

Azure Functions: Swiss Knife for your backend system

M. IBNU FADHIL



About Me

Muhammad Ibnu Fadhil

Coder.Entrepreneur.Maker

T: @mifmasterz / @gravicode

Founder of

PT Gravicode Multinovative Plexindo

Initiator of Gadgeteer Indonesia

<https://www.facebook.com/netgadgeteerindonesia>

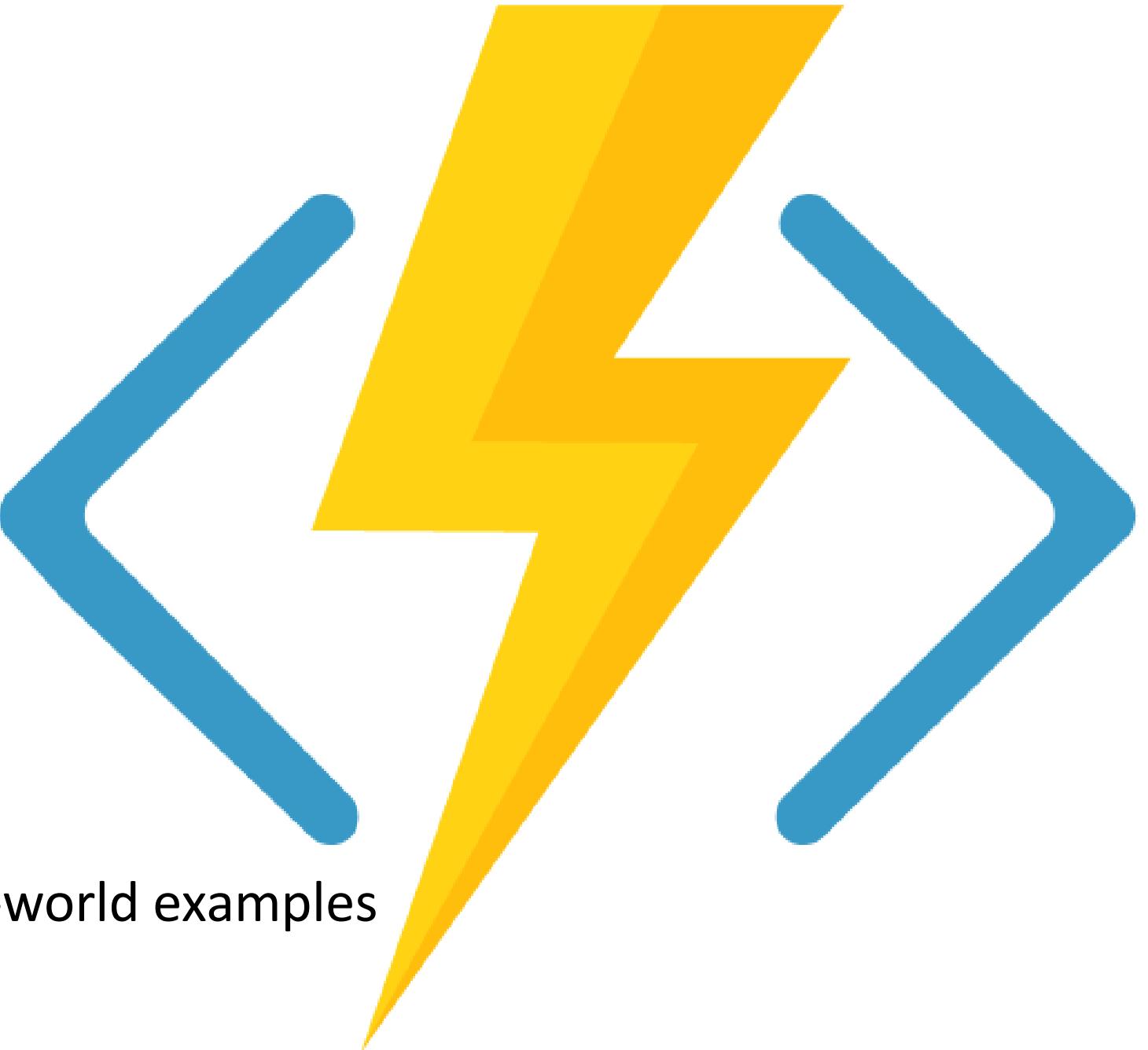
Contributor in Makers.ID

<http://makers.id>

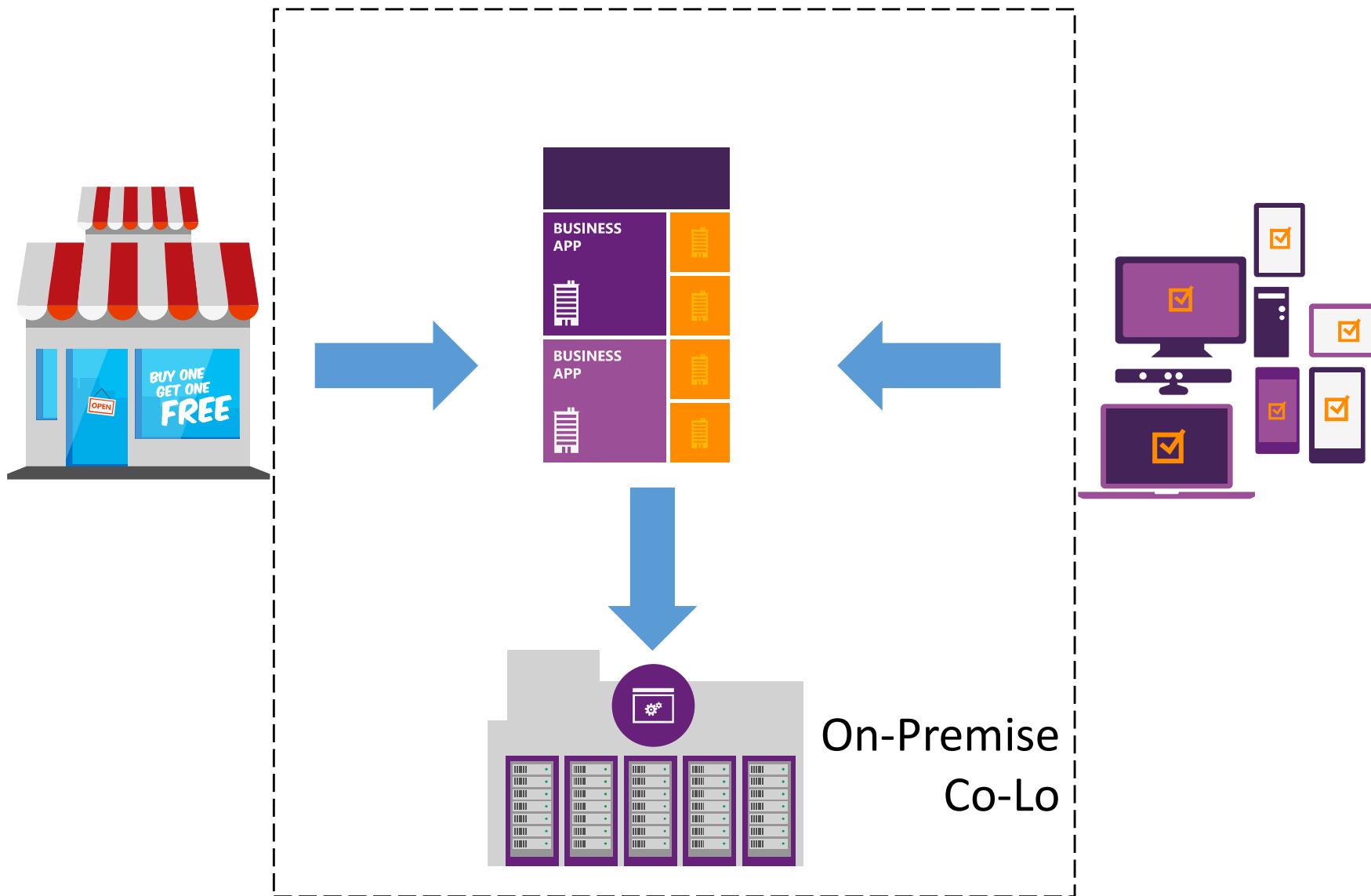


Agenda

- Why Serverless?
- What is serverless?
- Intro Azure Functions
- Serverless patterns / Real-world examples



Before cloud



Before cloud

What media should I use to keep backup?

