

Sponsors and Partners

Strategic Sponsors



Gold Sponsors



Silver Sponsors





The Cloud was made for APIs

Bartłomiej Zass, Microsoft CEE

APIs and API economy

Mobile

Hybrid IT

IoT

Cloud

Big data

Microservices

“Application programming interfaces (APIs) have been elevated from a development technique to a **business model driver** and **boardroom** consideration.

An organization’s core assets can be reused, shared, and **monetized** through APIs that can extend the reach of existing services or provide **new revenue streams**.

APIs should be managed like a **product** - one built on top of a potentially **complex technical footprint** that includes legacy and third-party systems and data.

From the [“API Economy”](#)

by George Collins and David Sisk
Deloitte Consulting LLP, 2015

Introducing the

Open Banking Standard

Helping customers, banks
and regulators take banking
into a truly 21st-century,
connected digital economy

Agenda

REST API & Swagger basics

Hosting APIs on App Service

Integrating APIs with Logic Apps

Exposing & monetizing APIs with API Management

Anatomy of REST APIs

REST = Representational State Transfer

Builds on-top of HTTP

Object State Operations are mapped to HTTP Verbs

HTTP Verbs:

Get – Select

Post – Create

Put – Update or Create

Delete – Delete

Options – which supported?

Head

...

REST – Where?

URL = Resource Locator w. Parameters

`http://myservice/customers/123`

`http://myservice/car/WE123123/owner/name`

REST – How?

Feels like „native“ HTTP

HTTP Verbs

Response Format Negotiation

Content-Type: <requested-MimeType>

Using HTTP Status Codes (200, 400, 404, 500, ...)

HTTPS / TLS Encryption

REST – What?

Content is NOT standardized

Could be XML / JSON / CSV / Images / HTML / Text / ...

HTTP Header:

Content-Type: application/json

```
{
  id: "123",
  licensePlate: "WE123123",
  owner: {
    name: "Jan Kowalski"
  }
}
```

JSON Schema (<http://json-schema.org>)

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "definitions": {
    "address": {
      "type": "object",
      "properties": {
        "street_address": { "type": "string" },
        "city":           { "type": "string" },
        "state":          { "type": "string" }
      },
      "required": ["street_address", "city", "state"]
    }
  },
  "type": "object",
  "properties": {
    "billing_address": { "$ref": "#/definitions/address" },
    "shipping_address": {
      "allOf": [
        { "$ref": "#/definitions/address" },
        { "properties": {
            "type": { "enum": [ "residential", "business" ] } },
          "required": ["type"]
        }
      ]
    }
  }
}
```

Type name

Data Type (object, numer, string, ...)

Complex Type Reference

```
{
  "shipping_address": {
    "street_address": "1600 Pennsylvania Avenue NW",
    "city": "Washington",
    "state": "DC"
  }
}

