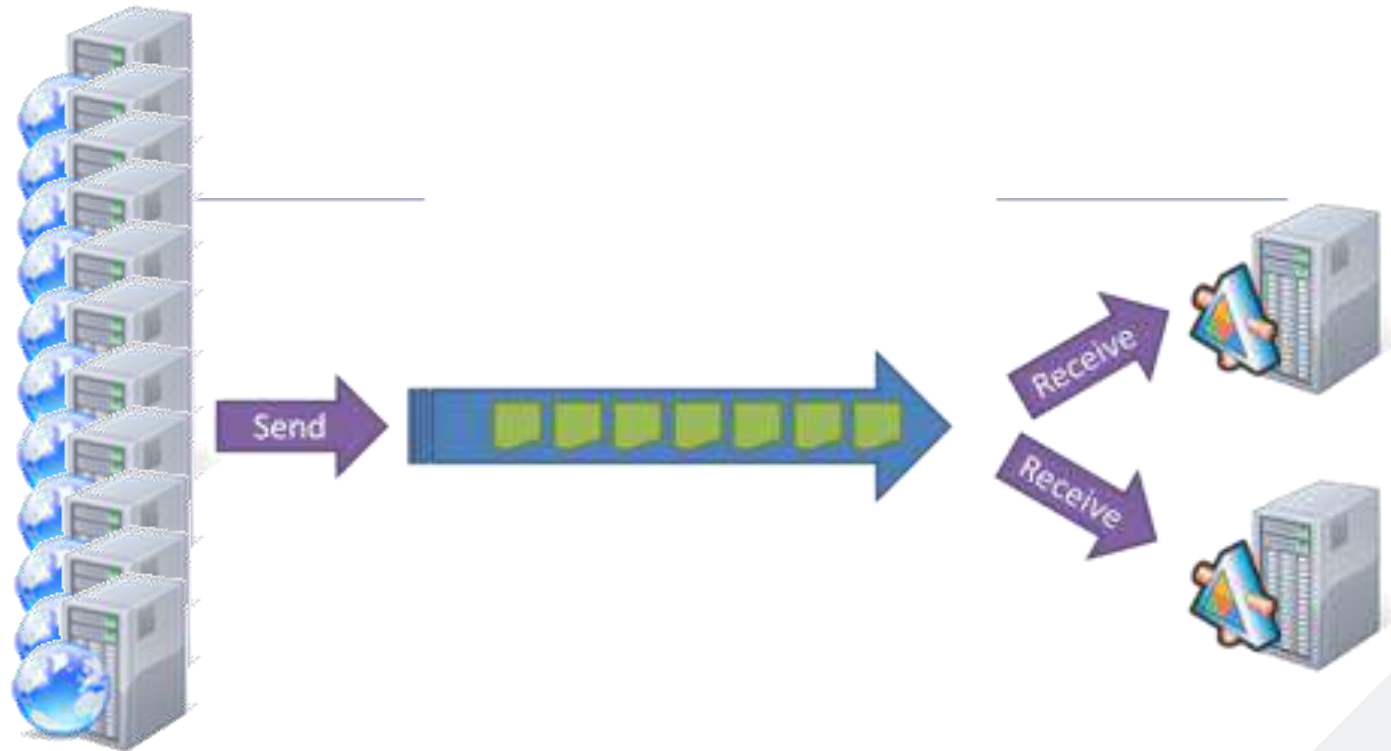
► End of Day Processing



<http://www.cloudcasts.net/devguide>

Service Bus/Queue Scenarios

► Hyper scale data ingress (Event Hub)



The background of the slide features two hot air balloons floating in a bright blue sky filled with soft, white clouds. The balloon on the left is white with blue and yellow checkered patterns. The balloon on the right is primarily orange with yellow and red accents. Both balloons have wicker baskets hanging from them.

Azure Service Bus

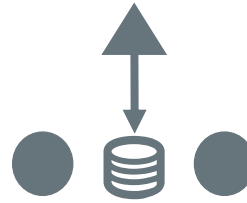
Communication Patterns



Azure Service Bus

➤ Messaging

- Queuing
- Topics (Pub/Sub)
- Reliable Transfer



➤ Connectivity

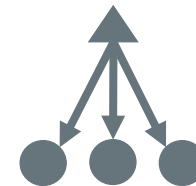
- Service Relay
- Protocol tunneling



➤ Hyper scale data ingestion

➤ Notification Hub

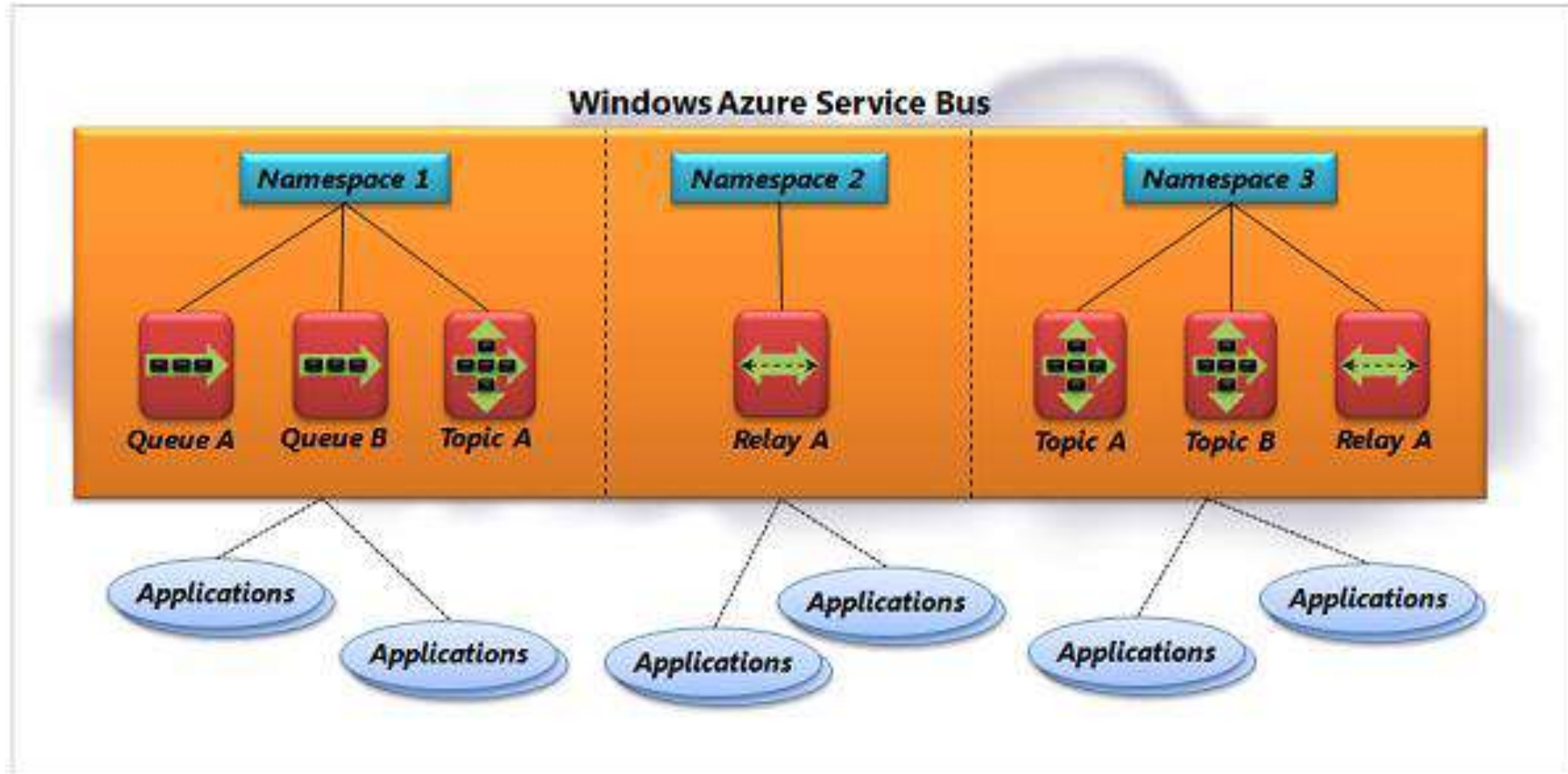
- Scalable Push notifications
- Multi platform



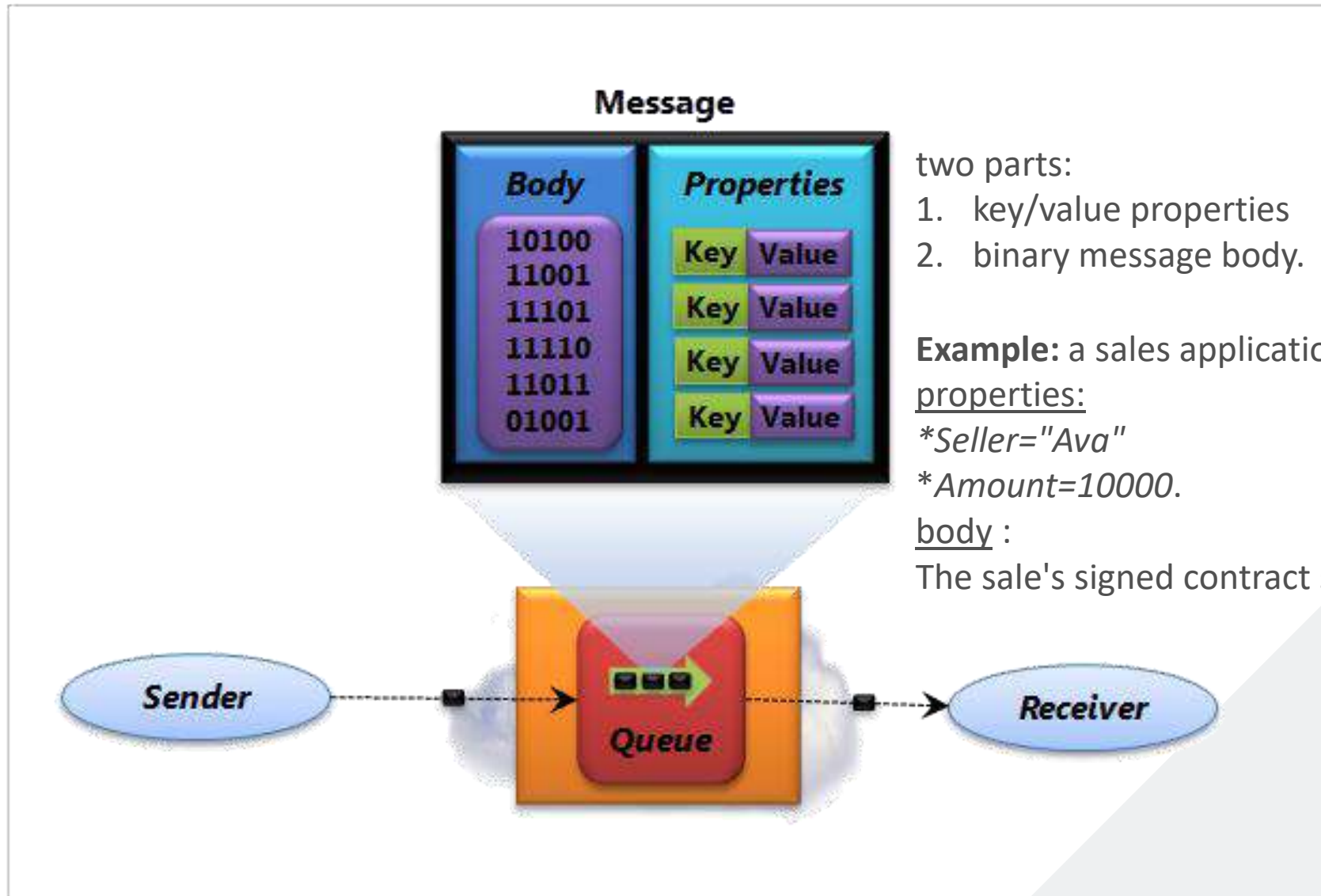
Service Bus Messaging

The background of the slide features a soft-focus image of two hot air balloons. One balloon, on the left, has a blue and white checkered pattern. The other, on the right, is primarily orange with some yellow and red accents. They are floating against a bright blue sky filled with wispy, white clouds. The overall aesthetic is clean and modern.

Service Bus Messaging



Service Bus Messaging - Queue



two parts:

1. key/value properties
2. binary message body.

Example: a sales application sends message properties:

**Seller="Ava"*

**Amount=10000.*

body :

The sale's signed contract scanned image

Service Bus Messaging - Queue

```
Uri managementUri =
    ServiceBusEnvironment.CreateServiceUri("sb", "ServiceBusNamespace", string.Empty);
var sharedSecretTokenProvider =
    TokenProvider.CreateSharedSecretTokenProvider( "[ServiceBusIssuerName]",
                                                    "[ServiceBusIssuerKey]");
var namespaceManager = new NamespaceManager(managementUri, sharedSecretTokenProvider);

var queueDescription=namespaceManager.CreateQueue("[QueuePath]");
var queueClient = QueueClient.Create(queueDescription.Path);

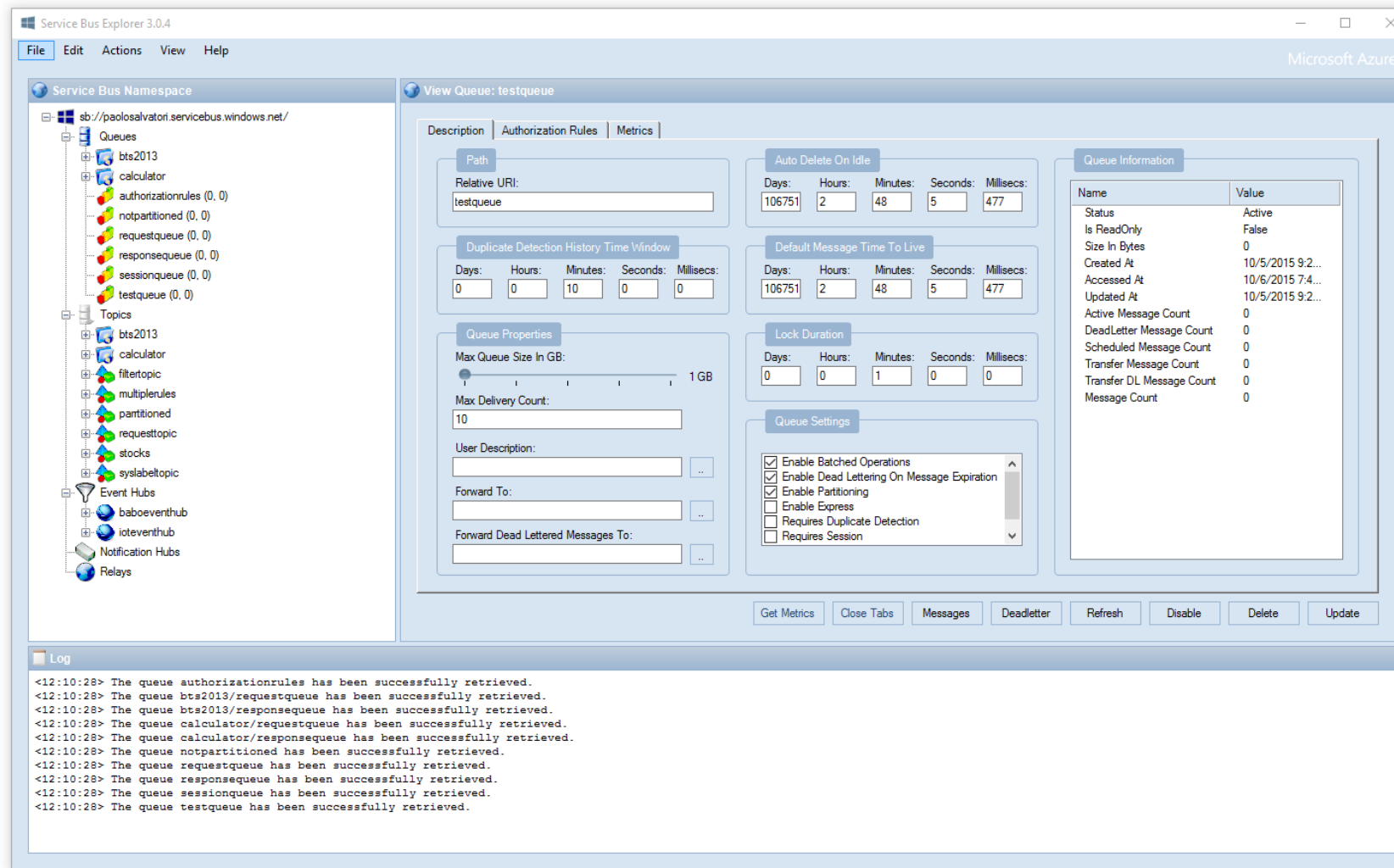
var someSerializableObject = new SomeSerializableType();
var brokeredMessageToSend = new BrokeredMessage(someSerializableObject);
brokeredMessageToSend.Properties["key"] = "val";
queueClient.Send(brokeredMessageToSend);

var recievedBrokerdMessage = queueClient.Receive();
var someSerializableType = recievedBrokerdMessage.GetBody<SomeSerializableType>();
var peekedBrokeredMessage = queueClient.Peek();
```




Service Bus Explorer

► <https://github.com/paoloselvatori/ServiceBusExplorer>



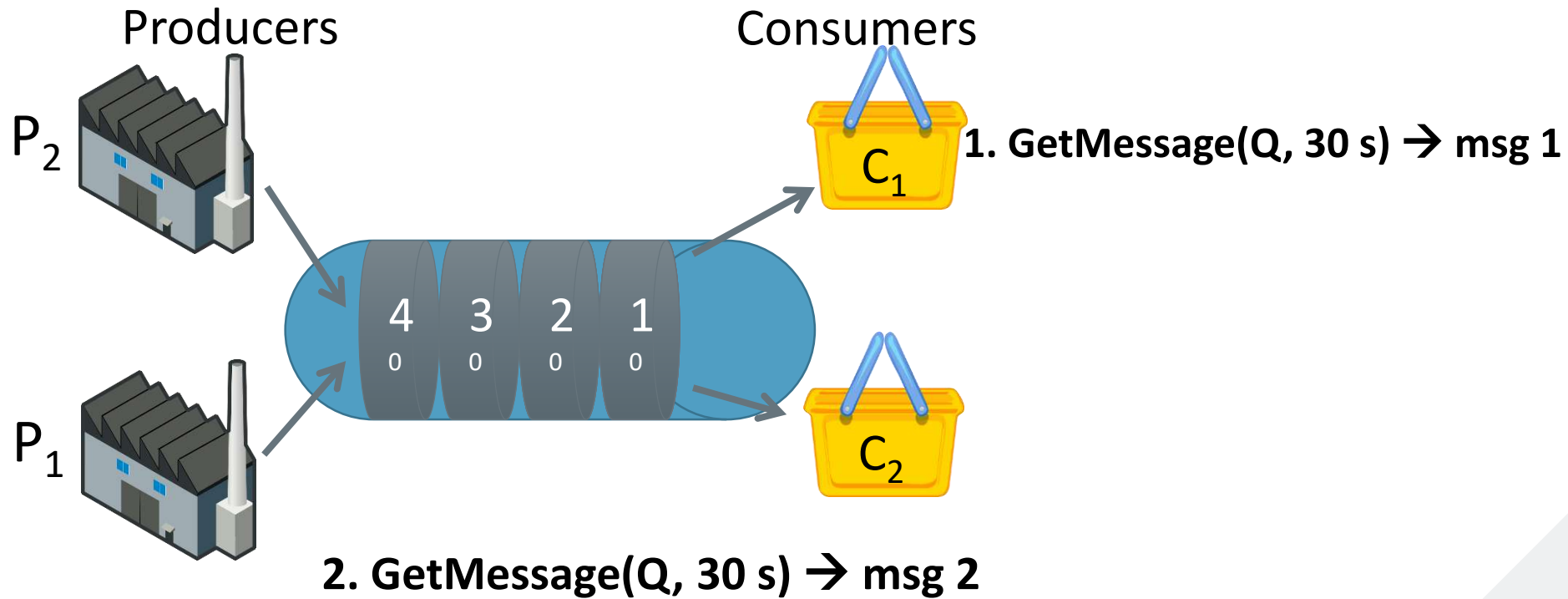
Service Bus Messaging – Queue – Event Driven

```
var eventDrivenMessagingOptions = new OnMessageOptions
{
    AutoComplete = true,
    MaxConcurrentCalls = 5
};
eventDrivenMessagingOptions.ExceptionReceived += OnExceptionReceived;
queueClient.OnMessage(OnMessageArrived, eventDrivenMessagingOptions);

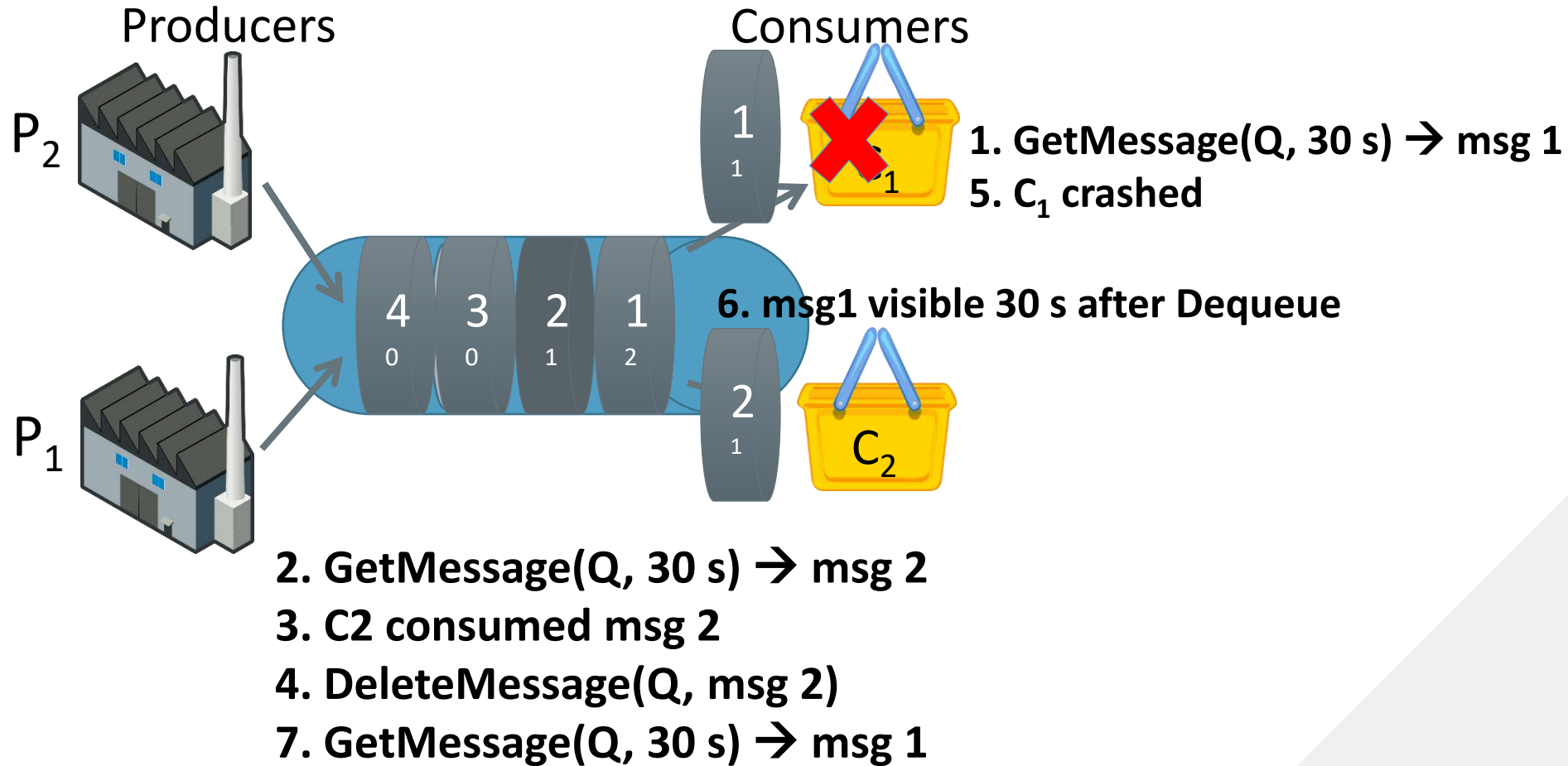
private void OnMessageArrived(BrokeredMessage obj)
{
    //do something
}

private void OnExceptionReceived(object sender, ExceptionReceivedEventArgs e)
{
    //do something
}
```

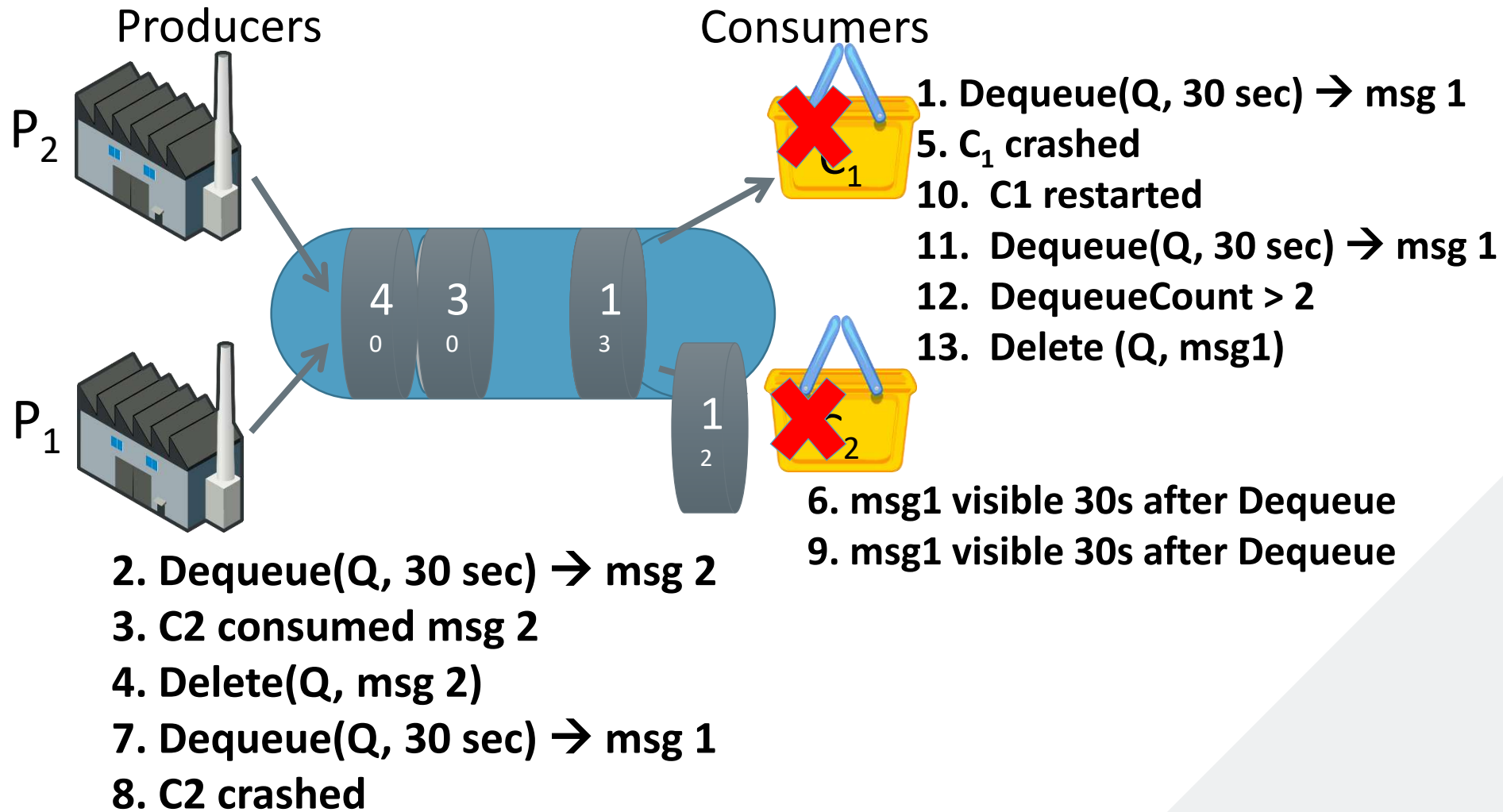

Removing Poison Messages



Removing Poison Messages



Removing Poison Messages





Poison Messages

- Message can cause the consumer to crash
- Detecting “Poison Messages”
 - For Storage Queues - examine the [DequeueCount](#) property of the message.
 - Two options
 1. Delete the message
 2. Store in Poison Queue/Table
- 1. Azure Service bus
 1. automatically done by setting the [QueueDescription.MaxDeliveryCount](#) and [SubscriptionDescription.MaxDeliveryCount](#) properties
 2. Explicitly calling the **DeadLetter()** method

Service Bus Messaging – Dead letters

- If a receiver finishes processing a message successfully then it should call **Completed** method
- If a receiver application is unable to process the message for some reason, then it can call the **Abandon** method
 - This will cause the Service Bus to unlock the message within the queue and make it available to be received again
- If the application crashes after processing the message but before the **Complete** request is issued, then the message will be redelivered to the application when it restarts
- We can mark failing message as dead-letter using the **DeadLetter()** method

Service Bus Messaging – Dead letters

- When we mark the message as dead letter. The message is moved to dead letter queue.
- the name of the sub-queue is [queueName]/\$DeadLetterQueue
- The path can be obtained using the **FormatDeadLetterPath** method of the QueueClient
- This sub-queue can be consumed by any other system or consumer and check the messages, log them and so on.
- It is not possible to add the message to the original queue.

Service Bus Messaging – Dead letters

```
BrokeredMessage receivedMessage;
while ((receivedMessage = queueClient.Receive(TimeSpan.FromSeconds(10))) != null)
{
    int retryCount = 0;
    while (retryCount < MaxRetryCount)
    {
        if (ProcessOrder(receivedMessage))
            break;
        else
            retryCount++;
    }

    if (retryCount == MaxRetryCount)
    {
        receivedMessage.DeadLetter( "UnableToProcess",
                                    "Failed to process in reasonable attempts");
    }
}
```

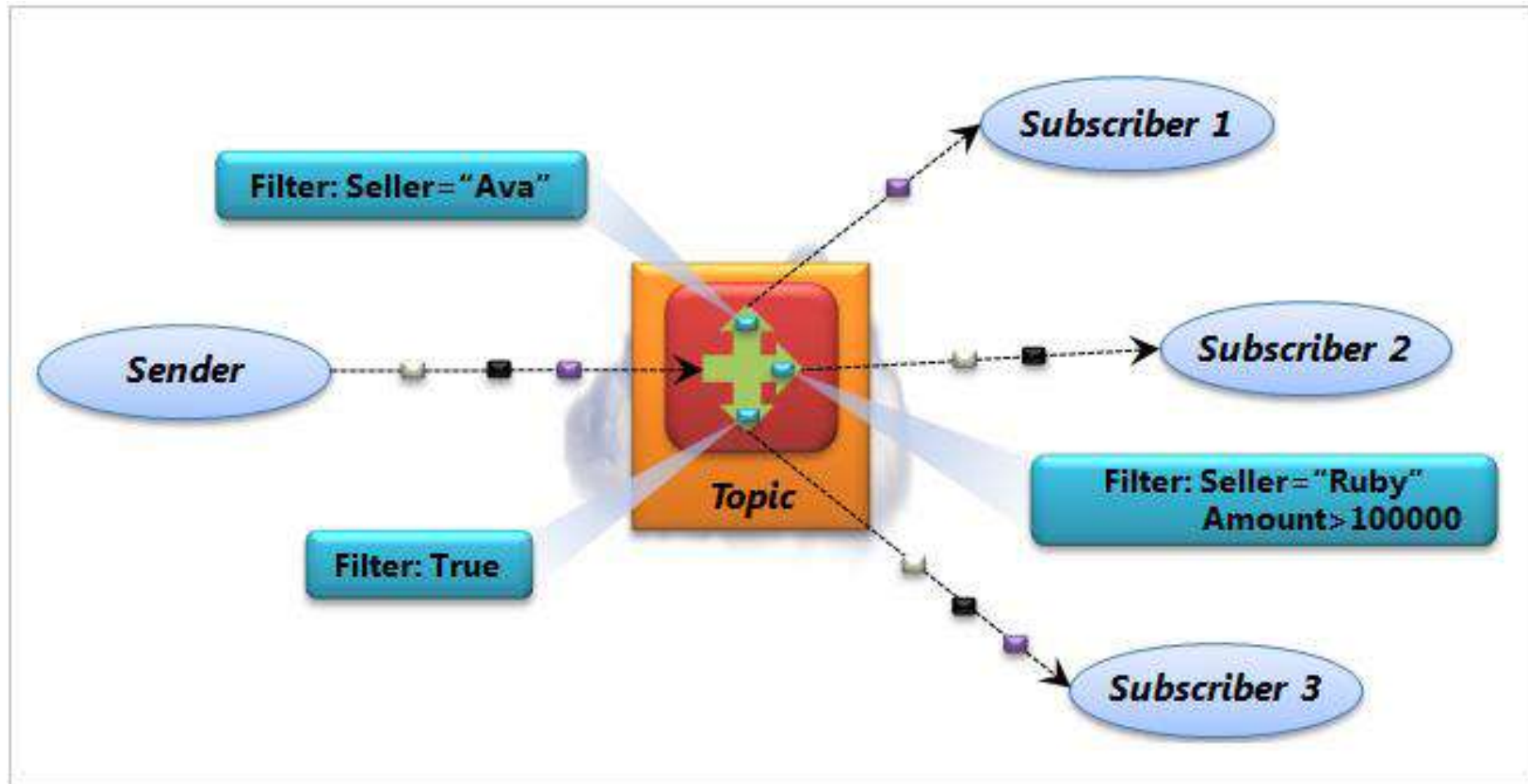

Service Bus Messaging – Dead letters

```
// Log the dead-lettered messages that could not be processed:
var deadQueuePath=
    QueueClient.Create(QueueClient.FormatDeadLetterPath(queueClient.Path)

QueueClient deadLetterCInt =
    QueueClient.Create(deadQueuePath, ReceiveMode.ReceiveAndDelete);

BrokeredMessage receivedDeadLetterMessage;
while ((receivedDeadLetterMessage = deadLetterCInt.Receive(TimeSpan.FromSeconds(10))) != null)
{
    LogOrder(receivedDeadLetterMessage);
}
```


Service Bus Messaging – Topics



Service Bus Messaging – Topics

- A topic is similar in many ways to a queue.
- topics let each receiving application create its own subscription by defining a *filter*.
- A subscriber will then see only the messages that match that filter.
- Unlike queues, however, a single message sent to a topic can be received by multiple subscribers.
- ***publish and subscribe***

Service Bus Messaging – Topics

- Up to 2000 rules per topic
- Each matched rule yield a message copy
- Types:
 - SqlFilter - SQL92 expressions over message properties
 - CorrelationFilter
 - FlaseFilter
 - TrueFilter
- Filter can help for
 - Routing based on content
 - Partitioning aware message distribution – without sender involvement

Service Bus Messaging – Topics

```
TopicDescription myTopic = namespaceManager.CreateTopic("DataCollectionTopic");
Uri serviceUri = ServiceBusEnvironment.CreateServiceUri("sb", "ServiceBusNamespace",
string.Empty);
var tokenProvider = TokenProvider.CreateSharedSecretTokenProvider("[ServiceBusIssuerName]",
                                                                    "[ServiceBusIssuerKey]");
MessagingFactory factory = MessagingFactory.Create(serviceUri, tokenProvider);
TopicClient myTopicClient = factory.CreateTopicClient(myTopic.Path);

var brokeredMessage = new BrokeredMessage(someSerializableObject);
brokeredMessage.Properties["MessageCategory"] = "Inventory";
myTopicClient.Send(brokeredMessage);
```

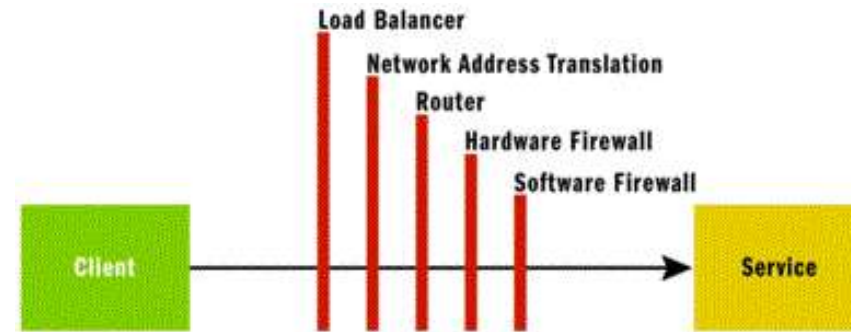

Service Bus Messaging – Topics

```
SqlFilter inventoryFilter =  
    new SqlFilter("MessageCategory = Inventory");//MessageCategory is a property in the message  
SubscriptionDescription myAgentSubscription =  
    namespaceManager.CreateSubscription(myTopic.Path, "Inventory");  
  
var subscriptionClient = SubscriptionClient.Create(myTopic.Path, "Inventory", inventoryFilter);  
BrokeredMessage message = subscriptionClient.Receive();
```




Service Bus– Relay

- IPv4 is running out
 - Dynamic DNS
 - Network Address Translation (NAT)
 - Load Balancers
 - Routers
 - Hardware Firewall
 - Software Firewall
-
- Web Services simply don't work
 - What about calling back to the client?



* From [MSDN Magazine](#)



Service Bus – Relay

- Intermediary Pass-Through Service (Push)
 - Overcome connectivity challenges
- Uses outbound connections only
 - Defaults to TCP with fallback to HTTP
- Relays client calls to service
- Hosted in the cloud
 - Scalability
 - Security
 - Management Portal

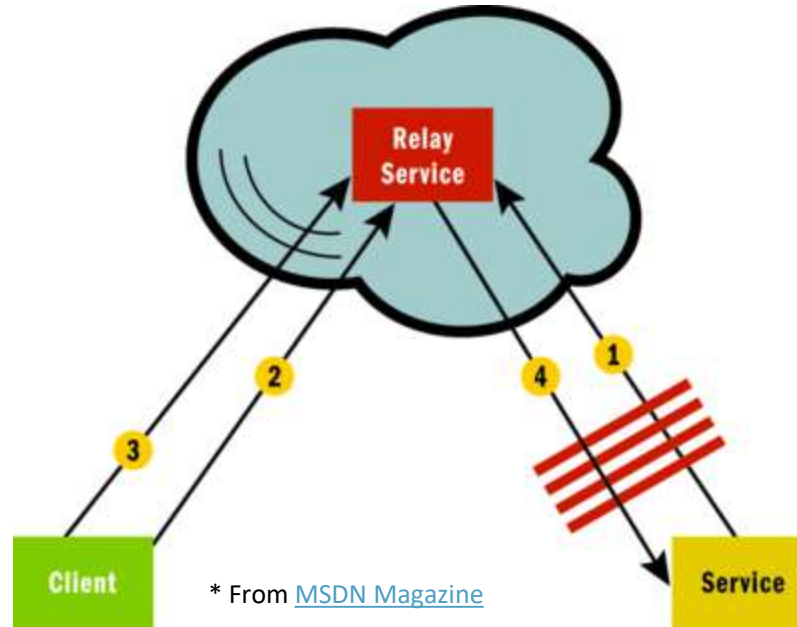


* From [MSDN](#)



Connectivity Relay

- Service connects and authenticates against the relay
 - Relay determines how to communicate with the service
- Client authenticates and calls the service
 - Relay forwards the message to the service



* From [MSDN Magazine](#)

Service Address and Registry

- Service Bus Address
 - `[scheme]:[ns].servicebus.windows.net/{uri}`
 - `sb://MyCompany.servicebus.windows.net/CalcService`
- Service Bus Registry
 - ATOM-based feed of online services
 - `http://MyCompany.servicebus.windows.net/`
- Need to enable publishing to registry
 - Add ServiceRegistrySettings endpoint behavior configured with public discovery type

Summary

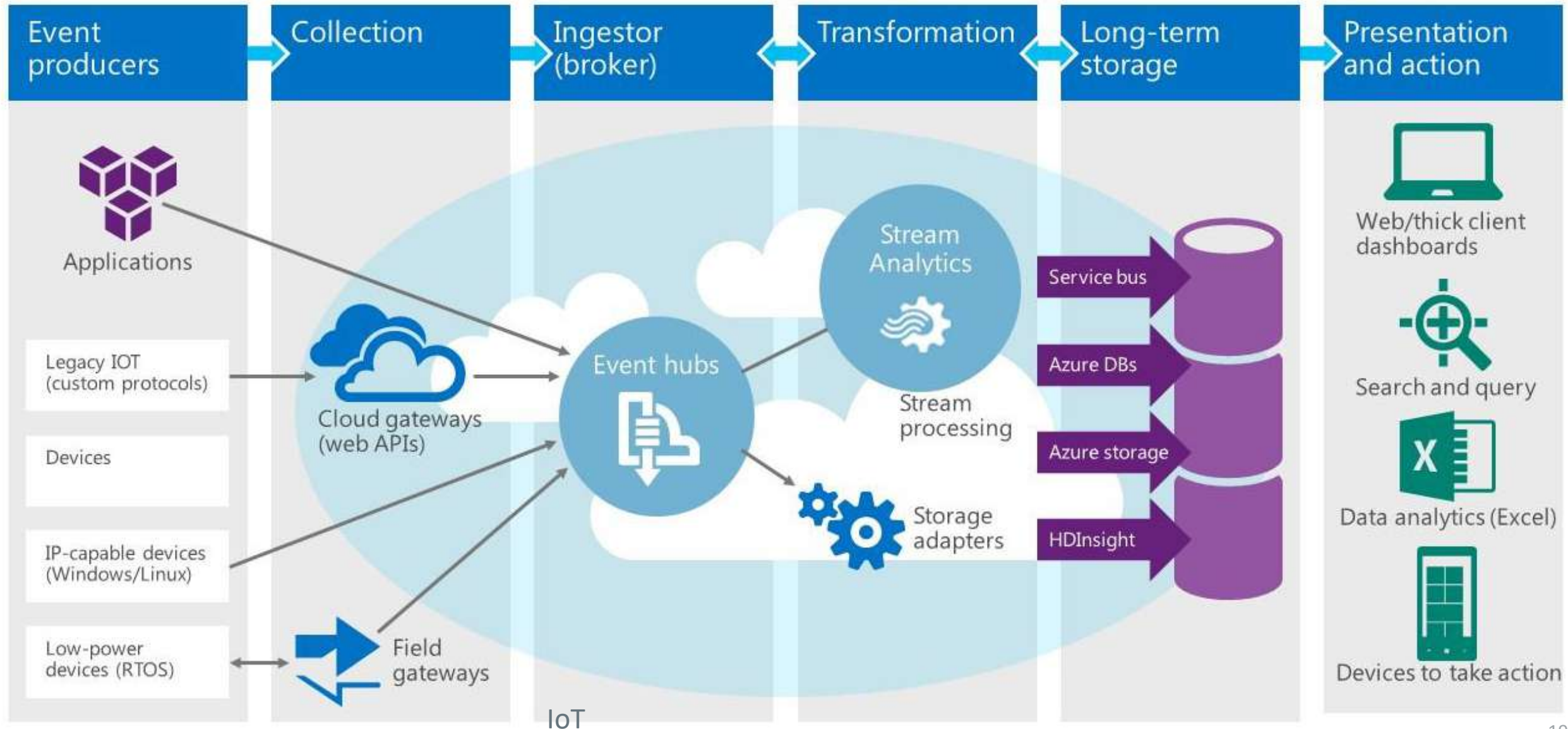
- Azure Service Bus is a key component for many connectivity scenarios
- Highly flexible and robust messaging & connectivity solution
 - Skype in a box
- Can be installed on premise
 - Service Bus for Windows Server
- PaaS – no maintenance or setup issues
 - Simply configure and use

Event Hubs

Event Hubs

- Highly scalable data ingress service
- Can ingest millions of events per second
- Act as the "front door" for an event pipeline
 - Once data is collected into an Event Hub, it can be transformed and stored using any real-time analytics provider or batching/storage adapters.
- Decouples the production of a stream of events from the consumption of those events
 - Event consumers can access the events on their own schedule
- Different from traditional queues
 - Journal Logging
 - Similar to Apache Kafka

Azure Based High-Throughput Ingest Architecture Microsoft



Event Hubs

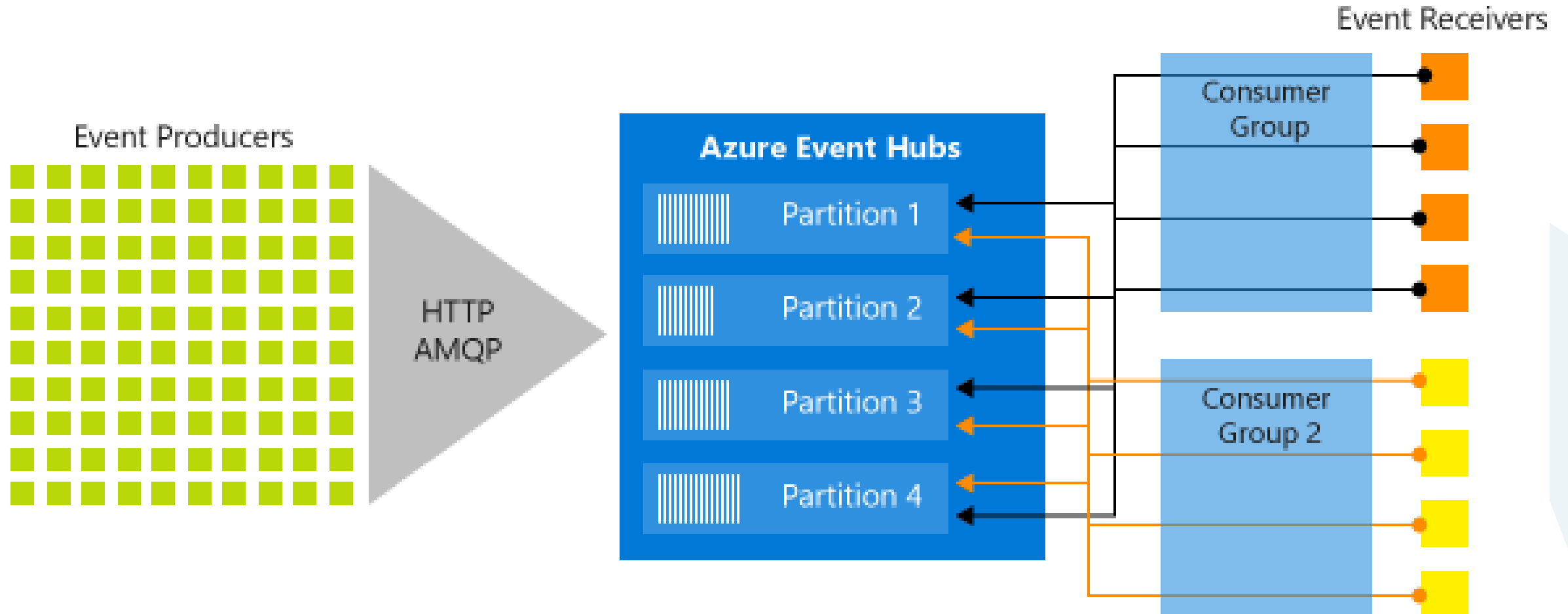
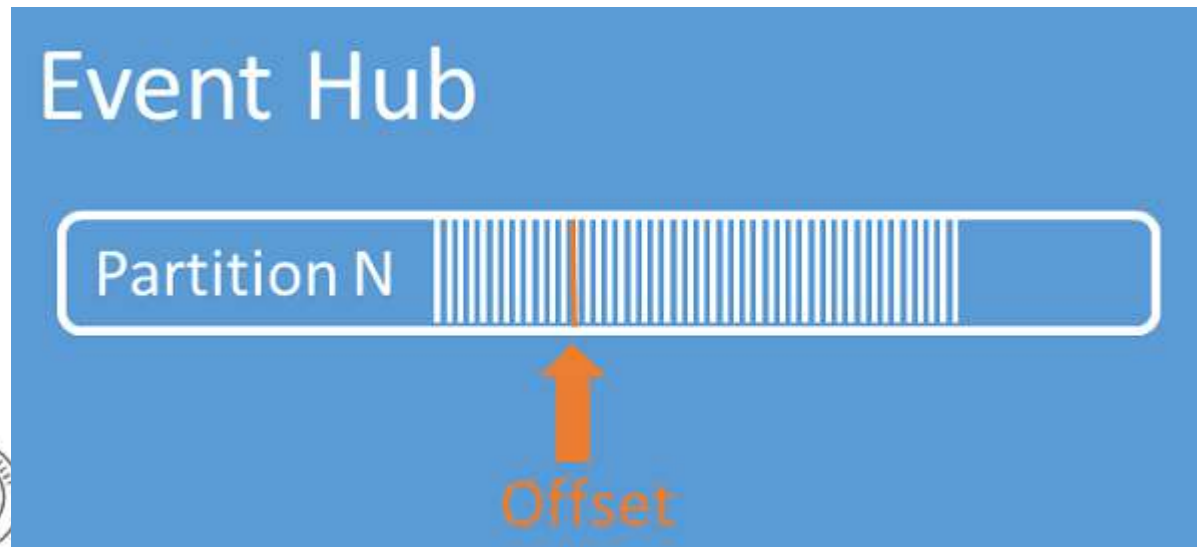
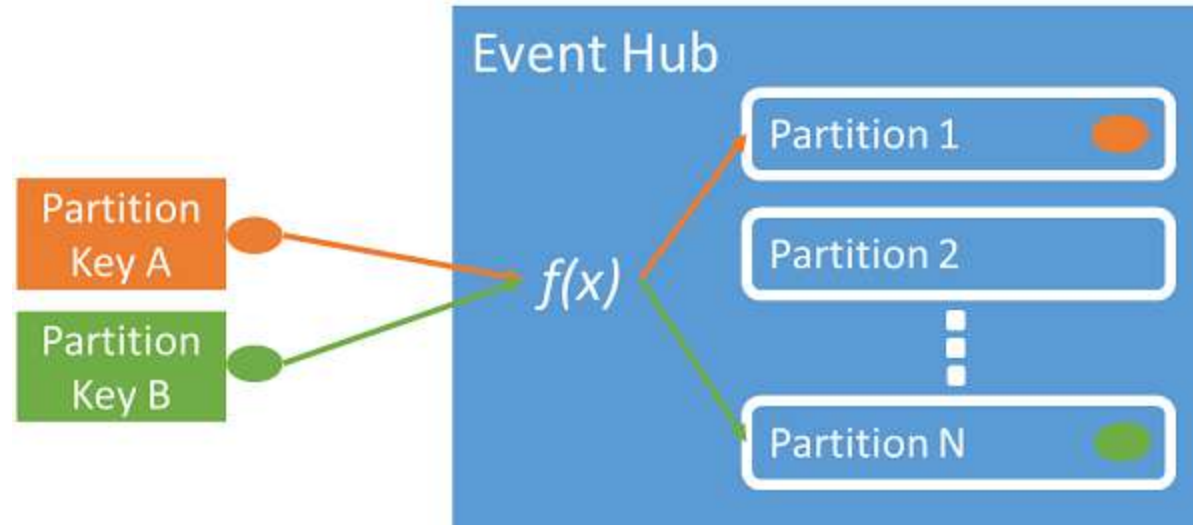


Image from <https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-features>



Event Hub Partition



Basic Programming model

Creating

```
var manager = new Microsoft.ServiceBus.NamespaceManager("mynamespace.servicebus.windows.net");  
var description = manager.CreateEventHub("MyEventHub");  
var client = EventHubClient.Create(description.Path);
```

Sending

```
var partitionedSender = client.CreatePartitionedSender();  
var partitionedSender = client.CreatePartitionedSender(description.PartitionIds[0]);
```

Receiving

```
EventHubConsumerGroup group = client.GetDefaultConsumerGroup();  
var receiver = group.CreateReceiver(client.GetRuntimeInformation().PartitionIds[0]);
```

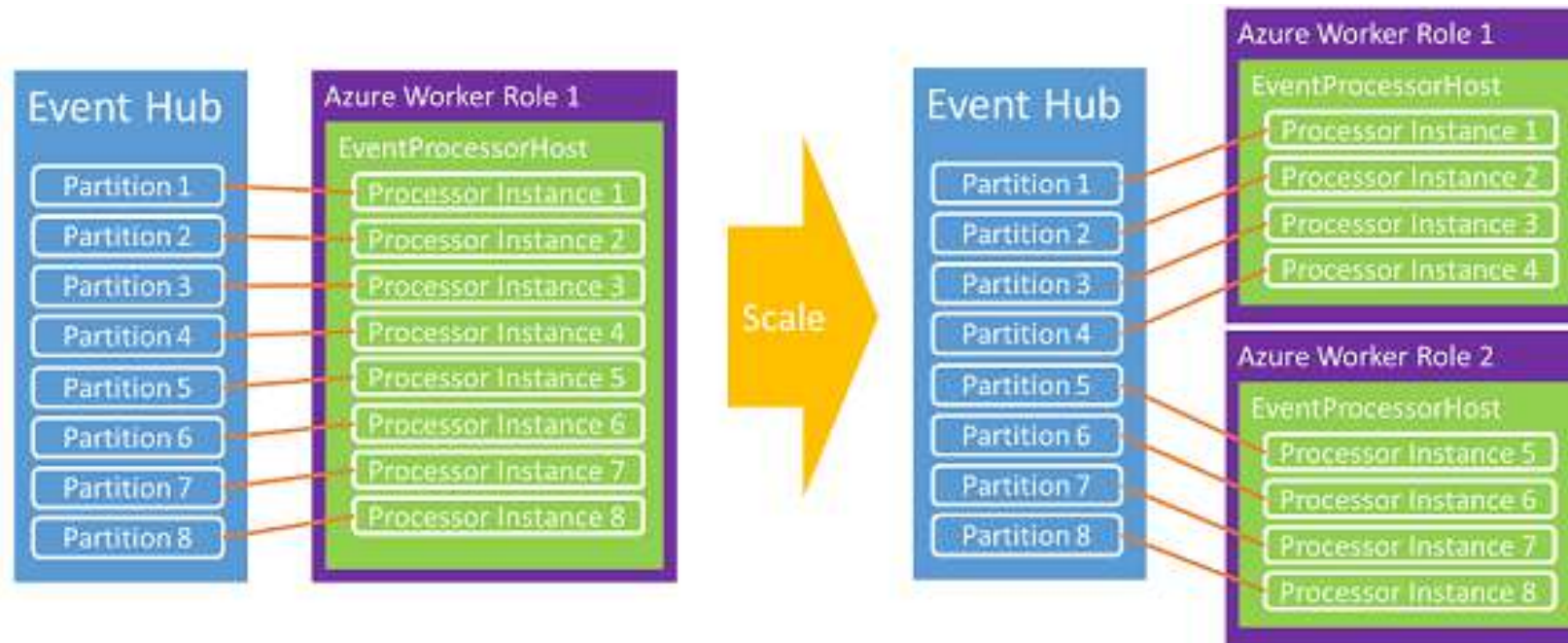



Event Processor Host

- [EventProcessorHost](#) provides a thread-safe, multi-process, safe runtime environment for event processor with checkpointing and partition lease
- Reside in Microsoft Azure Service Bus Event Hub – EventProcessorHost nuget package
- How it works:
 - Implement [IEventProcessor](#) with the logic of your message-processing:
 - Use [EventProcessorHost.RegisterEventProcessorAsync](#) to register [IEventProcessor](#)
 - The host will attempt to acquire a lease on every partition in the event hub using a "greedy" algorithm.
 - As new nodes,(worker instances), come online, they place lease reservations and over time the load shifts between nodes as each attempts to acquire more leases.