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



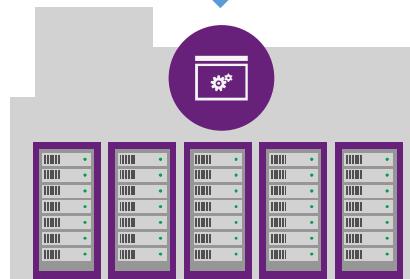
Do I need secondary network connection?

How many **servers** do I need?

It takes how long to **provision** a new **server**?

Who monitors my **Servers**?

What size of **servers** should I **buy**?



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



How I **scale** my app?

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

Who monitors my **App**?

Who has **physical** access to my **servers**?

Do I need a UPS?

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



Which OS should I use?

What happens if the power goes out?

How can I dynamically configure my app?

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

What storage I need to use?

Are my **server** in a secure location?

What happens in case of **server hardware** failure?

How can I increase **server** utilization?

Then came IaaS ...

What media should I use to keep backup?

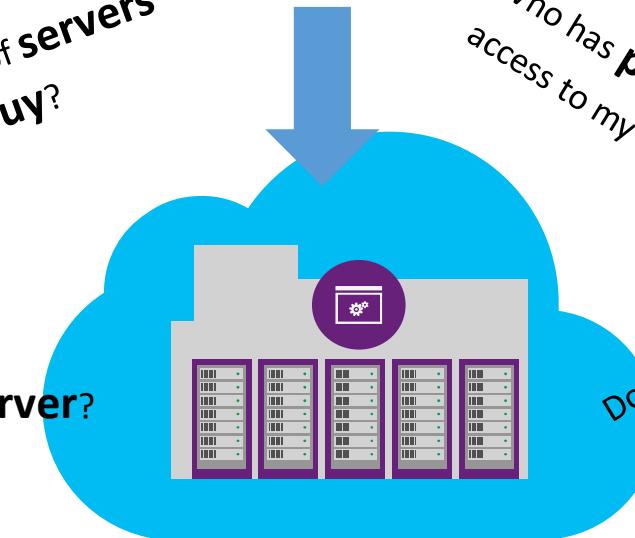
Which packages should be on my **server**?

Who monitors my **Servers**?

Do I need secondary network connection?

How many **servers** do I need?

It takes how long to **provision** a new **server**?



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

How I **scale** my app?

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

Who monitors my **App**?

Who has **physical** access to my **servers**?

Do I need a UPS?

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



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

What storage I need to use?

Are my **server** in a secure location?

What happens in case of **server hardware** failure?

How can I increase **server** utilization?

Which OS should I use?

What happens if the power goes out?

How can I dynamically configure my app?

Is it PaaS time?



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



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

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

How can I increase **server** utilization?

Who **monitors**
my App?

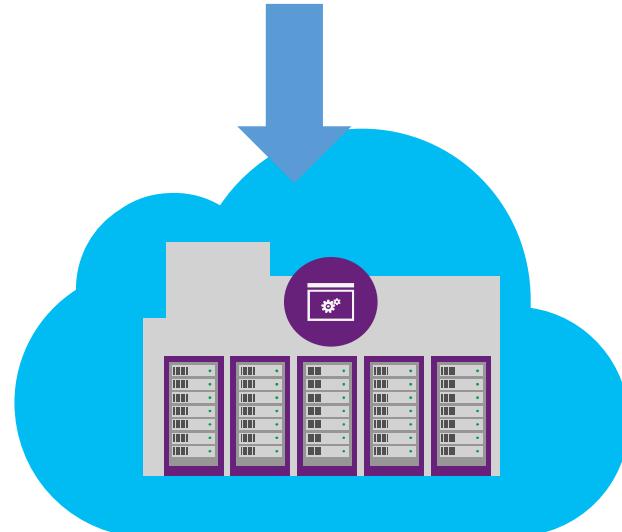


How often should I **backup**
my **server**?

Which OS should I use?

How can I dynamically
configure my app?

How many **servers**
do I need?



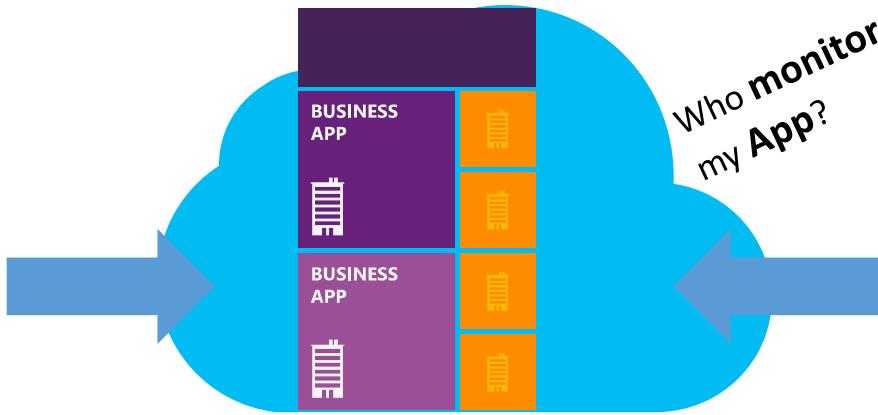
How often should
I **patch** my **servers**?

Is it PaaS time?



How many servers
do I need?

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



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

How I *scale* my app?

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

How can I increase **server** utilization?

Who monitors
my App?



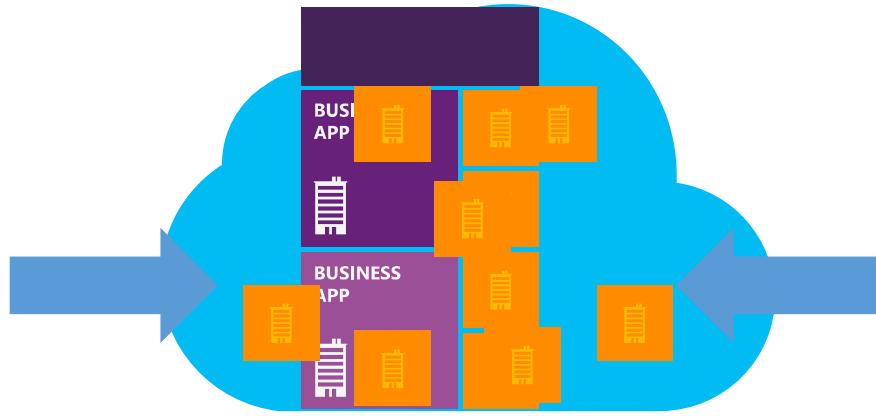
Which OS should I use?

How can I dynamically
configure my app?

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

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

Serverless . . .



*How many servers
do I need?*

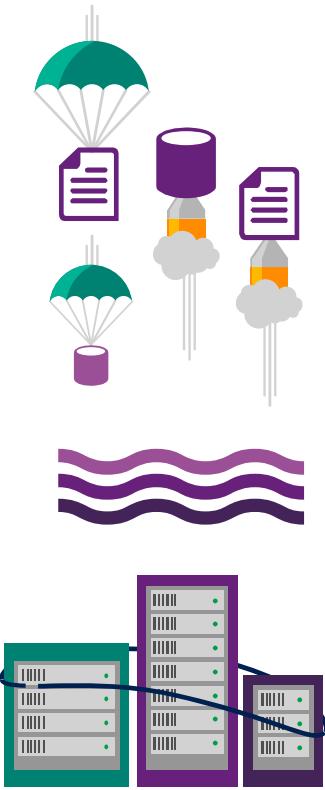


*What is the right **size** of
servers for my business needs?*

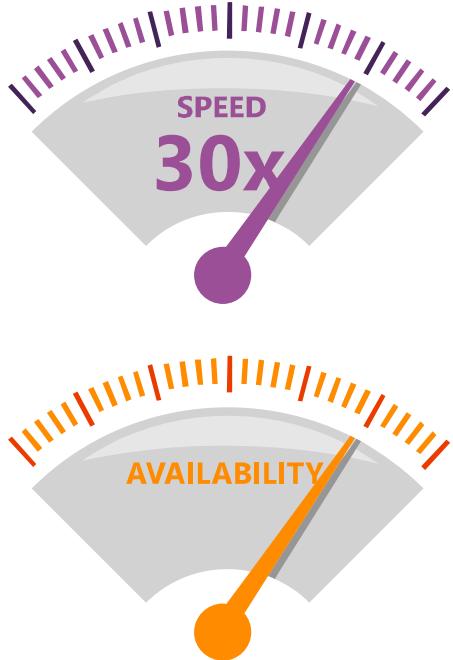
How I scale my app?

