



# I Find your lack of Azure Function Disturbing

Tiago Costa

“If having a coffee in the morning  
doesn’t wake you up, ...”

“... try to delete a production resource group”

# Tiago Costa

Cloud Architect and Advisor

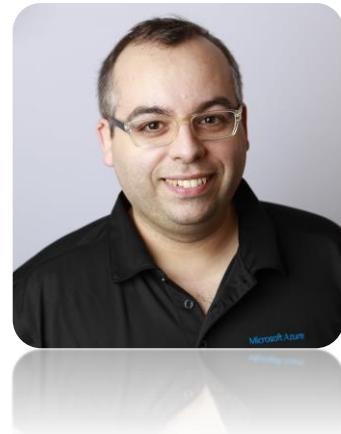
Trainer | Author | Speaker

Independent Contractor

MVP – Microsoft Azure

MCT – Microsoft Certified Trainer

MCT Regional Lead



[tiago.costa@tiagocosta.com](mailto:tiago.costa@tiagocosta.com) | [www.tiagocosta.com](http://www.tiagocosta.com) | [@tiagocostapt](https://twitter.com/tiagocostapt)

# Agenda

Why Azure?

Cloud Architecture – Serverless

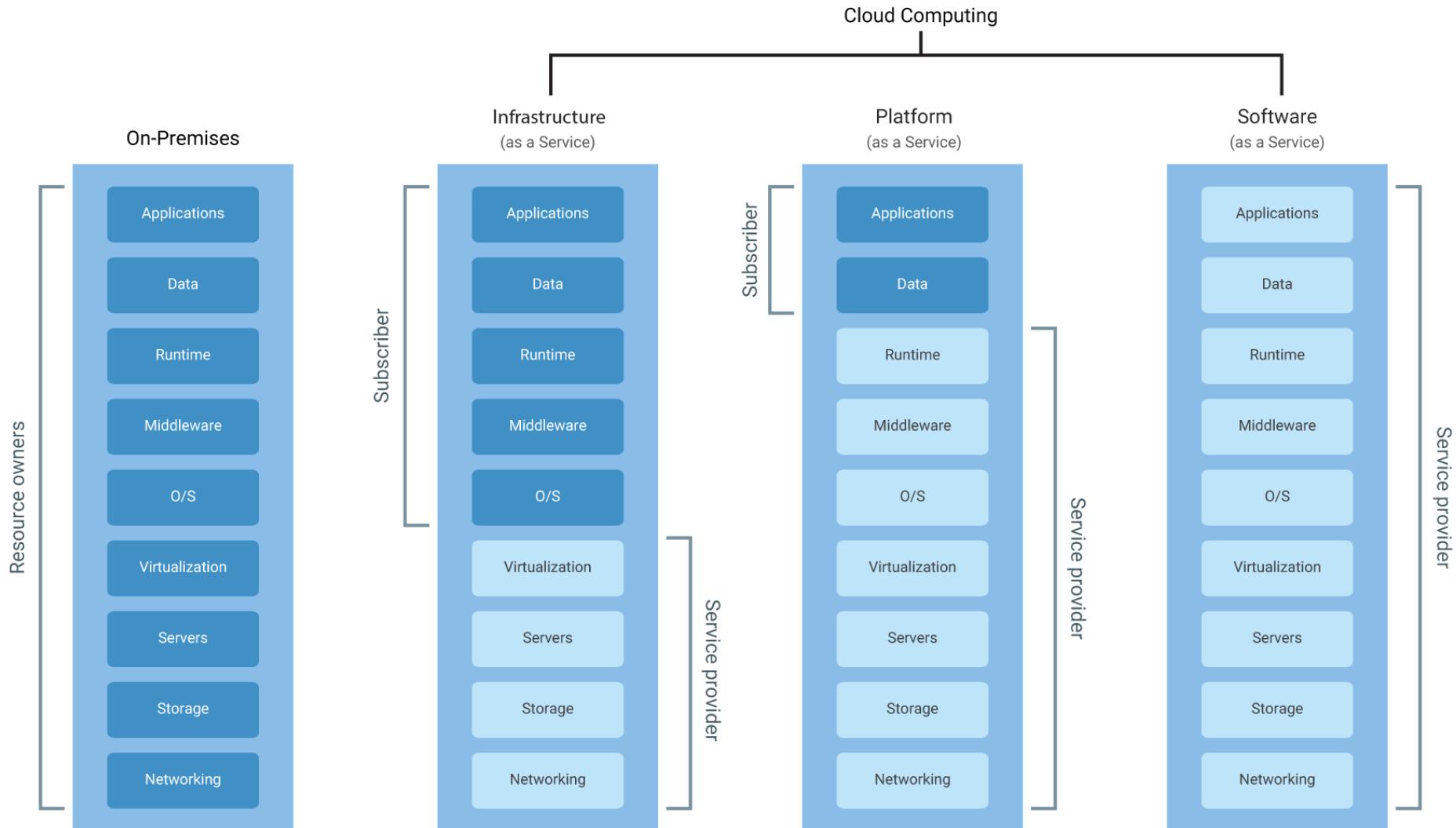
Azure Functions

Durable Functions



Why Azure?