{
  "shipping_address": {
    "street_address": "1600 Pennsylvania Avenue NW",
    "city": "Washington",
    "state": "DC",
    "type": "business"
  }
}
```



<http://json-schema.org/shared.html>

Benefits of Contracts / Schema

Contract = WYSIWYG (a meta description of the service)

...are the documentation

...can be used to auto-generate code

...are up-to-date

...can be used for validation



Just the JSON payload,
not for operations!

Examples of contracts: WSDL (for SOAP), XSD (for XML), JSON-Schema (for JSON)

Introducing Swagger

The WSDL of REST

<http://swagger.io>

Multi-Language Support

Generate SDK

Documentation

Test page



Swagger became OpenAPI Specification



The OAI is focused on **creating**, **evolving** and **promoting** a vendor neutral API Description Format based on the Swagger Specification.



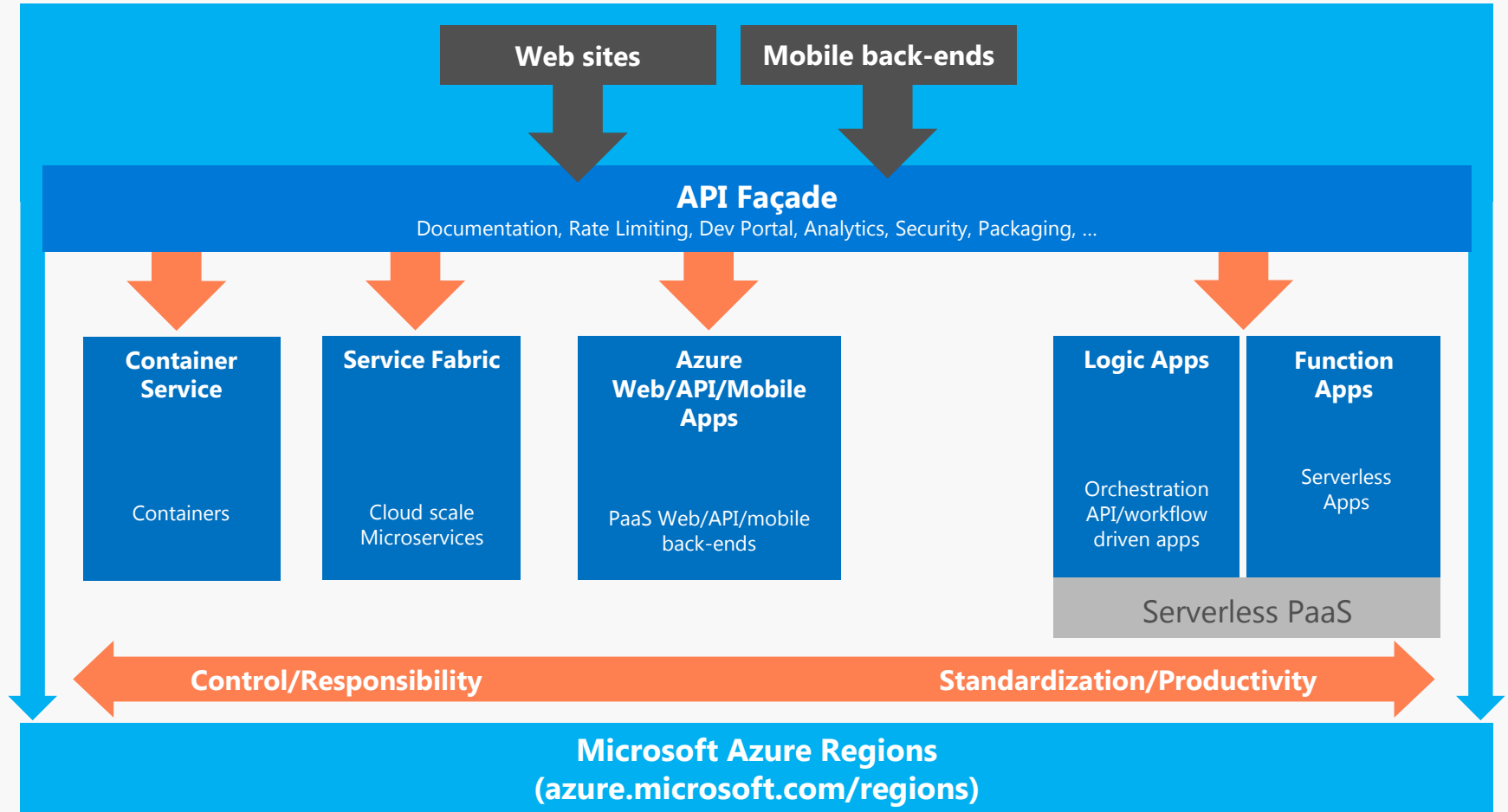
Demo

JSON Schema
Swagger
Hello, API Apps!

Hosting Your APIs

on Azure App Service

Azure App Continuum



Azure App Service



Web apps

Web apps that scale with your business



Mobile apps

Build mobile apps for any device



Logic apps

Automate business processes across SaaS and on-premises



API apps

Easily build and consume APIs in the cloud



Functions

Serverless event based development accelerator

App Service Core Capabilities

All features and capabilities are shared across all of App Service application (Web, Mobile, Functions and API)

Enterprise grade

Designed for secure mission-critical applications

- Premium Tier
- App Service Environments
- Hybrid Connections / VPN Support
- Scheduled Backup
- Azure Active Directory Integration
- Site Resiliency, HA, and DR
- Role Base Access Control
- Audit / Compliance
- Enterprise Migration
- Client Certs
- IP Restrictions/ SSL
- Dedicated IP address IP / NSG
- Web Sockets
- WW Datacenter Coverage

Fully managed

Optimized for Availability and Automatic scale

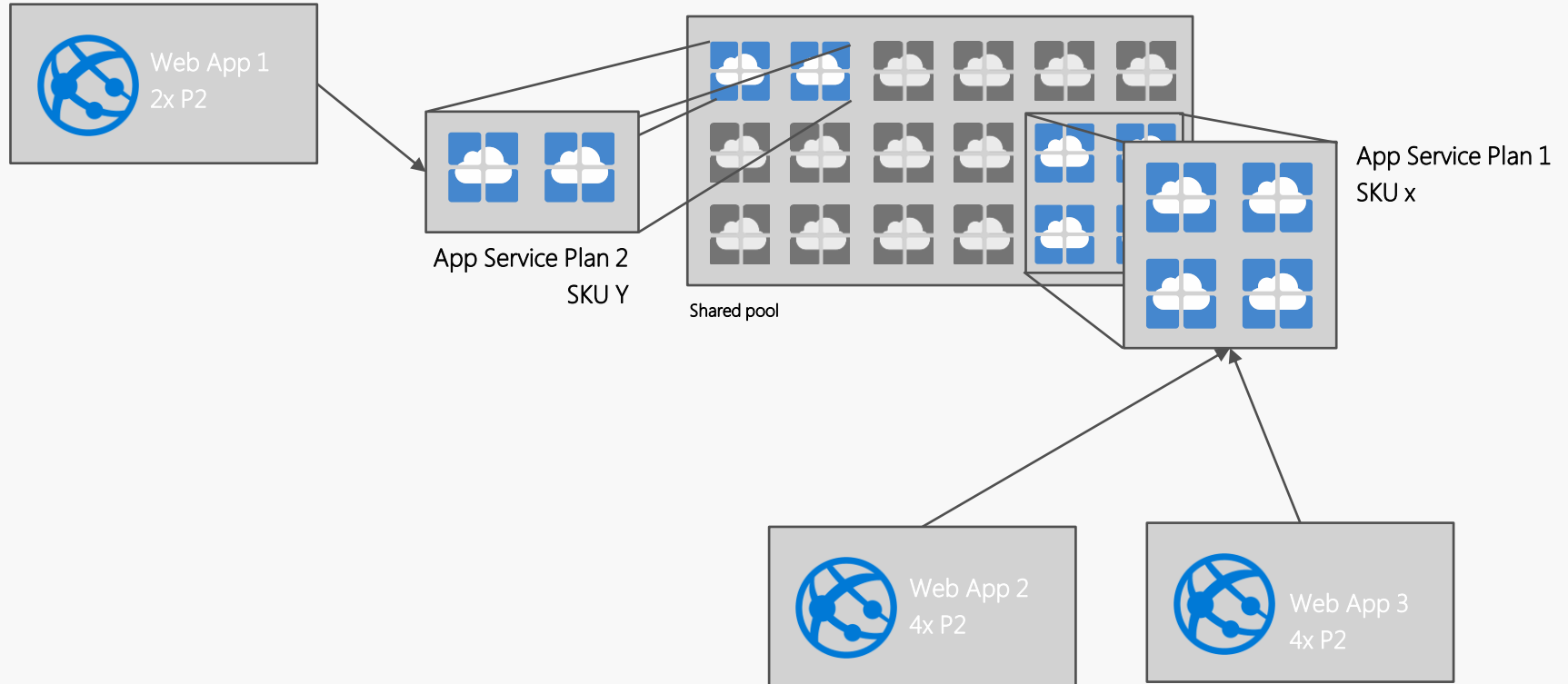
- Automated Deployment
- AutoScale
- Built-in Load Balancing
- WW Datacenter Coverage
- End Point Monitoring & Alerts
- Wildcard Support
- HTTP Compression
- WebJobs
- Sticky Sessions
- OS & Framework Patching
- Auto-Healing
- Local Cache
- Init Module
- Per Site Scaling
- Easy Auth

Built for DevOps

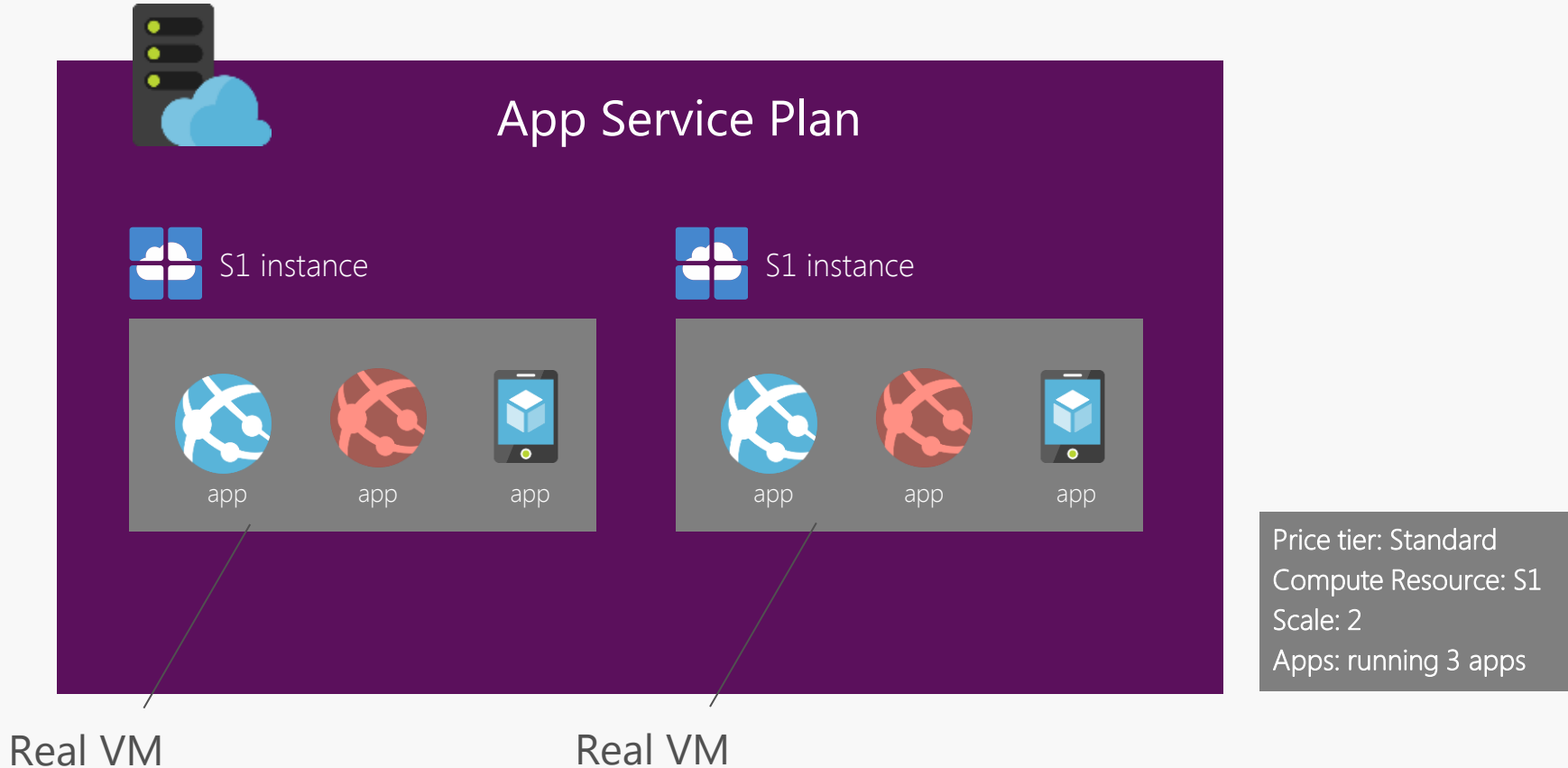
Agility through Continuous Deployment

- Remote Debugging w/ Visual Studio
- Site Staging Slots /Preview
- Traffic Routing
- Continuous Integration/Deployment
- Git/ Hub, Visual Studio Team Services
- App & Site Diagnostics
- Site Extensions/ Gallery
- NET, PHP, Python, Node, Java, Go
- Framework Installer
- Browser-based editing
- Logging and Auditing
- Admin-Site
- Support Portal
- Web Jobs / SDK 1.1
- Recommendation Engine
- Site Cloning

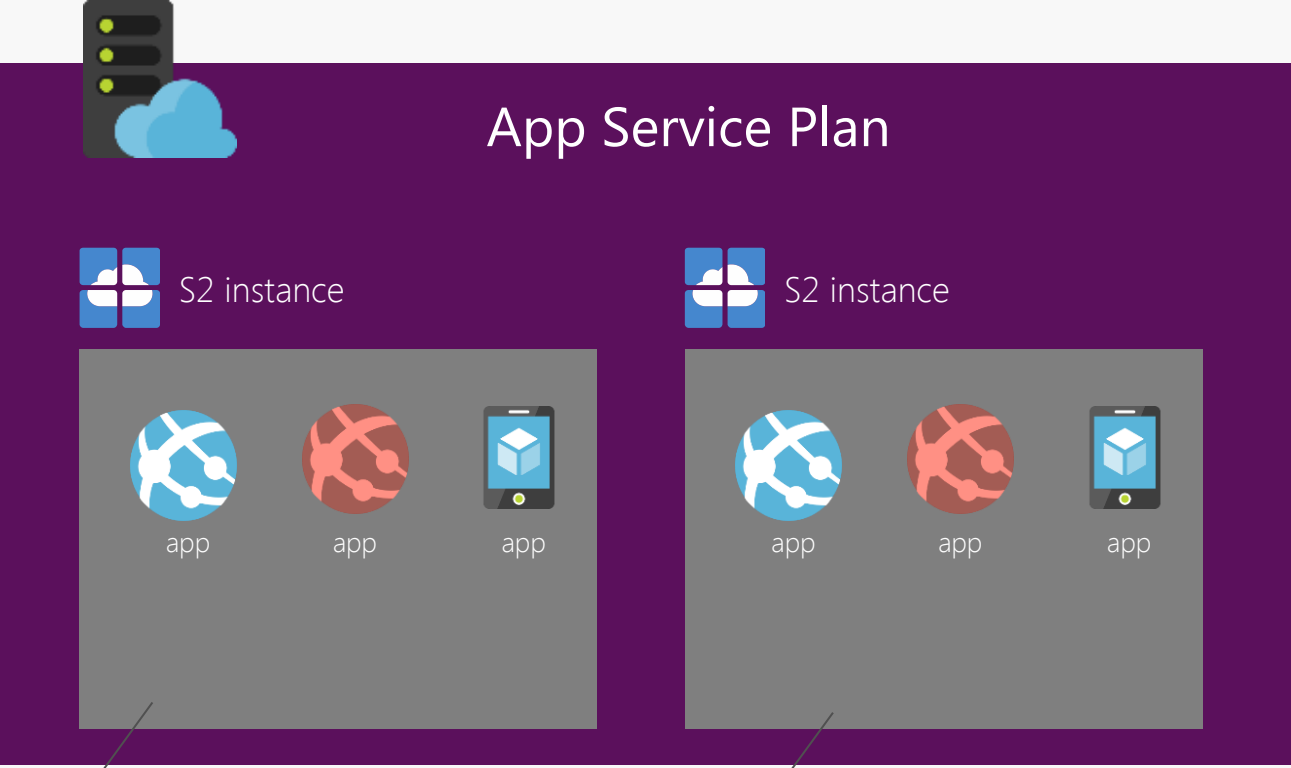
App Service Plans & Apps