Event Processor Host



- The [EventProcessorHost](#) class also implements an Azure storage-based checkpointing mechanism.
- This mechanism stores the offset on a per partition basis, so that each consumer can determine what the last checkpoint from the previous consumer was.



Event Processor Host - IEventProcessor

```
class SimpleEventProcessor : IEventProcessor
{
    async Task IEventProcessor.CloseAsync(PartitionContext context, CloseReason reason)
    {
        Console.WriteLine("Processor Shutting Down. Partition '{context.Lease.PartitionId}'."
            if (reason == CloseReason.Shutdown){
                await context.CheckpointAsync();
            }
    }
    Task IEventProcessor.OpenAsync(PartitionContext context){
        Console.WriteLine($"SimpleEventProcessor initialized. Partition: '{context.Lease.PartitionId}'");
        return Task.CompletedTask;
    }
    async Task IEventProcessor.ProcessEventsAsync(PartitionContext context,
        IEnumerable<EventData> messages){
        foreach (EventData eventData in messages) {
            string data = Encoding.UTF8.GetString(eventData.GetBytes());
            Console.WriteLine($"Message received.
                Partition: '{context.Lease.PartitionId}', Data: '{data}'");
        }
    }
}
```




Event Processor Host – Register IEventProcessor

```
string eventProcessorHostName = Guid.NewGuid().ToString();
EventProcessorHost eventProcessorHost =
    new EventProcessorHost(eventProcessorHostName,
                          eventHubName,
                          EventHubConsumerGroup.DefaultGroupName,
                          eventHubConnectionString,
                          storageConnectionString);

Console.WriteLine("Registering EventProcessor...");
var options = new EventProcessorOptions();
options.ExceptionReceived += (sender, e) => { Console.WriteLine(e.Exception); };
eventProcessorHost.RegisterEventProcessorAsync<SimpleEventProcessor>(options).Wait();

Console.WriteLine("Receiving. Press enter key to stop worker.");
Console.ReadLine();
eventProcessorHost.UnregisterEventProcessorAsync().Wait();
```

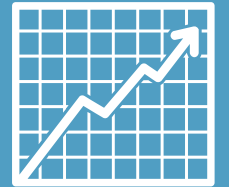

Notification Hubs

Push is Transforming Businesses

Broadcast breaking news to millions of customers using their preferences



Send notifications based on account changes or actions



Engage customer to improve your brand, customer satisfaction, and business metrics



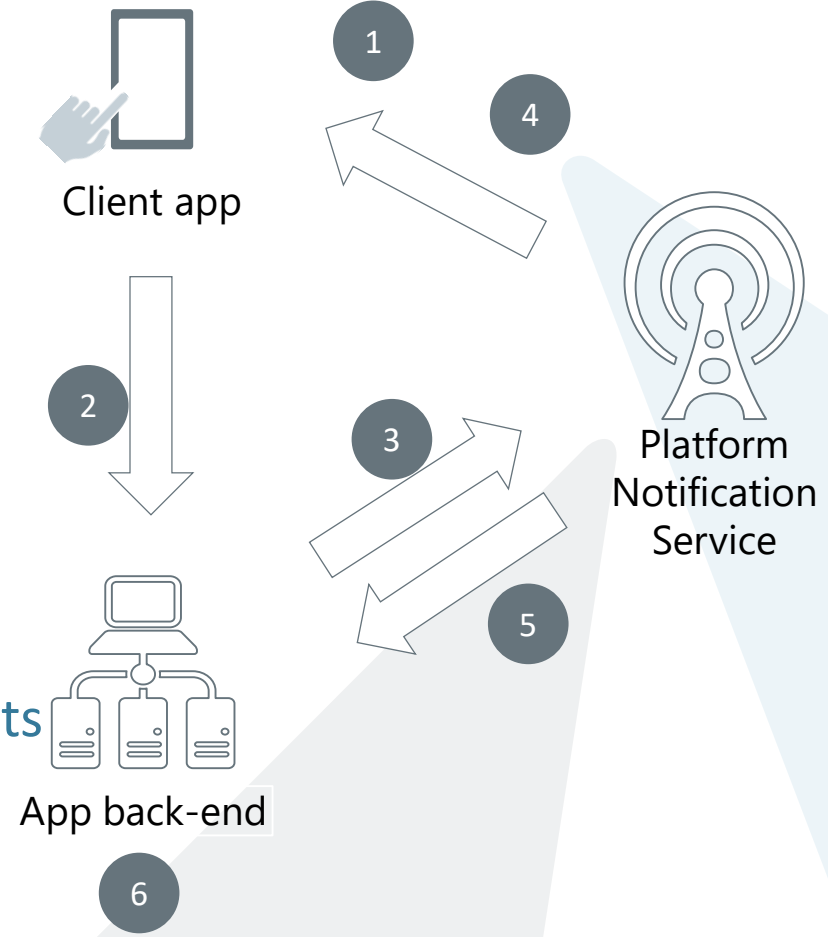
Increase employee productivity and responsiveness



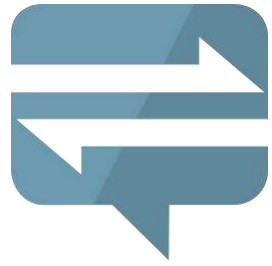


Push Notification 101

- Register device handle at app launch
 1. Client app retrieves handle from Platform Notification Service (PNS)
 2. Client app sends handle to your custom backend
- Send Notification
 3. **Your backend** connects to PNS and requests push
 - Your code** has to map between logical users and device handles
 4. PNS pushes notification to device
- Maintain backend device handles
 5. **Your code** must delete expired handles when PNS rejects them
 6. **Your code** must map between logical users and device handles



Azure Notification Hub



➤ Register device handle at app launch

1. Client app retrieves handle from Platform Notification Service
2. Client sends handle to your backend

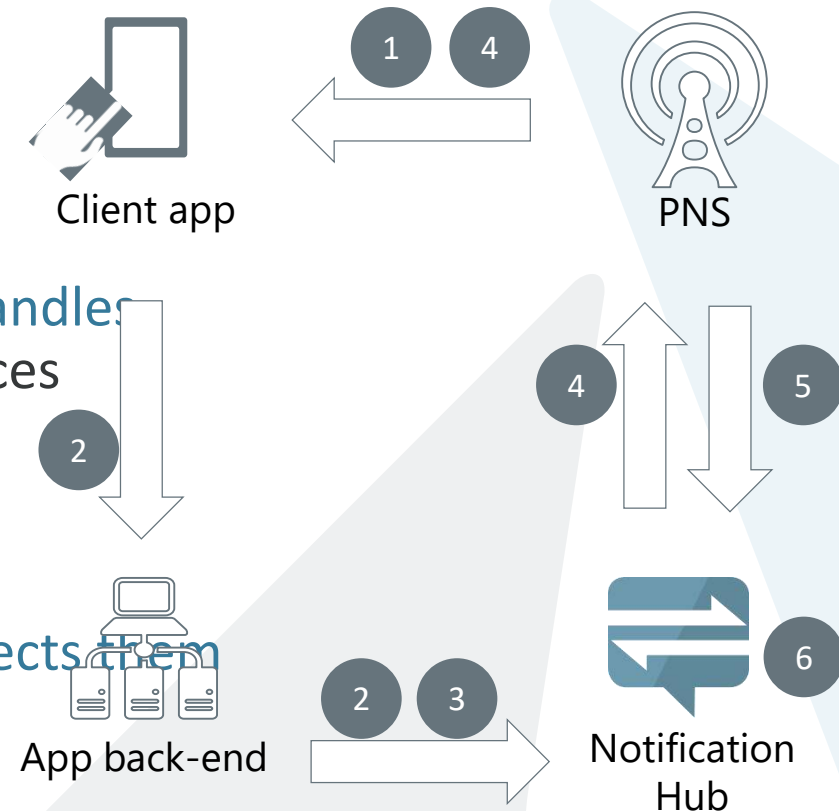
Backend **registers with Notification Hub** using tags to represent logical users and groups

➤ Send Notification

3. Backend sends request to Notification Hub using a tag
Notification Hub manages scale
Notification Hub maps logical users/groups to device handles
4. Notification Hub delivers notifications to matching devices via PNS

➤ Maintain backend device handles

5. **Notification Hub** deletes expired handles when PNS rejects them
6. **Notification Hub** maintains mapping between logical users/groups and device handles



Advantages of Notification Hub

- X-plat: one API to notify on any mobile platform
 - Backend can be on-prem or in the cloud, .NET, Java, PHP, Node, you name it
 - Support iOS, Android, Windows Phone, Windows, Kindle
- Avoid storing device information in your tables
- Work with logical users and segments
- Personalization and localization
 - Templates
- Broadcast at scale, multicast, unicast
- Rich Telemetry

Push Notification Patterns

- Notify a single logical user
- Broadcast to a segment based on interest
 - Tag Expressions
- Notify all users
- Notify all users, in batches
- Personalization and localization
 - Template Expressions
 - Geo-Targeting Tags

Event Grid



Modern computing
is all about events



Managing events is important but cumbersome

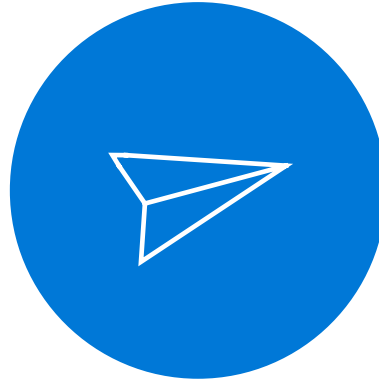
What if all these events
could be managed and
directed from one place?



Azure Event Grid



Fully-managed
event routing



Near real-time event
delivery at scale



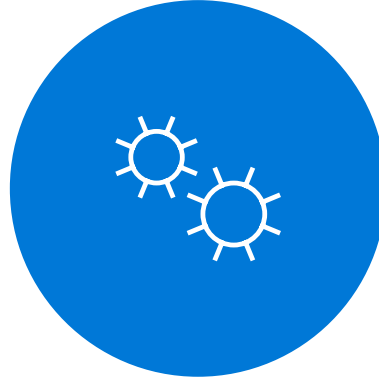
Broad coverage within
Azure and beyond

Backbone of event-driven computing

Benefits



Focus on innovation
and pay per event



Ensure reliability and
performance for your apps

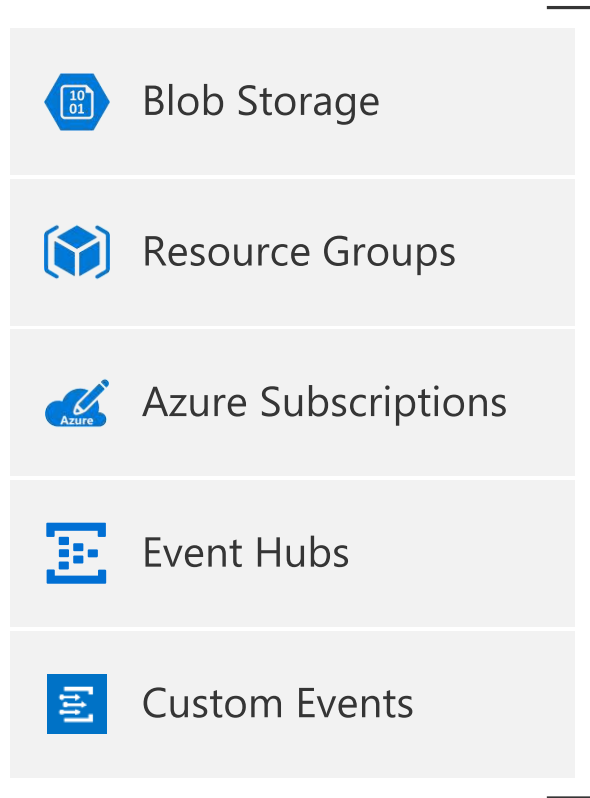


Unlock new scenarios
for your apps

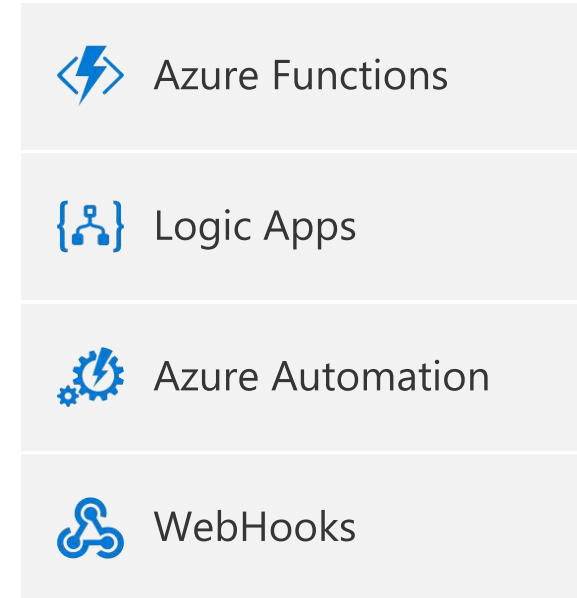
Manage all events in one place

Manage all events in one place

Event publishers

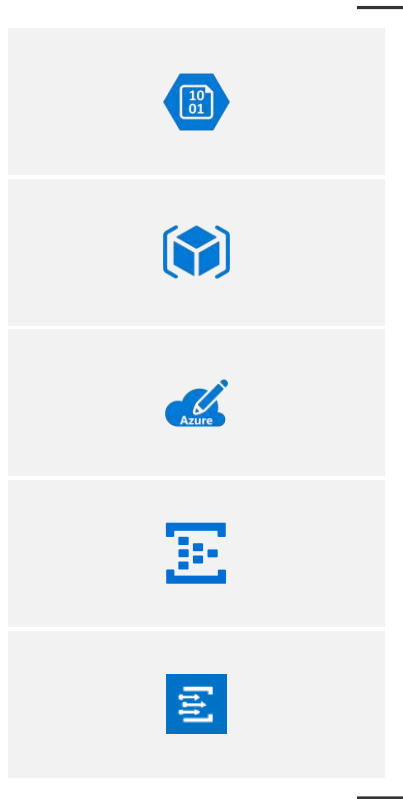


Event handlers



Manage all events in one place

Event publishers



Subscribe to pre-defined system events in Azure or create your own custom topics

Route events to any end-points, Azure or even beyond

Enable filtering and efficient routing of events



Create Event Subscription

Event Grid - PREVIEW

Name

Subscription

Resource group ☐ Use existing

Topic Type

Event Types

Subscriber Type

Prefix Filter Optional

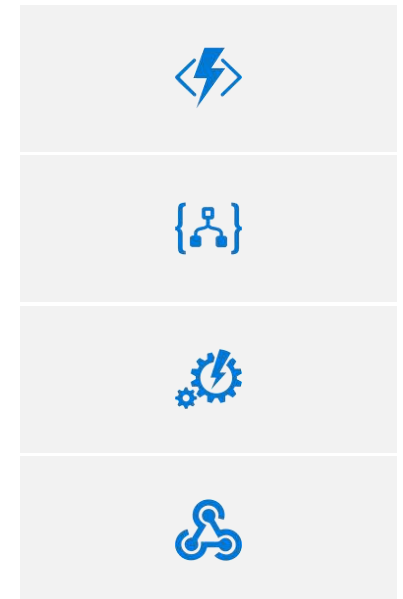
Suffix Filter Optional

☐ Filter Case Sensitive

Create



Event handlers



Ensure reliability and performance in your apps



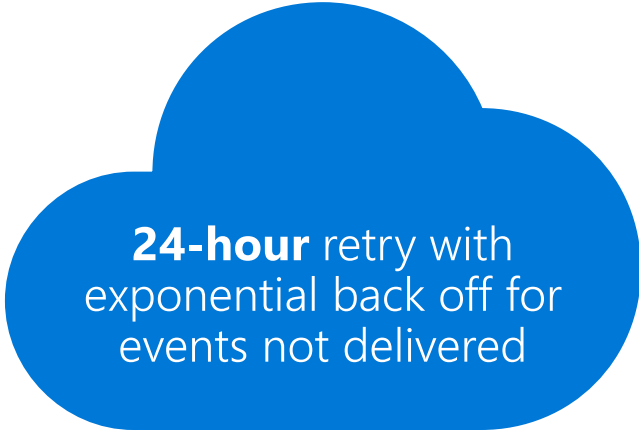
Sub-second
end-to-end latency in
the **99th percentile**

Near real-time



10,000,000 events
per second per region

Massive scale-out



24-hour retry with
exponential back off for
events not delivered

High reliability

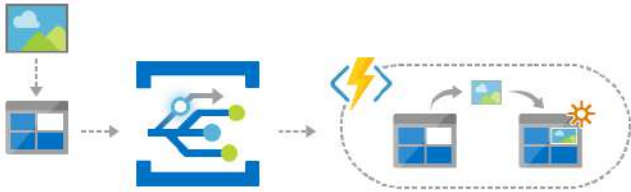
Benefit from broad coverage

Publishers	Subscribers
<p>Immediately available</p> <ul style="list-style-type: none">✓ Blob Storage✓ Resource Groups✓ Azure Subscriptions✓ Event Hubs✓ Custom Events <p>Coming soon</p> <p>Azure Automation, Azure Active Directory, API Management, Logic Apps, IoT Hub, Service Bus, Azure Data Lake Store, Cosmos DB</p>	<p>Immediately available</p> <ul style="list-style-type: none">✓ Azure Functions✓ Logic Apps✓ Azure Automation✓ WebHooks <p>Coming soon</p> <p>Fabric Controller, Service Bus, Event Hubs, Azure Data Factory, Storage Queues</p>

Scenarios

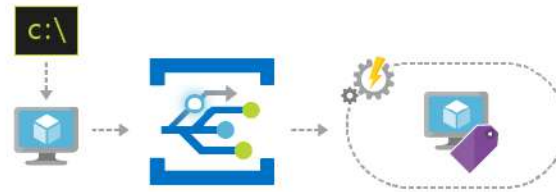
Serverless apps

Instantly trigger a serverless function to run analysis when a new file is added to a blob storage container.



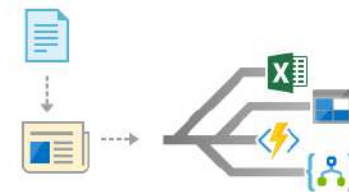
Ops automation

Speed up automation and simplify policy enforcement by notifying Azure Automation when underlying infrastructure is provisioned



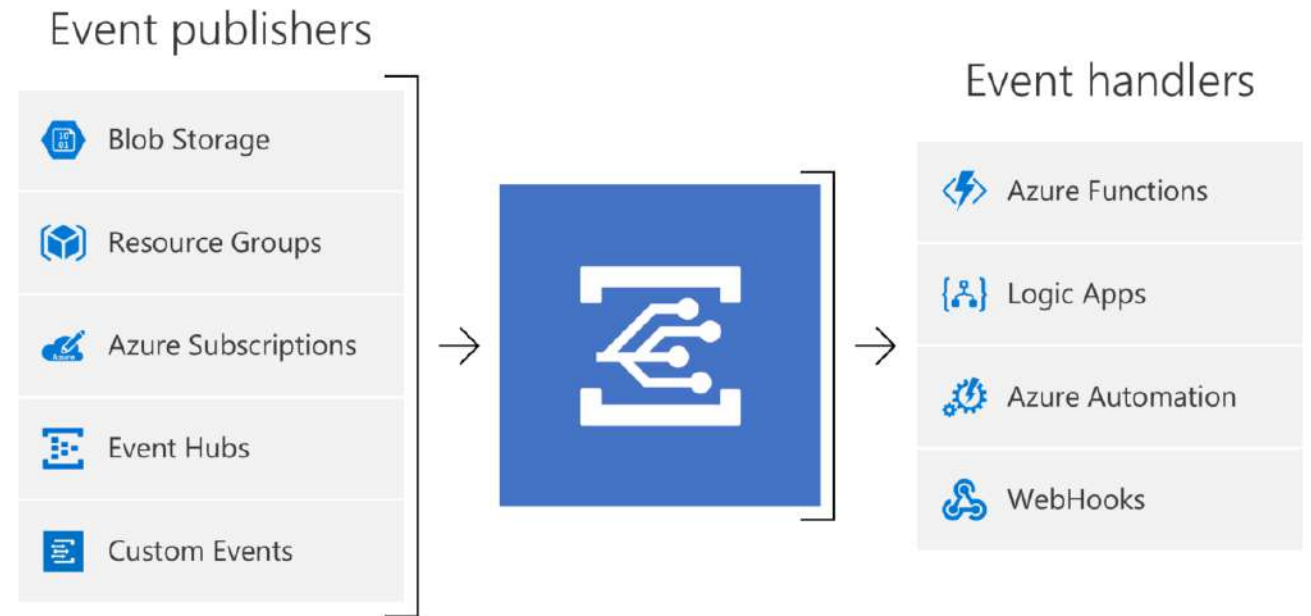
Application integration

Connects your app with other services. Create an application topic to route your app's event data to any desired destination

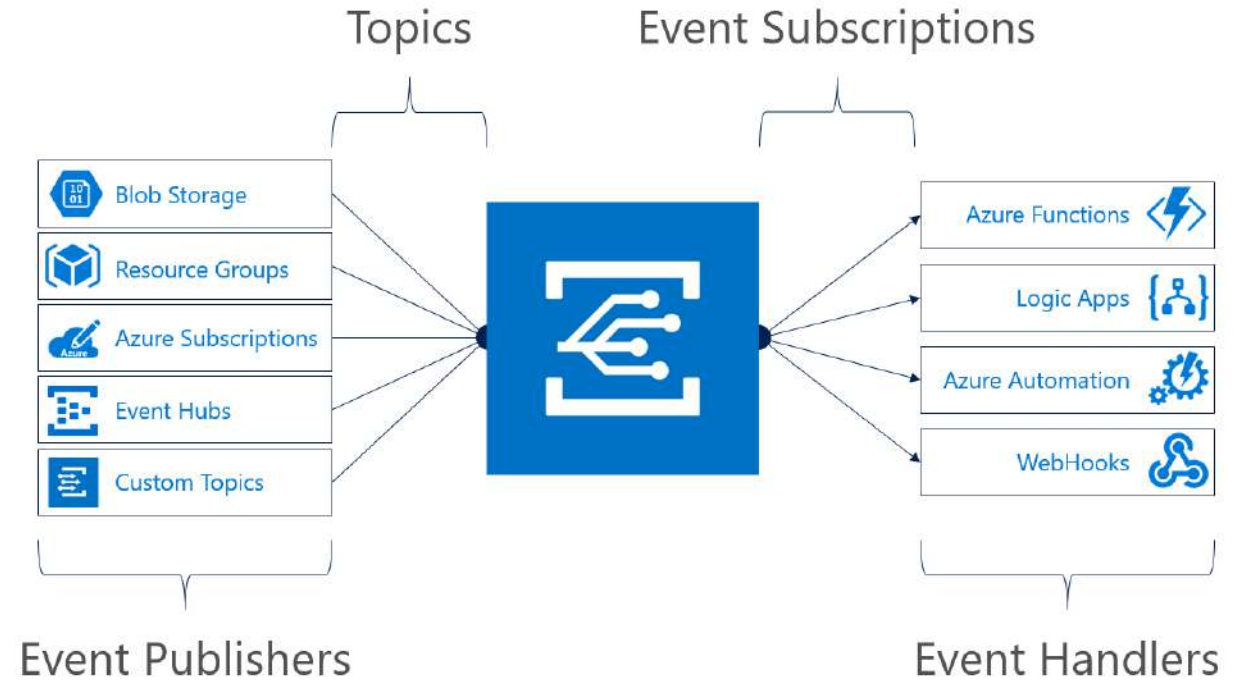


Concepts

1. Events: what happened
2. Event Publishers: where it took place
3. Topics: where publishers send events
4. Event Subscriptions: how you receive events
5. Event Handlers: the app or service reacting to the event



Concepts



1. Events: what happened
2. Event Publishers: where it took place
3. Topics: where publishers send events
4. Event Subscriptions: how you receive events
5. Event Handlers: the app or service reacting to the event

Event Grid guiding principles

- Always available
- Near real-time event delivery
- At least once delivery
- Dynamic scale
- Platform agnostic (WebHook)
- Language agnostic (HTTP protocol)

Target performance

- Sub-second end-to-end latency in the 99th percentile
- 99.99% availability
- 10,000,000 events per second per region
- 100,000,000 subscriptions per region
- 50 ms publisher latency
- 24 hour retry with exponential back off for events not delivered
- Transparent regional failover

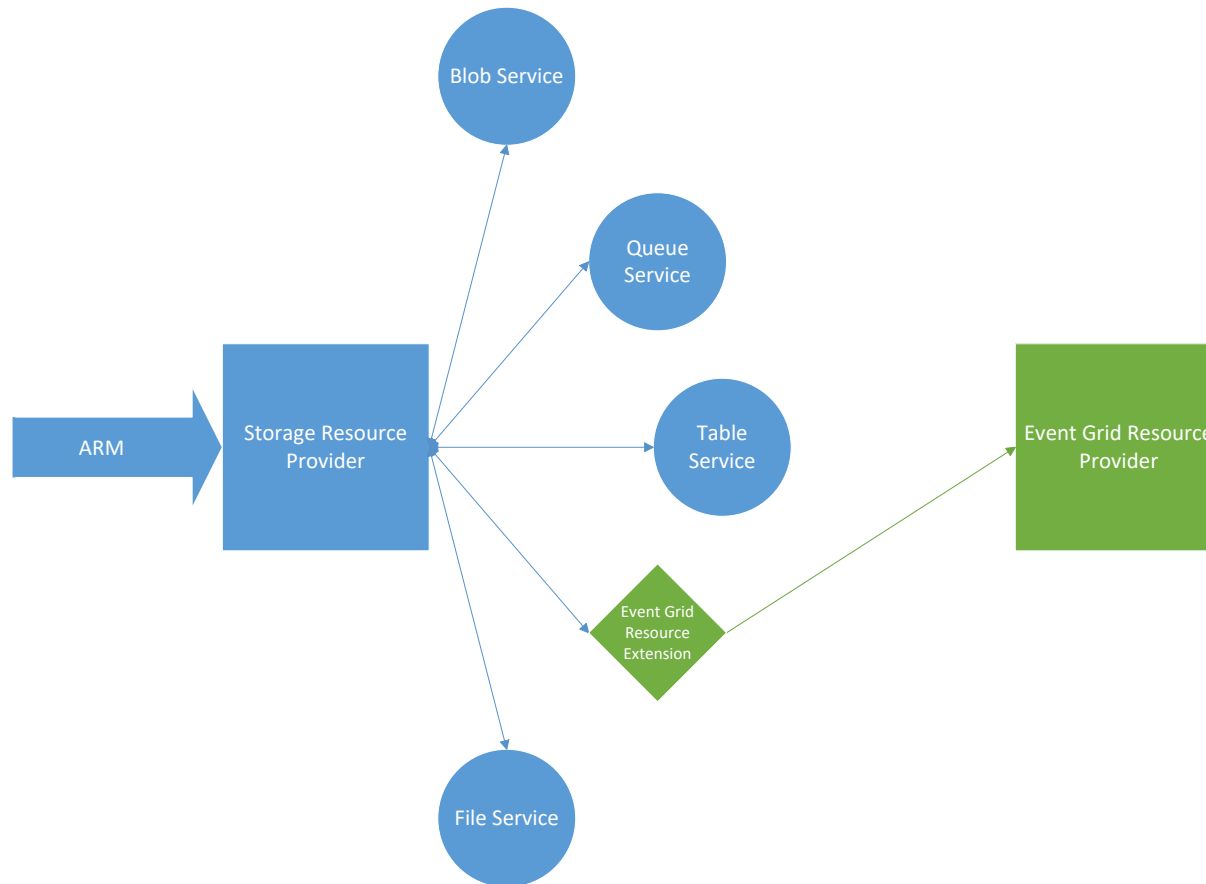
Event Schema

```
[
  {
    "topic": "/subscriptions/{subscription-id}/resourceGroups/Storage/providers/Microsoft.Storage/storageAccounts/xstoretestaccount",
    "subject": "/blobServices/default/containers/oc2d2817345i200097container/blobs/oc2d2817345i20002296blob",
    "eventType": "Microsoft.Storage.BlobCreated",
    "eventTime": "2017-06-26T18:41:00.9584103Z",
    "id": "831e1650-001e-001b-66ab-eeb76e069631",
    "data": {
      "api": "PutBlockList",
      "clientRequestId": "6d79dbfb-0e37-4fc4-981f-442c9ca65760",
      "requestId": "831e1650-001e-001b-66ab-eeb76e000000",
      "eTag": "0x8D4BCC2E4835CD0",
      "contentType": "application/octet-stream",
      "contentLength": 524288,
      "blobType": "BlockBlob",
      "url": "https://oc2d2817345i60006.blob.core.windows.net/oc2d2817345i200097container/oc2d2817345i20002296blob",
      "sequencer": "00000000000004420000000000028963",
      "storageDiagnostics": {
        "batchId": "b68529f3-68cd-4744-baa4-3c0498ec19f0"
      }
    }
  }
]
```


Resource model: extension resource

ARM calls are made to a parent resource

ARM reroutes all Event Grid calls to the Event Grid RP



Best Practices for using Custom Vision

- Use at least 30 images for each tag
- Images should be the focus of the picture
- Use sufficiently diverse images and backgrounds (ex: cats with red background and dogs with blue background)
- Train with images that are similar in {quality, resolution, lighting, etc.} to the images that will be used in prod
- Supports Microsoft accounts (MSA) and AAD

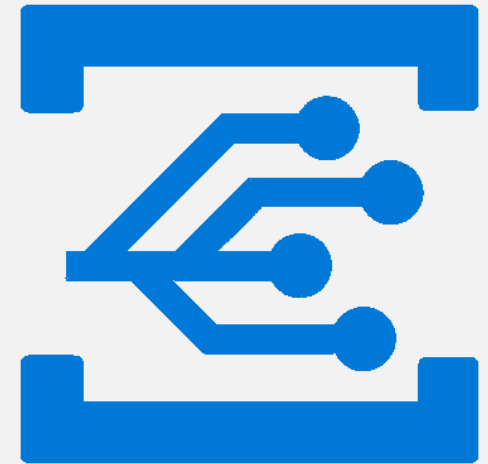
Pricing (public preview)

- ✓ \$0.30 per million operations (\$0.60 per million operations upon GA)
- ✓ Operations include:
 - Ingress events
 - Advanced matches
 - Delivery attempts
 - Management calls
- ✓ 100,000 free operations per month
- ✓ Management operations throttled to 10 per second
- ✓ 1,000 event subscriptions per account

Operation definitions, number of free operations, management throttling, and number of subscriptions per account subject to change upon GA



Learn more at
azure.com/EventGrid





Cloud Architecture

Micro Services



Agenda

- Introduction to Software Architecture
 - Requirements
 - Architecture
 - Design Principles
- Cloud Application Architecture
 - Architecture Attributes in Cloud Scale applications
 - High Availability, Management, Multi-Tenancy

Software Architecture



Software Architecture

- Architecture defines the system structure
 - Built from software components
 - The relationship between components
- Very young discipline
 - designing software architecture is still a mix of art and science!
- Usually architecture goes with high level design

➤ Requirements & Constrains

- **Non-Functional** requirements
 - Mainly during architecture phase
- **Functional** requirements
 - Mainly during design phase
- **Constraints**
 - Decision helpers

Non-Functional Requirements

It is hard to express these requirements in Use Cases or User Stories

- Express these requirements in a measurable way
 - Instead of: “The system has to be responsive”
 - Say: “The system has to respond to user action in less than 50 milliseconds”
- Non-Functional requirements have more influence on the architecture than functional requirements
 - Availability, scalability, security, ...

Architecture Attributes

➤ **Performance**

- Localize operations to minimize sub-system communication

➤ **Security & Identity**

- Use a layered architecture with critical assets in inner layers
- Provide Identity flow diagram for the whole system

➤ **Availability**

- Include redundant components in the architecture

➤ **Maintainability**

- Use fine-grain, self-contained components

➤ **Scalability**

- Include redundant components and handle state

More *-ilities

- Stability
- Backward compatibility
- Extensibility
- Reliability
- Maintainability
- Availability
- Security
- Usability
- Auditability
- Scalability
- Testability
- Composability
- Demonstrability
- Deployability
- Efficiency
- Learnability
- Manageability
- Operability

Cross Cutting Concerns

Those aspects that span across components and layers

- Identify each of the crosscutting concerns
 - Design separate components to manage these concerns
 - This approach provides better reusability and maintainability
- Avoid mixing the crosscutting code with the component code
 - You can use DI or AOP to decouple cross cutting concerns from your components
- Crosscutting Concern libraries must be deployed in such a way that all other component can use them

Cross Cutting Concerns

- Logging
- Authentication and Authorization (Identity & Security)
- Error Handling
- Communication & Hosting
 - Pub/Sub, Queues, Notifications, REST, SOAP
- Storage
- Report
- Caching
- Configuration
- Validation



The background of the slide features a photograph of two hot air balloons. One balloon, on the left, has a blue and white checkered pattern. The other, on the right, is primarily orange with some yellow and red sections. They are floating against a bright blue sky filled with soft, white, wispy clouds. The overall mood is light and airy.

Cloud Architecture



State Management

A Program is something that reads input, changes state and provides output

- The way you manage state has huge impact on your architecture and design
 - Sometime your state is transient
 - In many other cases you rely on durable storage
 - Transaction or Compensation (ACID vs BASE)
 - In-process state management vs. external state management
 - What does stateless mean?
- Bad state management can lead to poor performance, eliminate scalability or reduce correctness

The sun-light that you see indicates that the sun was still there 8 minutes ago!

- When you see in a shopping site that the product is available, is it?
- For Web/Cloud Scale application:
 - Prefer read-only data (Great for caching, CDN, Scale)
 - Understand your data flow & data consistency
 - Be prepared to compensate because of stale data

High Scale, High Availability, High Maintainability – to the Extreme!

- The Cloud technology enables cloud scale applications with much less effort
 - Answers many of the non-functional requirements and the cross-cutting concerns out of the box!
- Enables elastic, theoretically endless scale
- Surfaces more concerns:
 - More geographic options
 - Promotes “design for failure” (be prepared for VM shutdown)
 - Promotes Agility (continued integrations/deployments)



High Availability

Remove any single-point-of-failure throughout your application

- Make sure you have redundant services
- Distribute services and data across geographies
- Make sure your storage is HA
- Take storage snapshots
- Extreme monitoring & automatic responses
 - Elastic scaling
 - Self healing mechanisms
- Long latency is a failure!
 - Make sure your services respond quickly even on load
 - Spawn more instances, buy better QoS resources (I/O, CPUs, Memory, Network)
 - Load balance to reduce the pressure
- Have a storage only based failover web site for web applications



Load Balancing

- Use Auto Scaling feature to add or remove compute resources
 - [Azure Load Balancer](#) provides Layer 4 load balancing
 - LB across Fault Domains
 - Hash-based distribution
 - Port forwarding
 - [Azure Application Gateway](#) provides Layer 7 (HTTP) load balancing
 - Cookie based distribution (session affinity)
 - Configurable distribution rules
 - SSL offloading
 - Sticky Session – try to avoid it
 - [Azure Traffic Manager](#) provides DNS based Geo Load-Balancing
 - Failover
 - Round Robin
 - Performance
- Preferably use the CNAME record in the DNS and not a single server IP or Virtual IP

Queues in Cloud Applications

Load Balancing over time, and:

- A Transport
 - Asynchronous data distribution
 - Reliable delivery of messages
- Decoupled Services
 - Horizontal Scaling
 - S : R ratio
 - Set the number of senders and receivers according their workload
- A Buffer
 - Time Scaling (Buffer over time)
 - Reliable mechanism for partially offline services



Using Queues

➤ Azure [Queue Storage Service](#) / [Service Bus Queues](#)

- Since there is a size limit to a message
 - For large messages, put a URL to storage item in the message
- Some queues do not guarantee FIFO
 - Build your app to handle it (Add counter to the message)
- [Idempotency](#) is important (Handle duplicate message gracefully)
 - Azure Service Bus Queues can guarantee reliable messaging and duplicate detection
- Understand the access control of the queue
- Handle polling rate & batch operations wisely
- Set visibility timeout to handle message concurrency control
- Set the retention period (when the message will be deleted)
- Handle poison messages

Service Oriented Architecture

- An architectural pattern in which application components provide services to other components via a communications protocol
- A service is a self-contained unit of functionality
- Services can be combined to provide the functionality of a large software application
- SOA makes it easier for software components on computers connected over a network to cooperate.
- Every compute resource can run any number of services, and each service is built in a way that ensures that the service can exchange information with any other service in the network without human interaction and without the need to make changes to the underlying program itself



The 24/7 Challenge

- How do you update a system running 24/7/365?
 - How do you keep the application servers responsive?
 - How do you keep all application servers synced?
 - How do you update the data/schema?
 - How do you update all your clients' software?
 - Web, Mobile, Desktop...
 - How do you rollback on error?
 - How do you rollback data?
 - How do you know there is an error?



Microservices Architecture (MSA) - Wikipedia

- *“Microservices is a specialization of and implementation approach for service-oriented architectures (SOA) used to build flexible, independently deployable software systems”*
- *“Services are **small in size, messaging enabled, bounded by contexts, autonomously developed, independently deployable, decentralized** and built and released with **automated processes**”*
- *“The benefit of distributing different responsibilities of the system into different smaller services is that it **enhances the cohesion and decreases the coupling**”*

➤ Server Side Patterns – Micro Services

“Microservice applications are composed of small, independently versioned, and scalable customer-focused services that communicate with each other over standard protocols with well-defined interfaces.”

- Extreme SOA
 - A class is a service
 - Each service encapsulates a simple business functionality
 - Volatile boundary
 - Versioning is backed in to the method
- Requires hosting and management system
 - Azure Service Fabric for example
 - Docker or other Container technology

Micro Service Architecture

➤ Some principles:

- A micro service should be less than 100 lines of code
 - It should be easy to understand, fast to deploy, and cheap to reimplement (throw the old one)
- A micro service should be independently developed & deployed
- A micro service should have private data ownership
- Eventual Consistency
- Versioning



Microservices & SOA

➤ Microservices

- Small, independent processes that communicate with each other to form complex applications which utilize language-agnostic APIs
 - Small building blocks
 - Highly decoupled
 - Focused on doing a small task
- Facilitate a modular approach to system-building.
- The microservices architectural style is becoming the standard for building continuously deployed systems
- Advantages
 - Services are easy to replace
 - Services are organized around capabilities, e.g., user interface front-end, recommendation, logistics, billing, etc.
 - Services can be implemented using different programming languages, databases, hardware and software environment, depending on what fits best

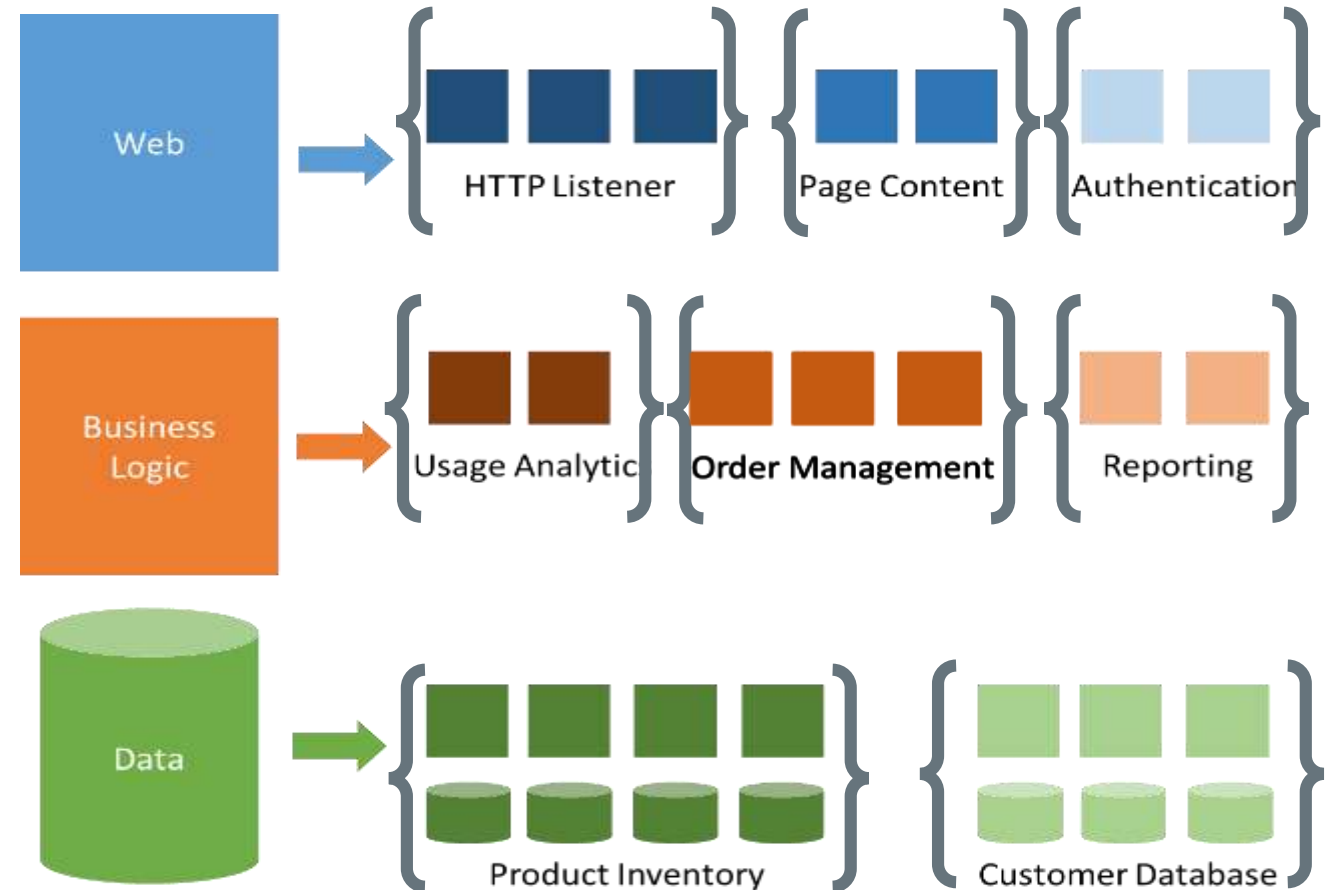


Microservices & SOA

- SOA aims at integrating various (business) applications
- Microservices belong to a single application
 - Naturally enforces a modular structure
 - Architectures are usually symmetrical rather than hierarchical
- Microservices sometimes referred to as a finer-grained SOA
 - Or SOA as SOA should have been
- How small is “micro”?
 - It depends!