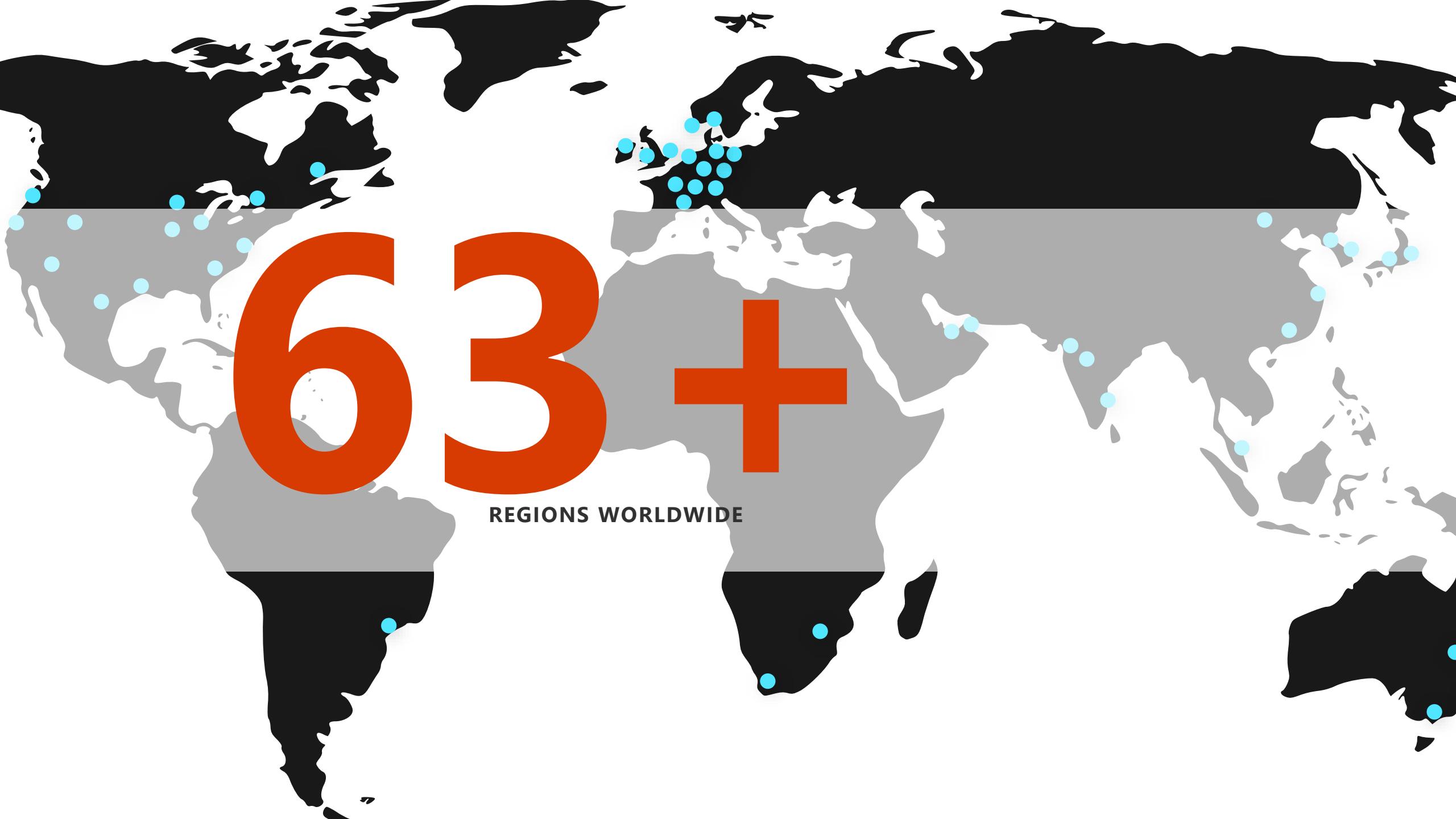
# Cloud Models





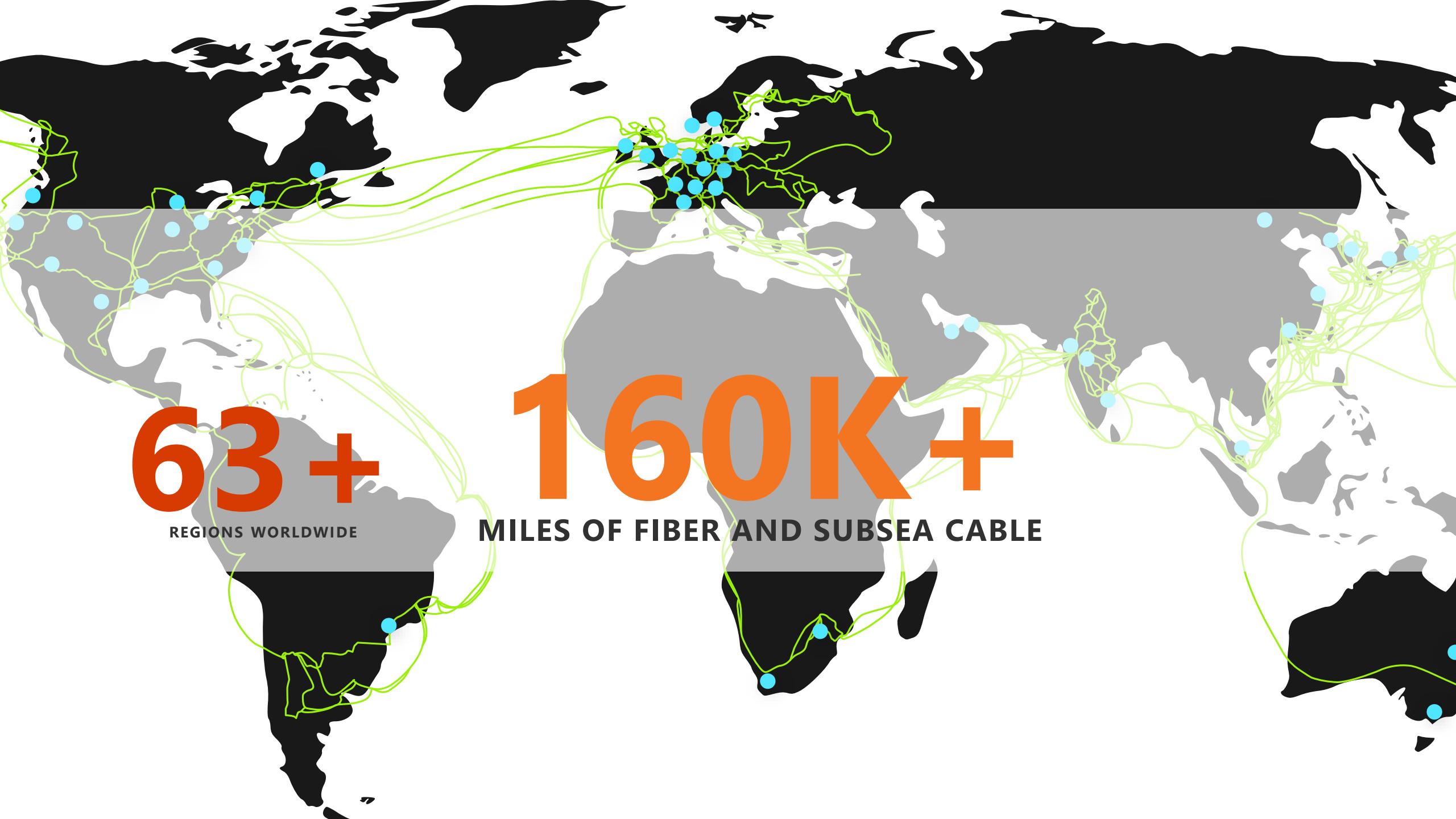
# 2nd

LARGEST PUBLIC CLOUD PROVIDER



**63+**

REGIONS WORLDWIDE

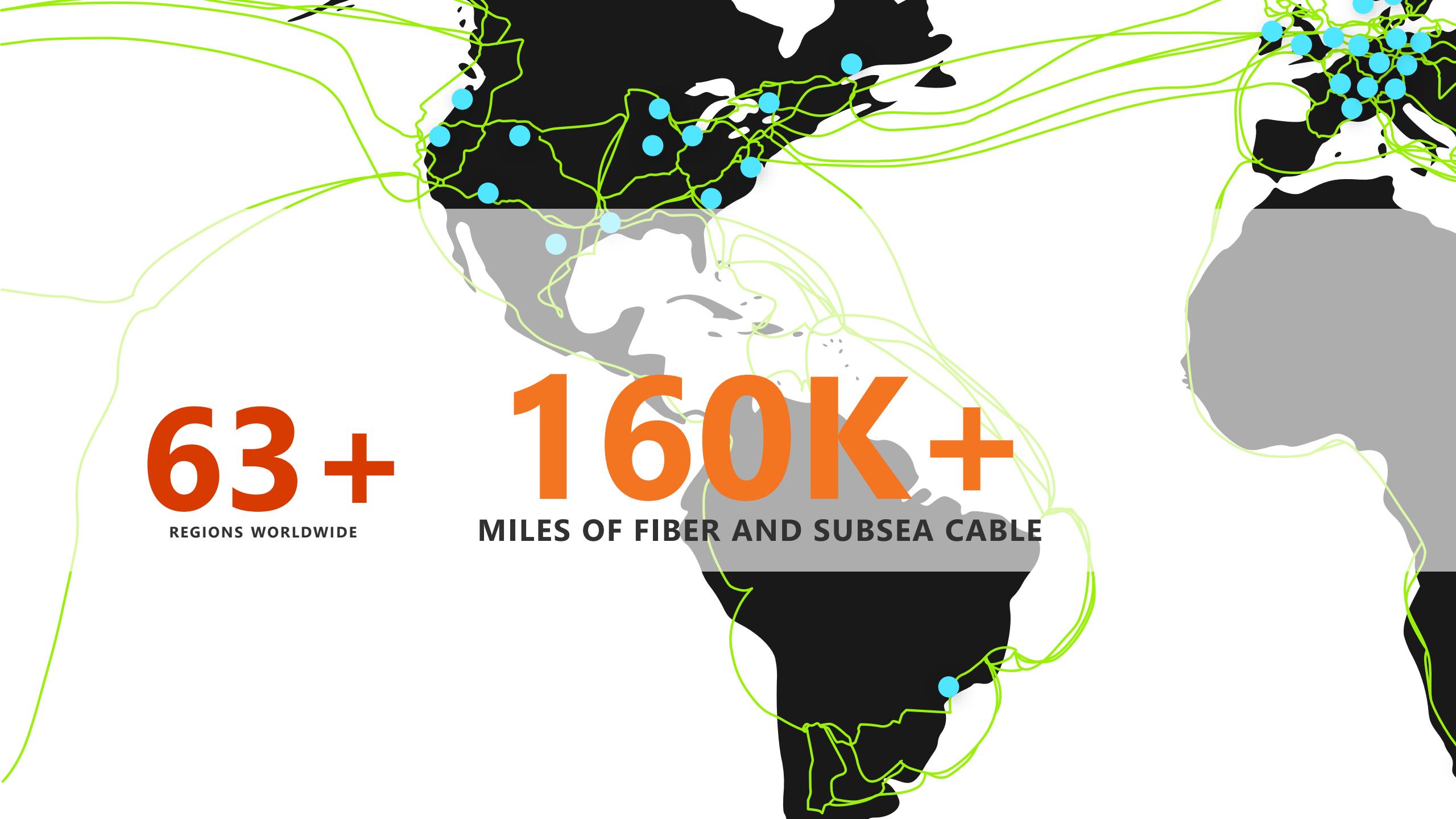


**63+**

REGIONS WORLDWIDE

**160K+**

MILES OF FIBER AND SUBSEA CABLE

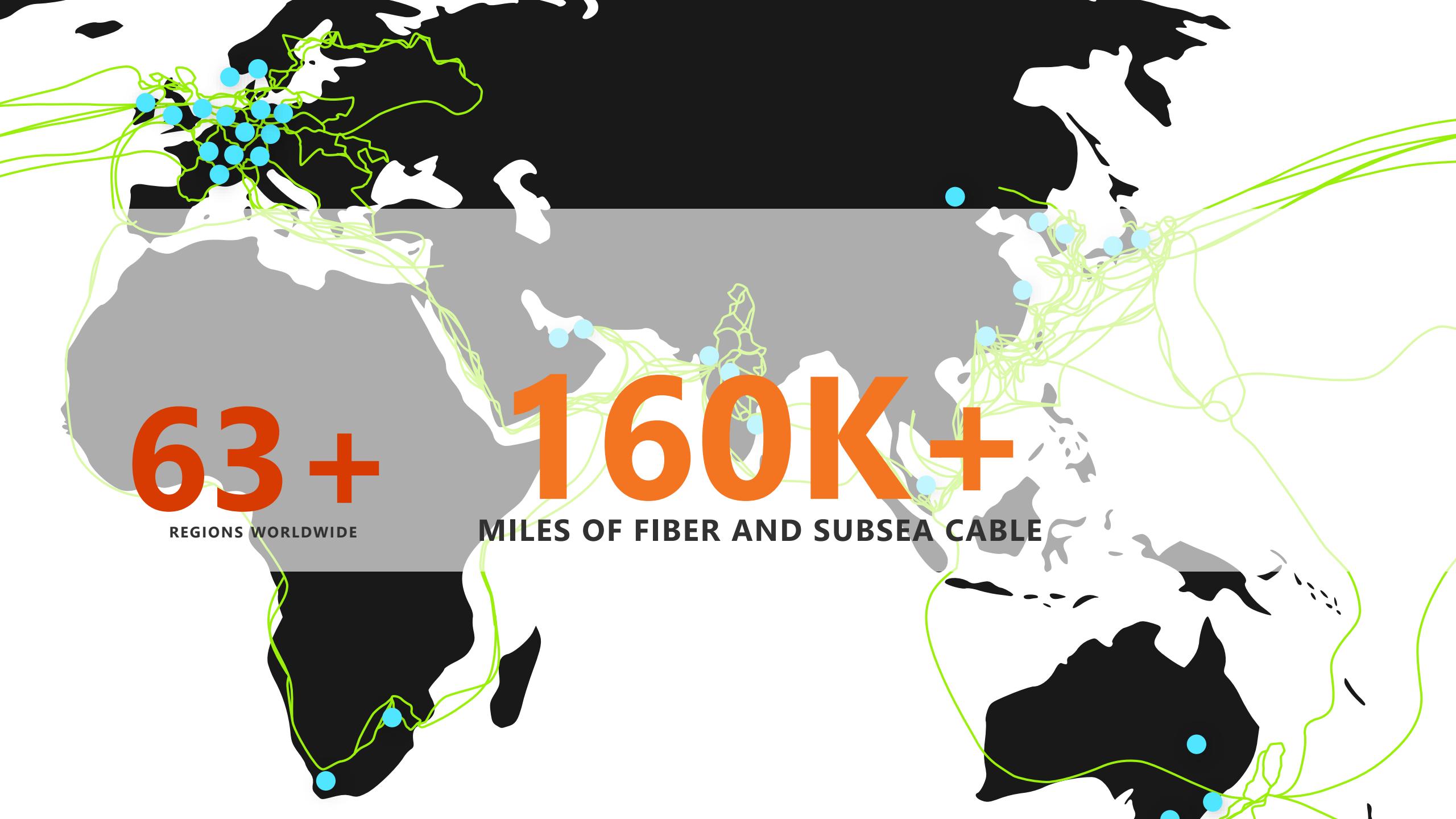


**63+**

REGIONS WORLDWIDE

**160K+**

MILES OF FIBER AND SUBSEA CABLE

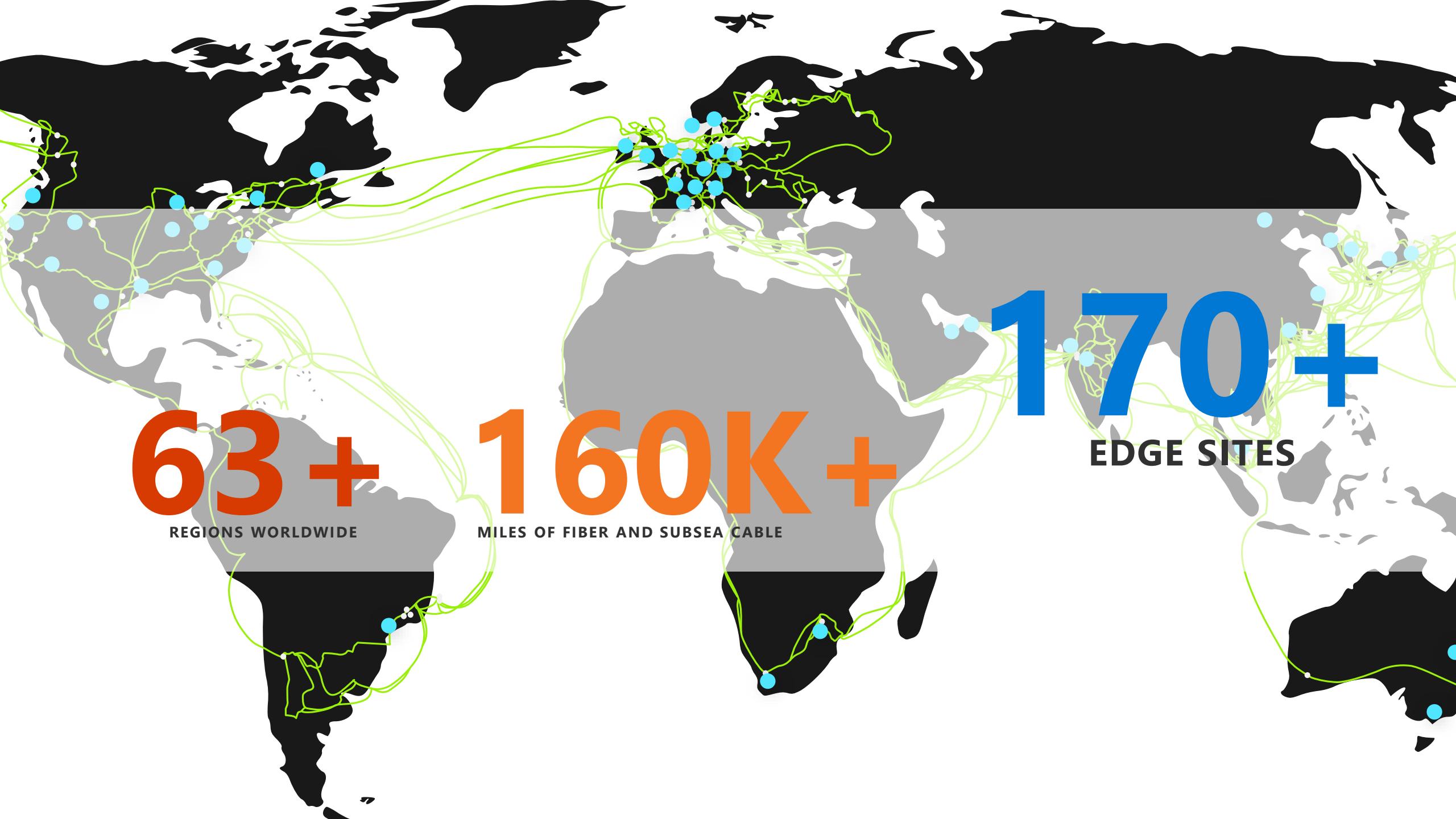


**63 +**

REGIONS WORLDWIDE

**160K+**

MILES OF FIBER AND SUBSEA CABLE



