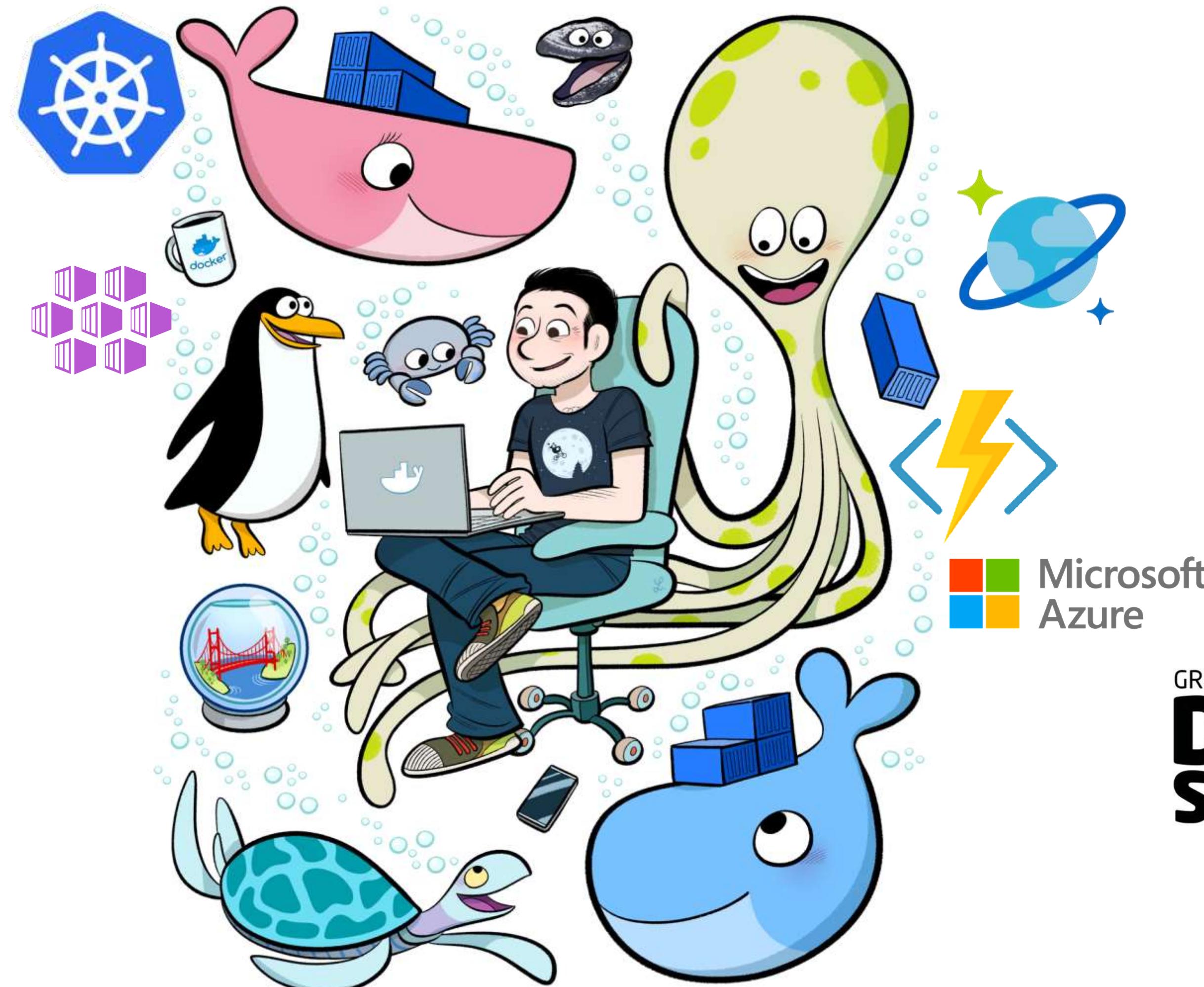


# Developing Apps with Containers, Functions and Cloud Services



Microsoft  
Azure

GREAT INTERNATIONAL  
**DEVELOPER**  
SUMMIT

Microsoft

Patrick Chanezon  
Cloud Advocate, Microsoft  
@chanezon





# @chanezon

1994-2005

Software Engineer

accenture



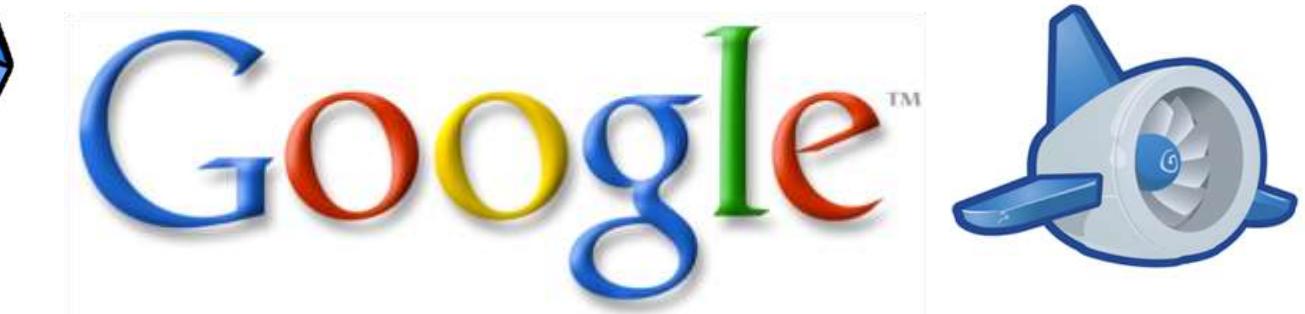
AOL

Sun  
microsystems



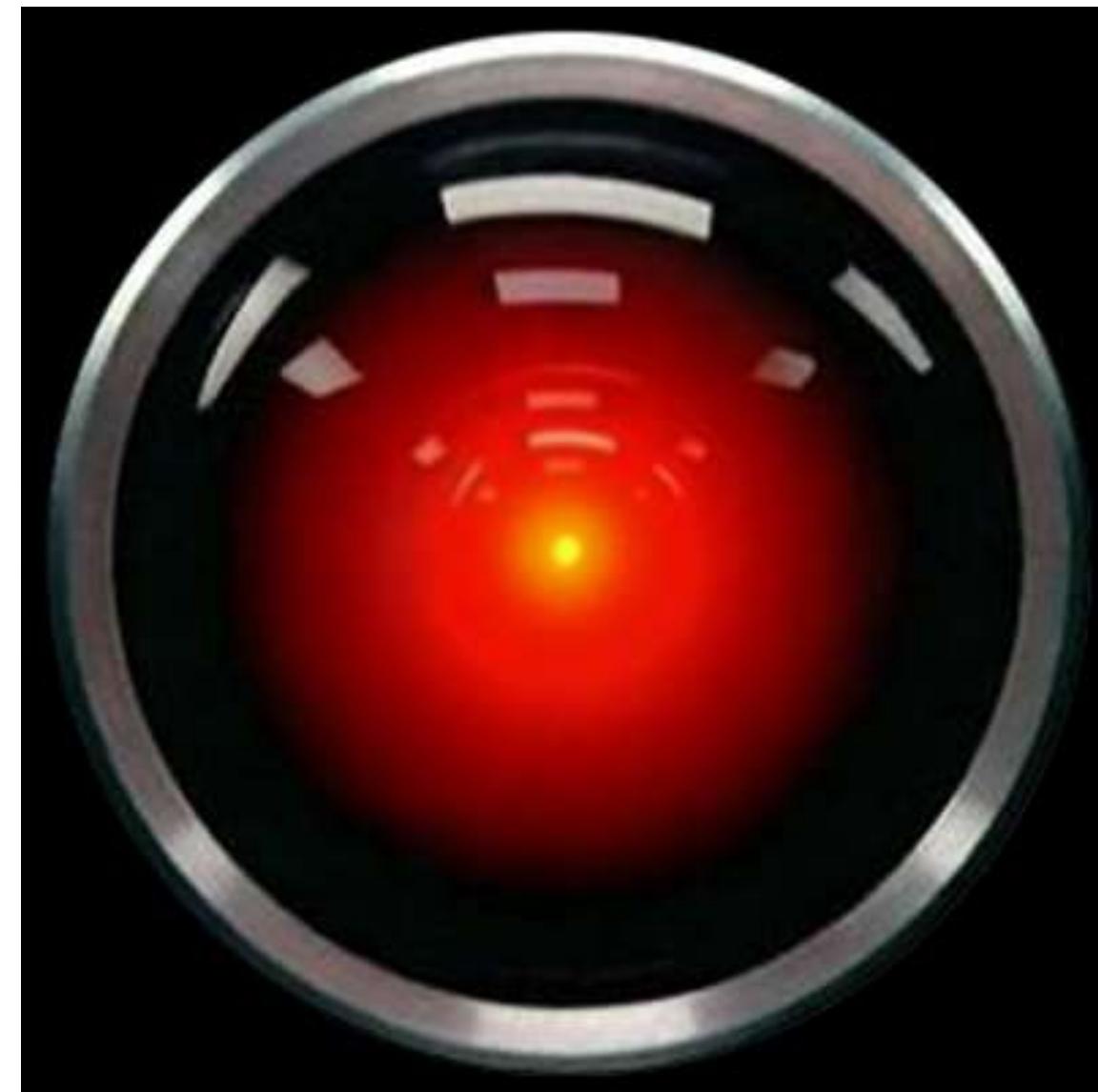
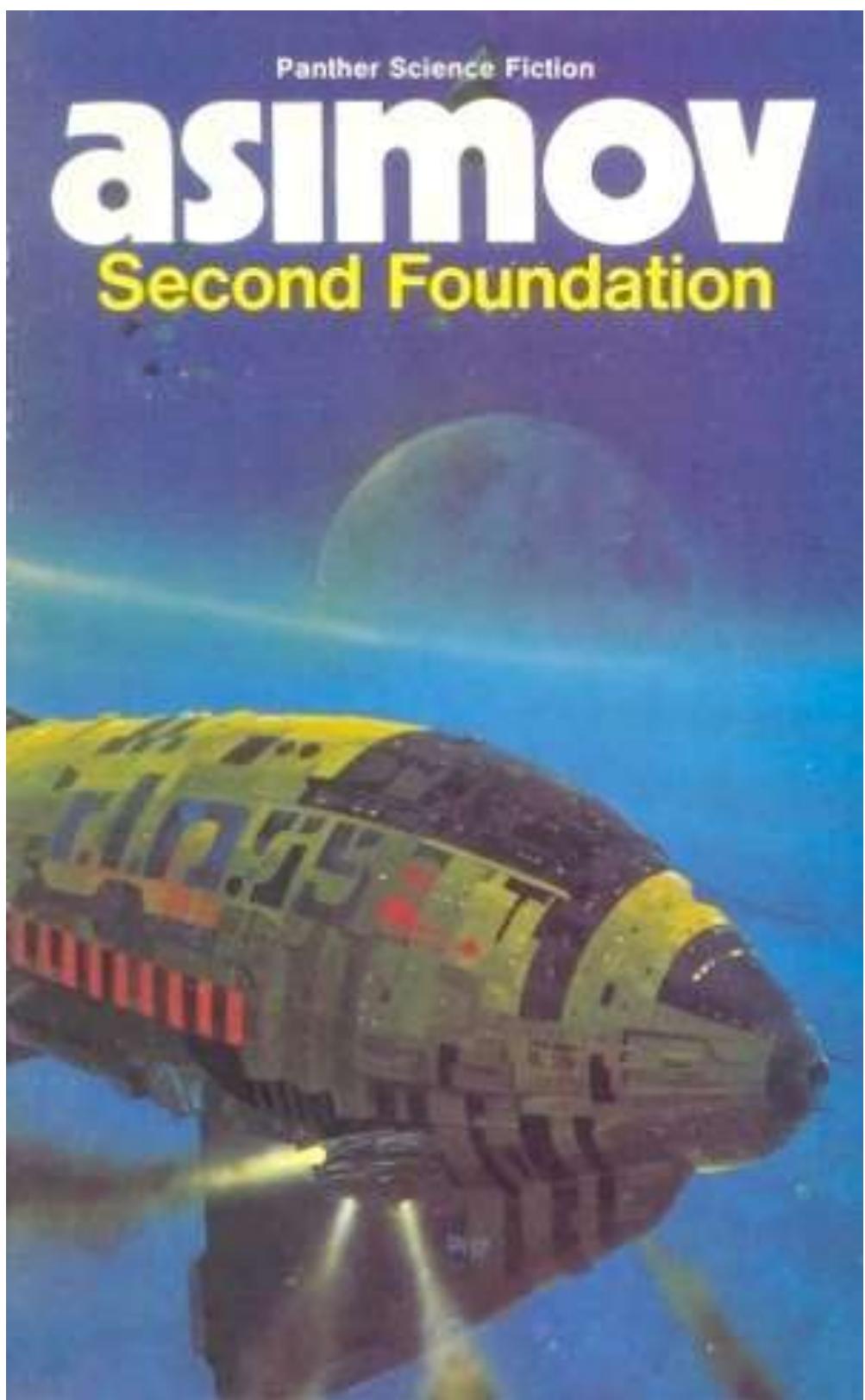
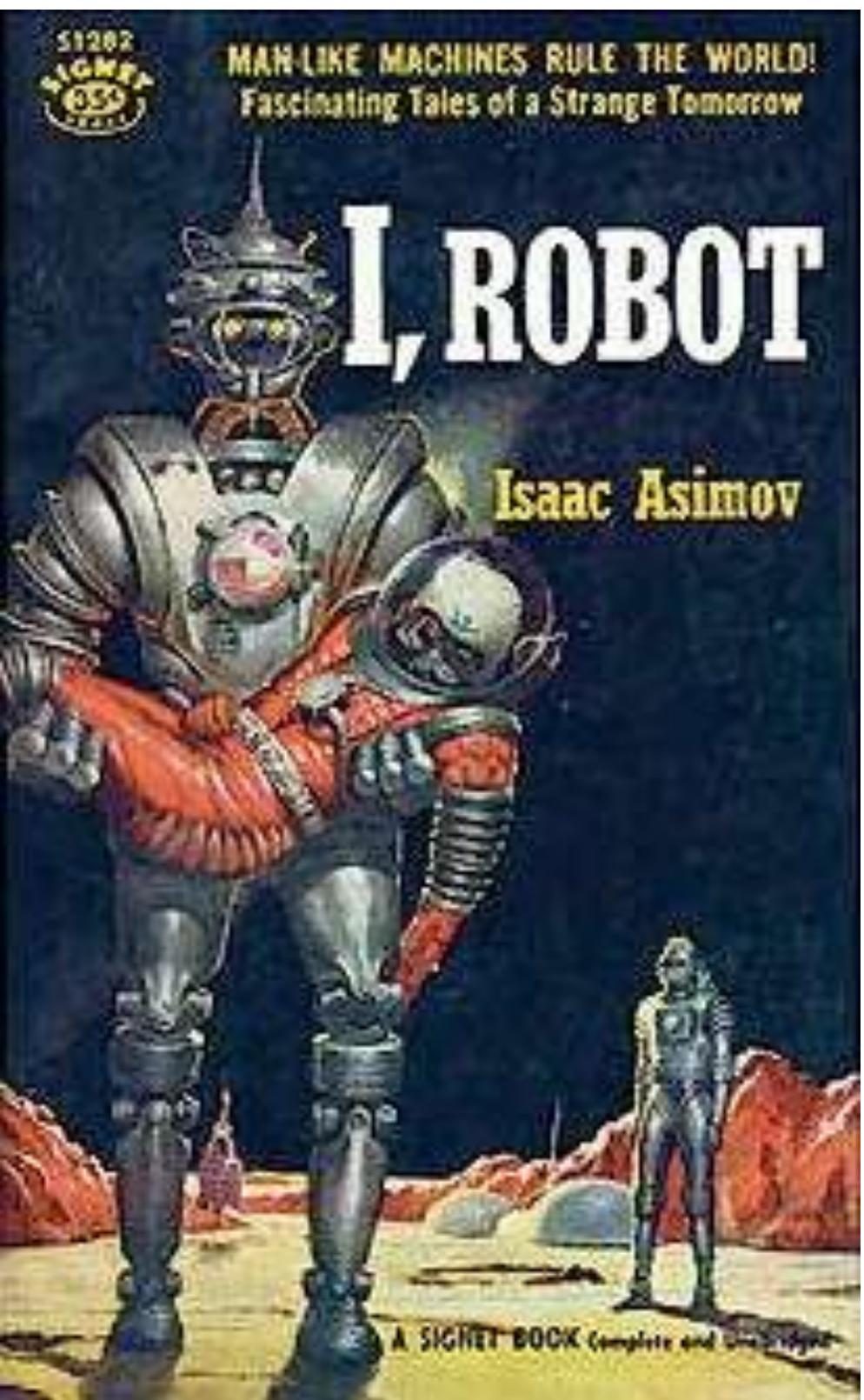
2005-2019

Developer Relations

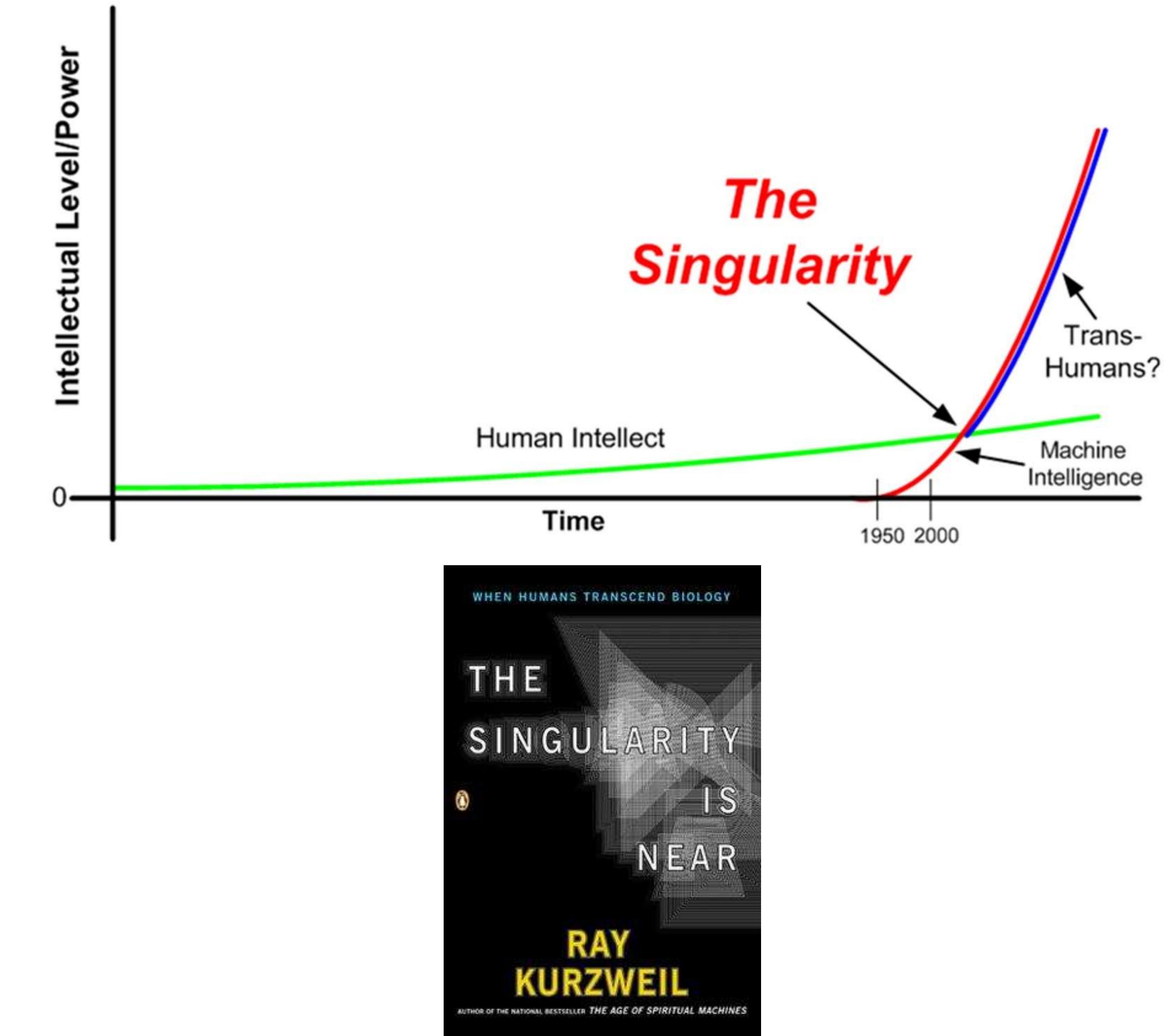
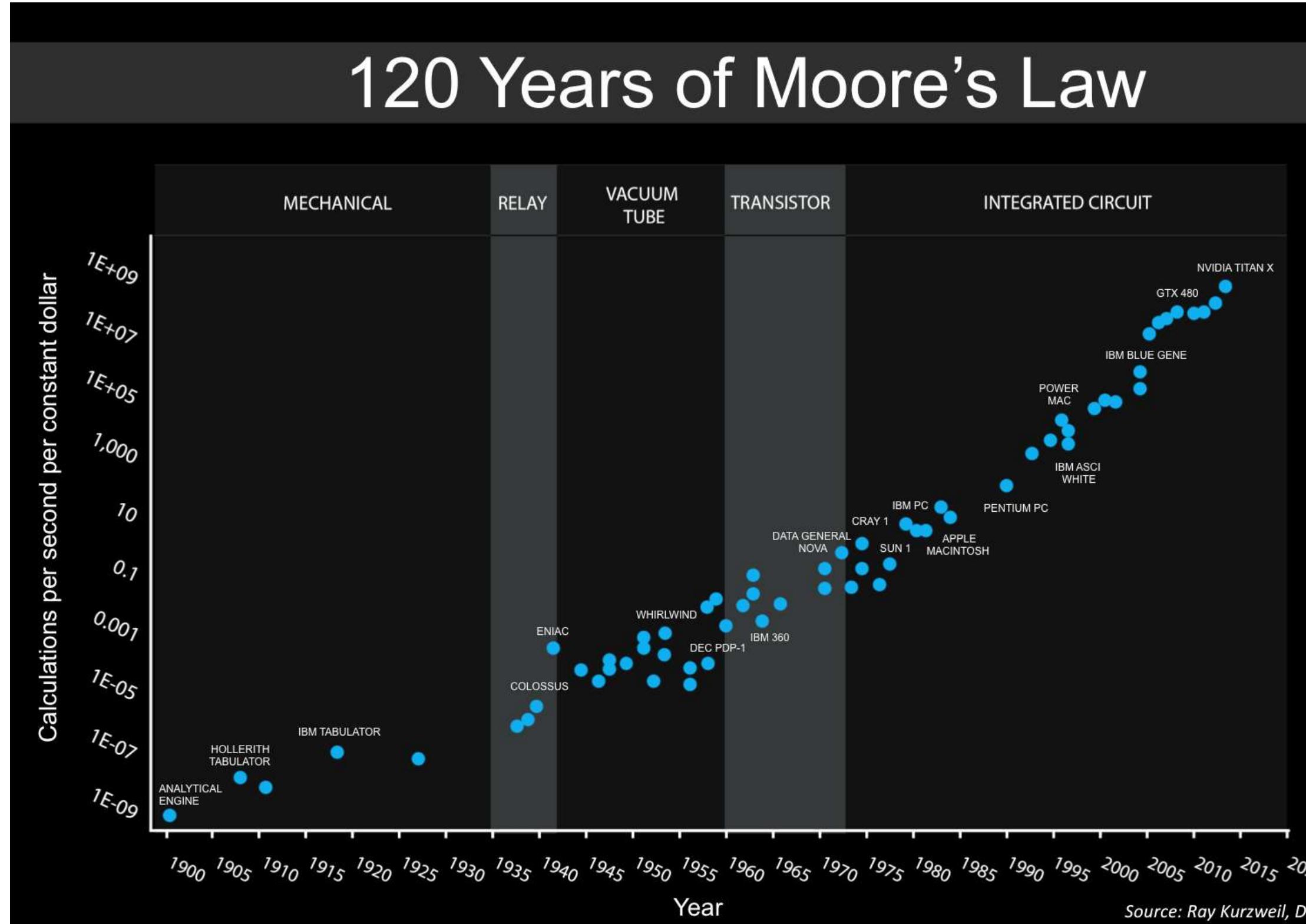