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

What is Serverless?



Abstraction
of servers

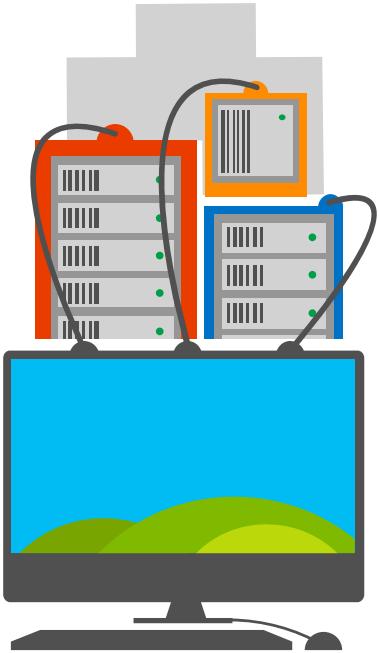


Event-driven/
instant scale

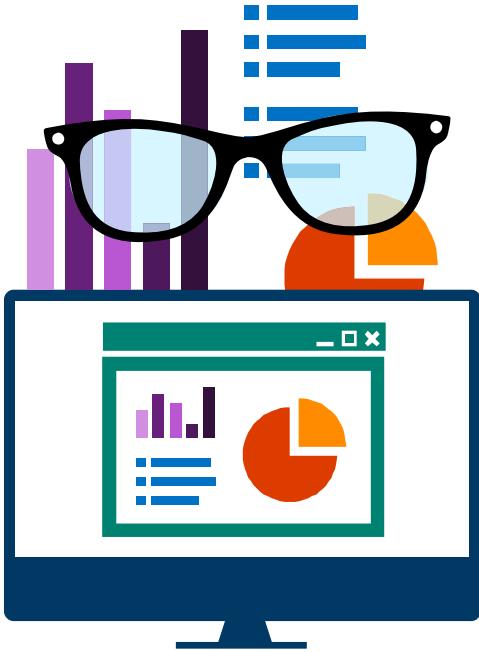


Sub-second
billing

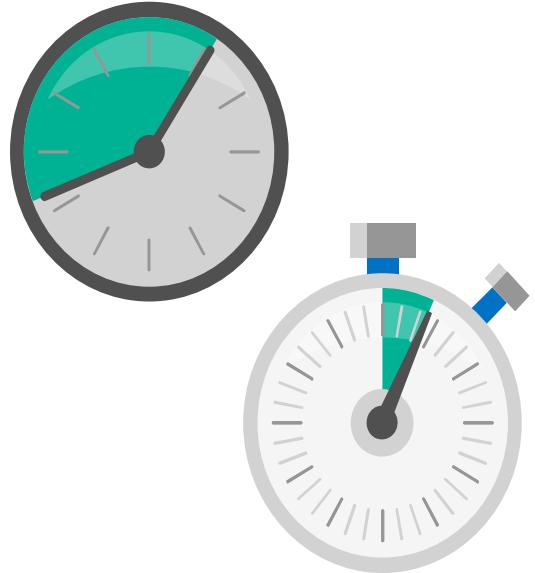
Benefits of Serverless?



Reduced
DevOps

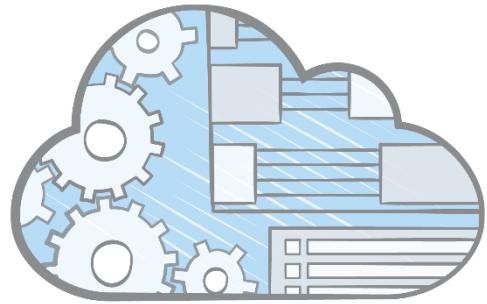


Focus on
Business
Logic

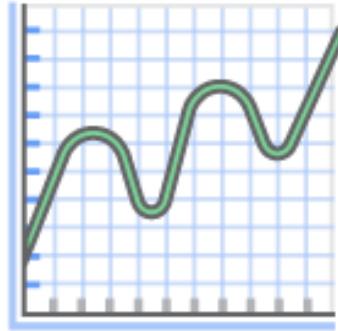


Reduced Time
To Market

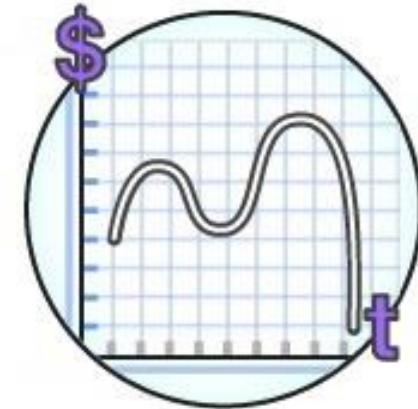
FaaS benefits



No servers to manage



Continuous scalability



Cost effectiveness



Azure Functions

Process events with Serverless code.

Make composing Cloud Apps insanely easy

Develop Functions in C#, Node.js, F#, Python, PHP, Batch and more

Easily schedule event-driven tasks across services

Expose Functions as HTTP API endpoints

Scale Functions based on customer demand

Easily integrate with Logic Apps

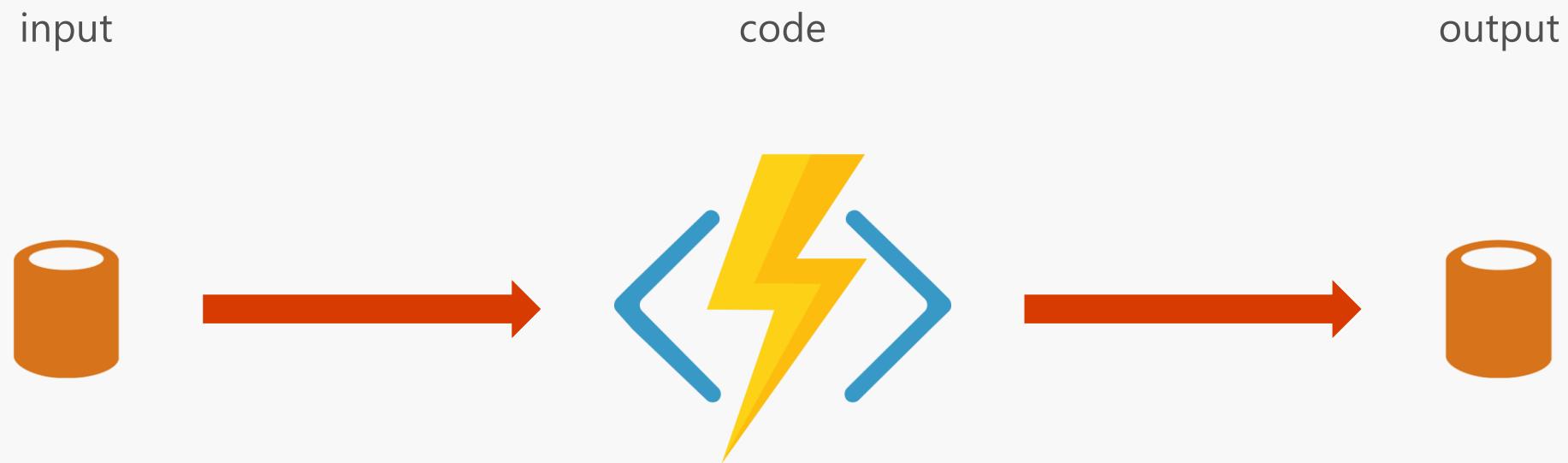
Use cases

- Mobile backend
- Serverless websites
- Data processing
- Event processing
- Scheduled tasks