**63+**

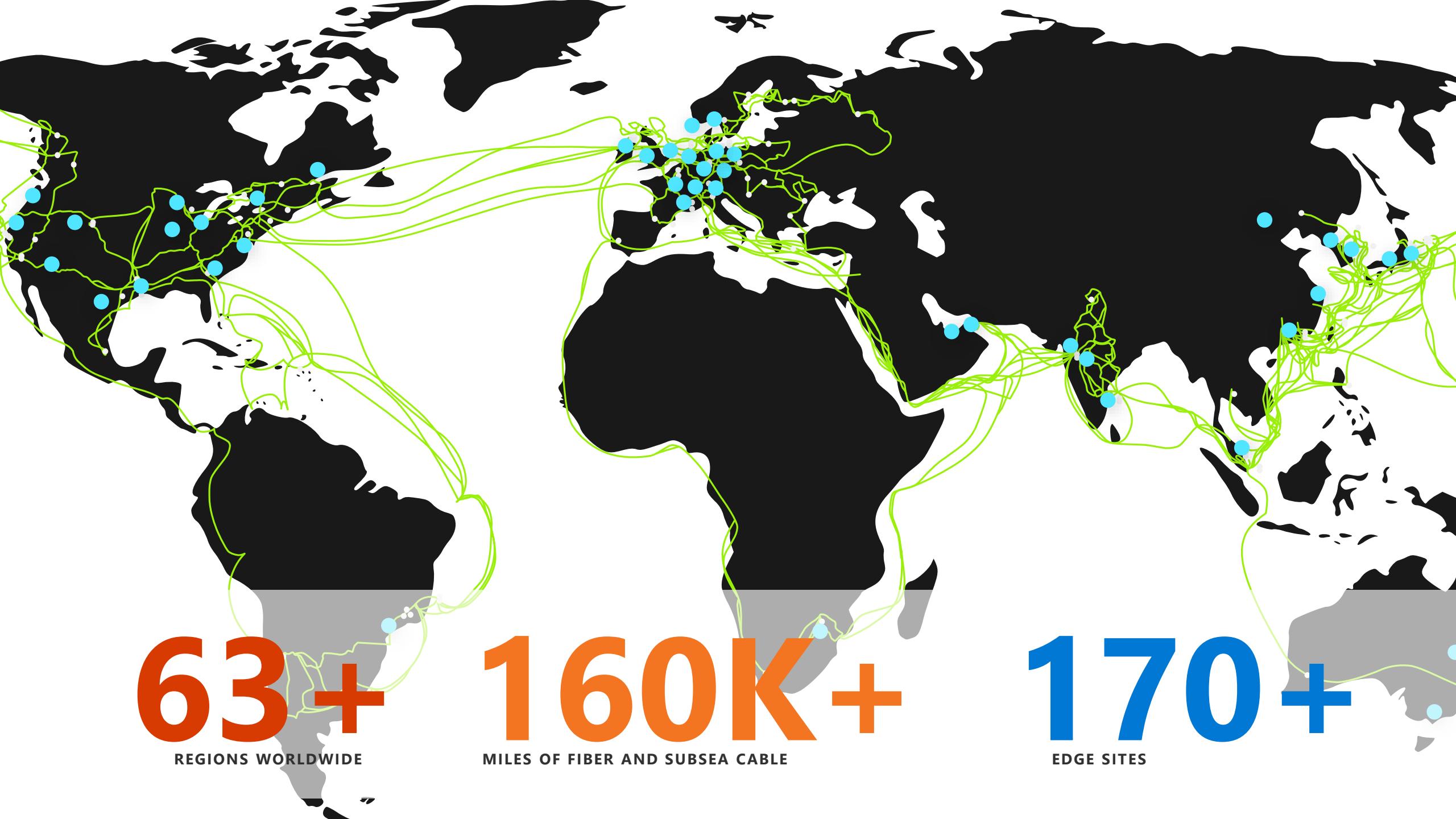
REGIONS WORLDWIDE

**160K+**

MILES OF FIBER AND SUBSEA CABLE

**170+**

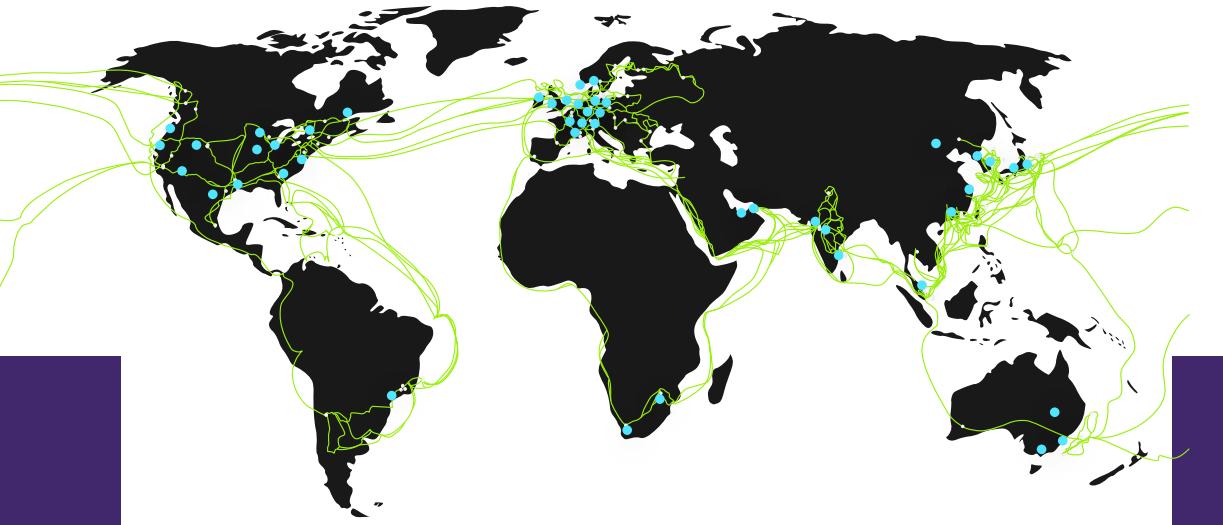
EDGE SITES



**63+**  
REGIONS WORLDWIDE

**160K+**  
MILES OF FIBER AND SUBSEA CABLE

**170+**  
EDGE SITES



---

**100 Gbps - fastest private**  
connection of any public cloud

1.6

million miles of  
fiber

 Datacenter

 CDN Locations

 Edge Node

 Internet  
Exchange

 Terrestrial Network

 Subsea Network

Virginia (2014)