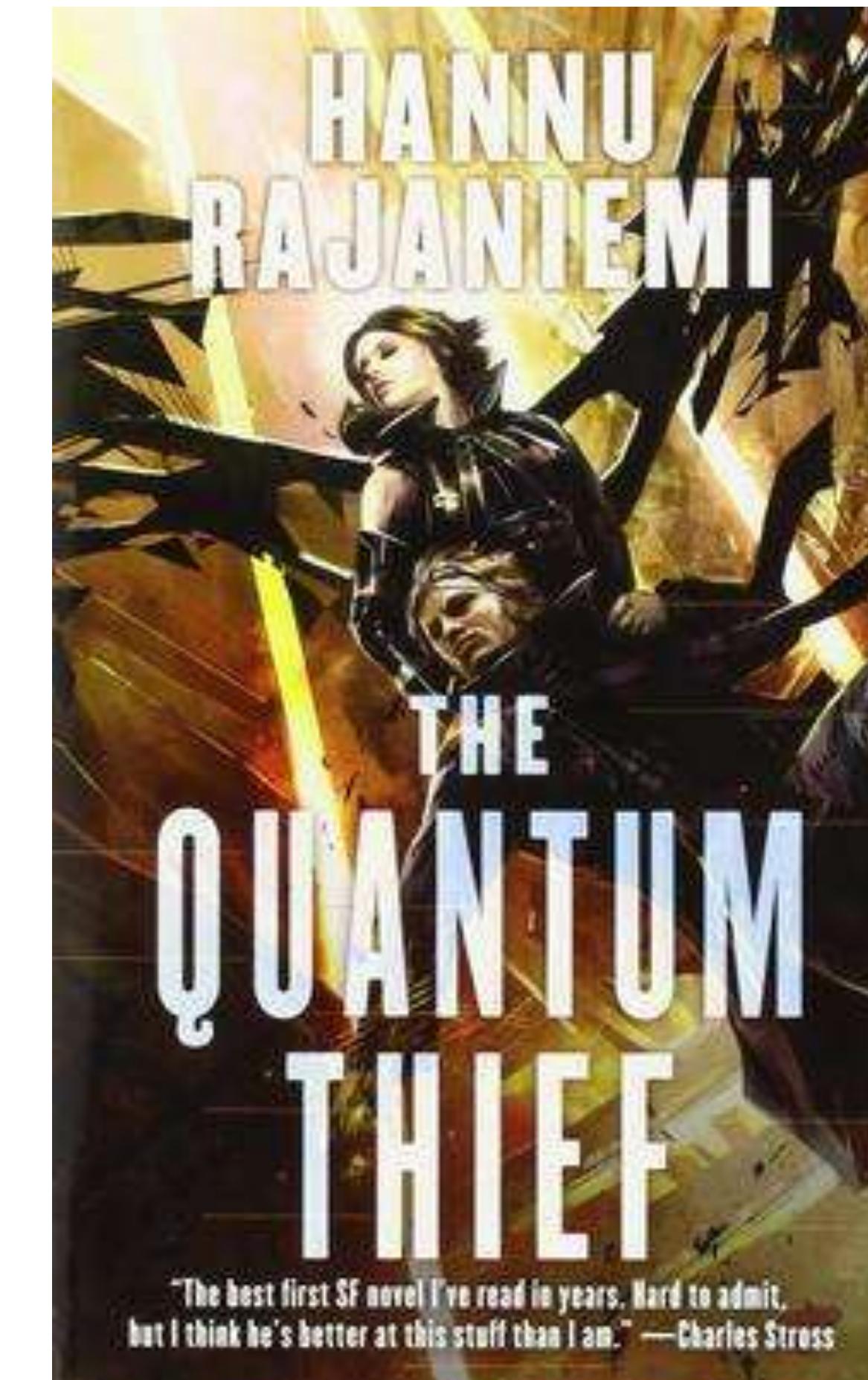
# Dreams of my childhood



# The Singularity... in a galaxy far far away



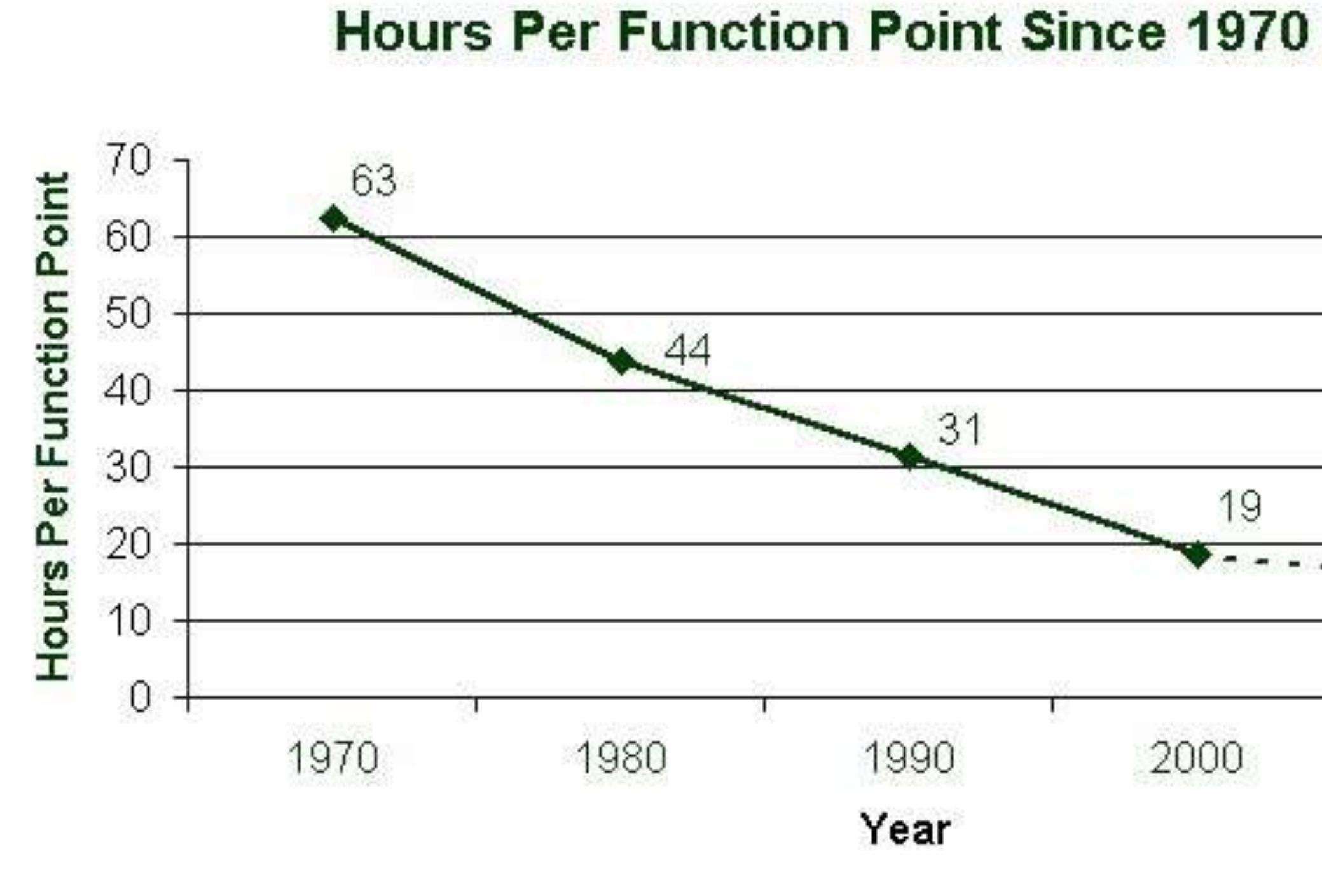
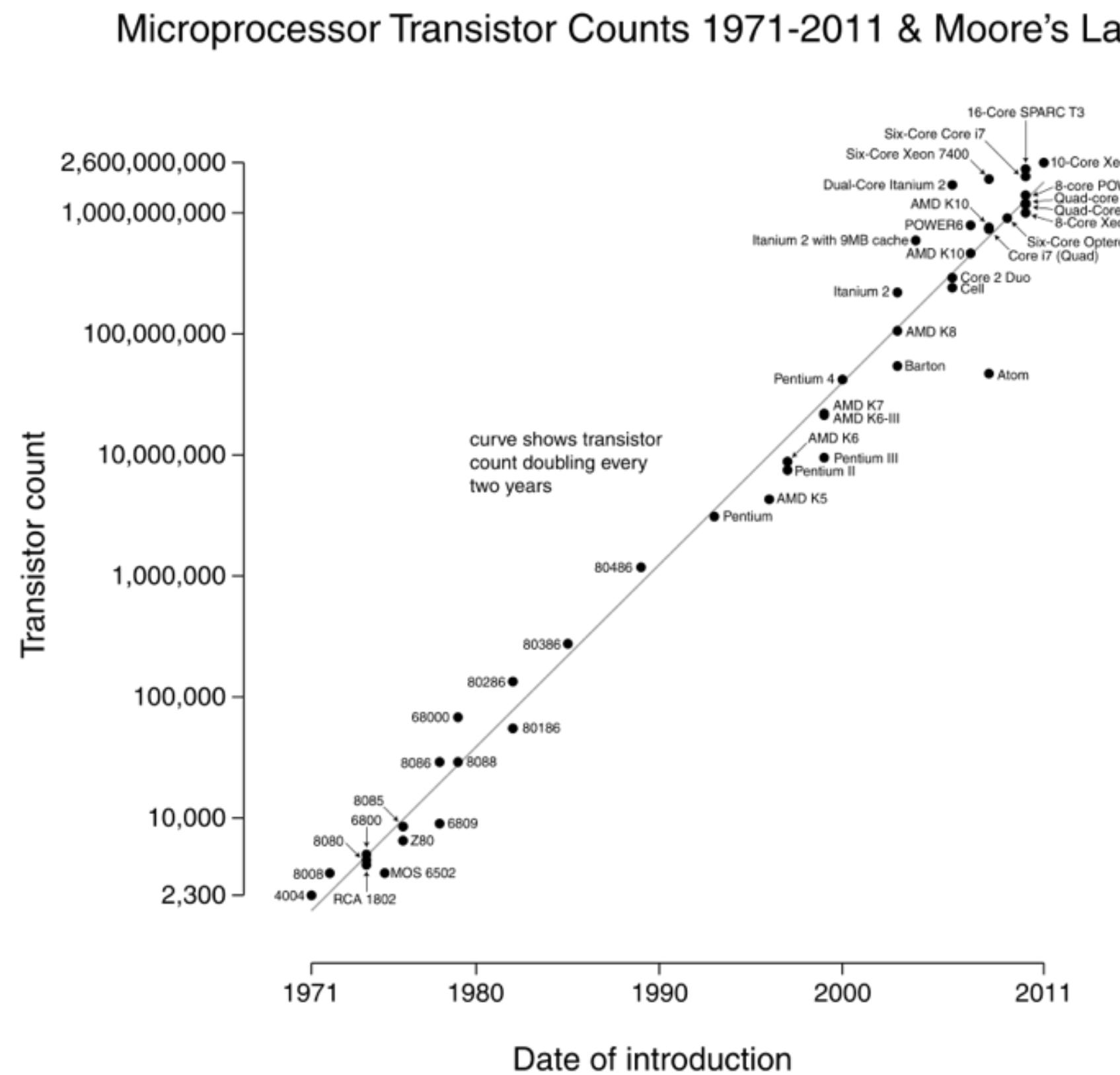
# Great Science Fiction



# Instead, here and now!



# Moore's Law... hardware only!



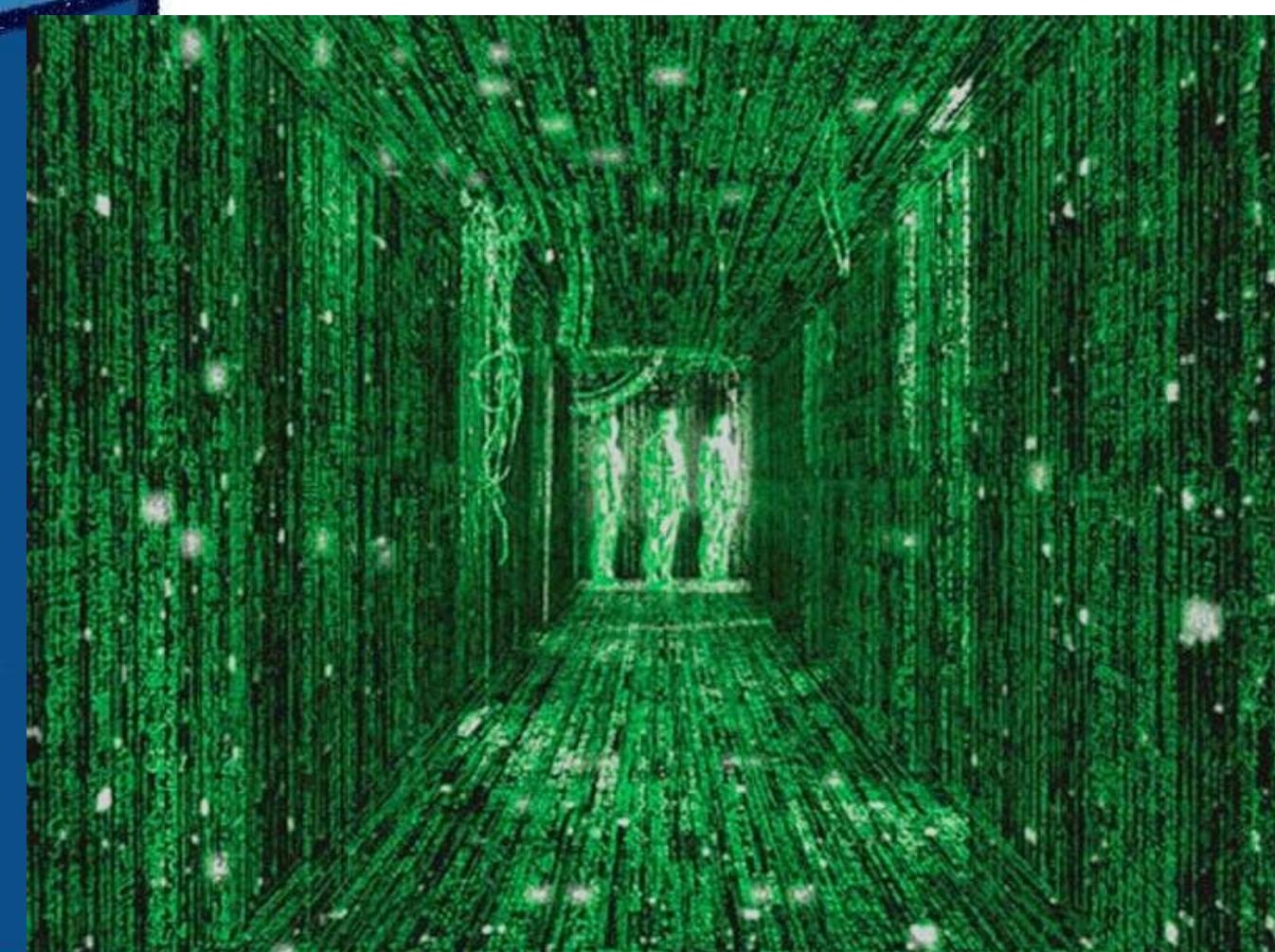
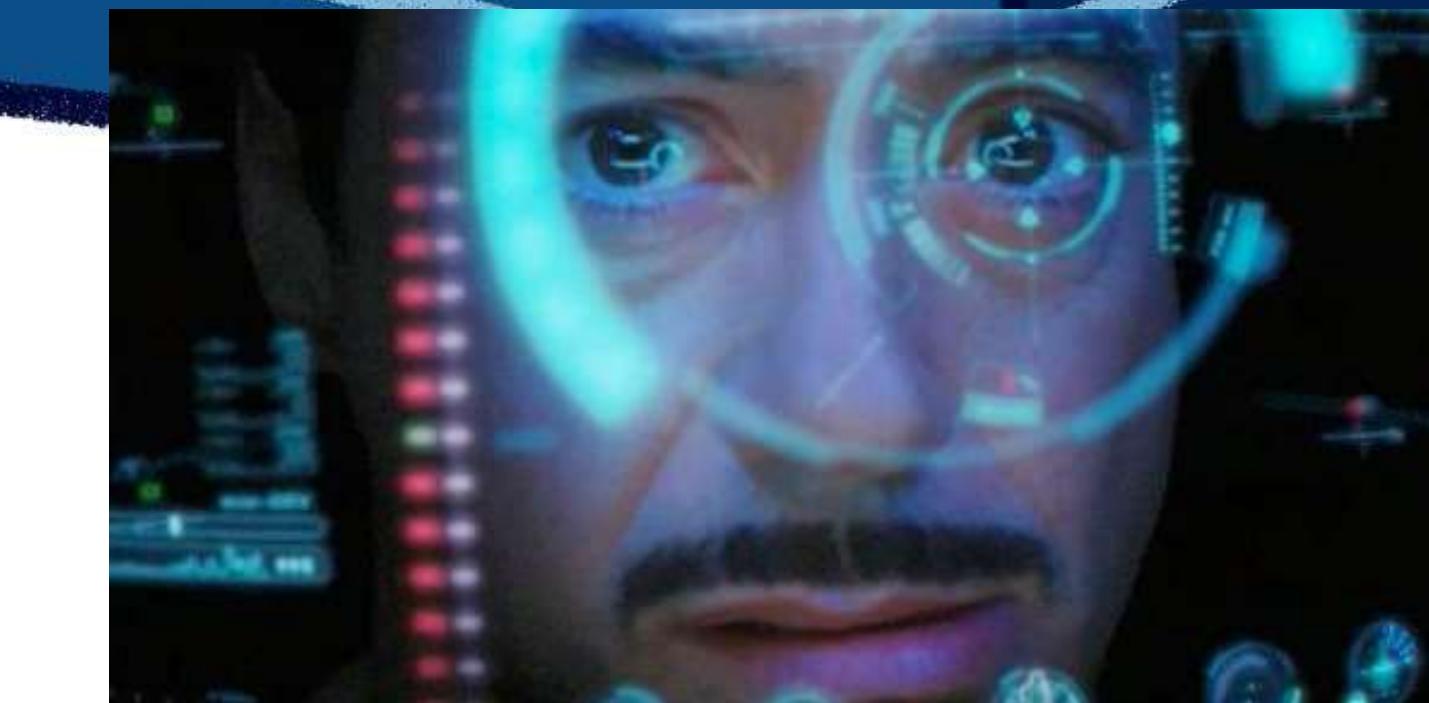


# Developer Productivity

# 3 Abstractions to build cloud apps

- Containers
- Functions, triggered by Events
- Managed Cloud Services

# Containers



# History of containerization

- 1960's mainframe
- 1990's hardware virtualization
- 1990's OS virt precursors: BSD Jails, Solaris zones
- 2006 Cloud IaaS
- 2009 platform virtualization (PaaS)
- 2013 Docker

See @bcantrill's deck <http://www.slideshare.net/bcantrill/docker-and-the-future-of-containers-in-production>