Functions Hello World

Demo

Azure Functions



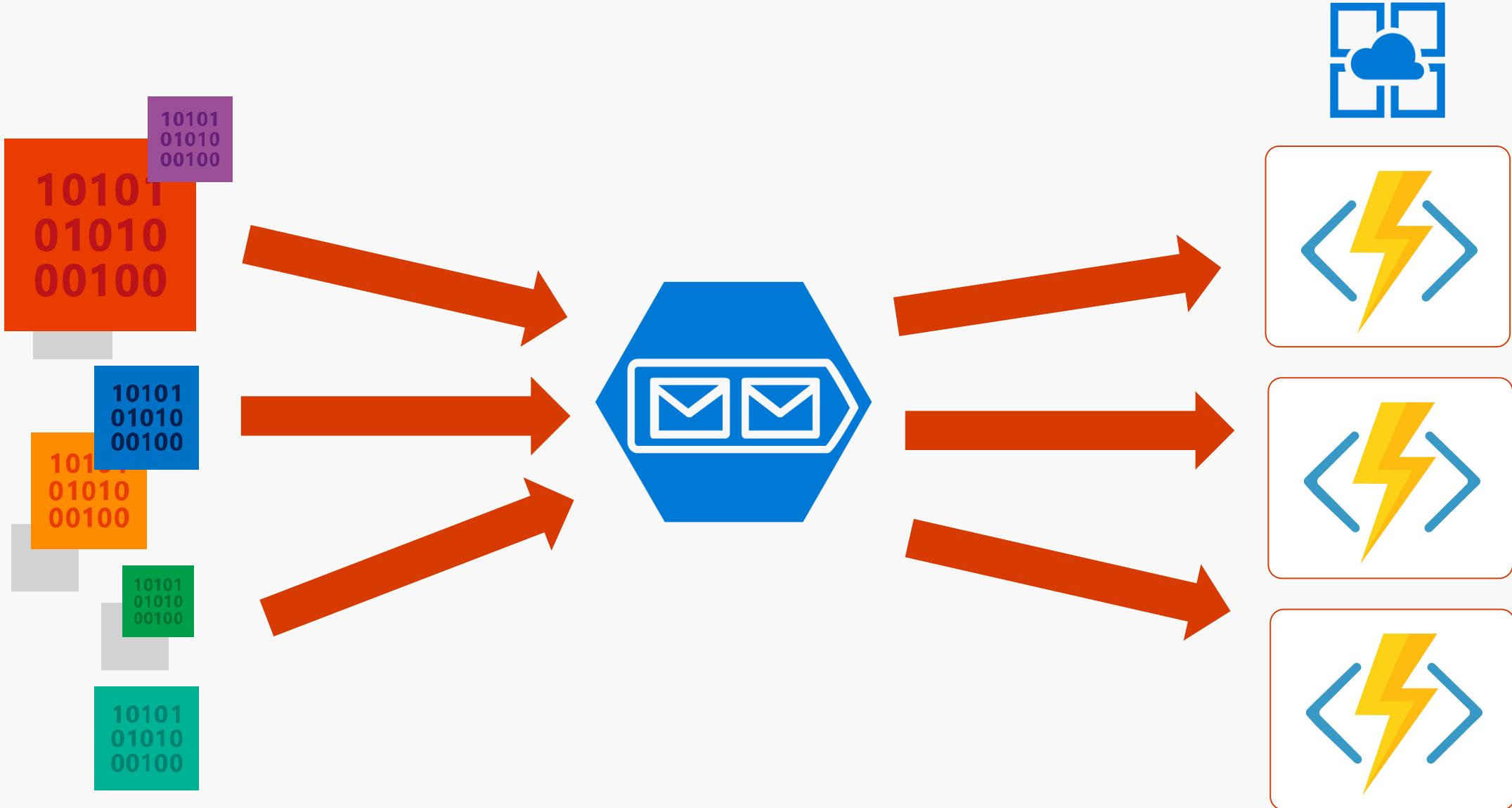
Supported bindings

| Type | Service | Trigger | Input | Output |
|------------------------|-------------------------------|---------|-------|--------|
| Schedule | Azure Functions | ✓ | | |
| HTTP (REST or WebHook) | Azure Functions | ✓ | | ✓ |
| Blob Storage | Azure Storage | ✓ | ✓ | ✓ |
| Queues | Azure Storage | ✓ | | ✓ |
| Tables | Azure Storage | | ✓ | ✓ |
| Tables | Azure Mobile Apps Easy Tables | | ✓ | ✓ |
| No-SQL DB | Azure DocumentDB | | ✓ | ✓ |
| Streams | Azure Event Hubs | ✓ | | ✓ |
| Push Notifications | Azure Notification Hubs | | | ✓ |

Supported Languages

- 1st class support
 - Node/JavaScript
 - C#
- Experimental support
 - F#
 - Python
 - PHP
 - Batch
 - Bash
 - PowerShell

Built to scale



Servers when you want them...

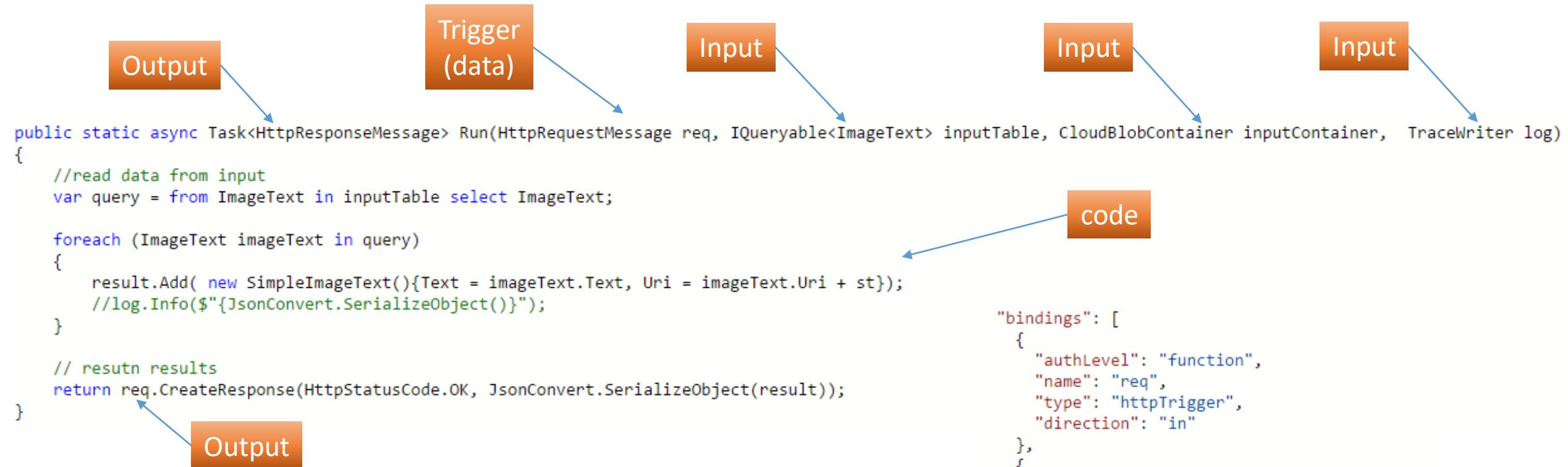
serverless when you don't

What is the “Functions” programming model?

- Function as the unit of work
- Functions are executed; they start and finish
- Functions have inputs and outputs

```
public async static Task ProcessQueueMessageAsync(CancellationToken(
    [QueueTrigger("blobcopyqueue")] string blobName,
    [Blob("textblobs/{queueTrigger}", FileAccess.Read)] Stream blobInput,
    [Blob("textblobs/{queueTrigger}-new", FileAccess.Write)] Stream blobOutput,
    CancellationToken token)
{
    await blobInput.CopyToAsync(blobOutput, 4096, token);
}
```

Functions Programming Model



- Function as a single unit of work
- Functions are executed per trigger
- Functions have inputs and outputs

