

Serverless apps with Azure Functions v2 and .NET Core



Lorenzo Barbieri

Cloud Solutions Architect - Microsoft

@_geniodelmale | lorenzo.barbieri@microsoft.com

 aspitalia.com

#netconfit



The “evolution” of application platforms

• What media should I use to keep backup?

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

• How can I **scale** my app?

• Do I need secondary network connection?

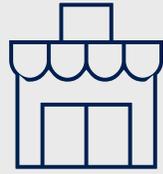
• How many **servers** do I need?

• Who **monitors** my **Servers**?

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

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

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



• Who **monitors** my **App**?

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

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

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

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

• Do I need a UPS?



• Which OS should I use?

• What happens if the power goes out?

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

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

• What storage I need to use?

• How can I dynamically configure my app?

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?



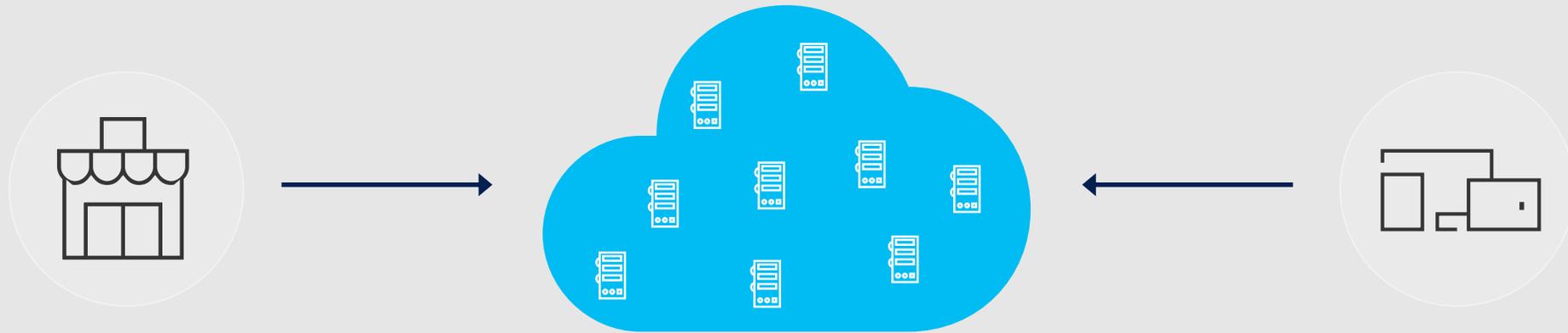
On-Premises

IaaS

PaaS

The “evolution” of application platforms

How do I **architect** my app?

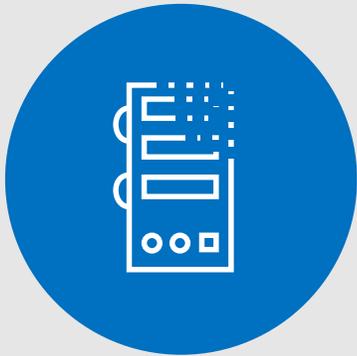


Serverless, the platform for next gen apps

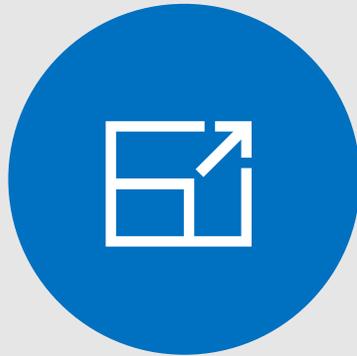


The "evolution" of application platforms

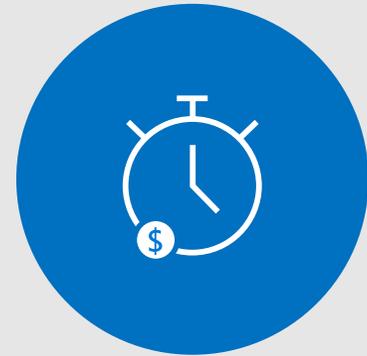
What is Serverless?