# Isolation using Linux kernel features

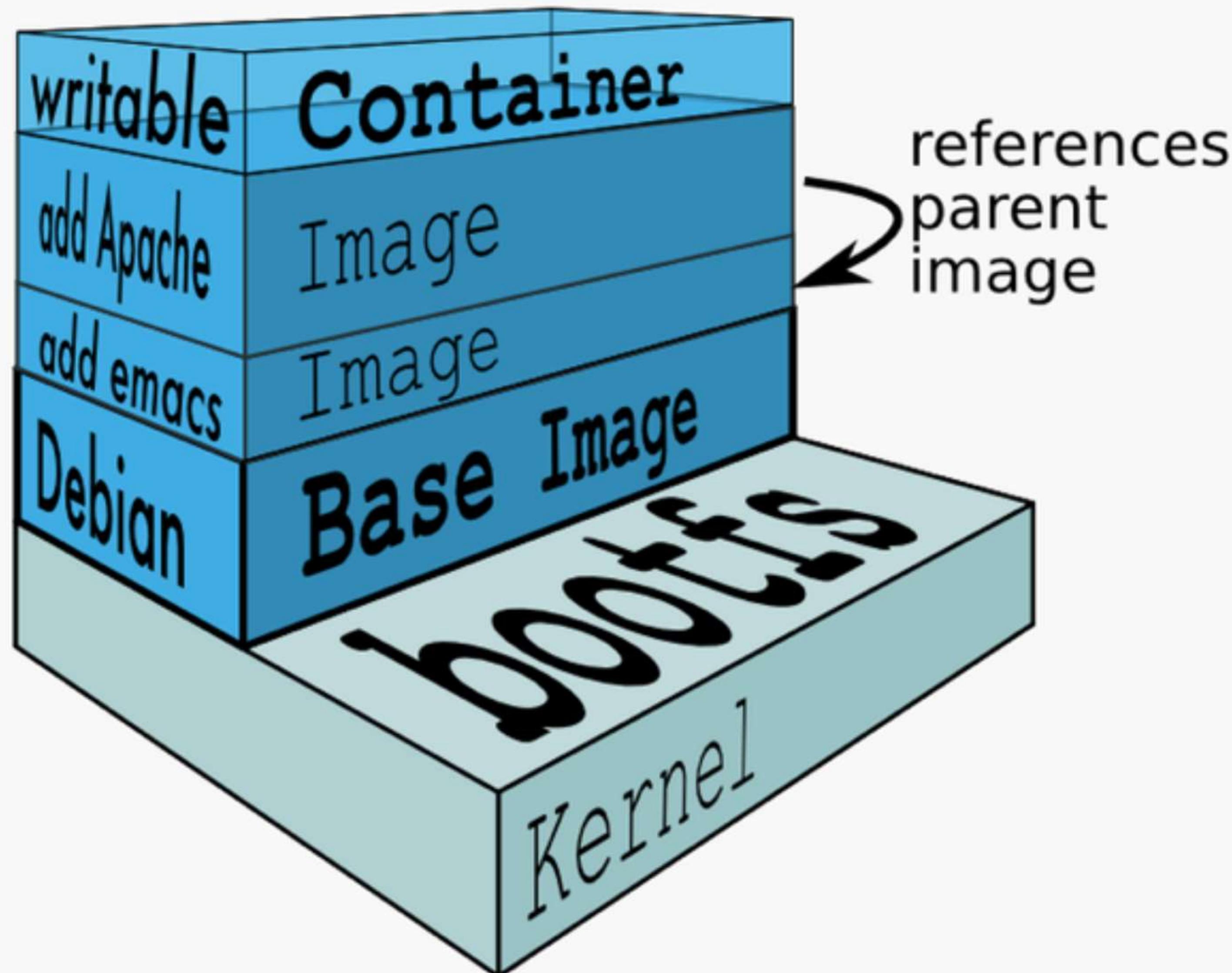
namespaces

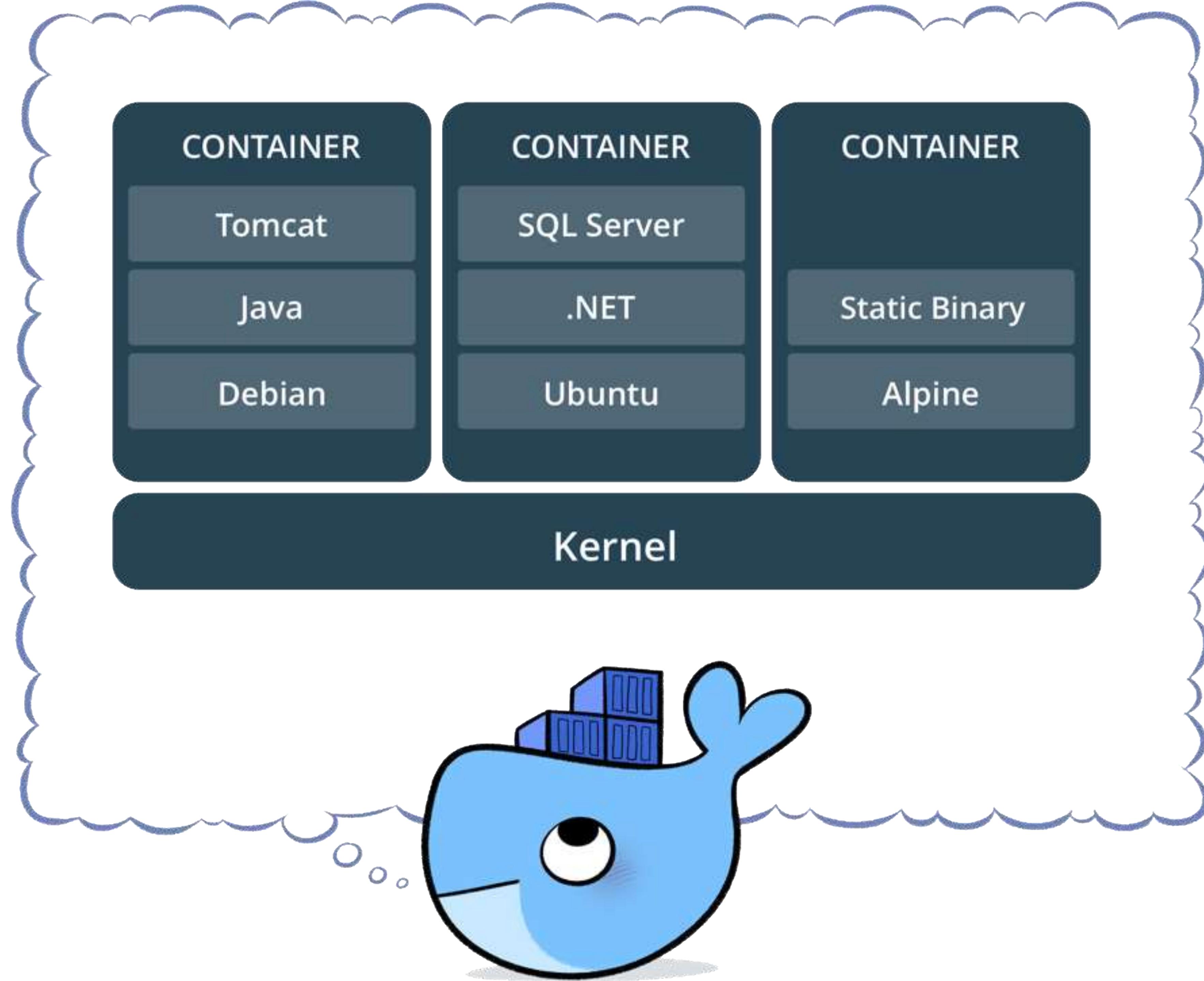
- pid
- mnt
- net
- uts
- ipc
- user

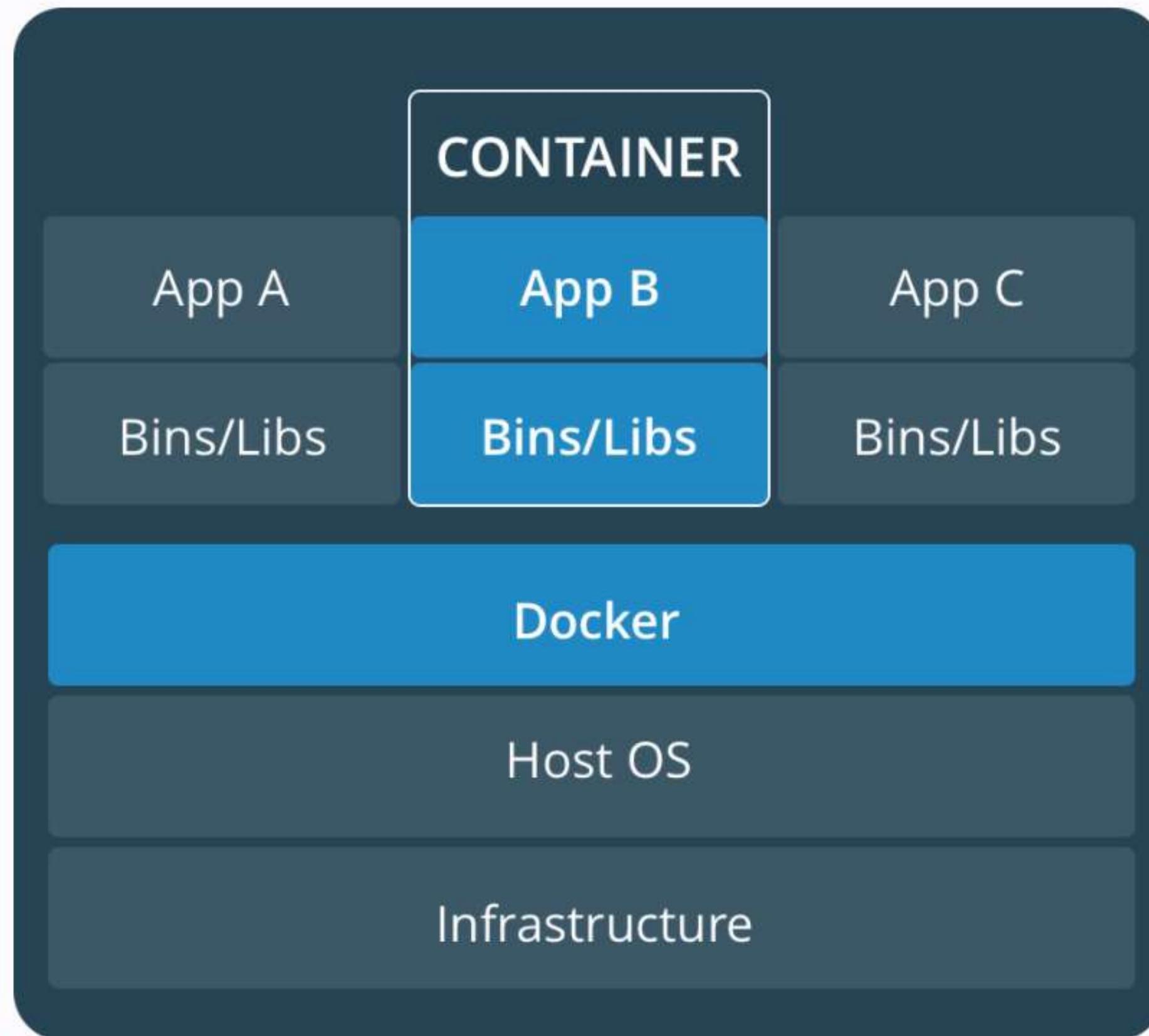
cgroups

- memory
- cpu
- blkio
- devices

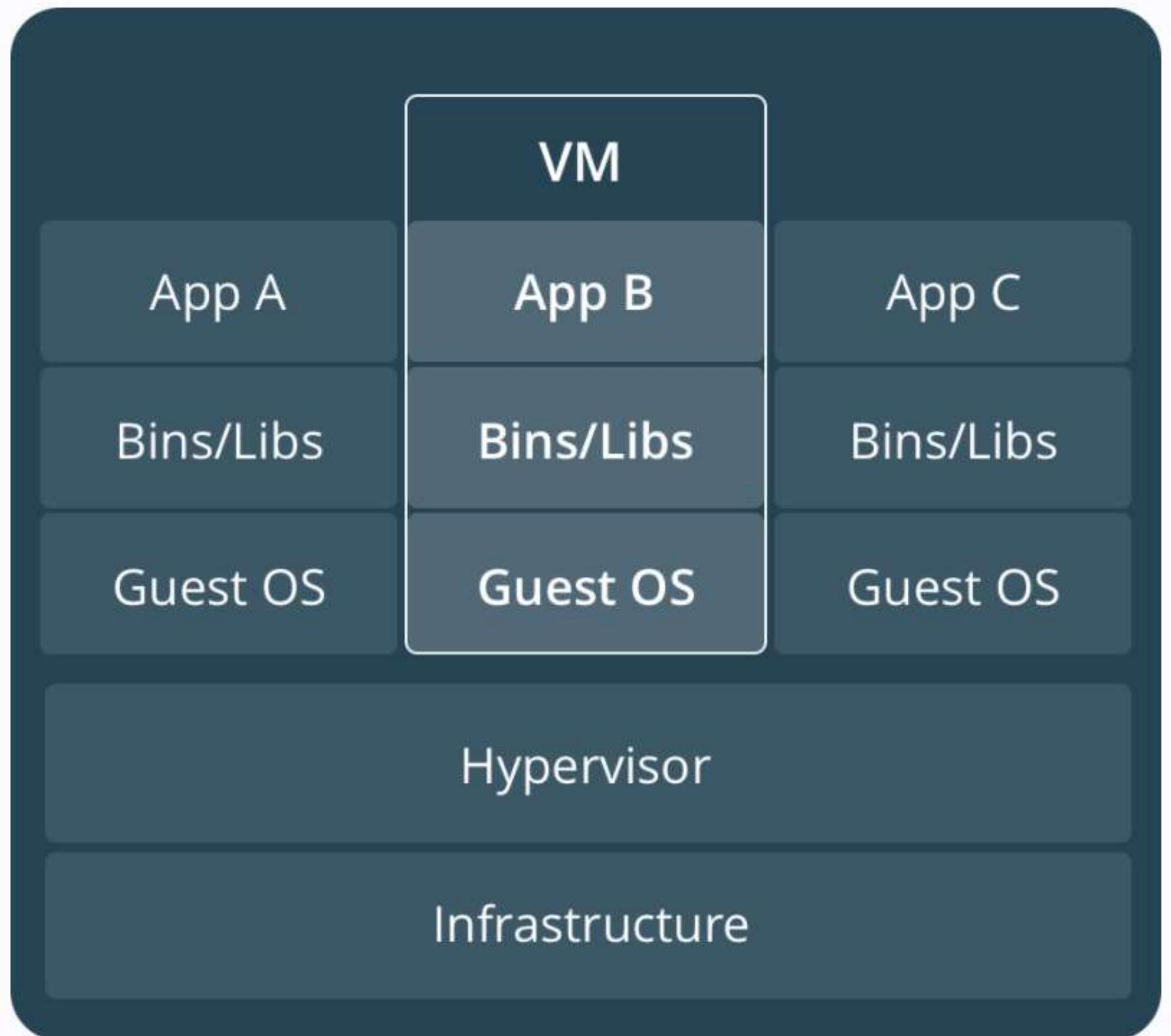
# Union File Systems & Image Layers







CONTAINERS



VIRTUAL MACHINES

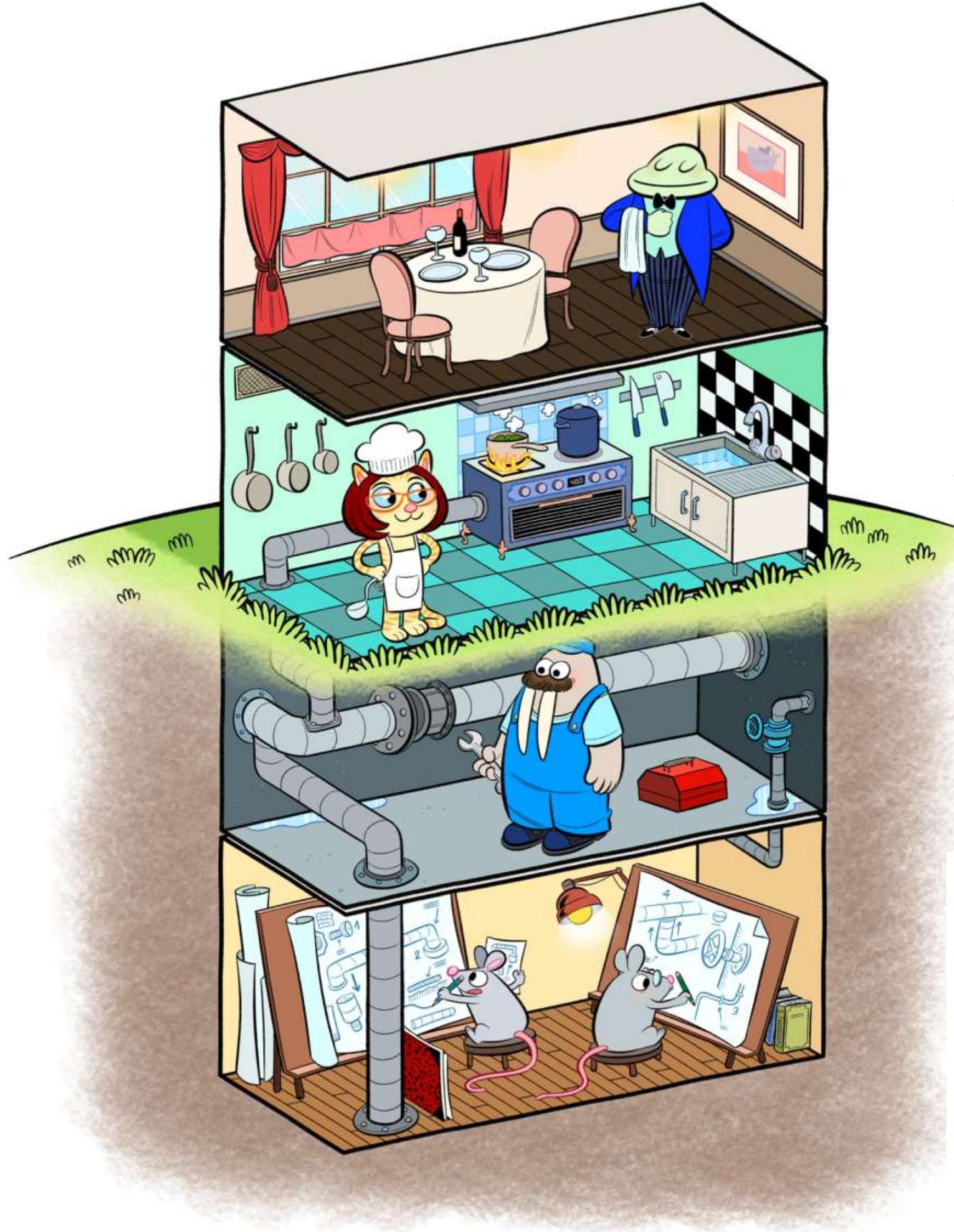
# Docker is building a stack to program the Internet

A commercial product,

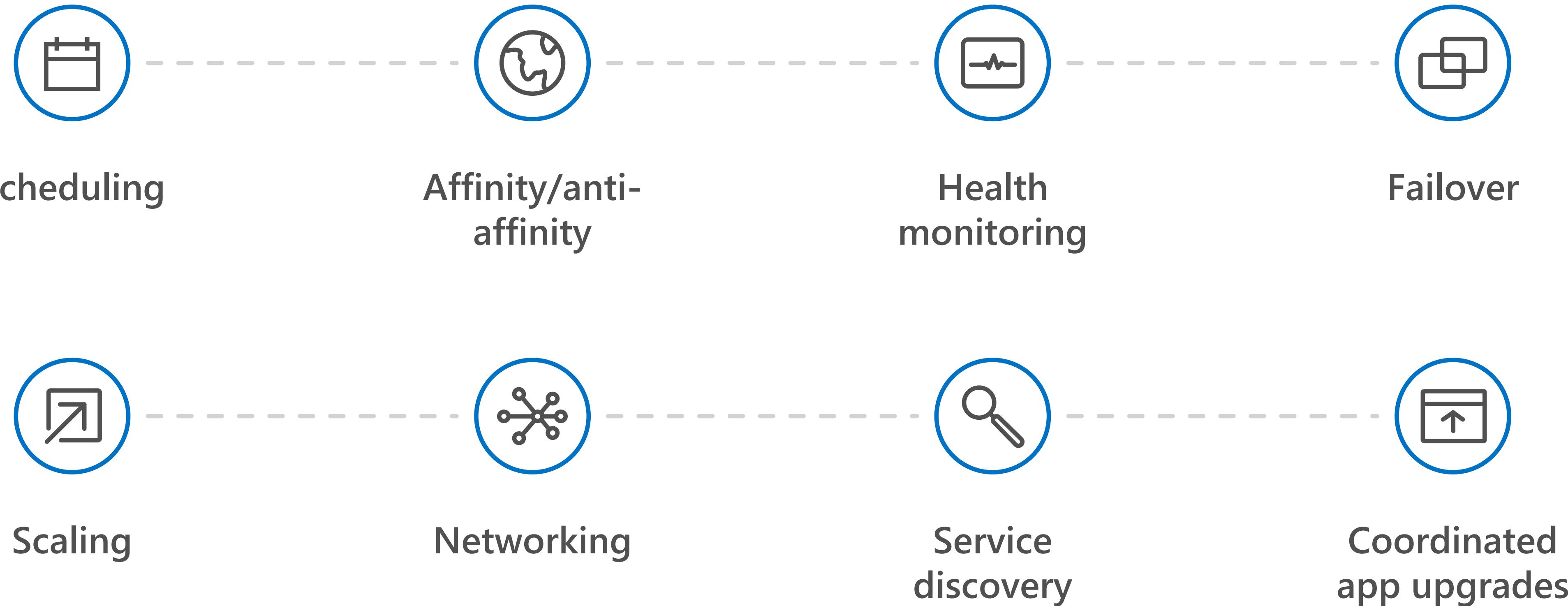
built on  
a development platform,

built on  
infrastructure,

built on  
standards.