# Virginia (2017)



# Azure Regions



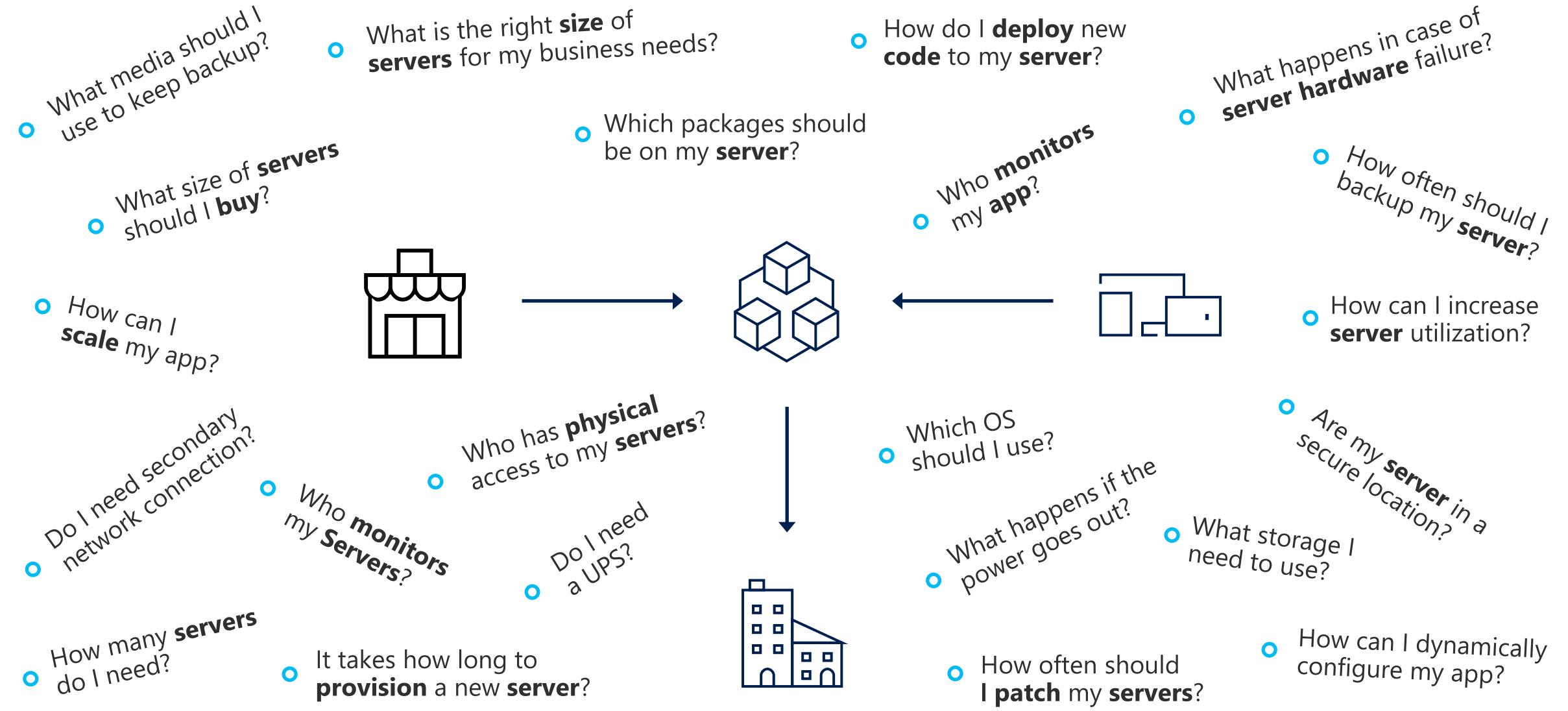
# Why Azure

- Huge demand from customers (Starting now in some geography's). Huge across Europe and USA.
- Fun to work with! Every project is a different project and we can use new services and/or new feature of already exiting services
- Pay for what you use



Serverless???

# Serverless



On-premises

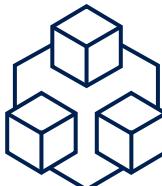
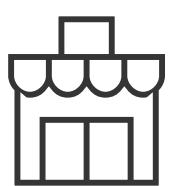
The "evolution" of application platforms

What is the right **size** of **servers** for my business needs?

How can I increase **server** utilization?

How many **servers** do I need?

How can I **scale** my app?



How often should I **patch** my **servers**?

How often should I backup my **server**?

Which packages should be on my **server**?

How do I **deploy** new **code** to my **server**?

**Which OS** should I use?

Who **monitors** my app?



On-premises

IaaS

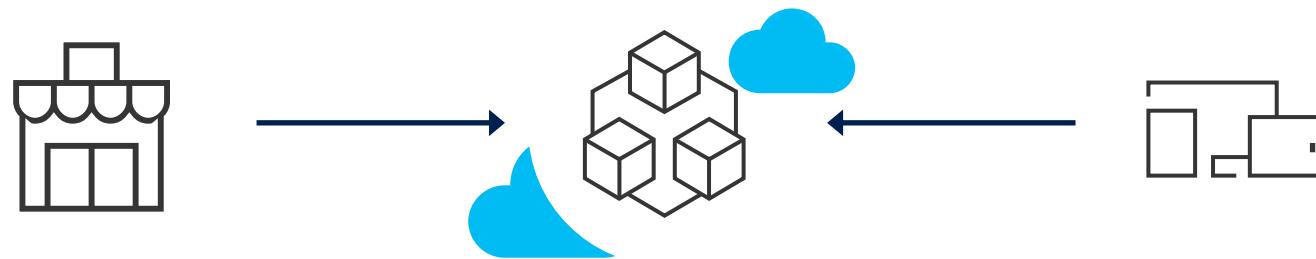
The “evolution” of application platforms

What is the right **size** of “**servers**” for my business needs?

How can I increase “**server**” utilization?

How many “**servers**” do I need?

How can I **scale** my app?



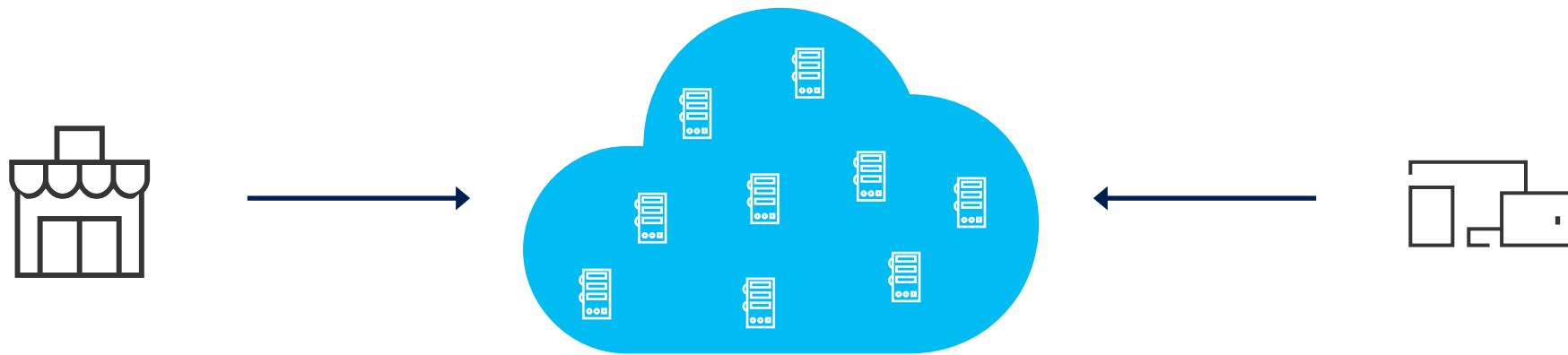
On-premises

IaaS

PaaS

The “evolution” of application platforms

How do I **architect** my app?

