

Welcome!

**Wi-Fi: API-SLA-CON
Pwd: conference**

Please complete the prerequisites:

<https://bit.ly/2Fdddnl>

Azure: Serverless Infrastructure as Code

Serverless Architecture Conference | August 17 | 2020

Mikhail Shilkov

© Mikhail Shilkov



Mikhail Shilkov

- Software engineer @ Pulumi
- Serverless, Azure Functions
- Functional programming
- Microsoft Azure MVP

<https://mikhail.io>
mikhail@pulumi.com
@MikhailShilkov

© Mikhail Shilkov



Stack Overflow Top 10 Users on Azure Functions

All Time

1.5k	514	 Mikhail Shilkov 27.7k • 3 • 43 • 76
751	164	 Fabio Cavalcante 10.9k • 3 • 27 • 39
709	308	 Jerry Liu 14.5k • 1 • 22 • 41
587	115	 mathewc 12.1k • 1 • 34 • 48
500	106	 David Ebbo 39.1k • 6 • 79 • 105
371	38	 Chris Anderson-MSFT 6,765 • 2 • 19 • 34
197	72	 Chris Gillum 12.4k • 4 • 38 • 47
185	44	 ahmelsayed 6,147 • 2 • 23 • 39
179	89	 Matt Mason 2,456 • 5 • 18
166	253	 Bowman Zhu 4,763 • 1 • 2 • 14

4x sessions, 1.5 hours each

~20 minutes - intro

~60 minutes - hands-on lab

~10 minutes - review and discussion

Agenda

09:00 - 10:30 **Infrastructure as Code**

10:30 - 11:00 **Coffee Break**

11:00 - 12:30 **Serverless, Azure Functions**

12:30 - 13:30 **Lunch**

13:30 - 15:00 **Event Hubs, Cosmos DB, API Management, CDN, ...**

15:00 - 15:30 **Coffee Break**

15:30 - 17:00 **Cost, scalability, cold starts, configuration of Azure Functions**

Questions

- 1. At the microphone**
- 2. Conference Slack**
- 3. In-person during breaks**

Infrastructure as Code



Cloud Evolution

Past

Virtual Machines
on demand =>
Lift and shift

Present

100s of managed
services =>
Cloud-native

Future

Abstractions for “the
most powerful
computer ever” =>
Cloud engineering

```
1 using System.Net;
2
3 public static HttpResponseMessage Run(HttpRequestMessage req, out object
4 {
5     string name = req.GetQueryNameValuePairs()
6         .FirstOrDefault(q => string.Compare(q.Key, "name", true) == 0)
7         .Value;
8
9     string task = req.GetQueryNameValuePairs()
10        .FirstOrDefault(q => string.Compare(q.Key, "task", true) == 0)
11        .Value;
12
13     string duedate = req.GetQueryNameValuePairs()
14        .FirstOrDefault(q => string.Compare(q.Key, "duedate", true) == 0)
15        .Value;
16
17     taskDocument = new {
18         name = name,
19         duedate = duedate.ToString(),
20         task = task
21     };
22 }
```

Azure Portal:
Good for exploration
Not reproducible

DevOps

The goal of DevOps is to
make software delivery
vastly more efficient

Faster delivery of **value**,
better **teamwork**,
reduce defect rates

Automate all routine

Infrastructure as Code (IaC)

Write and execute **code** to **define**, deploy, update, and destroy your **infrastructure**

Manage almost **everything in code**, including servers, databases, networks, log files, application configuration, documentation, automated tests, deployment processes

IaC flavors

Configuration management tools
(Ansible – install software on servers)

Server templating tools
(Docker, Vagrant)

Orchestration tools (Kubernetes)

Provisioning
(create servers and other infra)

IaC flavors

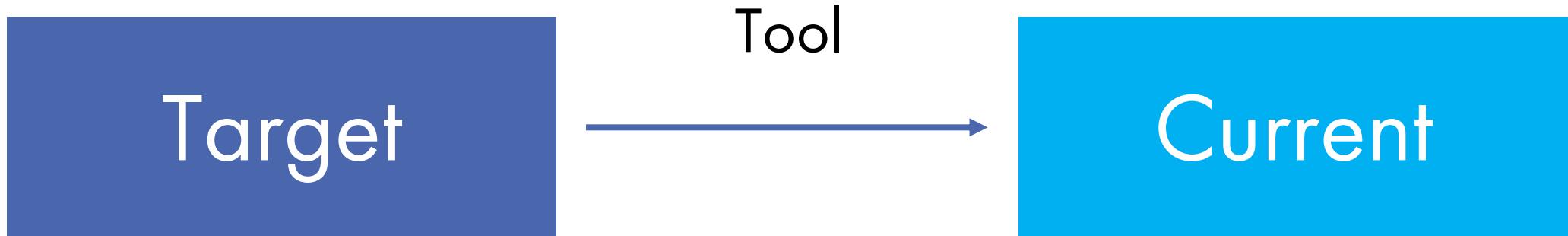
Configuration management tools
(Ansible – install software on servers)

Server templating tools
(Docker, Vagrant)

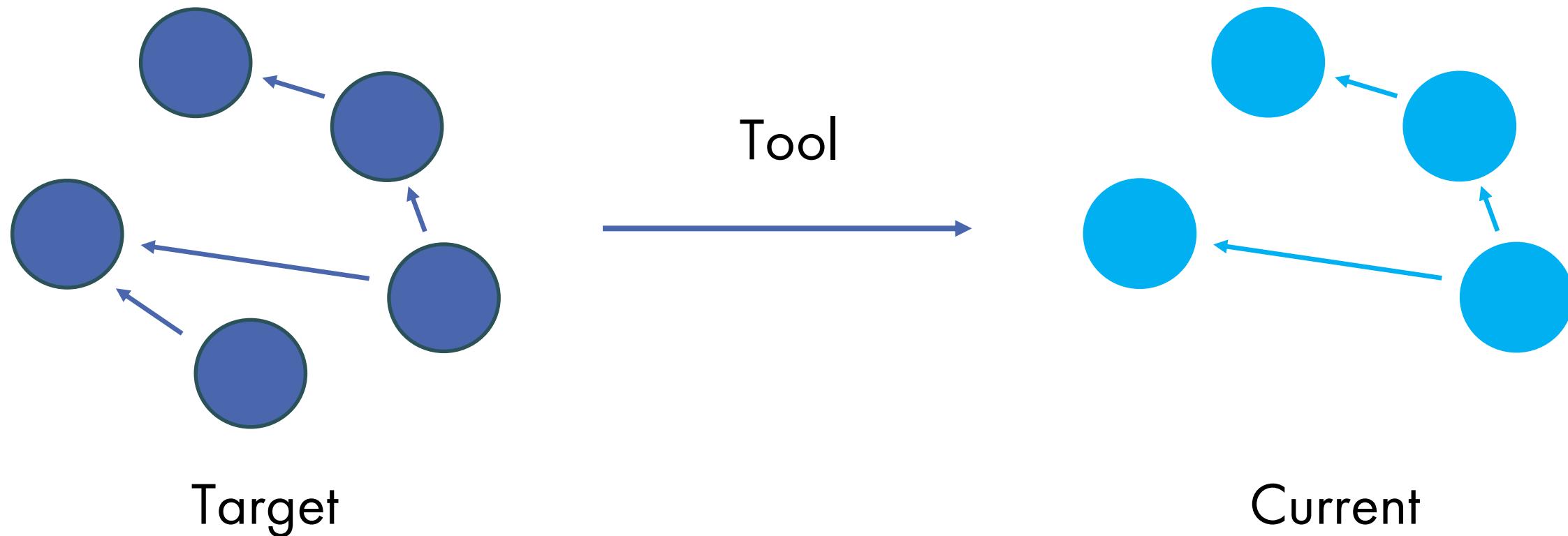
Orchestration tools (Kubernetes)

Provisioning
(create servers and other infra)

Desired State Configuration



Managing Resource Graphs



Benefits of IaC

- Self-service
- Version Control
- Reuse
- Documentation
- Speed and Safety
- Happiness

IaC Tools: Tradeoffs

- Procedural vs.
Declarative
- “Driver” Location
(Agent vs. Client vs. Server)
- Code Format and
Language
- Single-cloud vs.
Provider-based
- Feature Coverage
- Maturity and
Ecosystem

Pulumi



Pulumi

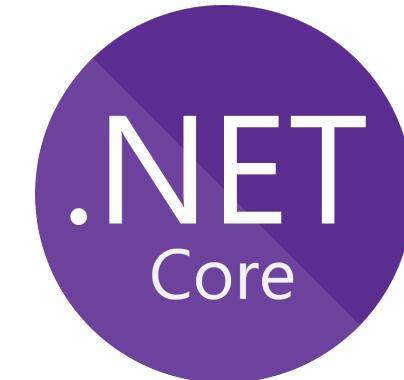
© Mikhail Shilkov

Made by Pulumi Corp.