Host on an App Service Plan



Scale-Up



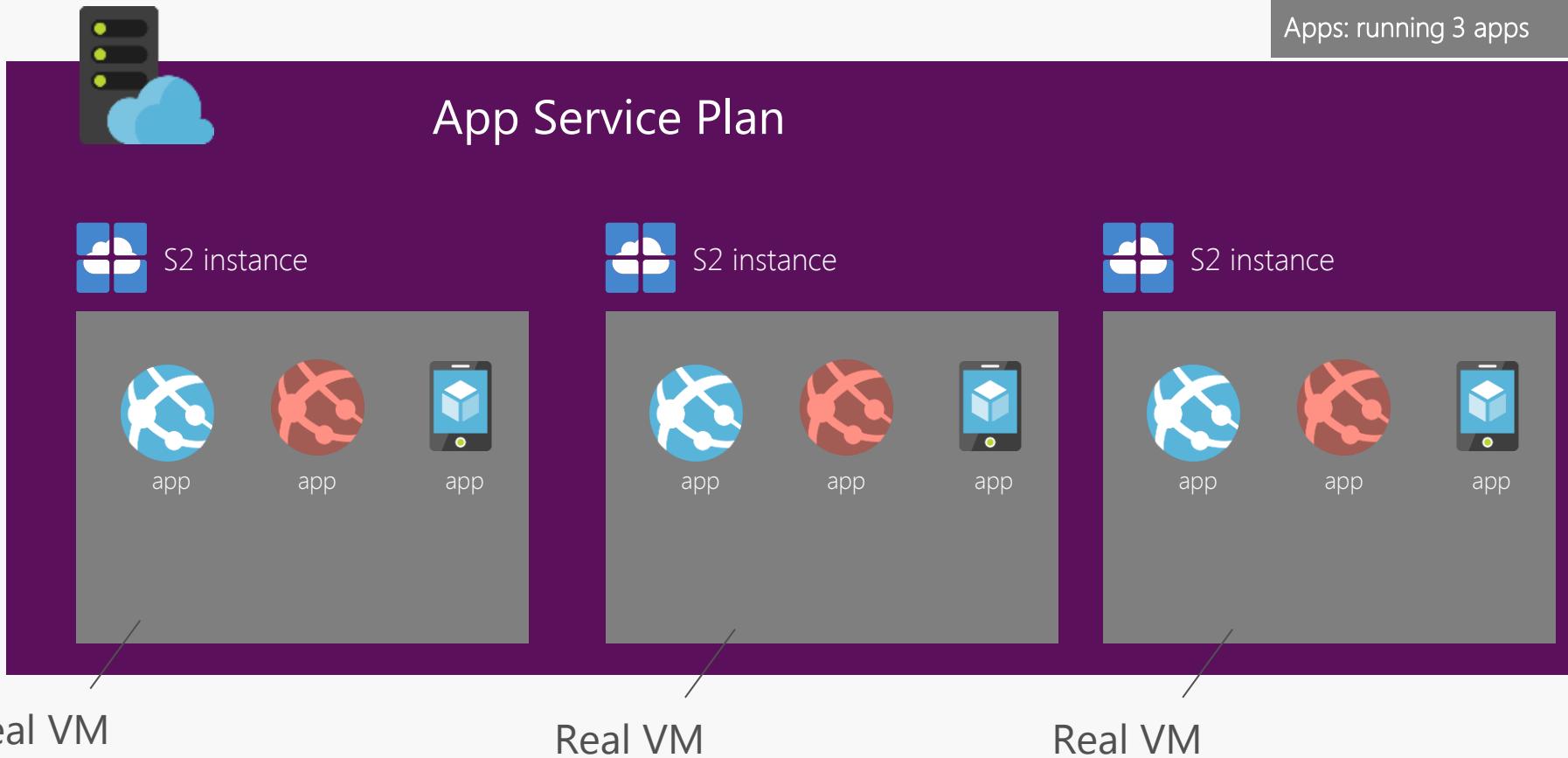
Real VM

Real VM

Price tier: Standard
Compute Resource: S2
Scale: 2
Apps: running 3 apps

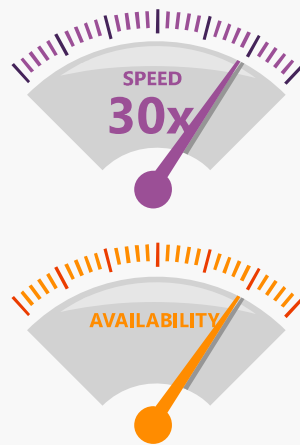
Scale-Out

Price tier: Standard
Compute Resource: S2
Scale: 3
Apps: running 3 apps



Auto scale

- Automatically provision/de-provision instances to support scale out/scale in
- Applies to App Service Plan
 - Does not apply to Web App, API App directly
- Enables scale by
 - Metric
 - Schedule



Web App Scaling by Instance / CPU (Basic, Standard or Premium Plan)

Scale	FIXITPLAN-A
Autoscale	On
Instances	1

INSTANCES
1

* Scale by

Description Create your own set of rules. Create a schedule that adjusts your instance counts based on time and performance metrics.

Default, scale 1 - 2

- CPU Percentage > 80 (increase count by 1)

Settings CPU Percentage < 60 (decrease count by 1)

[Add Rule](#)

[Add Profile](#)

Web App Auto-Scaling – Adding Metrics

New metrics also allow
scale up and scale down
rules

The screenshot displays the AWS Auto Scaling console interface. On the left, a timeline shows the number of instances (1) from September 15 to 21. Below the timeline, the scaling configuration is shown:

- Scale by:** schedule and performance rules
- Description:** Create your own set of rules. Create a schedule that adjusts your instance counts based on time and performance metrics.
- Default, scale 1 - 2:** CPU Percentage > 80 (increase count by 1)
- Settings:** CPU Percentage < 60 (decrease count by 1)
- Buttons:** Add Rule, Add Profile

On the right, a configuration panel for the selected resource 'fixitplan-a (serverfarms)' is shown:

- * Resource:** fixitplan-a (serverfarms)
- Please choose a metric:** CPU Percentage (highlighted)
- Memory Percentage**
- Disk Queue Length**
- Http Queue Length**
- Data In**
- Data Out**
- Duration (minutes):** 10
- Time aggregation:** Average
- * Action:** increase count by
- Value:** 1

Demo

API Apps on App Service

App Service Environment (ASE)

Deployment of App Service in your VNet



Private location

Supports very large scale

Isolation and secure network access

Liberal majority government

ALERTS 11:19 p.m. CBC News projects Kent Hehr (LIB) elected in Calgary Centre [Show all alerts](#)

National Results

RIDING TOTALS		VOTE SHARE	
MAJORITY	ELECTED	LEADING	TOTAL
	184	0	184
	99	0	99
	44	0	44
	10	0	10
	1	0	1

ELECTED	LEADING	TO COME: 0 RIDINGS

[Show all parties](#)

Riding Results

Search by postal code or riding name

MY FAVOURITES **RIDINGS TO WATCH** **CANDIDATES TO WATCH**

Top Ridings to Watch
[Show why these ridings matter](#)

- Ajax
- Beauport-Côte-de-Beaupré-Île d'Orléans-Charlevoix
- Berthier-Maskinongé
- Brampton Centre
- Brome-Missisquoi
- Burnaby North-Seymour