# The elements of **orchestration**



# Kubernetes: empowering you to do more

## Portable

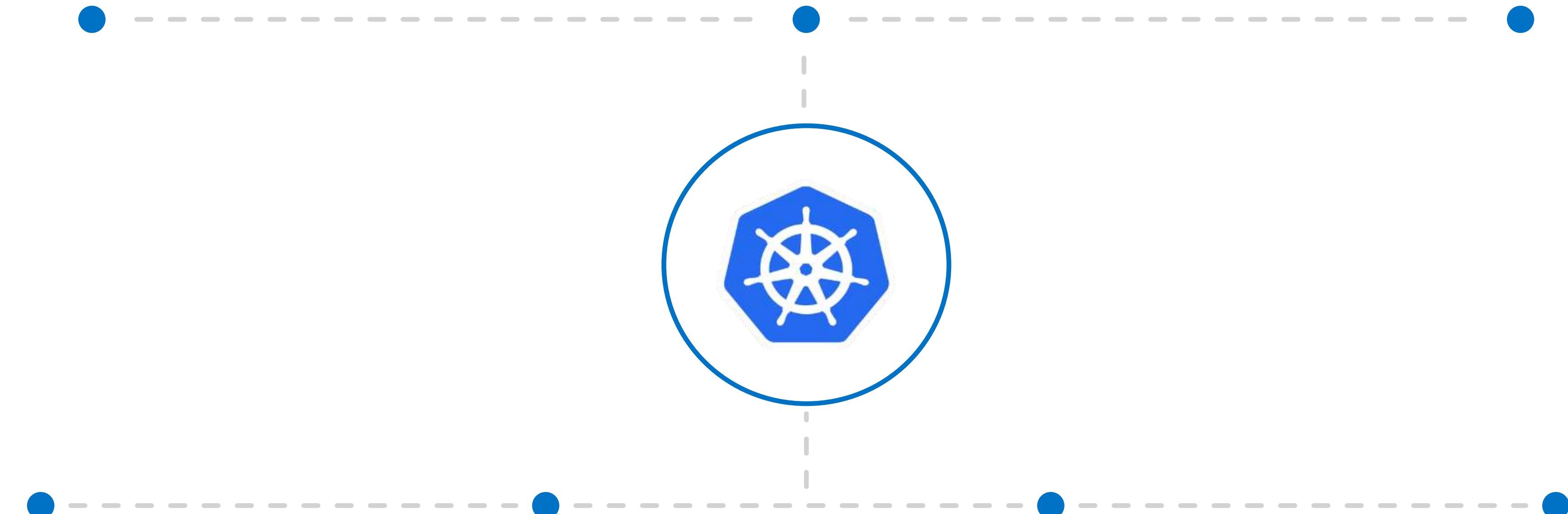
Public, private, hybrid,  
multi-cloud

## Extensible

Modular, pluggable,  
hookable, composable

## Self-healing

Auto-placement, auto-restart,  
auto-replication, auto-scaling



Deploy your  
applications quickly  
and predictably

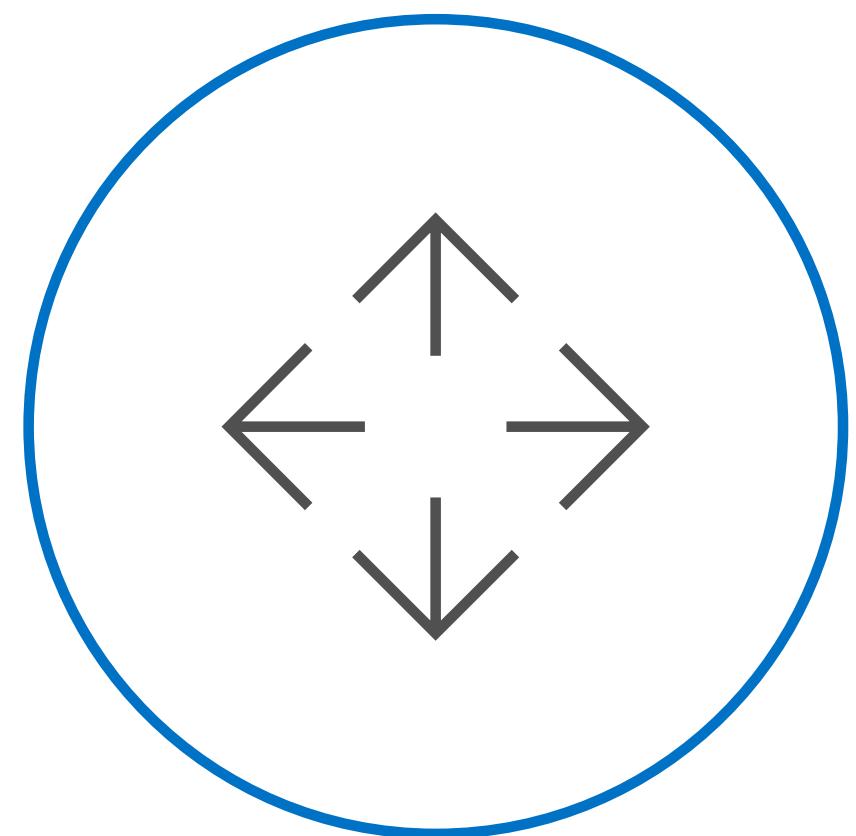
Scale your  
applications on  
the fly

Roll out  
new features  
seamlessly

Limit hardware  
usage to required  
resources only

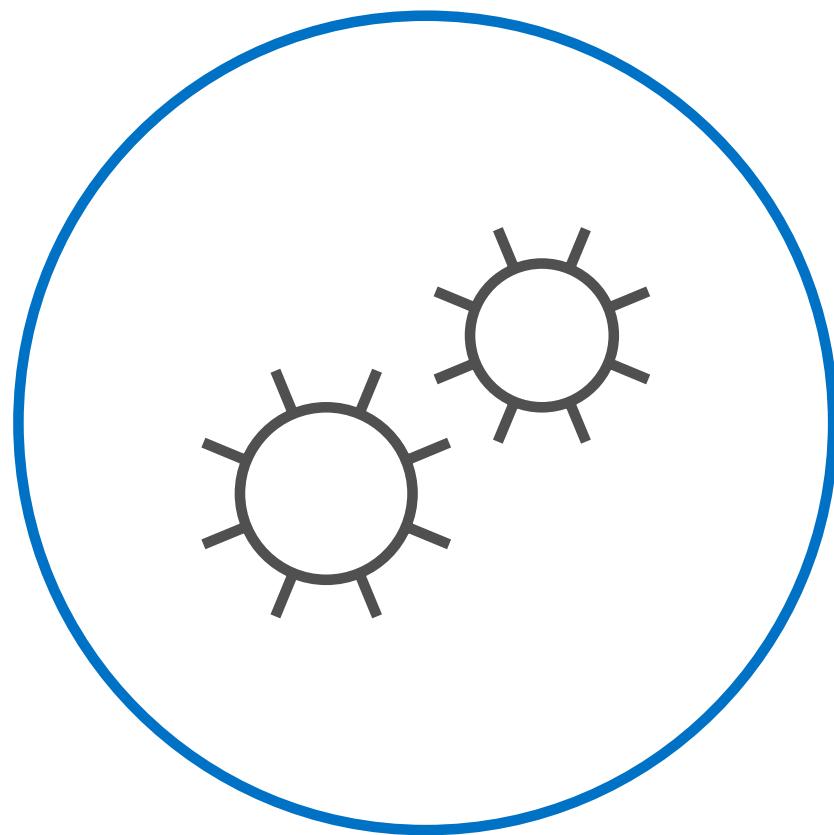
# Containers on Azure

# Containers on Azure



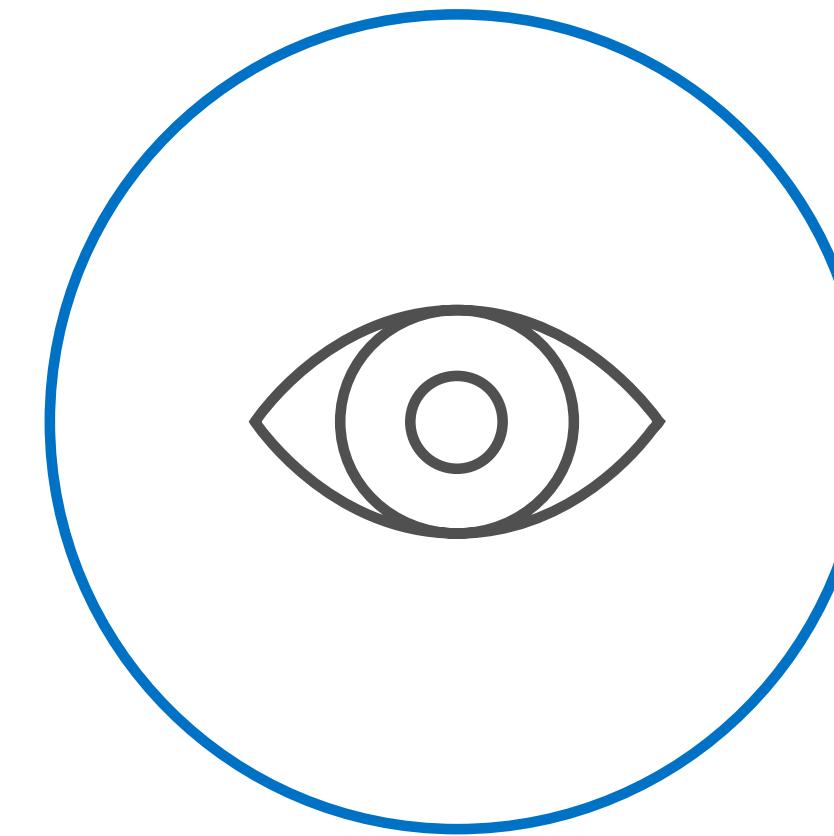
## Flexibility

Deploy containerized applications in your preferred environment



## Productivity

Accelerate containerized application development



## Trust

Manage, monitor, and help secure your containers

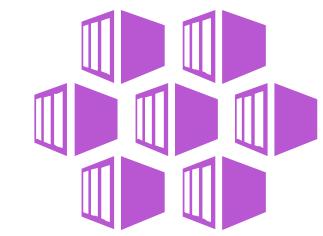
# Containers in Azure



App Service



Service Fabric



Kubernetes Service



Container Instance



Ecosystem

Deploy web apps or APIs using containers in a PaaS environment

Modernize .NET applications to microservices using Windows Server containers

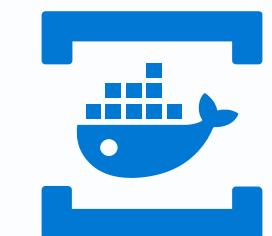
Scale and orchestrate Linux containers using Kubernetes

Elastically burst from your Azure Kubernetes Service (AKS) cluster

Bring your Partner solutions that run great on Azure



Azure Container Registry



Docker Hub

----- Choice of developer tools and clients -----

If you have a preferred container platform

Pivotal Cloud Foundry  Kubernetes  Docker Enterprise Edition

Red Hat OpenShift  Mesosphere DC/OS



Pivotal Cloud Foundry



Self-managed  
Kubernetes



Docker Enterprise  
Edition

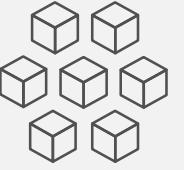


Red Hat OpenShift



Mesosphere DC/OS

You could bring that platform to Azure



Azure Kubernetes  
Service (AKS)



App Service



Azure Container  
Instances (ACI)



Service Fabric



Azure Batch



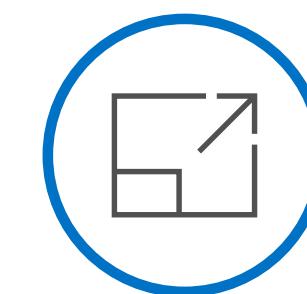
Azure Container  
Registry (ACR)

# Azure Kubernetes Service (AKS)

Simplify the deployment, management, and operations of Kubernetes



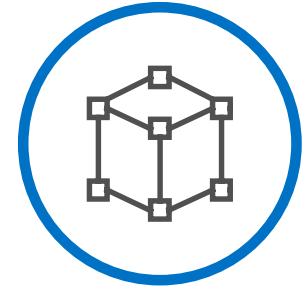
**Deploy and manage  
Kubernetes with  
ease**



**Scale and run  
applications with  
confidence**



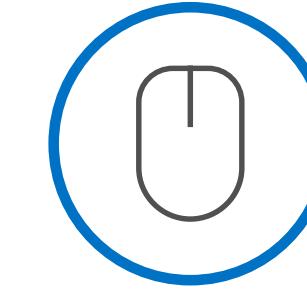
**Secure your Kubernetes  
environment**



**Accelerate  
containerized application  
development**

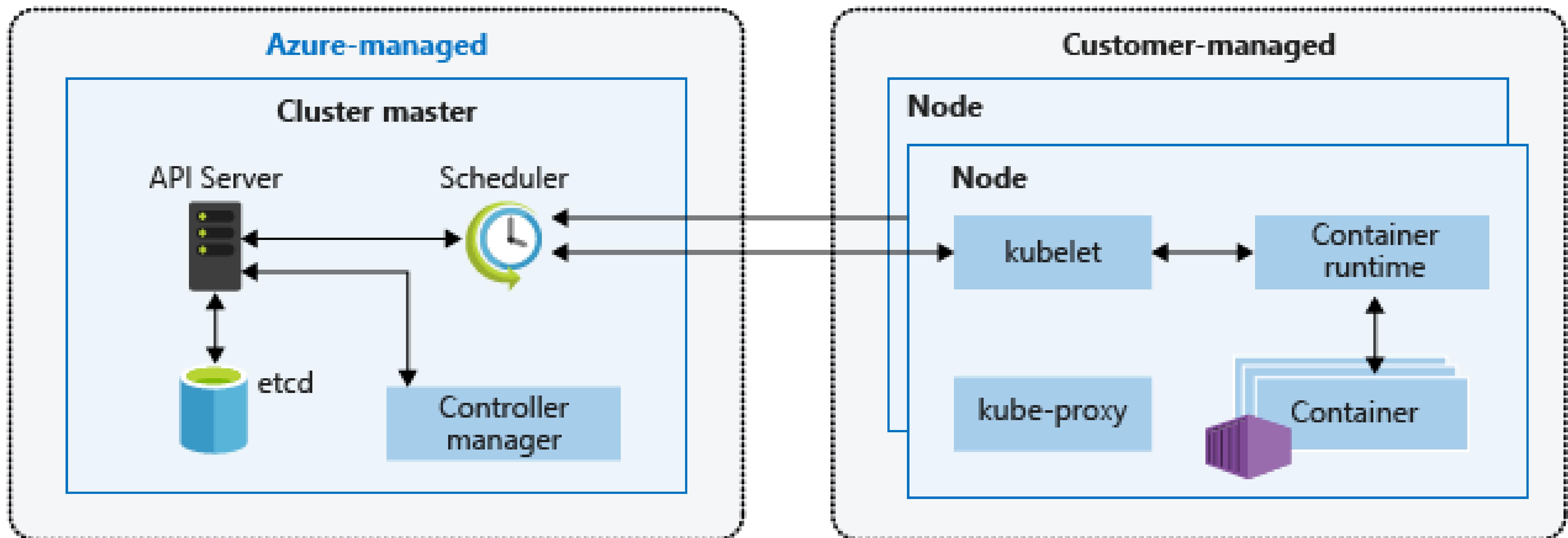


**Work how you want  
with open-source tools  
& APIs**



**Set up  
CI/CD in a  
few clicks**

# How managed Azure Kubernetes Service works



# Azure Kubernetes Service (AKS)

Get started easily

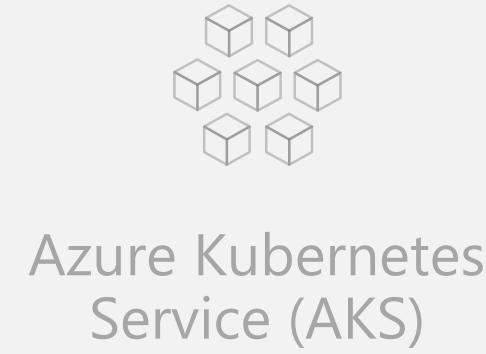
```
$ az aks create -g myResourceGroup -n myCluster --generate-ssh-keys  
\ Running ..
```

```
$ az aks install-cli  
Downloading client to /usr/local/bin/kubectl ..
```

```
$ az aks get-credentials -g myResourceGroup -n myCluster  
Merged "myCluster" as current context ..
```

```
$ kubectl get nodes
```

NAME	STATUS	AGE	VERSION
aks-nodepool1-29486530-0	Ready	4m	v1.11.9
aks-nodepool1-29486530-1	Ready	4m	v1.11.9
aks-nodepool1-29486530-2	Ready	4m	v1.11.9



Azure Kubernetes  
Service (AKS)



App Service



Azure Container  
Instances (ACI)



Service Fabric



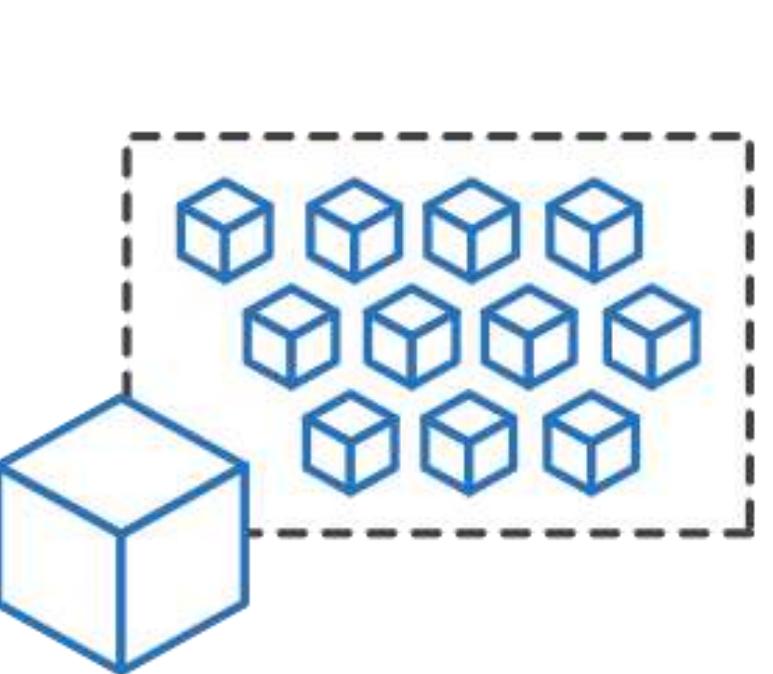
Azure Batch



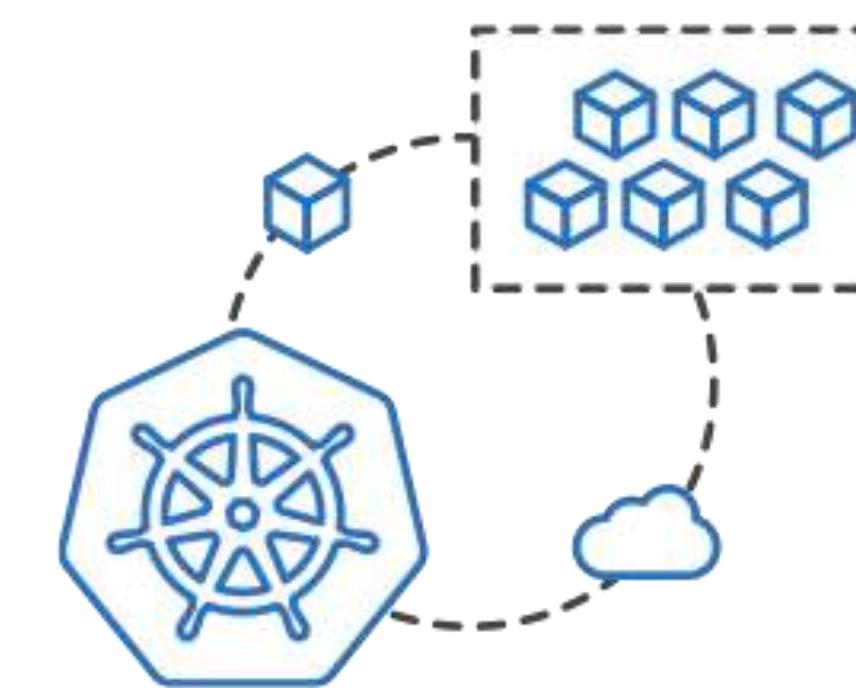
Azure Container  
Registry (ACR)

# Azure Container Instances (ACI)

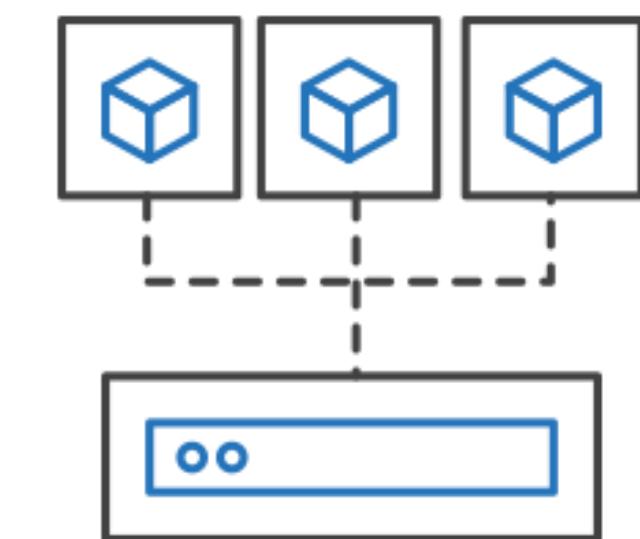
Easily run containers on Azure without managing servers