**Written in Go,
open source
Apache License 2.0**

Since 2018

General-purpose Programming Languages



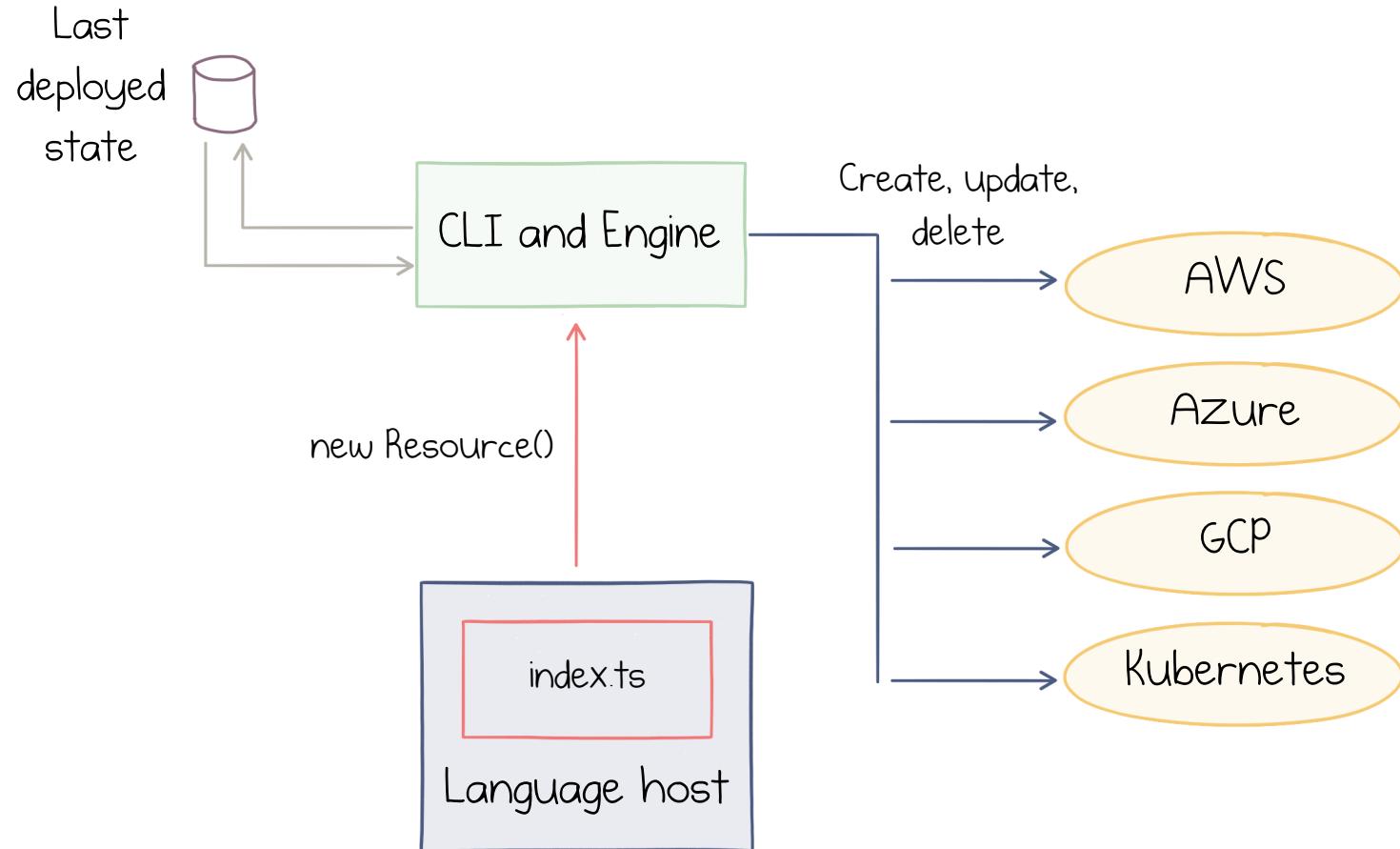
TypeScript example

```
const storageAccount = new azure.storage.Account("storage", {  
    resourceGroupName: resourceGroup.name,  
    accountReplicationType: "LRS",  
    accountTier: "Standard",  
});
```

Providers

- AWS
 - Azure
 - GCP
 - Digital Ocean
 - Cloudflare
-
- Docker
 - Kubernetes
 - OpenStack
 - PostgreSQL
 - New Relic

... and more



How Pulumi works

Workflow: Preview, Up, Destroy

pulumi preview

Displays a draft of changes, does nothing to the infra

pulumi up

Rolls the infra from its current state to the desired one

pulumi destroy

Deletes all resources in the current stack

Pulumi

- Procedural vs.
Declarative
- “Driver” Location
(Agent vs. Client vs. Server)
- Code Language:
TypeScript/Python/
Go/C#/F#
- Single-cloud vs.
Provider-based
- Feature Coverage:
Varies per Provider
- Open source

Labs

[**Lab 1 Modern Infrastructure as Code**](#)

Lab 2 Serverless Intro

[**Lab 3 Data processing pipeline**](#)

Lab 4 HTTP Azure Functions

[**Lab 5 Static web app**](#)

Lab 6 Azure API Management

[**Lab 7 Azure CDN**](#)

Lab 8 Azure Active Directory

Labs: Tips

**You don't have to complete
all the labs:
Understand, do NOT rush**

Ask questions

Enjoy!

Coding time!

Please follow the labs:

<https://bit.ly/3gPCJNv>

or

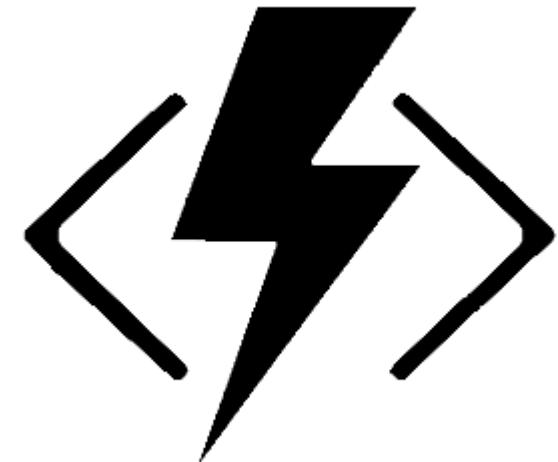
[https://github.com/mikhailshilkov/
azure-serverless-workshop](https://github.com/mikhailshilkov/azure-serverless-workshop)



Coffee Time
see you at 11:00

Serverless

Azure Functions



Key Features



Delegated
Management



Scalability and
High Availability
Out of the Box



Pay per Use

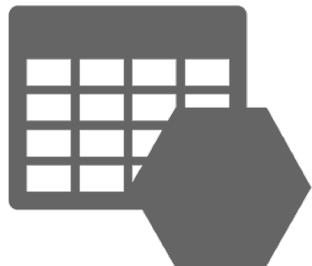
Paul Johnston

**“A simple definition
of Serverless”**

A Serverless solution is one
that costs you nothing to run
if nobody is using it
(excluding data storage costs)

Examples

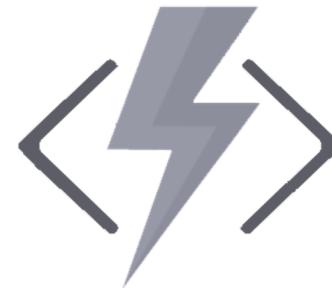
Azure Storage



Azure Event Grid



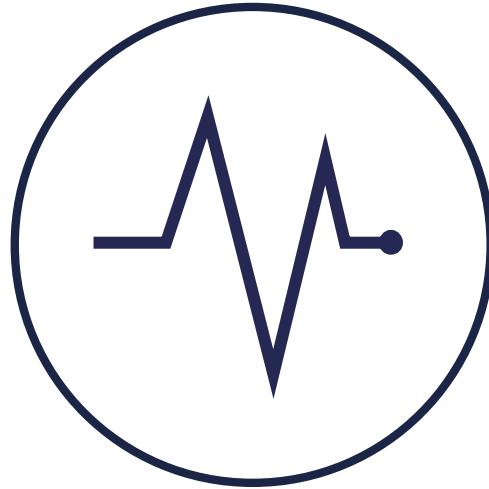
Azure Functions



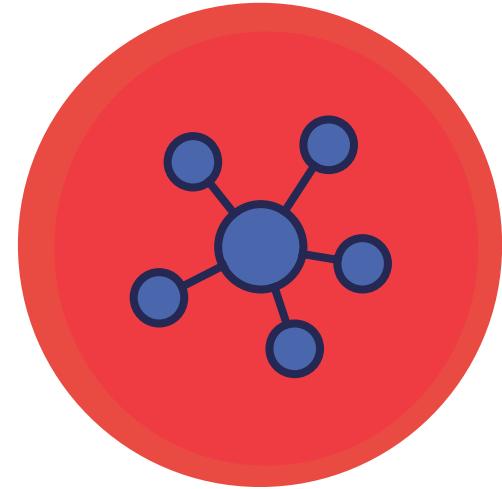
Azure Functions: Opiniated Development Model