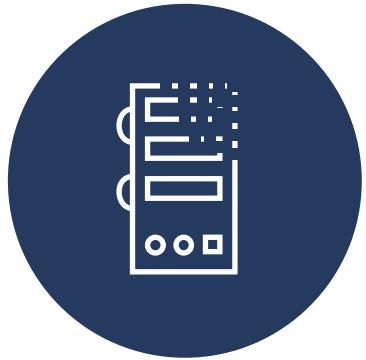


Serverless, the platform for cloud native apps

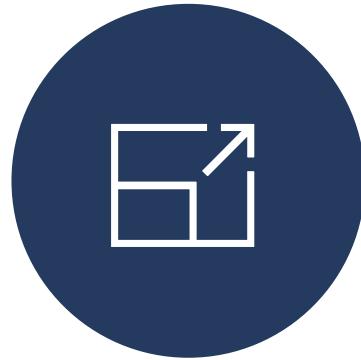
On-Premises                      IaaS                      PaaS                      **Serverless**

The “evolution” of application platforms

# What is Serverless?



Abstraction  
of servers

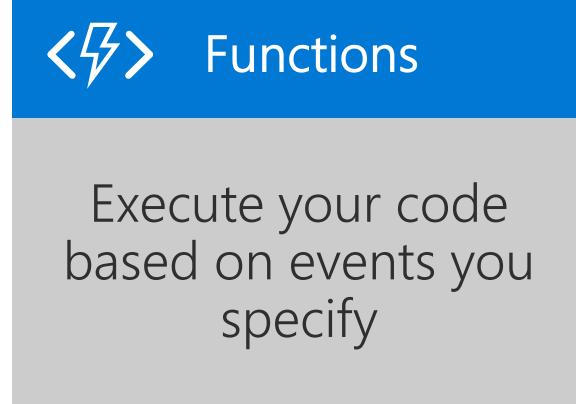


Event-driven/  
instant scale



Micro-billing

# Azure serverless platform components



# Azure serverless platform components

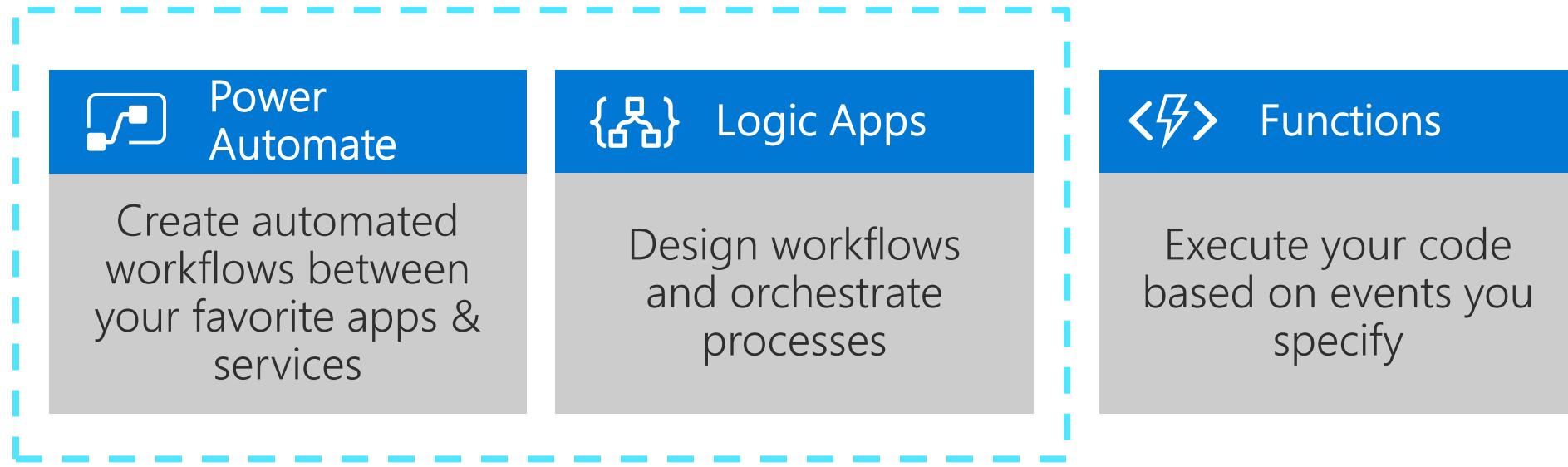
## {} Logic Apps

Design workflows  
and orchestrate  
processes

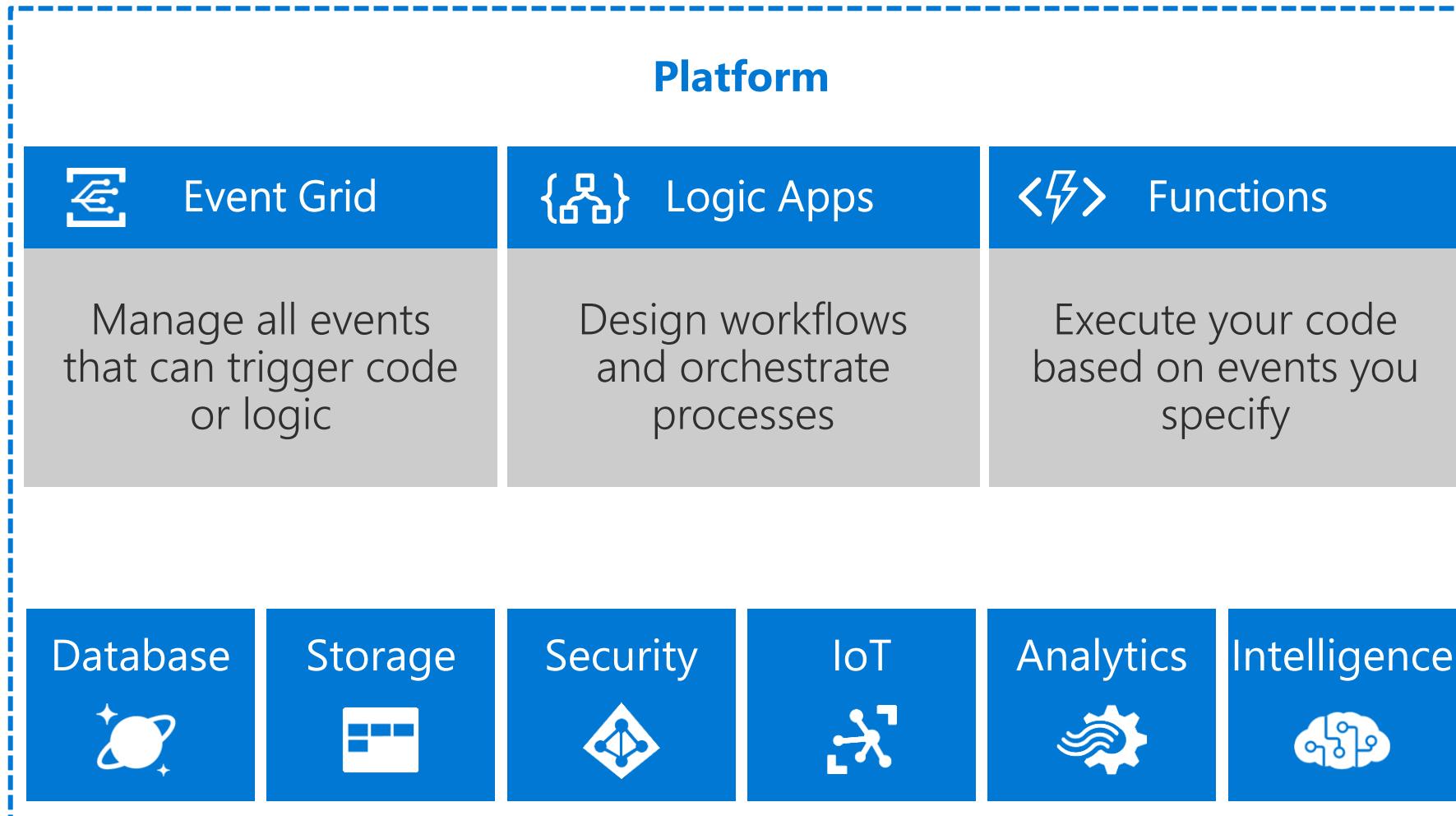
## <> Functions

Execute your code  
based on events you  
specify

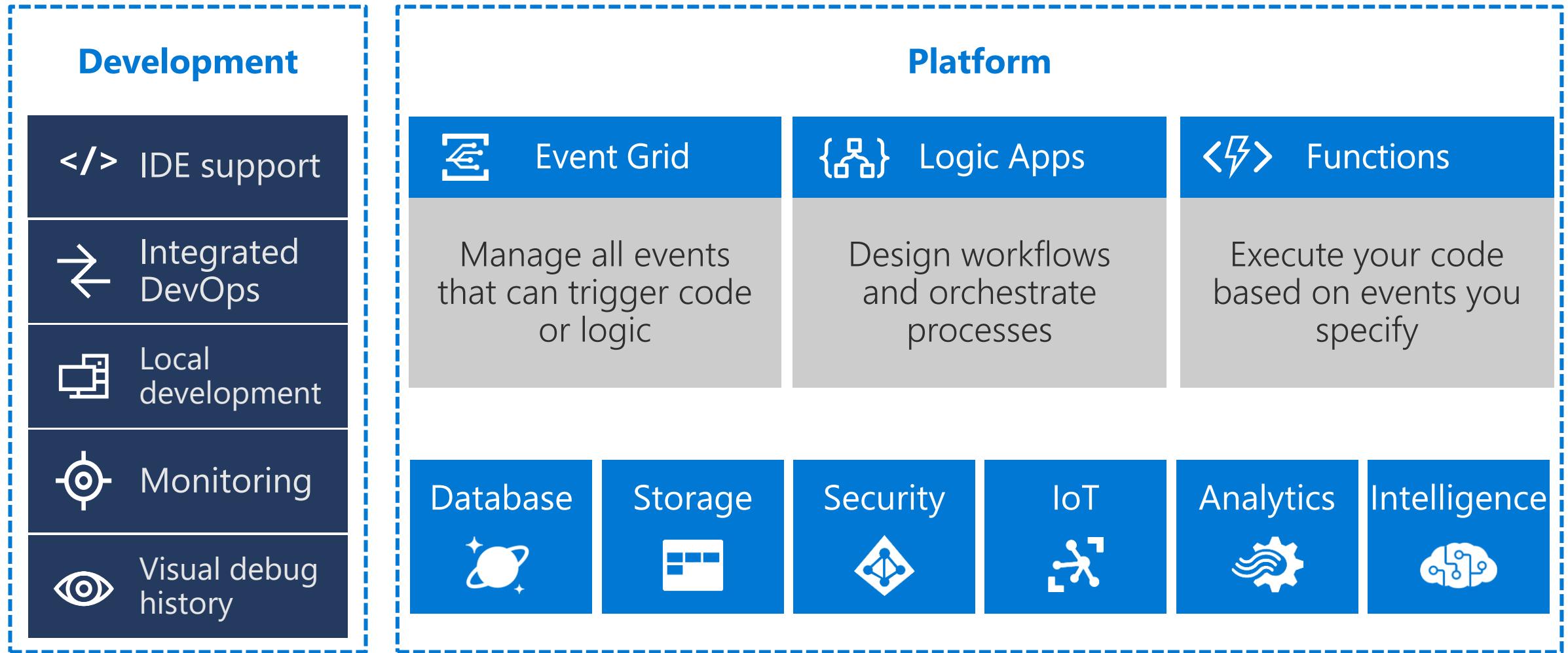
# Azure serverless platform components



# Azure serverless platform components



# Azure serverless platform components





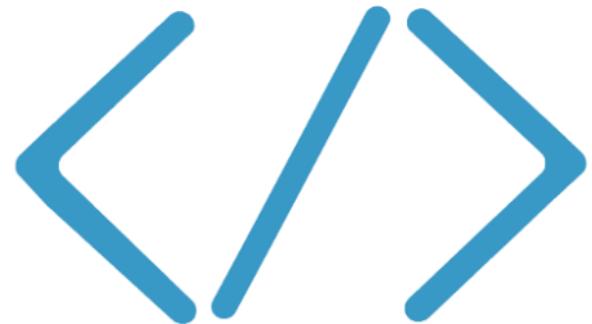
# Demo

Create a Logic App

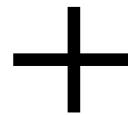
Azure  
Functions

# Introducing Functions

Code



Azure Functions