Abstraction
of servers



Event-driven/
instant scale



Micro-billing

Azure Functions

Azure Functions

Events



React to timers, HTTP, or events from your favorite Azure services, with more on the way

Code



Author functions in C#, F#, Node.JS, Java, and more

Outputs



Send results to an ever-growing collection of services

Full **integration** with Azure ecosystem

Functions is the center piece of the Serverless platform

Development

 IDE support

 Integrated DevOps

 Local development

 Monitoring

 Visual debug history

Platform

 Event Grid

Manage all events that can trigger code or logic

 Functions

Execute your code based on events you specify

 Logic Apps

Design workflows and orchestrate processes

Database



Storage



Analytics



Intelligence



Security



IoT



Functions 1.0 challenges

- Need for additional language support, e.g. Java, Python, Powershell
- Only able to host on Windows
- Assembly probing and binding issues for .NET developers
- Performance issues on a range of scenarios / languages
- Lack of UX guidance to production success

General Availability of
Functions 2.0
@
MS Ignite!



Functions 2.0 – What in this GA announcement?

- ✓ New Functions Quickstarts by Language
- ✓ Updated runtime built on .NET Core 2.1
- ✓ .NET Functions loading changes
- ✓ New extensibility model
 - ✓ Decoupled from language providers and bindings
- ✓ Run code from a package
- ✓ Tooling updates: CLI, Visual Studio & VS Code
- ✓ Durable Functions (GA)
- ✓ Consumption mode SLA

Functions runtime 1.0 and 2.0 key differences

	Functions 1.0	Functions 2.0
.NET Support	.NET Framework 4.7.1	.NET Core 2.1
Assembly isolation	No	Yes
Bindings versions	Runtime versions	User controlled
Language options	Limitations in languages and versions	Languages are external to the host
Node.js version	Node.js 6 only	Node.js 8 & 10 + future versions
Node.js native modules	Not supported	Supported
HTTP triggers	HTTP and specialized Webhooks	HTTP (supports Webhooks)
Language Runtime	Multiple languages per function app	Single language per function app
Functions Proxies	GA	GA
OpenAPI definition	Preview	Not yet available
Observability	Application Insights/WebJobs dashboard	App Insights

Demo

A tour of the Function Apps Portal



Bindings and integrations

Functions 1.0

Microsoft.NET.Sdk.Functions (.NET Framework 4.6)

- HTTP
- Timer
- Storage
- Service Bus
- EventHubs
- Cosmos DB

Functions 2.0

Microsoft.NET.Sdk.Functions (.NET Standard 2.0)

- HTTP
- Timer

Microsoft.Azure.WebJobs.Extensions.Storage 3.0.0

Microsoft.Azure.WebJobs.Extensions.ServiceBus 3.0.0

Microsoft.Azure.Webjobs.Extensions.EventHubs 3.0.0

Microsoft.Azure.WebJobs.Extensions.CosmosDB 3.0.0

Microsoft.Azure.Webjobs.Extensions.EventGrid 2.0.0

Microsoft.Azure.Webjobs.Extensions.MicrosoftGraph 1.0.0-beta6

Microsoft.Azure.WebJobs.Extensions.DurableTask 1.4.0

Microsoft.Azure.Webjobs.Extensions.SignalRService 1.0.0-preview1-10002

.NET Assembly Probing and Binding Improvements

- Runtime assemblies isolation
 - Function assemblies loaded in a custom Assembly Load Context
 - Automatic unification (implicit binding redirects)
- Improved probing/resolution behavior
 - Publish artifacts and deps file
 - Automatic resolution of runtime/platform dependencies and native assets
 - Consistent with .NET Core
- Improved extensibility consolidating on new load behavior
 - Common dynamic assembly load mechanism for all extensions
- Out-of-process .NET language worker (not in GA, it's in the roadmap low priority)
 - Serialization / Deserialization of triggers and bindings
 - More flexibility around .NET versions
 - Dependency injection