Event-
Driven



Short-
Lived



Stateless

Azure Functions: Sample Scenarios

Timer

Runs once per day,
executes a
PowerShell script in
5 minutes

HTTP

High-load web
service with 100s or
requests per second

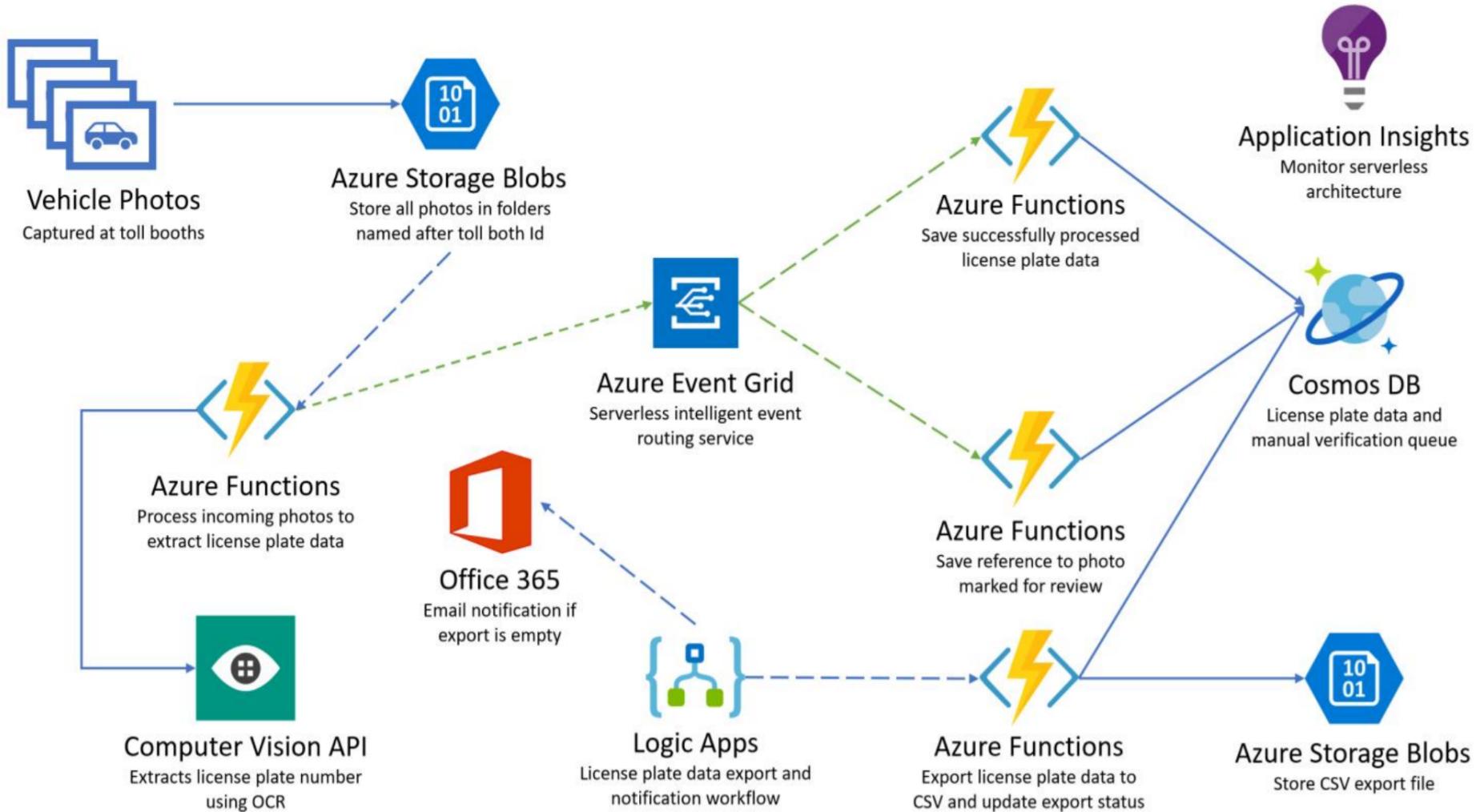
Queue

Bursts of incoming
messages, varying
processing time

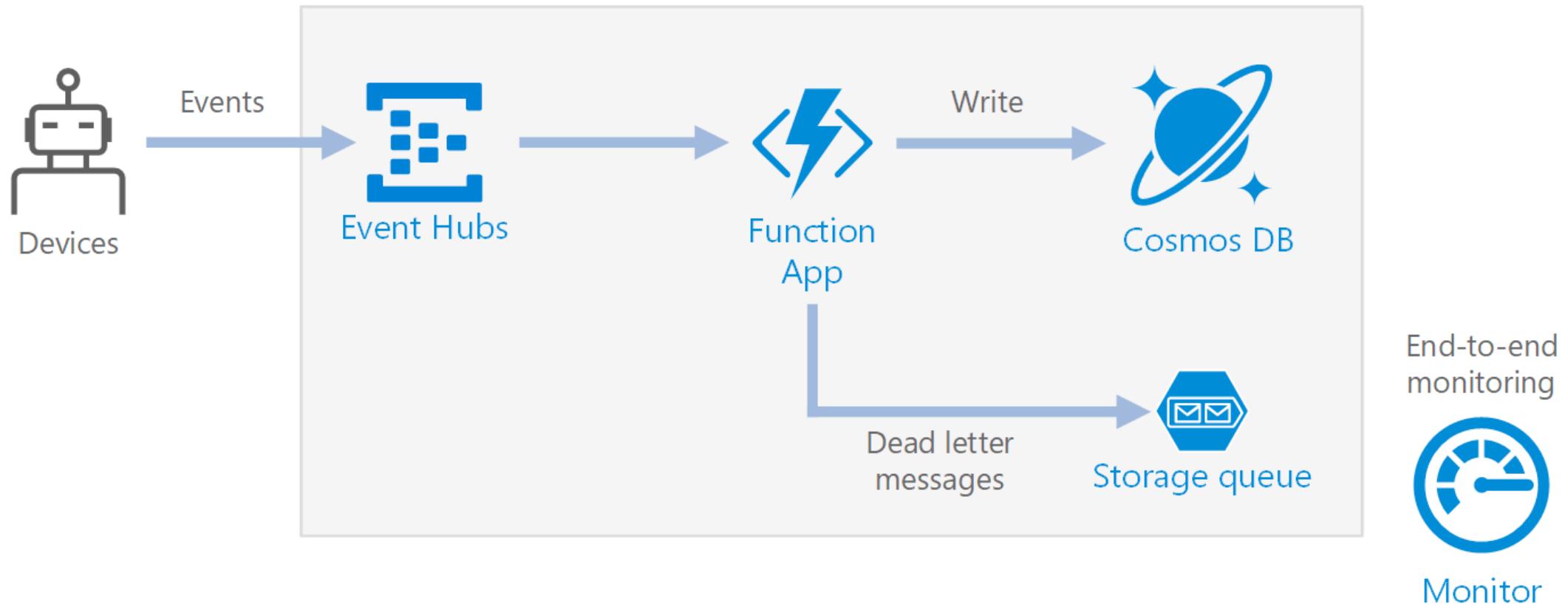
Azure Functions: Triggers and Bindings

```
[FunctionName("MyFunc")]
public static void Work(
    [TimerTrigger("0 */10 * * *")] TimerInfo timer
    [Blob("filename")] Stream stream,
    [Queue("myqueue")] out string message)
{
    // body
}
```

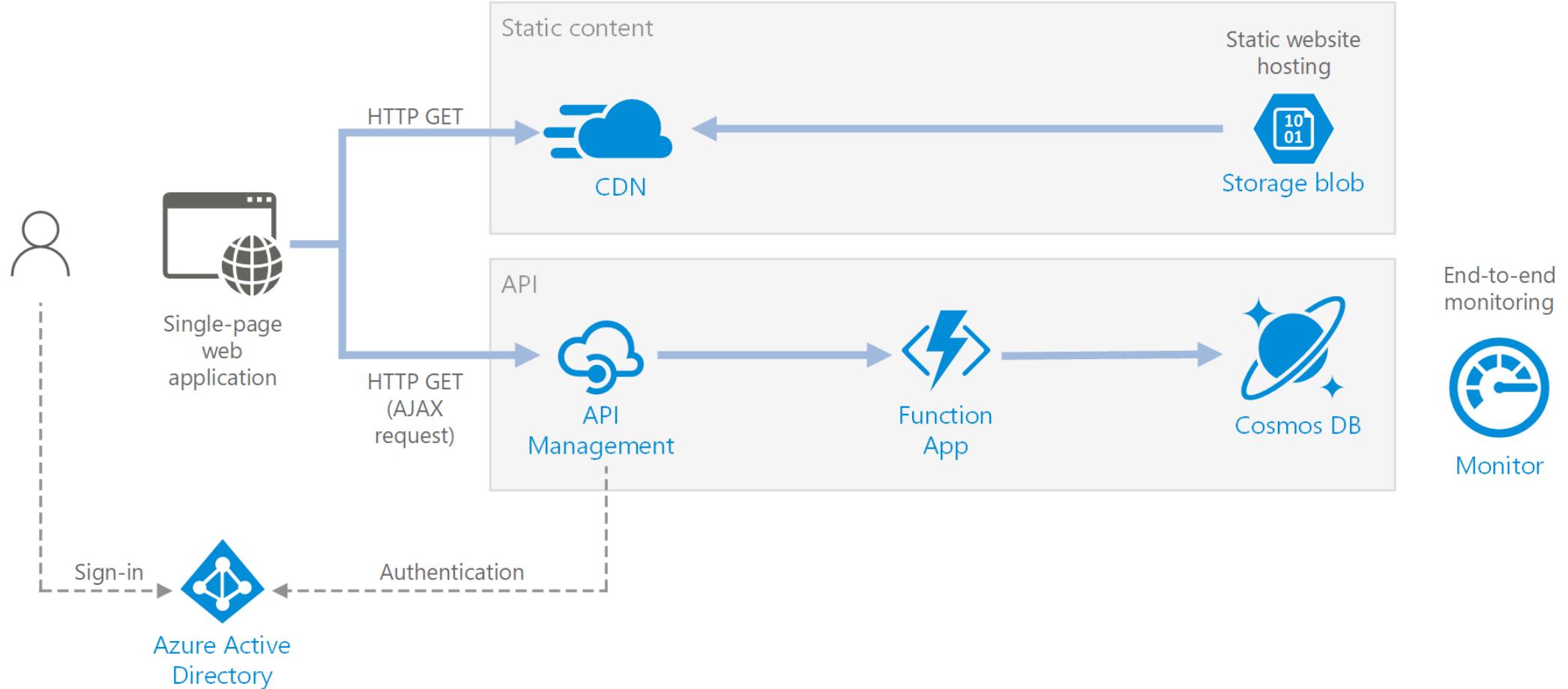
Cloud Glue



Serverless Event Processing



Serverless Web Applications



Coding time!