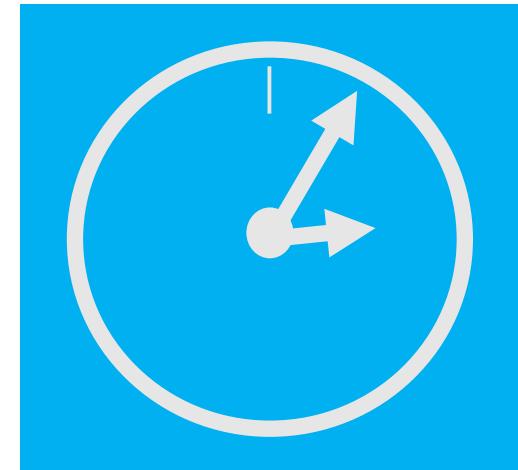
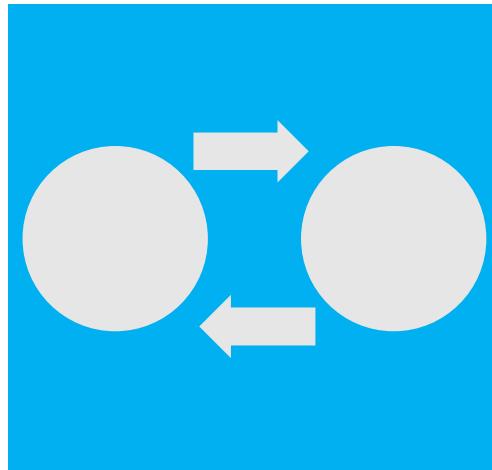
Triggers and bindings

| Type | Service | Trigger | Input | Output |
|------------------------|-------------------------|---------|-------|--------|
| Schedule | Azure Functions | ✓ | | |
| HTTP (REST or webhook) | Azure Functions | ✓ | | ✓* |
| Blob Storage | Azure Storage | ✓ | ✓ | ✓ |
| Events | Azure Event Hubs | ✓ | | ✓ |
| Queues | Azure Storage | ✓ | | ✓ |
| Tables | Azure Storage | ✓ | ✓ | |
| Tables | Azure Mobile Apps | ✓ | ✓ | |
| No-SQL DB | Azure DocumentDB | ✓ | ✓ | |
| Push Notifications | Azure Notification Hubs | | | ✓ |

(* - The http out binding requires an http trigger)

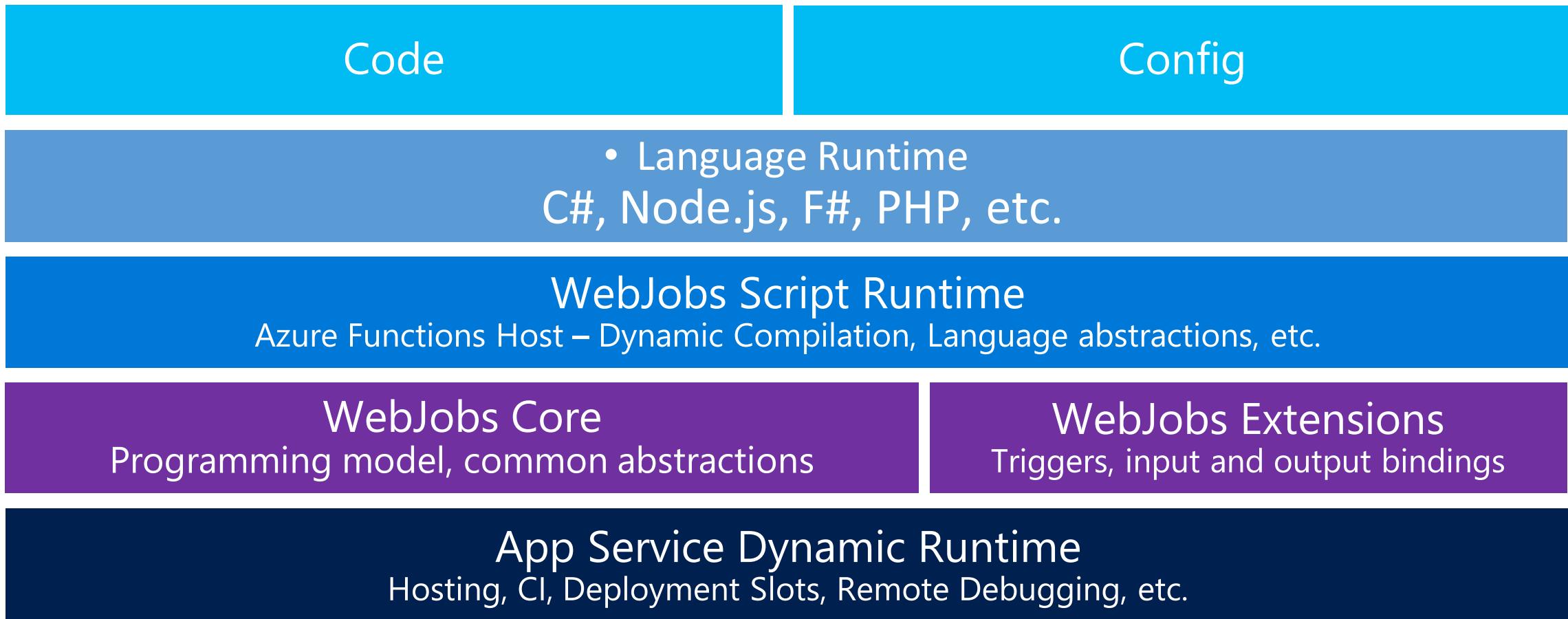
Functions Programming Model - Best Practices

- Functions should “do one thing”
- Functions should be stateless
- Functions should be idempotent
- Functions should finish as quickly as possible