Run containers  
without managing  
servers



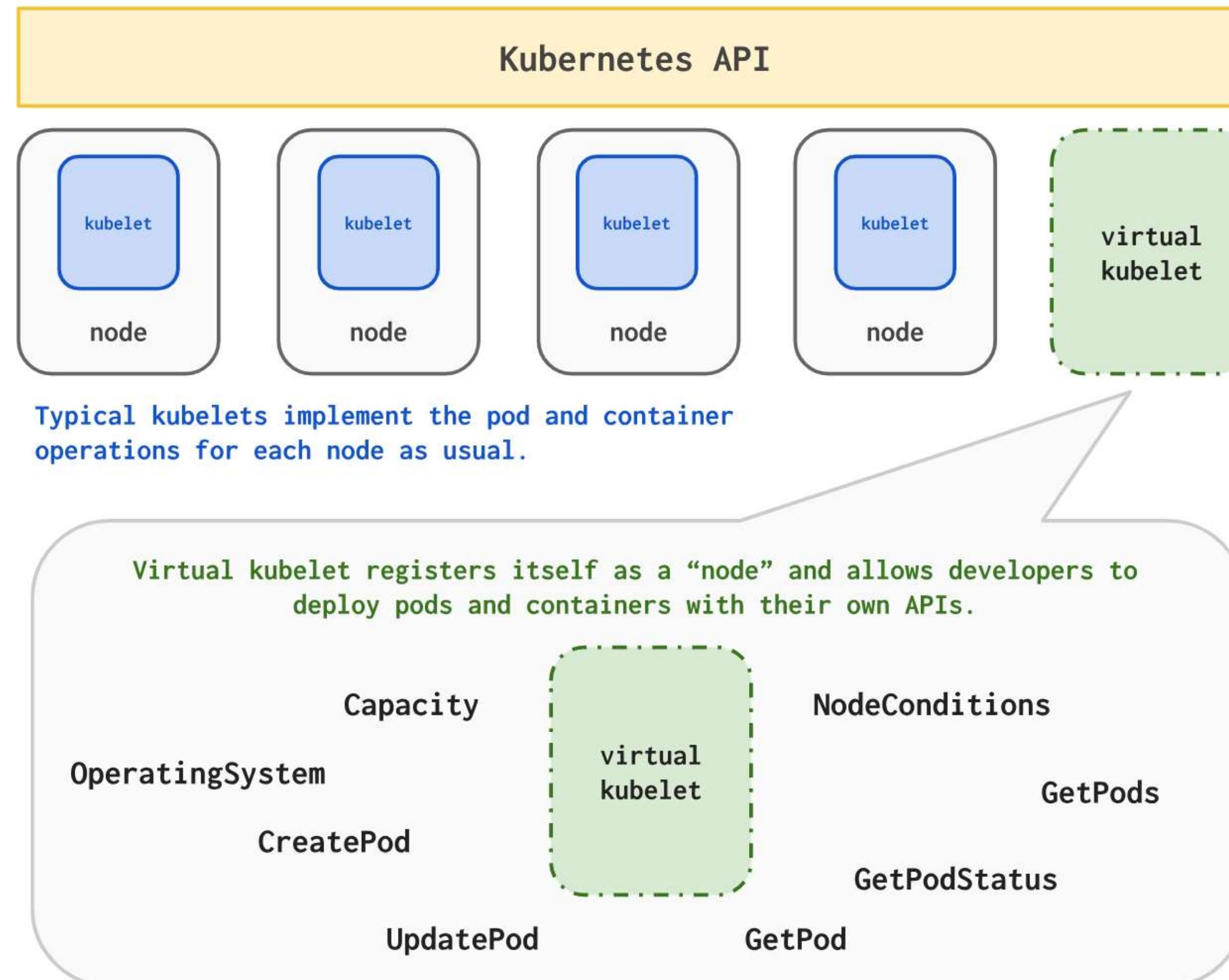
Increase agility with  
containers on  
demand



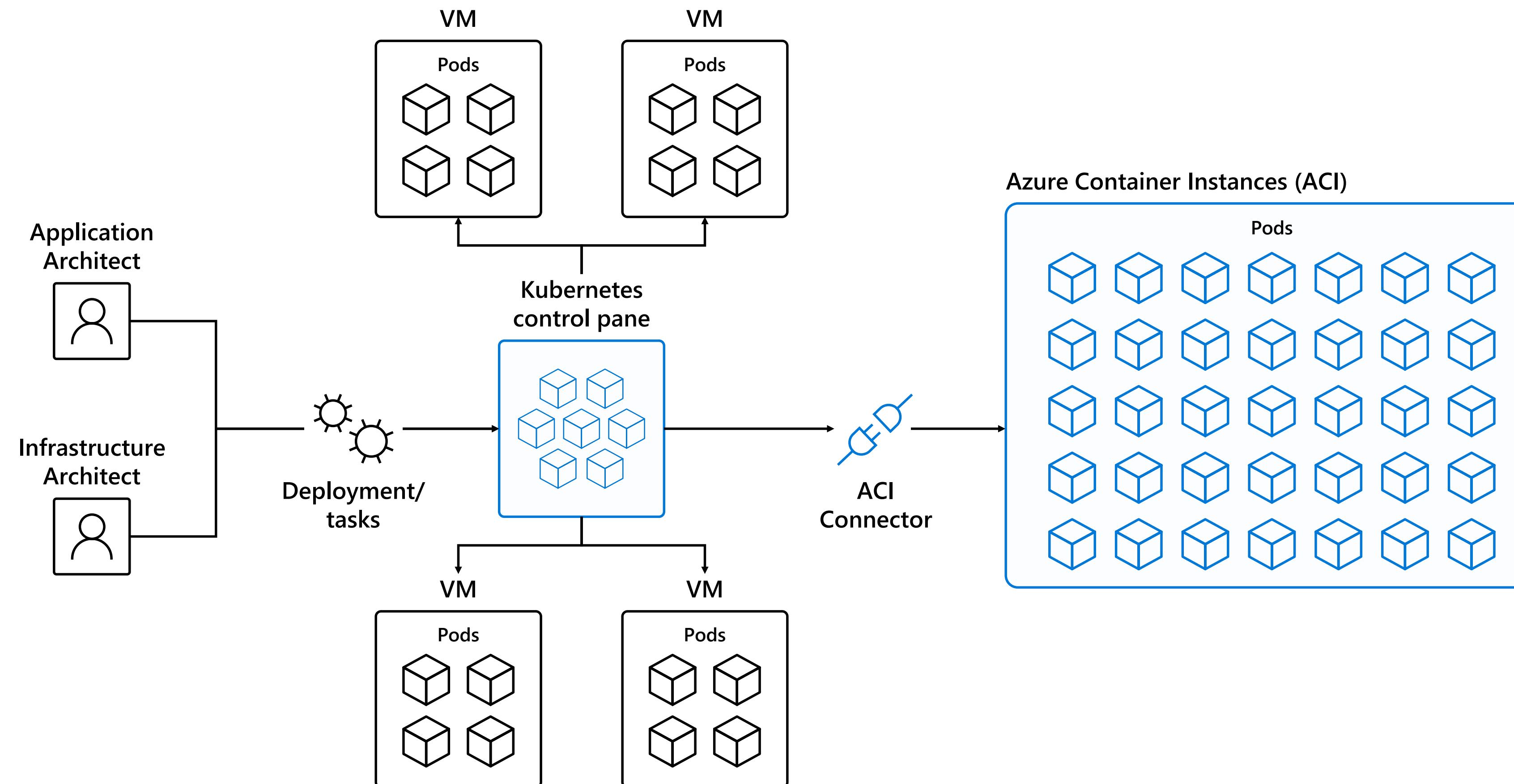
Secure applications  
with hypervisor  
isolation

# Virtual Kubelet

# Virtual Kubelet



# Bursting with the ACI Connector/ Virtual Kubelet



# Virtual Nodes

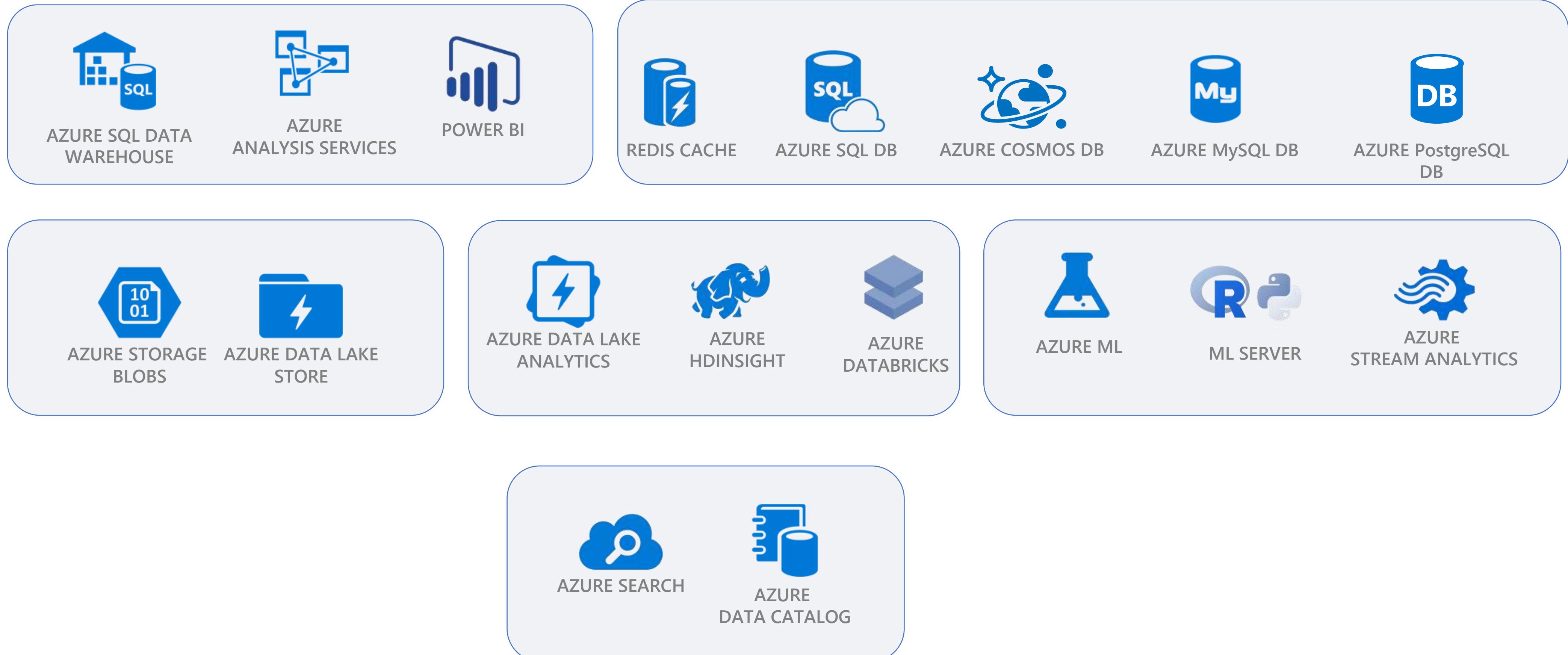
Public preview

- Seamlessly connect AKS to extra burst capacity within ACI
- Little to no operational overhead
- Eliminates the need to over provision clusters
- Service discovery & bring your own virtual network
- We develop in the open through Virtual Kubelet



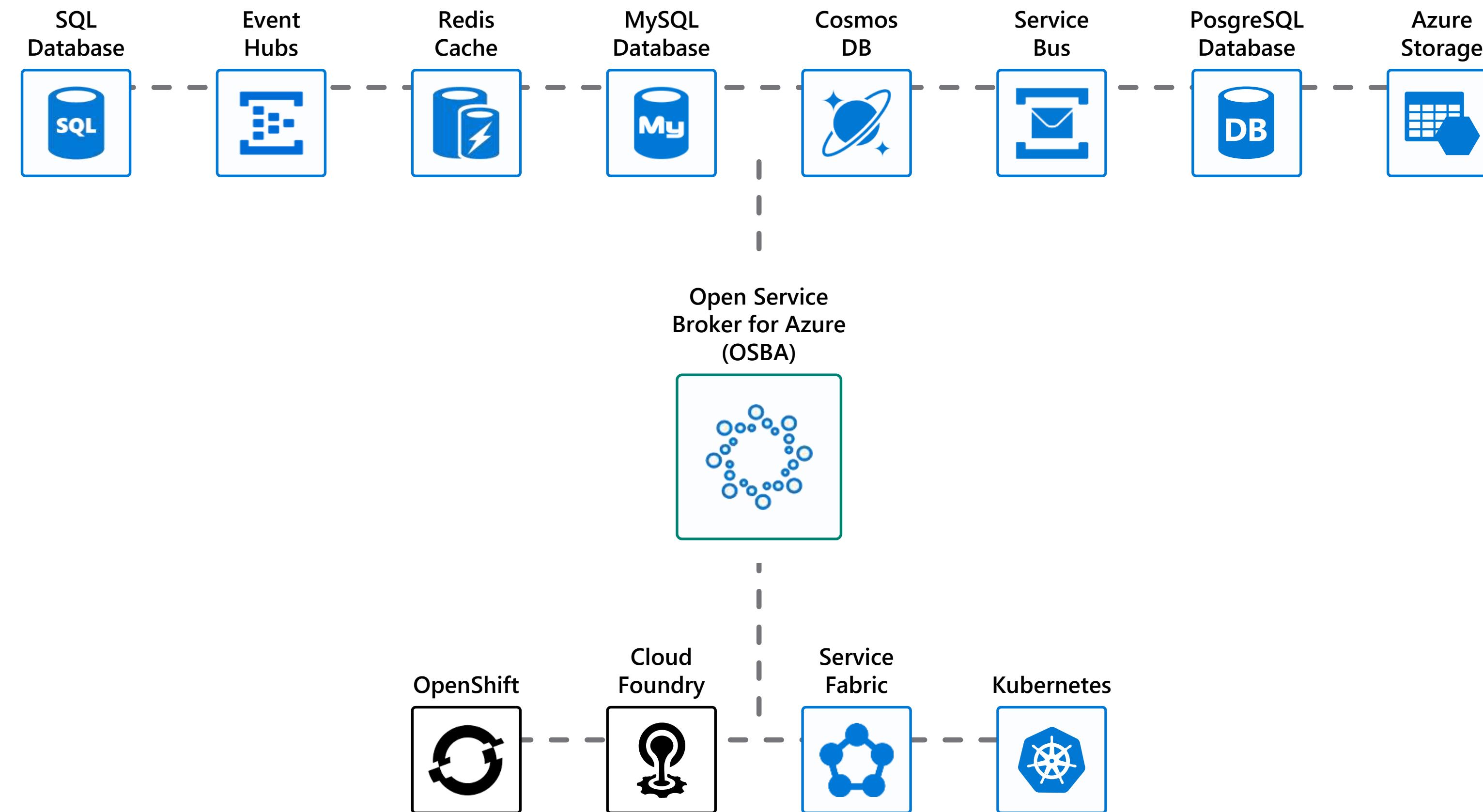
# Managed Cloud Services

# Use Azure Managed Data Platform Services



# Open Service Broker for Azure (OSBA)

Easily access to SLA-backed Azure Services such as Azure Database for MySQL



# Serverless



# What is serverless?



## Full abstraction of servers

Developers can just focus on their code—there are no distractions around server management, capacity planning, or availability.



## Instant, event-driven scalability

Application components react to events and triggers in near real-time with virtually unlimited scalability; compute resources are used as needed.

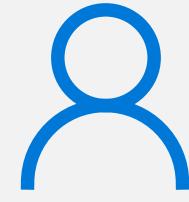


## Pay-per-use

Only pay for what you use: billing is typically calculated on the number of function calls, code execution time, and memory used.\*

# FaaS is at the center of serverless

Functions-as-a-Service programming model use functions to achieve true serverless compute



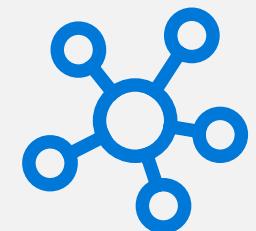
## Single responsibility

Functions are single-purposed, reusable pieces of code that process an input and return a result



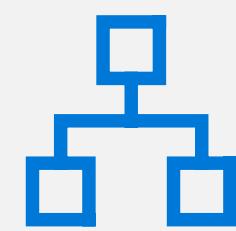
## Short lived

Functions don't stick around when finished executing, freeing up resources for further executions



## Stateless

Functions don't hold any persistent state and don't rely on the state of any other processes



## Event driven & scalable

Functions respond to predefined events, and are instantly replicated as many times as needed

# Azure Functions

# What is Azure Functions?

An event-based, serverless compute experience that accelerates app development

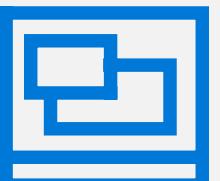


## Azure Functions = FaaS++



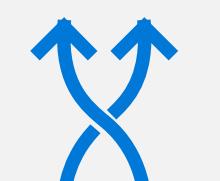
### Integrated programming model

Use built-in triggers and bindings to define when a function is invoked and to what data it connects



### Enhanced development experience

Code, test and debug locally using your preferred editor or the easy-to-use web based interface including monitoring



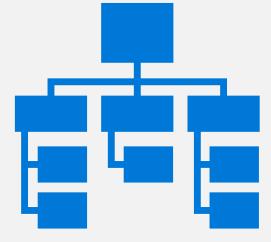
### Hosting options flexibility

Choose the deployment model that better fits your business needs without compromising development experience

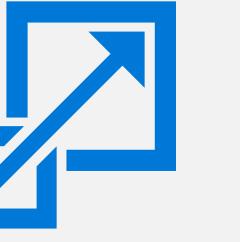




# Focus on code, not plumbing



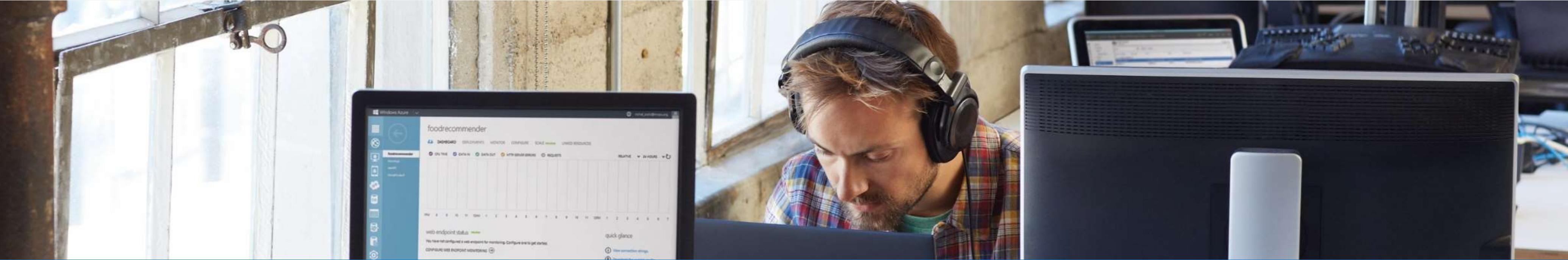
No infrastructure  
management



Auto-scale based  
on your workload



No wasted resources,  
pay only for what you use



# Boost development efficiency



## Triggers

Use triggers to define how functions are invoked  
Avoid hardcoding with preconfigured JSON files  
Build serverless APIs using HTTP triggers



## Bindings

Connect to data with input and output bindings  
Bind to Azure solutions and third-party services  
Use HTTP bindings in tandem with HTTP triggers



## Proxies

Define one API surface for multiple function apps  
Create endpoints as reverse proxies to other APIs  
Condition proxies to use variables



## Local debugging

Debug C# and JavaScript functions locally  
Use debugging tools in Azure portal, VS, and VS Code



## CI/CD

Save time with built-in DevOps  
Deploy functions using App Service for CI  
Leverage Microsoft, partner services for CD



## Monitoring

Integrate with Azure Application Insights  
Get near real-time details about function apps  
See metrics around failures, executions, etc.



# Gain flexibility and develop your way



## Multiple languages

Write code in C#, JavaScript, F#, and Java  
Continuous investment in new, experimental languages



## Durable Functions

Write stateful functions in a serverless environment  
Simplify complex, stateful coordination problems  
Add the extension to enable advanced scenarios



## Hosting options

Choose from six consumption plans to run Functions  
Run your first million function executions for free



## Dev options

Simplify coding for new users with native Azure portal  
Select from popular editors, like VS, VS Code, CLI, Maven\*



# Gain flexibility and develop your way



## Hosting options

### Consumption

Serverless



Only pay for what you use; charges apply per execution and per GB second

### AS Plan

Free, Basic, Standard, Premium



Gain all the advantages of Functions along with Microsoft's financially-backed SLA and the always-on features of an App Service Plan

### AS Environment

Network isolation



Use a dedicated App Service cloud environment (ASE) that comes with network isolation for apps, greater scale, and secure connectivity to local vNets

### Azure Stack

On-premises



Bring the power of the entire Azure stack to your own data centers

### Runtime

Functions on your server



Run Functions on your local server; does not include the entire Azure stack

### IoT Edge\*

On devices



Deploy custom Azure modules on IoT devices

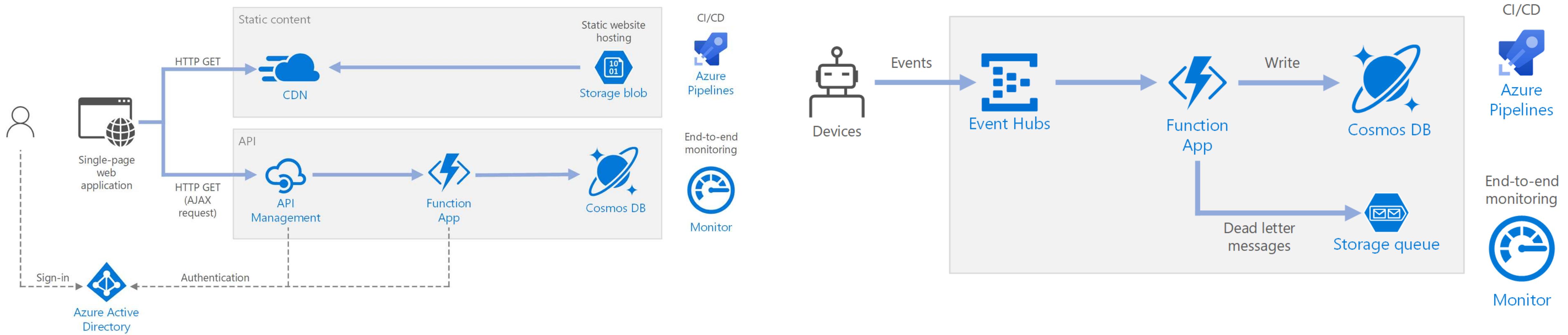
# Azure Functions is an **open-source** project