Get back to the (same) labs:

<https://bit.ly/3gPCJNv>

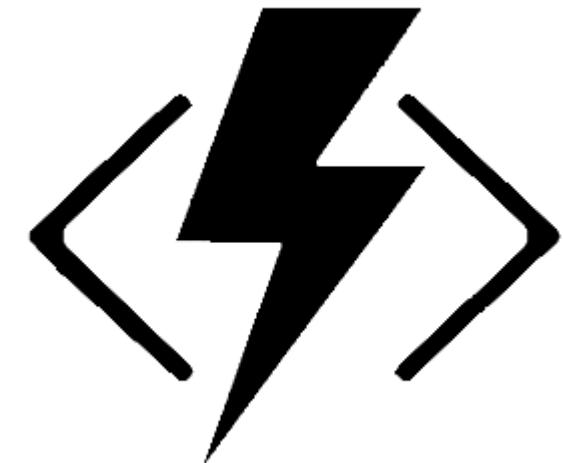
or

[https://github.com/mikhailshilkov/
azure-serverless-workshop](https://github.com/mikhailshilkov/azure-serverless-workshop)



A Tour of Azure Services

that you use in the labs



Resource Group

Each resource can only exist in one resource group

Resources in a group should share the same lifecycle

A resource group can contain resources located in different regions

Azure Storage

Blobs

Massively-scalable object storage for unstructured data

Queues

Basic messaging service

Tables

Semi-structured data that's highly available

Azure Event Hubs

What

Kafka-like
messaging service
based on event
logs

Why

Simple, scalable,
secure, pay-as-
you-go

Pricing

Per hour per
throughput unit,
per million events

Azure Cosmos DB

What

Distributed
managed NoSQL
database service

Why

Simple, scalable,
low latency,
compatible to
Mongo,
Cassandra, etc.

Pricing

Per hour for
provisioned
throughput with
autoscaling
options

Azure API Management

What

Single place for managing all your APIs

Why

Unified experience and solution to API management tasks

Pricing

Consumption (pay per use) or fixed (per hour)

Azure CDN

What

Content delivery
network

Why

Distributed,
global,
performant,
integrated in
Azure

Pricing

Per GB

Azure Application Insights

What

Application
Performance
Management
(APM), part of
Azure Monitor

Why

Telemetry,
logging, alerts,
dashboard,
troubleshooting

Pricing

Per GB

Key Features



Delegated
Management



Scalability and
High Availability
Out of the Box



Pay per Use

Coding time!

Get back to the (same) labs:

<https://bit.ly/3gPCJNv>

or

[https://github.com/mikhailshilkov/
azure-serverless-workshop](https://github.com/mikhailshilkov/azure-serverless-workshop)