Modernization with Microservices

- Individually built and deployed
- Small, independently executing services
- Integrate using published API calls for overall application's functionality
- Fine-grained, loosely coupled application



MSA - Signs that you do something wrong

- If you find the need to deploy services together, you're doing something wrong
- If you have a single codebase for all your services, you're doing something wrong
- If you find you have a service talking directly to the database of another service, you're doing something very wrong
- If you have to send out a warning before each deploy of a service, you're doing something wrong



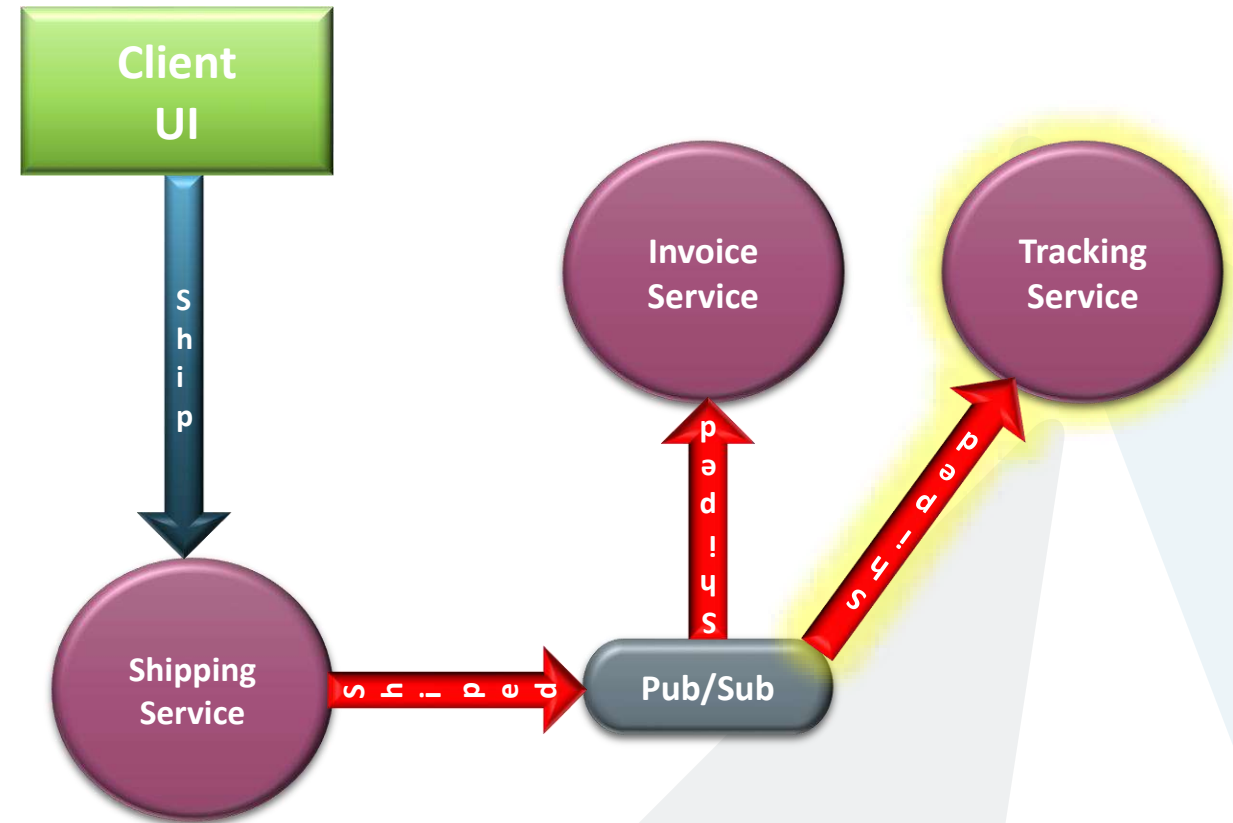
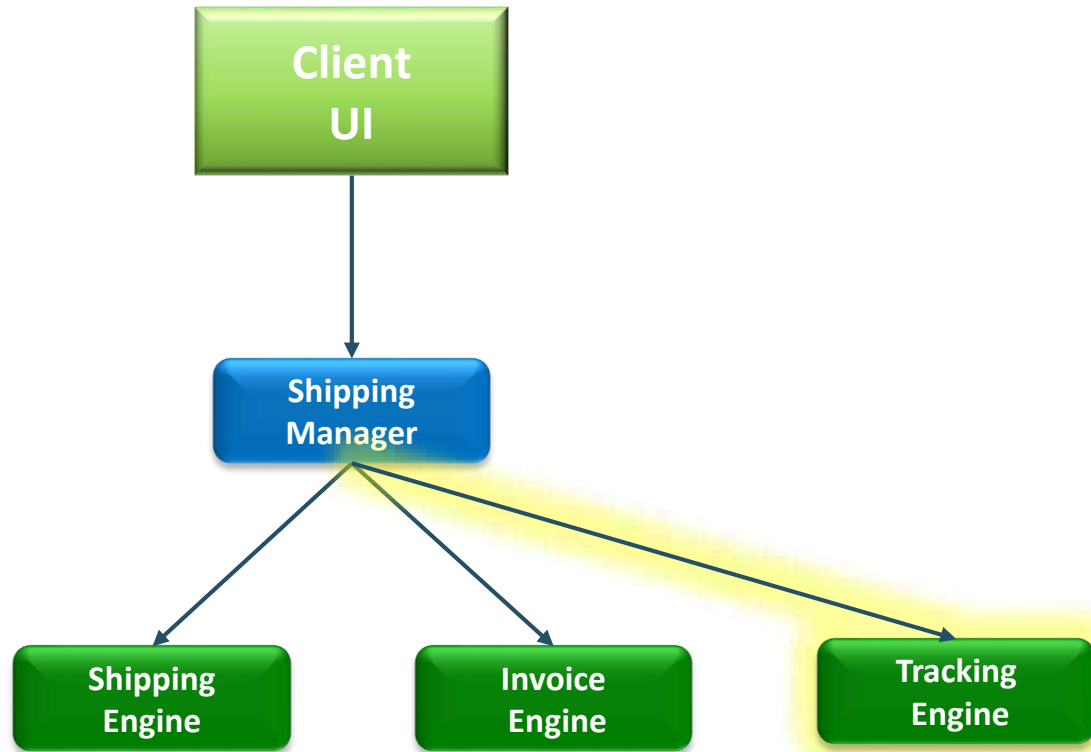
Microservices

- Microservices is traditionally hard
 - Deploy & manage multiple (high) amount of distinct components
 - Ensure scalability & HA of each component separately
 - Ensure sufficient compute resources for each component separately
 - Distribution of components across a server farm
- Today we have tools to manage these concerns for us
 - Or at least help us manage them correctly
- Two popular options on Azure
 - [Azure Container Service](#) (Docker, [Managed Kubernetes](#))
 - [Azure Service Fabric](#)

Decoupling Business Workflows

- Orchestration vs Choreography
- The Manager service in IDesign method encapsulates the business process
 - A major change in the business-flow requires a change in the manager
- In a non-restrictive MSA, any service can consume any message and call any other service – this is the common approach!
 - Either by using an API, or by sending a message to a broker
 - Event Driven Messaging, Pub/Sub, Queues, etc.
 - With this approach one can change the business flow just by introducing a new micro service!!!
 - However, the complexity of the system is much higher!!!
- **Does MSA flexibility worth the complexity?**
- Server-less application is another form of MSA, and introduces even higher complexity

Orchestration vs Choreography

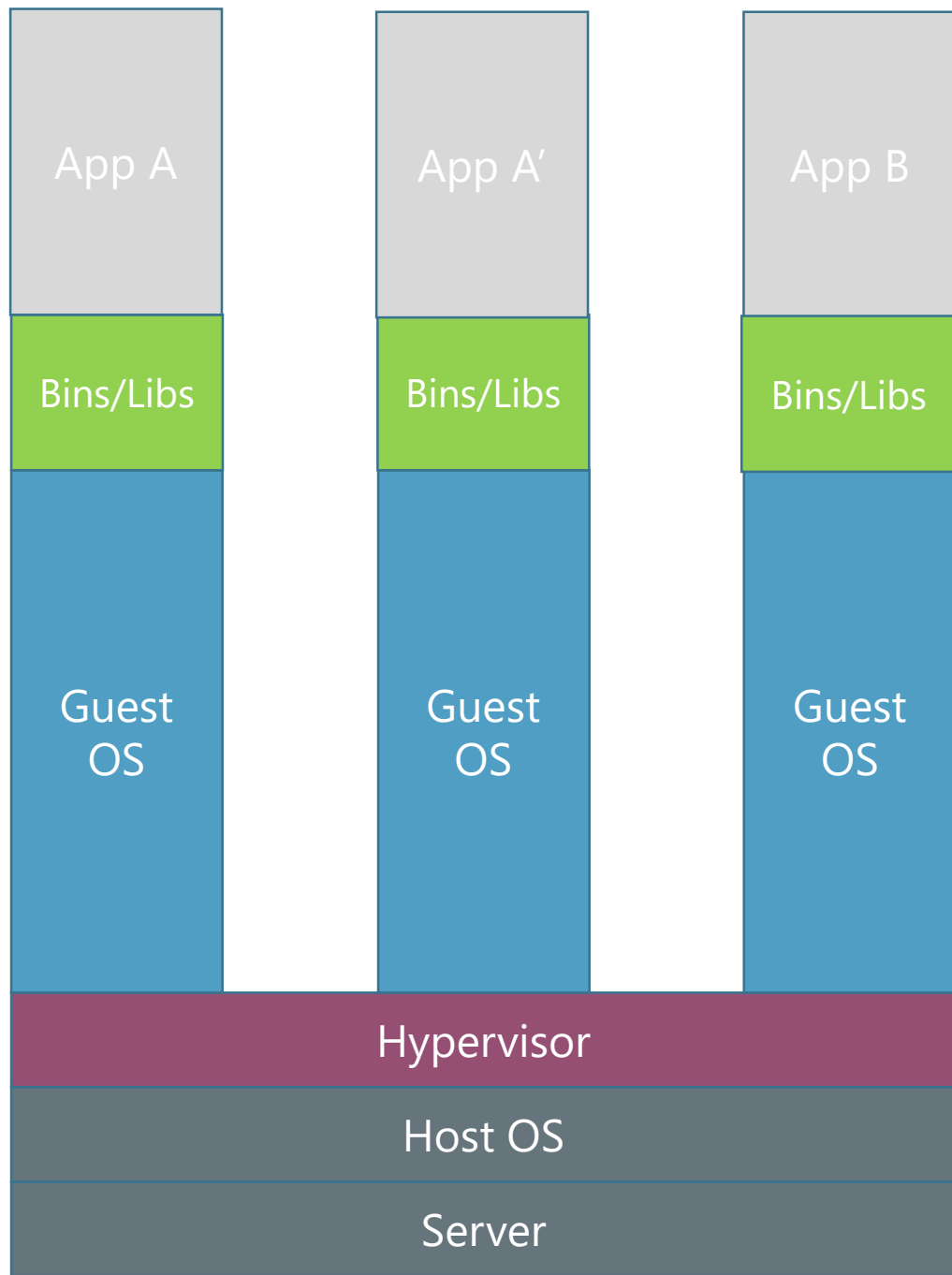


The Seek for the Perfect Host

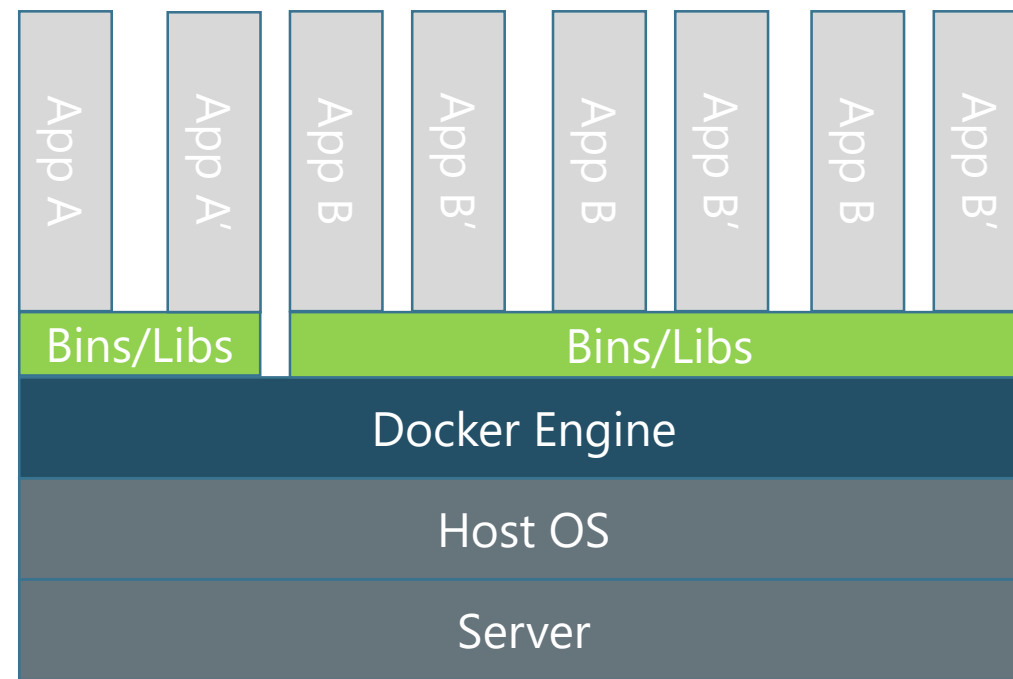


- To deal with MSA complexity the requirements are:
 - Isolation, Security
 - Maintainability, Manageability, Traceability
 - Fast Auto Scaling, Health monitoring, Self Healing
- Famous Hosts:
 - Process
 - (Console) Application
 - Windows Service
 - Unix/Linux Demon
 - Web Servers
 - IIS, Apache, NGINEX
 - Framework and language runtimes
 - .NET AppDomain, JVM
 - Virtual Machines
 - Hyper-V, VMWare, VirtualBox
 - Containers
 - Docker Containers, Windows Containers, Windows Hyper-V Containers, Service Fabric
- Containers are the current MSA favorite hosting model





Containers are isolated,
but share OS and, where
appropriate,
bins/libraries





Microservices

- Use Azure Container Service
 - Existing Docker enabled systems
 - Usage of existing 3rd party tools which integrate with Docker
 - Want to customize various elements of the framework
 - Registry service
 - Image repository
 - Cluster orchestration
- Use Azure Service Fabric
 - Prescribed holistic solution
 - New software taking advantage of the service fabric SDK
 - Stateful Services
 - Actor Model
 - Easier to use and get started
- Integration is possible!
 - Run Docker containers in Service Fabric
 - Run Service Fabric in a Docker container

Multitenancy

One Application to Rule Them All!

- The cloud provides many benefits
 - You develop and deploy to the cloud
 - Staging/Production cloud environments
 - Cloud management & monitoring
- You'd like to share those effort for all of your customers
- A Multi-tenant application is one application that served many isolated application instances





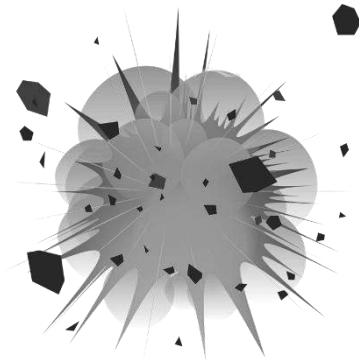
Multitenancy

- Consider which resources are shared and which are separate
 - Beware of privacy!
- Multi-factor optimization question
 - Cost of resources
 - Maintenance difficulty
 - Failures
 - Updates
 - Security constraints
 - The more separation, the more security...
 - Strive to avoid potential code issues which can retrieve wrong data

Azure IoT



Traffic Light History Facts



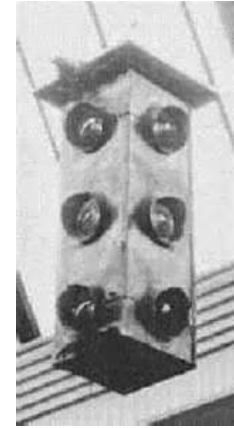
1868
London



1912
SLC



1914
Cleveland



1920
Detroit



1922
Automatic timer
controlled



1950s
Computer
Detection

*light*TRAFFIC

Agenda

- Introduction
- The Simple System
- IoT Device Lifecycle
- The Modern IoT System
- Azure IoT PaaS & SaaS
- Azure IoT Hub
- Device Registry & Provisioning Service
- Twin, Routing and Jobs
- Smart cloud & intelligence edge
- Summary



Technology advances in “Buzzwords” steps

- It starts with the basic technology and slowly evolves
- After several years the world is ready to embrace the technology
- And here is where the big buzz begins
- For the Internet of Things:
 - The basic technology is already here for almost a decade by now
 - Amazon Web Services cloud has started in 2006
 - Azure was announced in October 2008 and released on February 1, 2010
 - Devices such as the [Amtel AVR](#) are more than 20 years old
 - The [Arduino](#) family that many IoT hobbyists are using it as a cheap IoT end device is 10 years old

IoT - More Than The Core Technology

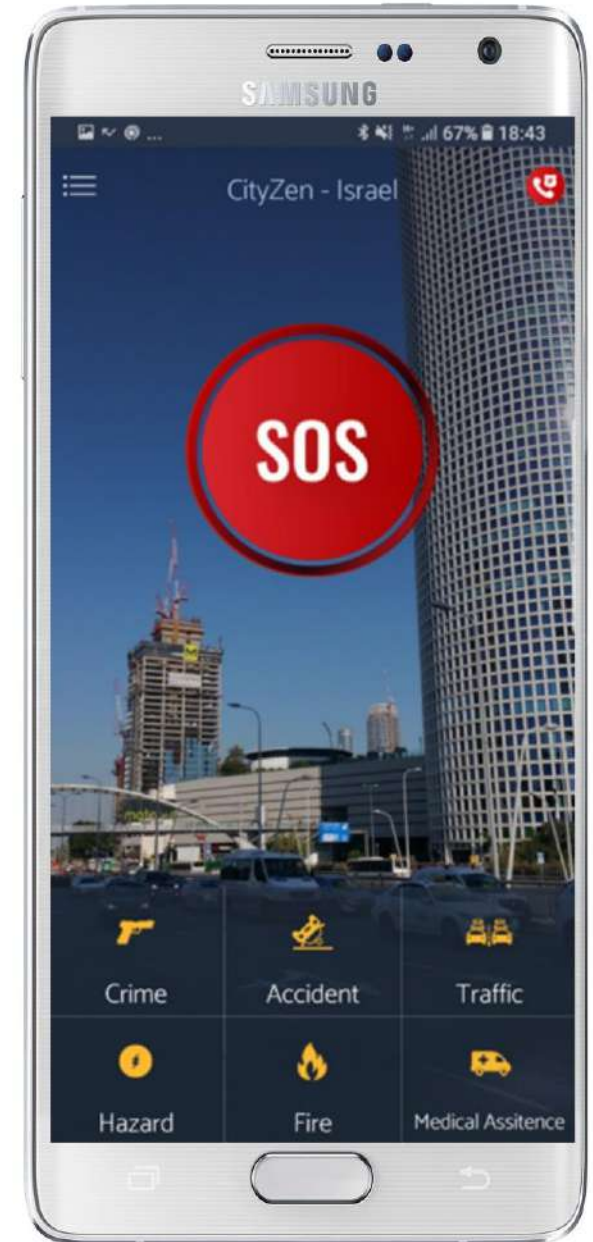
- It is not (just) the technology that makes IoT what it is
- It is the
 - Concepts, perception, commitment and the challenges
 - Facts that the entire industry is dealing with it nowadays
- The IoT Challenge:
 - ***vast** amount of **devices** using different hardware and software technologies, are **connected** between them and to the **cloud** which in turn provides many **services**, which handle a **huge stream of data** and **analyze** it and extract vital **information** about the current state of the system and via extended processing it can even **predict** future state*



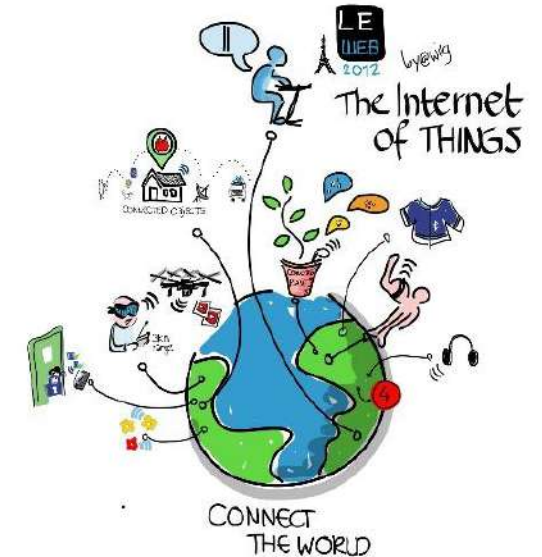
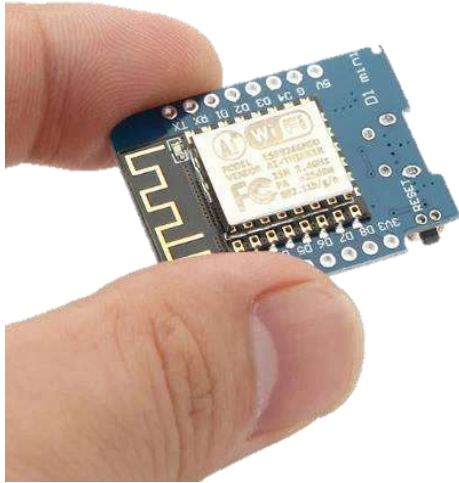
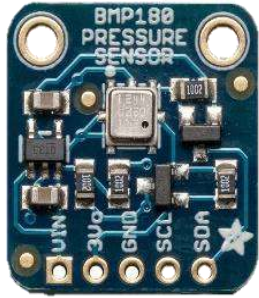
The Citizen App

- Make any citizen a sensor(s)
 - Text, picture, video, audio, location
- Large city scale – 10s Millions
- Server-less Architecture:
 - IoT Hub per city
 - Stream Analytics
 - CosmosDB
 - Azure Functions
 - File uploads
- Small team: 2-3 developers, 6-8 months

Amazing!



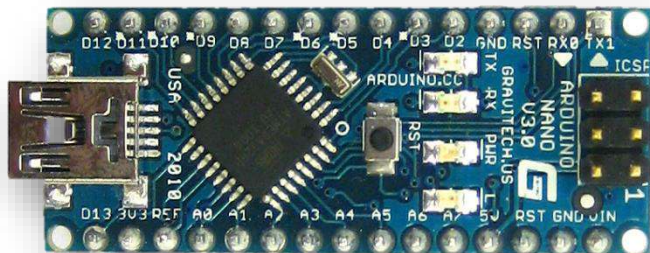
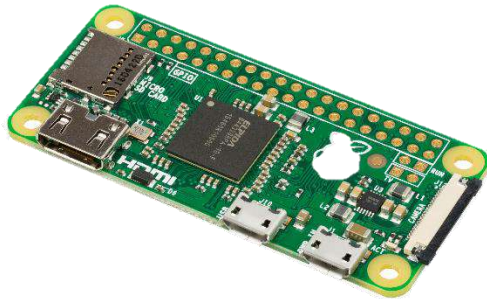
IoT System Basic Components



```
var sensorData = await _bmp180.GetSensorDataAsync(Bmp180.UltraHighResolution);  
var messageString = JsonConvert.SerializeObject(sensorData);  
var message = new  
    Microsoft.Azure.Devices.Client.Message(Encoding.ASCII.GetBytes(messageString));  
await deviceClient.SendEventAsync(message);
```


The Device

- There are many System on a Chip (SoC) devices to choose from
- Raspberry Pi family
- Arduino Compatible Family
 - ESP 8266 based devices
- Intel devices
- ...



Raspberry Pi Kit
Windows 10 and Raspbian
Samples in C and C#



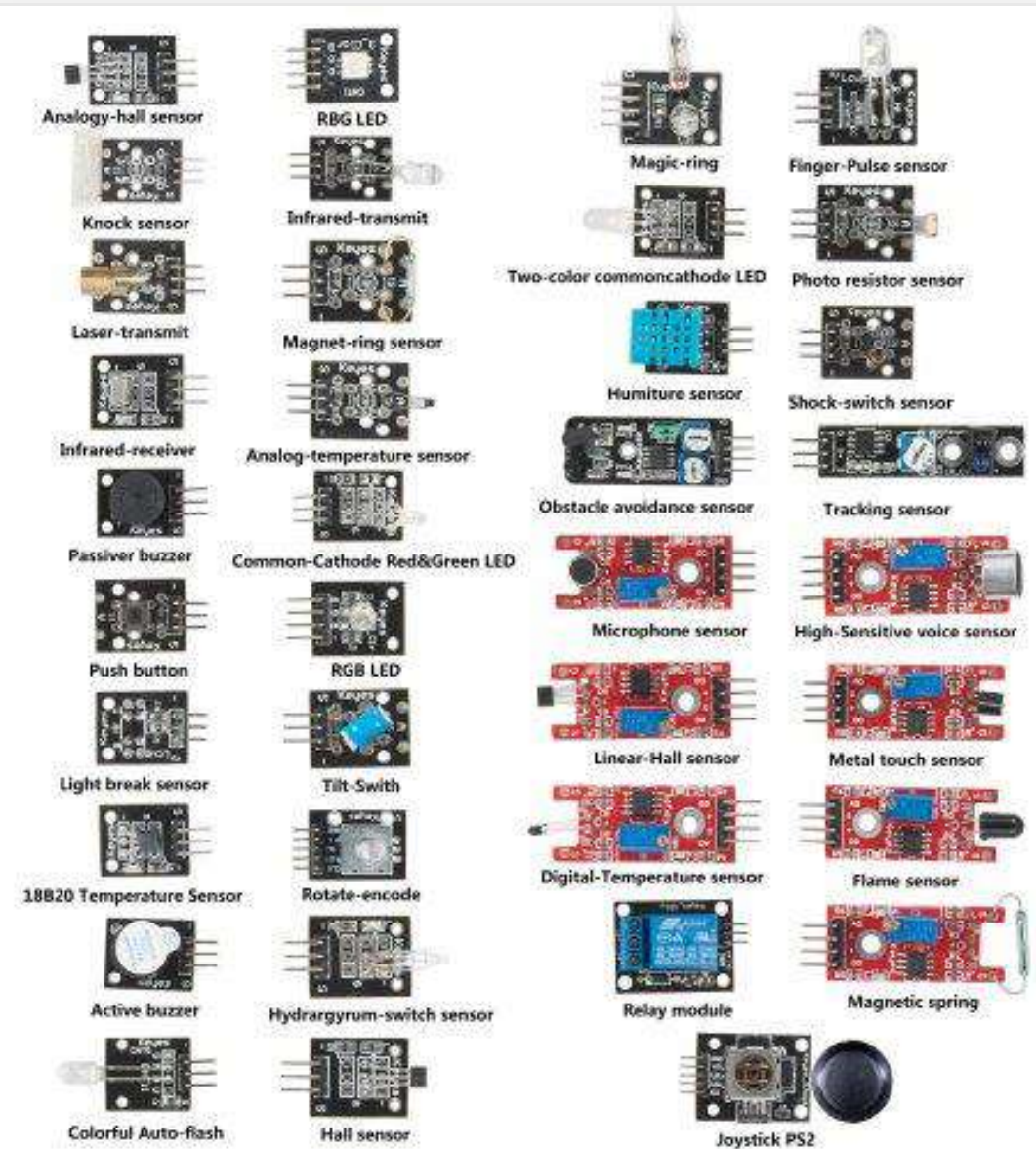
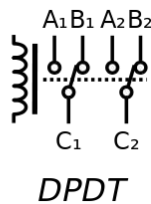
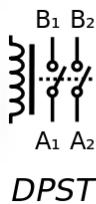
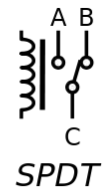
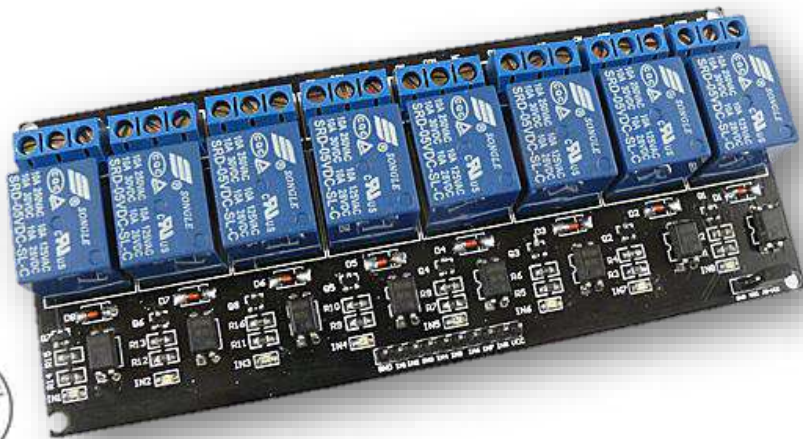
Intel Edison Kit
Linux Yocto
Samples in JavaScript (Node.js)



Feather Huzzah ESP8266 Kit
RTOS
Samples in Arduino IDE and C

Sensors, Actuators, Motors

► There are plenty of them





How do I play with it?

- Pick your weapon
 - A prototype board that has networking capabilities
 - [Raspberry Pi](#), [WeMos](#), [Intel IoT](#), [Tessel](#), [NetDuino](#)
 - The complete list @ [Azure IoT hardware catalog](#)
- Some Electronics & Hardware Programming
 - Understand how to connect sensors and communicate with them
- Pick your Cloud Services and technologies
 - Microsoft Azure, AWS
 - Do something with the (Big) data



Electronics 101

► Ohm's law: $V = R * I$

► $V \rightarrow$ Volt (V, mV)

► $I \rightarrow$ Ampere (A, mA)

► $R \rightarrow$ Ohm (Ω)


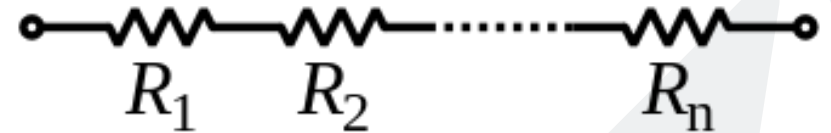


Diagram of a resistor with four color bands: brown, black, red, and gold. The bands are connected to a table below.

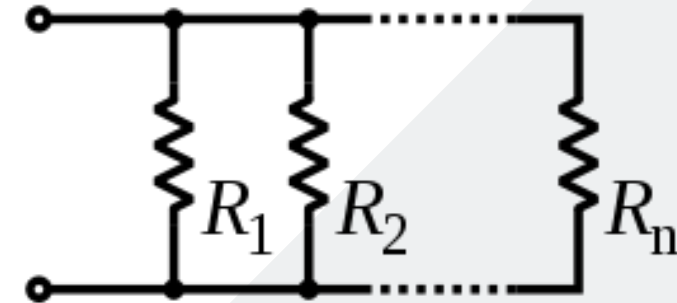
1st digit	2nd digit	Multiplier	Tolerance
0	0	x 1	
1	1	x 10	±1%
2	2	x 100	±2%
3	3	x 1K	
4	4	x 10K	
5	5	x 100K	
6	6	x 1M	
7	7		
8	8	x 0.1	±5%
9	9	x 0.01	±10%

► Series and Parallel Resistors

► Series: $R_{EQ} = R_1 + R_2 + \dots + R_n$



► Parallel: $1/R_{EQ} = 1/R_1 + 1/R_2 + \dots + 1/R_n$



➤ Resistive divider, LED current limiter

➤ Voltage divider: $V_{OUT} = V_{IN} * (R_2 / (R_1 + R_2))$

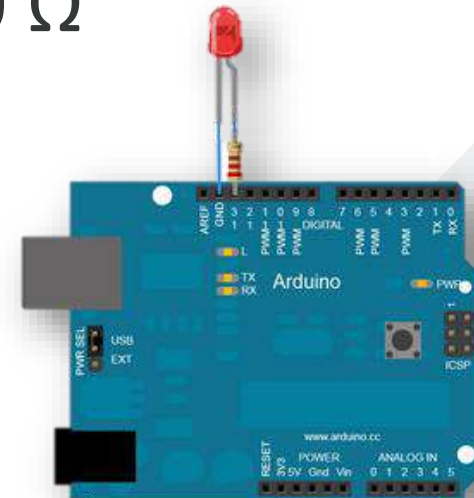
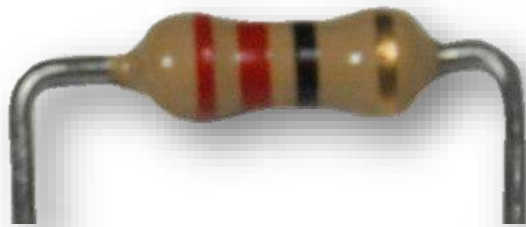
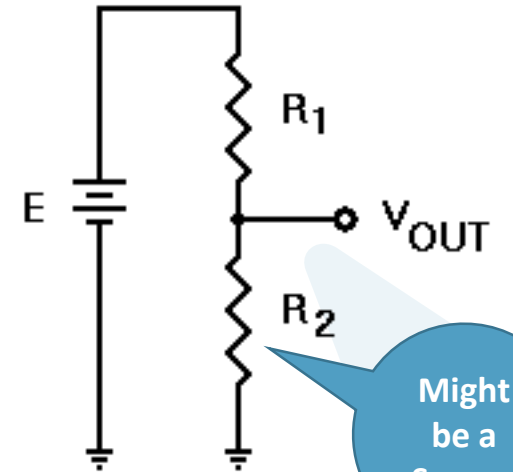
➤ 3.3V supply, $R_1 = 5k\Omega$, $R_2 = 10k\Omega$

$$V_{OUT} = 3.3V * (10k\Omega / (10k\Omega + 5k\Omega)) = 2.2V$$

➤ LED resistor: $R = (V_{SUPPLY} - V_{LED}) / I_{LED}$

➤ 5V supply, 0.7V 20mA LED: $R = (5V - 0.7V) / 20mA = 215\Omega$

➔ Nearest higher rated resistor 220 Ω



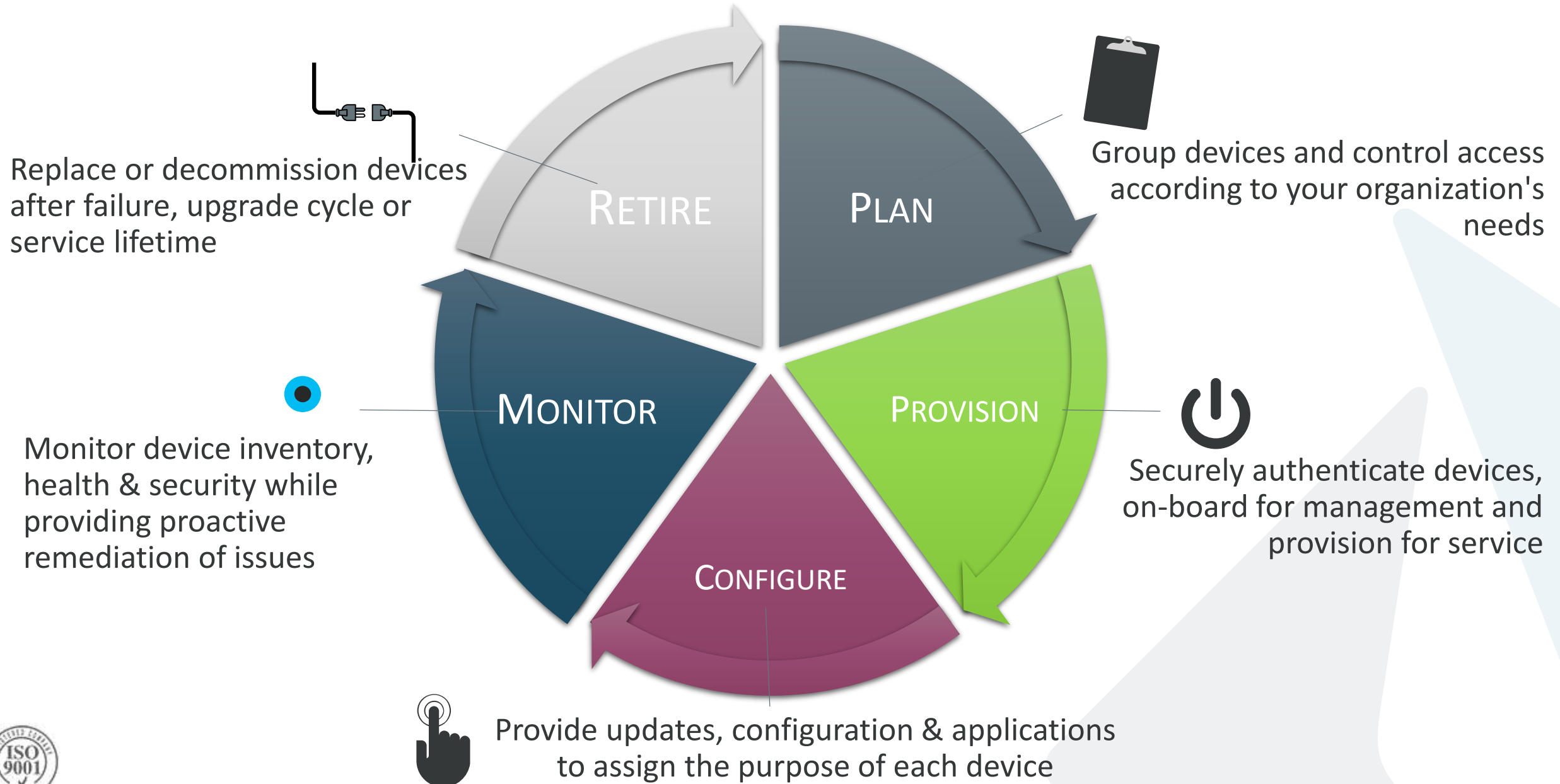
The Modern IoT System

- Most large IoT systems include one or more of the following:
 - Many different end **devices** with **sensors** and **actuators**
 - Local **gateways**
 - A collection of **cloud services** that enables:
 - **Registration** of end devices
 - **Management** of end devices
 - **Controlling** of end devices
 - Different **communication protocols** that provide reliability and security
 - The ability to **collect a vast amount** of data in a very **high rate**
 - The ability to **analyze** the **stream** of information in **close to real-time** manner
 - The ability to **analyze** the **current** and **historical** collected information
 - The ability to **show** the resulted **conclusion** and the **collected data**

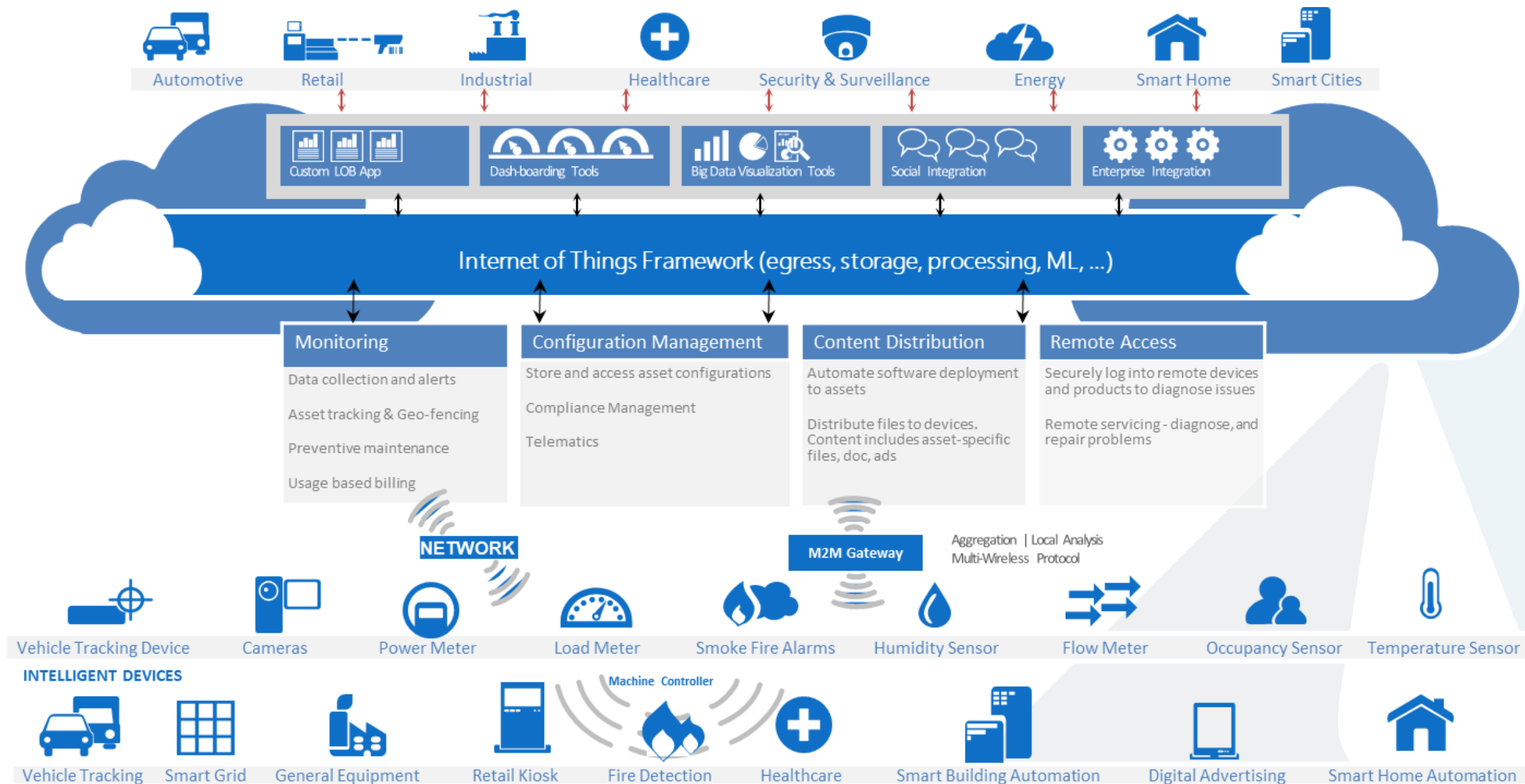
➤ The IoT Challenge - Pets Vs Cattle – Pettle?



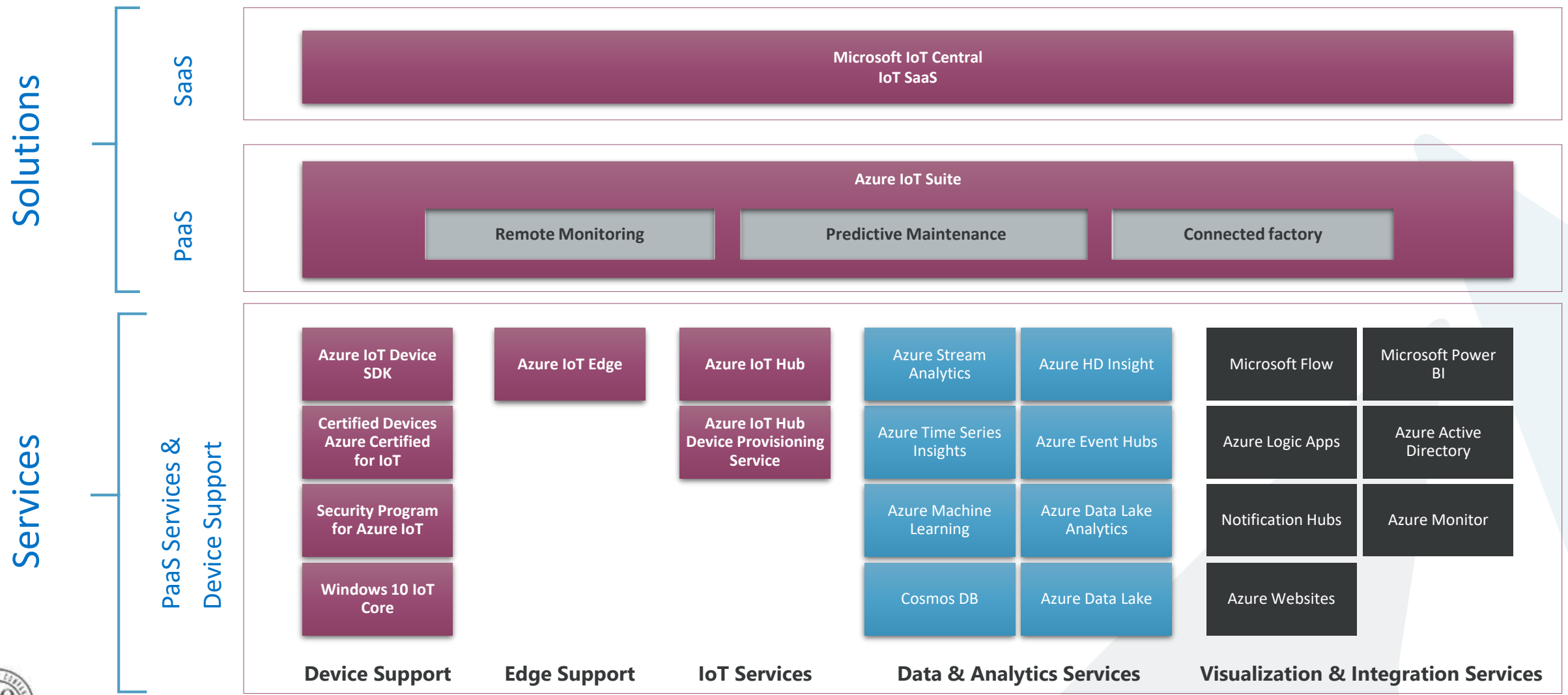
IoT Device Lifecycle



High Level Architecture



Comprehensive set of capabilities for IoT solutions



Azure IoT solutions approach

➤ SaaS – Microsoft IoT Central

- Fully managed IoT SaaS
- No cloud solution development expertise required
- Configurable to your needs
- Ideal for straightforward IoT needs

➤ PaaS – Azure IoT Suite

- Preconfigured solutions
- Deploy in minutes
- Accelerate time to value
- Ideal for solutions that require ultimate control



IoT Central Features



Connectivity Hub & Telemetry ingestion

Connects a variety of devices to the cloud through an open platform

Device management

Enables understanding, control, and optimization of investments

Analytics & dashboards

Provide simple and consumable reports and visualizations for any platform

Rules engine

Real time data processing

Time-series insights

Identify trends among millions of IoT events

Digital twin management

Enables actionable insights through modeling and simulation

User and identity management

Delivers customized levels of permissions across users and protect from unauthorized access

Create Application

We just need a few things from you, so we can create your application

Application Name * ⓘ
Contoso Vending

URL * ⓘ
contoso-vending.microsoftiotcentral-ppe.com

Directory * ⓘ
Microsoft (microsoft.onmicrosoft.com) ▾

Azure Subscription * ⓘ
IOTC_CLIENT_WHITELIST_PRODUCTION ▾
Don't have a subscription? [Create subscription](#)

Resource Group * ⓘ
contoso-vending ✕ ▾

Region * ⓘ
East US ▾

Application Template

☒ Custom Application
Start with a blank template and define your application from scratch.

☐ Sample Contoso
Get started with a predefined application for a connected device.

☐ Sample Devkits
Want to connect a Raspberry Pi or MXChip IoT DevKit? Start with this predefined app and get them connected in minutes.

Payment plan

- ☐ Free 30 Day Trial Application
- ☒ Paid Application
500 USD per application per month (includes 100 devices)
0.50 USD per additional device per month after that
30 USD per additional GB of data

By clicking create, you agree to the Microsoft IoT Central [Terms of use](#) and [Privacy Statement](#).

Create

Refrigerated Vending Machine (1.0.0)

Measurements Settings Properties Rules Dashboard

+ New Rule

Save X Cancel

Name

Temperature Monitor

Enable rule for all devices of this template ⓘ

☒ On

Conditions

- Temperature is greater than 15

Actions

Select Action



Email

Coming Soon



Webhook

Invoke a webhook to trigger external custom workflows.

Coming Soon



SMS

Send SMS to one or more recipients to notify about alert.

Coming Soon



SAP

Create service case automatically in your existing SAP instance.

Coming Soon



Logic Apps

Invoke Azure Logic Apps to simplify and implement scalable integrations and workflows in the cloud.

Coming Soon



Azure Functions

Invoke serverless code that enables you to run code on-demand in response to rule events.

Coming Soon



Microsoft Dynamics 365

Integrate with Microsoft Dynamics 365 to automatically create service tickets and schedule proactive maintenance.

Coming Soon



Salesforce

Create service case automatically in your existing Salesforce instance.



Refrigerated Vending Machine (1.0.0)

Refrigerated Vending Machine - SN01255

Measurements Settings Properties Rules Dashboard

Connect this device Delete

Machine Info

Installation Address
Installation Date
11/8/2017 7:38:42 AM
Model
Double Zone
Serial Number
SN00001

Maintenance Info

Installation Address
Installation Date
11/8/2017 7:38:42 AM
Maintenance Contract
true
Tampering Threshold
0
Temperature Alert Thres...

Customer Info

Customer Address
Customer Contact Email
Customer Contact Name
Customer Contact Phone
() - -
Customer Name

Max Temperature (de...

22.7

Past 1 week

Max Humidity (%)

58

Past 1 week

Average Pressure (hPa)

1K

Past 1 week

Internal Temperature



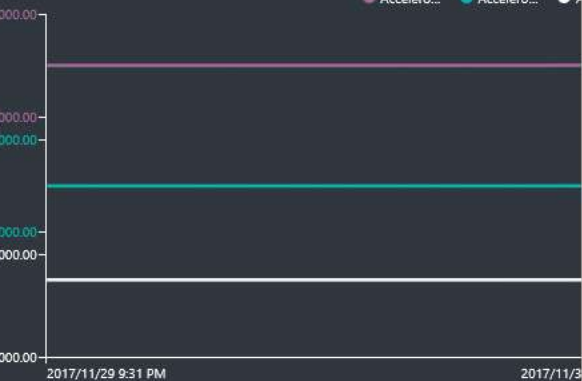
Internal Temperature Trend



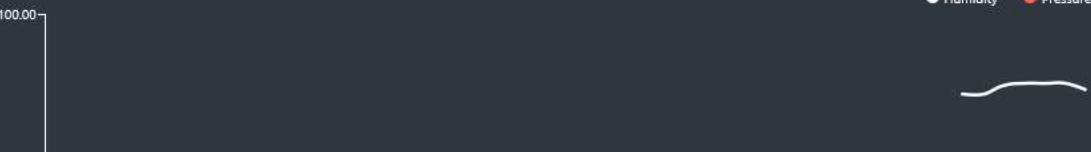
Machine Orientation (X,Y,Z Axis)



Tampering Monitor (X,Y,Z Axis)



Enviromental Data Trend



Min Temperature (de...

Average Temperature (degC)

18.3

Past 1 week

Mag Field (X Axis) (m...

Mag Field (Y Axis) (m...

Mag Field (Z Axis) (m...

Microsoft IoT Central - Simplified predictable pricing

FREE

Trial for 30 days

Includes 10 devices
and 100MB data traffic

\$0.50 USD

Per device, per month

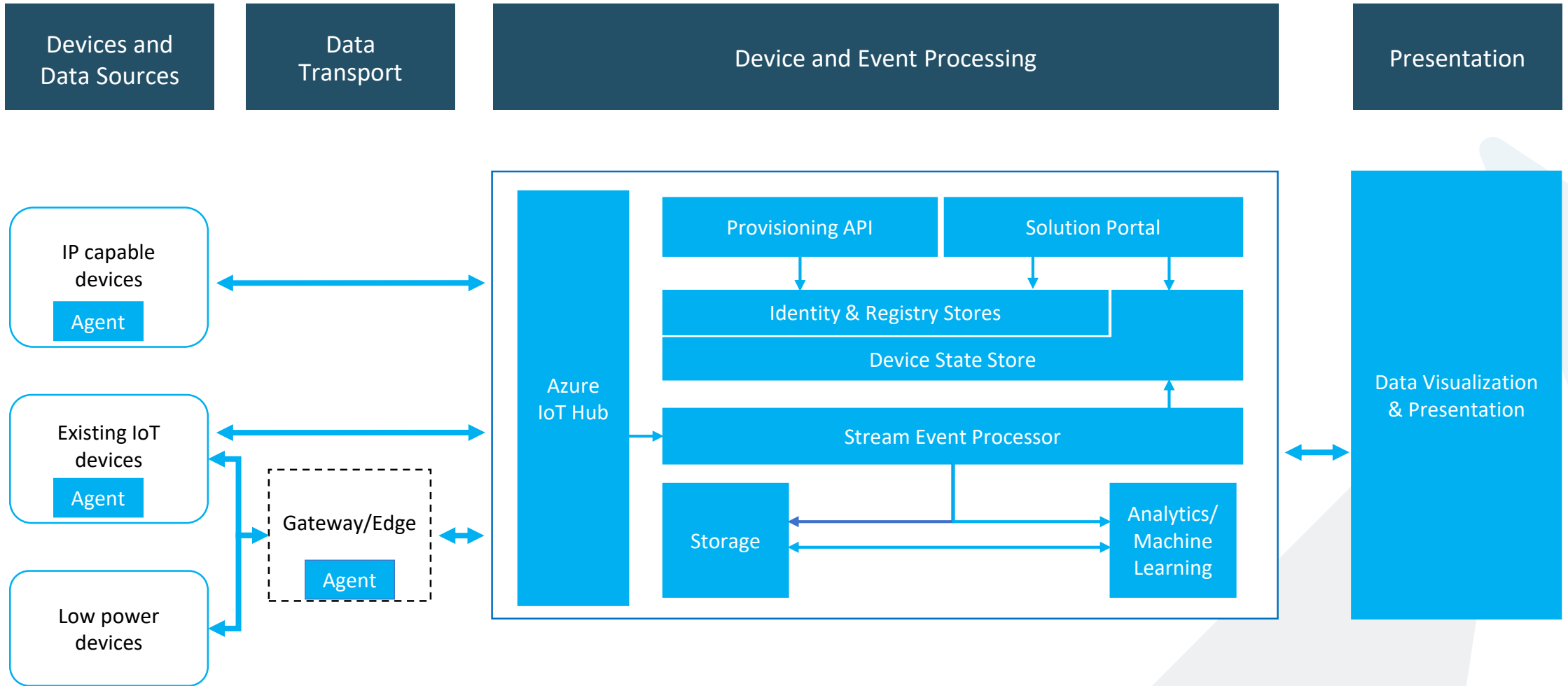
\$500 fixed fee per month *includes first 100 devices and 1000 MB data traffic*

Additional devices \$0.50 USD
includes 10 MB data traffic

Additional data traffic \$30 per 1 GB



Azure IoT Services Reference Architecture

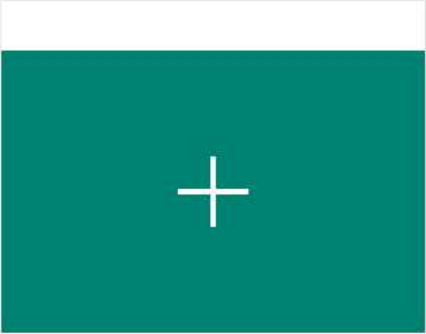


Azure IoT Suite

Azure IoT Suite

Microsoft Azure IoT Suite

Provisioned solutions ↻



Create a new solution
Create your own fully integrated provisioning solution


Feedback

Azure IoT Suite

Microsoft Azure IoT Suite

Alon Fliess
CODEVALUE LTD.

New




Remote monitoring preview
Connect and monitor your devices to analyze untapped data and improve business outcomes by automating processes.
This updated version of remote monitoring provides enhanced functionality and operator scenarios built on a new architecture that facilitates customization.
The new version includes:

- Redesigned user interface
- Microservices-based architecture
- Availability in both .NET and Java
- View an [interactive demo](#)


Get the [previous version](#).

Select




Connected factory
Accelerate your journey to Industrie 4.0 – connect, monitor and control industrial devices for insights using OPC UA to drive operational productivity and profitability.

Select



Predictive maintenance
Anticipate maintenance needs and avoid unscheduled downtime by connecting and monitoring your devices for predictive maintenance.