Assembly Isolation - Bindings: 1.0 Model

Azure Functions Host – Functions 1.0

Host Assembly Load Context (default)

Job host

Bindings

Runtime dependencies

WindowsAzure.Storage 7.2.0

LoadFrom Context

Functions Assemblies

Functions dependencies

WindowsAzure.Storage 9.3.2

```
public static Run(..., CloudBlockBlob blob,...)
{
    //Function code...
}
```



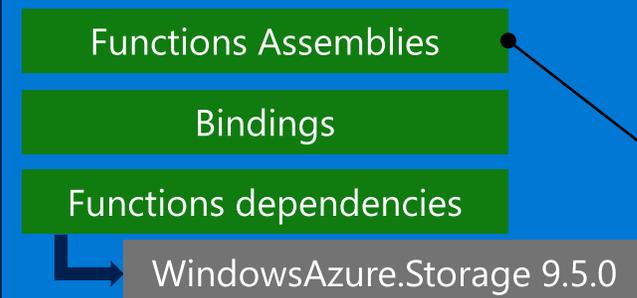
Assembly Isolation - Bindings: 2.0 Model

Azure Functions Host – Functions 2.0

Host Assembly Load Context (default)

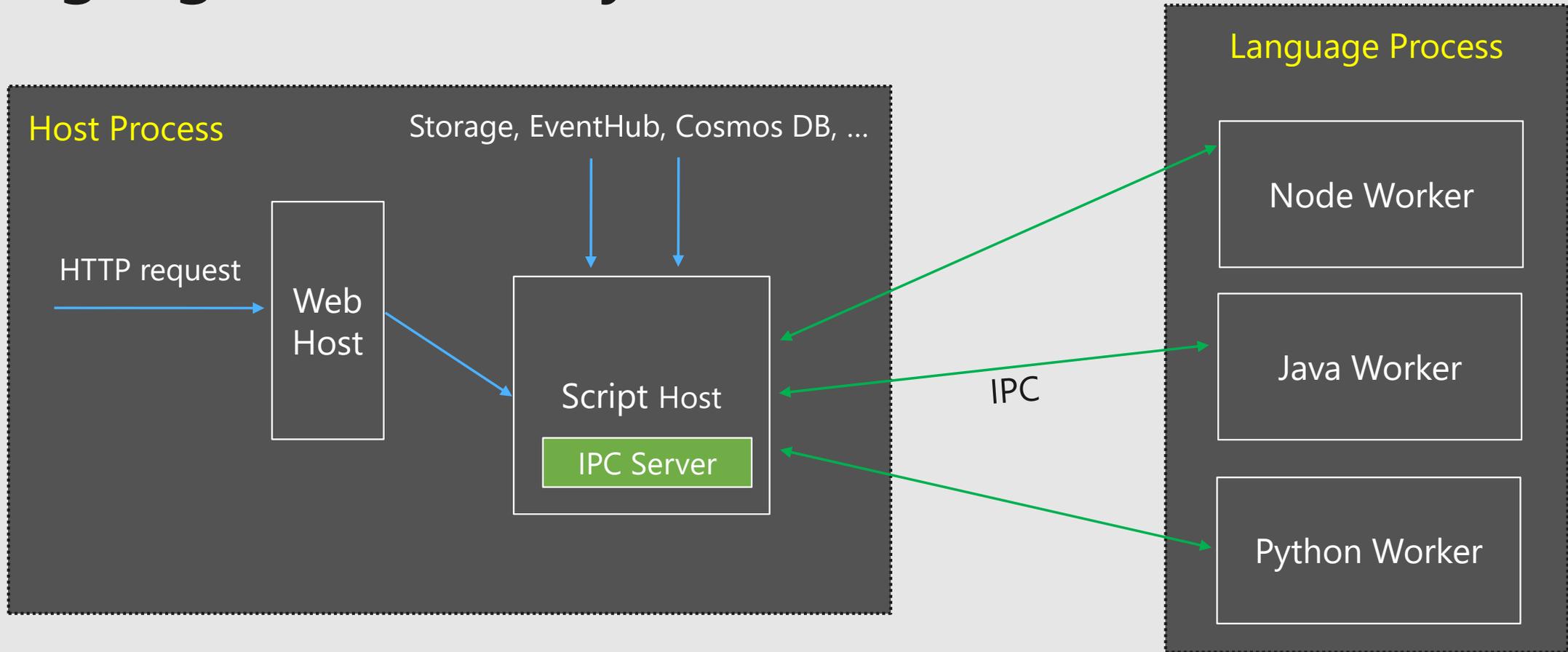


Function Assembly Load Context



```