Coffee Time
see you at 15:30

Azure Functions



Hosting Options

App Service

Per-per-hour

Consumption

Pay-per-use

Premium

A combination:
Per-per-use with a
fixed component

Azure Functions Consumption

Single SKU

Single-core CPU

1.5GB RAM

Undisclosed SKU

Azure Functions Premium

EP1

3.5GB RAM
1 Dv2 Core

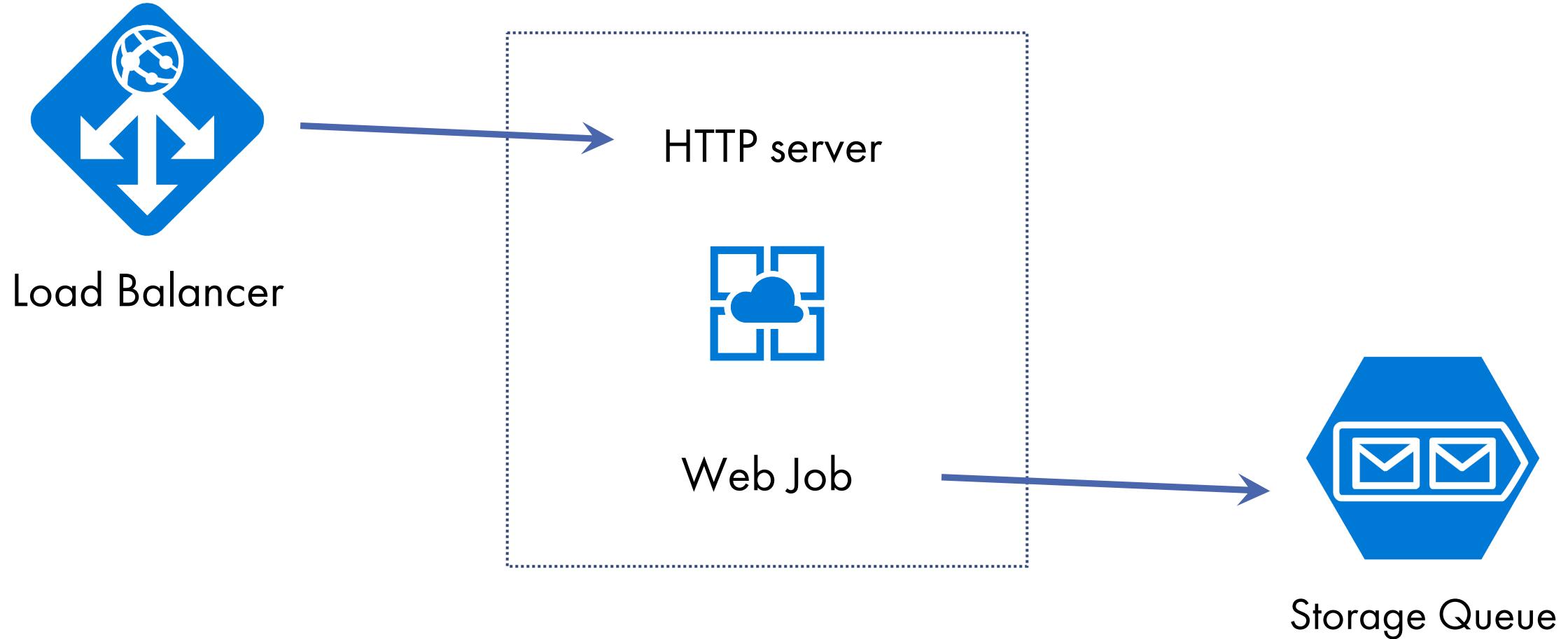
EP2

7GB RAM
2 Dv2 Cores

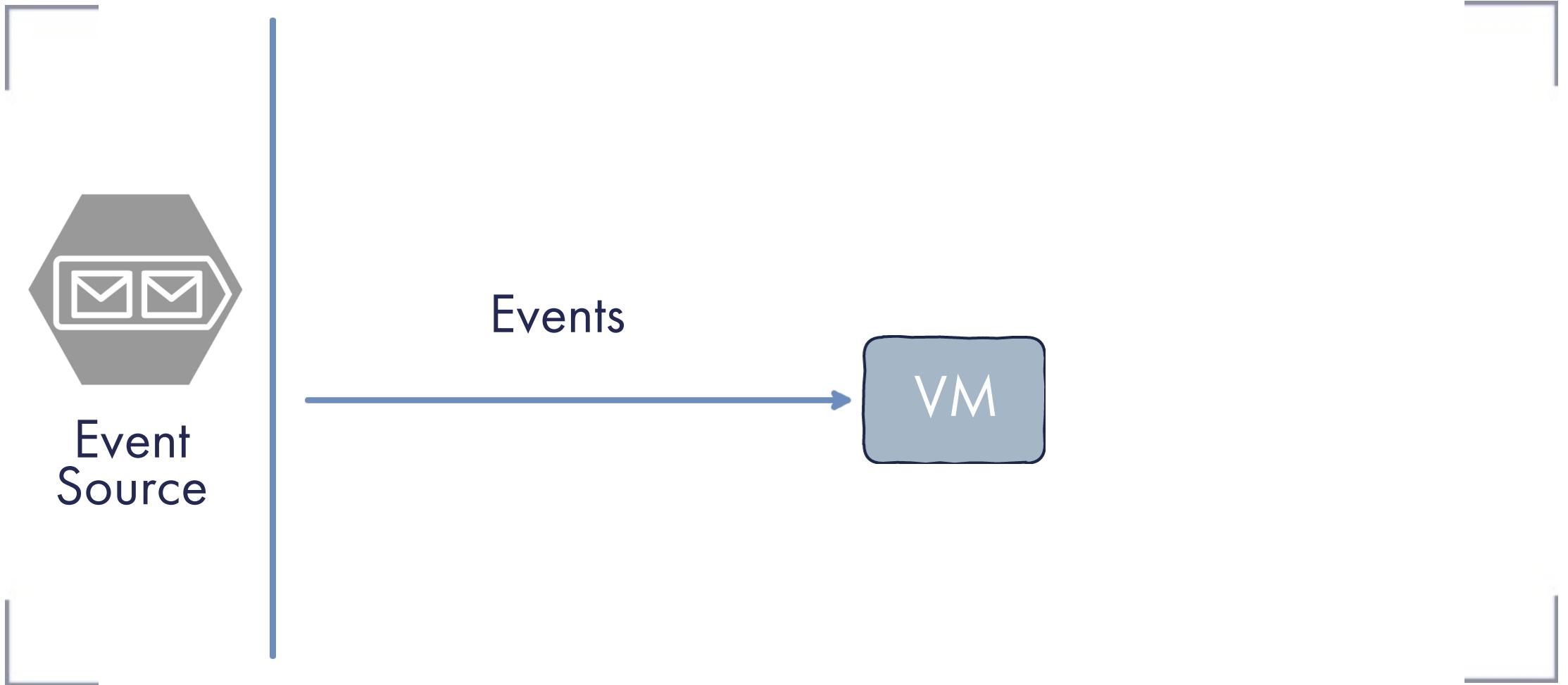
EP3

14GB RAM
4 Dv2 Cores

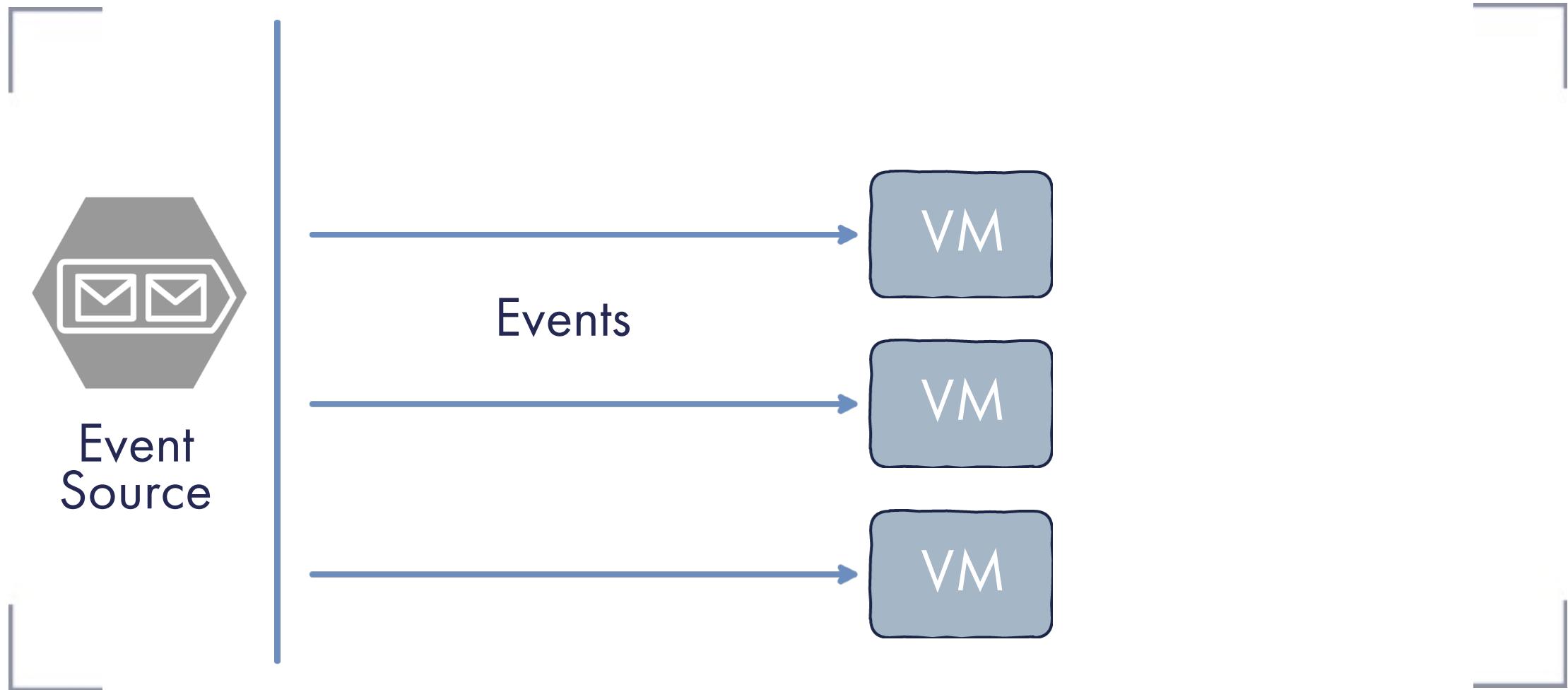