Select



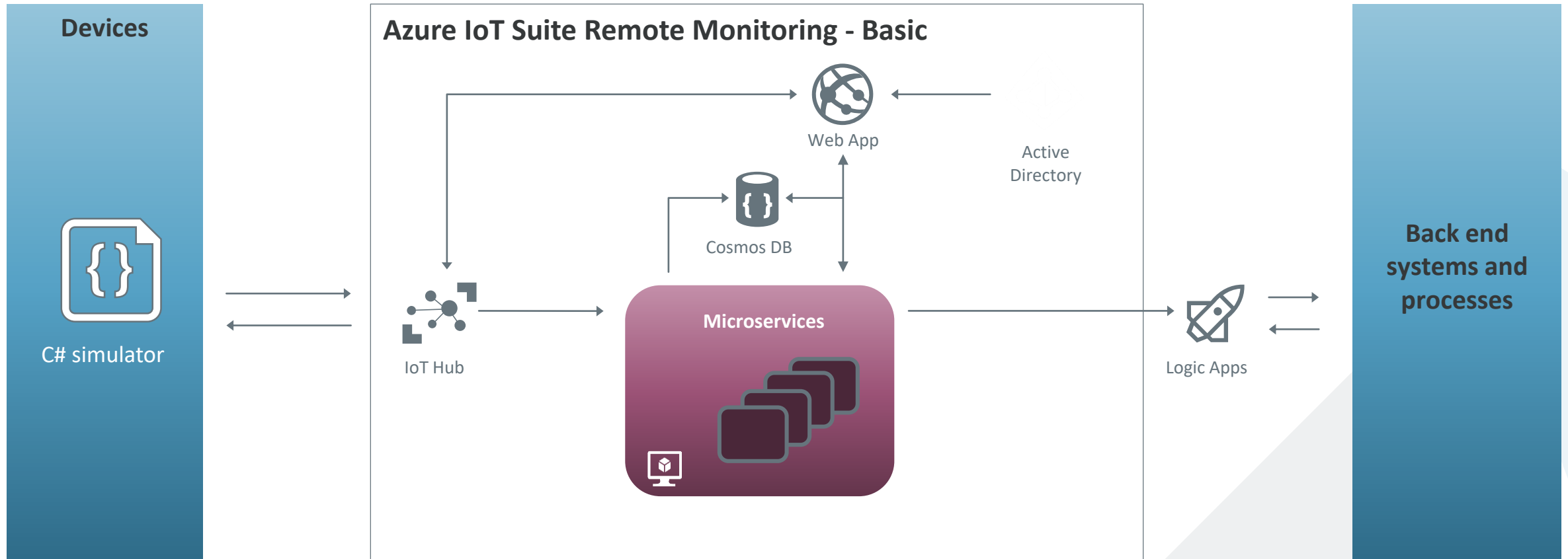
Device simulation
Streamline your IoT solution development by using simulated IoT devices to both build and test your solution throughout the software development lifecycle.

Select

Feedback



Azure IoT Suite solution – PaaS (almost) like a SaaS

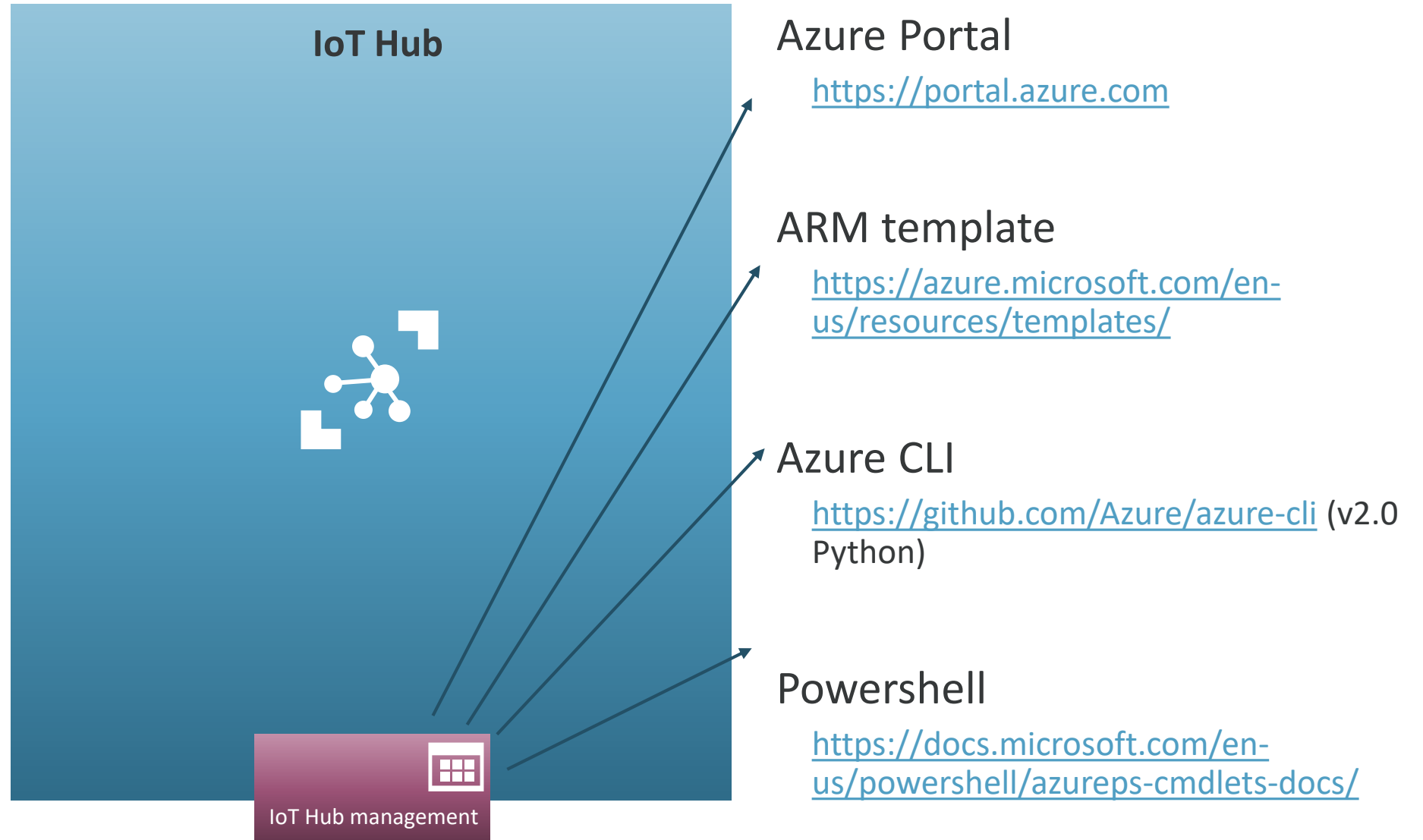


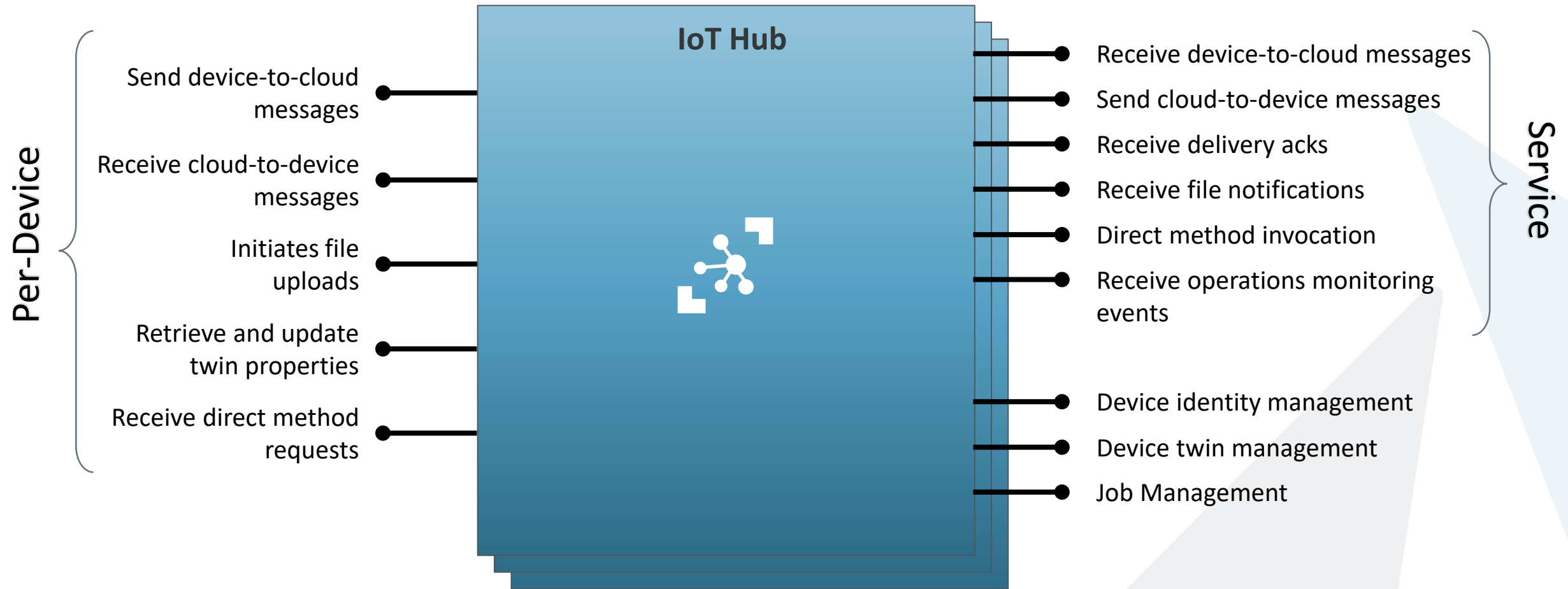


Introducing Microsoft Azure IoT Hub

- IoT Hub is available as a stand-alone service or as one of the services used in the new Azure IoT Suite
- Azure IoT Hub is designed to connect your devices to Azure. It supports:
 - Millions of simultaneously connected devices
 - Per-device authentication
 - High throughput data ingestion
 - Scale device management
 - Reliable command and control

Pick your favorite to create a hub







Azure IoT Hub SDKs

➤ **Device-facing**

- For devices and field gateways

➤ **Platforms**

- [Many devices](#)
- RTOS (FreeRTOS)
- Linux (Ubuntu, Debian, Fedora, Raspbian, Angstrom)
- Windows 7/8/10
- ARM mbed
- Android
- iOS

➤ **Device SKD by programming language**

- For device side development
- [Azure IoT device SDK for C](#)
- [Azure IoT device SDK for .NET](#)
- [Azure IoT device SDK for Java](#)
- [Azure IoT device SDK for Node.js](#)
- [Azure IoT device SDK for Python](#)

➤ **Service-facing SDK by programming language**

- For back-ends and cloud gateways
- [Azure IoT service SDK for .NET](#)
- [Azure IoT service SDK for Node.js](#)
- [Azure IoT service SDK for Java](#)
- [Azure IoT service SDK for Python](#)

➤ **Azure IoT Gateway SDK**

- Infrastructure and modules to create IoT gateway solutions

➤ **Azure IoT Hub REST API**

- For all the rest...

➤ **Advance IoT Hub topics**

- [IoT Hub endpoints](#)
- [IoT Hub query language for device twins and jobs](#)
- [Quotas and throttling](#)
- [IoT Hub MQTT support](#)

C Language Device SDK

- Many low price, low energy, SoC can be developed only by using the C language
- The IoT team has built a full-blown C SDK to connect and communicate with the IoT Hub
 - It supports all IoT Hub Device capabilities, including:
 - Secure connection and communication using three protocols (HTTP, AMQP, MQTT)
 - Sending telemetry messages using JSON serialization and set of macros to provide message serialization
 - Receiving messages from the cloud
 - Handling device twin synchronization
 - Invoke a function with request-reply message exchange pattern when the IoT Hub calls
 - Upload files
- There are two levels of functions:
 - With `*_LL_*` - low level API – for device that has no threading capabilities
 - With no `*_LL_*` - support background message processing using threads
- Follow this [intro](#) to understand the various functions

C
Language

Connecting and Defining A Model (C SDK)

```
if ((IoTClientHandle = IoTClient_LL_CreateFromConnectionString(connectionString, MQTT_Protocol)) ==  
{  
    (void)printf("ERROR: IoTClientHandle is NULL!\r\n");  
}  
else  
{
```

```
BEGIN_NAMESPACE(WeatherStation);
```

```
DECLARE_MODEL(ContosoAnemometer,  
WITH_DATA(ascii_char_ptr, DeviceId),  
WITH_DATA(int, WindSpeed),  
WITH_ACTION(TurnFanOn),  
WITH_ACTION(TurnFanOff),  
WITH_ACTION(SetAirResistance, int, Position)  
);
```

```
END_NAMESPACE(WeatherStation);
```



```
ContosoAnemometer* myWeather = CREATE_MODEL_INSTANCE(WeatherStation, ContosoAnemometer);  
if (myWeather == NULL)  
{  
    (void)printf("Failed on CREATE_MODEL_INSTANCE\r\n");  
}  
else  
{
```




IoT Hub and IoT Device Communication Protocols

- IoT Hub supports three protocols:
 - HTTP – Use for devices that cannot support other protocols or that are rarely connected
 - [AMQP](#) – Use on field and cloud gateways to take advantage of connection multiplexing across devices
 - [MQTT](#) – Extremely lightweight, Use on all devices that do not require to connect multiple devices
- You can choose to use any protocol, however you need to take some protocol characteristics into considerations:
 - HTTP does not have an efficient way to implement server push
 - As such, when you are using HTTP, devices poll IoT Hub for cloud-to-device messages
 - AMQP returns errors for many conditions, while MQTT terminates the connection
 - As a result your exception handling logic might require some changes
 - MQTT does not support the *reject* operations when receiving [cloud-to-device messages](#)
 - If your back-end app needs to receive a response from the device app, consider using [direct methods](#)
 - The MQTT and HTTP libraries have a smaller footprint than the AMQP libraries

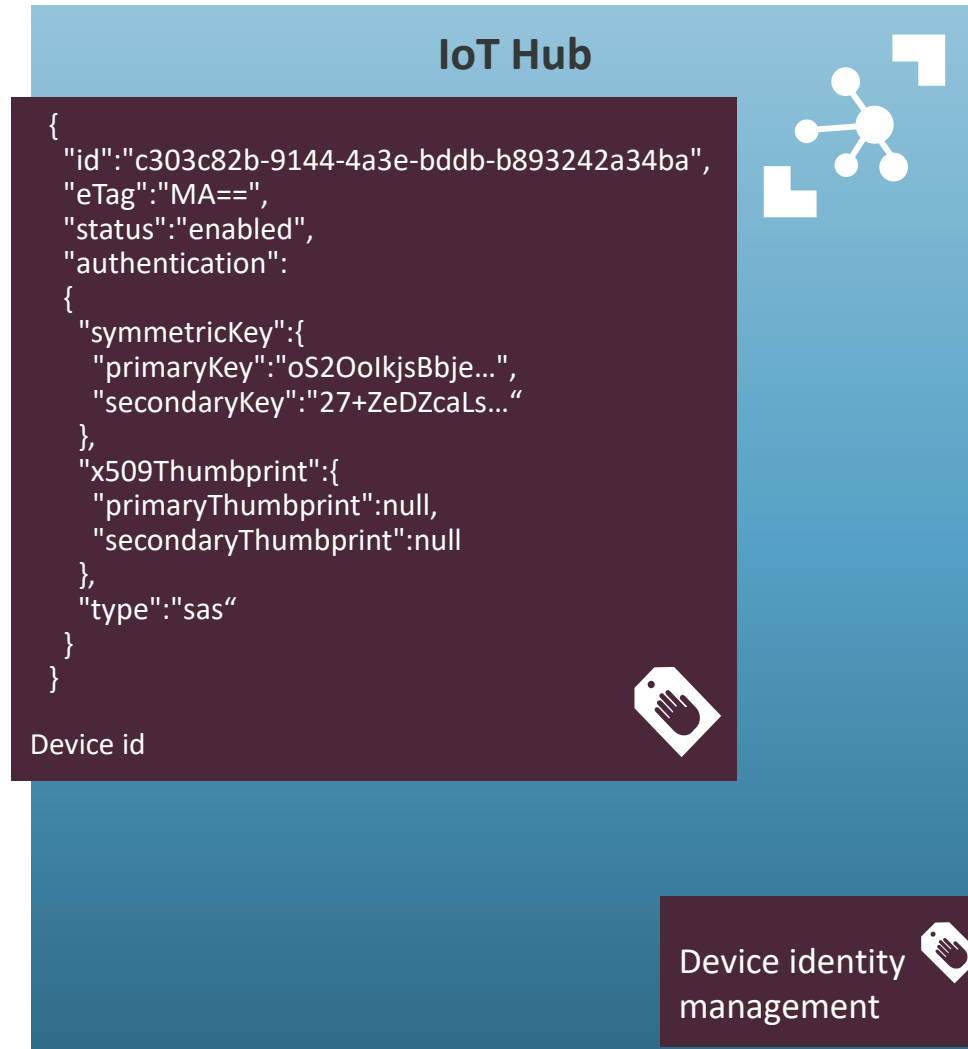


Supported Protocols & Port Numbers

Protocol	Port
MQTT	8883
MQTT over WebSockets	443
AMQP	5671
AMQP over WebSockets	443
HTTP	443

Device registry

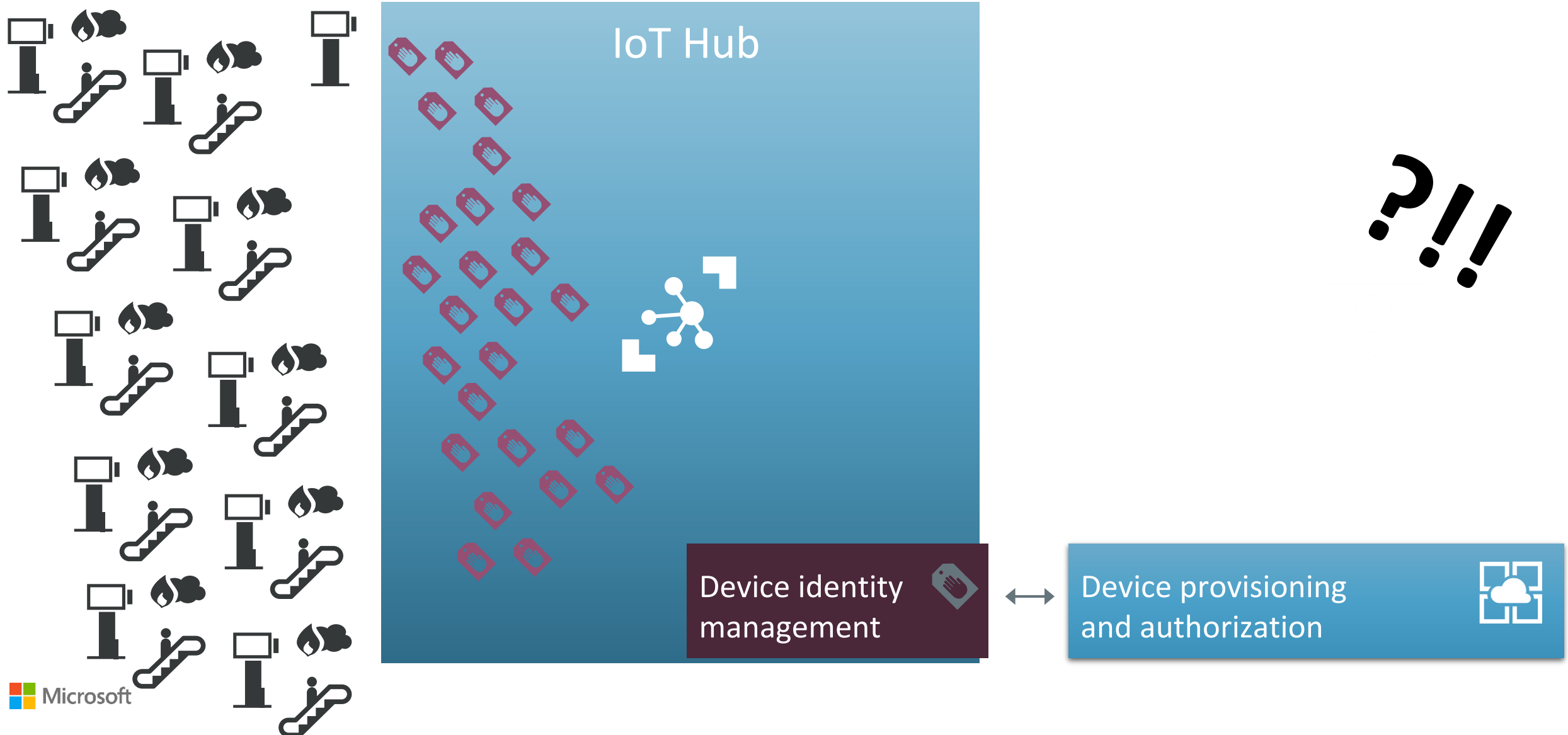
- Unique id for each device
- Unique credentials for authentication
 - Private Key/SAS Token
 - X.509 Certificates
- Device Twin



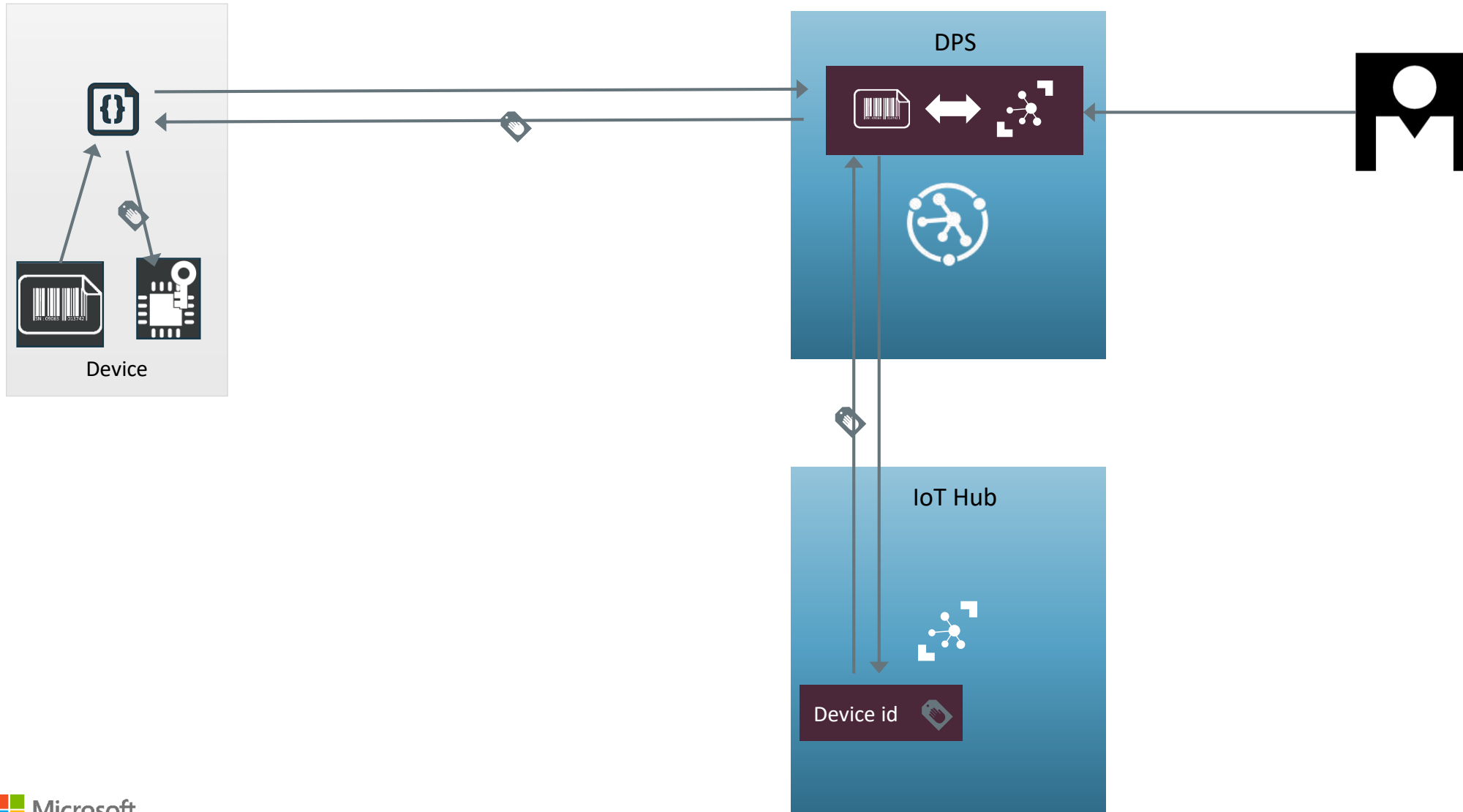
- Azure Portal
- Development tools
 - Azure CLI
 - VSCode extension
- Using a client SDK
 - .Net
 - Node
 - Java
 - Python

Device provisioning and authorization

Provisioning at scale



Device Provisioning Service





Device Twins

- Device twins are JSON documents that store device state information:
 - metadata, configurations, and conditions
- The IoT Hub persists a device twin for each registered device
- Use device twins to:
 - Store device-specific metadata in the cloud
 - Report current state information such as available capabilities and conditions from your device app
 - Synchronize the state of long-running workflows between device app and cloud app
 - Query your device metadata, configuration, or state
 - Get notified when a twin is modified



Device Twin

Device Code

Read,
Receive change
notifications

Read, Write

Device Twin

Tags

Properties

Desired

Reported

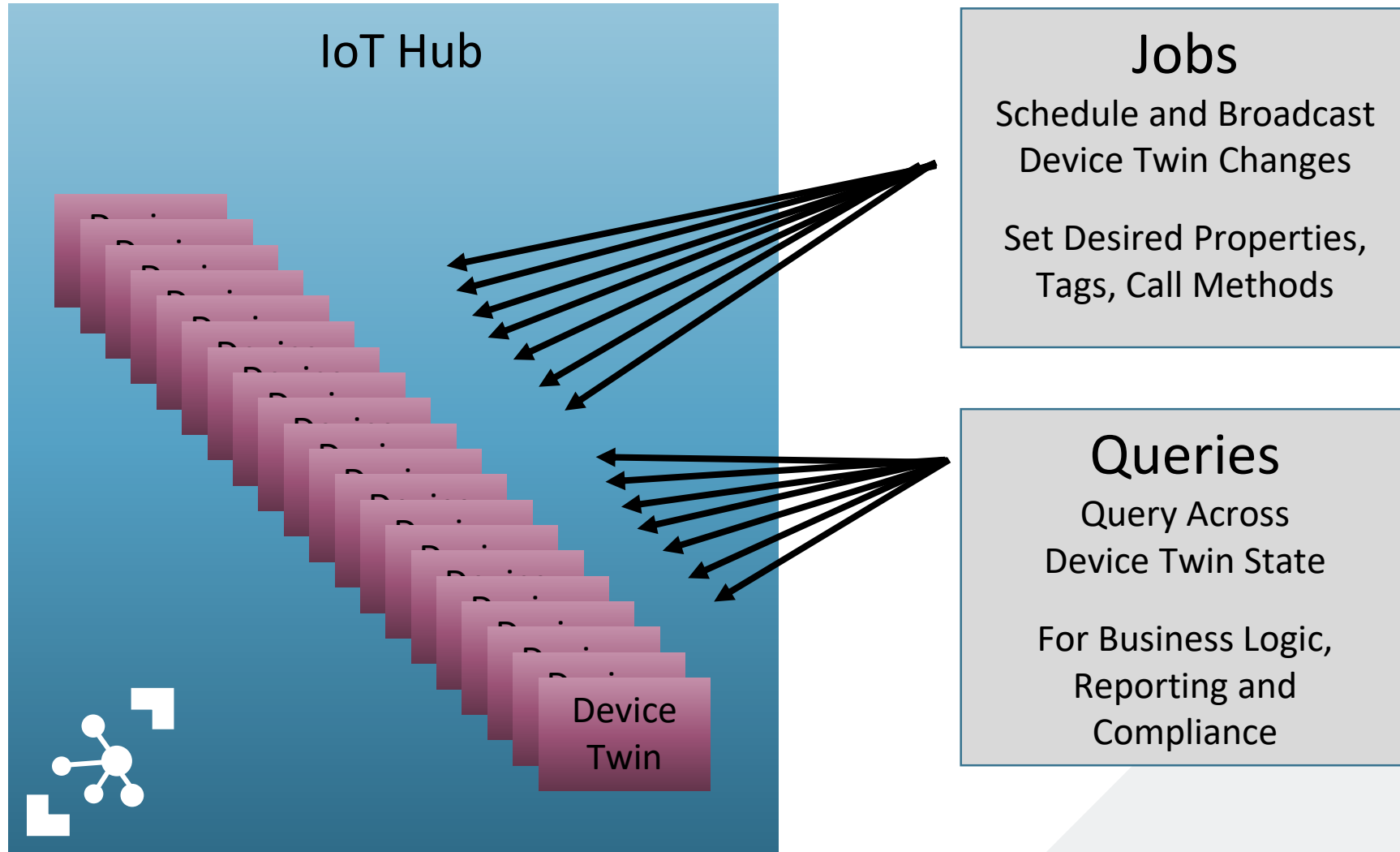
Back End Code

Read,
Write change
notifications

Read,
Write change
notifications

Read change
notifications

Designed for IoT Scale

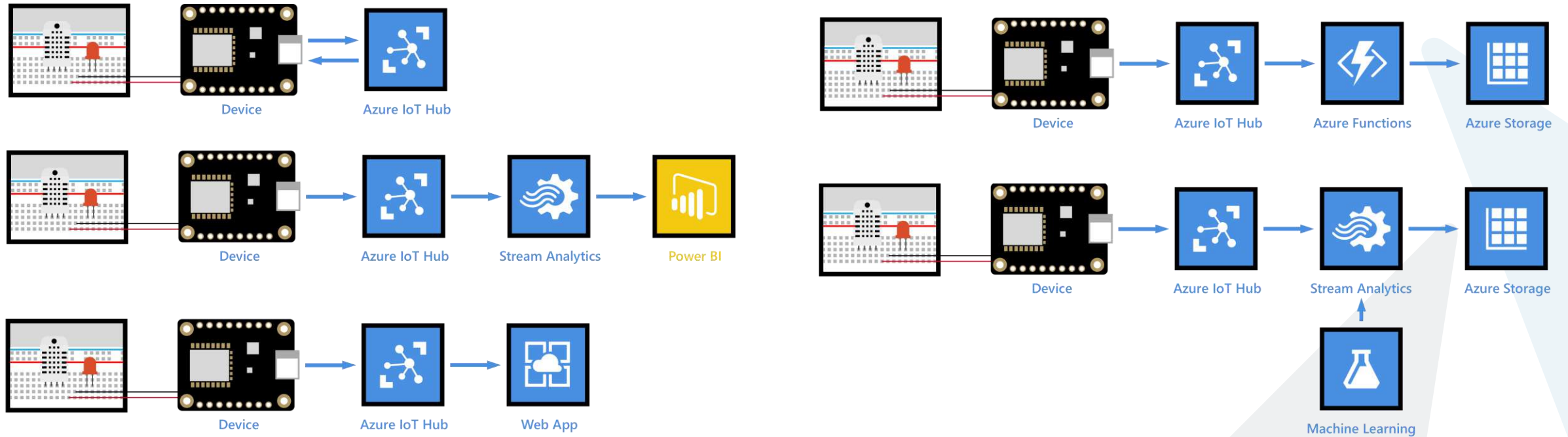




Device Jobs

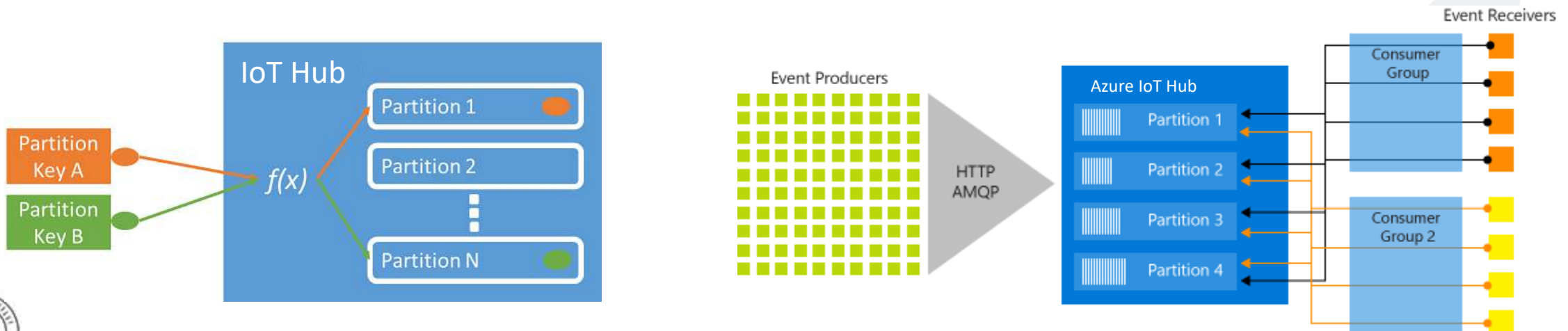
- To handle massive amount of devices and to communicate with offline devices, use Jobs:
 - Jobs encapsulate the execution of device twin updates and direct methods against a set of devices at a schedule time
 - The job is described as a JSON document
- Jobs are initiated by the cloud app and maintained by IoT Hub
 - Once a job is initiated, querying for jobs enables the cloud app to refresh the status of running jobs
- [More information](#)

What's next now that I have data flowing in?



Processing IoT Hub Messages – Event Hub

- You can process IoT Hub device to cloud messages using either:
 - The built-in Event-Hub compatible endpoint
 - Rout the events to an Azure Service Bus queue
- Azure Event Hub is a very powerful telemetry ingestion service that was created by the Service Bus team
 - The key to scale for Event Hubs is the idea of **partitioned consumers**
 - **Partitioned consumers** enables very high scale by removing the contention bottleneck and facilitating end to end parallelism



Routing telemetry data

Header:

```
$content-encoding="utf-8"  
$content-type =  
"application/json"  
my-message-type = "alert"
```

Body

```
{  
  "Weather":{  
    "Temperature":50,  
    "Time":"01:23:12Z"  
  }  
}
```

Device



IoT Hub

```
SELECT * FROM Devices  
WHERE tags.location = "bldg43"
```

```
my-message-type = "alert"
```

```
$body.Weather.Temperature > 75
```

Routing rules (query)

D2C receive endpoint

Custom endpoints
connectors

Event processing
(hot path)

Event Hub

Service Bus Topics

Service Bus Queues

Storage Blob

Event Grid

Event processing (hot and cold path)





Direct Method - Calling a function in the device

- As opposed to other IoT Hub message exchange patterns that are one-way, a method call is a request-reply interaction
 - Other cloud to device communication are based on sending messages to the device, or setting desired properties
- Each device method targets a single device
 - [Jobs](#) provide a way to invoke direct methods on multiple devices, and schedule method invocation for disconnected devices
- Direct methods are synchronous and either succeed or fail
 - Failure occurs after a timeout period (default: 30 secs, settable up to 1 Hour)
- Great for interactive scenarios such as turning on a light from a phone
- Direct method are HTTP-only from the cloud side, and MQTT-only from the device side
- The payload for method requests and responses is a JSON document up to 8KB

Direct Method

```
private static async Task InvokeMethod()
{
    var methodInvocation = new CloudToDeviceMethod("writeLine") { ResponseTimeout = TimeSpan.FromSeconds(30) };
    methodInvocation.SetPayloadJson("'a line to be written'");

    var response = await serviceClient.InvokeDeviceMethodAsync("myDeviceId", methodInvocation);

    Console.WriteLine("Response status: {0}, payload:", response.Status);
    Console.WriteLine(response.GetPayloadAsJson());
}
```

```
serviceClient = ServiceClient.CreateFromConnectionString(connectionString);
InvokeMethod().Wait();
Console.WriteLine("Press Enter to exit.");
Console.ReadLine();
```

Device Side C SDK – Handling direct method

```
else if (IoTHubClient_LL_SetDeviceMethodCallback(iotHubClientHandle, DeviceMethodCallback, myWeather) != IOTHUB_CLIENT_OK)
{
    (void)printf("Failed on IoTHubClient_SetDeviceMethodCallback\r\n");
}
```




Upload Files

- Use file upload to send media files and large telemetry batches
- You must first link an Azure Storage account to the IoT Hub
 - You can do that using the portal
- The device initiates an [upload](#)
- When the upload completes, the device [notifies the IoT hub](#)
- See [file upload notifications](#)
- The SDK makes it easy: (C#)

```
private static async void SendToBlobAsync()
{
    string fileName = "image.jpg";
    Console.WriteLine("Uploading file: {0}", fileName);
    var watch = System.Diagnostics.Stopwatch.StartNew();

    using (var sourceData = new FileStream(@"image.jpg", FileMode.Open))
    {
        await deviceClient.UploadToBlobAsync(fileName, sourceData);
    }

    watch.Stop();
    Console.WriteLine("Time to upload file: {0}ms\n", watch.ElapsedMilliseconds);
}
```


Waves of Innovation

The smart cloud & Intelligent Edge

Cloud

Globally available, unlimited compute resources

IoT

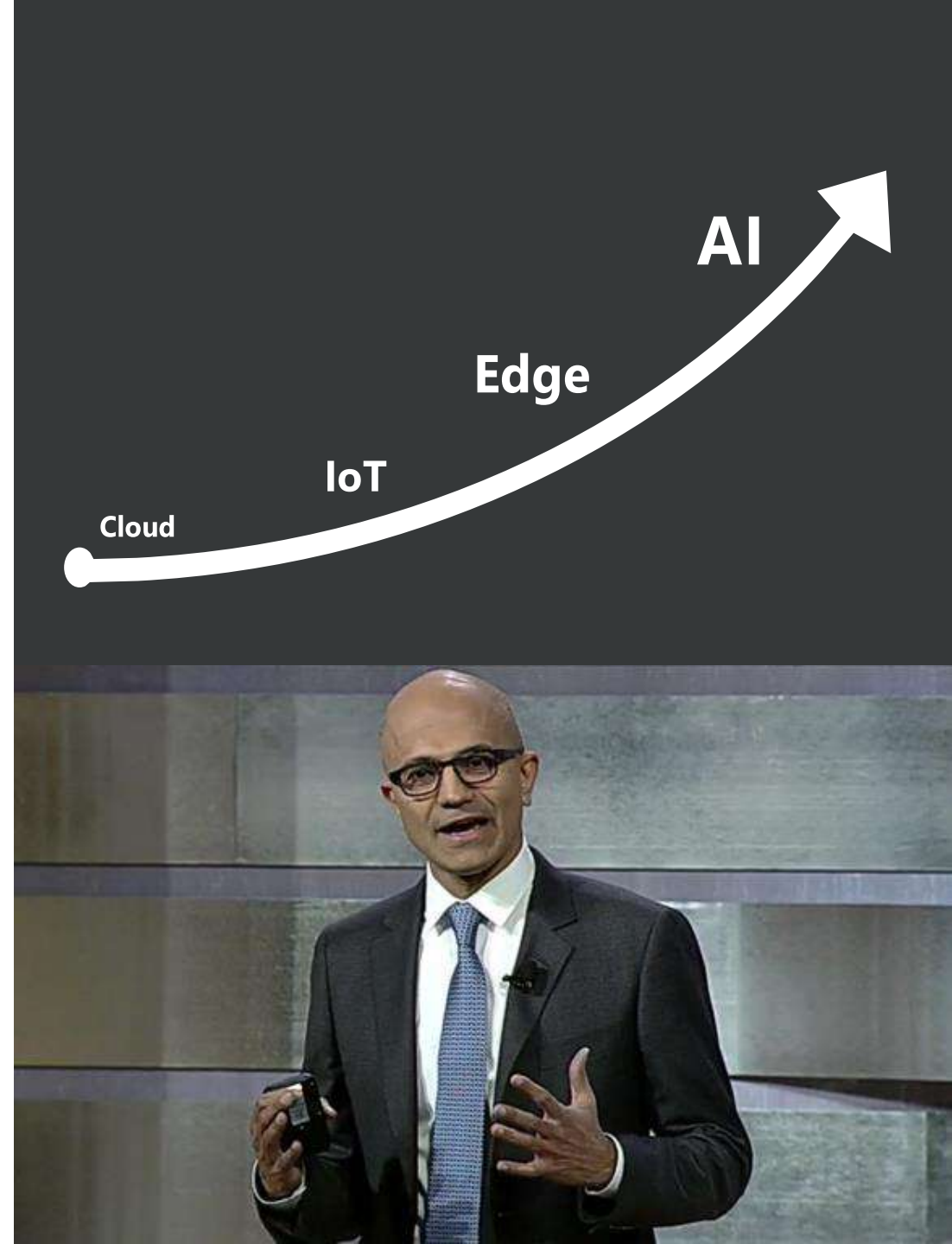
Harnessing signals from sensors and devices, managed centrally by the cloud

Edge

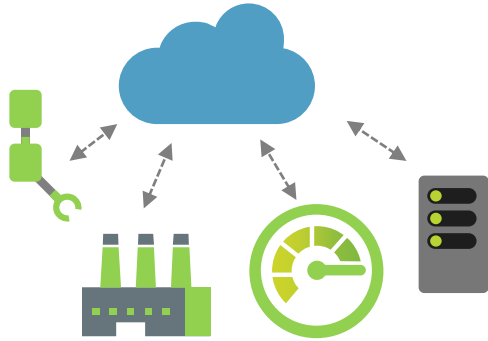
Intelligence offloaded from the cloud to IoT devices

AI

Breakthrough intelligence capabilities

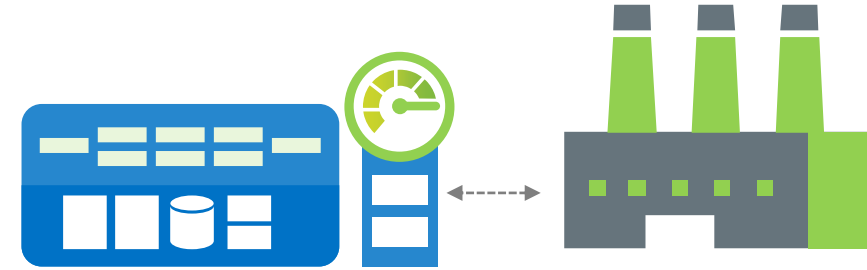


Why the edge?



IoT in the Cloud

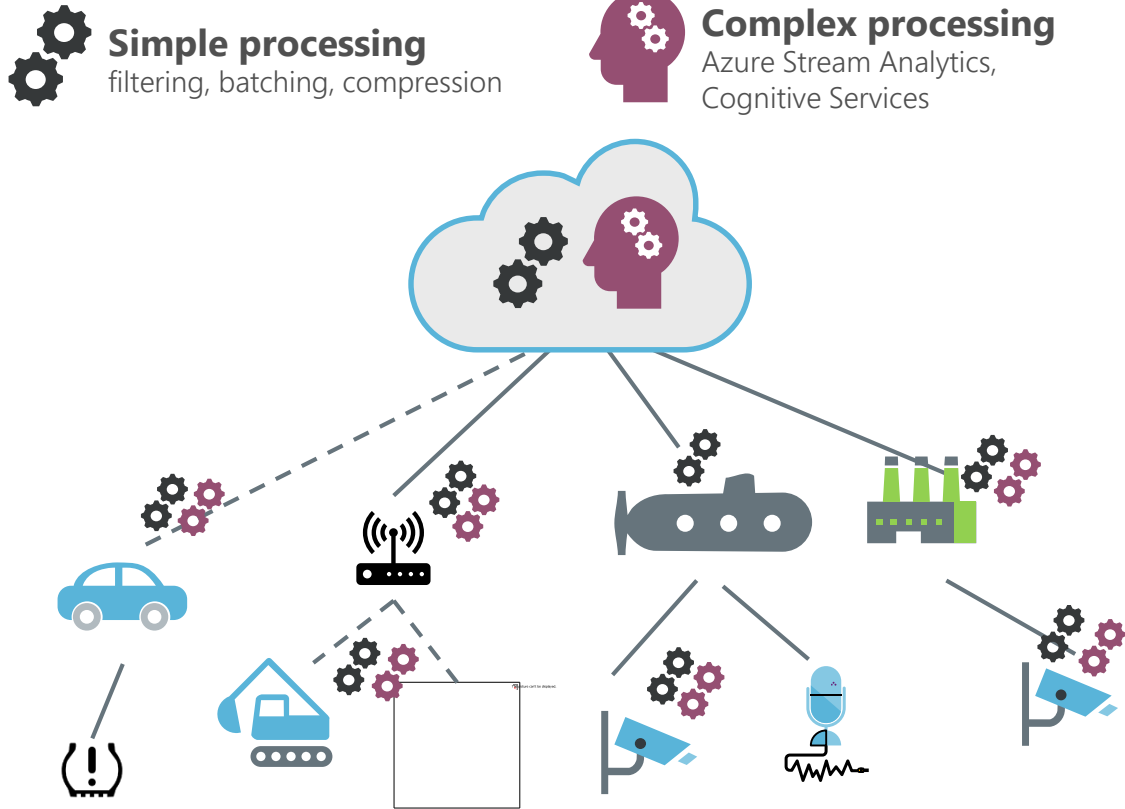
- Remote monitoring and control
- Merging remote data from across multiple IoT devices
- Near infinite compute and storage to train machine learning and other advanced AI tools



IoT on the Edge

- Low latency tight control loops require near real-time response
- Public internet inherently unpredictable
- Privacy of data and protection of IP

Azure IoT Edge



➤ Secure

- A Secure connection to the Azure IoT Edge
- Collect state and telemetry and monitor security of the device

➤ Cloud Managed

- Enable rich management from Azure

➤ Cross-Platform

- Enable Azure IoT Edge on both Windows and Linux

➤ Portable

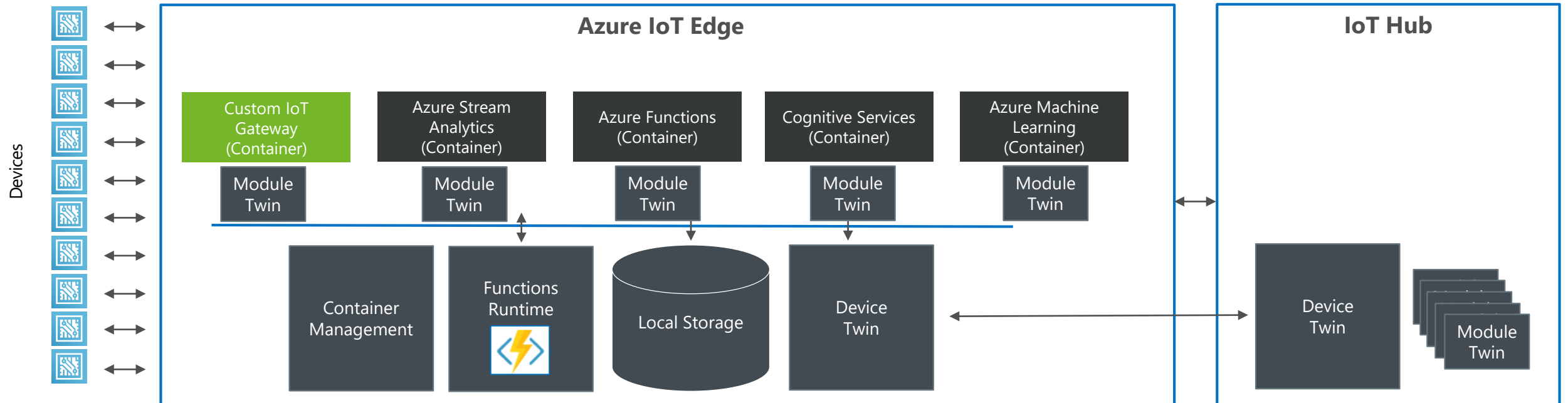
- Enable creating Docker Images that target multiple architecture

➤ Extensible

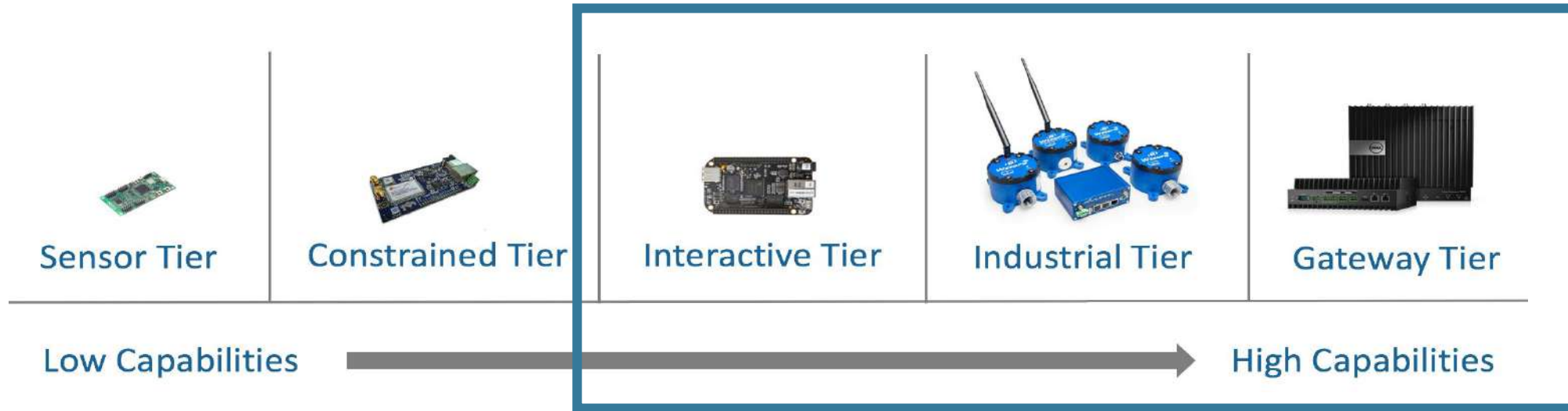
- Enable seamless deployment of advanced capabilities modules such as **AI**, **Azure Function**, **Stream Analytics** and **3rd party**

Azure IoT Edge

- Container based modules
- Azure Functions
- Azure Stream Analytics
- Azure Machine Learning
- Cognitive Services
- Offline / Synchronized Device Twins
- Local Storage
- Cloud Management & Deployment
- High Availability / Fault Tolerance
- Cloud Dev/Test Support



Hardware for Azure IoT Edge



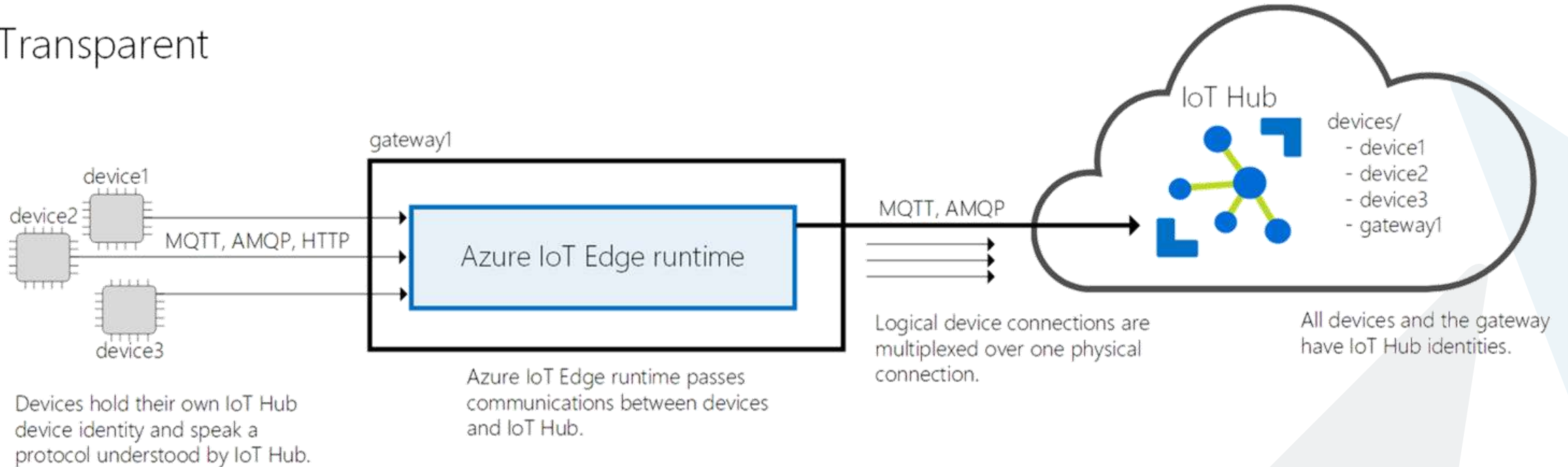
Ability to run on devices smaller than a Raspberry Pi