Azure Functions architecture

Built on top of App Service and WebJobs SDK





Serverless Patterns

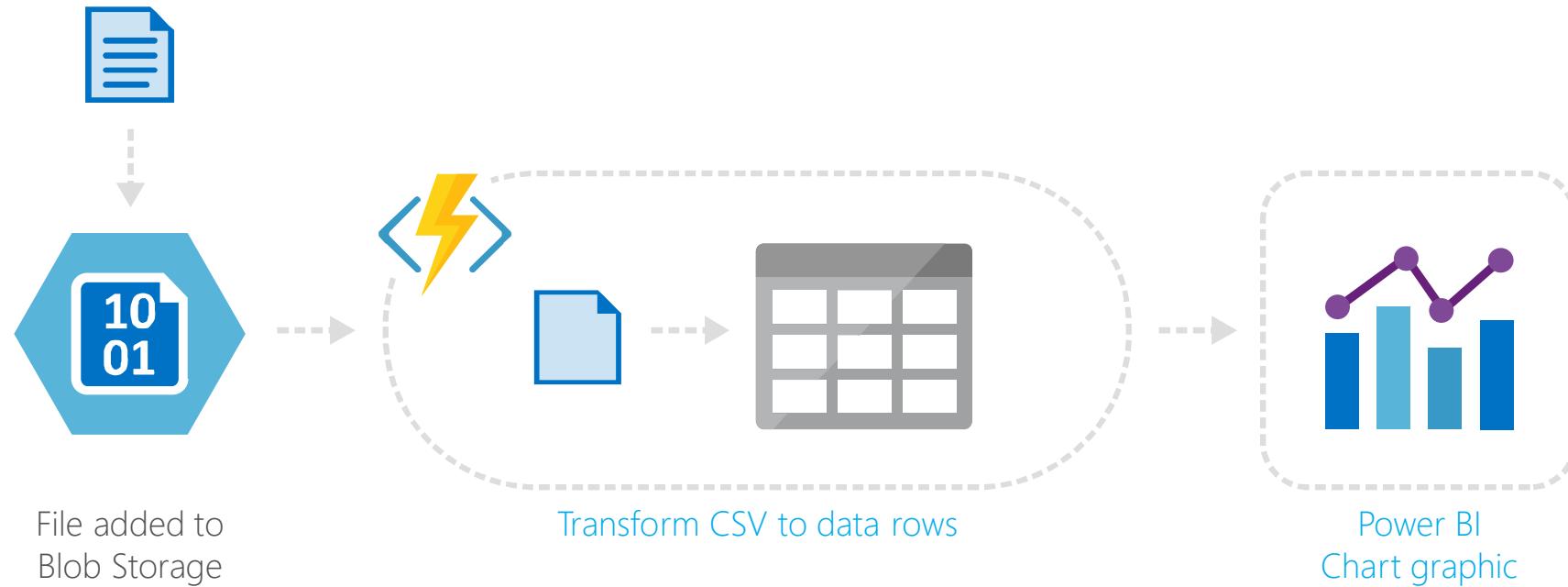
Applications

Example: Timer based processing



Applications

Example: Azure service event processing



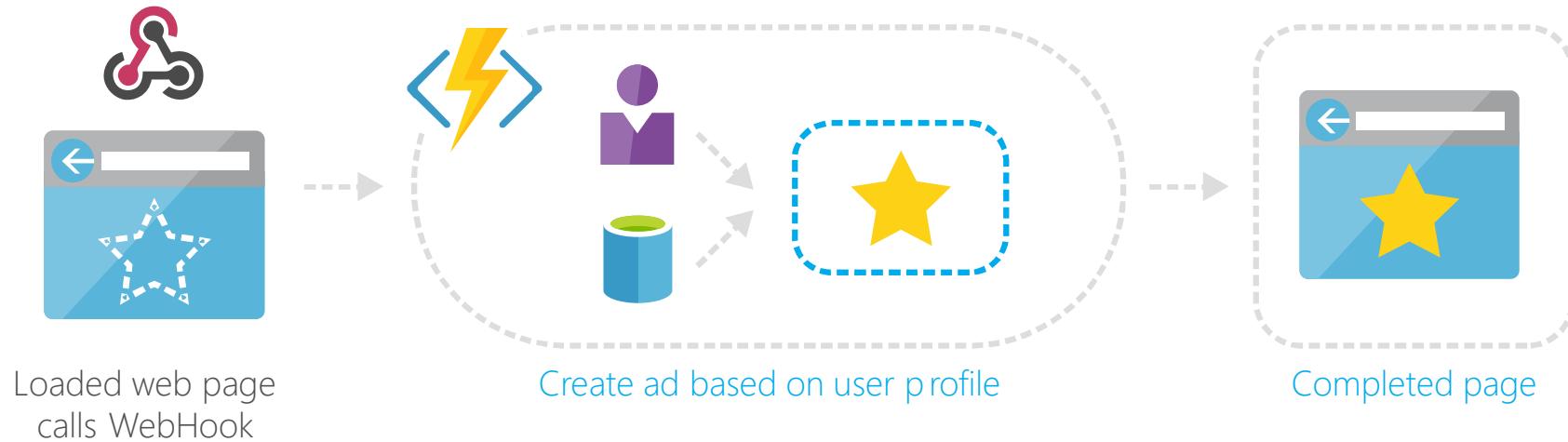
Async background processing

Example: Serverless Mobile back ends



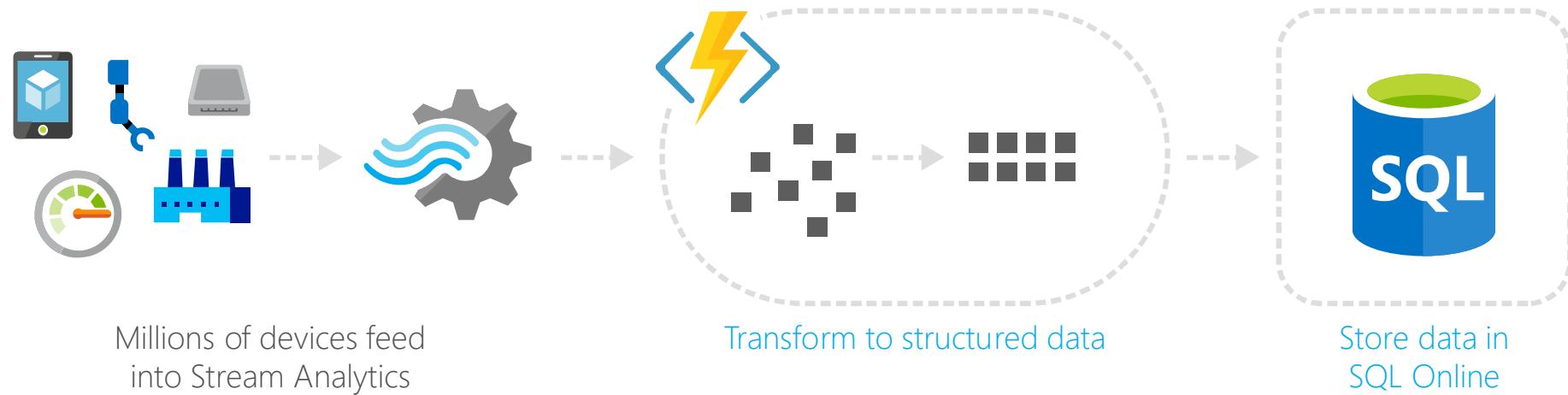
Applications

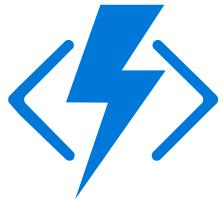
Example: Serverless Web Applications architectures



Applications

Example: Real-time stream processing





Pricing

1

$$\frac{\text{Allocated RAM}}{1024} * \text{Total Execution Time} * \text{GB}\backslash\text{sec Cost}$$

2

Total Request count * Request Cost

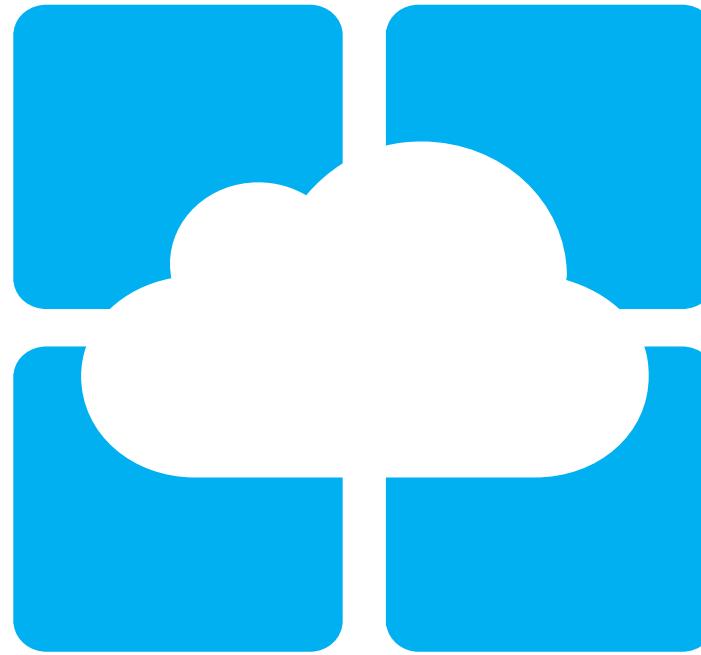
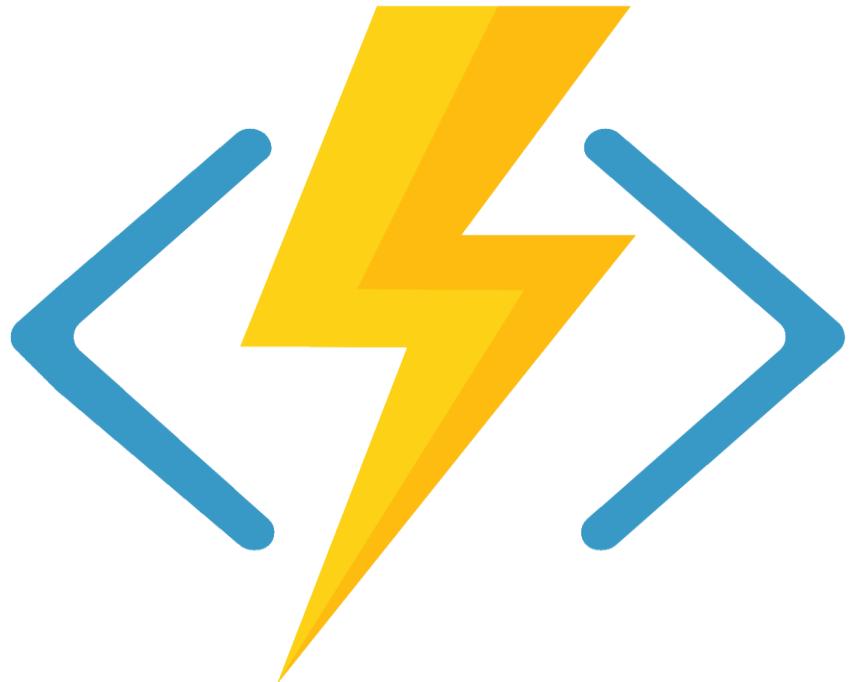
3

Azure Storage cost

Microsoft and Serverless

Try Functions – <https://functions.azure.com>

Try App Service – <https://tryappservice.azure.com>



Questions?