Events



# Why Azure Functions?

- 1     Supports C#, Node/JavaScript, Python, F#, PHP, PowerShell, Bash, Batch
- 2     Supports NuGet and NPM
- 3     Can even run EXE's or call into DLL's

# More reasons?!

## Triggers

Blob Storage

Cosmos DB

Event Hub

HTTP

Queues

Service Bus

Timer

Webhook

## Bindings

File

Table

Excel

OneDrive

Email

Mobile app

Notification

More...

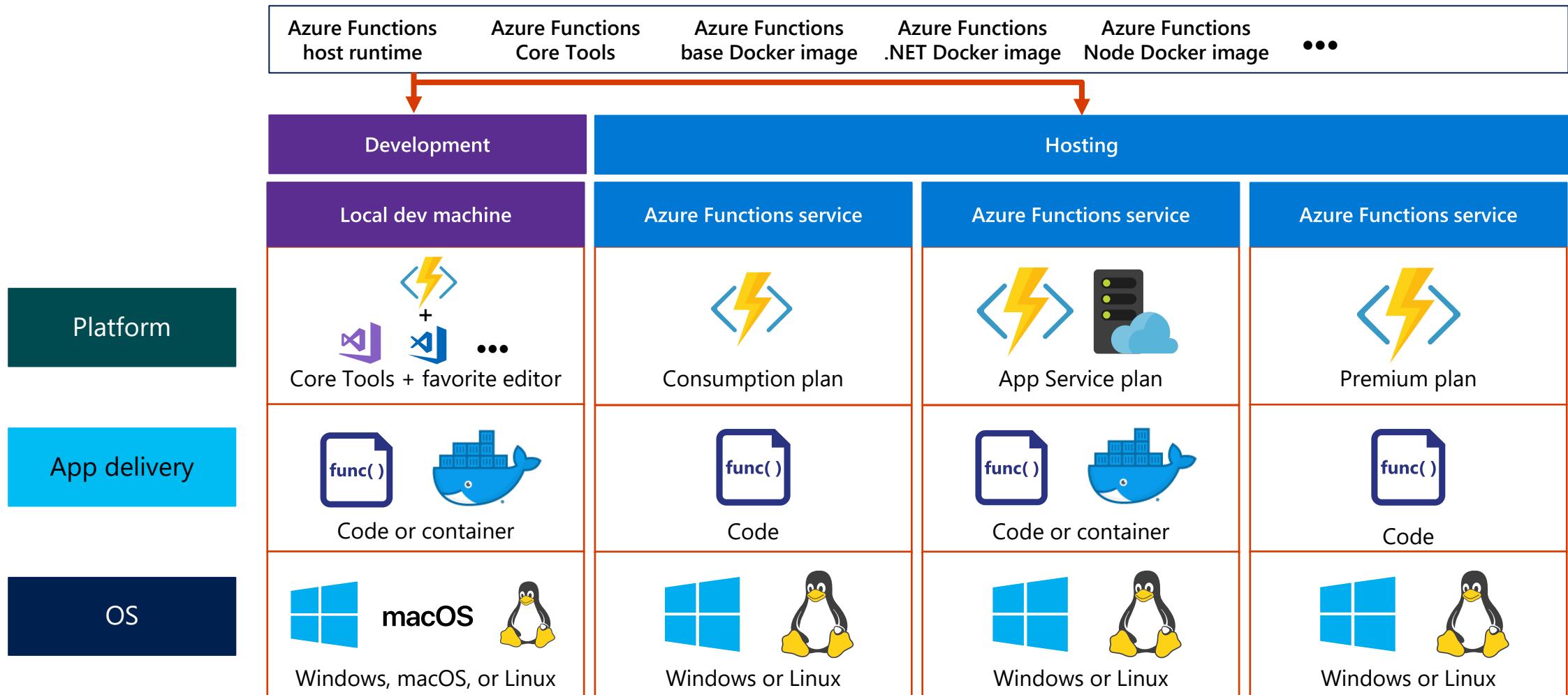
# Scale and hosting

- You can choose between two types of plans
  - Consumption
    - Instances are dynamically instanced, and you are charged based on compute time
  - App Service plan
    - Traditional App Services model used with Web Apps, API Apps, and Mobile Apps
- The type of plan controls:
  - How host instances are scaled out
  - The resources that are available to each host

# Azure Functions hosting



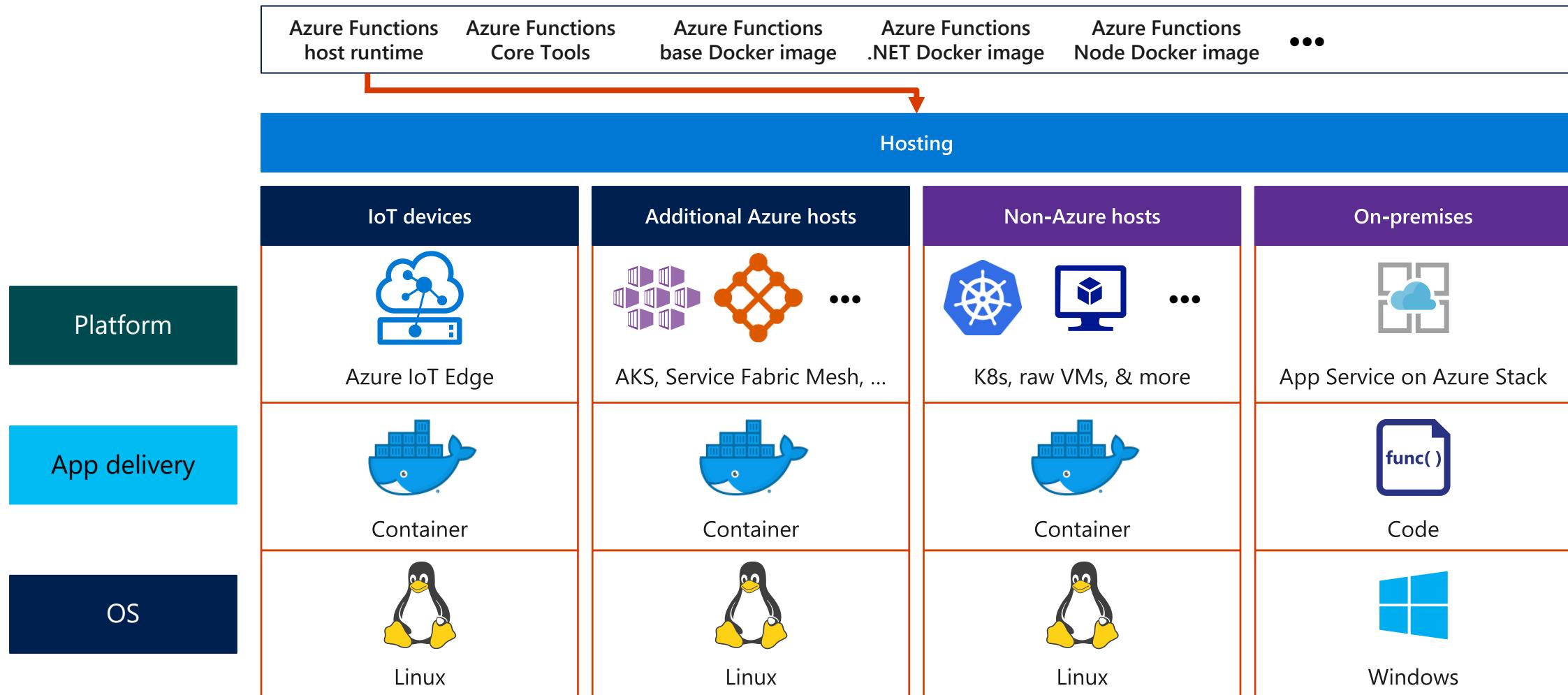
<https://github.com/azure/azure-functions-host> (+other repos)



# Azure Functions hosting (continued)



<https://github.com/azure/azure-functions-host> (+other repos)