Functions runtime and all extensions are fully open source

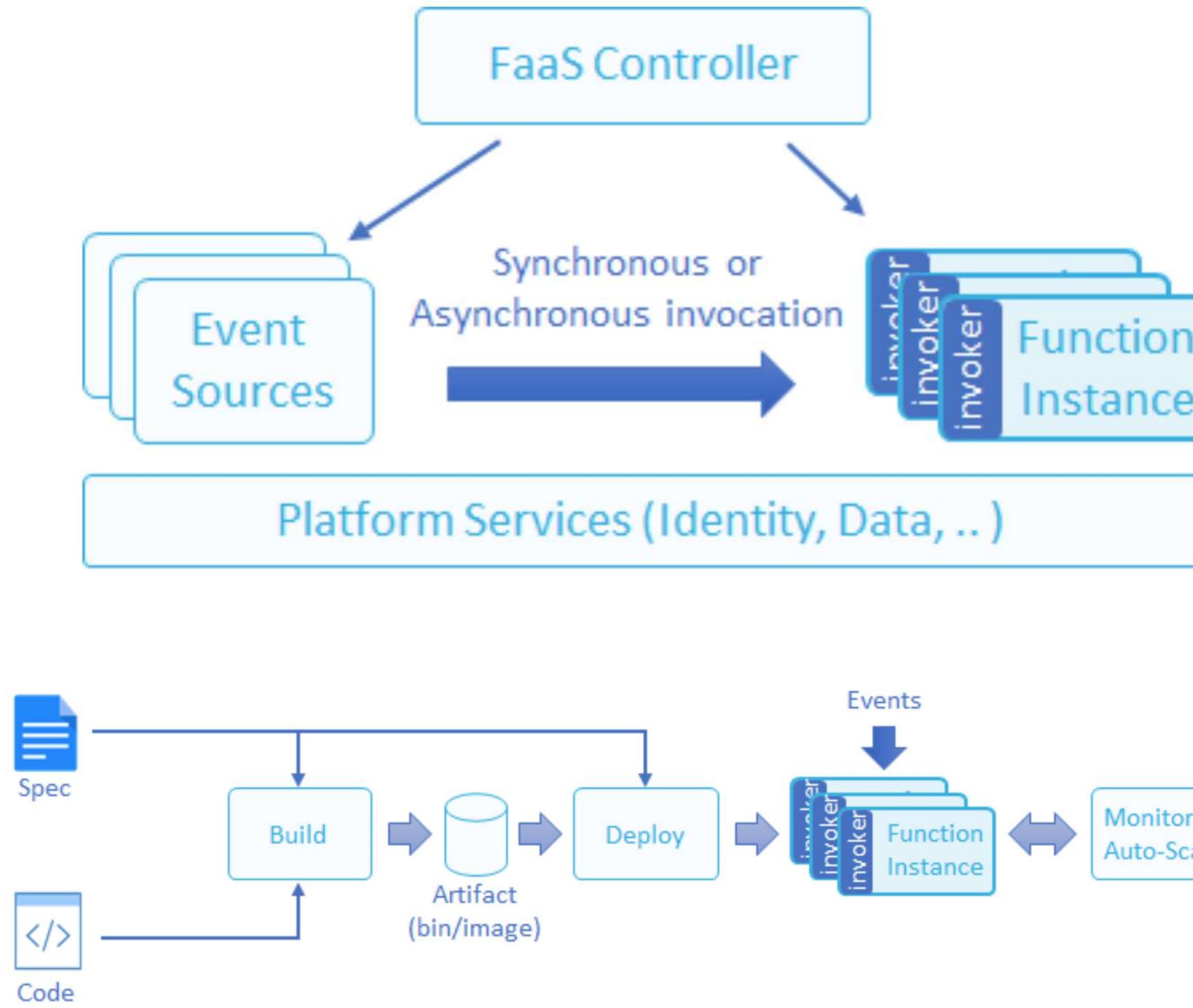


<https://github.com/Azure/Azure-Functions>

# Serverless Apps on Azure



# Portable Serverless Platforms



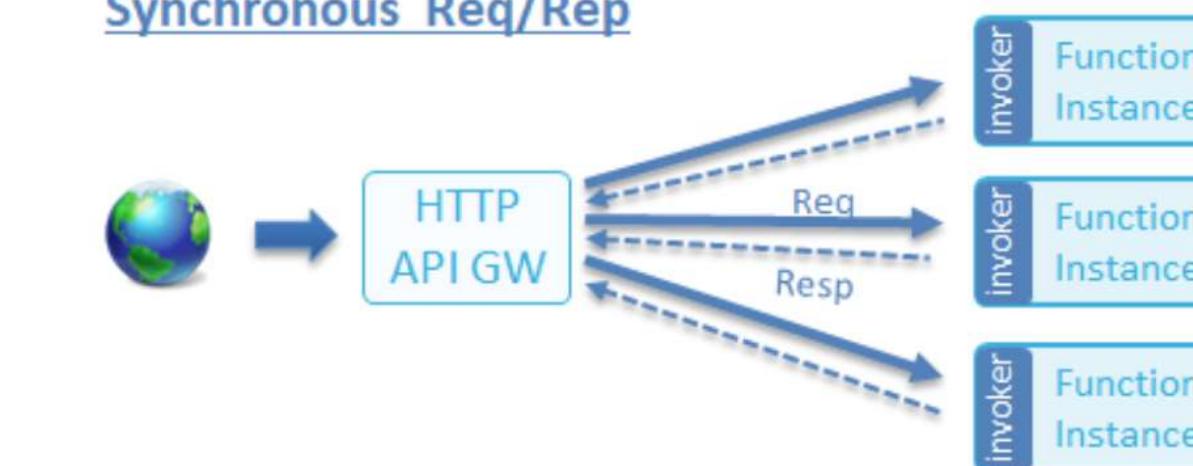
Nuclio

OpenFaaS

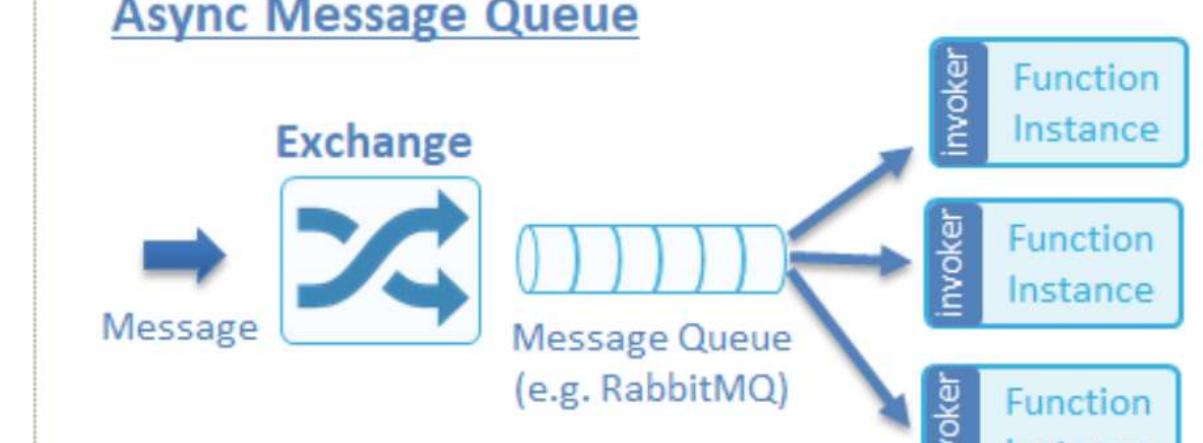
Galactic Fog

OpenWhisk

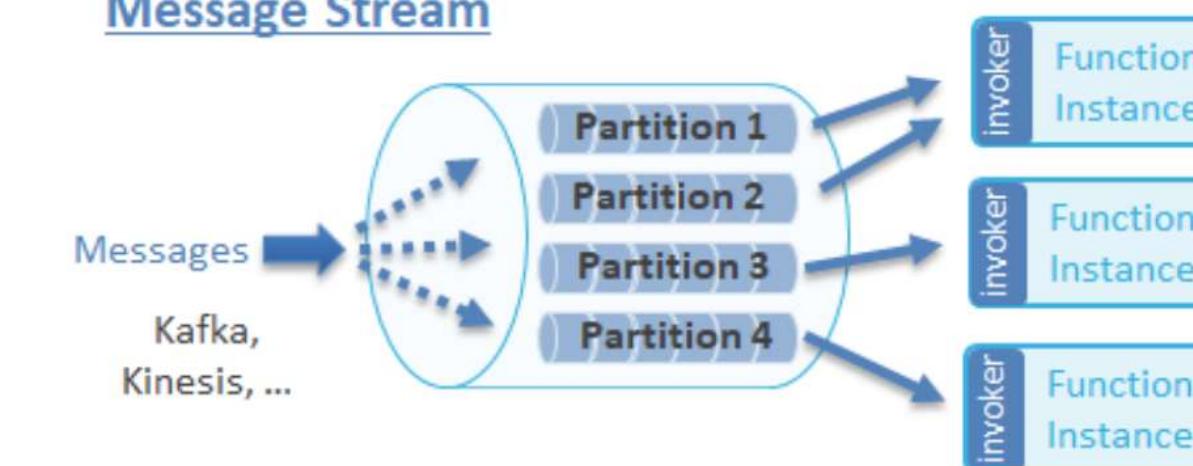
## Synchronous Req/Rep



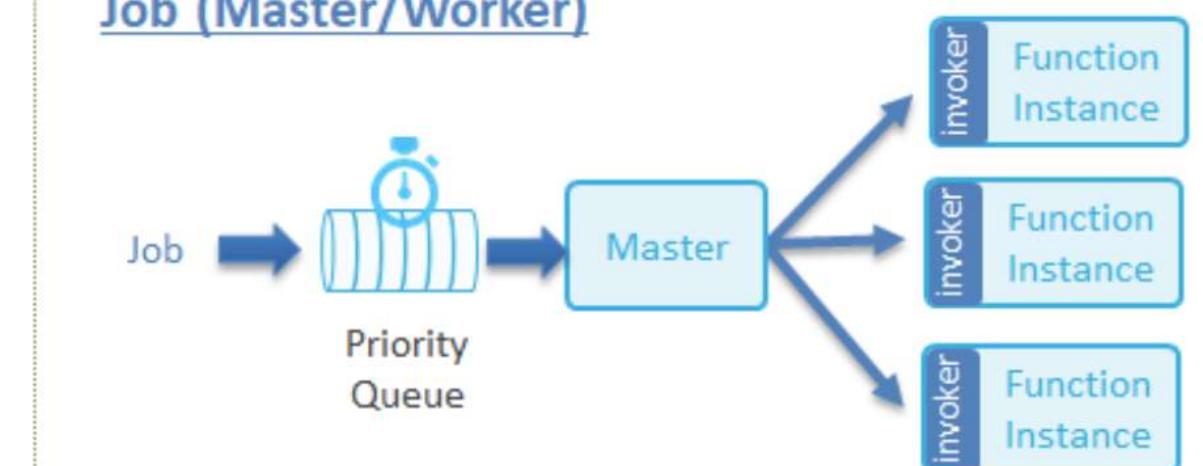
## Async Message Queue



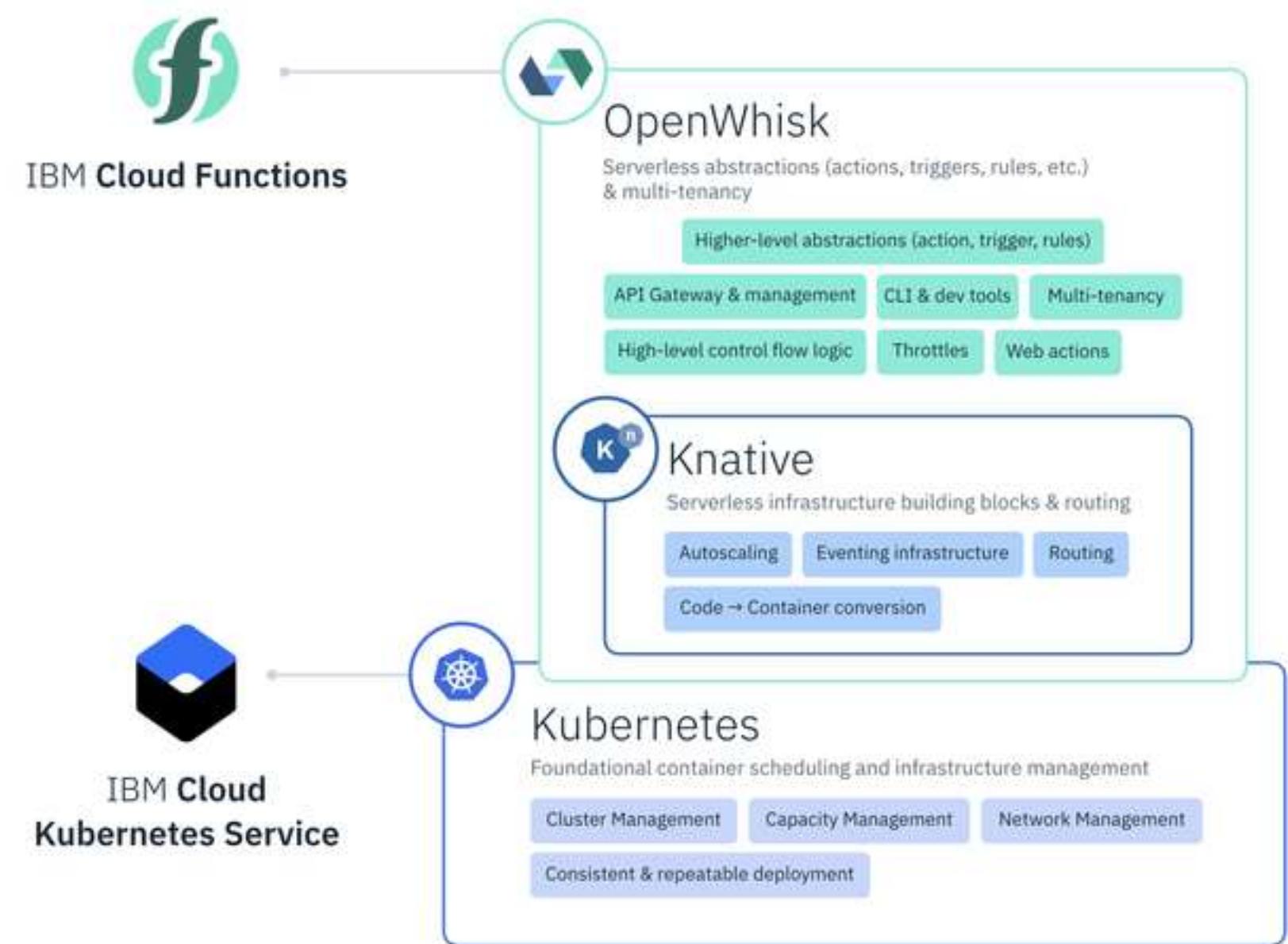
## Message Stream



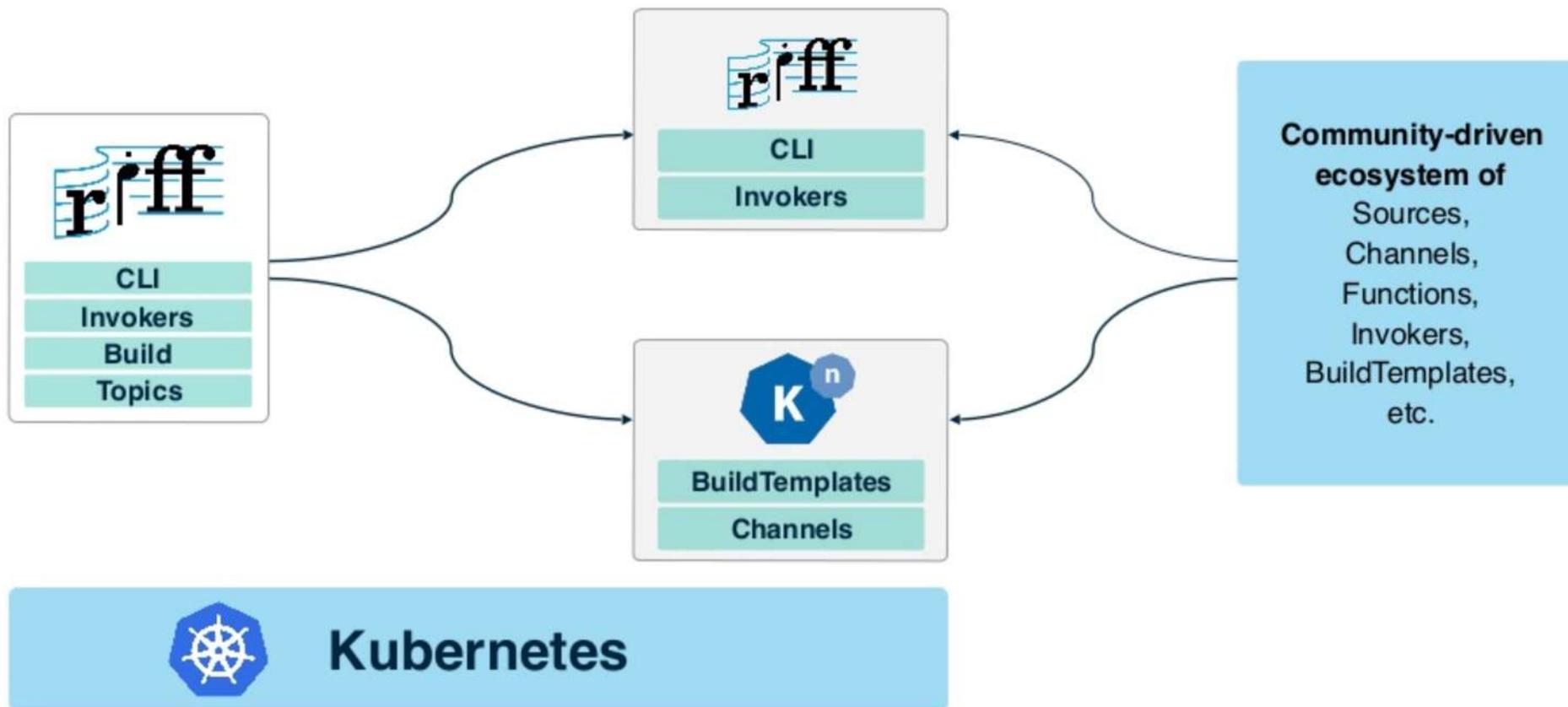
## Job (Master/Worker)



# KNative & Friends



## riff on Knative



# Open Source Tooling

# Microsoft drives community-led innovations for Kubernetes



68% of Kubernetes users\* prefer Helm as their package manager



Visual Studio Code Kubernetes Extensions has 11K monthly active users

---

## Microsoft also maintains...

---

Helm



Draft



Brigade



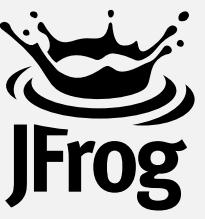
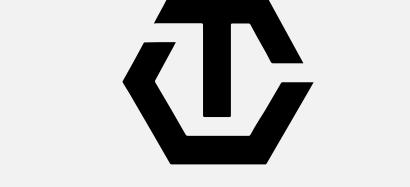
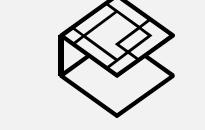
Cloud Native  
Application Bundles  
(CNAB)



Virtual  
Kubelet



# Work how you want with opensource tools and APIs

	Development	DevOps	Monitoring	Networking	Storage	Security
<p>Take advantage of services and tools in the Kubernetes ecosystem</p>    	   	   	    	 	 	  
<p>Leverage 100+ turn-key Azure services</p>  		 DevOps  ARM	 Azure Monitor	  Azure Policy	 Azure Storage	 Container Registry  Azure Active Directory  Key Vault



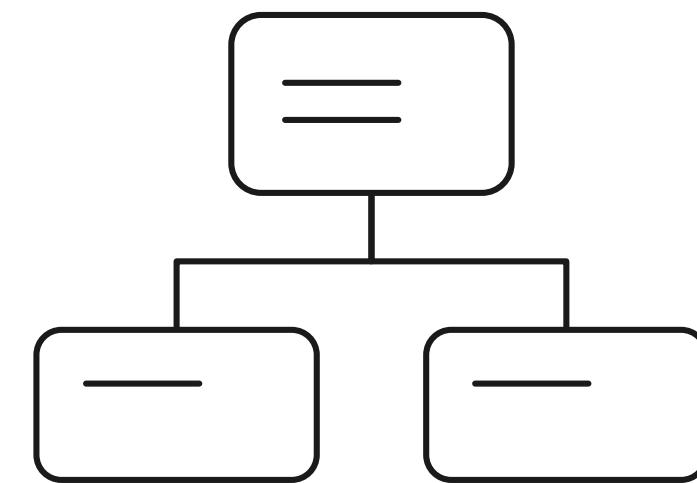
# The package manager for Kubernetes

Helm is the best way to find, share, and use software built for Kubernetes.