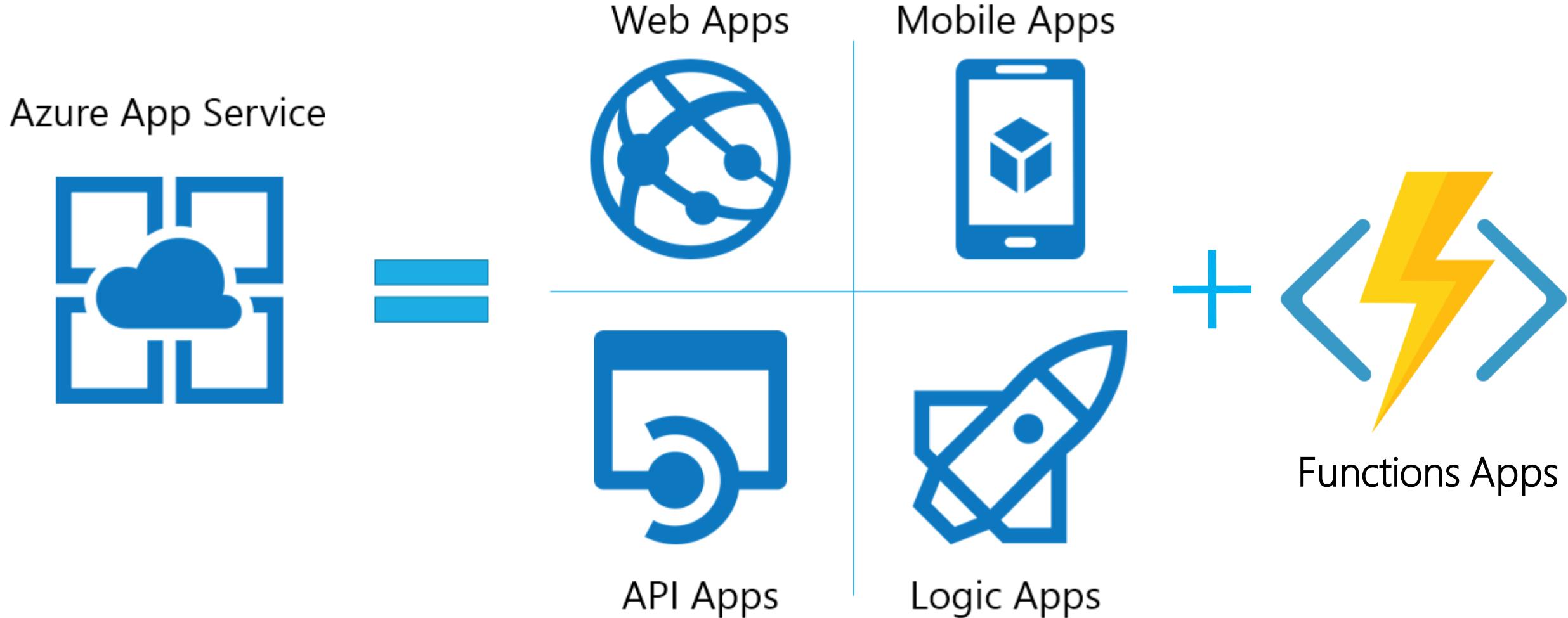


Thank You!

“Empower your business with intelligence app”

fadhil@Gravicode.com

App Service Eco-System



WebJobs SDK vs Azure Functions

- **Generalizing**

WebJobs SDK == more freedom -> more work

Azure Functions == less responsibilities -> less work

- **Commonalities**

Programming model differences

Hosting model differences

Commonalities

- Both use the “Function” oriented programming model
- Both support “bindings” for trigger/input/output
- Both support WebJobs SDK extensions model
- Both support external libraries being used
- Both can run locally and be debugged
- Both have runtime telemetry via the WebJobs Dashboard

Programming Model Differences

WebJobs SDK

- ✓ C# only
- ✓ Attributes for configuring bindings
- ✓ Traditional .NET developer experience (Visual Studio, NuGet, MSBuild)
- ✓ Many functions per class
- ✓ Can access and manipulate many core SDK features
- ✓ Can't listen for HTTP requests*

Azure Functions

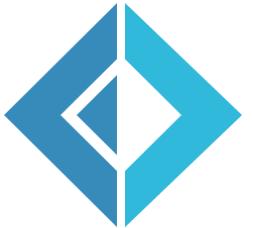
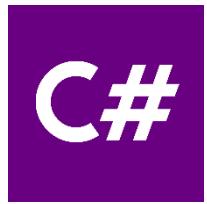
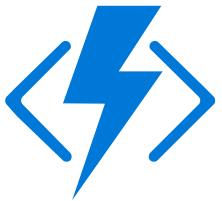
- ✓ C# & Node.js + more
- ✓ Config files for configuring bindings
- ✓ More diverse development experience (Web portal, VSCode, dynamically builds itself)
- ✓ Limited access to manipulate core SDK features, but (C# only) still some access
- ✓ Supports HTTP!

Hosting Model Differences

- WebJobs SDK
 - You configure host
 - Build a console app which is run
- WebJobs and Dedicated App Service plans
 - Runs the service in the background of Web/Mobile/API app
 - Runs any console app (not just SDK based ones)
 - You manage scale
- Azure Functions
 - Limited control over the host
 - Just give it your code/config
- Function App + Dynamic App Service Plans
 - Function app owns the whole host, including web frontend.
 - Only runs Azure Functions stuff – no other things
 - Scale is managed for you