public static Run(..., CloudBlockBlob blob,...)
{
    //Function code...
}
```

Language Extensibility



- Worker and host broken into 2 separate processes
- Development of new language workers can happen independently
- Worker process crashes doesn't bring down the host

Azure Functions Tooling Options

- Visual Studio



- VS Code



- CLI

- Portal



- Deployment Options



Deployment options: Run from package

Classic Deployment Issues:

1. Not atomic => inconsistent files
2. Files in use get locked
3. Multi-region inconsistencies
4. Difficult rollback

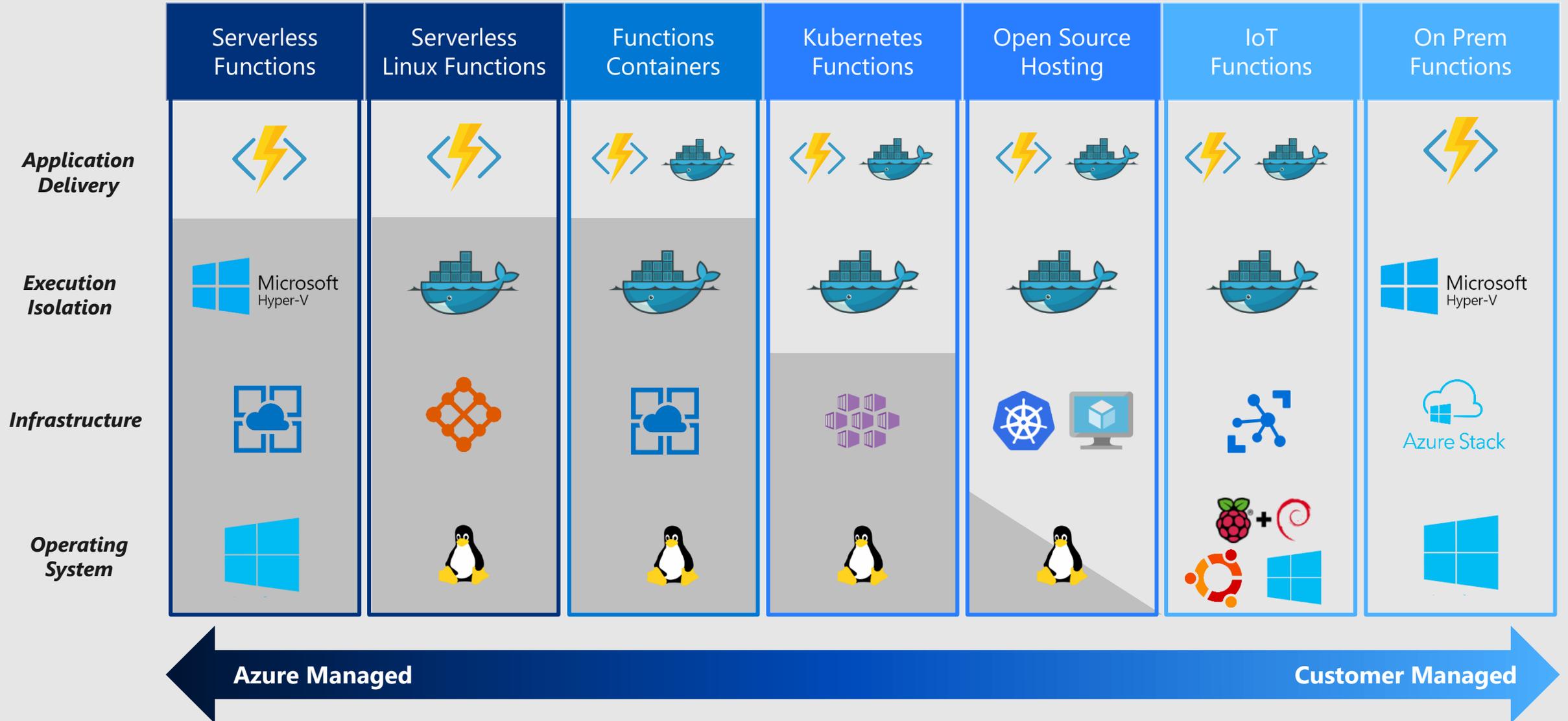
Solutions:

1. Externally hosted zip file
2. Zip file hosted within your app

Azure Functions 2.0 - Recap

- Cross platform
- Assembly probing and binding issues addressed
- Decoupled bindings/extensions
- Language extensibility – out of process language workers
- Additional deployment options
- New tooling options
- New languages
- Closing a very important issue
- Engineering work to enable future scenarios

Functions Hosting Models



Try out the new Functions models

- Linux Consumption – *Preview*
 - aka.ms/functions-consumptionlinux-preview
- Python support - *Preview*
 - aka.ms/linuxconsumptionfunctions
- Functions on Kubernetes – *Preview*
 - <https://medium.com/@asavaritayal/azure-functions-on-kubernetes-75486225dac0>

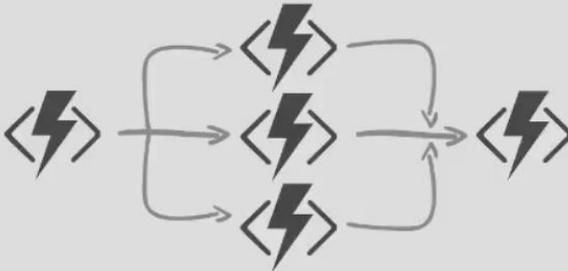
Durable Functions



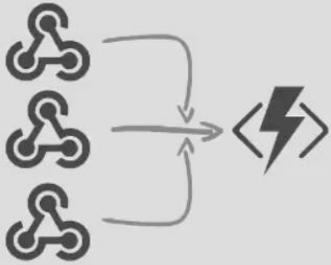
What's still hard?



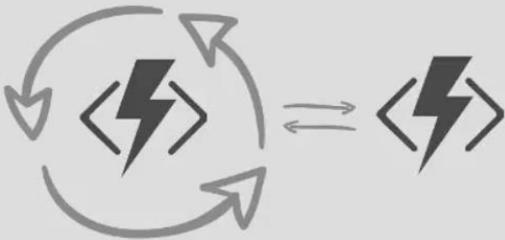
Manageable Sequencing
+ Error Handling / Compensation