	184 seats 39.5% of vote		99 seats 31.9% of vote		44 seats 19.7% of vote		10 seats 4.7% of vote		1 seat 3.5% of vote
--	----------------------------	--	---------------------------	--	---------------------------	--	--------------------------	--	------------------------

Select riding or enter postal code

Seats and popular vote: 2015
 338 seats, 17,552,402 votes

Party	Seats	Popular Vote
LIB	184 (54%)	39.5% (6,928,514)
CON	99 (29%)	31.9% (5,597,565)
NDP	44 (13%)	19.7% (3,460,288)
BQ	10 (3%)	4.7% (818,652)
GRN	1 (0%)	3.5% (605,637)

Riding Review
Support for Liberals surges at campaign close

The Duffy trial, the Syrian refugee crisis, the niqab ban, "old-stock Canadians" – these are among the issues that defined a fought 78-day campaign. CBC's Poll Tracker analyzed polls from across the country and monitored how the parties gained through the charted timeline and the linked stories below to see how the issues influenced party support.

The NDP slipped in the polls following the French-language debate, when Mulcair accused Harper of using the niqab as "a distraction." The Conservatives surpassed the NDP in October when they announced a plan to introduce a "hardcore" clause as the details of the Trans-Pacific Partnership deal. Meanwhile, the Liberals observed an uptick in support following Trudeau's performance in the foreign policy debate at the end of September.

By the end of the campaign, support for the Liberals surged to 37.2 per cent, ahead of the Conservatives' 30.9 per cent and

Canadian Broadcasting

OBJECTIVES

Provide real-time results of Canadian election at very high scale (nationwide). While keeping the experience for mobile and desktop devices responsive and impressive

TACTICS

Using App Service Environment to (auto) scale across 3 different geo-regions and utilize (peak time) close to 1300 cores. Used App Service apps to provide API, Web and Mobile experiences

RESULTS

Successfully served 3.6 billion requests over six hours, at a peak of 800K RPS.



Home



About us



Mass Calling



Premium SMS



Mobile Payment



App Solutions

EUROVISION SONG CONTEST

Eurovision Song Contest

12 points go to ... digame mobile. Since 2004 we are the official partner of the European Broadcasting Union (EBU) and responsible for the centralized televoting via telephone, SMS, MHP and for the official Eurovision App, which has been developed by digame mobile.

[mehr](#)



Digame Mobile

OBJECTIVES

Support Live TV with online voting

Provide rich experience, allowing customers to vote, purchase songs, view standing via Mobile and Web

TACTICS

Cloud-based web and mobile apps running on App Service, connecting to backend production systems

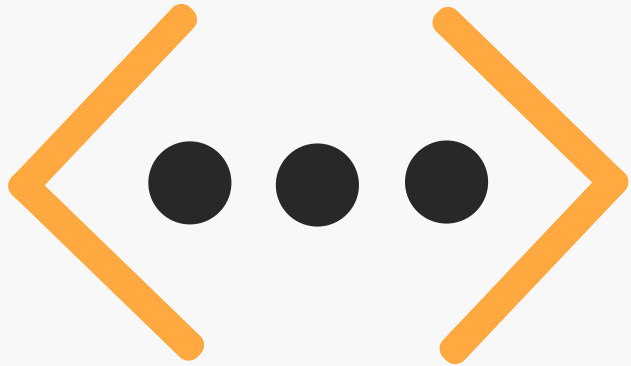
Auto-scale for peak times

RESULTS

Scalable solution supporting very large number of voters

Added real-time feedback to production