Legacy of App Service



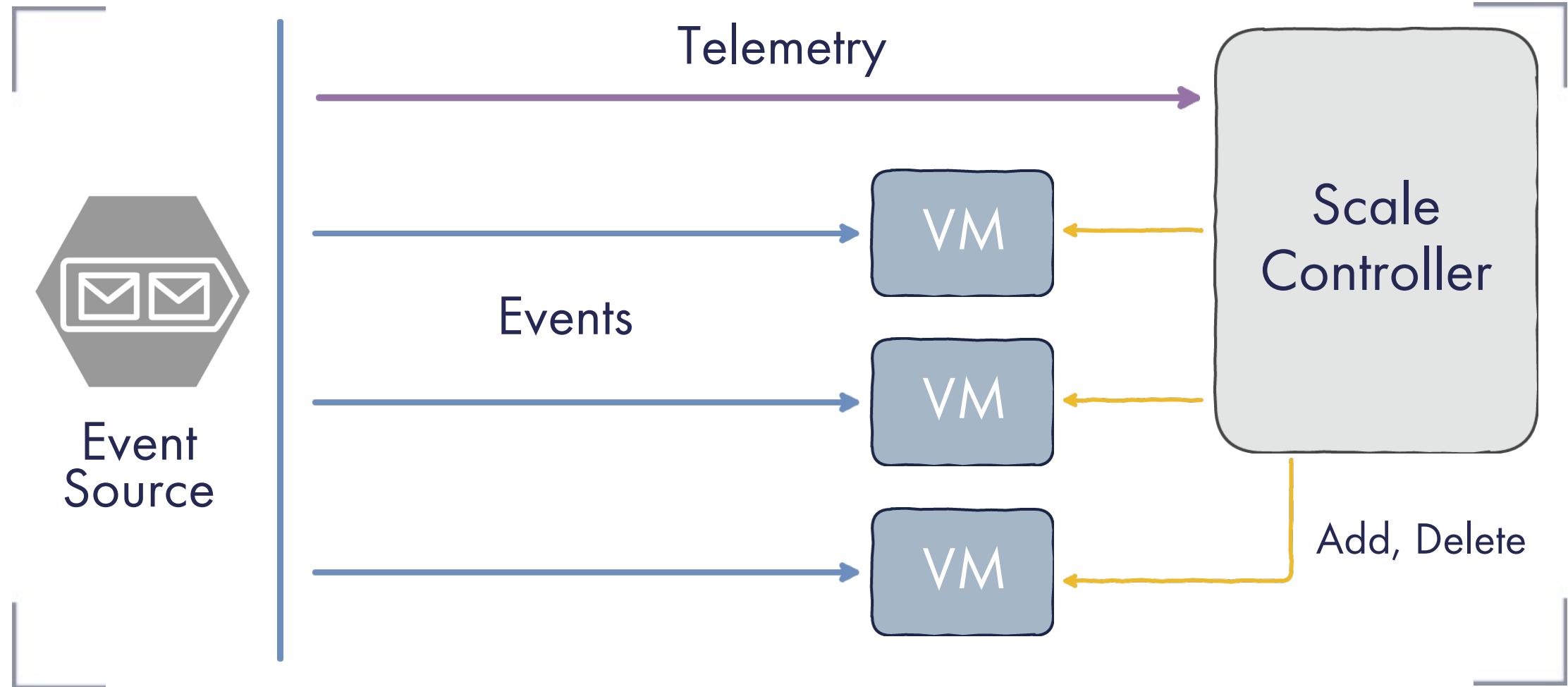
Legacy of App Service



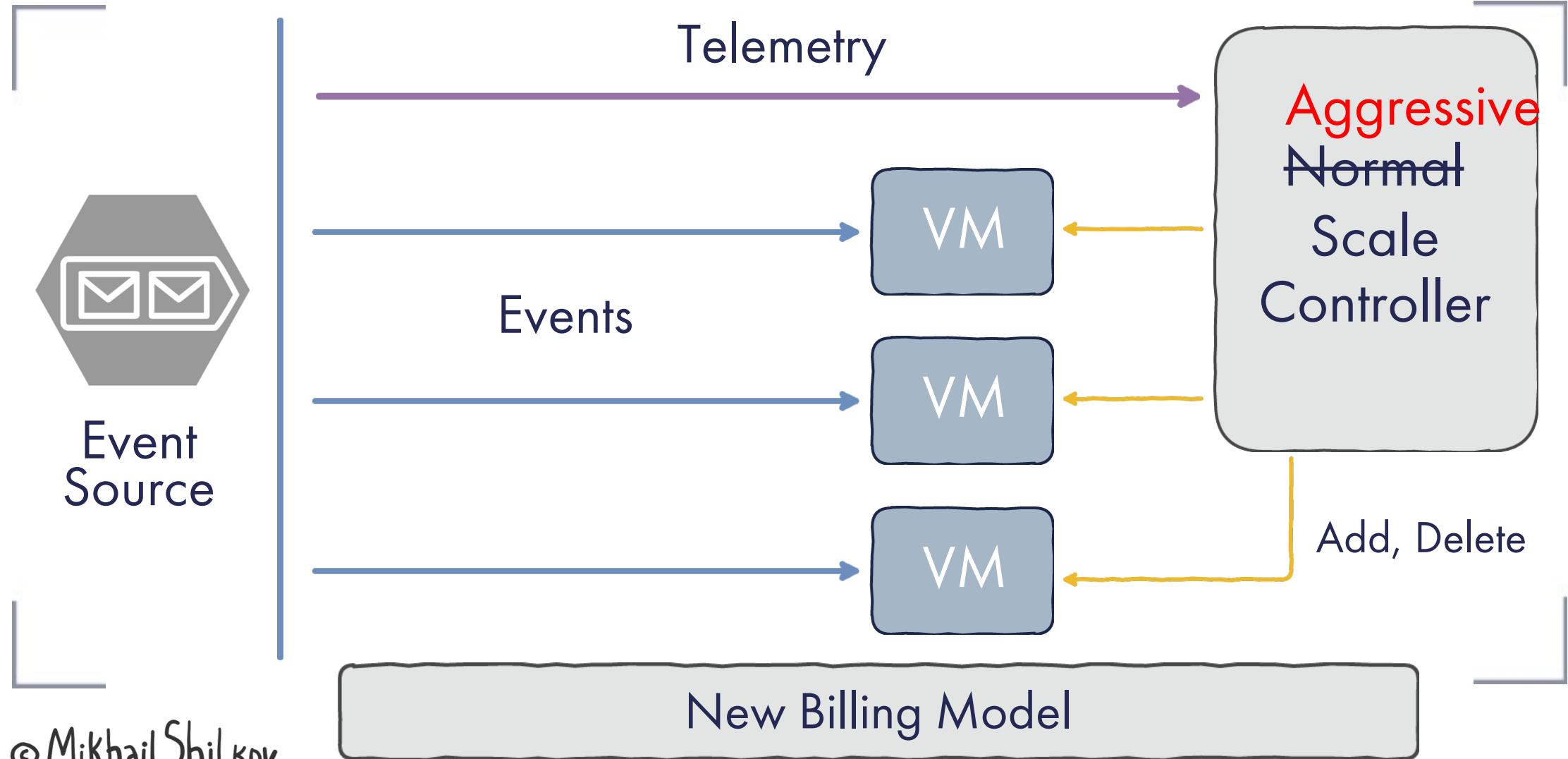
App Service: scalability



App Service: scalability



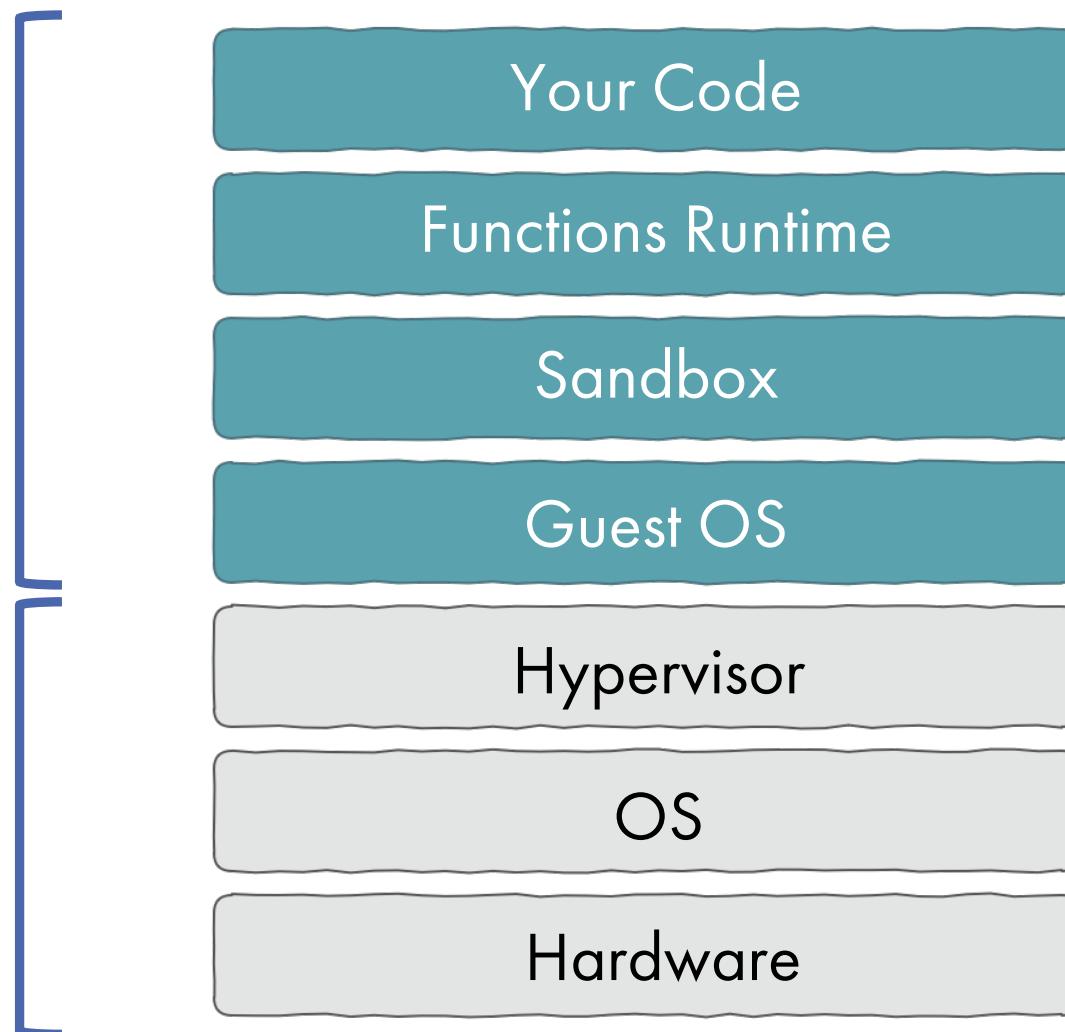
Function App Consumption Plan



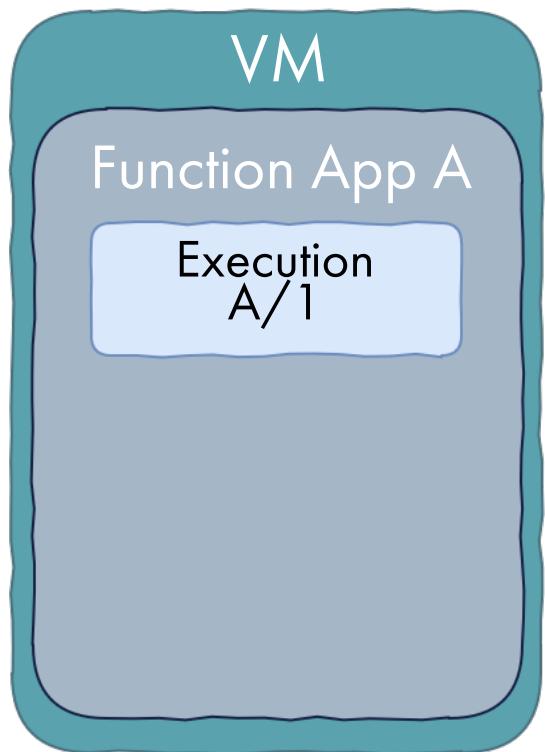
Isolation Layers

One Function App
(VM)

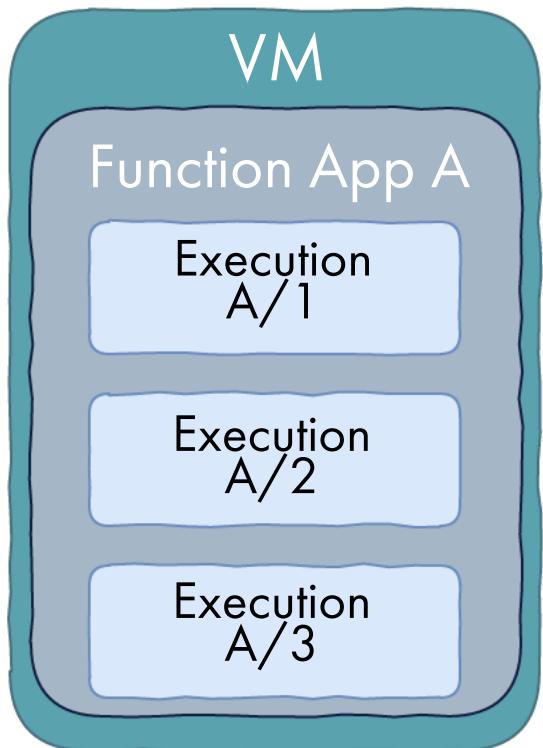
Difference Apps



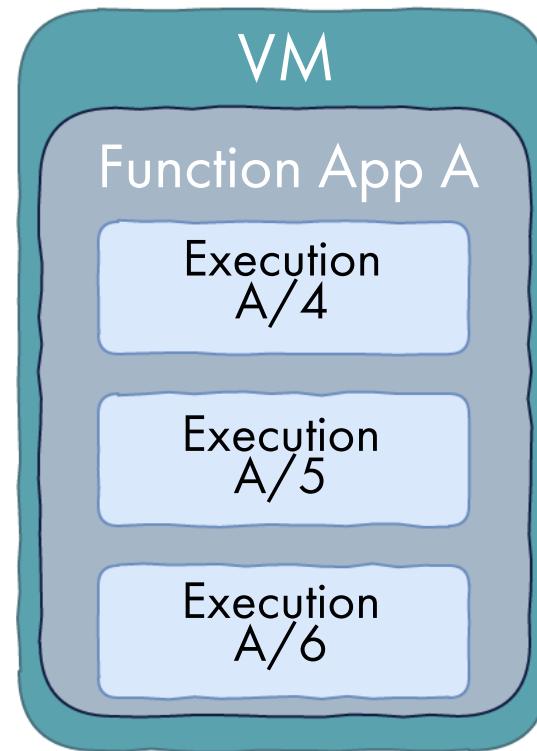
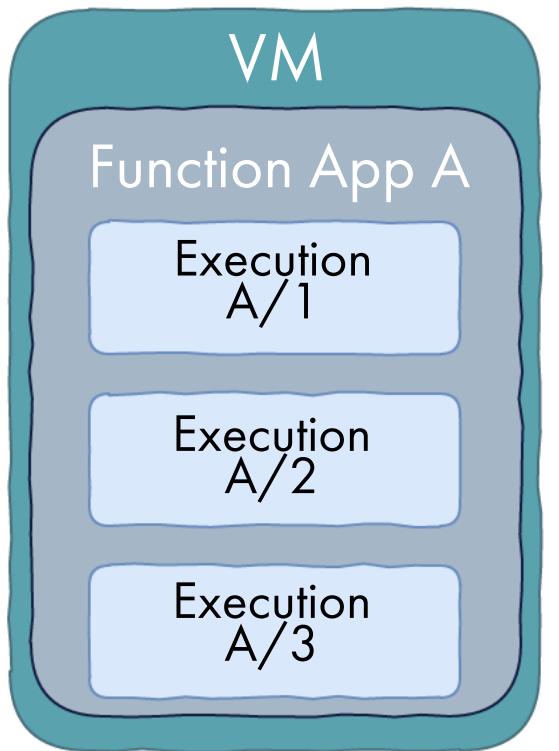
Simultaneous Executions