Fanning-out & Fanning-in



External Events Correlation



Flexible Automated Long-running
Process Monitoring



Http-based
Async Long-running APIs

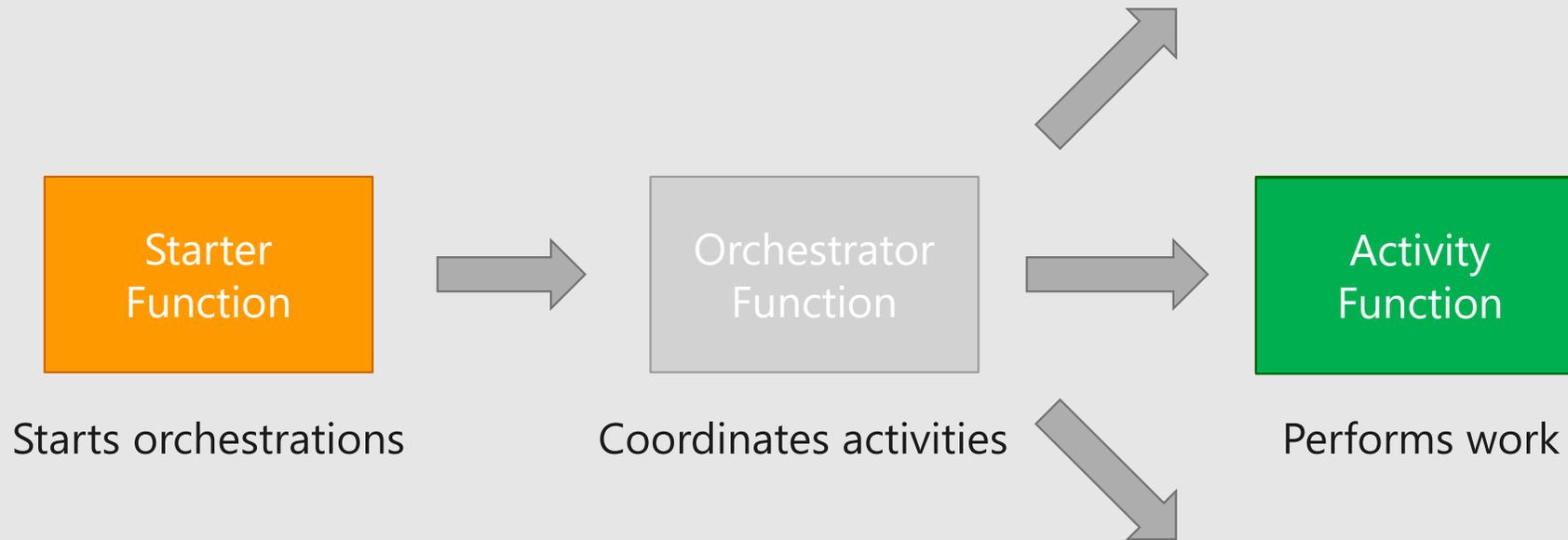


Human Interaction

Durable Functions

- Write **long-running orchestrations** as a **single function** while **maintaining local state**.
- **Simplify complex transactions and coordination (chaining, etc.)** Easily call a Function from another Function, synchronously or asynchronously.
- All of the above using code-only. No JSON schemas. No graphical designer.
- Announcing: **Generally available (v2) - C# and JavaScript**

Components



OrchestrationClient

OrchestrationContextTrigger

ActivityTrigger

What It Looks Like

```
// calls functions in sequence
```

Orchestrator Function

```
public static async Task<object> Run(DurableOrchestrationContext ctx)
```

```
{
```

```
    try
```

```
    {
```

```
        var x = await ctx.CallActivityAsync("F1");
```

```
        var y = await ctx.CallActivityAsync("F2", x);
```

```
        return await ctx.CallActivityAsync("F3", y);
```

```
    }
```

```
    catch (Exception)
```

```
    {
```

```
        // global error handling/compensation goes here
```

```
    }
```

```
}
```

Activity Functions

Takeaways

- Azure Functions is advancing quickly, but there are tons of resources to stay up to date
 - aka.ms/functionslinks
- Azure Functions is open source with plenty of contribution – so thanks to all of you!
- Feedback is never enough, so please continue to use and contribute
 - GitHub / StackOverflow / MSDN
 - Twitter: @azurefunctions



Grazie!



@_geniodelmale

lorenzo.barbieri@microsoft.com

Materiale su

<http://aspit.co/netconf-18>