Azure Virtual Network(VNet)



Private network in the Azure cloud

Enables network based security and isolation

Can be used with VPNs to create hybrid cloud applications

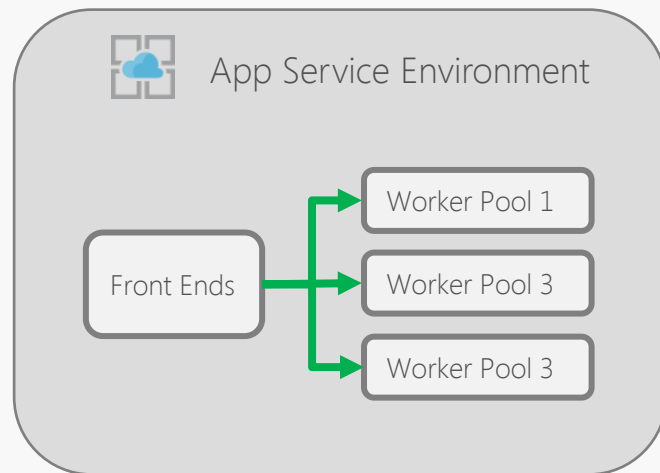
ASE roles

Front Ends:

- HTTP endpoints
- Distribute requests to workers

Worker Pools:

- Host the actual apps
- Can have up to 3 pools of workers
- Can use sizes of P1 through P4
- Can have up to 50 workers



ASE management

In an ASE you control:

- Quantity of Front Ends
- VM size used for Front Ends
- Quantity of Workers in each Worker Pool
- VM size used for each Worker Pool
- Quantity of IP addresses that can be used by individual apps

When to use ASE vs multi-tenant

Do you need....

Network Isolation?

Network level control on inbound or outbound traffic

Very large scale?

Can scale up to 50 instance

More powerful workers?

Can leverage P4 VMs: 8 Cores / 14 GB RAM

More disk space?

Can have up to 500GB of storage

Typical customer scenarios

Tenant isolation

Front-end pool isolated to a single customer

Network isolation

Internal apps in corporate network

WAF (vs Multi-tenant options with IP filtering IIS level)

Public facing apps with VNET integration to backends on Azure or on-premise

Scale

Spikes over and beyond what multi-tenant App Service can offer

Continuous high load

Dedicated front-end load balancer layer

Bigger machine sizes

ASE Front Ends

Front Ends:

- Are the HTTP endpoints
- Perform HTTPS decryption/encryption
- Work with the database to send traffic to the correct workers

Primary resource consumed is normally memory

BUT

If handling many HTTPS requests, may need more CPU

Scaling Front Ends

Monitor CPU

Monitor memory

Plan ahead for periods of high load

Scale out and don't scale up – Add more, not bigger

Autoscale and Front Ends

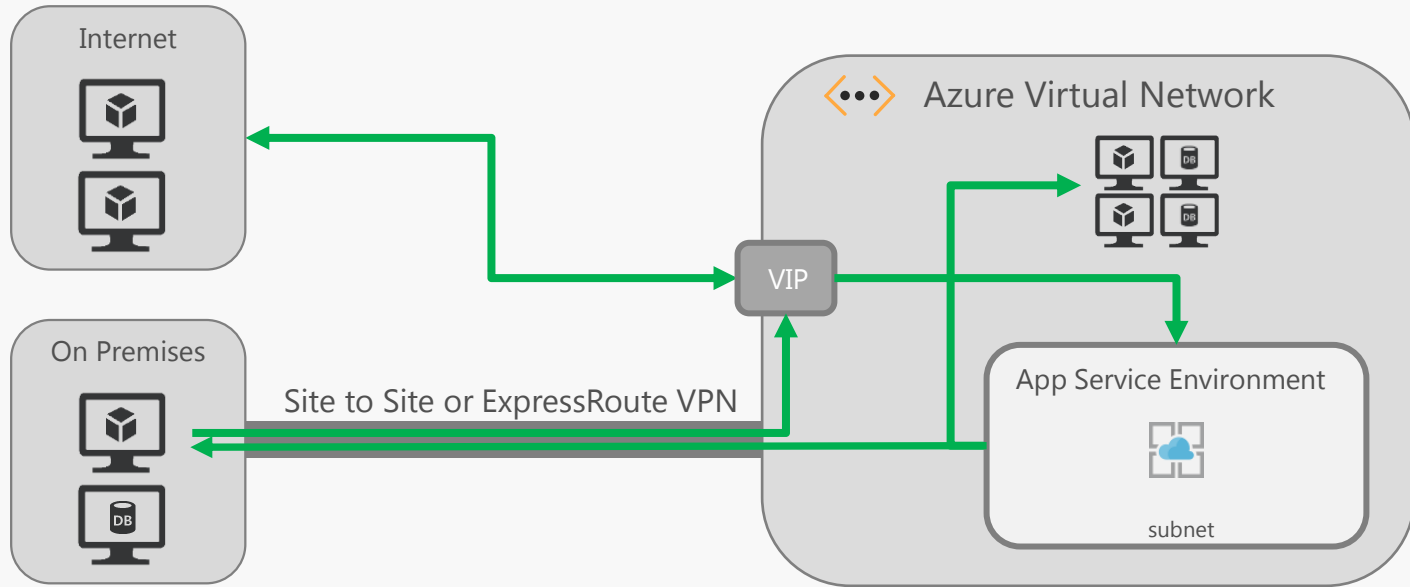
As with worker pool, autoscaling can be done on:

- Scheduled times of day/week
- Based on performance metrics
- Fixed date (one time)

VNet VPN technologies

- ExpressRoute: Enterprise class VPN solution set up with a telco
- Site to site (S2S): Network to network VPN
- Point to site (P2S): Host to network VPN

ASE high level network



Network Security Groups (NSGs)

- Access Control List (ACL) rules to Allow/Deny traffic
- Applies to incoming/outgoing traffic
- Assign NSG to subnet ASE
- Default set of rules assigned when creating new NSG
 - Cannot be removed, only overwritten
- By default, ASE does not have NSG assigned

VIP Types

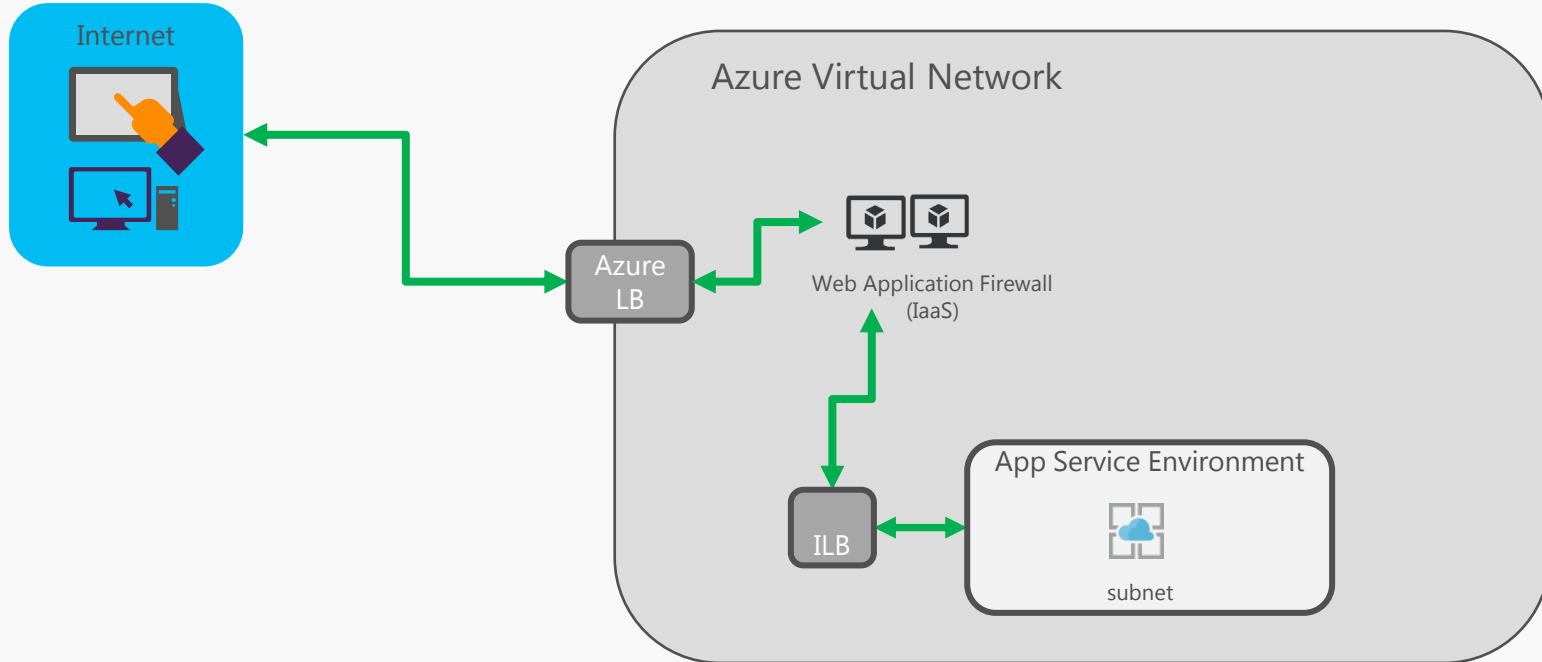
An ASE can be deployed with either an:

External VIP – enables hosting internet accessible apps

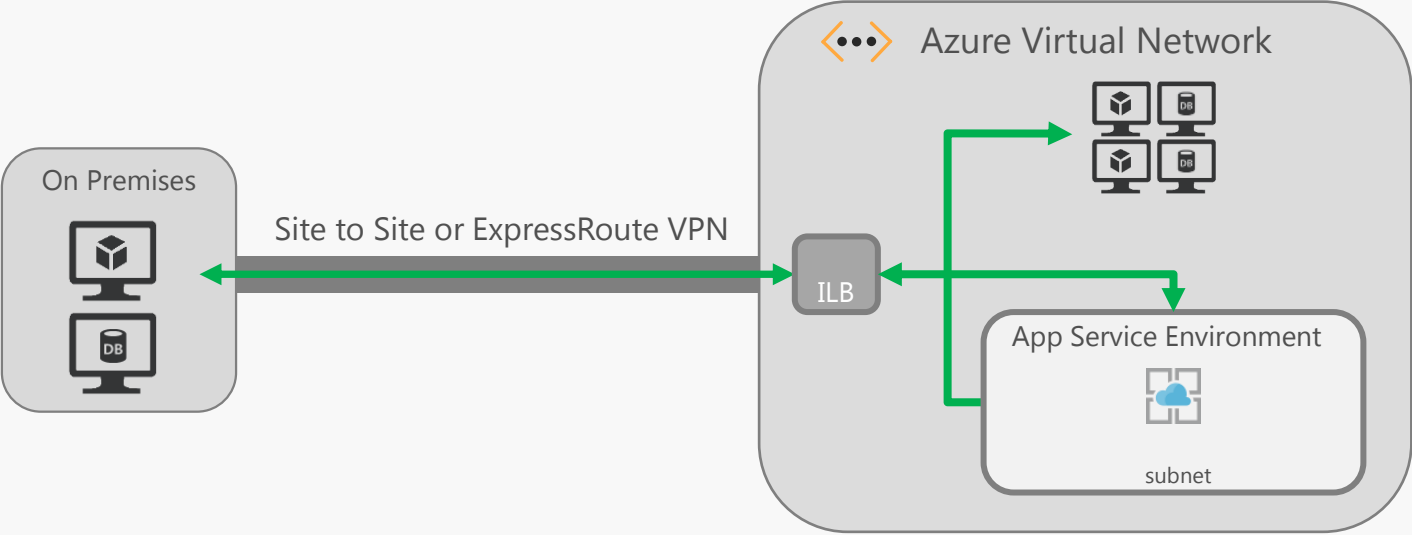
All examples thus far were with an External VIP

Internal VIP – enables hosting apps on a VNet IP address. This uses an Internal Load Balancer(ILB) instead of a publicly addressable VIP