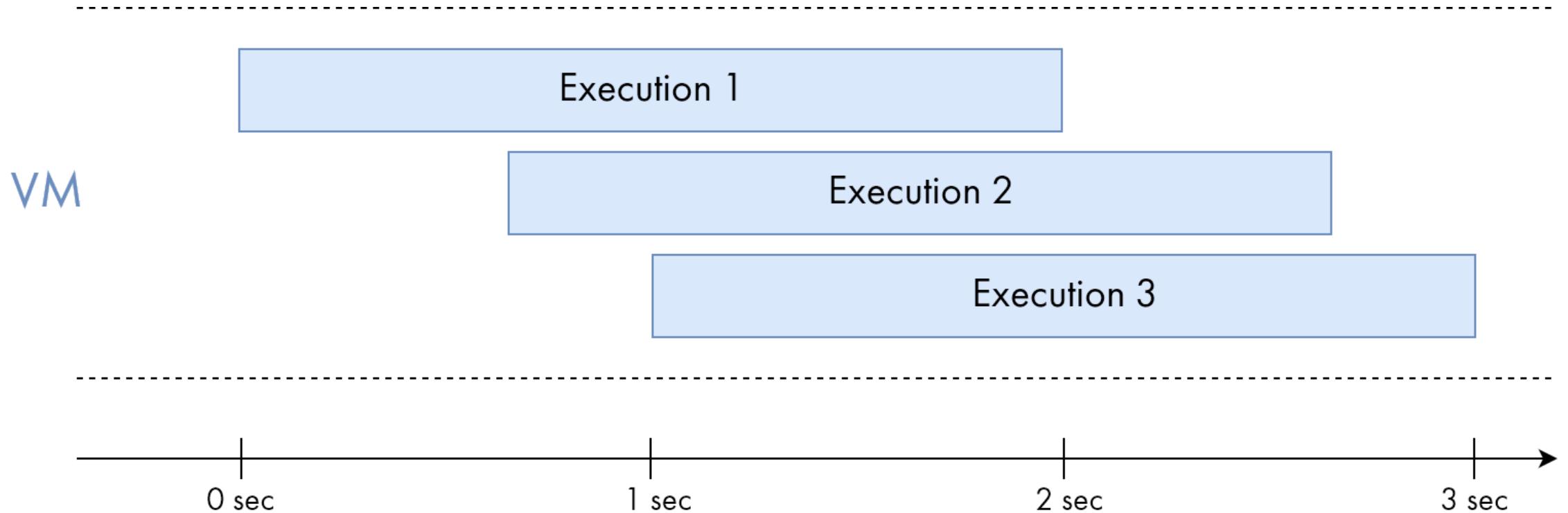
Simultaneous Executions



Simultaneous Executions



Simultaneous Executions



Cost



Azure Functions Pricing

Nominal

\$0.20 per million
requests

\$16 per million
GB^{*}seconds

Instances

Charged per
active instances,
not requests

Charged for

Actual RAM
usage
(128MB min)

Per 1 ms
(100ms min)