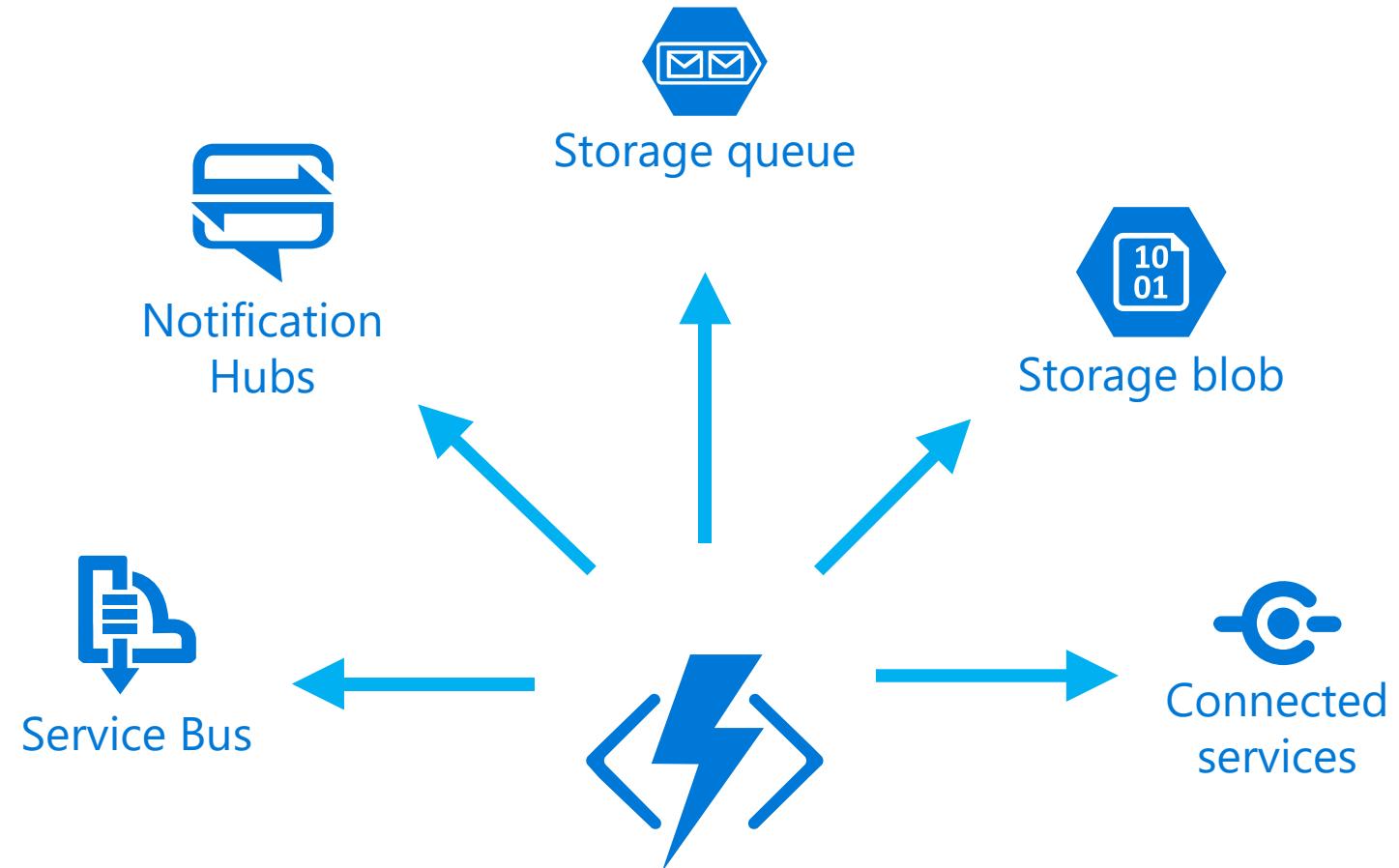
Azure Functions Vs Aws Lambda

Supported languages



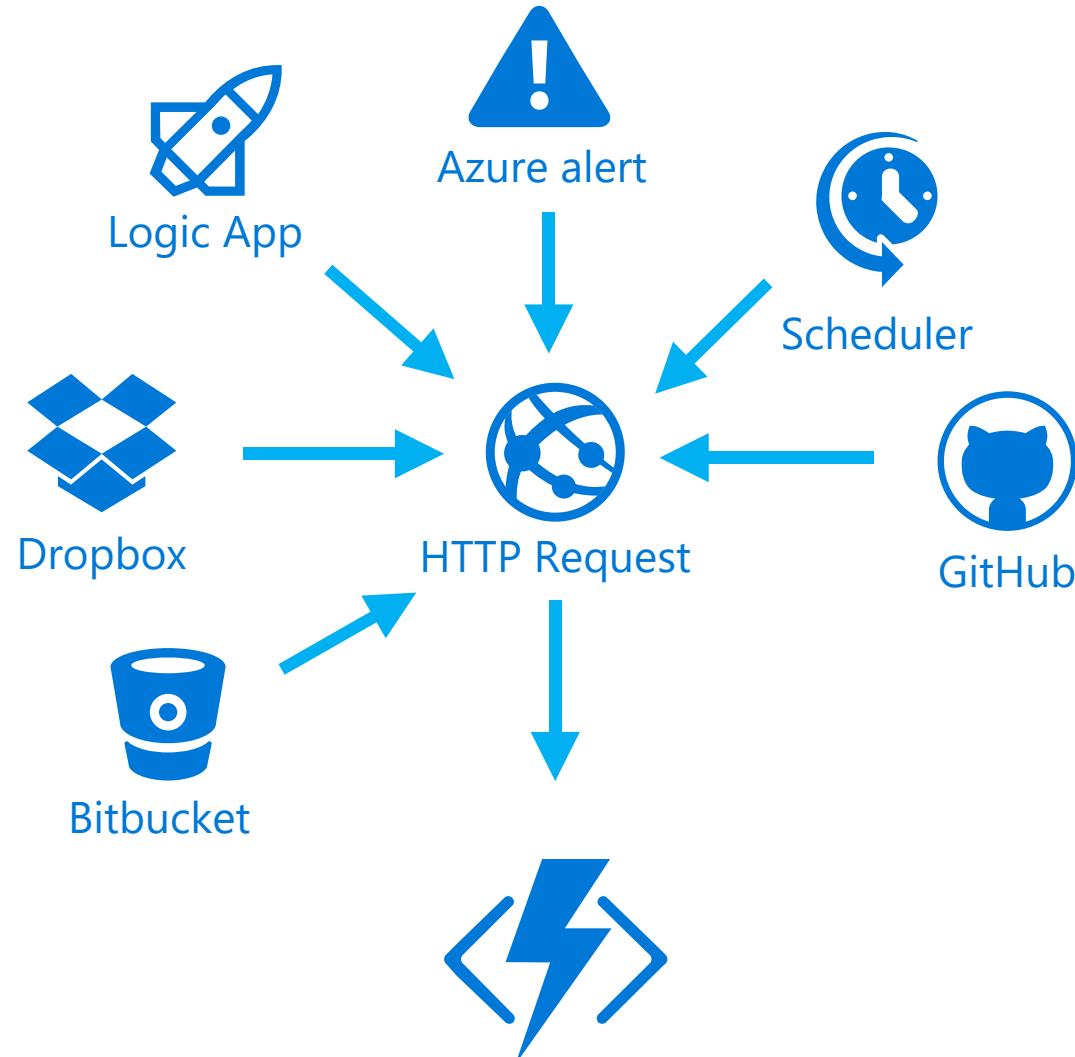
Event models: Azure way

Pull model



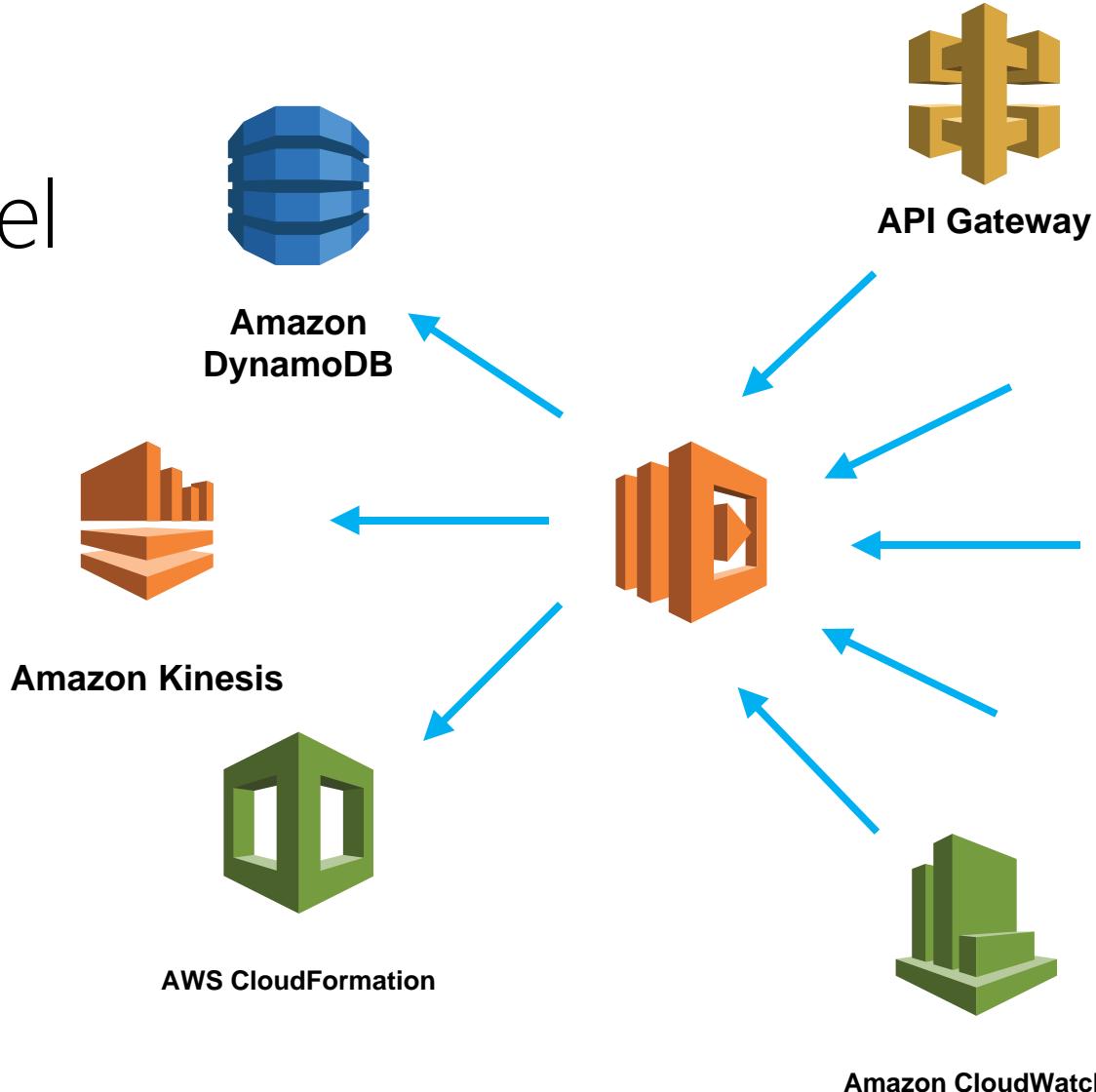
Event models: Azure way

Push model



Event models: Lambda way

Pull model



Push model