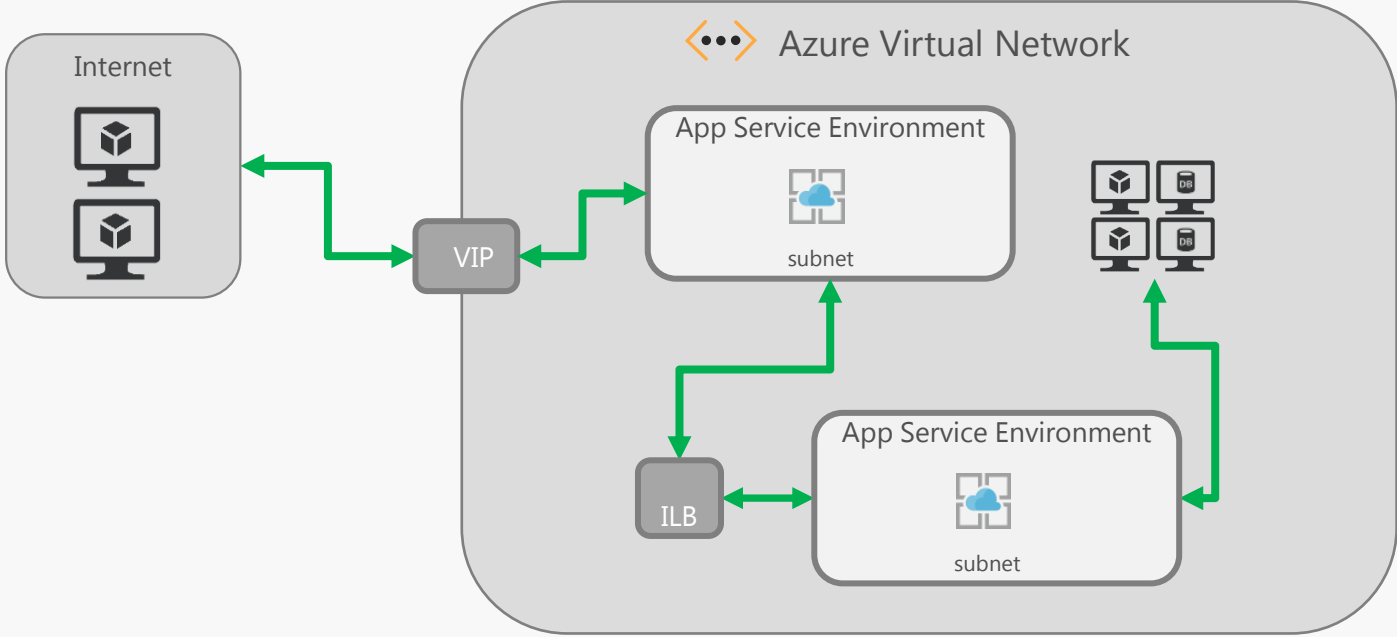
Scenario: Web Application Firewall



ILB ASE – Intra-net app



ILB ASE – 2 tier application



Demo

App Service Environment

Integrating APIs

with Azure Logic Apps

Microsoft Enterprise Integration

API Management



- Create API frontends for backends in the cloud and on-premises
- Protect, monitor and monetize the APIs
- Onboard external and internal app developers

BizTalk Server



- Automate mission critical processes
- Integrate on-premises LOB to SaaS apps
- Support for new Microsoft Platforms

Logic Apps



- Connect SaaS applications
- Serverless flows, auto scaling, fully managed.
- Leverage Azure Services

Service Bus



- Cloud-based messaging service
- Connect to the cloud
- Secure and predictable message delivery

Automate business processes in the cloud with Logic Apps

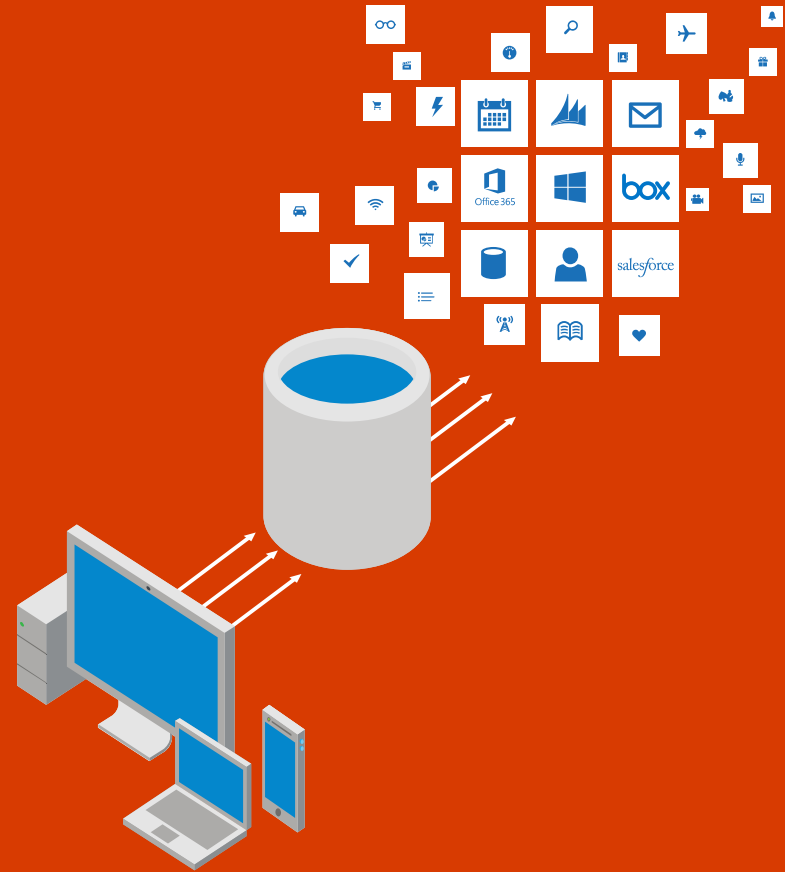
Create business processes and workflows visually

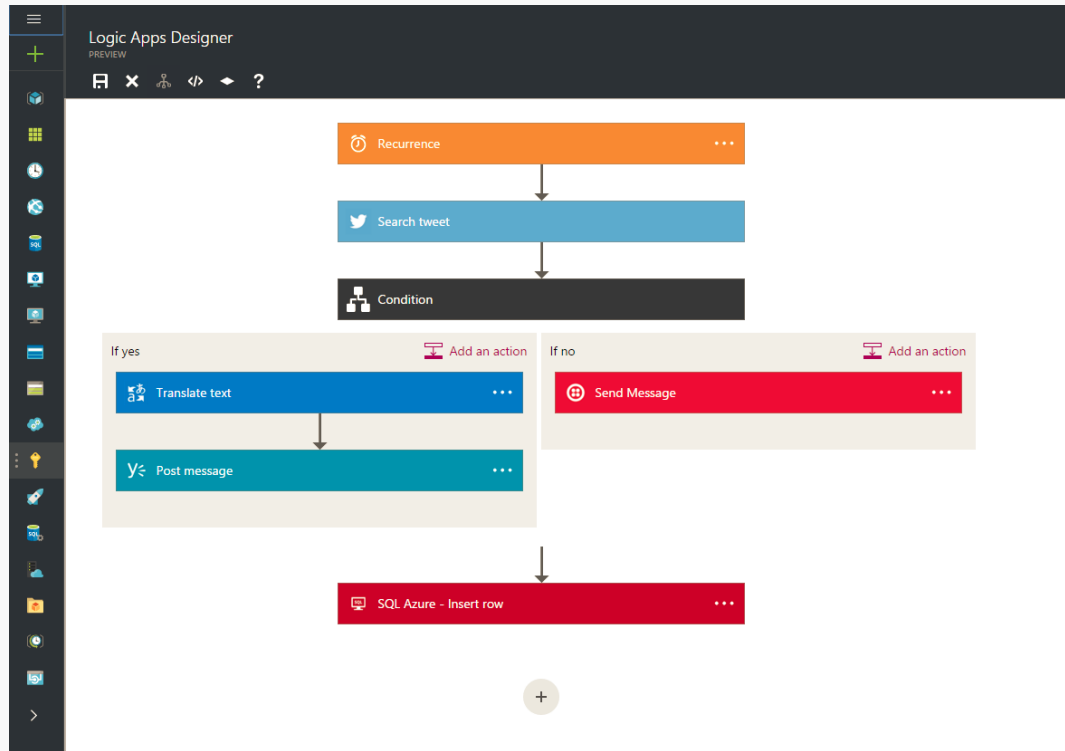
Integrate with SaaS and enterprise applications

Unlock value across on-premises and cloud

Automate EAI, B2B/EDI, and business processes

Leverage Integration Platform as a Service (iPaaS)





Visual designer based on declarative language
Stock library of SaaS and format connectors
Leverage Azure Functions as custom steps
Invoke with a timer and Web hooks
Scalable runtime

Out-of-box connectors

SaaS

- Azure Service Bus
- Azure Storage Blob
- Bing Search
- Box
- Dropbox
- Dynamics CRM Online
- Facebook
- GitHub
- Google Calendar
- Google Drive
- Google Sheets
- Google Tasks
- Instagram
- MailChimp
- Mandrill
- Microsoft Project Online
- Microsoft Translator
- Office 365
- Office 365 Users
- OneDrive
- OneDrive for Business
- Outlook.com
- Project Online
- Salesforce
- SendGrid
- SharePoint Online
- Slack
- SparkPost
- SQL Azure
- Trello
- Twilio
- Twitter
- Wunderlist
- Yammer
- YouTube

Protocols

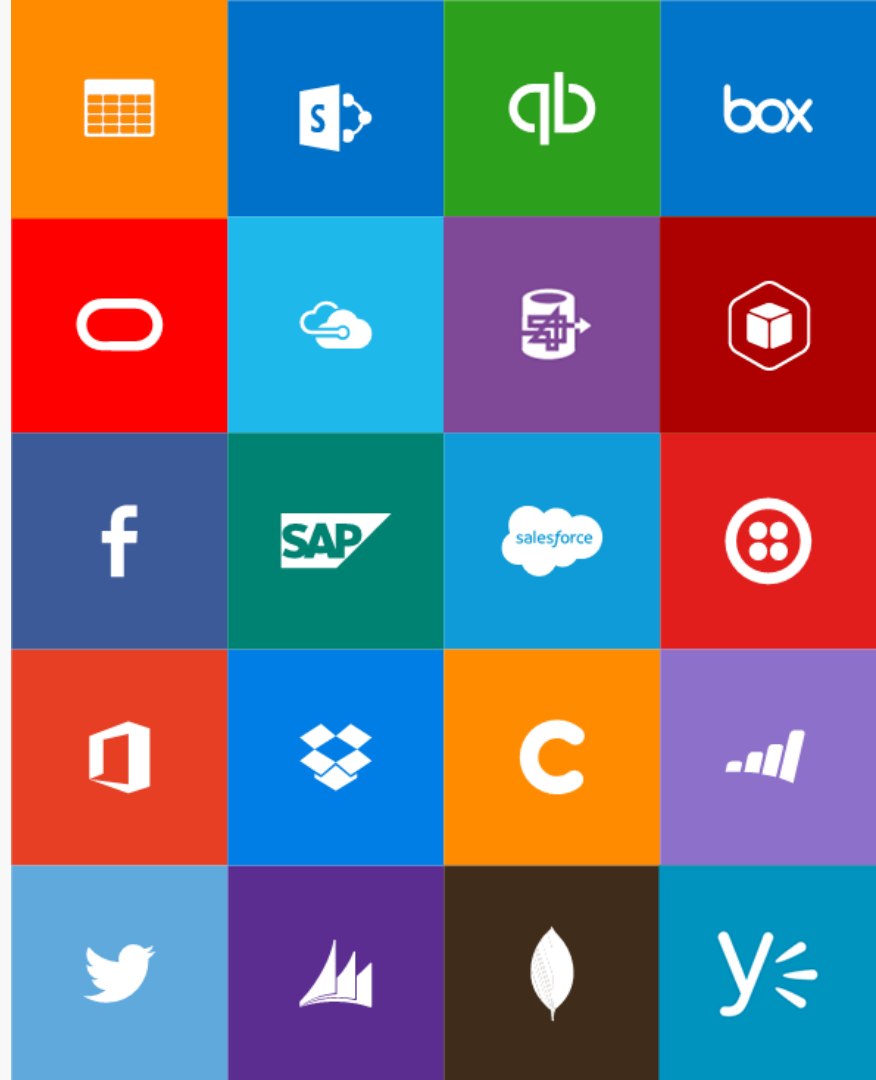
- HTTP, HTTPS
- HTTP Webhook
- FTP, SFTP
- SMTP
- RSS
- Delay
- Workflow

Enterprise messaging

- Validate
- Transform (+Mapper)
- Convert (XML-FF)
- X12
- AS2

Hybrid

- SQL Server



Upcoming connectors

SaaS

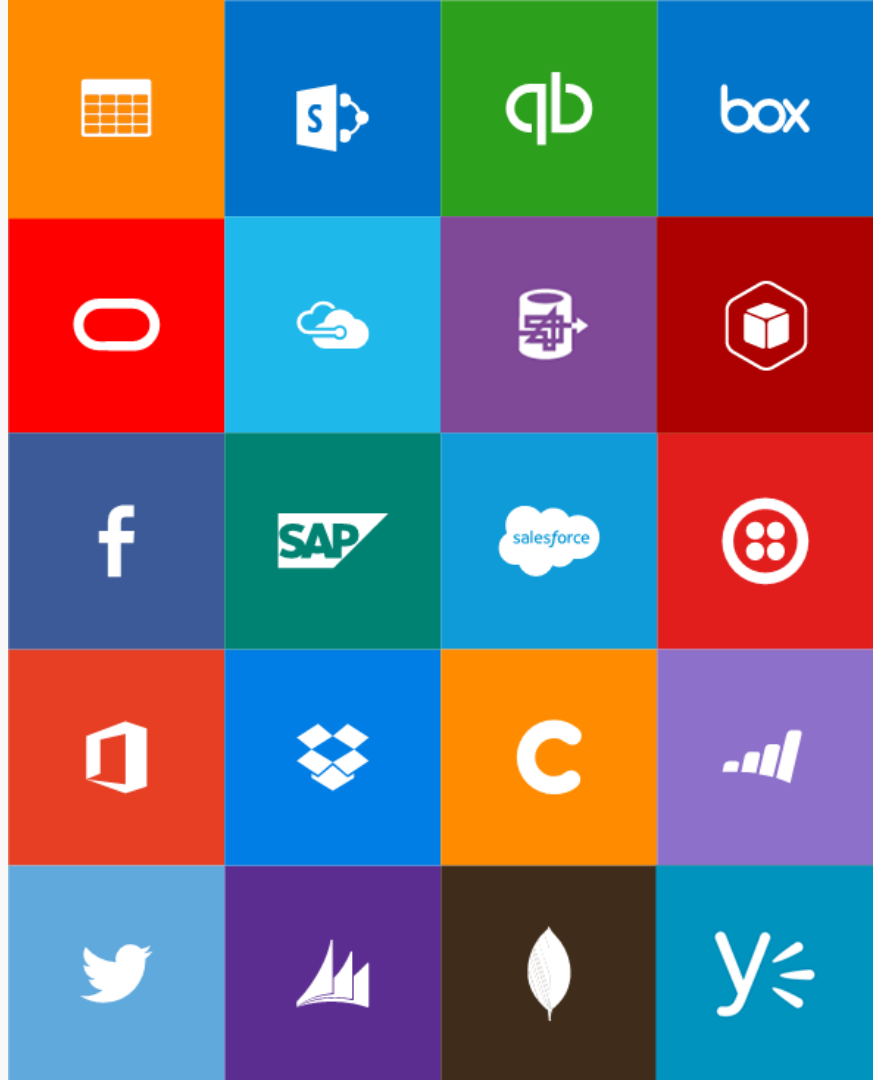
- DocuSign
- Google Mail
- Lithium
- OneNote
- Stripe
- Bing Search
- UserVoice
- Trello
- VSTS
- ZenDesk

Hybrid

- IBM DB2
- Informix
- SharePoint Server
- Oracle DB
- File
- Websphere MQ
- SAP
- Oracle EBS
- CICS

Enterprise messaging

- EDIFACT
- Party resolution



Logic Apps for Orchestration Across API Apps

Implicitly – when output referenced

Explicit “dependsOn” condition

Explicit “expression” condition - function