128MB memory

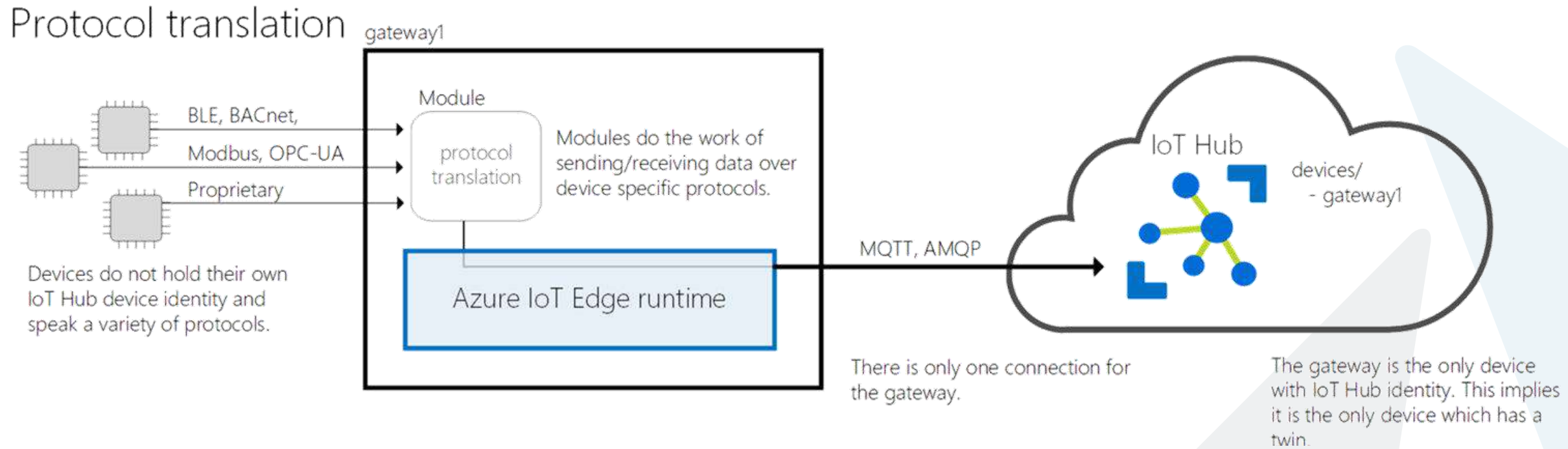
Support best in class operating systems such as Windows, and Linux

IoT Edge as a Gateway - Transparent

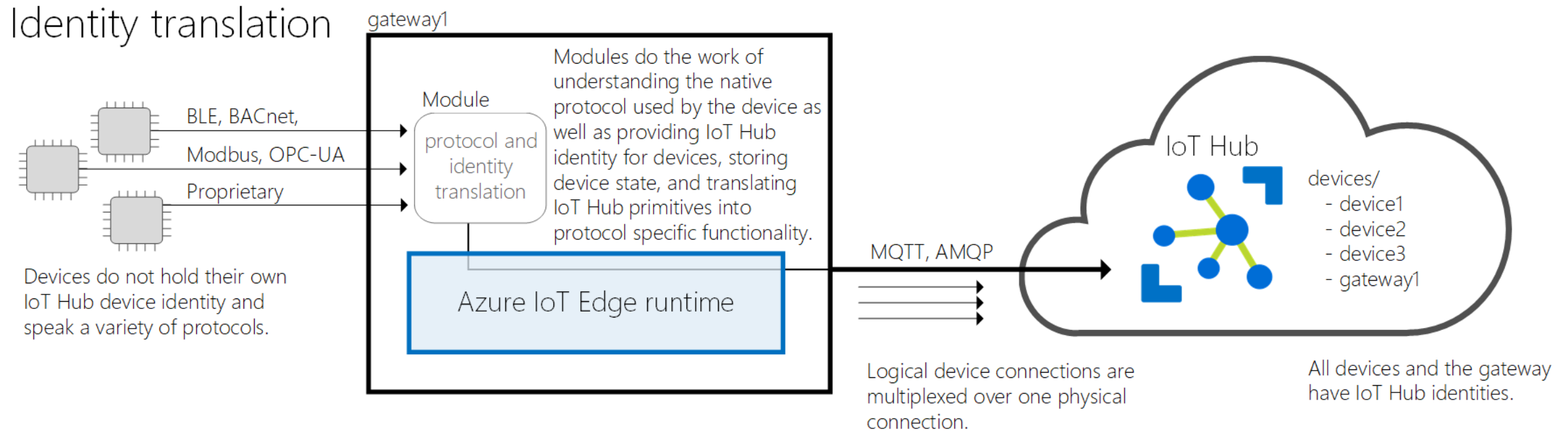
Transparent



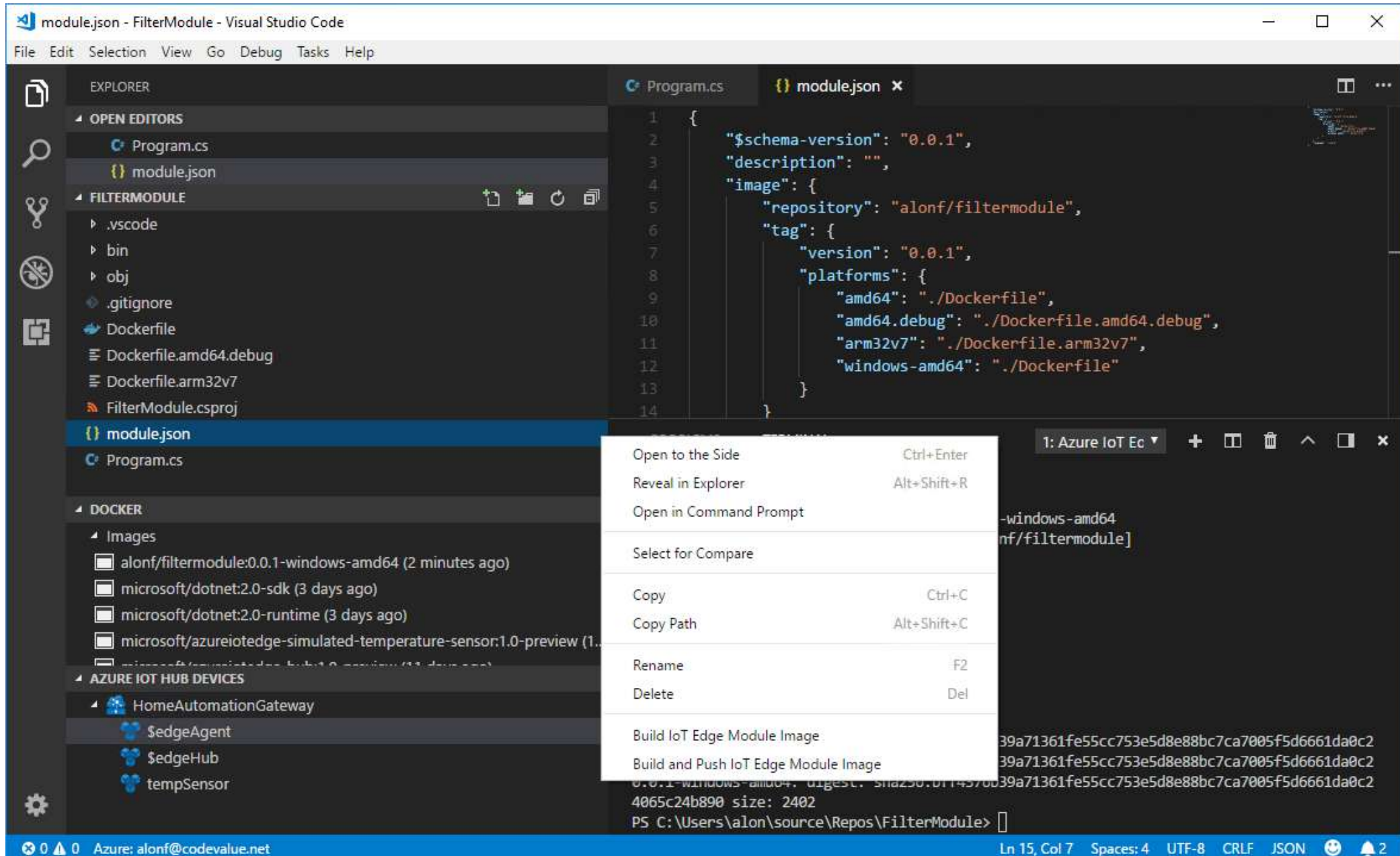
IoT Edge as a Gateway - Protocol Translation



IoT Edge as a Gateway – Identity Translation



Visual Studio Code IoT Edge Extension



The screenshot displays the Visual Studio Code interface with the IoT Edge extension installed. The Explorer sidebar on the left shows the project structure for 'FilterModule', including files like 'Program.cs', 'module.json', and 'Dockerfile'. The Docker sidebar at the bottom left lists Docker images and Azure IoT Hub devices. The main editor area shows the 'module.json' file, which contains metadata for the IoT Edge module. A context menu is open over the 'module.json' file, providing options to manage the module, such as 'Build IoT Edge Module Image' and 'Build and Push IoT Edge Module Image'.

module.json - FilterModule - Visual Studio Code

File Edit Selection View Go Debug Tasks Help

EXPLORER

OPEN EDITORS

- Program.cs
- module.json

FILTERMODULE

- .vscode
- bin
- obj
- .gitignore
- Dockerfile
- Dockerfile.amd64.debug
- Dockerfile.arm32v7
- FilterModule.csproj
- module.json
- Program.cs

DOCKER

- Images
 - alonf/filtermodule:0.0.1-windows-amd64 (2 minutes ago)
 - microsoft/dotnet:2.0-sdk (3 days ago)
 - microsoft/dotnet:2.0-runtime (3 days ago)
 - microsoft/azureiotedge-simulated-temperature-sensor:1.0-preview (11 days ago)
- AZURE IOT HUB DEVICES
 - HomeAutomationGateway
 - \$edgeAgent
 - \$edgeHub
 - tempSensor

Program.cs

```
1 {
2     "$schema-version": "0.0.1",
3     "description": "",
4     "image": {
5         "repository": "alonf/filtermodule",
6         "tag": {
7             "version": "0.0.1",
8             "platforms": {
9                 "amd64": "./Dockerfile",
10                "amd64.debug": "./Dockerfile.amd64.debug",
11                "arm32v7": "./Dockerfile.arm32v7",
12                "windows-amd64": "./Dockerfile"
13            }
14        }
15    }
16 }
```

module.json

```
1 {
2     "$schema-version": "0.0.1",
3     "description": "",
4     "image": {
5         "repository": "alonf/filtermodule",
6         "tag": {
7             "version": "0.0.1",
8             "platforms": {
9                 "amd64": "./Dockerfile",
10                "amd64.debug": "./Dockerfile.amd64.debug",
11                "arm32v7": "./Dockerfile.arm32v7",
12                "windows-amd64": "./Dockerfile"
13            }
14        }
15    }
16 }
```

1: Azure IoT Ec

Open to the Side Ctrl+Enter

Reveal in Explorer Alt+Shift+R

Open in Command Prompt

Select for Compare

Copy Ctrl+C

Copy Path Alt+Shift+C

Rename F2

Delete Del

Build IoT Edge Module Image

Build and Push IoT Edge Module Image

39a71361fe55cc753e5d8e88bc7ca7005f5d6661da0c2

39a71361fe55cc753e5d8e88bc7ca7005f5d6661da0c2

0.0.1-windows-amd64: digest: sha256:0b14570039a71361fe55cc753e5d8e88bc7ca7005f5d6661da0c2

4065c24b890 size: 2402

PS C:\Users\alon\source\Repos\FilterModule>

Ln 15, Col 7 Spaces: 4 UTF-8 CRLF JSON



Price & Capabilities

FEATURE	BASIC	STANDARD
Device-to-cloud telemetry	✓	✓
Per-device identity	✓	✓
Message Routing, Event Grid Integration	✓	✓
HTTP, AMQP, MQTT Protocols	✓	✓
DPS Support	✓	✓
Monitoring and diagnostics		✓
Cloud-to-device messaging		✓
Device Management, Device Twin		✓
IoT Edge		✓



Price & Capabilities

BASIC TIER	PRICE PER UNIT (PER MONTH)	TOTAL NUMBER OF MESSAGES/DAY PER UNIT	MESSAGE METER SIZE	Max # of Units
B1	\$10	400,000	4 KB	200
B2	\$50	6,000,000	4 KB	200
B3	\$500	300,000,000	4 KB	10

STANDARD TIER	PRICE PER UNIT (PER MONTH)	TOTAL NUMBER OF MESSAGES/DAY PER UNIT	MESSAGE METER SIZE	Max # of Units
FREE	FREE	8,000	0.5 KB	1
S1	\$25	400,000	4 KB	200
S2	\$250	6,000,000	4 KB	200
S3	\$2,500	300,000,000	4 KB	10

Operation throttles & Other Limits:

<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-quotas-throttling>



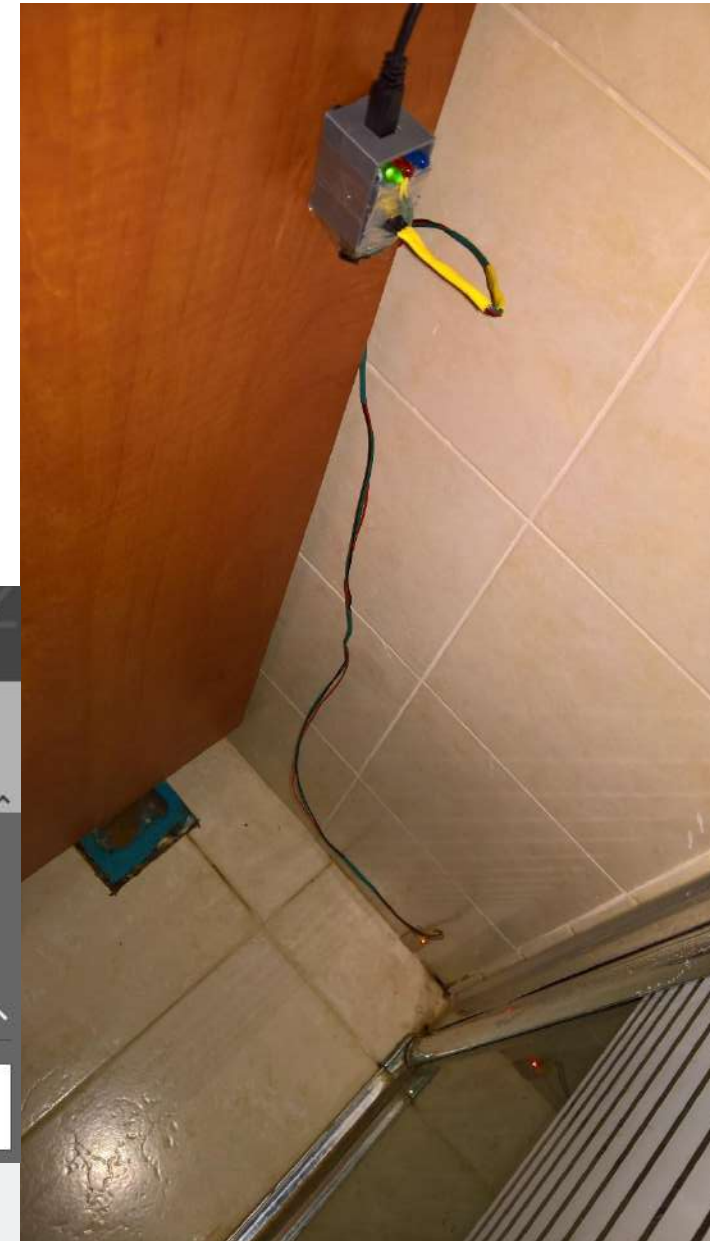
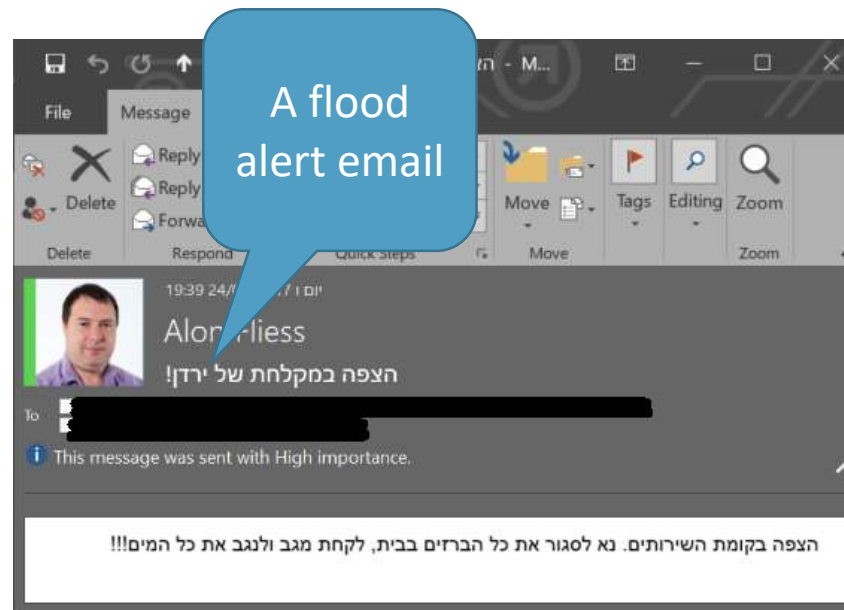
Limits, Quota & Throttling

Tier	Sustained throughput	Sustained send rate
B1, S1	Up to 1111 KB/minute per unit (1.5 GB/day/unit)	Average of 278 messages/minute per unit (400,000 messages/day per unit)
B2, S2	Up to 16 MB/minute per unit (22.8 GB/day/unit)	Average of 4,167 messages/minute per unit (6 million messages/day per unit)
B3, S3	Up to 814 MB/minute per unit (1144.4 GB/day/unit)	Average of 208,333 messages/minute per unit (300 million messages/day per unit)



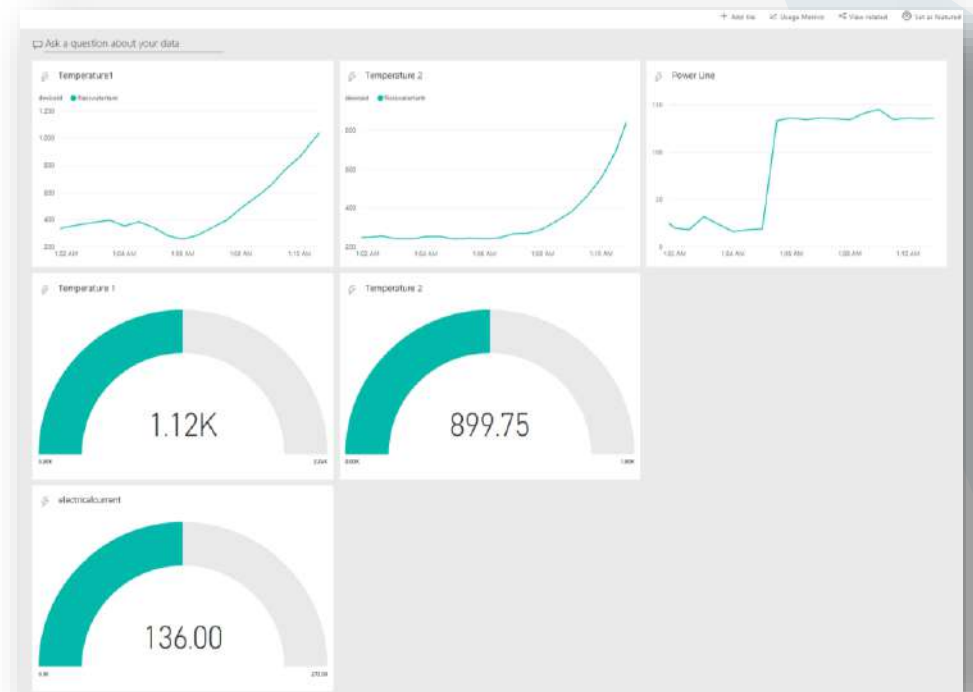
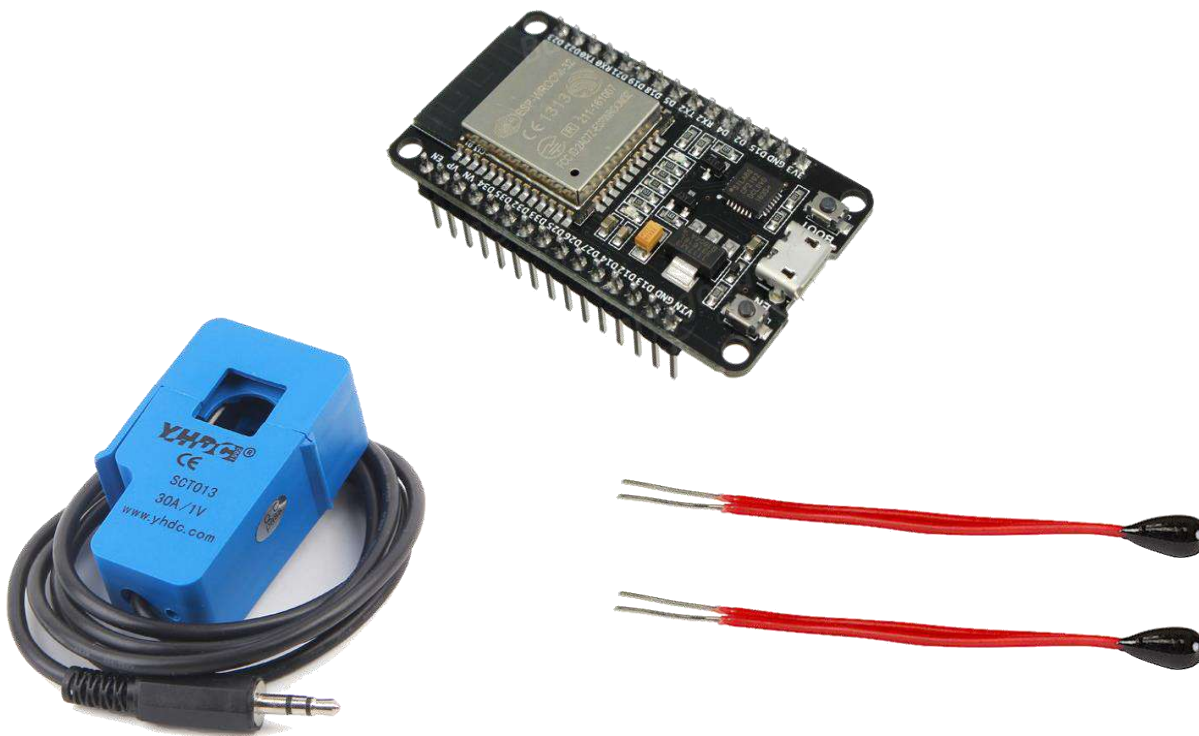
A True War Story

- My wife is shouting:
 - ALON!!! There's water everywhere!!!
- By the time the plumber came to fix it, in 4 hours I had this:
- A water flood detector
 - Using small device + Azure IoT Hub + Service Bus Queue + Azure Logic App + Office 365 Email + Twilio SMS



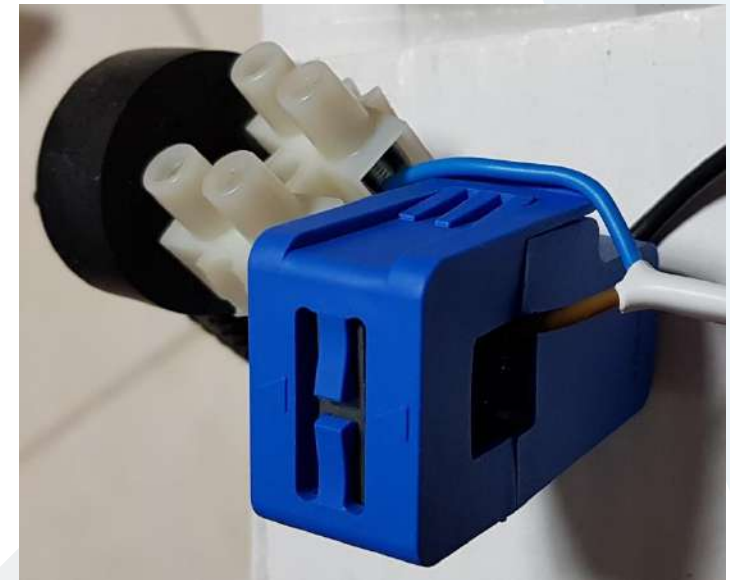
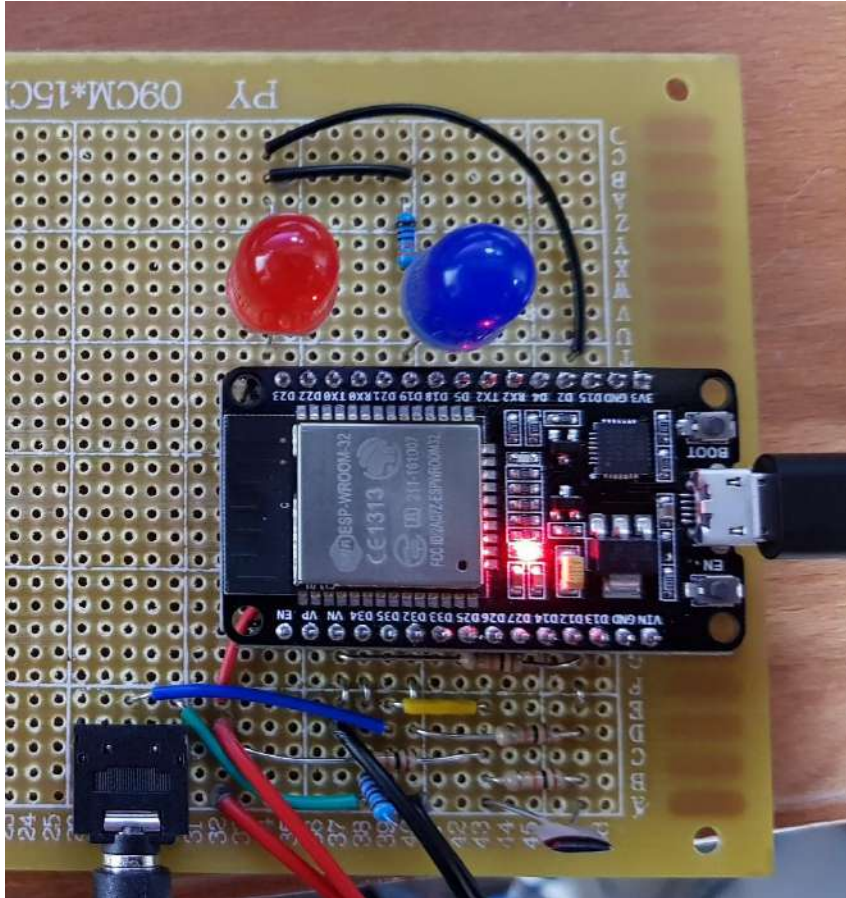
The Water-Tank Boiler System

- Based on ESP32 + 2 NTC Thermistors + Current Sensor
- Hot water, when needed while saving energy!
- Over the air firmware update



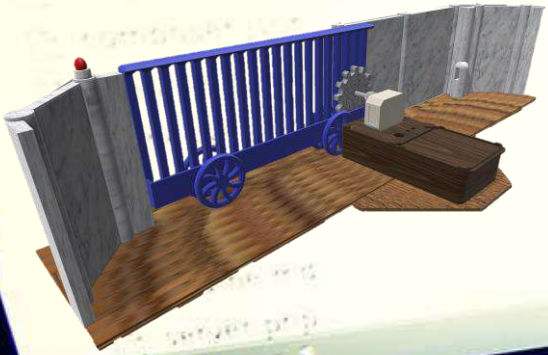
A Smart Boiler System

IoT Hub + Routing to ServiceBus Queue + Stream Analytics + Azure Function + PowerBI + Stream Insight





Demo



Line 81, Column 1

```
68 $mockQueryBuilder->shouldReceive('newQuery')->once()->andReturn($query);
69 $relation->expects($this->once())->method('touchIfTouching');
70
71 $this->assertTrue($relation->detach());
72 }
73
74 public function getRelation()
75 {
76     list($builder, $parent) = $this->getRelationArguments();
77
78     return new MorphToMany($builder, $parent, 'taggable', 'taggables', 'taggable_id', 'tag_id');
79 }
80
81 public function getRelationArguments()
82 {
83     $parent = m::mock('Illuminate\Database\Eloquent\Model');
84     $parent->shouldReceive('getMorphClass')->andReturn(get_class($parent));
85     $parent->shouldReceive('getKey')->andReturn(1);
86 }
87
```




Smart Boiler - To-do...

- Use Azure IoT edge
 - Run the Azure Function locally
 - Run stream analytics locally
- Add AI module
 - Water usage patterns
 - Save even more energy
- Scale to many tenant
 - Device provisioning portal
 - Device settings and management portal





Azure IoT Summary

- IoT system architecture is a bit different than other cloud architecture
 - A “Pettle” – each device count!
- Microsoft provides SaaS and PaaS solutions
 - Azure IoT Central, Azure IoT Suite, Azure IoT Hub and cloud services
- Azure IoT Hub is designed to connect your devices to Azure. It supports:
 - SDKs, Millions of simultaneously connected devices, Per-device authentication, High throughput data ingestion, Scale device management
 - HTTP, MQTT, AMQP communication protocols
 - Cloud to Device and Device to Cloud messaging
 - State transfer with device twins
 - Query language, Job Management, File Upload
- Smart cloud & intelligent Edge

Activating a Real Gate



הפעל

OK

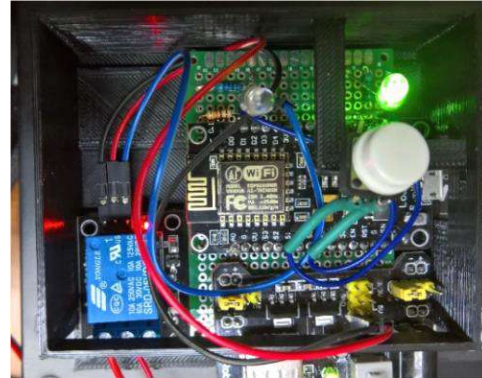
**Welcome to the Fliess &
Fisher home gate control
system**



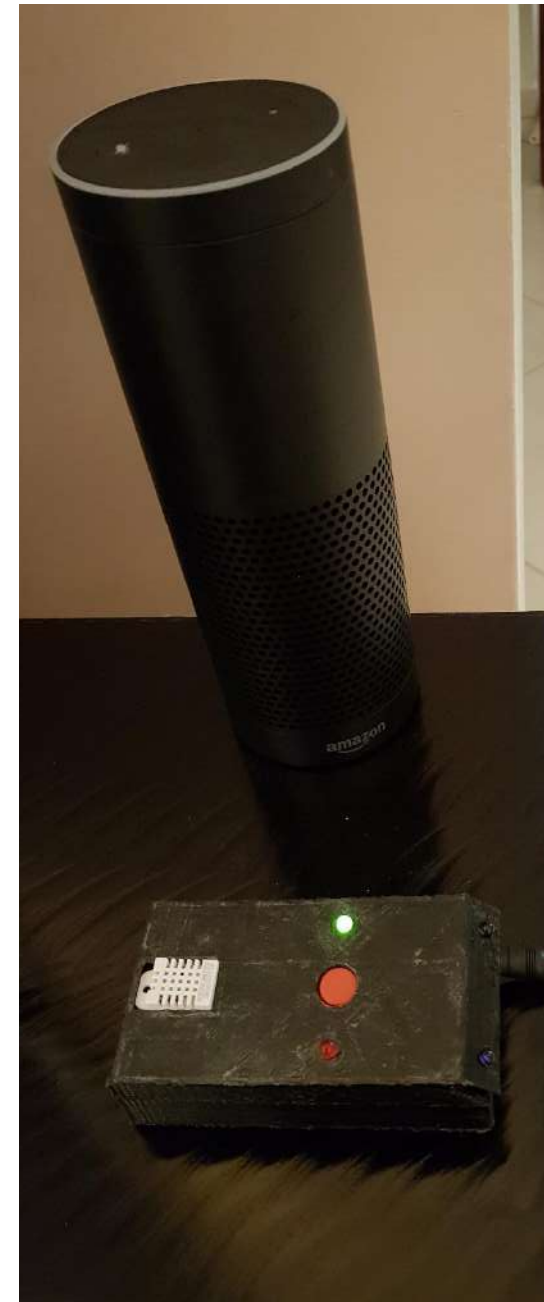
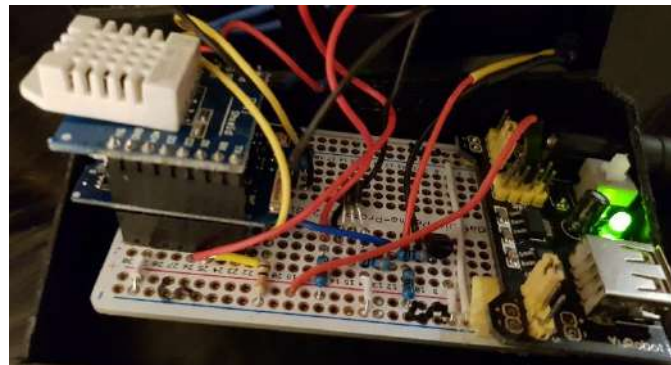


Other Usage

- My 3D Printer LED light



- Alexa: Turn on ground level AC
- Alexa set ground level AC to 20 degrees





Resources

- Demo code:
 - <https://github.com/alonf/BasicGateController>
 - Setup IoT Hub video: <https://youtu.be/vq5AeLlsWx4>
- My MSDN articles:
 - [Introduction to the Internet of Things – From the Device to Microsoft Azure Cloud](#)
 - https://blogs.msdn.microsoft.com/microsoft_press/2015/04/27/from-the-mvps-introduction-to-the-internet-of-things-from-the-device-to-microsoft-azure-cloud/
 - [Efficient IoT With Azure](#)
 - <https://blogs.msdn.microsoft.com/mvpawardprogram/2016/11/15/efficient-iot-with-azure/>
 - [Secure Provisioning of IoT device using Azure IoT Hub device SDK](#)
 - <https://blogs.msdn.microsoft.com/mvpawardprogram/2017/03/14/provisioning-of-iot-device/>
- Thingiverse
 - <http://www.thingiverse.com/thing:2253418>
- Azure IoT
 - [IoT SDKs](#) - <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide-sdks>
 - [GitHub](#) - <https://github.com/Azure/azure-iot-sdks>
 - [Azure IoT Suite](#) - <https://azure.microsoft.com/en-us/suites/iot-suite/>
 - [Azure IoT Hub](#) - <https://azure.microsoft.com/en-us/services/iot-hub/>

C# and Xamarin





- A multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented and component-oriented programming disciplines
- Developed by Microsoft and approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006)
- C# is intended to be a simple, modern, general-purpose, object-oriented programming language
- The most recent version is C# 6.0, which was released on July 20, 2015

C# Basics

- The platform – Compiler, Jitter, CLR, GC, Libraries (references), .NET Native
- Types, Value types and reference types (boxing), Metadata (ILDASM)
- Class and Interfaces, explicit interface, polymorphism (abstract, override, new), partial
- Methods, ref and out parameters, params, lambda expression
- Exceptions, checked and unchecked, finally
- Control Flow (if-else, while, do-while, for, foreach, switch)
- Operator overloading
- Properties
- Delegates & Events
- Attributes
- Generic, constraints
- Enumerators (yield return)
- LINQ
- Task, Async/Await

https://en.wikipedia.org/wiki/Comparison_of_C_Sharp_and_Java

Xamarin (Forms)



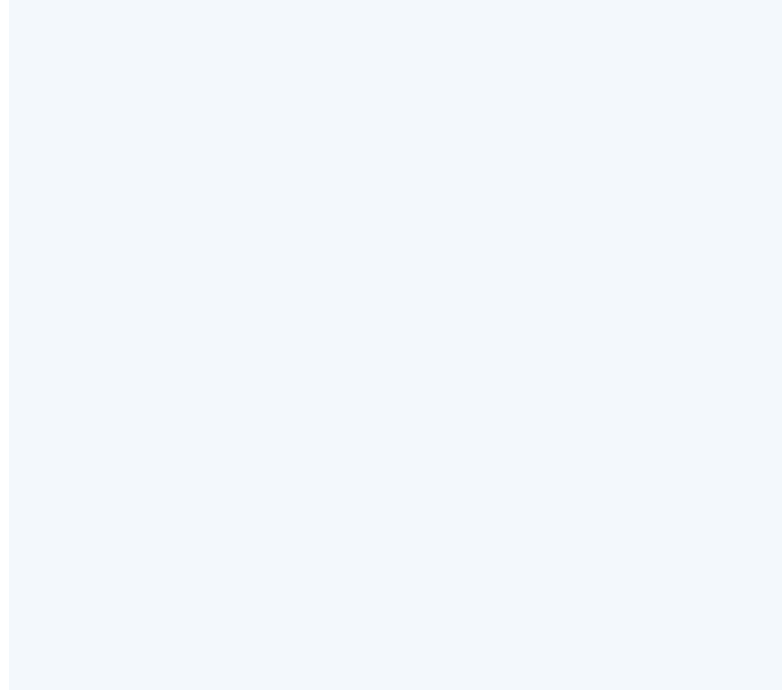
What is native?



The Anatomy of a Native App




Native User Interfaces



Architecting

Mobile Apps

A decorative geometric pattern on the right side of the slide, consisting of large, light gray triangles that form a larger triangular shape pointing to the right.



The Silo Approach

Build App
Multiple Times



iOS App

Objective-C
XCode



Android App

Java
Eclipse

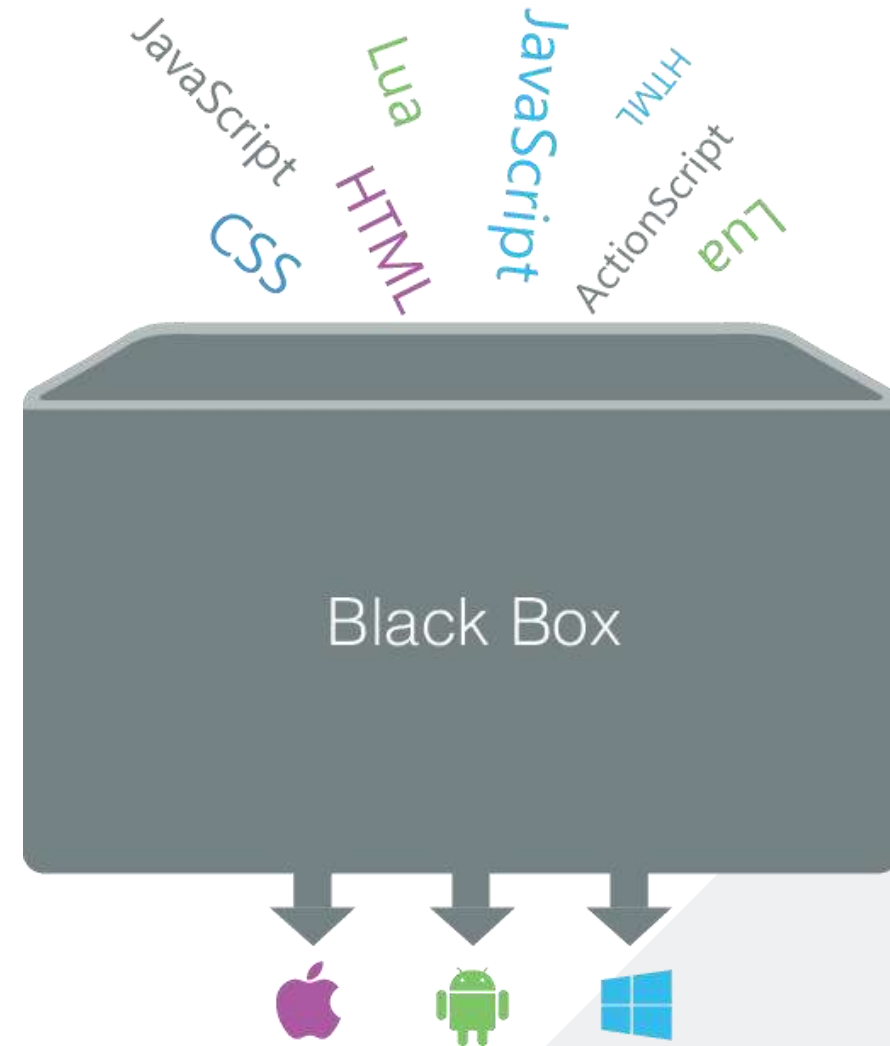


Windows App

C#
Visual Studio

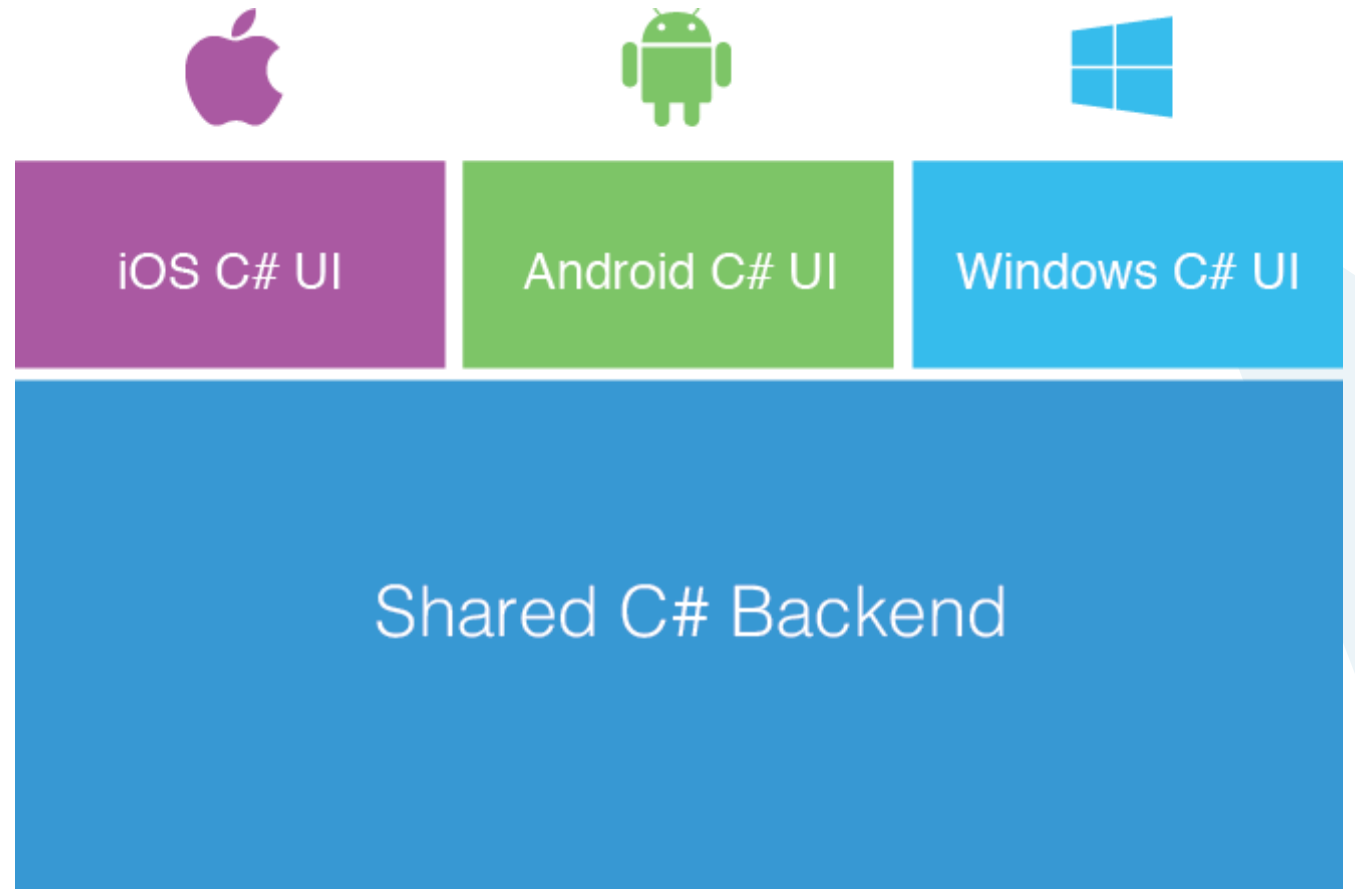
The Write-Once-Run-Anywhere Approach

Lowest Common
Denominator

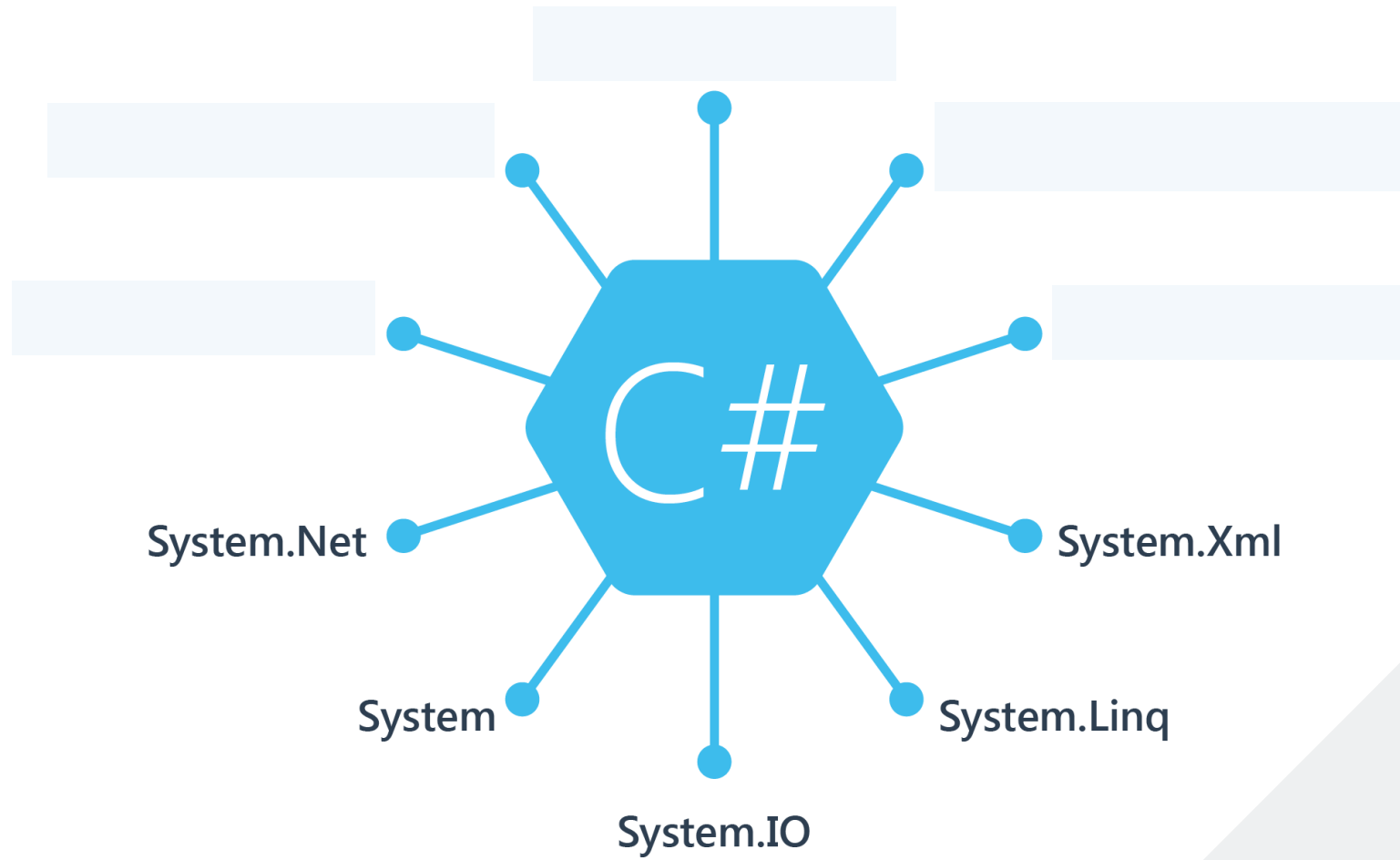


Xamarin's Unique Approach

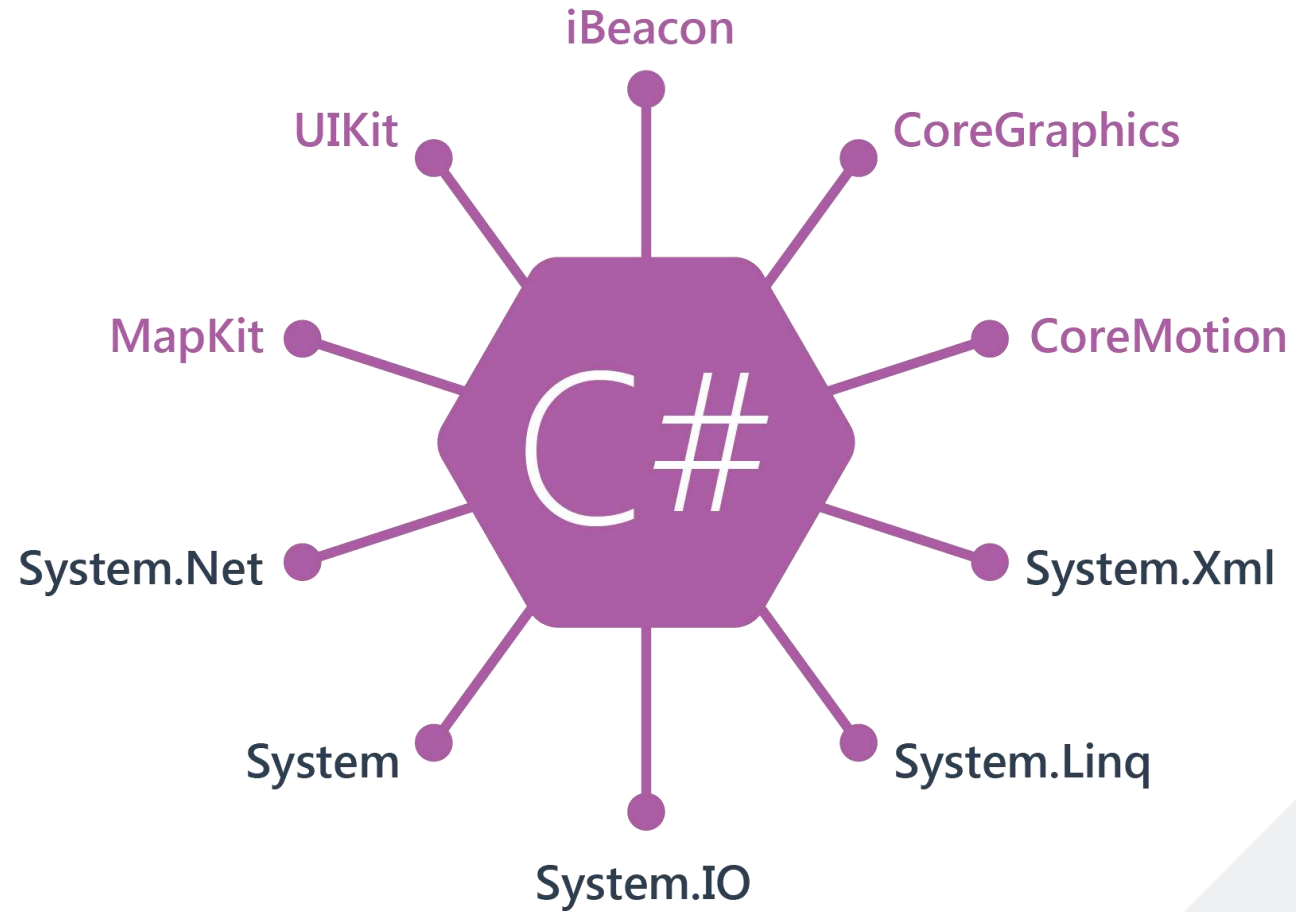
Native With
Code Sharing



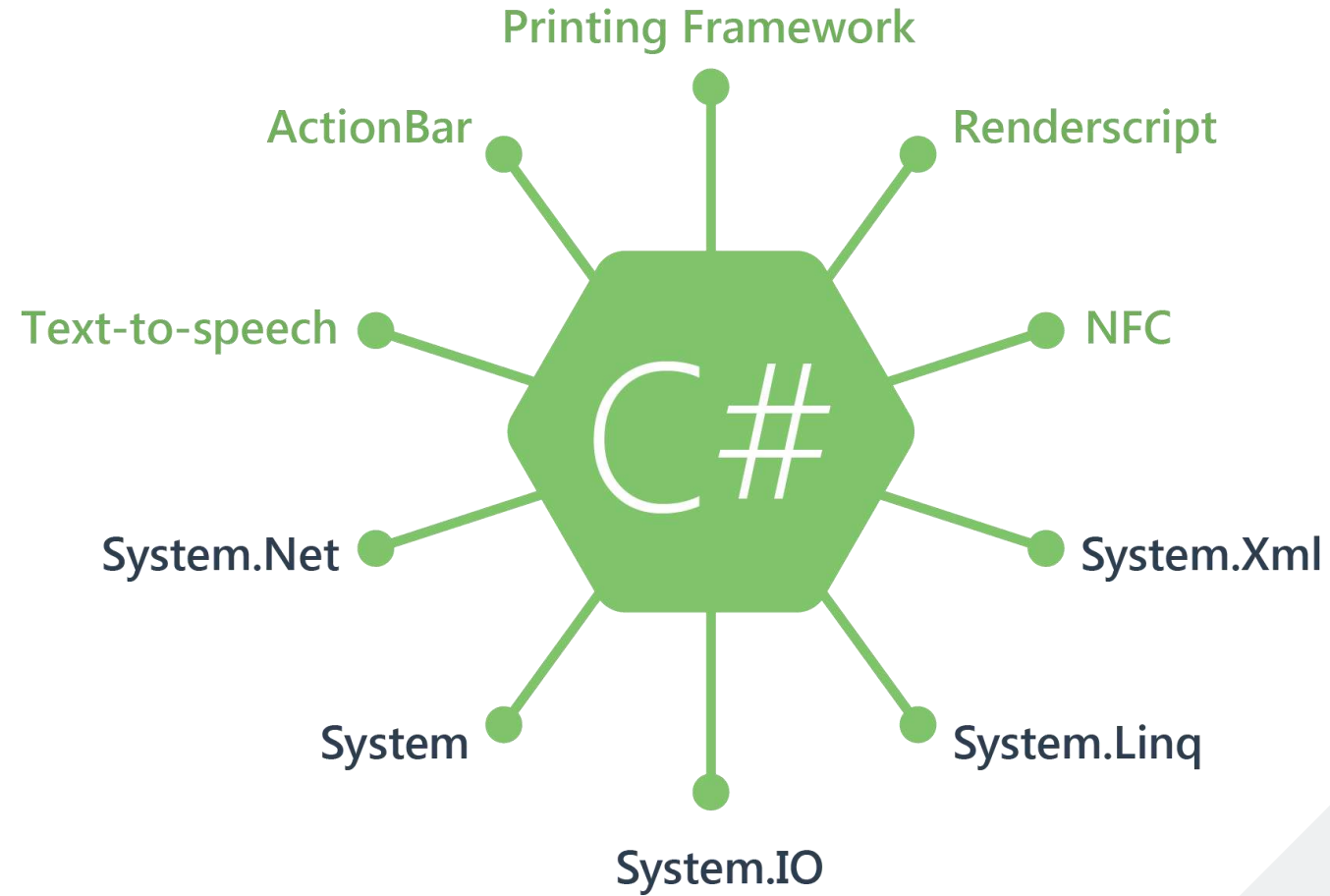
Windows APIs



iOS APIs | 100% Coverage



Android APIs | 100% Coverage





Anything you can do in Objective-C, Swift, or Java
can be done in C# with Xamarin using Visual Studio