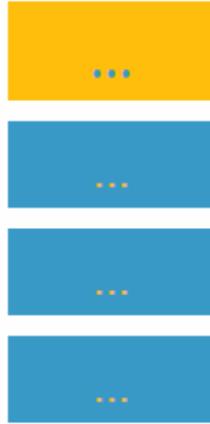
Cold Starts



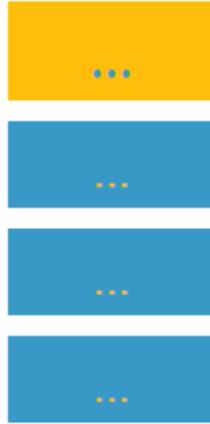
Example: Loading a map



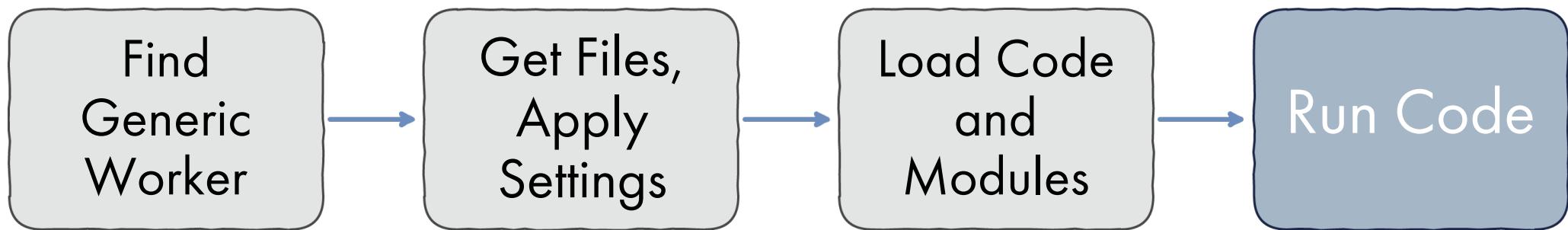
Example: Loading a map



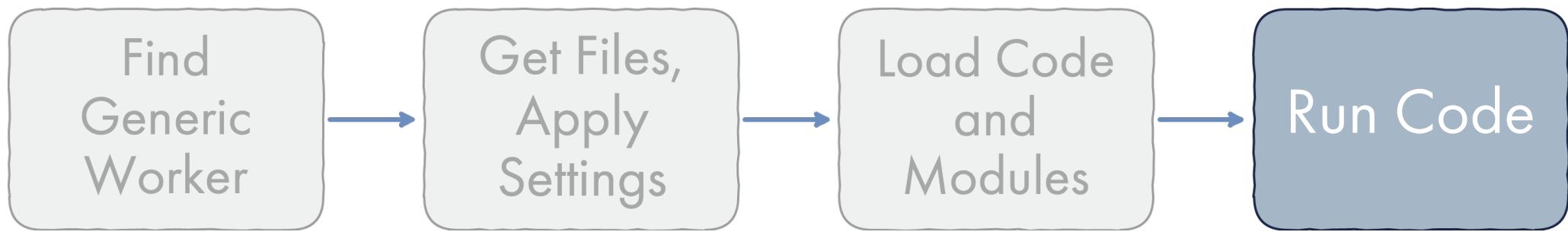
Example: Loading a map



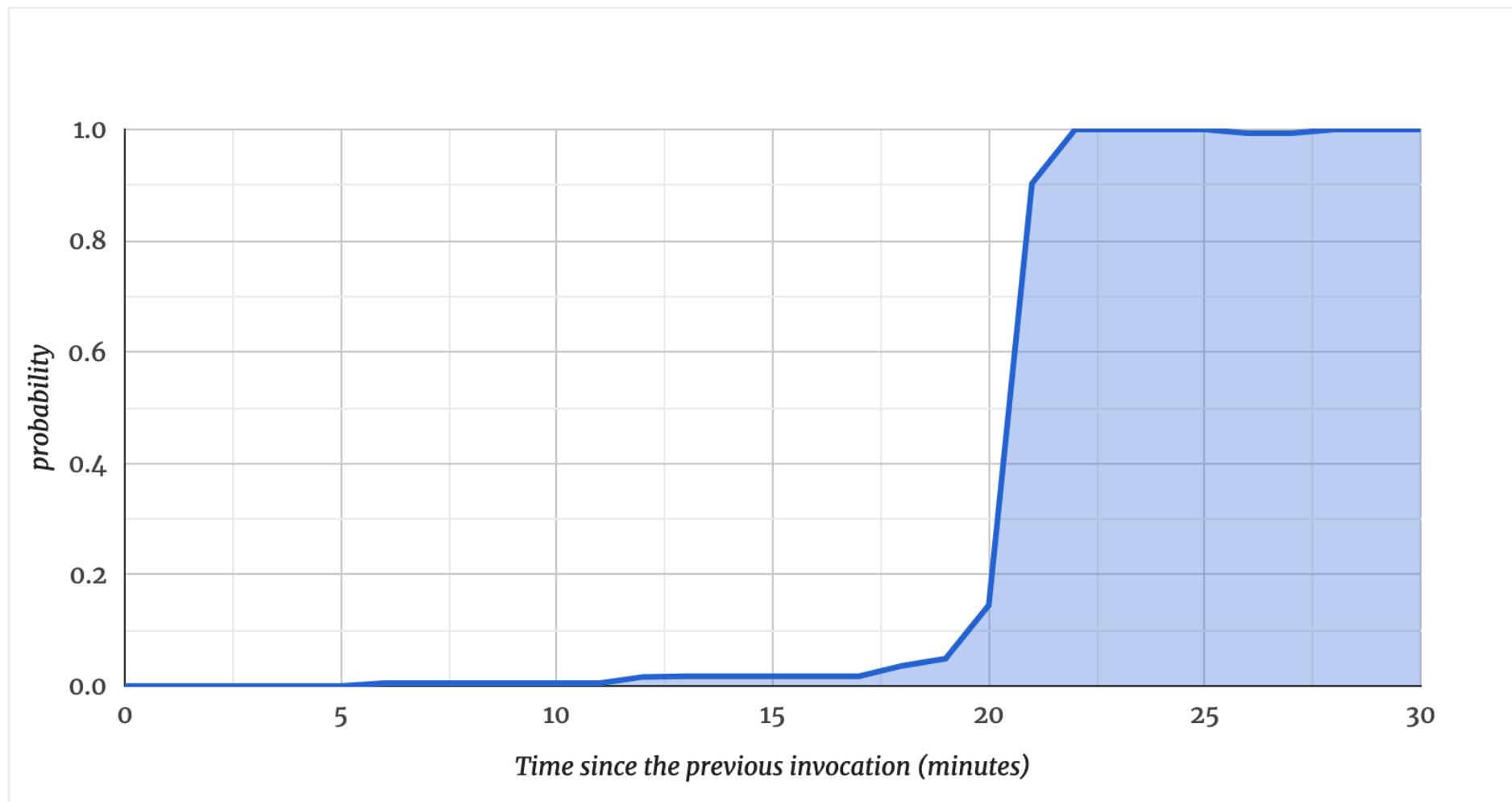
Cold Start



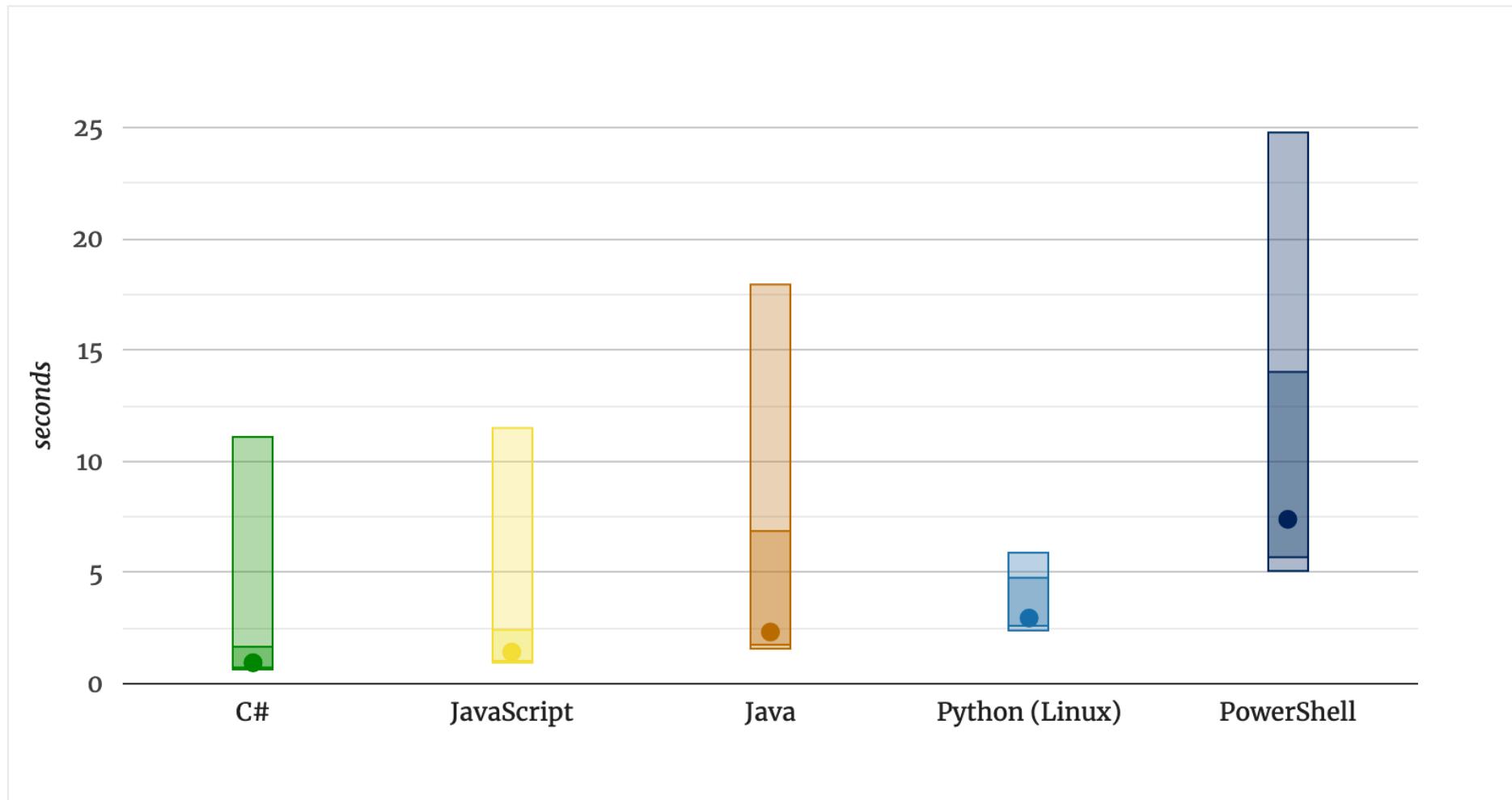
Warm Start



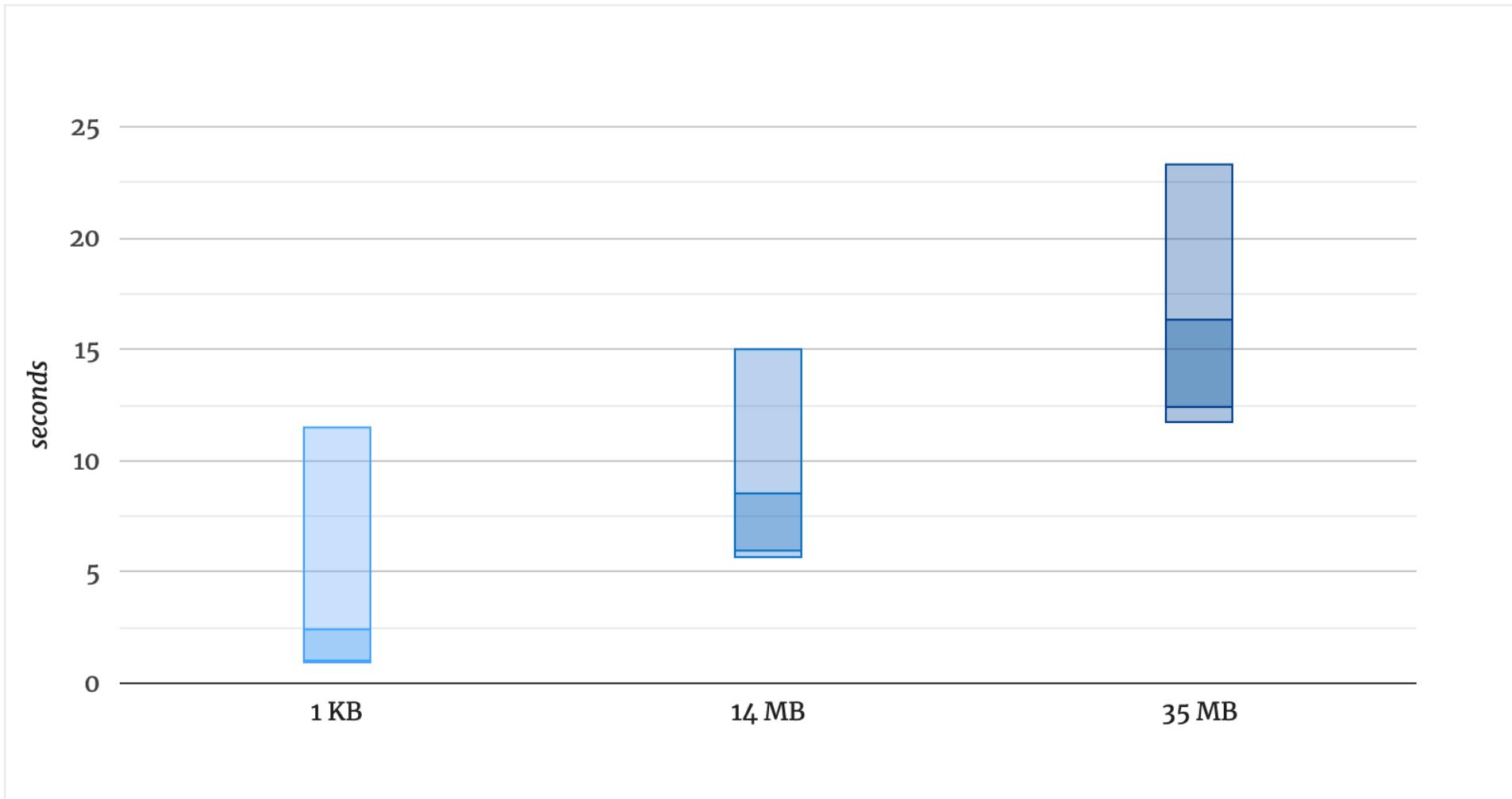
Period Before a Subsequent Cold Start



Cold Starts: Language Comparison



Dependencies Add to Cold Start



Coding time!

Get back to the (same) labs:

<https://bit.ly/3gPCJNv>

or

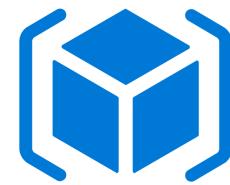
[https://github.com/mikhailshilkov/
azure-serverless-workshop](https://github.com/mikhailshilkov/azure-serverless-workshop)

Conclusions



Making
Cloud
Apps?

USE INFRASTRUCTURE- AS-CODE



Fine-grained components

Automation



Thank you!

@MikhailShilkov
<https://mikhail.io>