The screenshot displays a Logic App workflow configuration. The top step is a trigger named "When a new tweet appears" with a Twitter icon. Below the trigger, the "QUERY TEXT" field contains "#MicrosoftAzure". A status bar indicates it is "Connected to Twitter" with a "Change connection" link. A downward arrow points to the second step, "Create file", which has a Dropbox icon. This step is configured with a "FOLDER PATH" of "/", a "FILE NAME" of "tweet.txt", and "FILE CONTENT" selected as "Tweet text" from a dropdown menu. Below the configuration, a section titled "You can insert data from previous steps..." shows "Outputs from When a new tweet appears" with buttons for "Tweet text", "Tweeted by", "Tweet id", "Created at", and an ellipsis. The bottom status bar shows it is "Connected to Dropbox" with a "Change connection" link.

Triggering a Logic App "Run"

Recurring schedule – "every X hours"

Polling an API for a response

A 200 response means "run" – a 202 response means "wait"

Can use trigger state to get information on the previous execution

Manual Webhook

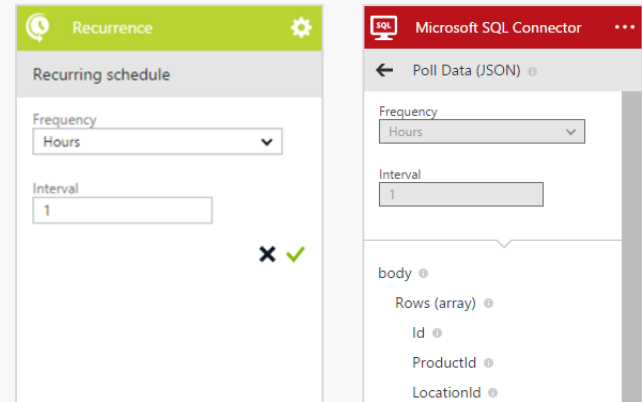
Every workflow has an endpoint you can POST to from any web service using a SAS URL

Webhook subscription

Subscribe to a web services webhook

On Demand

User can click the "Run Now" button in the portal



Logic Apps operations

Response

SplitOn

Retry

Scopes

Conditionals

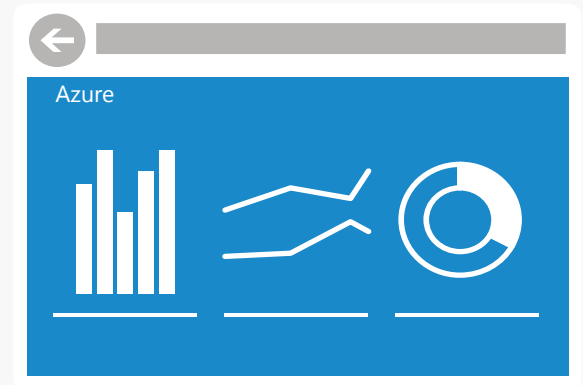
Repeating / iterating over list

Do...until

...

Debugging and History

- Trigger history shows polling results
- Run history shows details for each logic app run
- Monitoring view shows step by step results
- Diagnostics & Alerts



Demo

Logic Apps

What is Azure Functions?

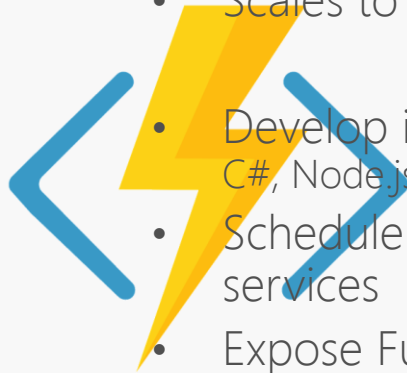
Code Azure Functions Events + data

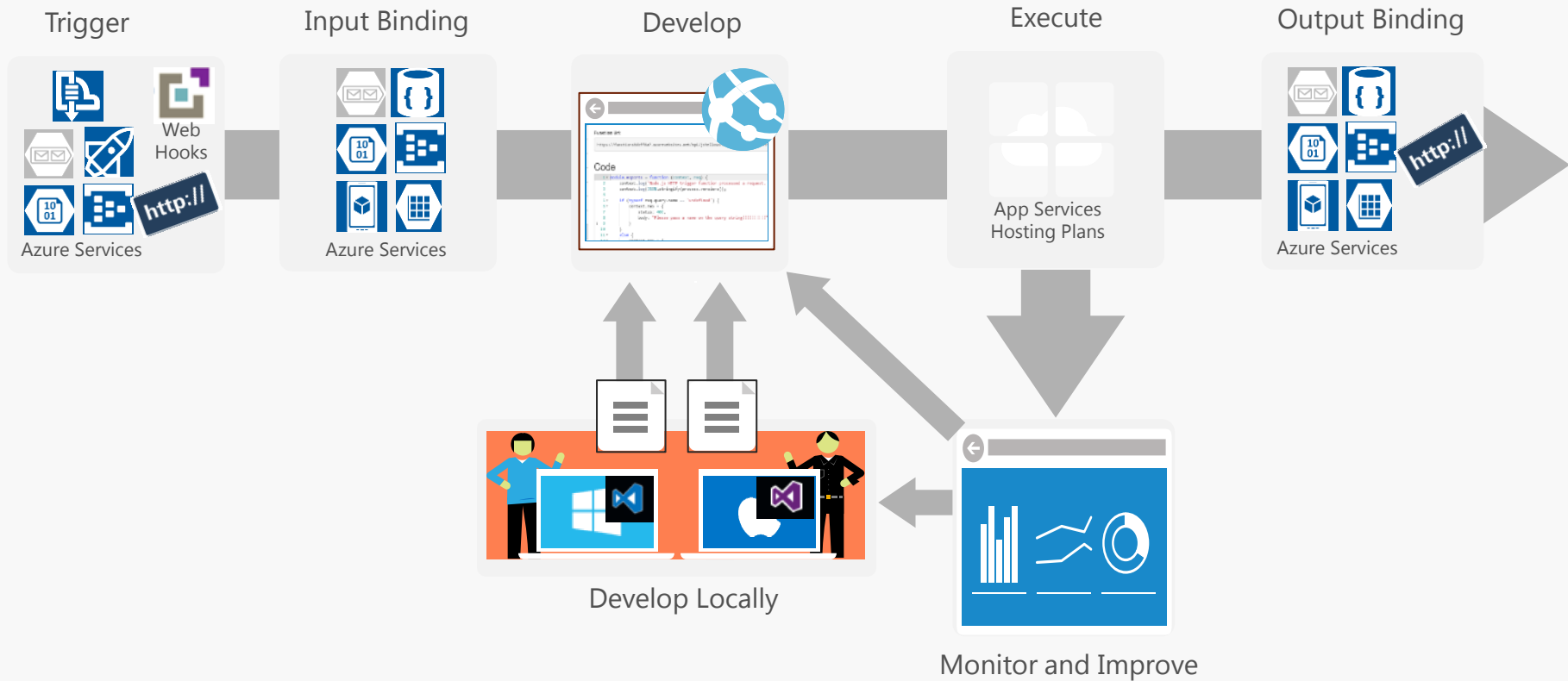


What is Azure Functions?

Azure Functions

- Cloud-scale Event Handlers in no time
- Composing cloud apps becomes simple
- Scales to demand & pay for what you use
- Develop in:
C#, Node.js, Python, PHP, and more
- Schedule event-driven tasks across services
- Expose Functions as HTTP API endpoints
- Fully Open Source
- Running on Serverless Infrastructure





Demo

Functions + Logic Apps

Exposing & monetizing APIs

with API Management

APIs as products

How do you make your ledger

How do you protect your

How do you enforce

How do you

How do you

How do you measure use and impact?

API Management

How do you enforce policies?

How do you

Azure API Management

API consumers



AZURE API MANAGEMENT

Abstraction

decouple
modernize
optimize

...

Middleware

secure
protect
cache

...

Monitoring

usage
health
monetization

...

Developer

discover
document
on-board

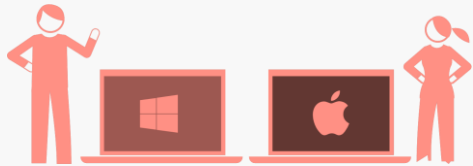
...

APIs on Azure

Azure APIs

On-prem APIs

3rd party APIs



APP DEVELOPERS



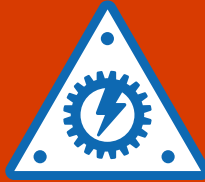
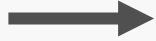
AZURE API MANAGEMENT