Native Performance



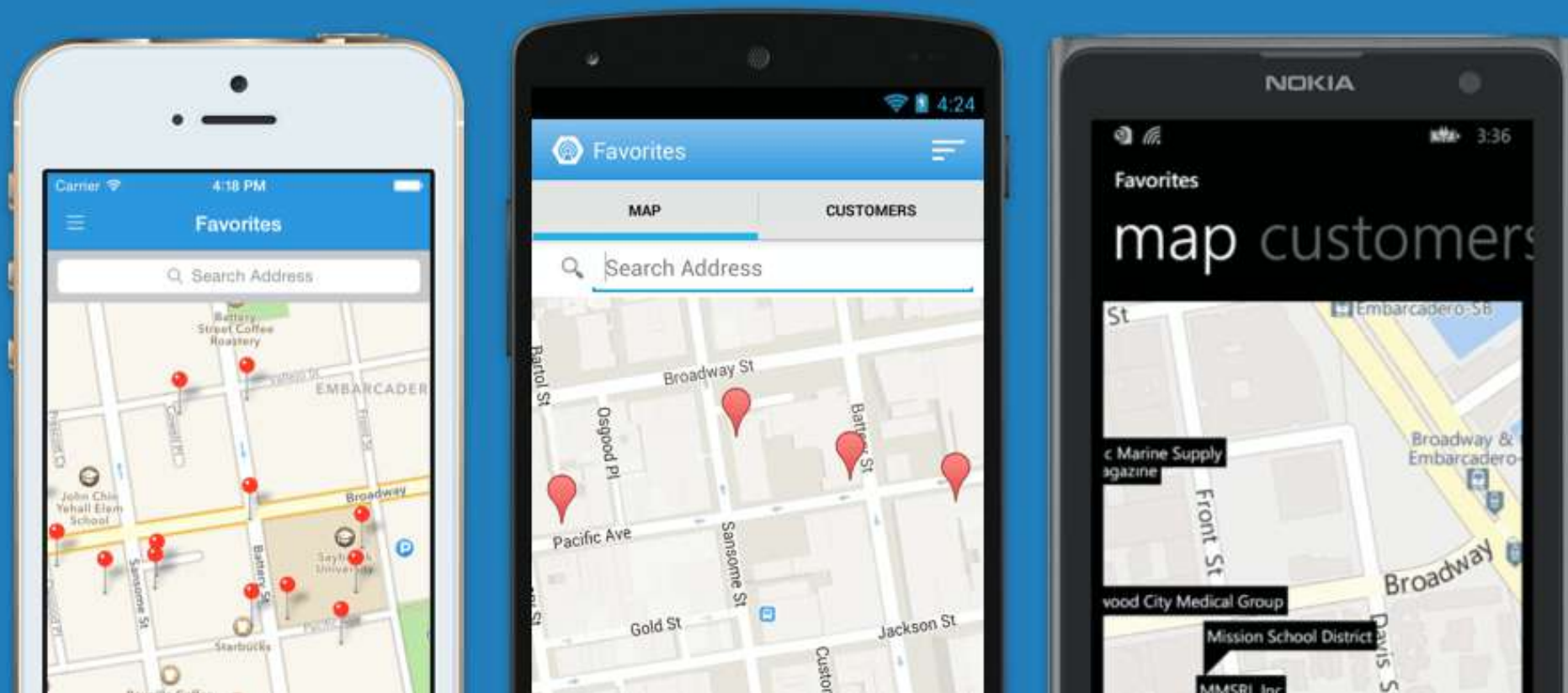
Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.



Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

Meet Xamarin.Forms

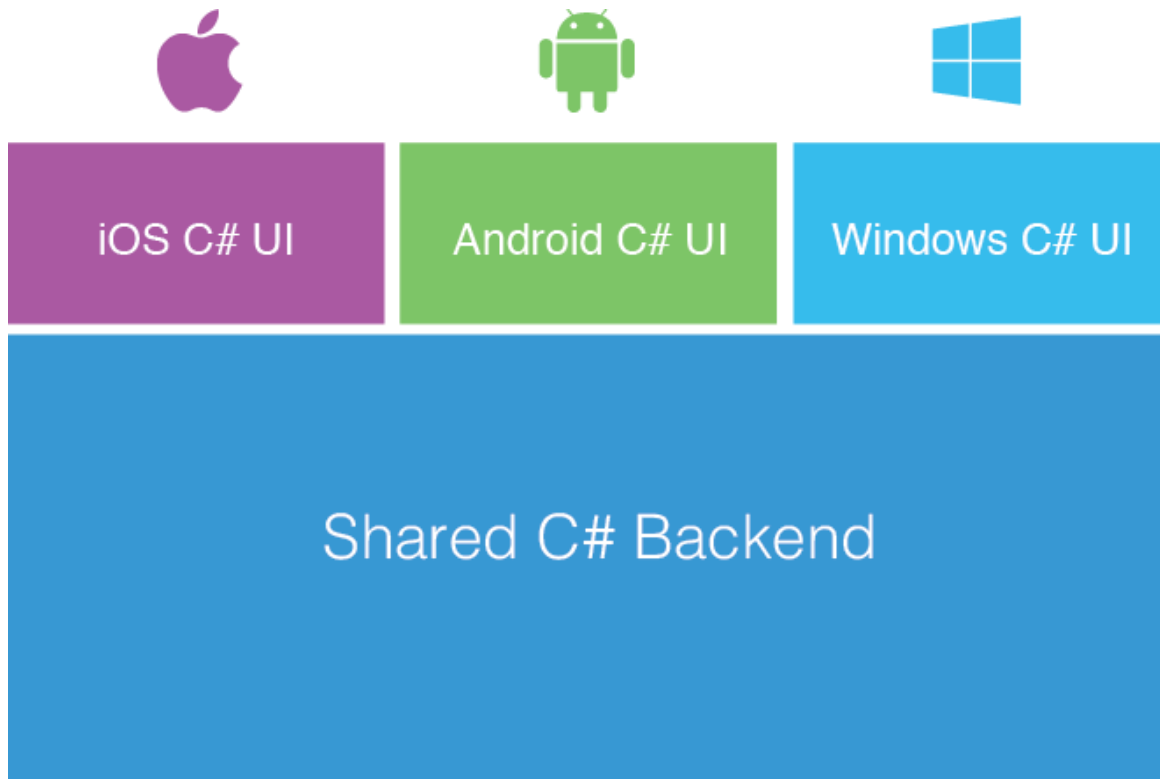
Build native UIs for iOS, Android and Windows Phone from a single, shared C# codebase.



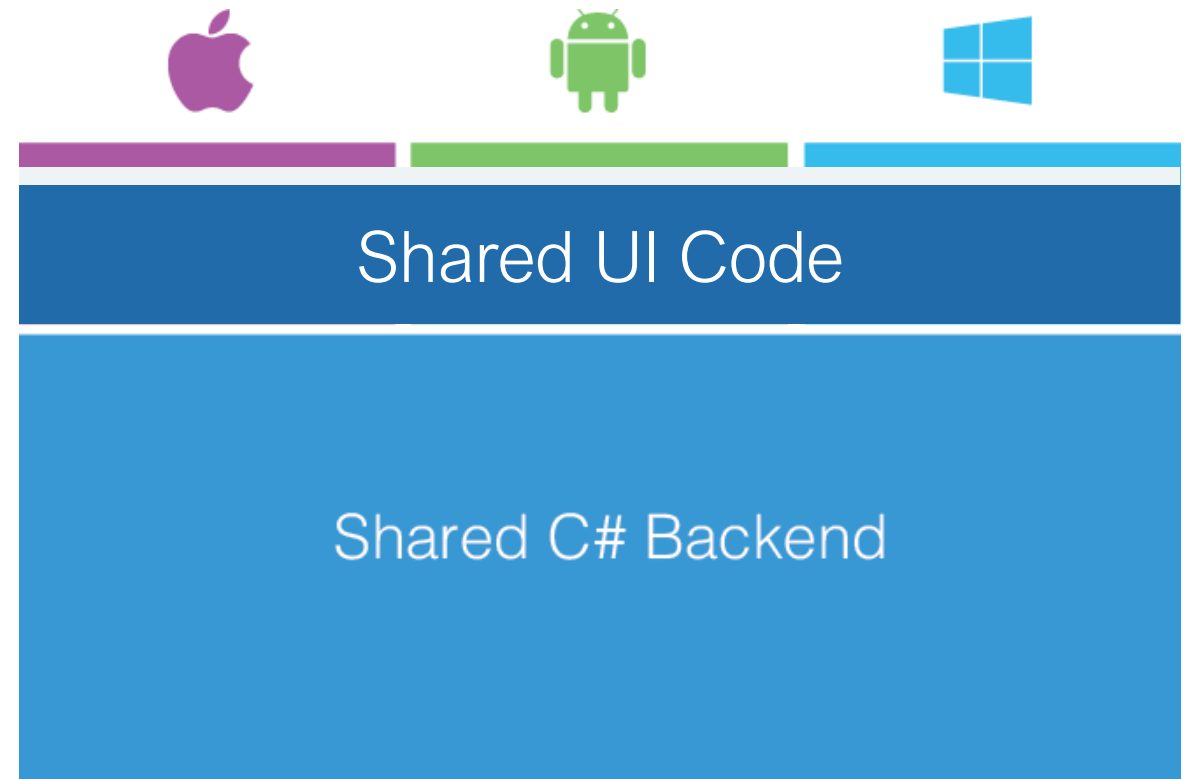
Xamarin + Xamarin.Forms



Traditional Xamarin approach



With Xamarin.Forms:
more code-sharing, native controls



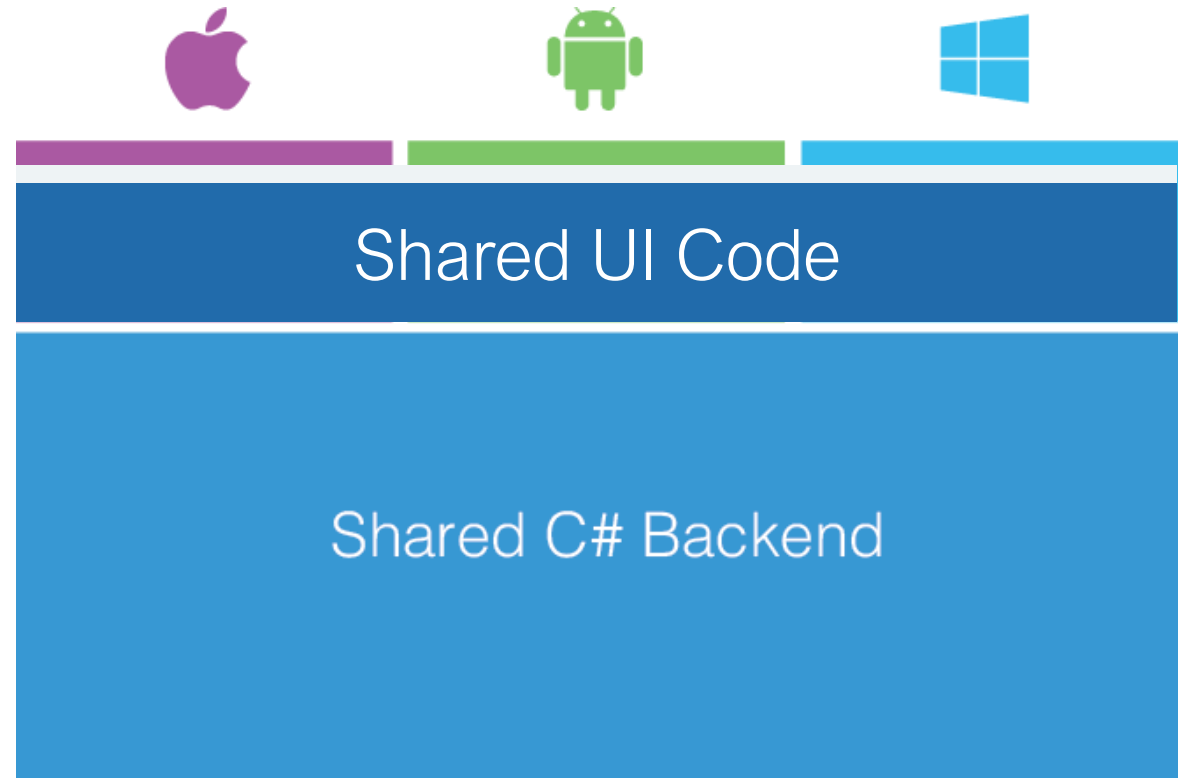
Xamarin + Xamarin.Forms



Quickly and easily build native user interfaces using shared code

Xamarin.Forms elements map to native controls and behaviors

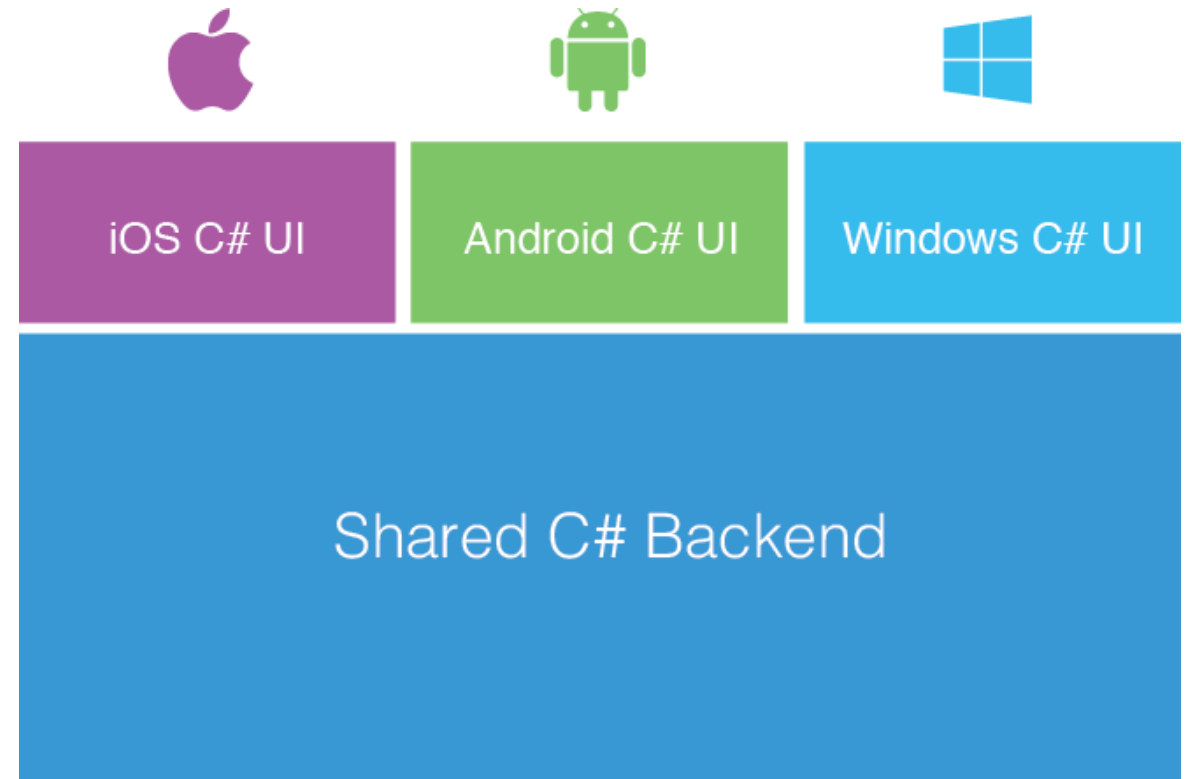
Mix-and-match Xamarin.Forms with native APIs



What's Included



- 40+ Pages, Layouts, and Controls
 - Build from code behind or XAML
- Two-way Data Binding
- Navigation
- Animation API
- Dependency Service
- Messaging Center



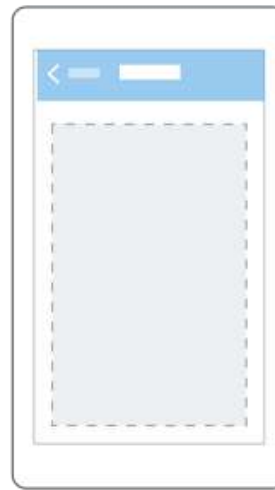
Pages



Content



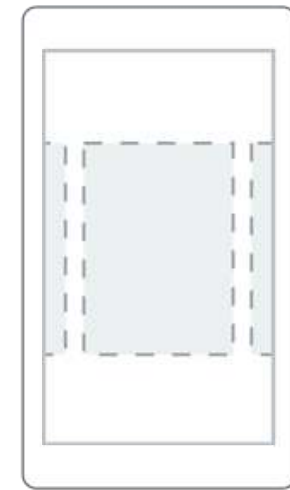
MasterDetail



Navigation

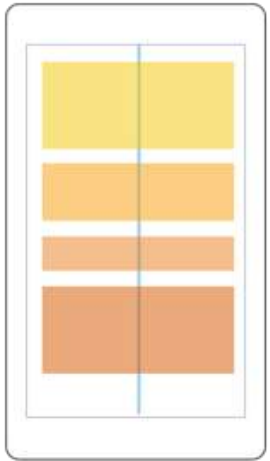


Tabbed

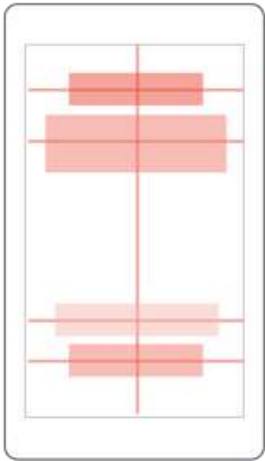


Carousel

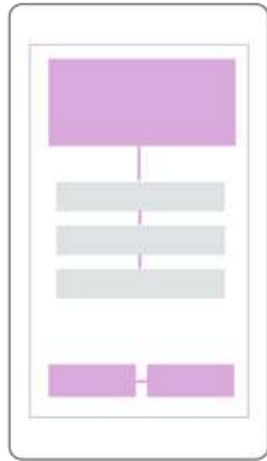
Layouts



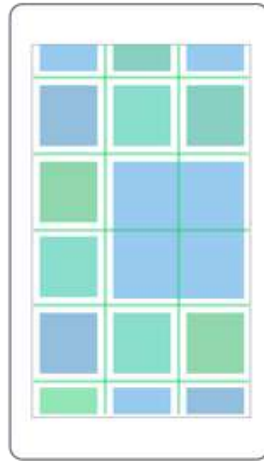
Stack



Absolute



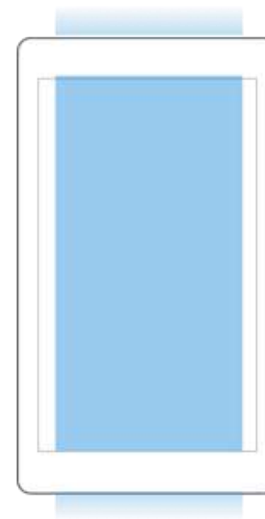
Relative



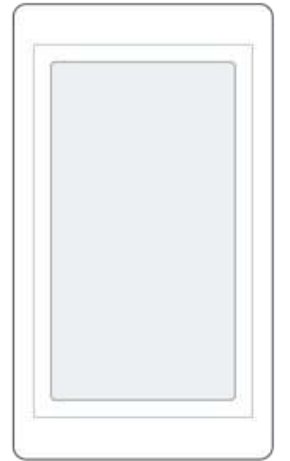
Grid



ContentView



ScrollView



Frame

Controls

ActivityIndicator

BoxView

Button

DatePicker

Editor

Entry

Image

Label

ListView

Map

OpenGLView

Picker

ProgressBar

SearchBar

Slider

Stepper

TableView

TimePicker

WebView

EntryCell

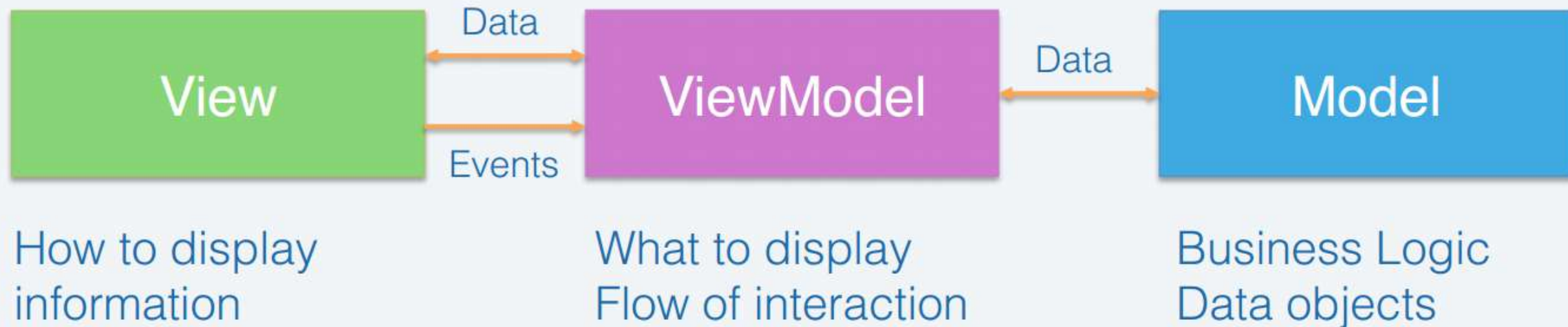
ImageCell

SwitchCell

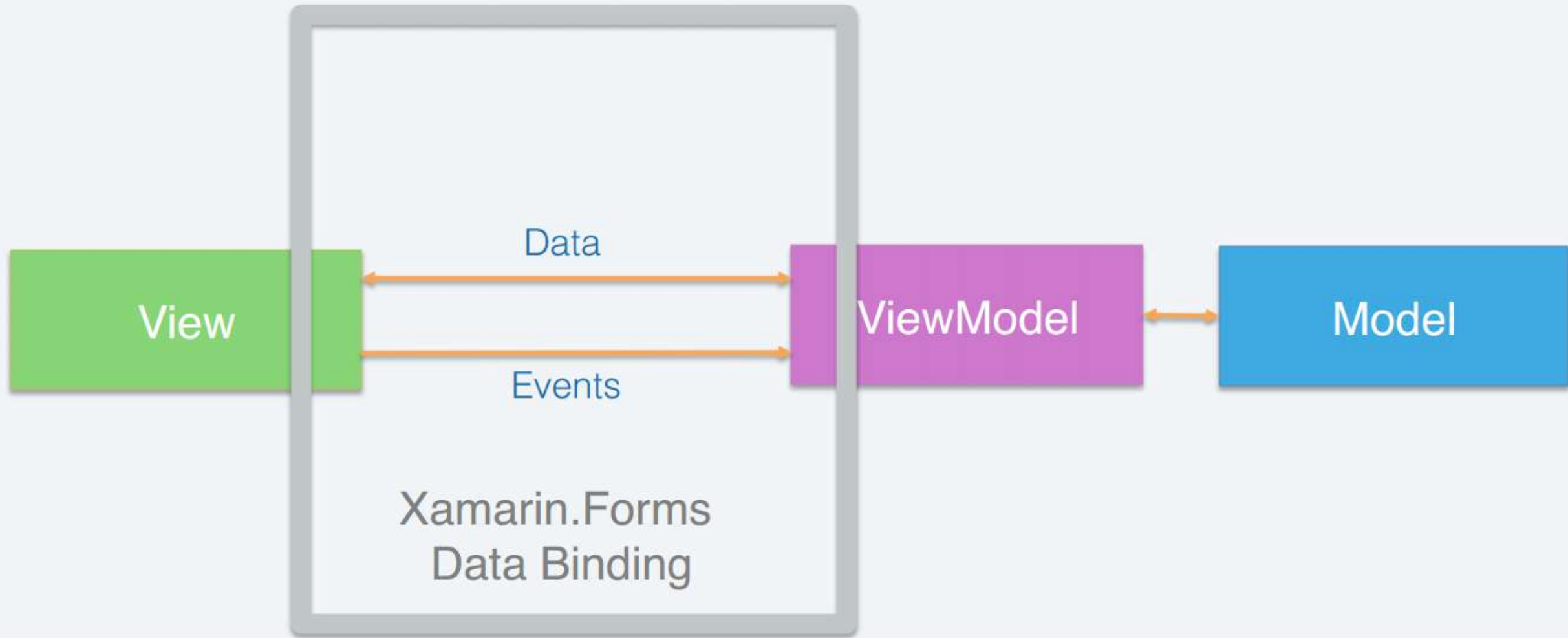
TextCell

ViewCell

Model-View-ViewModel



Model-View-ViewModel



DataBinding

- Xamarin support rich DataBindings mechanism.
- Support for INotifyPropertyChanged notifications.
- Declare Bindings in Code/XAML

```
var label = new Label() {VerticalOptions = LayoutOptions.Center};  
label.SetBinding(Label.TextProperty, new Binding("MyName"));
```

```
<Entry Placeholder="Please input your User Name"  
Text="{Binding UserName, Mode=TwoWay}"/>
```


- Used to Execute a method when an action is performs, such as button click.
- Ability to pass parameter
- Ability to have CanExecute

```
public interface ICommand
{
    //
    // Methods
    //
    bool CanExecute (object parameter);

    void Execute (object parameter);

    //
    // Events
    //
    event EventHandler CanExecuteChanged
}
```


Commands

- Command type is part of the Xamarin Forms framework (no need for the 3rd Party).

```
private Command _remindMeCommand;

public Command RemindMeCommand
{
    get
    {
        return _remindMeCommand ?? (_remindMeCommand = new Command(
            () =>
            {
                UserName = "Alon Fliess";
            }));
    }
}
```

```
<Button Text="Remind Me..." Command="{Binding RemindMeCommand}"/>
```


- Cross-platform animations
- Platform-specific animation APIs
- Async/Await API

box.to|

- ☒ FadeTo
- ☒ LayoutTo
- ☒ RelRotateTo
- ☒ RelScaleTo
- ☒ RotateTo
- ☒ RotateXTo
- ☒ RotateYTo

```
public Task  
FadeTo (  
    double opacity,  
    uint length = 250,  
    Easing easing = null  
)
```

Extension Method from
Xamarin.Forms.ViewExtensions

Login ViewModel

```
public class LoginViewModel : INotifyPropertyChanged
{
    private string username = string.Empty;
    public string Username
    {
        get { return username; }
        set { username = value; OnPropertyChanged ("Username"); }
    }

    private string password = string.Empty;
    public string Password
    {
        get { return password; }
        set { password = value; OnPropertyChanged ("Password"); }
    }

    public Command LoginCommand
    {
        get {
            return new Command (() => {
                //Log into Server here
            });
        }
    }
}
```


Login Page – Code Behind

```
public class LoginPage : ContentPage
{
    public LoginPage()
    {
        //set binding context
        this.BindingContext = new LoginViewModel ();

        //create UI & bind to properties
        var username = new Entry { Placeholder = "Username" };
        username.SetBinding (Entry.TextProperty, "Username");

        var password = new Entry { Placeholder = "Password", IsPassword = true };
        password.SetBinding (Entry.TextProperty, "Password");

        var loginButton = new Button {
            Text = "Login",
            TextColor = Color.White,
            BackgroundColor = Color.FromHex("77D065")
        };

        loginButton.SetBinding (Button.CommandProperty, "LoginCommand");

        //set main content of page
        Content = new StackLayout{
            VerticalOptions = LayoutOptions.Center,
            Padding = 50, Spacing = 10,
            Children = { username, password, loginButton }
        };
    }
}
```


Login Page – XAML

```
<?xml version="1.0" encoding="UTF-8" ?>
<ContentPage
  xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="LoginExampleForms.LoginPageXAML">
  <ContentPage.Content>

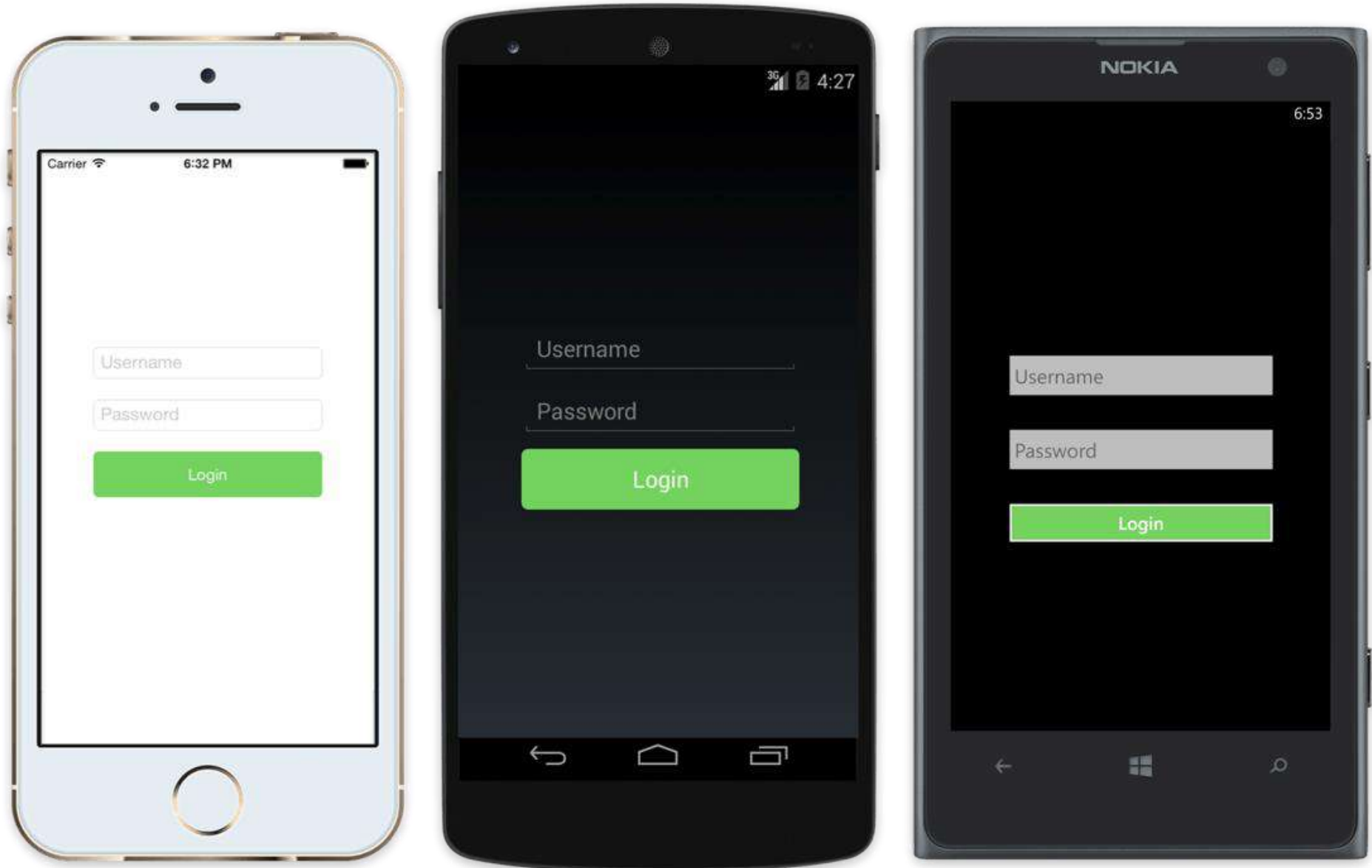
    <StackLayout VerticalOptions="Center" Padding="50" Spacing="10">

      <Entry Placeholder="Username" Text="{Binding Username}"/>
      <Entry Placeholder="Password" Text="{Binding Password}"/>
      <Button Text="Login"
        TextColor="#FFFFFF"
        BackgroundColor="#77D065"
        Command="{Binding LoginCommand}"/>

    </StackLayout>

  </ContentPage.Content>
</ContentPage>
```


Login Page



Summary

- Quickly and easily build native user interfaces using shared code
- Xamarin.Forms elements map to native controls and behaviors
- Mix-and-match Xamarin.Forms with native APIs

XAML Fundamentals

What is XAML?

- XML based language
- Enable separation of UI and behavior (code)
- Windows Phone related tools emit XAML
- XAML allows
 - Creation of objects
 - Setting of properties
 - Connection to events
 - Custom behaviors
- XAML cannot call methods directly

XAML vs. Code

- Anything that can be done in XAML can be done in code
 - But not vice versa
- XAML is usually shorter and more concise than the equivalent code
 - Thanks to type converters and markup extensions
- XAML should be used for initial UI
- Code will handle events and change items dynamically

Simple XAML Example

```
<Button xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
        Content="OK" />
```



```
Windows.UI.Xaml.Controls.Button b = new  
Windows.UI.Xaml.Controls.Button();  
b.Content = "OK";
```

- Visual Studio UI designer generates XAML on each control picked from the toolbox
- XAML Can be visually viewed in the UI designer



XAML Namespaces

- The default XAML namespace is assigned a value that is mapped to some of the runtime namespaces contain UI elements
- Other XAML namespaces may be mapped to custom namespaces and other runtime namespaces
- The “x” namespace is mapped to a special namespace, contains XAML parser specific types
- XAML namespace can be defined on each element level



XAML Example

```
<!--ContentPanel - place additional content here-->
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
    <Button x:Name="buttonOk"
        Width="200"
        Height="200"
        Content="OK"
        Click="buttonOk_Click" />
</Grid>
```



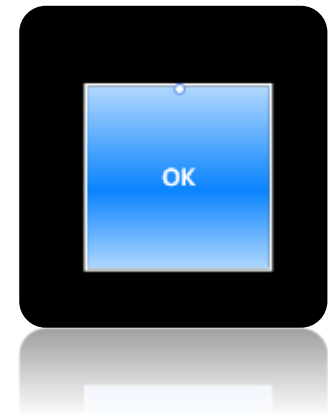
Elements and Attributes

- Elements with type names only designate object creation (via the default constructor)
- Attributes indicate property or event values
 - Event values are event handlers (methods) names
- Complex properties are designated using a **<Type.Property>** element



XAML Example

```
<!--ContentPanel - place additional content here-->
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
    <Button x:Name="buttonOk"
        Width="200"
        Height="200"
        Content="OK"
        Click="buttonOk_Click" >
        <Button.Background>
            <LinearGradientBrush EndPoint="0.5,1"
                                StartPoint="0.5,0">
                <GradientStop Color="#FFB2D9FF" Offset="0.004"/>
                <GradientStop Color="#FFB0D8FF" Offset="1"/>
                <GradientStop Color="#FF0A85FF" Offset="0.571"/>
            </LinearGradientBrush>
        </Button.Background>
    </Button>
</Grid>
```



XAML And Code Behind

- A root element, usually **Page** or **UserControl** classes, can have code behind file
- The name of the code behind file is correlated to the XAML file name
- For example: MainPage.xaml and MainPage.xaml.cs
- The code behind full class name is specified from XAML using the **x:Class** directive

```
<Page
  x:Class="UWPDemo.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:UWPDemo"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">
  <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
    <Image x:Name="image" HorizontalAlignment="Left" VerticalAlignment="Top" Stretch="Fill" Opacity="0.5"/>
    <Button x:Name="button" Content="Button" HorizontalAlignment="Left" Height="61" Width="127"/>
  </Grid>
</Page>
```




Child Elements

- Child elements (that are not property elements) can be one of
 - The **Content** property of the object
 - A property adorned with the attribute **Windows.UI.Xaml.Controls.ContentProperty**
 - Collection items
 - The object implements **ICollection** or **IDictionary**
 - A value that can be type-converted



Content Property

- A single property that is designated with the **ContentProperty** attribute on the type
- Allows shortening the markup

```
<Button Content="OK" >  
</Button>
```



```
<Button>  
    OK  
</Button>
```

```
<Button>  
    <Button.Content>  
        <Rectangle Fill="Blue"/>  
    </Button.Content>  
</Button>
```



```
<Button>  
    <Rectangle Fill="Blue"/>  
</Button>
```




Collection Items

► List (IList)

```
<ListBox>  
<ListBox.Items>  
<ListBoxItem Content="Item 1"/>  
<ListBoxItem Content="Item 2"/>  
</ListBox.Items>  
</ListBox>
```

► Dictionary (IDictionary)

```
<ResourceDictionary>  
<SolidColorBrush x:Key="br1" Color="Aqua" />  
<Rectangle x:Key="rc1" Fill="Brown" />  
</ResourceDictionary>
```


Summary of XAML Rules

- XML Element – create a new instance
- XML attribute – set a property or register an event
 - Type converter may execute
- **Type.Property** – set a “complex” property
- **ContentProperty** attribute – no need to specify **Type.Property**
- Property of type **IList** or **IDictionary**
 - Add child elements (XAML calls appropriate **Add** method)
 - Need a **x:Key** in case of a dictionary



Naming Elements

- Elements can be named using the **x:Name** XAML attribute
- The code-behind file will contain a field with that name
- Elements deriving from **FrameworkElement** contain a **Name** property that can be used in code to locate elements
 - **x:Name** and **Name** cannot be set on the same element



XAML Keywords

Keyword	Valid on	Meaning
x:Class	Root element	The class that derives from the element type
x:ClassModifier	Root element, must be used with x:Class	The class visibility (public by default)
x:FieldModifier	Element, must be used with x:Name	Visibility of the field created behind the element
x:Key	Element that its parent implements IDictionary	Key in a dictionary
x:Name	Element	The element's name, used for a field name for that element
X:Uid	Element	Identifies elements that should use localized resources



Mapping custom types to XAML namespaces

- You can define your own custom types in C# and then reference your custom types in XAML markup
- To use XAML for custom types - those that come from libraries other than the Windows Runtime core libraries:
 - You must declare and map a XAML namespace with a prefix
 - Use that prefix in element usages to reference the types that were defined in your library
 - You declare prefix mappings as **xmlns** attributes
- For example:
 - the attribute syntax to map a prefix **myTypes** to the namespace **myCompany.myTypes** is
 - **xmlns:myTypes="using:myCompany.myTypes"**
 - The representative element usage is: **<myTypes:CustomButton/>**

XAML Markup Extensions

- Represent some kind of "shortcut" that enables a XAML file to access a value or behavior that isn't simply declaring elements based on backing types
- In XAML attribute syntax, curly braces "{" and "}" indicate a XAML markup extension usage
- A XAML parser calls code that provides behavior for that particular markup extension
 - That code provides an alternate object or behavior result that the XAML parser needs
- Examples:
 - {x:Bind} {Binding} {StaticResource} {ThemeResource} {TemplateBinding} {RelativeSource} {CustomResource} {x:Null}

Markup Extension Example

```
<Canvas.Resources>
```

```
    <Style TargetType="Border" x:Key="PageBackground">
```

```
        <Setter Property="BorderBrush" Value="Blue"/>
```

```
        <Setter Property="BorderThickness" Value="5"/>
```

```
    </Style>
```

```
</Canvas.Resources>
```

```
...
```

```
<Border Style="{StaticResource PageBackground}">
```

```
...
```

```
</Border>
```


XAML and .NET Events

- XAML has a syntax for attaching event handlers to objects in the markup
- You specify the name of the event as an attribute name on the object where the event is handled
 - For the attribute value, you specify the name of an event-handler function that you define in code
- The XAML processor uses this name to create a delegate representation in the loaded object tree, and adds the specified handler to an internal handler list

```
<Button Click="showUpdatesButton_Click">Show updates</Button>
```


Summary

- XAML is mainly used to create a Windows app user interface
- It declaratively allows object creation, property and event assignment
- A code-behind file will usually contain the procedural logic
- Sharing with designers is easier
- Tools such as Expression Blend generate XAML that is immediately usable

Event Processing

Azure Stream Analytics



Real time event processing

➤ **Uncover real time insights**

- Perform real time analytics across multiple streams

➤ **Rapid Deployment**

- Use simple SQL syntax, auto distributed for scale

➤ **Mission critical reliability**

- Fully managed, low latency, high throughput

➤ **Create real time alerts**

- Flag alerts and alarms for attention

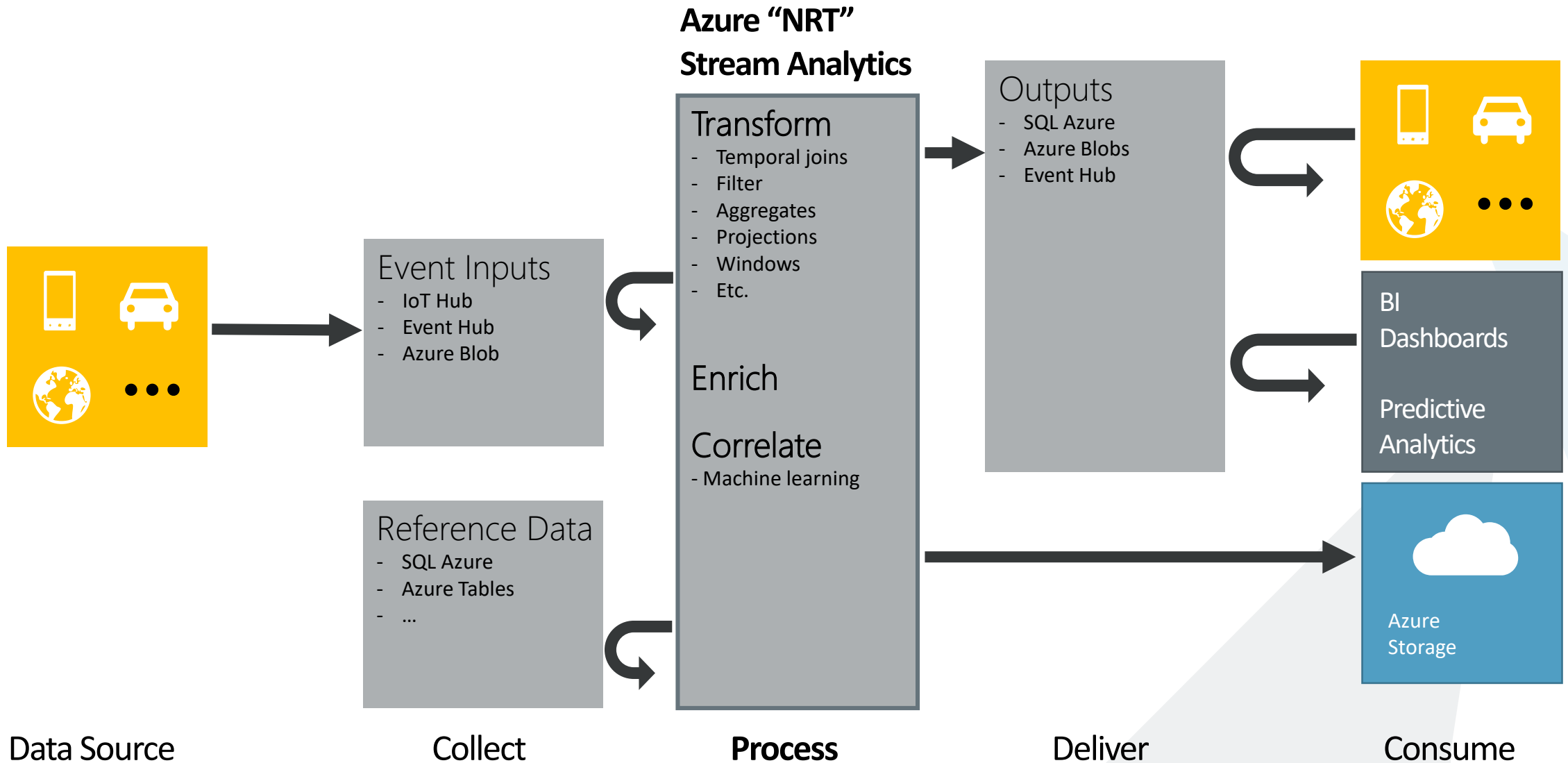
➤ **High volume**

- Analyze millions of data points per second

➤ **Highly scalable**

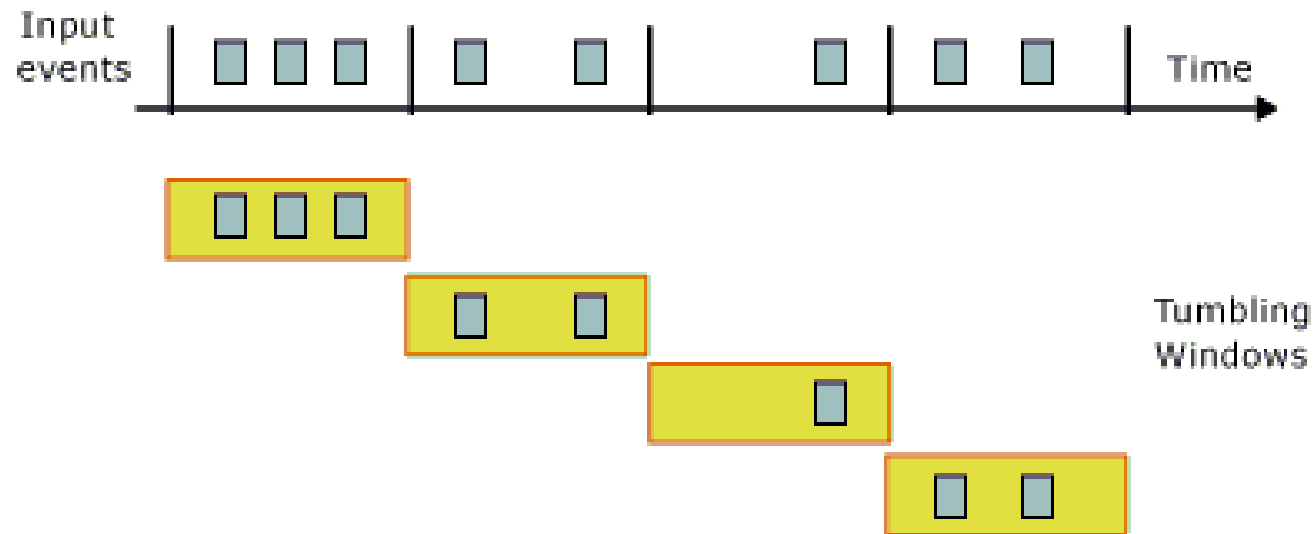
- Enterprise grade, predictable solution.