# Demo

Create an Azure Function using Azure Portal

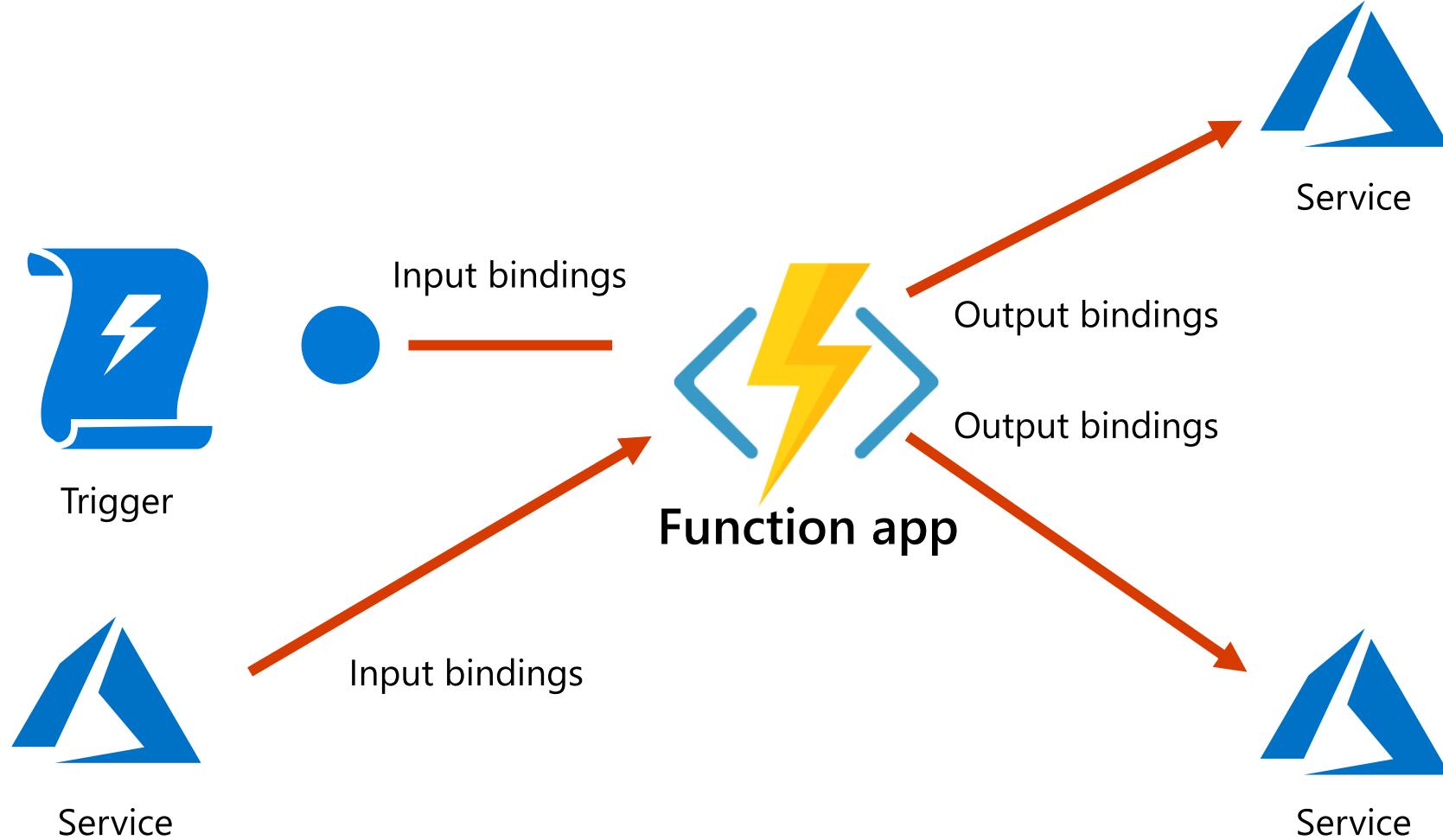
# Triggers



# Trigger types

- Triggers based on Azure services
  - Cosmos DB
  - Blob and queues
  - Service Bus
  - Event Hub
- Triggers based on common scenarios
  - HTTP request
  - Scheduled timer
- Triggers based on third-party services
  - GitHub
- And more...

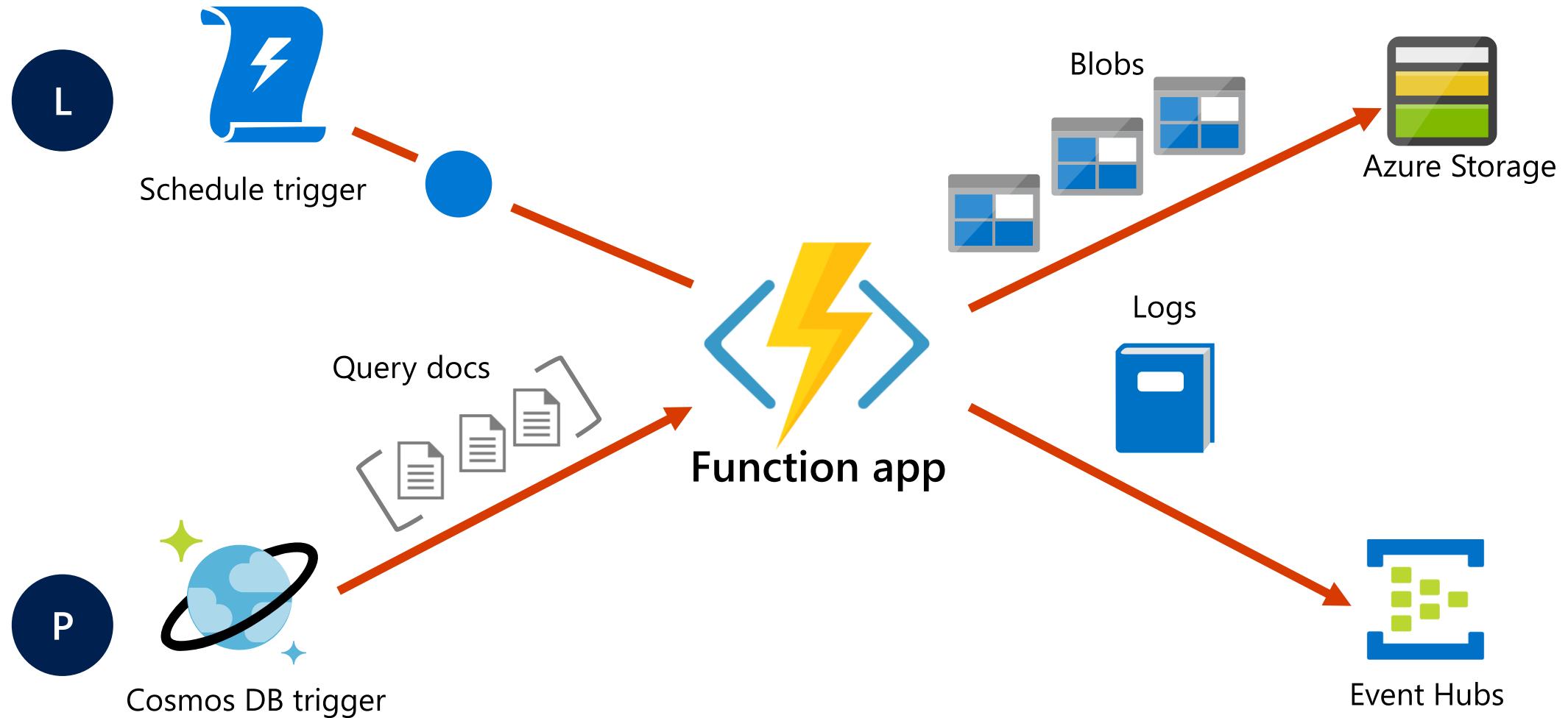
# Input and Output Bindings



# Bindings

- Declarative way to connect to data from your code
  - Connect to services without writing plumbing code
  - Service credentials are not stored in code
  - Bindings are optional
- Function can have multiple input and output bindings
- Output bindings can send data to Azure services such as
  - Storage
  - Azure Cosmos DB
  - Service Bus

# Trigger and Bindings example



# Azure Functions in Visual Studio

- Visual Studio project type
  - Develop, test and deploy C# functions to Azure
- Use WebJobs attributes to configure functions in C# code
- Pre-compile C# functions
  - Better cold-start performance

# Function code

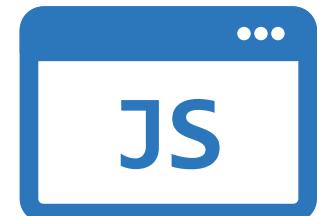
```
using System;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Host;

namespace FunctionApp1
{
    public static class Function1
    {
        [FunctionName("QueueTriggerCSharp")]
        public static void Run([QueueTrigger("myqueue-items", Connection =
"QueueStorage")]string myQueueItem, TraceWriter log)
        {
            log.Info($"C# Queue trigger function processed: {myQueueItem}");
        }
    }
}
```



# Bindings

```
{  
  "bindings": [  
    {  
      "name": "order",  
      "type": "queueTrigger",  
      "direction": "in",  
      "queueName": "myqueue-items",  
      "connection": "MY_STORAGE_ACCT_APP_SETTING"  
    },  
    {  
      "name": "$return",  
      "type": "table",  
      "direction": "out",  
      "tableName": "outTable",  
      "connection": "MY_TABLE_STORAGE_ACCT_APP_SETTING"  
    }  
  ]  
}
```

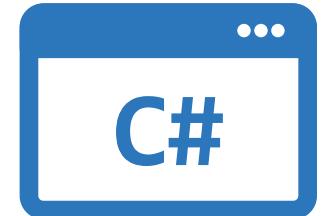


# Binding-based code

```
#r "Newtonsoft.Json"

using Microsoft.Extensions.Logging;
using Newtonsoft.Json.Linq;

public static Person Run(JObject order, ILogger log)
{
    return new Person() {
        PartitionKey = "Orders",
        RowKey = Guid.NewGuid().ToString(),
        Name = order["Name"].ToString(),
        MobileNumber = order["MobileNumber"].ToString()
    };
}
```