Developer Portal



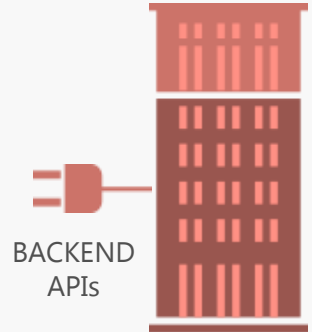
APPS



Gateway



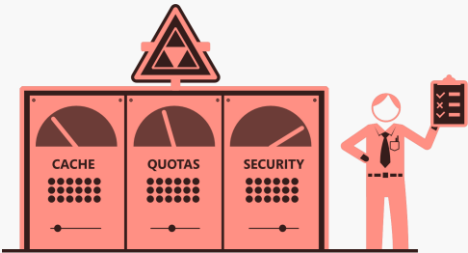
DIRECT OR VPN



BACKEND APIs

Hosted **anywhere**.

Developed using **any** technology.



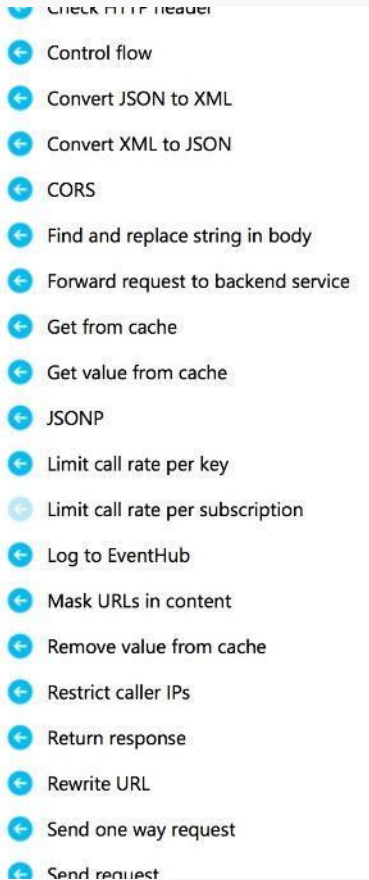
API PUBLISHERS



Publisher portal



Policies and expressions



Policies

Encapsulate common API management capabilities

Mutate request context or change API behavior

Can be set in the inbound and outbound directions

Have global, product, API or operation scope

Expressions

C# "snippets" embedded in policies

Have read-only access to the request context

Can use whitelisted .NET types only

Are used to dynamically configure policies

Demo

API Management

Azure API Management Product Roadmap



Public

👁 Subscribed

⋮ Show Menu

About

API management powered by
Microsoft Azure

☰ 1

How do I use this board?

☰ 2

How do I propose a feature?

☰

Backlog

Improved Policy Editor UI

☰

Bring your own cache (BYOC)

☰

Version APIs

☰

Manage API lifecycle

☰

Deploy gateway on-premises

☰ 1

Validate requests and responses (OWASP)

☰

Capture request traces on demand

☰

Setup and receive alerts

☰

Monetize APIs

☰

Enterprise API catalog

☰ 1

Visual Studio integration

☰

Next up

Enforce role-based access for admins

☰

Import API from an API App

☰

In Progress

Import API from a Logic App

☰

Query gateway log entries via
management API

☰

Enable internal only access to API
Management

☰

Join Azure Resource Manager VNET

☰

Azure "Classic" portal UX in the new
Azure portal

☰

Publisher portal UX in the new Azure
portal

☰

"Restify" SOAP backends

☰ 5

Import API from a Function App

☰

Done!

Publish SOAP backends

☰

Transform content using XSLT policy

☰

🗨 1

Retry a group of policies

☰

🗨 3

Manage API Management service
instance using GIT

☰

Manage and create values and
secrets using properties

☰

Customize developer portal pages
using templates

☰

Custom caching policy

☰

Flexible request throttling policies

☰

<http://aka.ms/apimroadmap>

Key takeaways

Swaggerize your APIs

Integrate with Logic Apps

Use functions for serverless patterns

Expose with API Management

Resources

[App Service](#)

[Logic Apps](#)

[Azure Functions](#)

API Management

Learn more - <http://aka.ms/apimdocs>

Get in touch - <http://aka.ms/askapim>

Follow progress - <http://aka.ms/apimblog>,
<http://aka.ms/apimroadmap>

Propose a feature - <http://aka.ms/apimwish>

Contact our team

Planning interesting project on App Service Environment, API Management, Logic Apps, Service Fabric?

Want some support or visibility?

bzass@microsoft.com



.NET **DEVELOPER** **DAYS**

net.developerdays.pl
[@DeveloperDaysPL](https://twitter.com/DeveloperDaysPL)

Sponsors and Partners

Strategic Sponsors



Gold Sponsors



Silver Sponsors