Streaming Architecture



Stream Analytics - Tumbling Windows

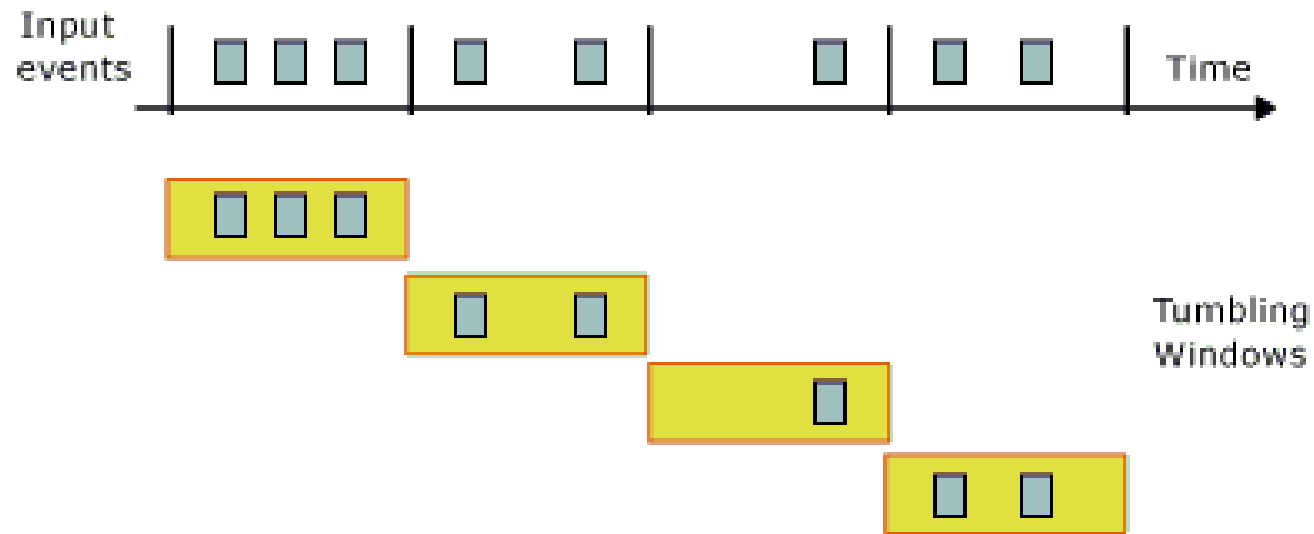
- How many vehicles entered each toll booth every 5 minutes?



```
SELECT TollId, COUNT(*) FROM EntryStream  
GROUP BY TollId, TumblingWindow(minute,5)
```

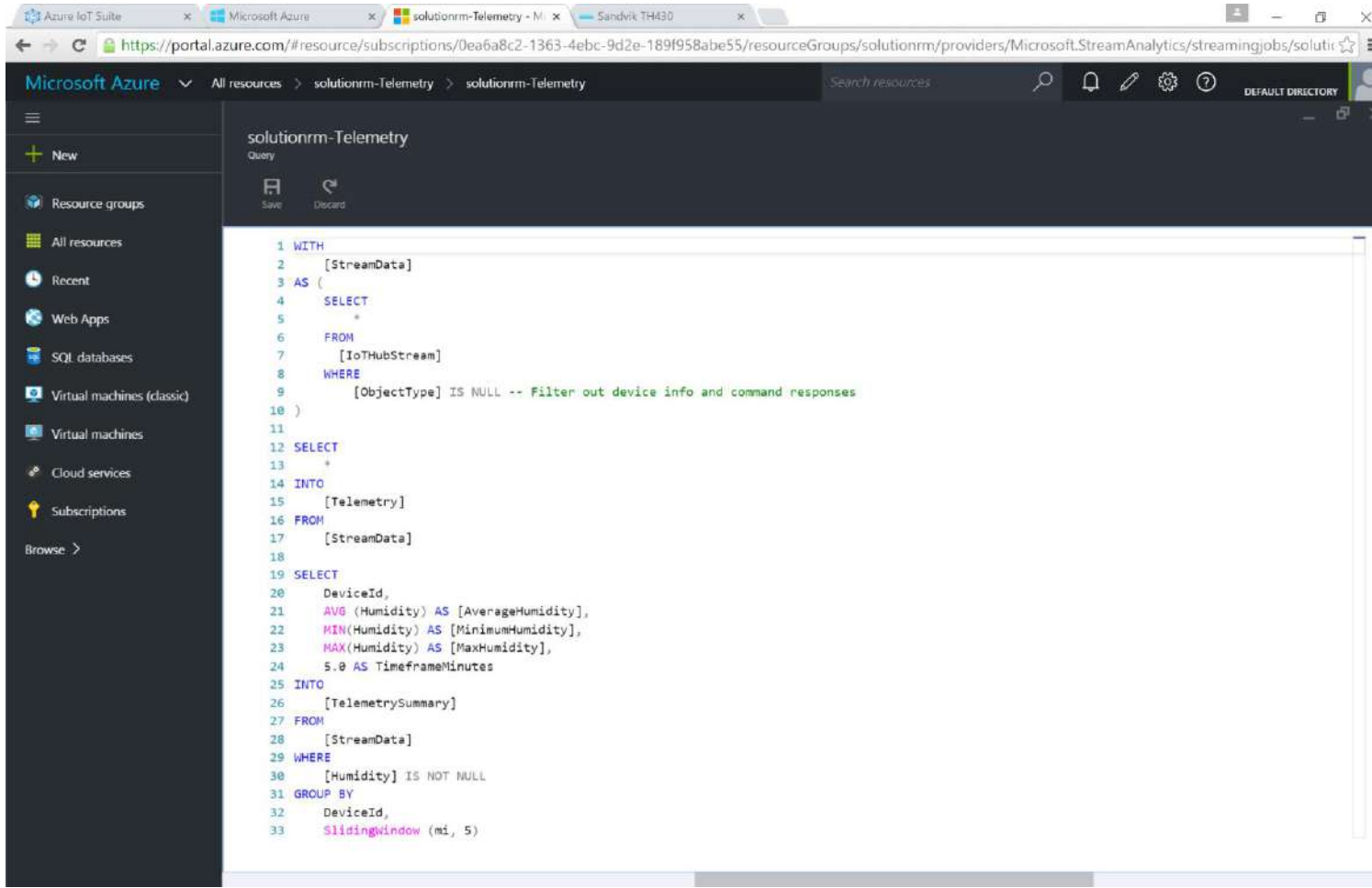

Stream Analytics - Tumbling Windows

How many vehicles entered each toll booth every 5 minutes?



```
SELECT TollId, COUNT(*) FROM EntryStream  
GROUP BY TollId, TumblingWindow(minute,5)
```


Add or edit jobs using simple interface



Rule based interface

Simple implementation and rule development using ASA UI.

Multi-channel

Analyze multiple channels of information simultaneously, in real time.

Data Visualization

Power BI



Data visualization with PowerBI

Highly accessible analytics

Cloud based dashboard and analytics tool,

360° view of business KPI's

Customize dashboards to address concerns and performance metrics.

Cross platform support

View data via web platform, on any device

Pre-built dashboards

Utilize standard dashboards for rapid deployment, based on popular solution demands.

Real time capabilities

Ingest, analyze and display data as it happens

Secure access

Secure, live communication with data source

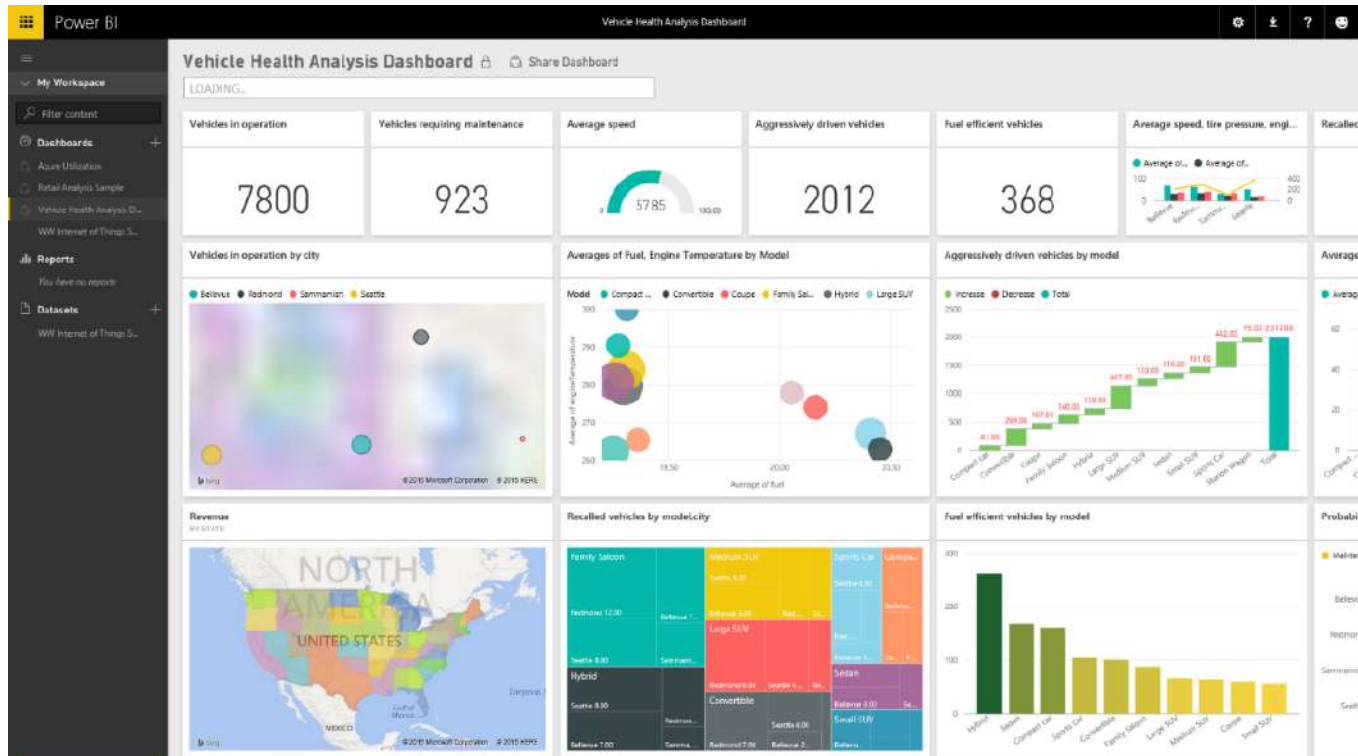
Query data

Intuitive, natural language query tool

Integrated systems

Integrate with other business systems and enrich device data with intelligence from other business systems, eg: CRM, ERP

Data visualization with PowerBI



Rich visuals

Standard and custom graphing options

Custom dashboards

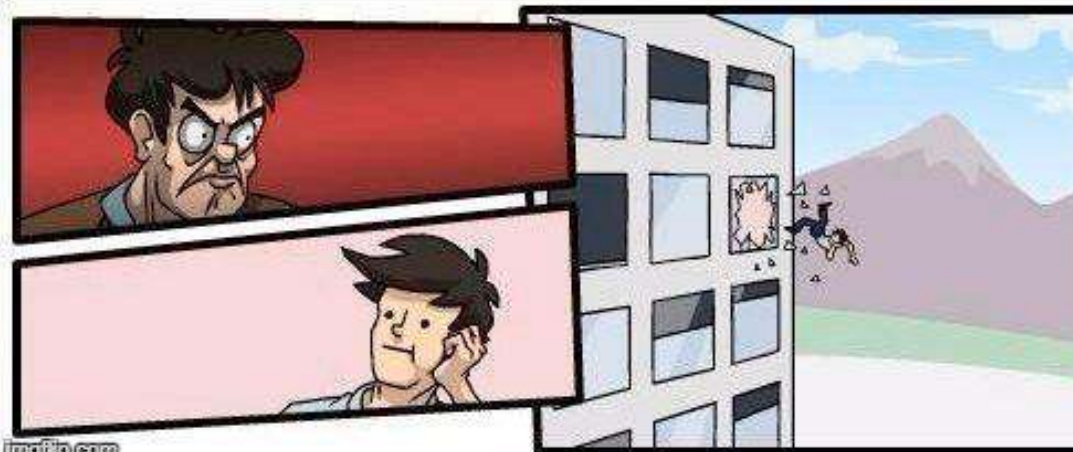
Build heat maps and visually track data

External data

Integrate external data feeds to add value to device data, or pull in external information such as weather or market information.

Artificial Intelligence





WHAT IF I TOLD YOU

**THAT ARTIFICIAL INTELLIGENCE
CAN ACTUALLY BE EASY?**

INTELLIGENCE AS A SERVICE?



SHUT UP AND TAKE MY MONEY!

The background of the slide features two hot air balloons floating in a bright blue sky filled with soft, white clouds. The balloon on the left is smaller and has a blue and white checkered pattern. The balloon on the right is larger and has a warm-toned pattern of orange, yellow, and red. Both balloons have small baskets hanging from them.

Microsoft Cognitive Services

Microsoft Cognitive Services

Give your apps a human side



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, & 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

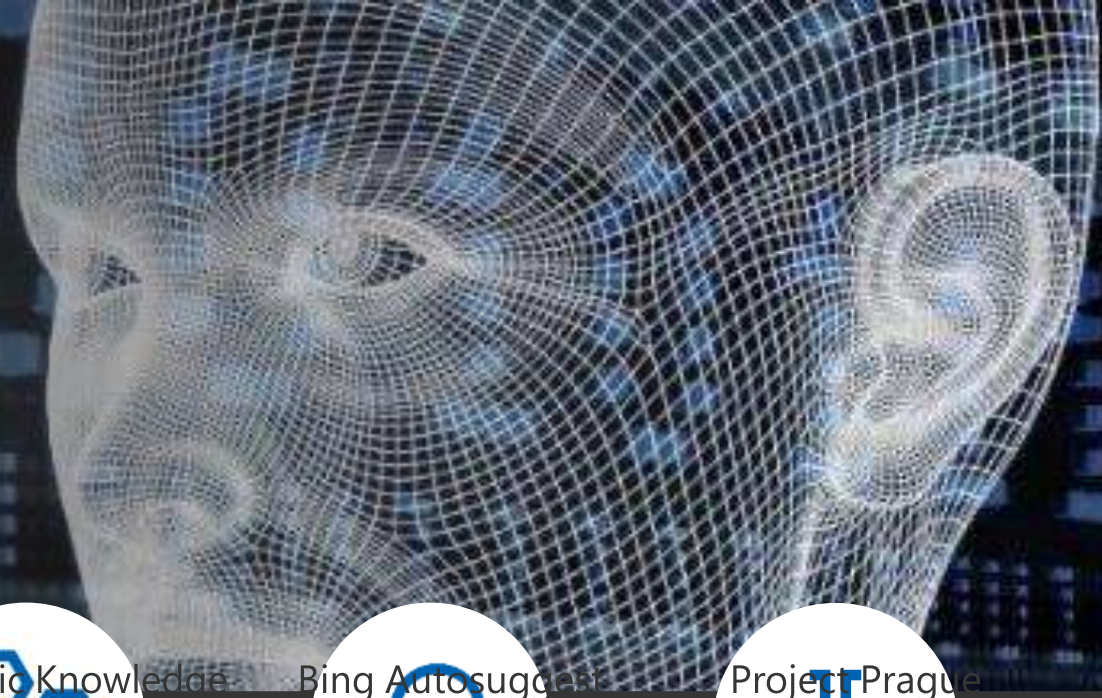


Labs

An early look at emerging Cognitive Services technologies: discover, try & give feedback on new technologies before general availability

Microsoft Cognitive Services

Give your apps a human side



Computer Vision
Content Moderator
Emotion
Vision
Video
Video Indexer
Custom Vision Service

Bing Speech
Speaker
Recognition
Translator Speech
Speech
Custom Speech
Service

Bing Spell Checker
Linguistic Analysis
Text Analytics
Translator Text
Language
Web Language
Model
Language
Understanding

Academic Knowledge
Entity Linking
Knowledge
Exploration
Knowledge
Recommendations
QnA Maker
Custom Decision
Service

Bing Autosuggest
Bing Image Search
Bing News Search
Bing Video Search
Search
Bing Web Search
Bing Entity Search
Bing Custom
Search

Project Prague
(gesture)
Project Cuzco (events)
Project Johannesburg
(routing)
Labs
Project Nanjing
(isochrones)
Project Abu Dhabi
(distance matrix)
Project Wollongong
(location)

Microsoft Cognitive Services

Give your apps a human side



Vision

Computer Vision
Content Moderator
Emotion
Face
Video
Video Indexer



Speech

Bing Speech
Speaker Recognition
Translator Speech



Language

Bing Spell Check
Linguistic Analysis
Text Analytics
Translator Text
Web Language Model



Knowledge

Academic Knowledge
Entity Linking
Knowledge Exploration
Recommendations
QnA Maker



Search

Bing Autosuggest
Bing Image Search
Bing News Search
Bing Video Search
Bing Web Search
Bing Entity Search



Labs

Project Prague (gesture)
Project Cuzco (events)
Project Johannesburg (routing)
Project Nanjing (isochrones)
Project Abu Dhabi (distance matrix)
Project Wollongong (location)

CUSTOMIZATION

Custom Vision Service

Custom Speech Service

Language Understanding

Custom Decision Service

Bing Custom Search

Why Microsoft Cognitive Services?

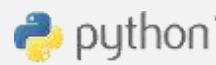
Easy

Roll your own with REST APIs
Simple to add: just a few lines of code required



Flexible

Integrate into the language and platform of your choice
Breadth of offerings helps you find the right API for your app
Bring your own data for your custom experience























Tested

Built by experts in their field from Microsoft Research, Bing, and Azure Machine Learning
Quality documentation, sample code, and community support



A variety of real-world applications

Vision 	Speech 	Language 	Knowledge 	Search 								
 What is in the image?	 Give me directions to the nearest local branch.	 Play today's customer call recording.	 Top publications in customer lifecycle trends?	 Search for 'fraud prevention'								
Computer Vision	Bing Speech	Language Understanding	Knowledge Exploration	Bing News Search								
 <table border="1" data-bbox="104 1079 550 1300"><tr><td>Category</td><td>People; 5 faces</td></tr><tr><td>Adult/Racy?</td><td>False/False</td></tr><tr><td>Dominant colors</td><td></td></tr><tr><td>Accent color</td><td></td></tr></table>	Category	People; 5 faces	Adult/Racy?	False/False	Dominant colors		Accent color		 →  Convert spoken audio to text Convert text to spoken audio Extract intent of user	Natural Language Processing <div>Intent: PlayCall Content: Customer# DateTime.date: today</div>  Now Playing 11/29/2016 Customer Call	Here are the top results: Customer Relationship Management – 5 Key Trends for 2014 CRM Oct 28, 2015 – Here are FIVE key trends in 2014 that would help marketers in rolling ... Of late, marketers are looking at customer lifecycle management (CLM) Predictive Customer Lifecycle Management (CLM) The purpose of Customer Life-cycle Management (CLM) is to maximize both customer retention and Predictive trend analysis provides business visibility. Trends 2016: The Future of Customer Service Jan 5, 2016 – The top 10 customer service trends for 2016 that North American Consumer Language Around Customer Lifecycles in the Banking Industry View PDF	Here is what I found:  Information Communications Media Market News It also investigates the top three expected Fraud Detection and Prevention programs, in terms of demand in key markets...  The Big Question: In-House or Outsourced Fraud Protection? First, let's point out that there is not one absolute answer—there are "pros" and "cons" to each. Those who favor in-house...  How to Protect Your Business from Online Fraud this Holiday Season Michael heads fraud prevention tool. Online and mobile shopping are expected to continue growing apace...
Category	People; 5 faces											
Adult/Racy?	False/False											
Dominant colors												
Accent color												

How do I use them?

```
using Microsoft.ProjectOxford.Vision;  
using Microsoft.ProjectOxford.Vision.Contract;
```

```
AnalysisResult analysisResult;  
var features = new VisualFeature[] { VisualFeature.Tag  
VisualFeature.Description };
```

```
using (var fs = new FileStream(@"C:\Vision\Sample.jpg",  
{  
    analysisResult = await visionClient.AnalyzeImageAsy  
})
```

```
POST https://api.projectoxford.ai/vision/v1.0/analyze?  
&subscription-key=<Your subscription key>
```

```
{  
  "tags": [  
    { "name": "outdoor",  
      "score": 0.976 },  
    { "name": "bird",  
      "score": 0.95 } ],  
  "description":  
    { "tags":  
      [ "outdoor", "bird" ],  
      "captions": [  
        { "text": "partridge  
          in a pear tree",  
          "confidence": 0.96 }  
      ]  
    }  
}
```


Vision



Vision



Computer Vision API

Distill actionable information from images



Face API

Detect, identify, analyze, organize, and tag faces in photos



Emotion API

Personalize experiences with emotion recognition



Video API

Analyze, edit, and process videos within your app



Content Moderator

Machine-assisted moderation of text and images, augmented with human review tools



Custom Vision Service

Customizable web service that learns to recognize specific content in imagery



Video Indexer

Process and extract smart insights from videos

Computer Vision API

Analyze an image

Understand content within an image

OCR

Detect and recognize words within an image

Generate thumbnail

Scale and crop images, while retaining key content

Recognize celebrities

Thanks to domain specific models, ability to recognize 200K celebrities from business, politics, sports, and entertainment around the world



Analyze image

Type of image

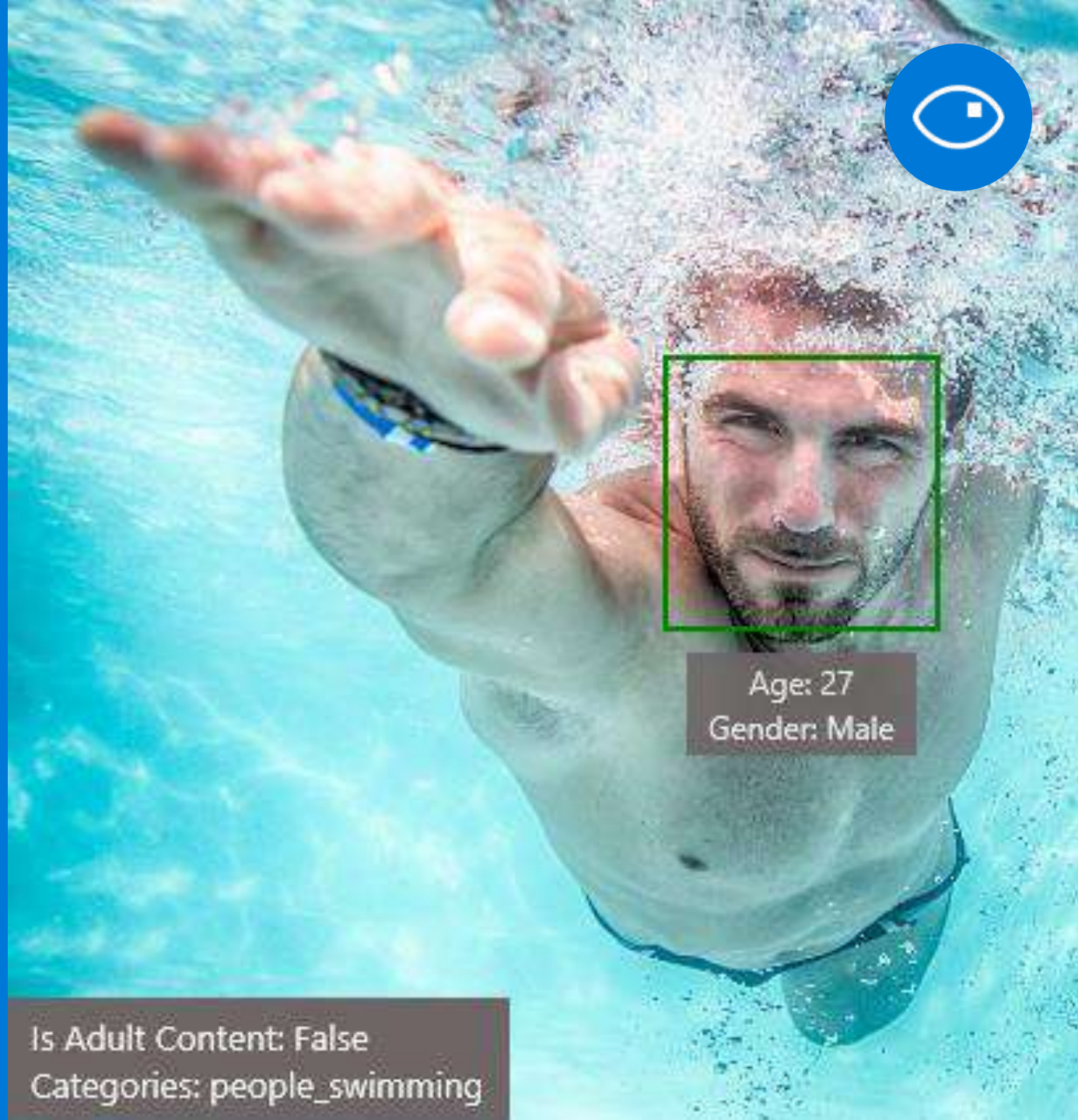
Clip Art Type	0 Non-clipart
Line Drawing Type	0 Non-Line Drawing
Black & White Image	False

Content of image

Categories	[{ "name": "people_swimming", "score": 0.099609375 }]
Adult Content	False
Adult Score	0.18533889949321747
Faces	[{ "age": 27, "gender": "Male", "faceRectangle": { "left": 472, "top": 258, "width": 199, "height": 199 } }]

Image colors

Dominant Color Background	White
Dominant Color Foreground	Grey
Dominant Colors	White
Accent Color	



Age: 27
Gender: Male

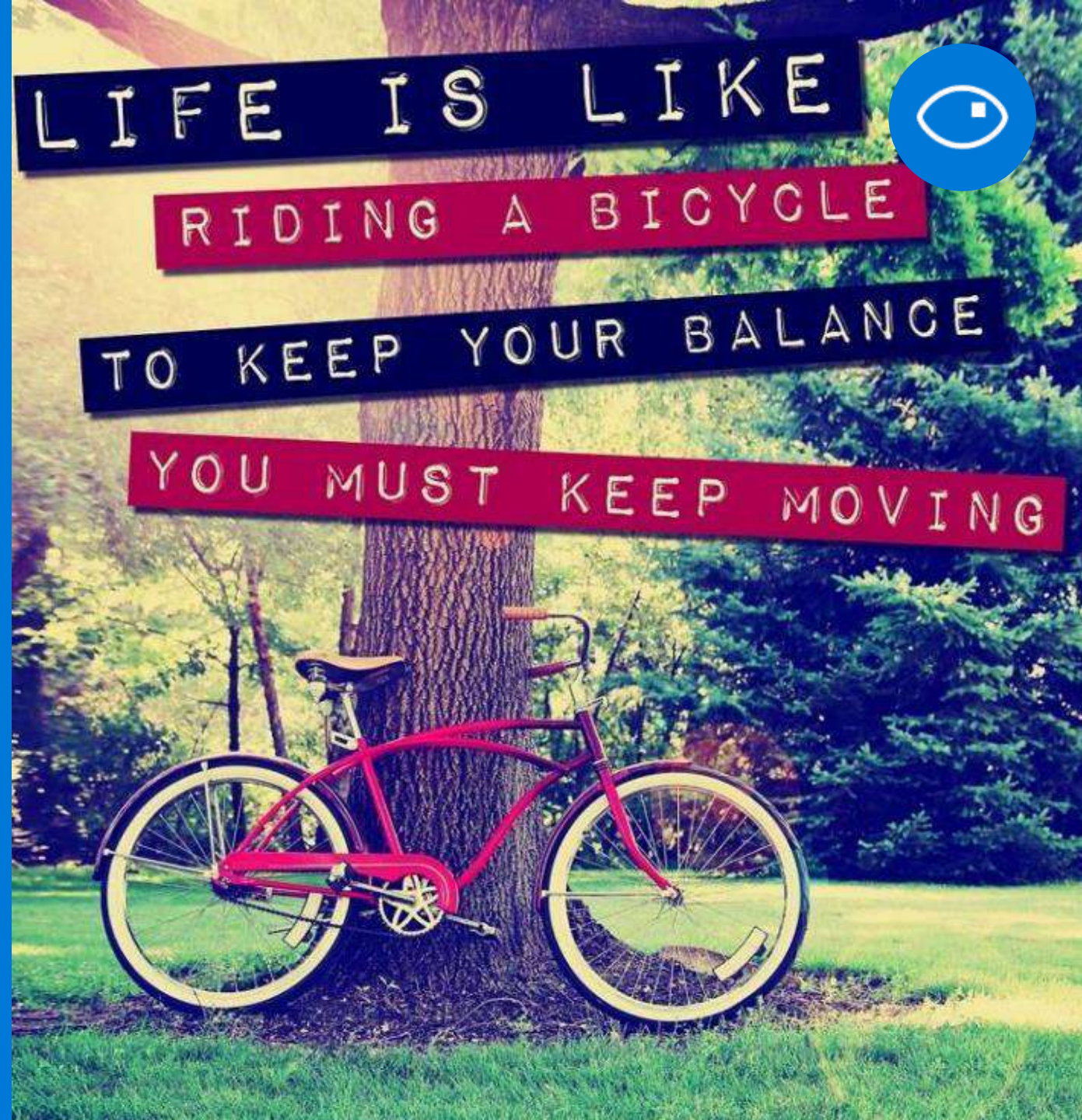
Is Adult Content: False
Categories: people_swimming

OCR

JSON:

```
{
  "language": "en",
  "orientation": "Up",
  "regions": [
    {
      "boundingBox": "41,77,918,440",
      "lines": [
        {
          "boundingBox": "41,77,723,89",
          "words": [
            {
              "boundingBox": "41,102,225,64",
              "text": "LIFE"
            },
            {
              "boundingBox": "356,89,94,62",
              "text": "IS"
            },
            {
              "boundingBox": "539,77,225,64",
              "text": "LIKE"
            }
          ]
        }
      ]
    }
  ]
}
```

...



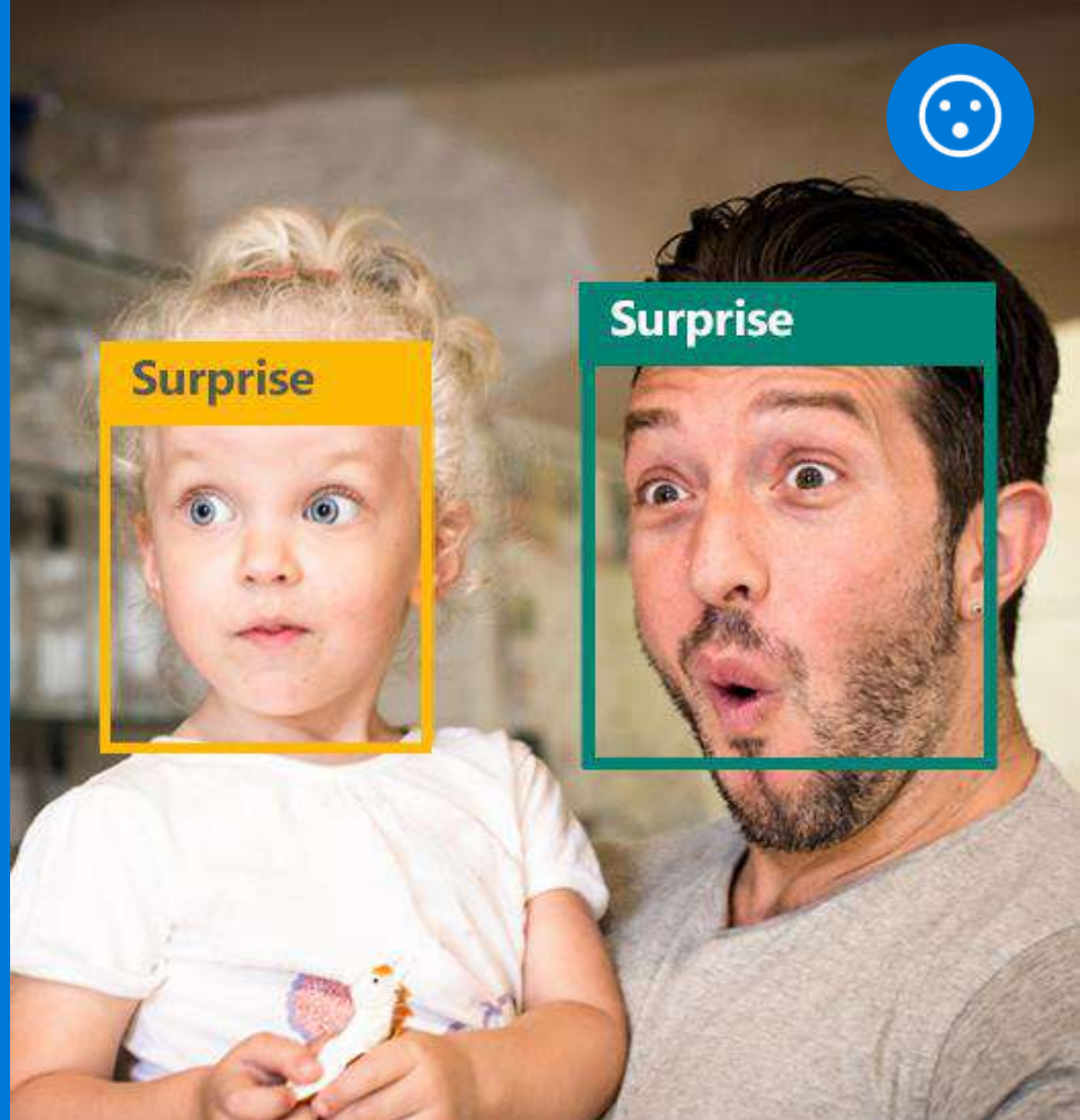
Emotion API

Face detection

```
"faceRectangle": {"width": 193,  
  "height": 193,  
  "left": 326,  
  "top": 204} ...
```

Emotion scores

```
"scores": { "anger": 5.182241e-8,  
  "contempt": 0.0000242813,  
  "disgust": 5.621025e-7,  
  "fear": 0.00115027453,  
  "happiness": 1.06114619e-8,  
  "neutral": 0.003540177,  
  "sadness": 9.30888746e-7,  
  "surprise": 0.9952837}
```



Face API

Face detection

Detect faces and their attributes within an image

Face verification

Check if two faces belong to the same person

Similar face searching

Find similar faces within a set of images

Face grouping

Organize many faces into groups

Face identification

Search which person a face belongs to



Face API

Detection

```
"faceRectangle": {"width": 193, "height": 193,  
"left": 326, "top": 204}  
...
```

Feature attributes

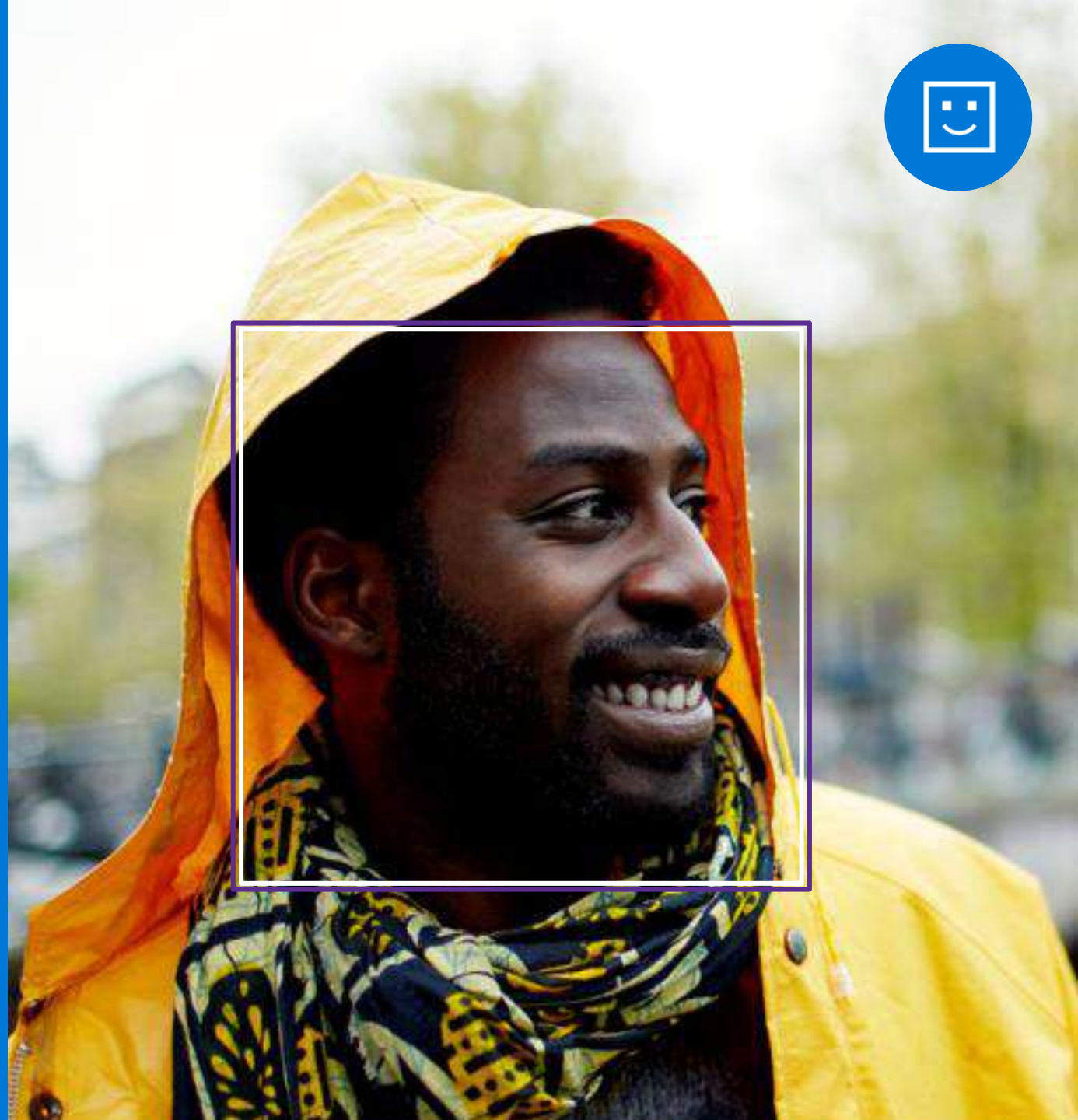
```
"attributes": { "age": 42, "gender": "male",  
"headPose": { "roll": "8.2", "yaw": "-37.8",  
"pitch": "0.0" }}
```

Grouping



Identification

Jasper Williams



Demo

Face and Emotion: <http://microsoft.com/cognitive>
Intelligent Kiosk: <http://aka.ms/kioskapp>

Video Indexer

Unlock video insights

Upload your video and go

Start turning your video into insights right away.

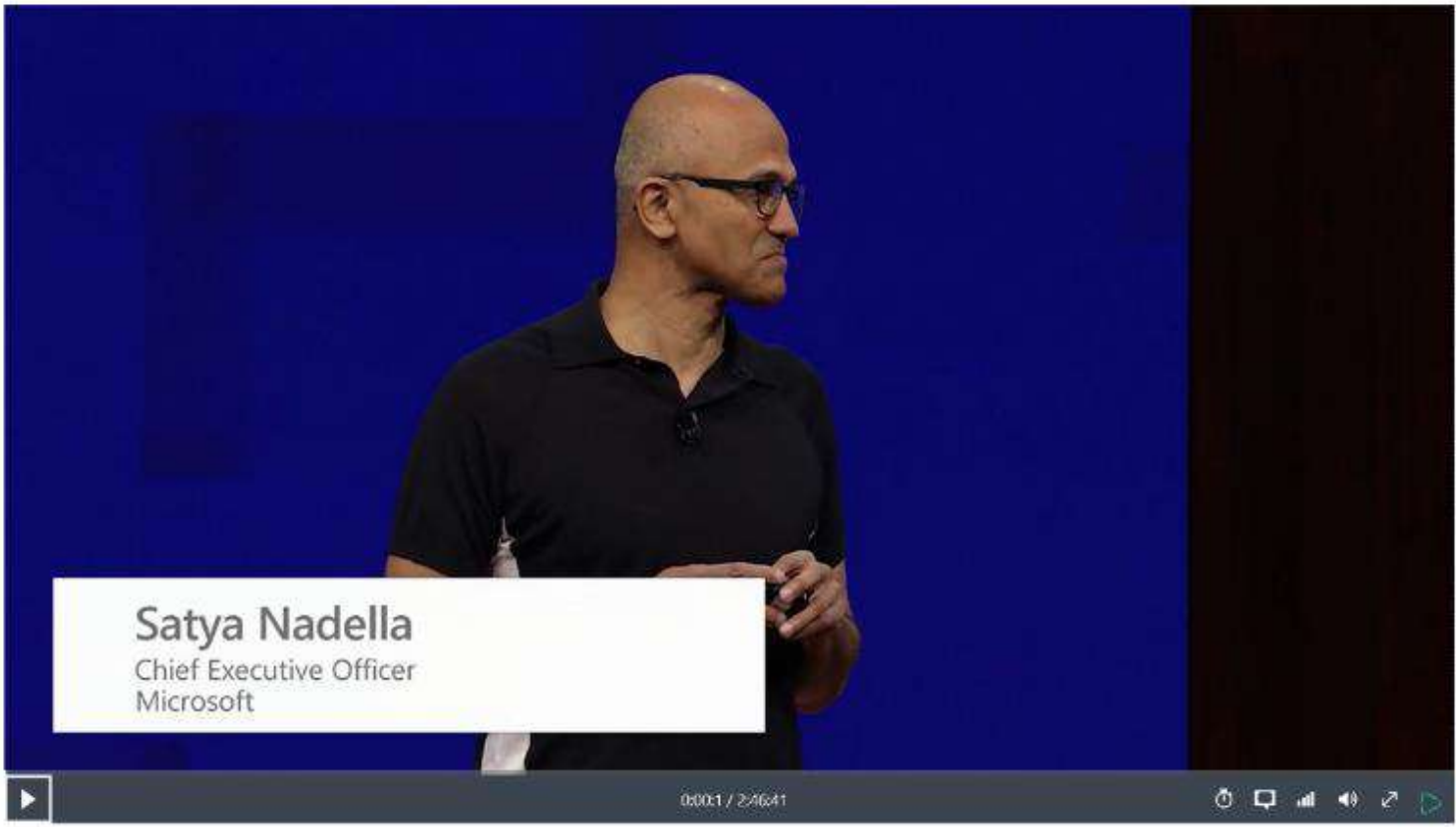
Make your content more discoverable

Quickly and easily extract insights such as spoken words, faces, characters, and emotions

Improve engagement with your video

Metadata extracted by Video Indexer can be used to build powerful engagement experiences with recommendations, highlight clips, and interactive videos





Build 2017 Keynote

Public Created 5 months ago by Video Indexer ▶ 4313 Views ⓘ Report



[Insights](#) [Transcript](#) [English](#)

People

Satya Nadella ⓘ
Microsoft CEO
[Show biography](#) [Find in Bing](#)

Appears for 17% of the video's duration.

Keywords

Show all 19

graph

Azure stack

Azure Cosmos

Azure Functions

Azure sequel

Microsoft Services

vision with Microsoft

Microsoft graph in Office

Microsoft stream

platform in Microsoft graph

Microsoft Graph Common Data

Azure Batch

App Service

Microsoft Azure Stack

Microsoft Graph Common Data Model

Annotations

Show all 120

person

indoor

man

screenshot

abstract

standing

monitor

floor

laptop

computer

Speech sentiment

Positive (76%)

Neutral (19%)

Negative (5%)

Demo

Video Indexer: <https://www.videoindexer.ai/>

Best practices for Devs

Samples and SDKs exist

For ObjectiveC/Swift/iOS, Java/Android, C#/Windows, and Python (Jupyter notebook)

<https://www.microsoft.com/cognitive-services/en-us/SDK-Sample?api=computer%20vision>

Limitations

Computer Vision API describes images in English only

Face API detects up to 64 human faces in one image

Facial detection: JPEG, PNG, GIF (first frame), and BMP supported, image file size of 1KB-4MB, detectable face size 36x36-4096x4096 pixels, returned faces ordered by face rect size desc

Fun random details

FindSimilarFace has 2 modes: matchPerson (default, same person) and matchFace (similar faces)

FaceGroup API takes between 2-1000 candidate faces

Documentation: <https://www.microsoft.com/cognitive-services/en-us/documentation>

Data

Computer Vision

Description, tags, clip art, line drawing, black & white, IsAdultContent/Score, IsRacy/Score, categories, faces, dominant colors, accent color

<https://www.microsoft.com/cognitive-services/en-us/computer-vision-api>

Emotions

Anger, contempt, disgust, fear, happiness, sadness, surprise, and neutral

<https://www.microsoft.com/cognitive-services/en-us/emotion-api>

Face

Bounding box, 27 facial landmarks, age, gender, head pose, smile, facial hair, glasses

<https://www.microsoft.com/cognitive-services/en-us/face-api>



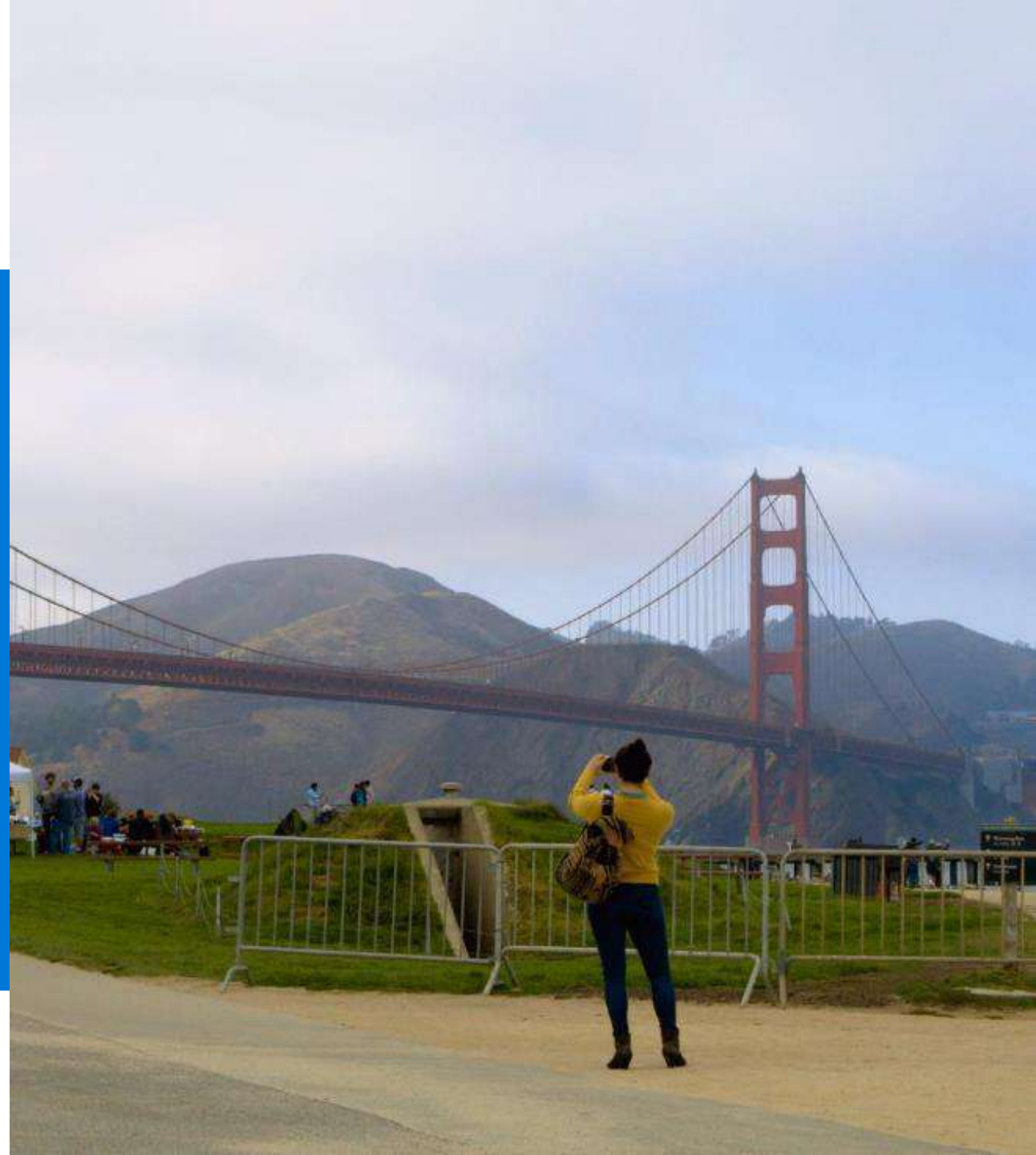
"Thousands of partners sign in to our platform every hour. The response time from the Face API is incredible, enabling us to verify our drivers without slowing them down."

Dima Kovalev, Product Manager, Uber

[Face API](#)

[Read case study here](#)

[See video here](#)



Custom Vision Service



Custom Vision Service

A customizable web service that learns to recognize specific content in imagery

Upload Images

Upload your own labeled images, or use Custom Vision Service to quickly tag any unlabeled images.

Train

Use your labeled images to teach Custom Vision Service the concepts you want it to learn.

Evaluate

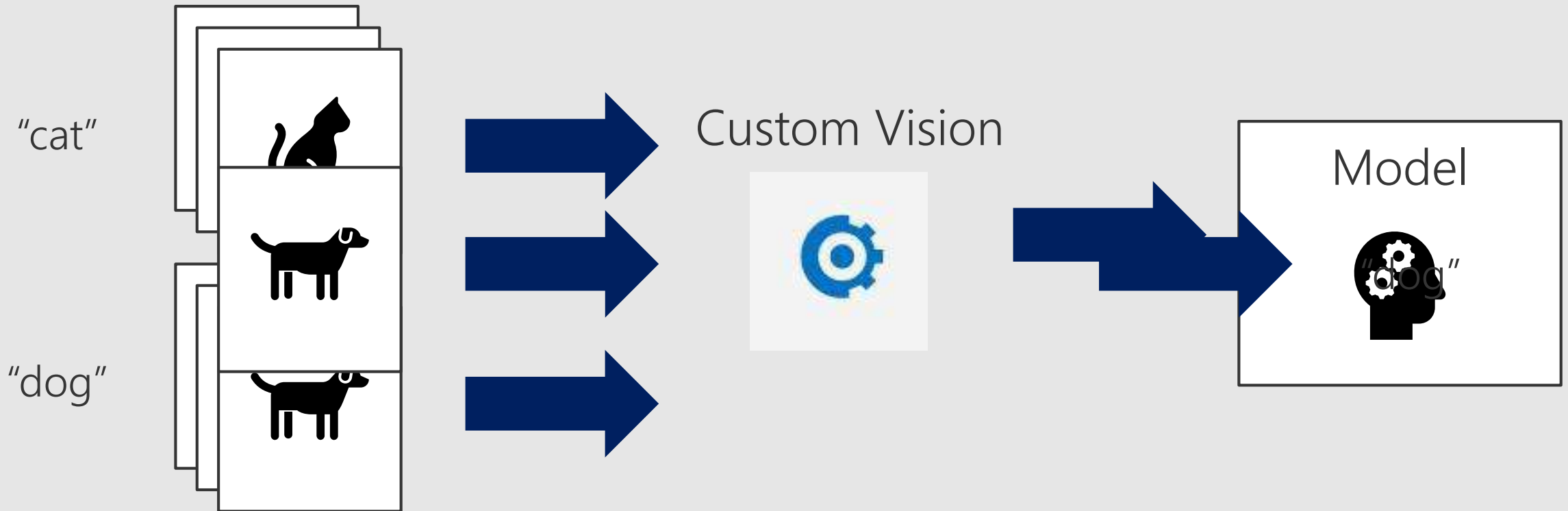
Use simple REST API calls to quickly tag images with your new custom computer vision model.

Active learning

Images evaluated through your custom vision model become part of a feedback loop you can use to keep improving your classifier.

What is it?

Custom Vision Service is an easy-to-use tool for prototyping, improving, and deploying a custom image classifier to a cloud service, without any background in computer vision or deep learning required.



Export models to mobile!