# Demo

Creating an Azure Functions project

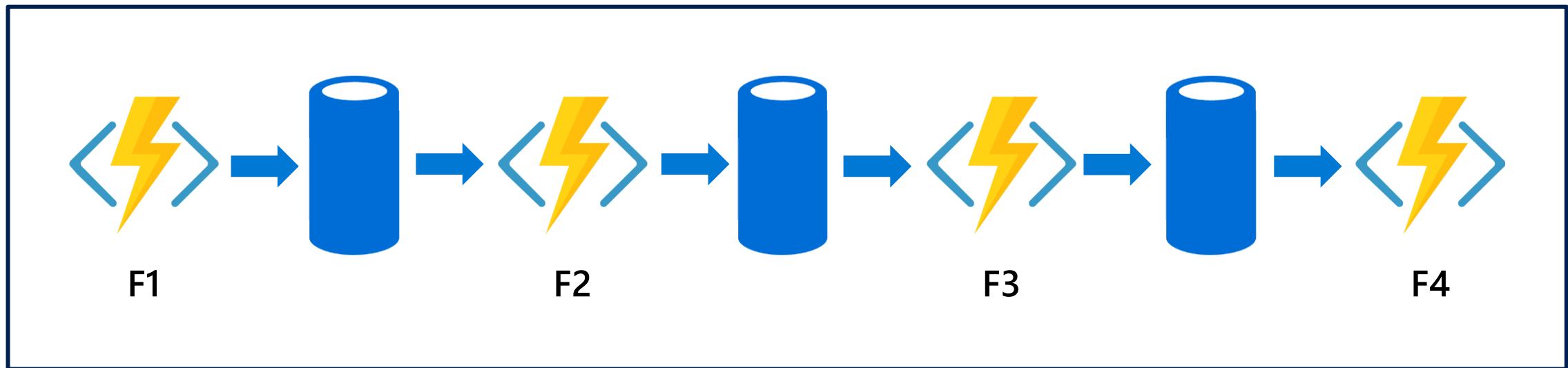
Durable  
Functions

# Durable Functions

- Write stateful functions in a stateless environment
- Manages state, checkpoints, and restarts
- Defines an Orchestrator function
  - Workflows are defined in code
  - Calls other functions synchronously or asynchronously
  - Checkpoint progress whenever function awaits

# Durable Function scenario - Chaining

Function chaining refers executing a sequence of functions in a particular order. Often, the output of one function needs to be applied to the input of another function.



# Durable Function scenario - Chaining code

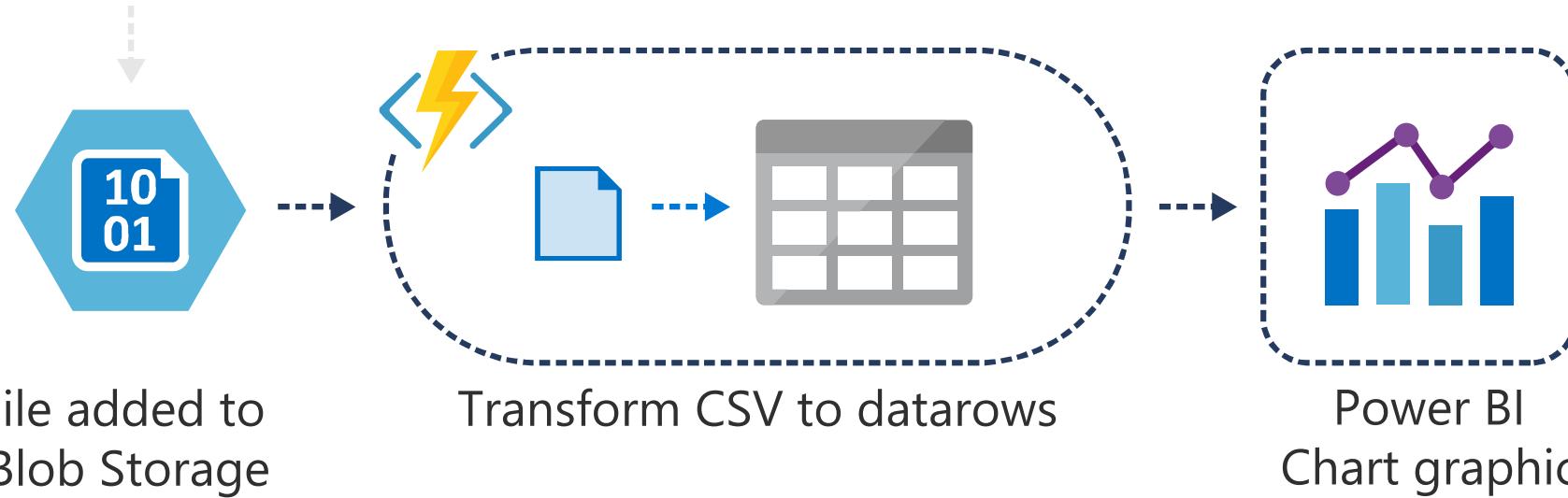
```
public static async Task<object> Run(DurableOrchestrationContext ctx)
{
    try
    {
        var x = await ctx.CallActivityAsync<object>("F1");
        var y = await ctx.CallActivityAsync<object>("F2", x);
        var z = await ctx.CallActivityAsync<object>("F3", y);
        return await ctx.CallActivityAsync<object>("F4", z);
    }
    catch (Exception)
    {
        // error handling/compensation goes here
    }
}
```



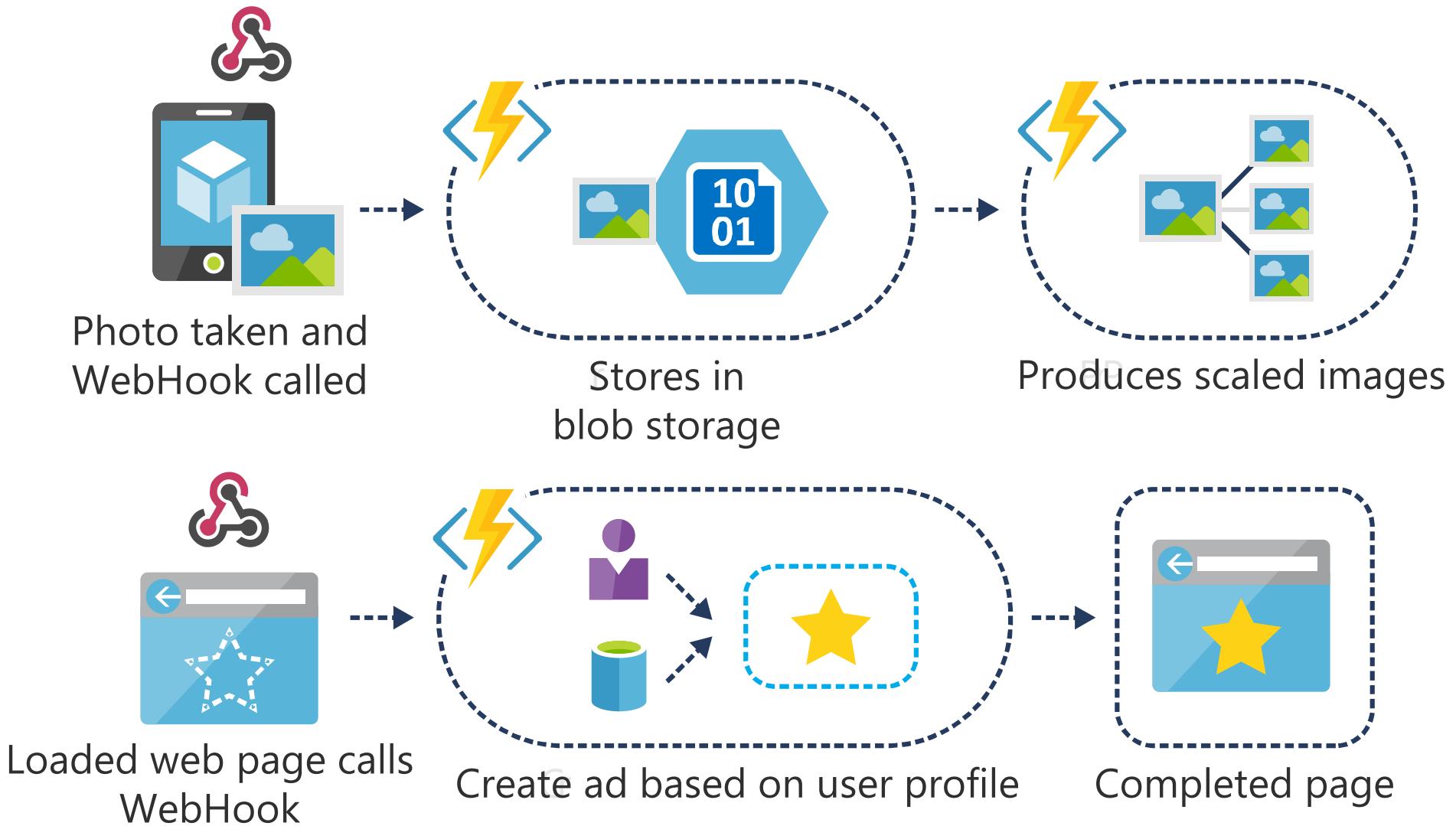
# Best practices

- Avoid long-running functions
  - Functions that run for a long time can time out
- Use queues for cross-function communication
  - If you require direct communication, consider Durable Functions or Azure Logic Apps
- Write stateless functions
  - Functions should be stateless and idempotent
  - State data should be associated with your input and output payloads
- Code defensively
  - Assume that your function might need to continue from a previous fail point

# Scenarios



# Scenarios



# Resources

## Tools & Resources

- Visual studio - <https://visualstudio.microsoft.com/>
- Visual Studio Code - <https://code.visualstudio.com/>
- Git - <https://git-scm.com/>
- Azure Functions Docs - <https://docs.microsoft.com/en-us/azure/azure-functions/>
- Azure QuickStart Templates: <https://github.com/Azure/azure-quickstart-templates>

Thank you!

My Name is Tiago Costa and this is how you  
**Use Azure Functions**

Contact me:

- @TiagoCostaPT
- [www.tiagocosta.com](http://www.tiagocosta.com)
- [tiago.costa@tiagocosta.com](mailto:tiago.costa@tiagocosta.com)



# Global Microsoft 365 Developer Bootcamp

<https://aka.ms/Microsoft365DevBootcampSurvey2020>