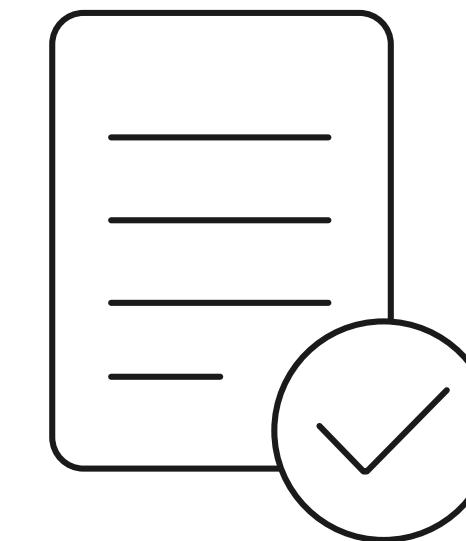




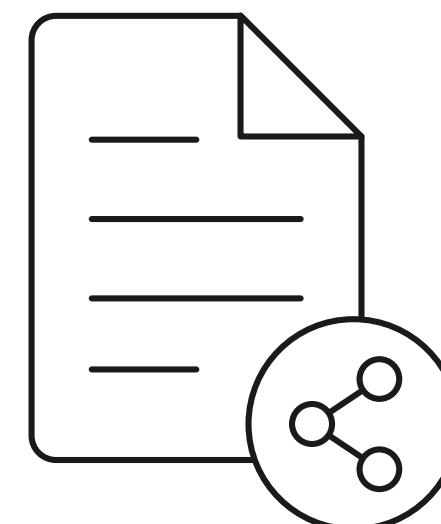
Find, share, and use software built for k8s



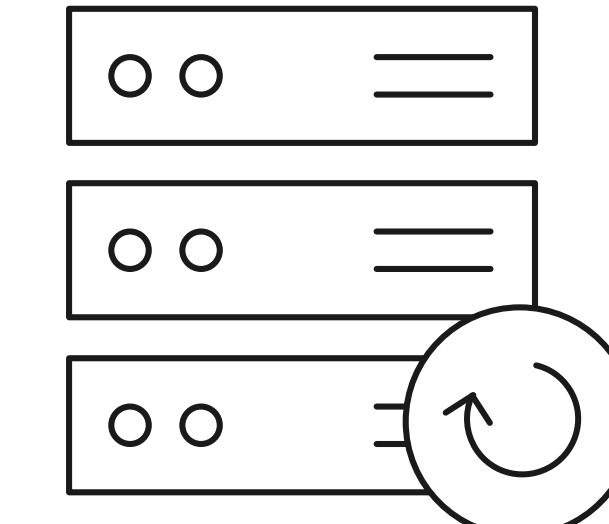
Manage complexity



Easy updates



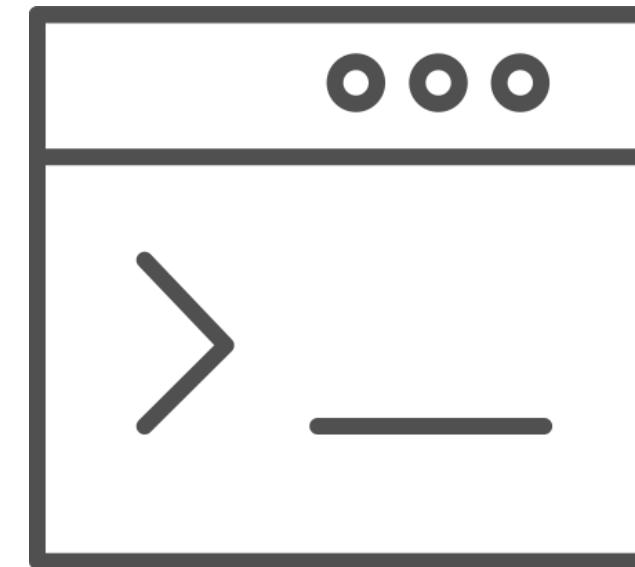
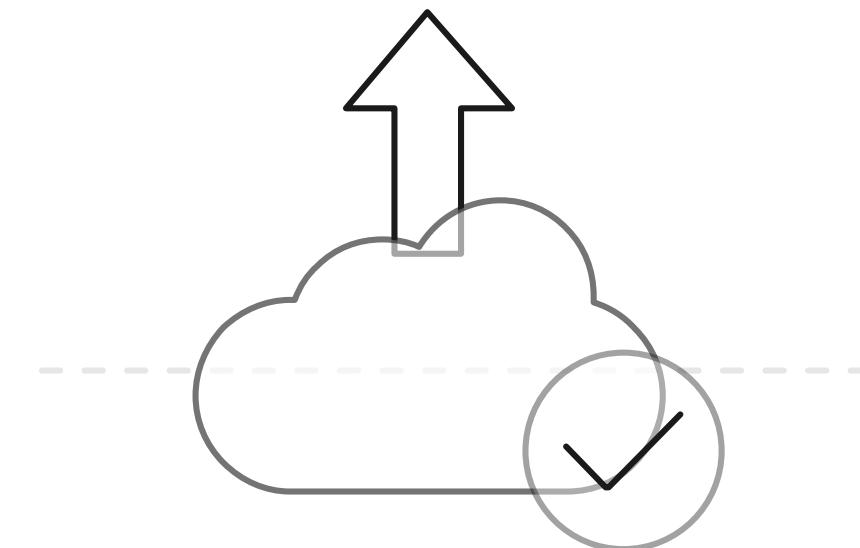
Simple sharing



Rollbacks



## Simple app development and deployment – into any Kubernetes cluster



### Simplified development

Using two simple commands, developers can now begin working on container-based applications without requiring Docker or even installing Kubernetes themselves

### Language support

Draft detects which language your app is written in, and then uses packs to generate a Dockerfile and Helm Chart with the best practices for that language

# Developer Workflows

# 3 Abstractions

- Containers
- Functions, triggered by Events
- Managed Cloud Services

# Many tools!



# Developer Workflows

Local

Cloud

Code

Container

Functions

Services

# Docker Desktop

# Docker Desktop



# Docker Compose

- docker-compose for local dev
- docker-compose Kubernetes CRD

# Developer Workflows

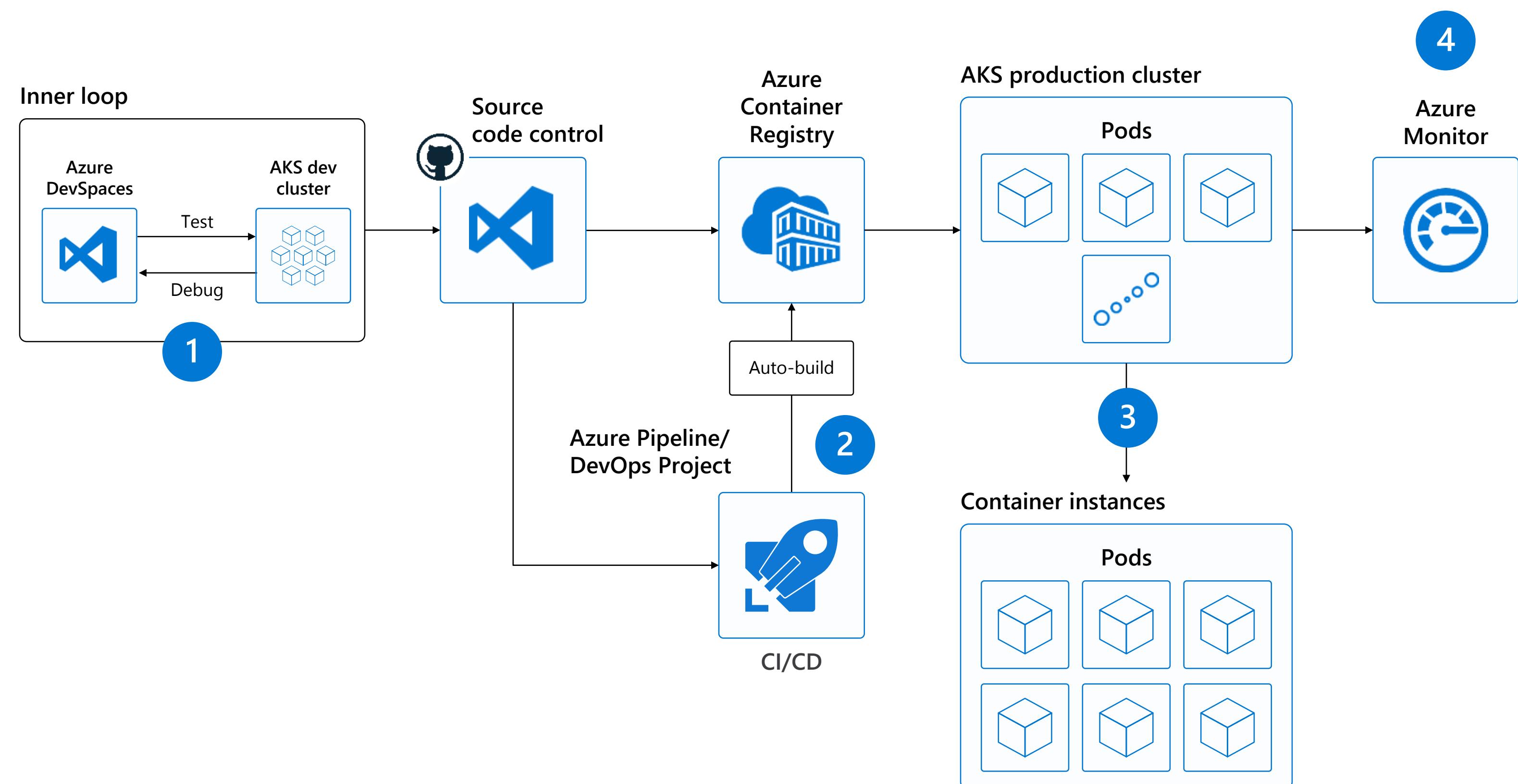
	Local	Cloud
Code	code mounted in local containers inotify to restart if needed	
Container	docker-compose	
Functions	Azure Functions Runtime	
Services	mysql, mongo images	CosmosDB

# Azure Dev Spaces

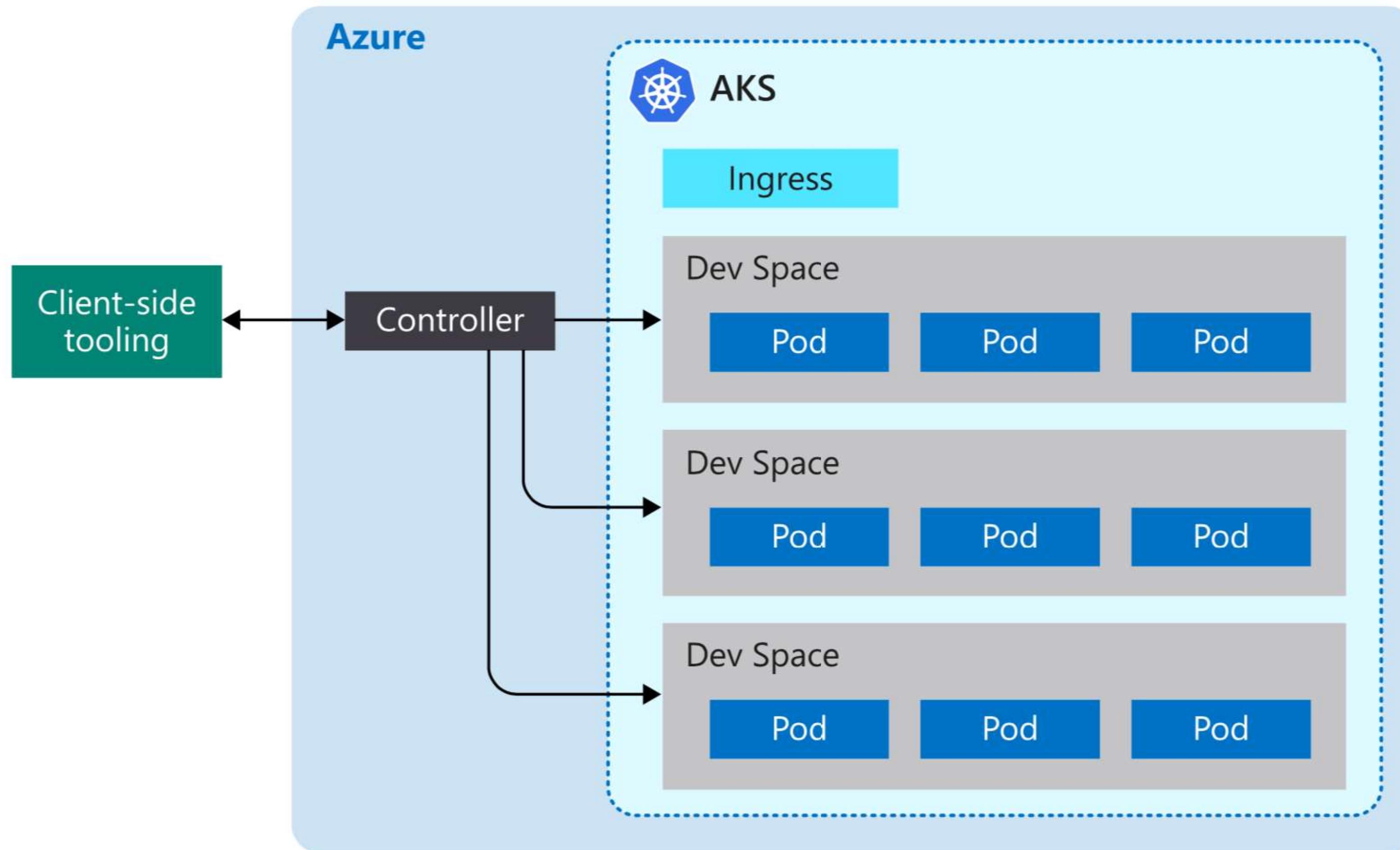
# Integrated end-to-end Kubernetes experience

## Capabilities

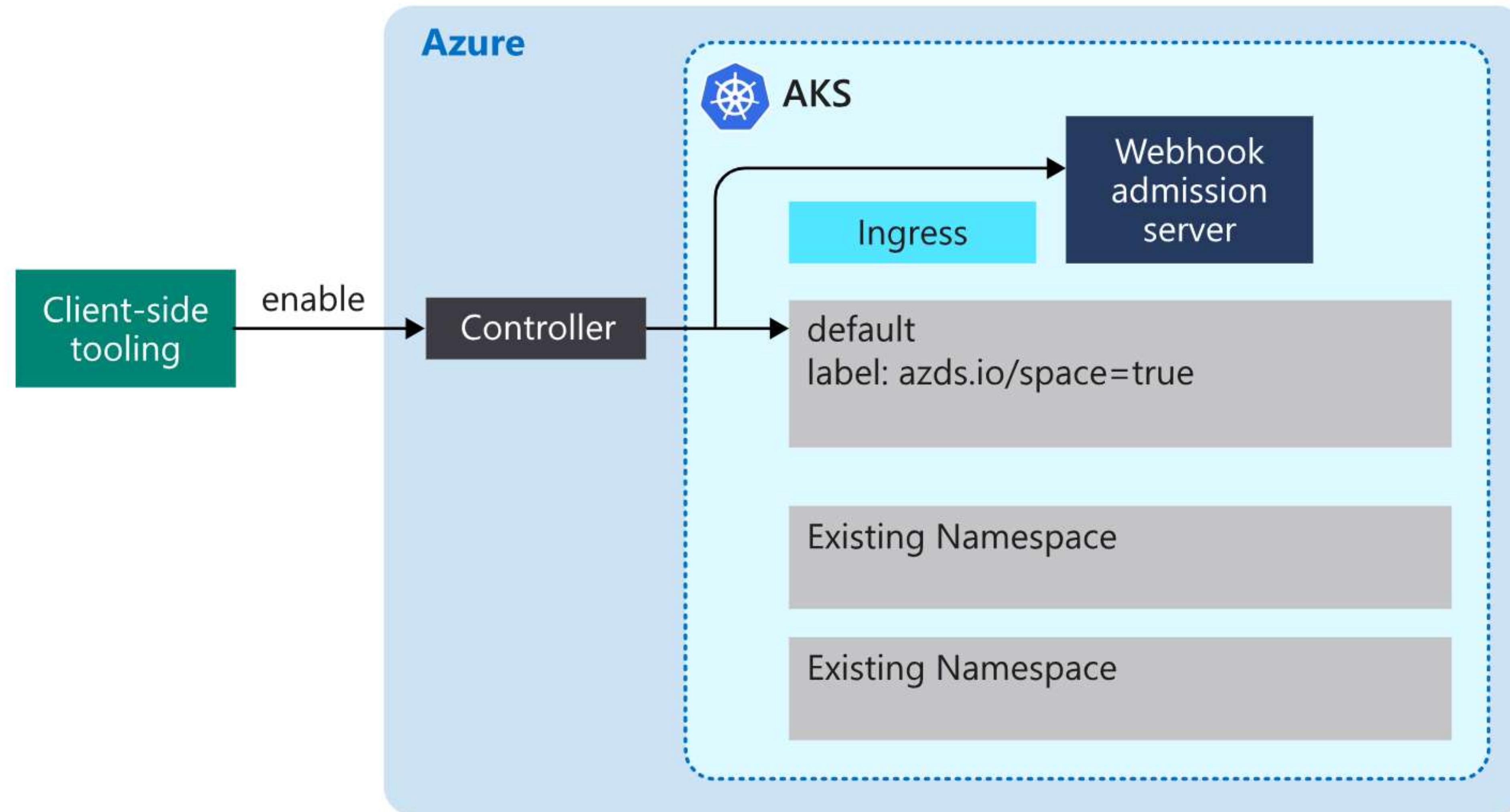
1. Use **Azure Dev Spaces** to iteratively develop, test, and debug microservices targeted for AKS clusters.
2. **Azure DevOps** has native integration with Helm and helps simplifying continuous integration/continuous delivery (CI/CD)
3. **Virtual node**—a Virtual Kubelet implementation—allows fast scaling of services for unpredictable traffic.
4. **Azure Monitor** provides a single pane of glass for monitoring over app telemetry, cluster-to-container level health analytics.