- Announcement:
<https://aka.ms/cvsexport>
- Sample:
<https://github.com/Azure-Samples/cognitive-services-ios-customvision-sample>
- Xamarin port:
<https://github.com/Xamarin/ios-samples/tree/master/ios11/CoreMLAzureModel>



Demo

Dog vs. cat classifier: <http://customvision.ai>

Intelligent Kiosk: <https://aka.ms/kioskapp>

Export to CoreML on iOS11: <https://aka.ms/cvsexport>

Best Practices for using Custom Vision

- Use at least 30 images for each tag
- Images should be the focus of the picture
- Use sufficiently diverse images and backgrounds (ex: cats with red background and dogs with blue background)
- Train with images that are similar in {quality, resolution, lighting, etc.} to the images that will be used in prod
- Supports Microsoft accounts (MSA) and AAD

Gotchas to watch for

- V1 doesn't currently do object detection with bounding boxes within an image
- Intended to be robust to subtle differences, so V1 is not well suited to tasks like defect detection/quality assurance
- Current project limitations while in preview: 1000 images, 50 tags, 20 iterations saved
- Current account limitations while in preview: 20 projects, 1000 predictions per day

Example Customer Scenarios

Customer Support

- Enable a customer to identify a product for support by taking a photo.
No finding the manual or pulling the appliance out to identify it!

Service Engineers

- Identify parts for ordering

Manufacturing

- Fault detection on assembly lines to avoid machine downtime and drop in production rates (provided differences are obvious)

Data Scientists

- Automatic tagging instead of manual, to create features or labels

Resources: Custom Vision Service

Get started at <http://customvision.ai>

Build 2017 Talk:

<https://channel9.msdn.com/Events/Build/2017/T6022>

Programmatic API access using C# (Python and Node SDKs coming soon): <https://github.com/Microsoft/Cognitive-CustomVision-Windows>

Speech



Speech



Bing Speech API

Convert speech to text and back again, and understand its intent



Speaker Recognition API

Give your app the ability to know who's talking



Custom Speech Service

Fine-tune speech recognition for anyone, anywhere

Demo

Speech: <http://dictate.ms> in Office

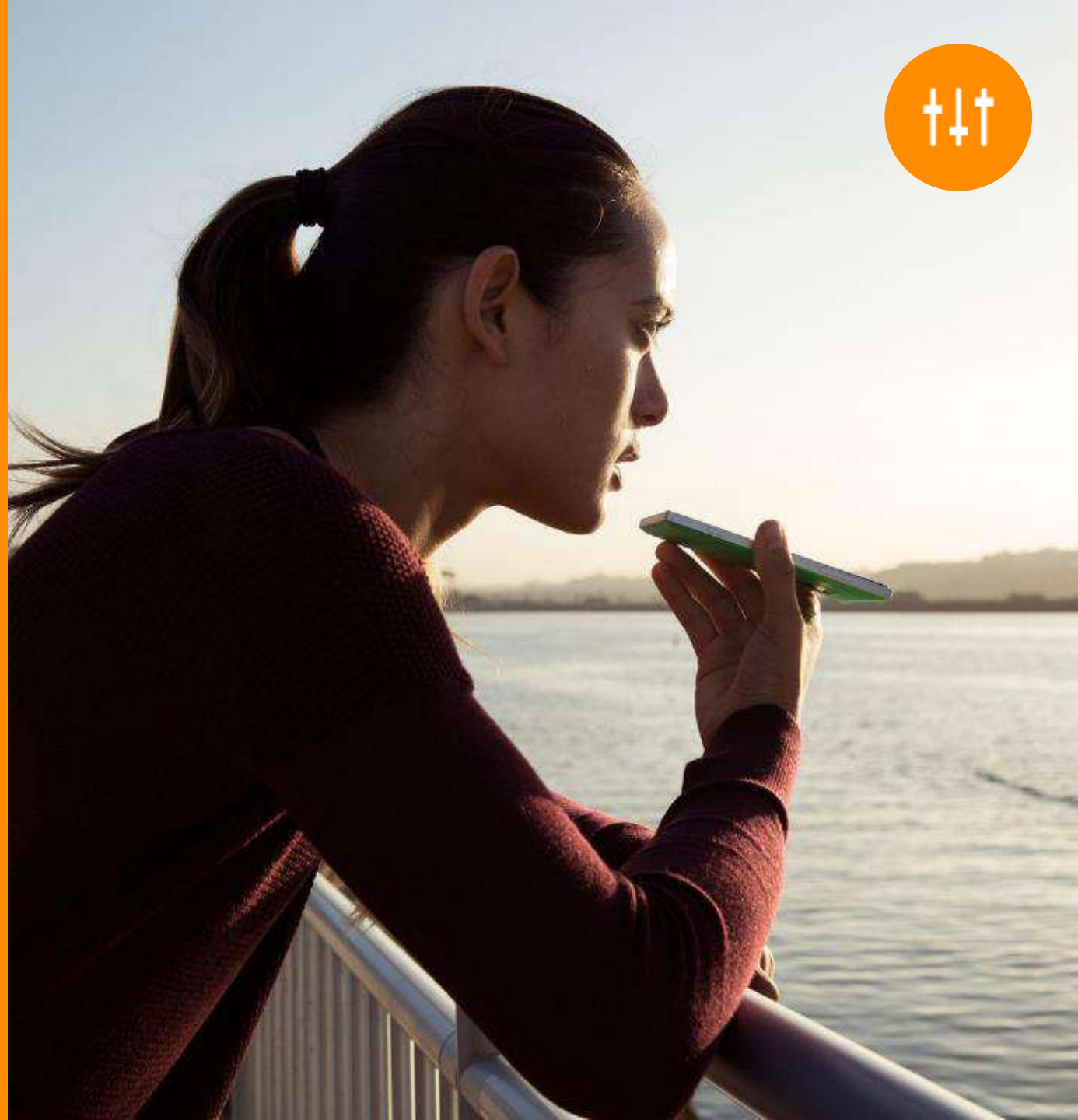
Speaker Recognition: <http://microsoft.com/cognitive>

Custom Speech Service

Custom Speech Service

Customize both language and acoustic models

Tailor speech recognition to your app and environment



Custom Speech Service

Create custom language models for the vocabulary of the application

Adapt acoustic models to better match the expected environment of the application's users

Deploy to a custom endpoint and access from any device



Record audio



Transcribe



Adapt

Deploy



STARSHIP COMMANDER

"The only reason we can build a product like this is because we are building on the deep learning and speech recognition expertise at Microsoft to deliver an entertainment experience that will be revolutionary."

Alexander Mejia, Owner and Creative Director,
Human Interact

[Custom Speech Service, LUIS](#)

[Read case study here](#)

[See video here](#)



Example Customer Scenarios

- “Hands Free” or Kiosk scenarios across Manufacturing, Retail, etc.
- Drive Through
- Call Center

Code Sample

Biology sample data:

<https://github.com/Microsoft/Cognitive-Custom-Speech-Service>

Walkthrough: <https://docs.microsoft.com/en-us/azure/cognitive-services/custom-speech-service/cognitive-services-custom-speech-get-started>

Service creation: <http://cris.ai>

Demo

Biology data: <https://cris.ai/AccuracyTests>
Airport Kiosk

Summary

Microsoft Cognitive Services

Give your apps a human side



Vision

Computer Vision
Content Moderator
Emotion
Face
Video
Video Indexer



Speech

Bing Speech
Speaker Recognition
Translator Speech

CUSTOMIZATION

Custom Vision Service

Custom Speech Service



Language

Bing Spell Check
Linguistic Analysis
Text Analytics
Translator Text
Web Language Model

Language Understanding



Knowledge

Academic Knowledge
Entity Linking
Knowledge Exploration
Recommendations
QnA Maker

Custom Decision Service



Search

Bing Autosuggest
Bing Image Search
Bing News Search
Bing Video Search
Bing Web Search
Bing Entity Search

Bing Custom Search



Labs

Project Prague (gesture)
Project Cuzco (events)
Project Johannesburg (routing)
Project Nanjing (isochrones)
Project Abu Dhabi (distance matrix)
Project Wollongong (location)

Resources

- Website: <http://microsoft.com/cognitive>
- UserVoice: <https://cognitive.uservoice.com>
- StackOverflow tag: microsoft-cognitive
- Microsoft Ignite content can be found at <https://channel9.msdn.com/Events/Ignite>

Resources: Cognitive Services

Microsoft Cognitive Services Developer Code of Conduct:

<https://azure.microsoft.com/en-us/support/legal/developer-code-of-conduct>

Microsoft Cognitive Services Terms - applicable to free previews, excluding generally available Microsoft Azure and Volume Licensing

<https://go.microsoft.com/fwlink/?LinkId=533207>

Online Services Terms – applicable to generally available Cognitive Services use via Microsoft Azure or Volume Licensing (once available)

<http://www.microsoftvolumelicensing.com/DocumentSearch.aspx?Mode=3&DocumentTypeId=31>

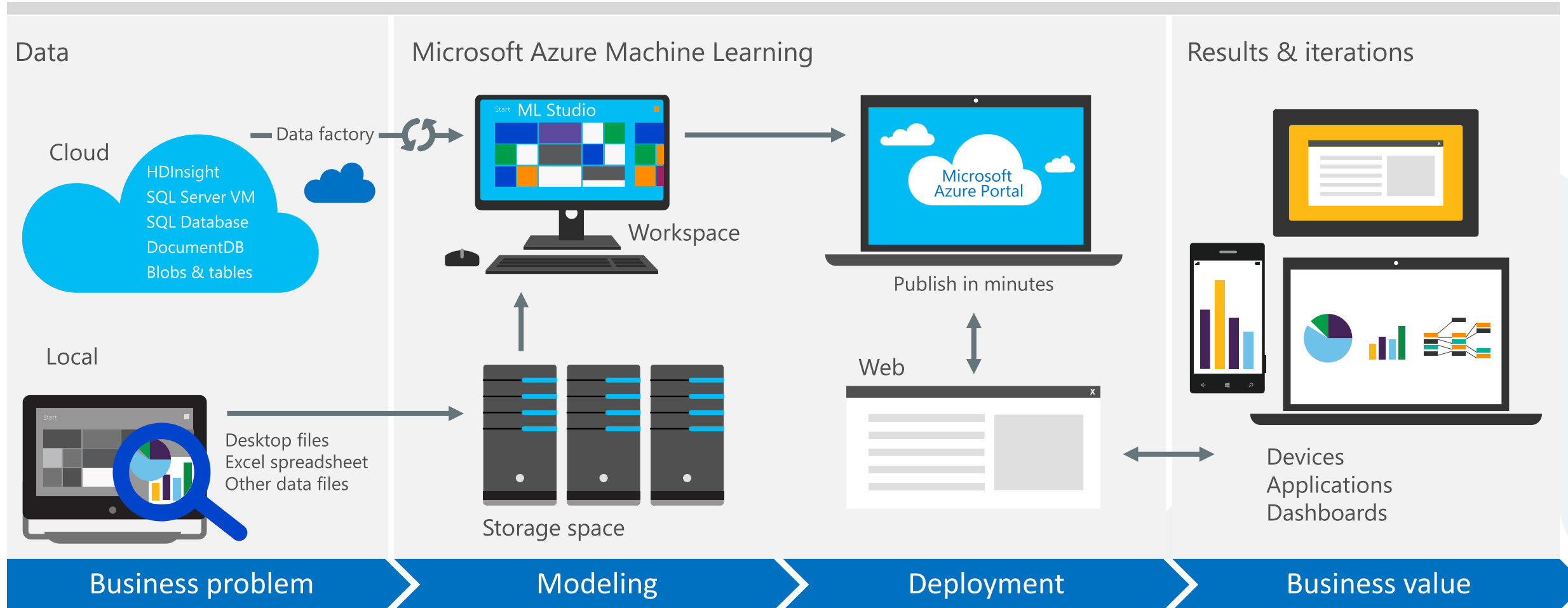
Predictive Analytics

Azure Machine Learning



Azure Machine Learning

The power of machine learning to predict future trends or behaviors



AML - Drag & Drop + Best in Class Algorithms

The screenshot displays the Microsoft Azure Machine Learning Studio interface. The top navigation bar includes links for Home, Studio, and Gallery. The main workspace is titled "Predictive Maintenance Demo - Training - Original" and is in "In draft" status. A left-hand sidebar contains a search bar and a list of experiment items: Saved Datasets, Trained Models, Transforms, Data Format Conversions, Data Input and Output, Data Transformation, Feature Selection, Machine Learning, OpenCV Library Modules, Python Language Modules, R Language Modules, Statistical Functions, Text Analytics, Web Service, and Deprecated. The central workspace shows a drag-and-drop workflow with the following steps: 1. "TrainingData.csv" (data source) connects to "Project Columns" (data transformation). 2. "Project Columns" connects to "Split" (data transformation). 3. "Split" connects to "Multiclass Decision Forest" (machine learning algorithm) and "Train Model" (training step). 4. "Train Model" connects to "Score Model" (scoring step). 5. "Score Model" connects to "Evaluate Model" (evaluation step). A right-hand sidebar contains the "Properties" panel, which includes sections for "Experiment Properties" (Start Time, End Time, Status Code: InDraft, Status Details: None), "Summary" (Training predictive maintenance demo), and "Description" (Enter the detailed description for your experiment). A "Quick Help" section is also visible at the bottom of the right sidebar. The bottom of the interface features a toolbar with icons for New, Run History, Save, Discard Changes, Run, Set Up Web Service, and Publish to Gallery.

Microsoft Azure Machine Learning | Home Studio Gallery PREVIEW

Predictive Maintenance Demo - Training - Original In draft

Search experiment items

- Saved Datasets
- Trained Models
- Transforms
- Data Format Conversions
- Data Input and Output
- Data Transformation
- Feature Selection
- Machine Learning
- OpenCV Library Modules
- Python Language Modules
- R Language Modules
- Statistical Functions
- Text Analytics
- Web Service
- Deprecated

Workflow steps:

- TrainingData.csv
- Project Columns
- Split
- Multiclass Decision Forest
- Train Model
- Score Model
- Evaluate Model

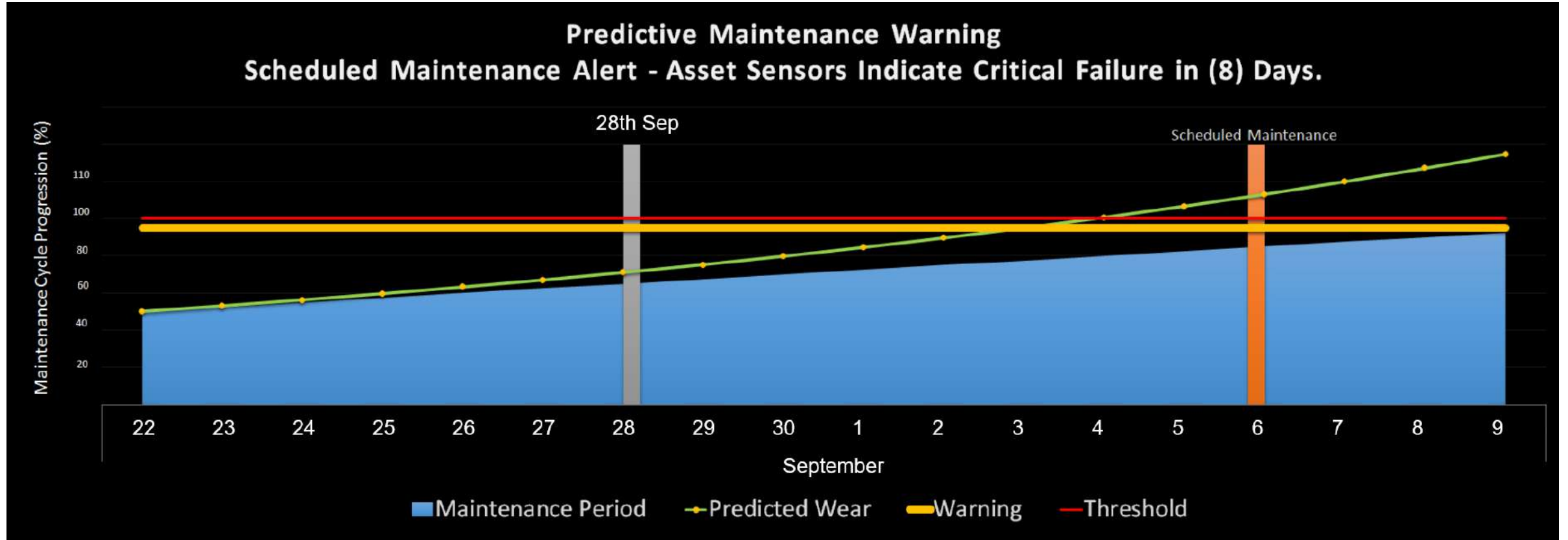
Properties

- Experiment Properties
 - START TIME -
 - END TIME -
 - STATUS CODE InDraft
 - STATUS DETAILS None
 - Go to web service
 - Prior Run
- Summary
 - Training predictive maintenance demo
- Description
 - Enter the detailed description for your experiment.

Quick Help

+ NEW RUN HISTORY SAVE DISCARD CHANGES RUN SET UP WEB SERVICE PUBLISH TO GALLERY

Integrated predictive analytics



Empower with proactive analysis

Machine learning solutions allow for powerful predictive analytics solutions, leveraging historical data and real time device ingestion input.

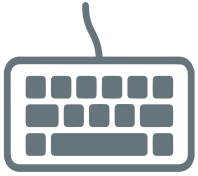


Azure Bot Service

[About](#)

Why a bot?

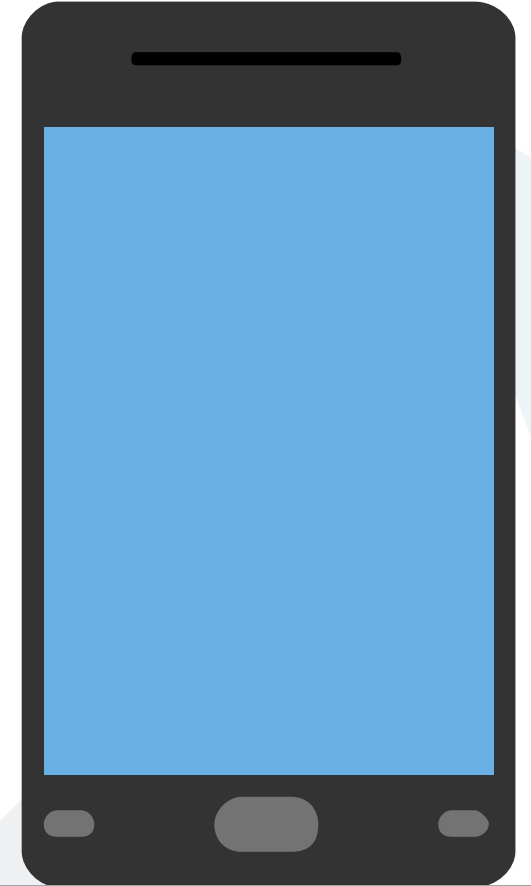
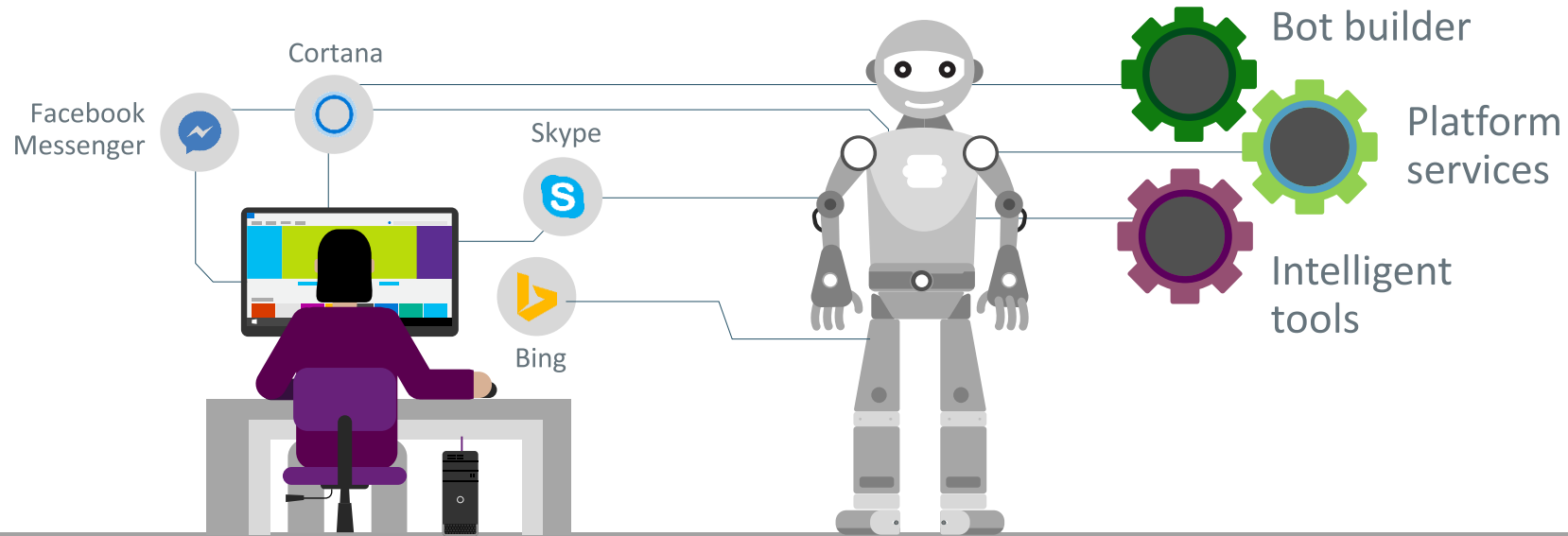
TYPE



TALK



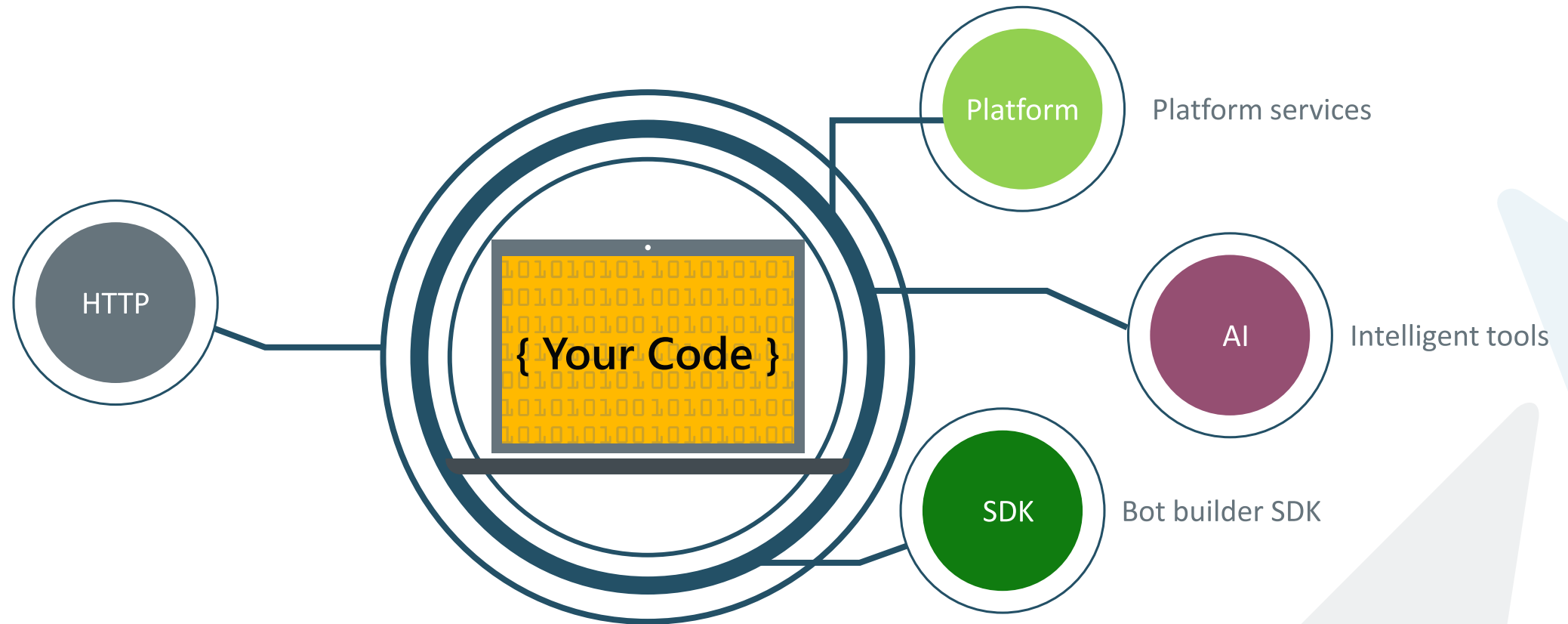
TAP



Kinds of bots

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 a bot?



REST endpoint
[Direct Line Protocol](#)



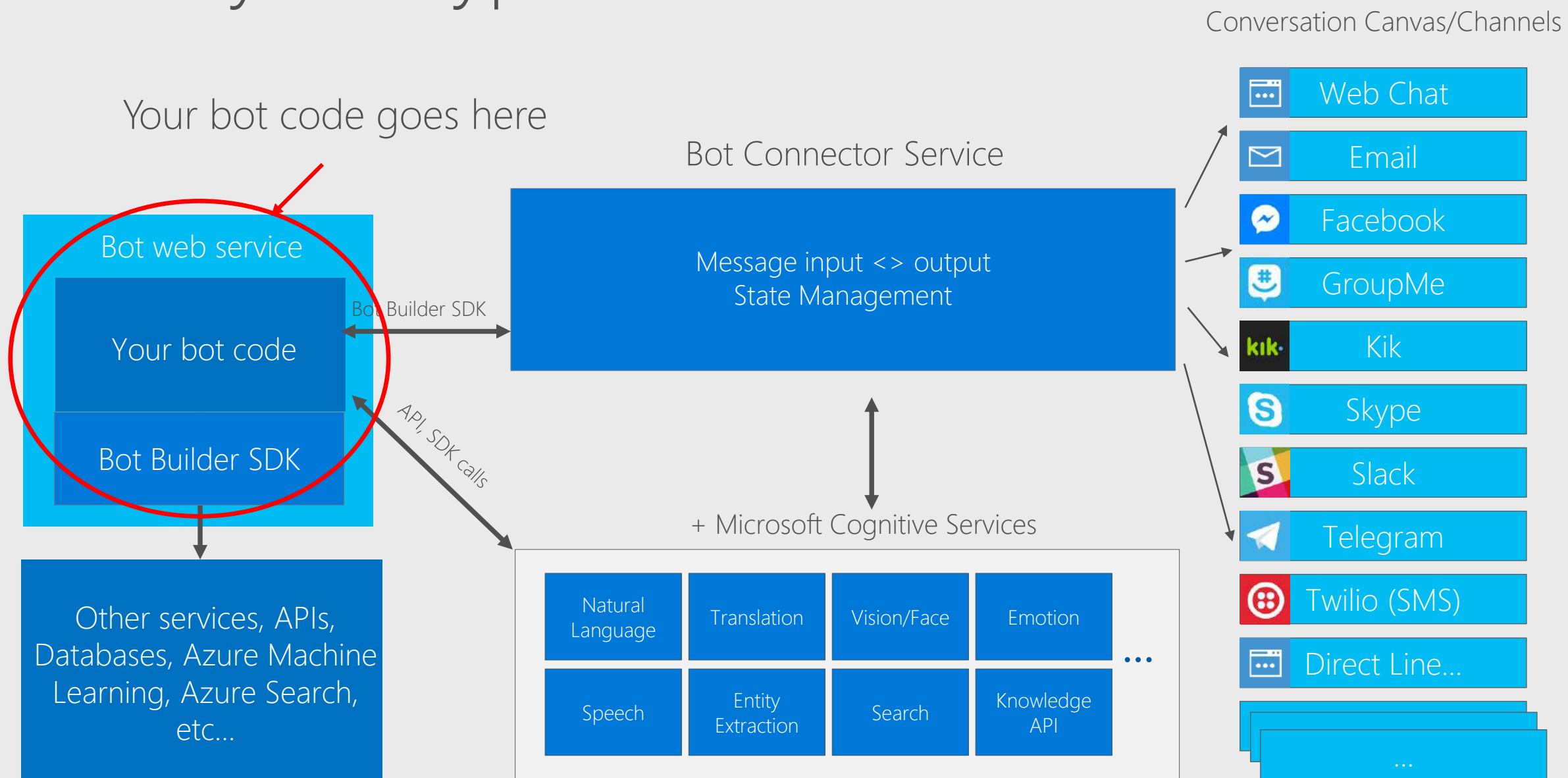
Conversational and
business logic

Canvas aware



Context
sensitive

An x-ray of a typical bot





Building Bots

- The hard way – DIY
 - Implement a REST API
- The easy(-ier) way – Bot Builder SDK
 - .NET
 - Node.js





Getting Started with Predefined Templates

- Bot Service includes five templates to help you get started with building bots (C# and Node.js)
 - Basic bot
 - A bot that uses dialogs to respond to user input
 - The bot echoes back to the user whatever they type in
 - You can use this template to get started building conversation flow in your bot
 - Form bot
 - A bot that collects input from a user via a guided conversation
 - For example, a bot that is designed to obtain a user's sandwich order
 - Language Understanding bot
 - A bot that uses natural language models to understand user intent
 - This template leverages Language Understanding (LUIS) to provide natural language understanding
 - Question and Answer bot
 - A bot that distills semi-structured data like question and answer pairs into distinct, helpful answers,
 - This template leverages the QnA Maker service to parse questions and provide answers
 - Proactive bot
 - A bot that can send proactive messages to the user
 - For example:
 - if a bot sets a timer or reminder, it may need to notify the user when the time arrives
 - if a bot receives a notification about an external event, it may need to communicate that information to the user



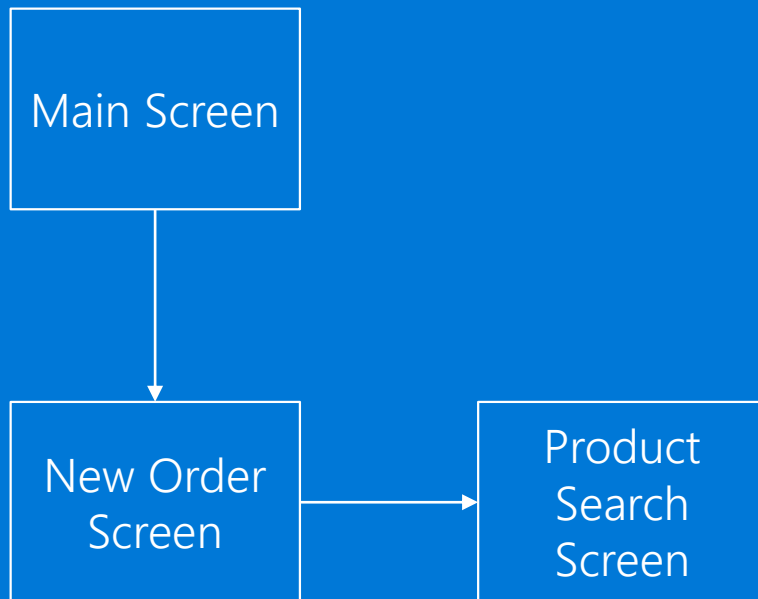
Deployment Options

- With the App Service plan
 - a bot is a standard Azure web app you can set to allocate a predefined capacity with predictable costs and scaling.
- With a Consumption plan
 - a bot is a serverless bot that runs on Azure Functions and uses the pay-per-run Azure Functions pricing.

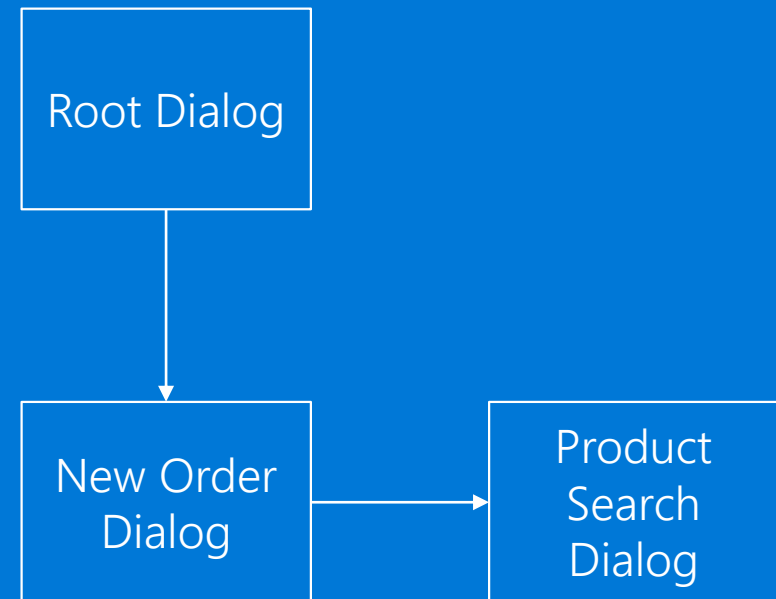
Dialogs are for bots like screens are for apps

They separate concerns and organize flows, exactly the same way:

Traditional Application



Bot



Dialogs are serialized into stacks

Root Dialog

This is how we know how to navigate “back”
(Users won’t necessarily think that way)

Dialog 1

So careful with too much “modality”: Things can get complex quickly

Dialog 2

...

[Read more](#)



Implementing Dialogs

- A dialog is a class implementing a piece of the bot's business logic
- One method to implement:

```
public interface IDialog<out TResult>
{
    Task StartAsync(IDialogContext context);
}
```




Conversation Flow

- The active dialog takes control of conversation flow
- Using the SDK you can:
 - *Context.Wait()*
 - *Context.Done()*
 - *Context.Fail()*
 - *Context.Foreward()*
 - *Context.Call()*

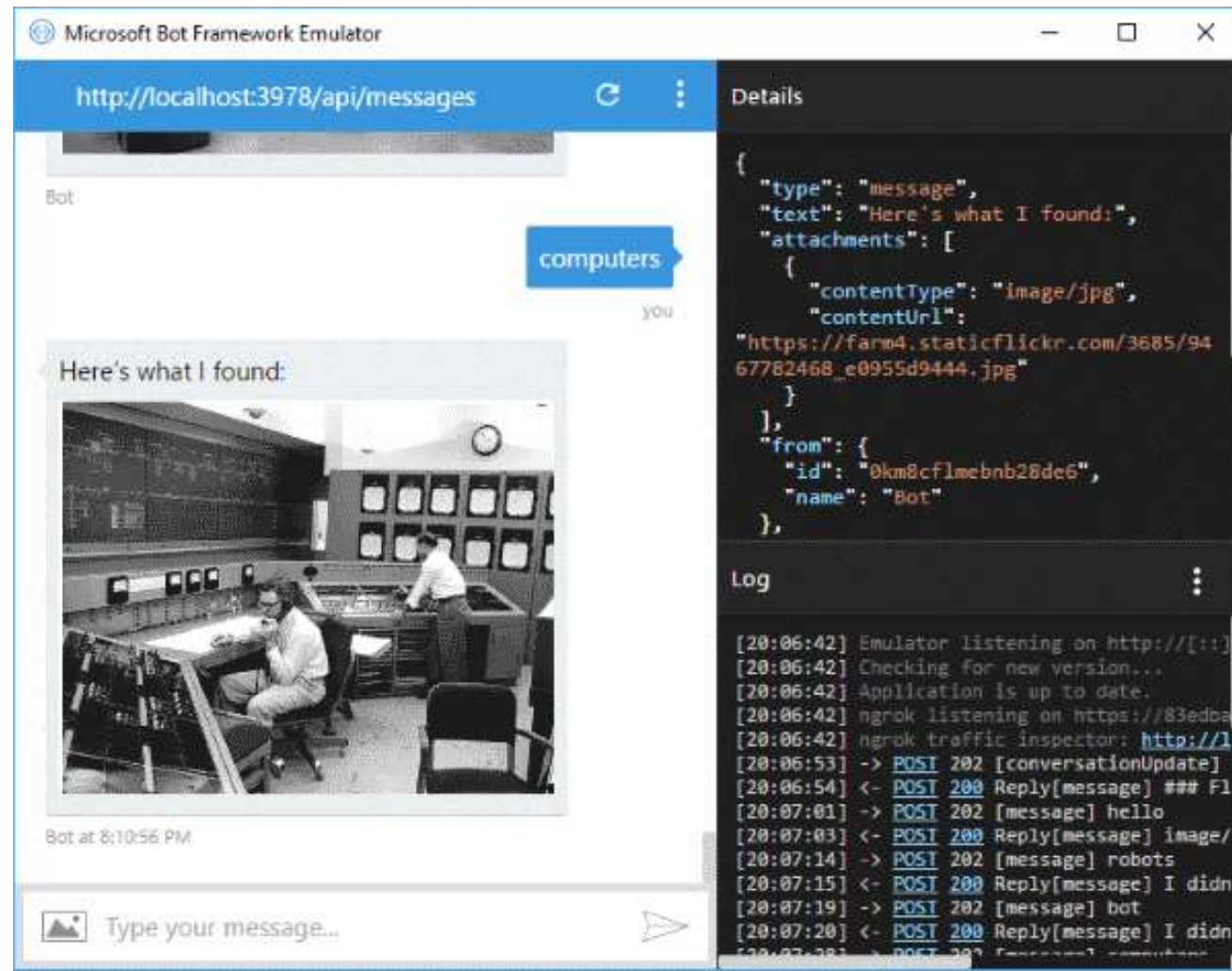


Dialog State

- Code continuations allow modeling a state machine

```
if (...)
{
    context.Call(OrderDialog.CreateDialog(), BeerOrderedAsync);
    return;
}
```


Bot Framework Emulator



Chain API

- A Fluent API for building dialogs
- Available using the ***Chain*** class

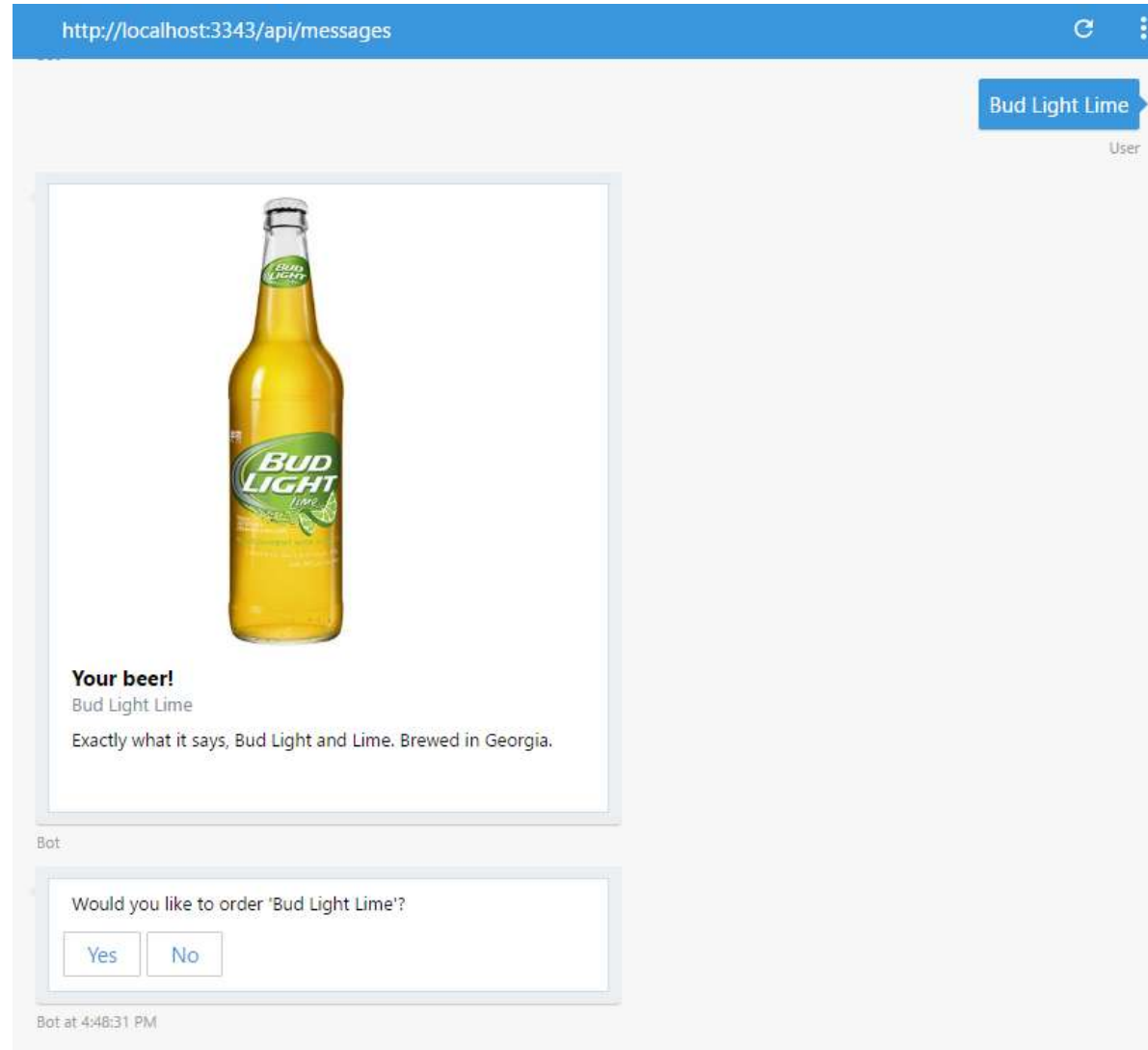
```
public static readonly IDialog<Beer> Dialog = Chain
    .From(() => new PromptDialog.PromptChoice<RecommendationOptions>(
        new[] { RecommendationOptions.Category, RecommendationOptions.Origin, RecommendationOptions.Name },
        "How would you like me to recommend your beer?",
        "Not sure I got it. Could you try again?",
        3, descriptions: new [] { "By Beer Category" , "By Beer Origin", "By Beer Name" }))
    .Switch(
        Chain.Case<...>(option => option == RecommendationOptions.Category, (context, option) => CategoryRecommendation),
        Chain.Case<...>(option => option == RecommendationOptions.Origin, (context, option) => CountryRecommendation),
        Chain.Case<...>(option => option == RecommendationOptions.Name, (context, option) => NameRecommendation)
    )
    .Unwrap();
```


- ▶ Sometimes we need the user to input some predefined fields




```
public class BeerOrder
{
    [Prompt("What beer would you like?")]
    public string BeerName { get; set; }
    [Prompt("Which chaser would you like next to your beer? {||}")]
    public Chaser Chaser { get; set; }
    [Prompt("How about something to eat? {||}")]
    public SideDish Side { get; set; }
}

public static IDialog<BeerOrder> CreateDialog(string beerName = null)
{
    return new FormDialog<BeerOrder>(
        new BeerOrder { BeerName = beerName }, ...);
}
```



Creating a Card

```
.ContinueWith(async (context, beerAwaitable) =>
{
    var chosenBeer = await beerAwaitable;
    Uri imageUrl = await ImageSearchService.SearchImage($"{chosenBeer.Name} beer");
    var card = new HeroCard("Your beer!", chosenBeer.Name, chosenBeer.Description,
        new List<CardImage> { new CardImage(imageUrl.ToString()) });

    var message = context.MakeMessage();
    message.AttachmentLayout = AttachmentLayoutTypes.Carousel;
    message.Attachments = new List<Attachment> {card.ToAttachment()};
    await context.PostAsync(message);
    return Chain.Return(chosenBeer);
});
```


“Typing” Indicator



Sending a Notification

- Sending a Typing notification is easy

```
var typingMessage = context.MakeMessage();  
typingMessage.Type = ActivityTypes.Typing;  
await context.PostAsync(typingMessage);  
await Task.Delay(1000);    // Do Work
```

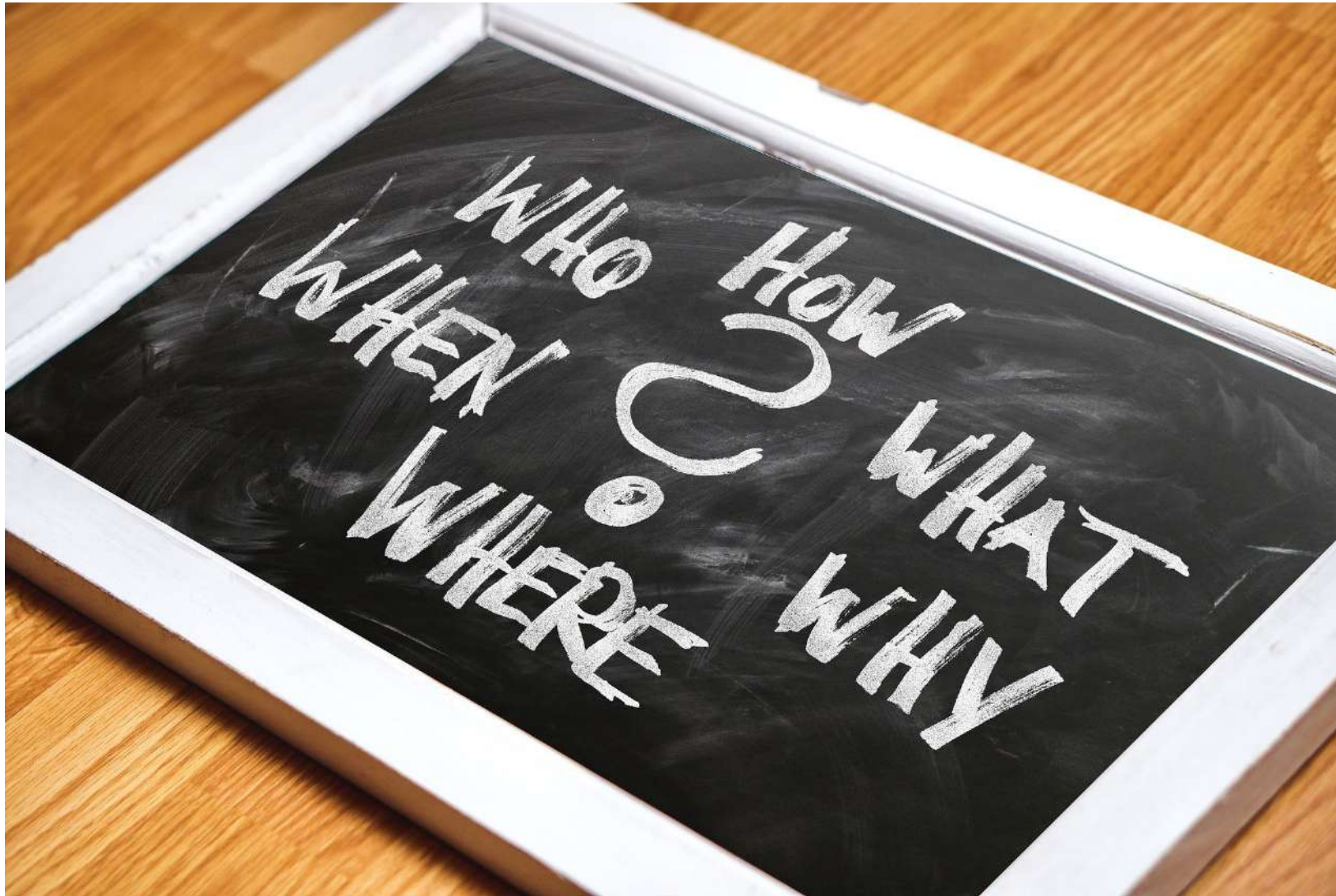



Bot State

- Use the session object
 - User Data
 - Conversation Data
 - Private Conversation Data
 - Dialog Data
- Persisted and managed by the Bot Framework State Service
 - IBotStorage
- Various providers exist
 - In-Memory
 - Table storage
 - CosmosDB
 - DIY...



Natural Language



Language Understanding Intelligent Service

Add conversational intelligence to your apps.

[Sign in or create an account](#)

What is the **weather** **today** in **Seattle**?

intent = weather
place = seattle, wa
date = May 3 2017, 8am

Do I need my **umbrella**?

LUIS & Bot Framework

```
[LuisIntent("Bye")]
public async Task OnByeAsync(...)
{
    await context.PostAsync("Bye bye. See you soon!");
    context.Done((object) null);
}

[LuisIntent("RecommendBeer")]
public Task OnRecommendBeerAsync(...)
{
    var beerName = GetEntity(luisResult, BeerNameEntityName);
    var brewery = GetEntity(luisResult, BreweryEntityName);
    var category = GetEntity(luisResult, CategoryEntityName);
    var country = GetEntity(luisResult, CountryEntityName);

    context.Call(RecommendationDialog.CreateDialog(beerName, brewery, category, country), BeerRecommendedAsync);
    return Task.FromResult((object) null);
}
```


Demo

```
68 $mockQueryBuilder->shouldReceive('newQuery')->once()->andReturn($query);
69 $relation->expects($this->once())->method('touchIfTouching');
70
71 $this->assertTrue($relation->detach());
72 }
73
74 public function getRelation()
75 {
76     list($builder, $parent) = $this->getRelationArguments();
77
78     return new MorphToMany($builder, $parent, 'taggable', 'taggables', 'taggable_id', 'tag_id');
79 }
80
81 public function getRelationArguments()
82 {
83     $parent = m::mock('Illuminate\Database\Eloquent\Model');
84     $parent->shouldReceive('getMorphClass')->andReturn(get_class($parent));
85     $parent->shouldReceive('getKey')->andReturn(1);
86 }
87
```

Line 81, Column 1

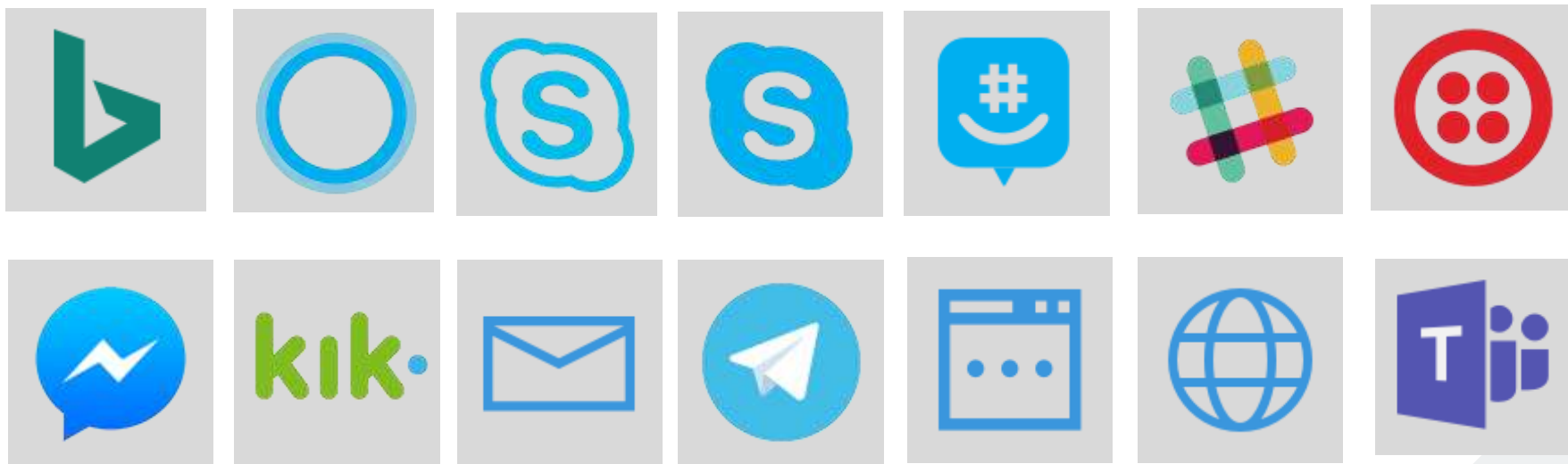
LUIS Integration

Deployment

- The Bot is just a REST API
- Can be hosted anywhere
- Azure App Service is an easy choice
 - Azure Functions also supported



Channels



Takeaways

- Chat bots are another form of UI (NLUI)
- Microsoft Bot Framework makes it easier to write your bot
 - Standard connection to various channels
 - Bot Builder SDK
 - Integration with Cognitive Services
- The trickiest part is still designing the conversation



Thank you!