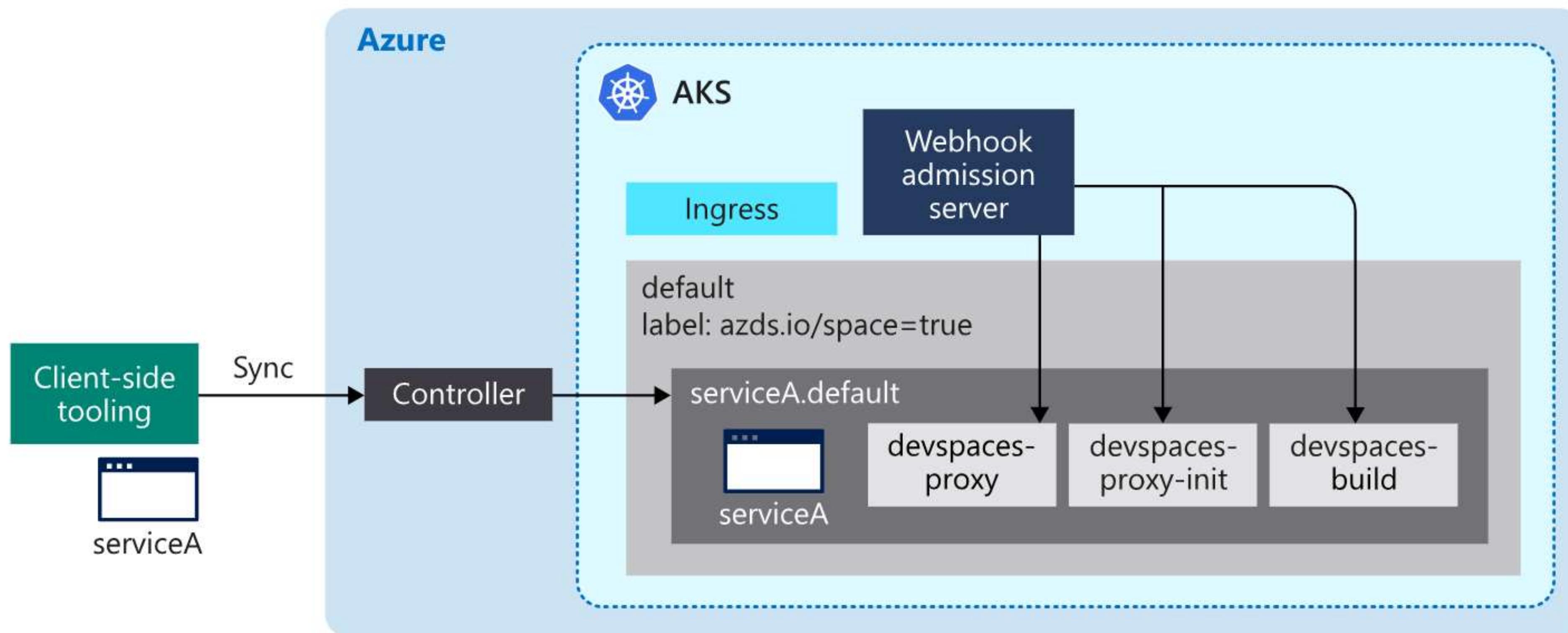
# Azure Dev Spaces



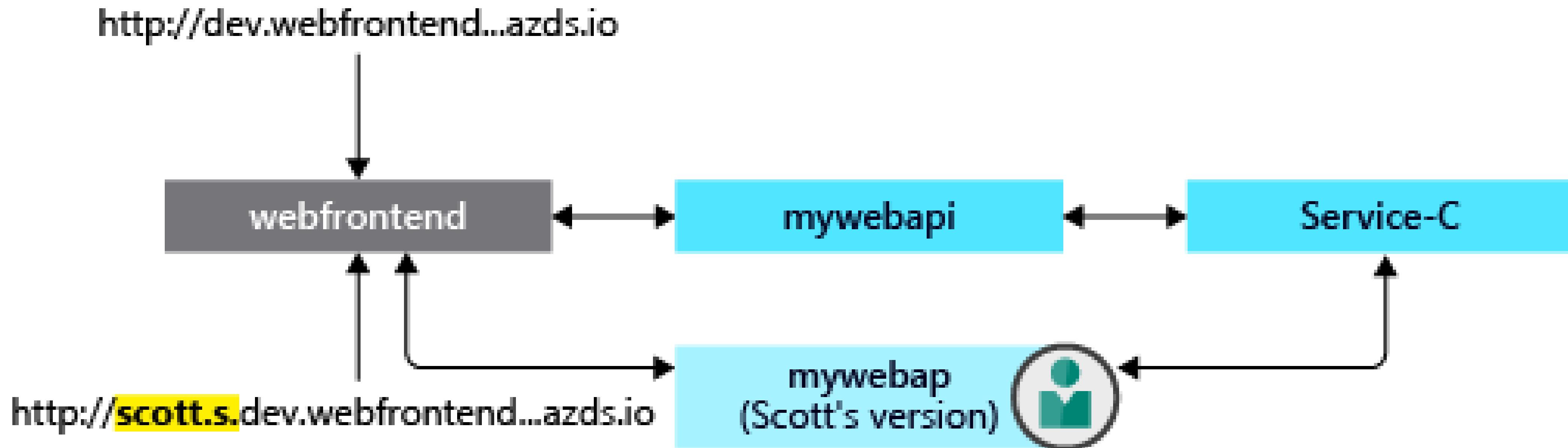
# Azure Dev Spaces



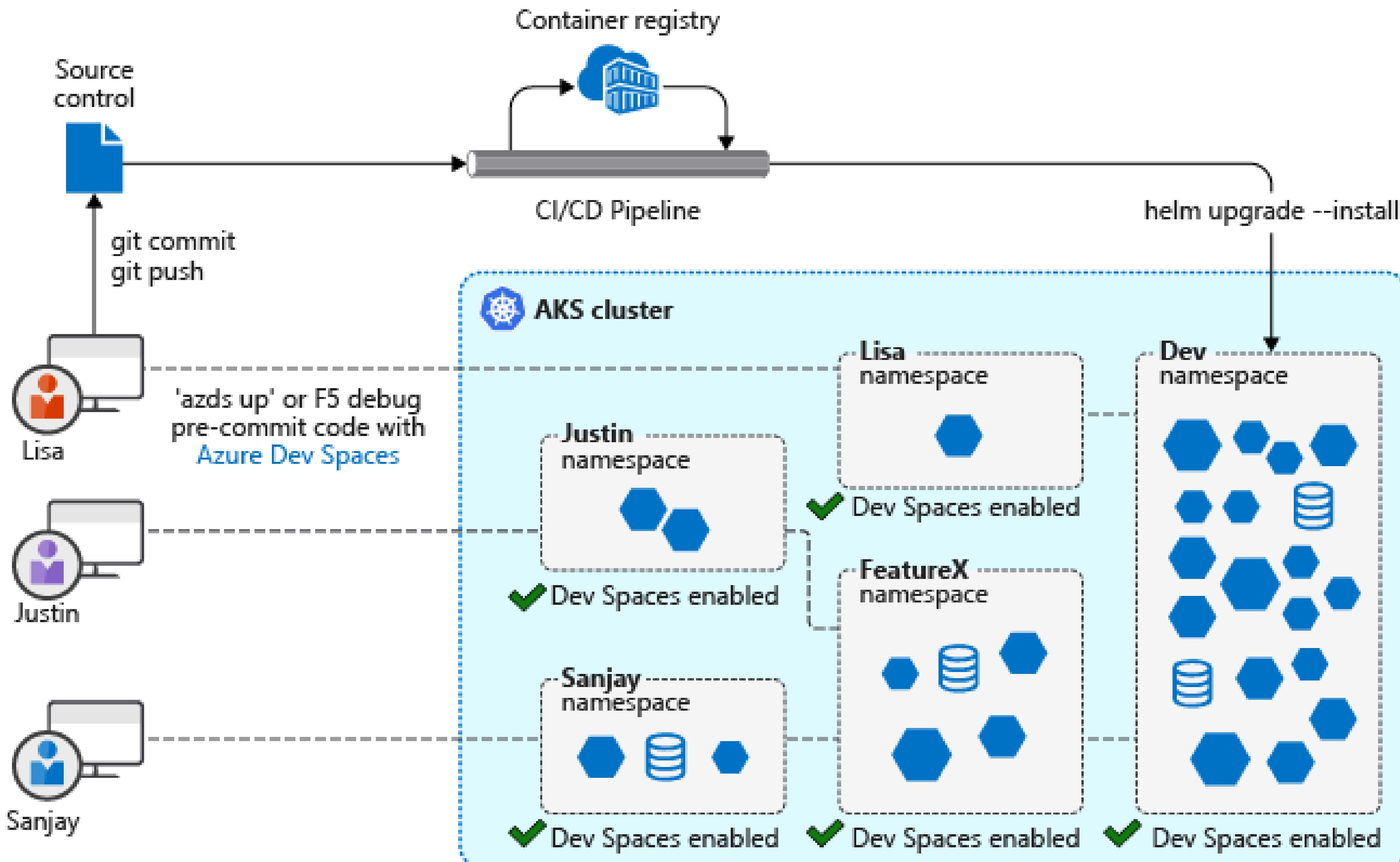
# Azure Dev Spaces



# Azure Dev Spaces



# Azure Dev Spaces



# Developer Workflows

Local

Code

code synced to cloud

Container

Functions

Services

Cloud

code built, started

AKS

Azure Functions

Azure

# VS Code Live Share

# VS Code Live Share

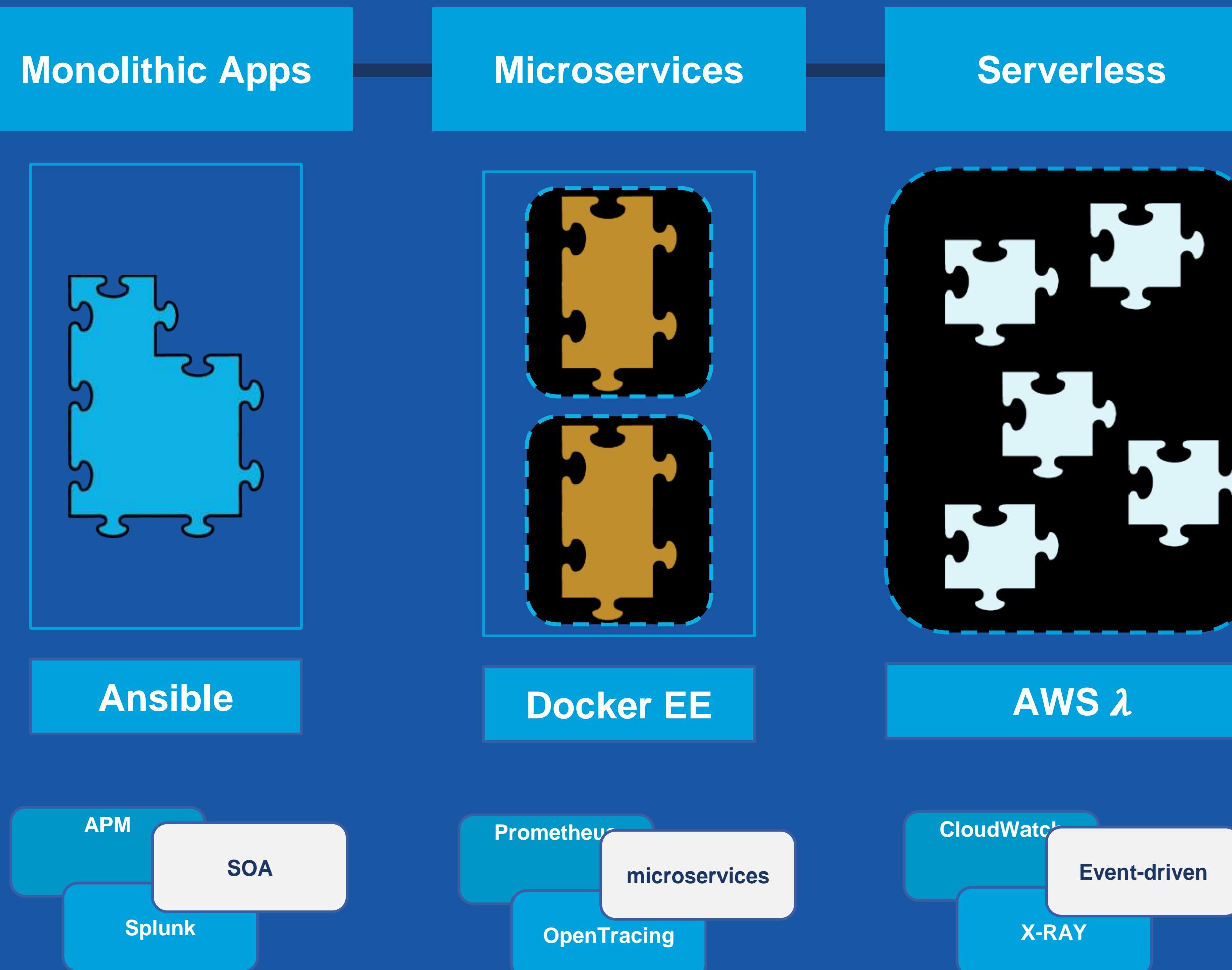
- You just need VS Code locally
- Code and all setup on your collaborator's machine
- Code together without setting anything up
- Access services on remote machine from localhost
- Works with Azure Dev Spaces on machine sharing the session: double jump to AKS

# Gloo





# THE PROBLEM: Disparate ecosystems, hard transition

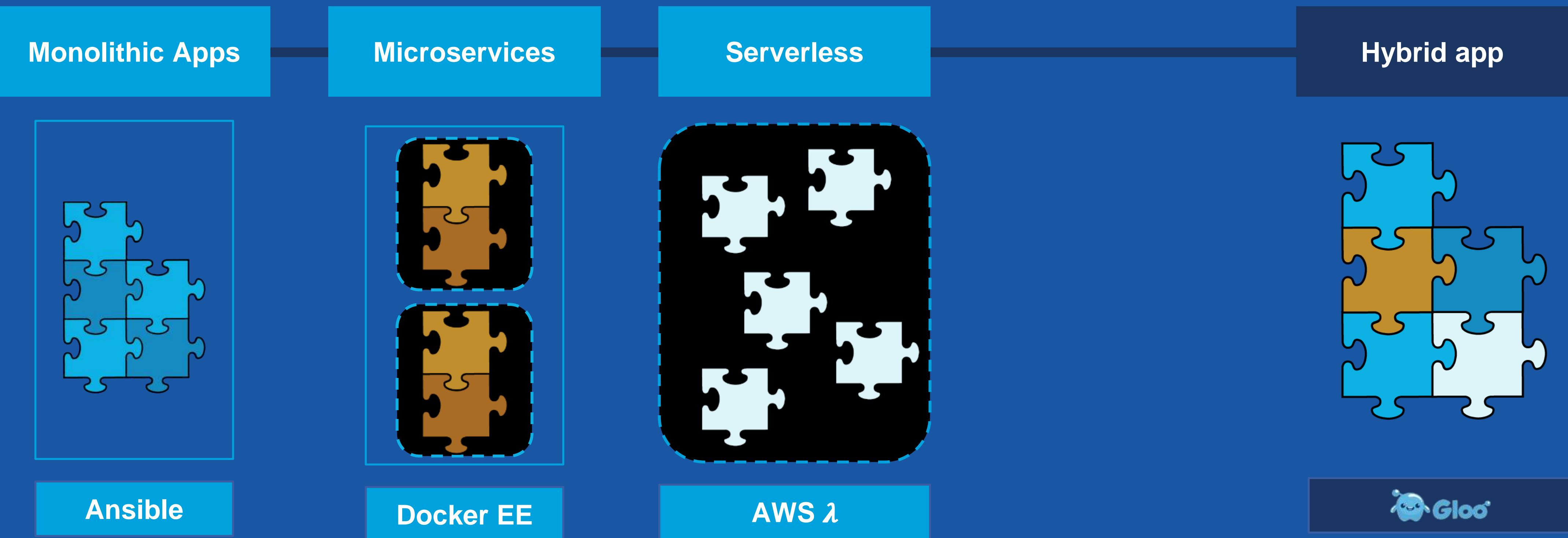


Enterprise faces 4 main problems in adopting innovative architectures:

1. Insolation between brown and green field
2. Transition is lengthy and diverts essential personnel from core mission
3. Duplicate redundant tools
4. Requires silo teams

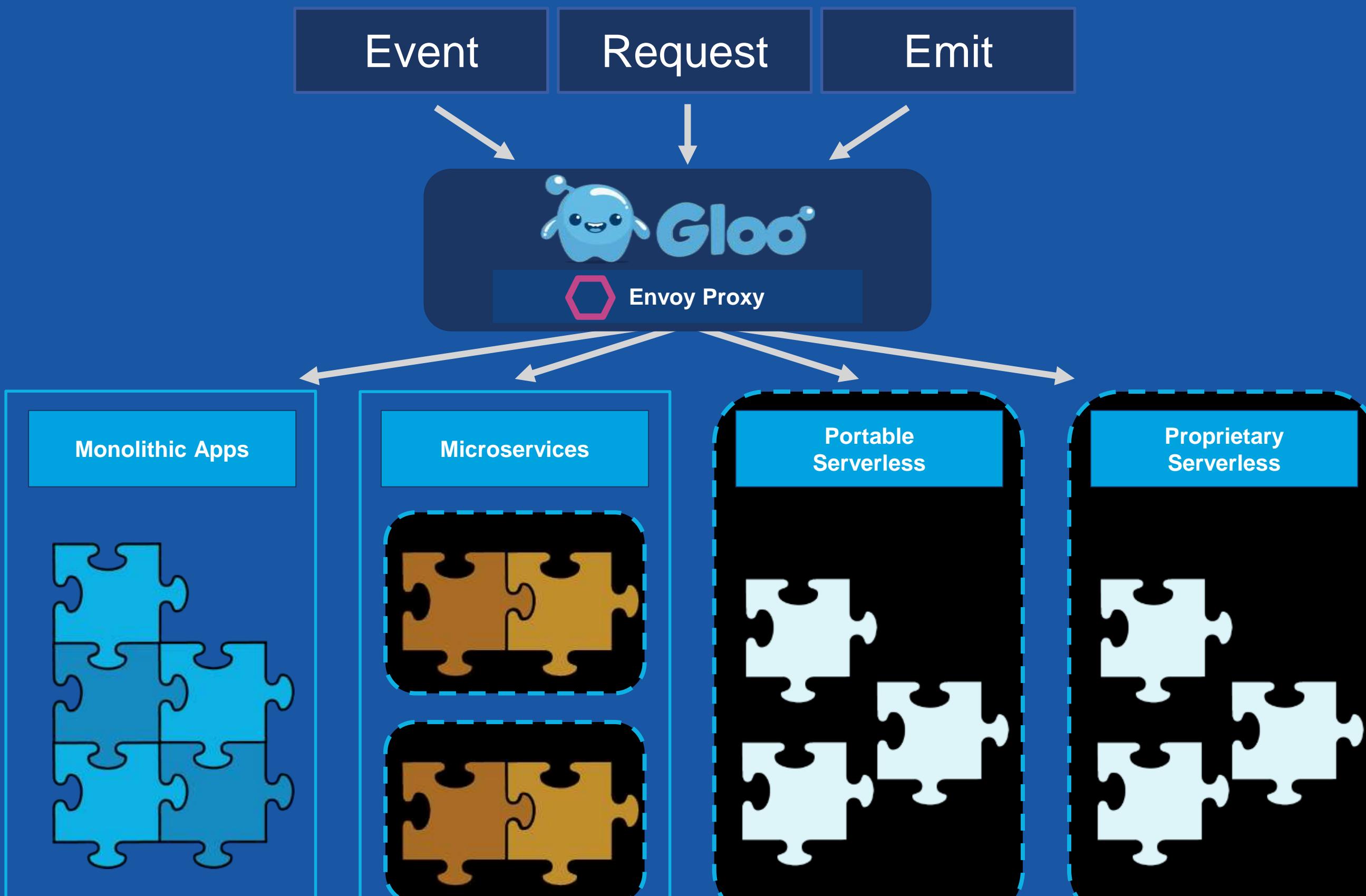


# OUR VISION: Hybrid App



**Pros:** fast delivery of new features in modern architectures; stop “digging the hole”; gradual transition = minimal interruption; unified tooling.

# Gloo will glue together your hybrid environment



**Centralized place** to manage security, observability, traffic

**Function-level routing:** canary deployment, security, caching

**Date plane:** builds around and extends the Envoy proxy (C++)

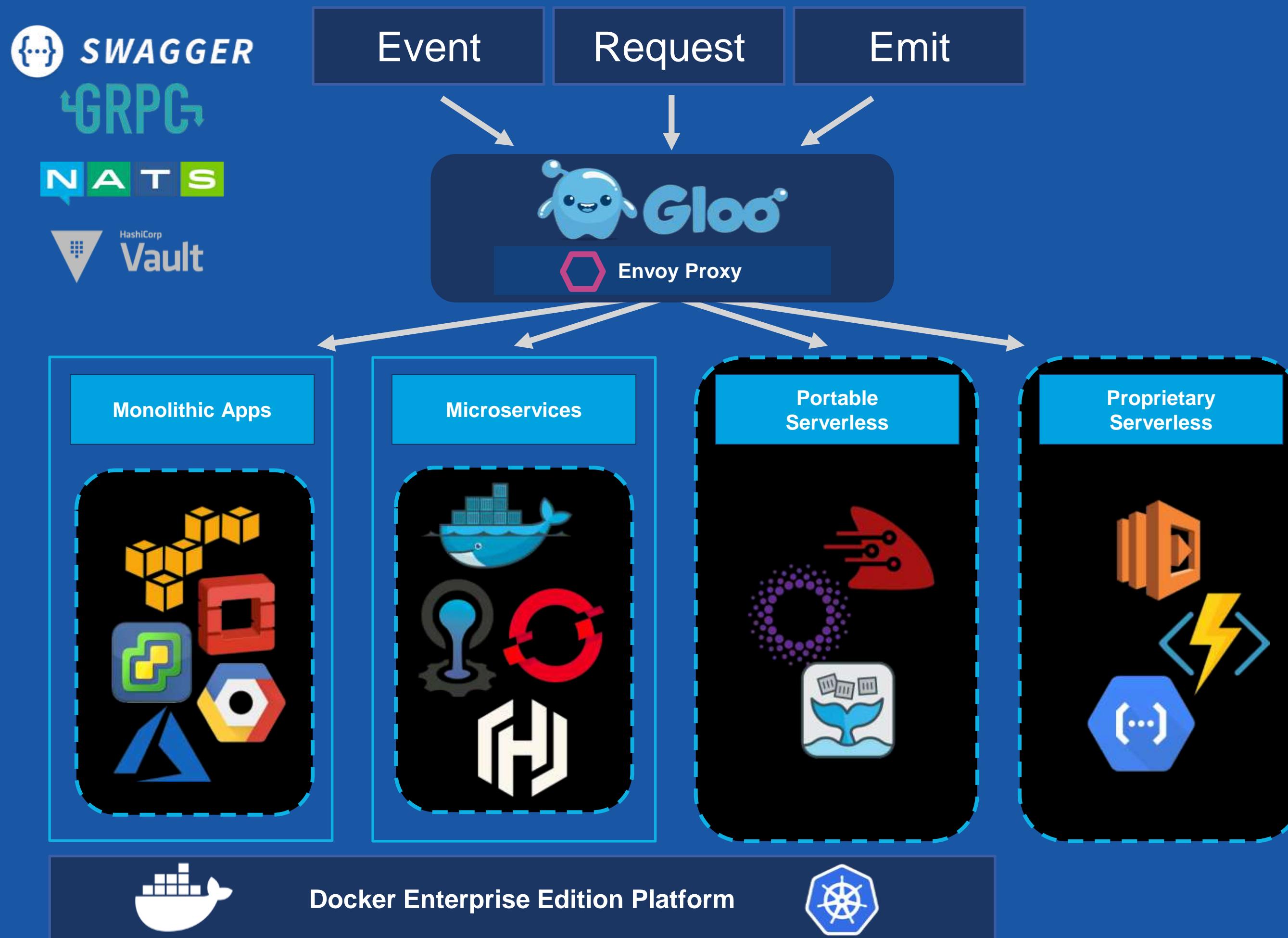
**Control plane:** pluggable, extensible architecture (GO)



Docker Enterprise Edition Platform



# Gloo will glue together your hybrid environment



**Automatic discovery** of all your functions, services & apps

**Full integration** with all components of your environment: **all clouds, all platforms, all technologies**



docker  
con SF18

# Let Gloo build your hybrid apps

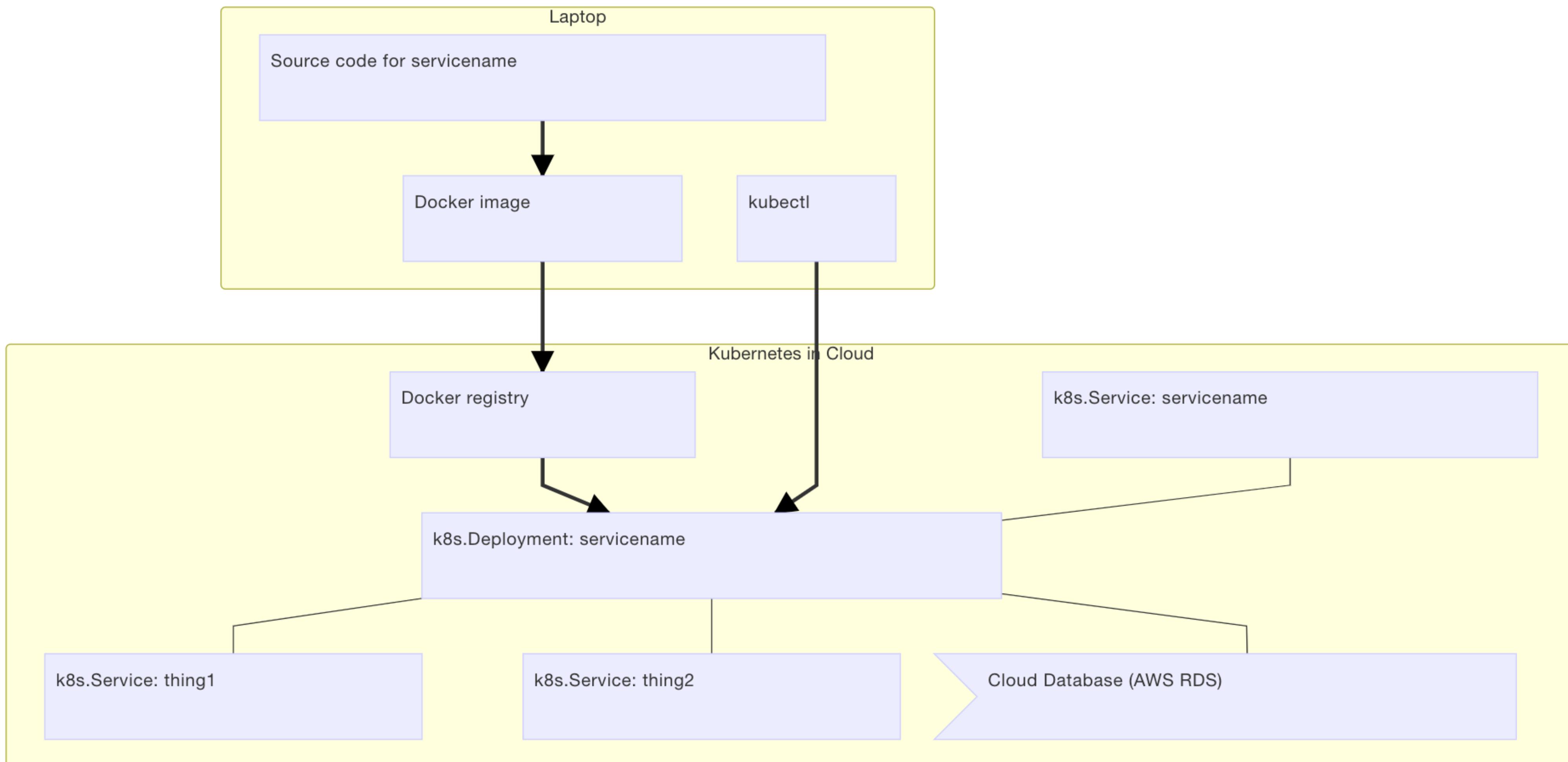


# Developer Workflows

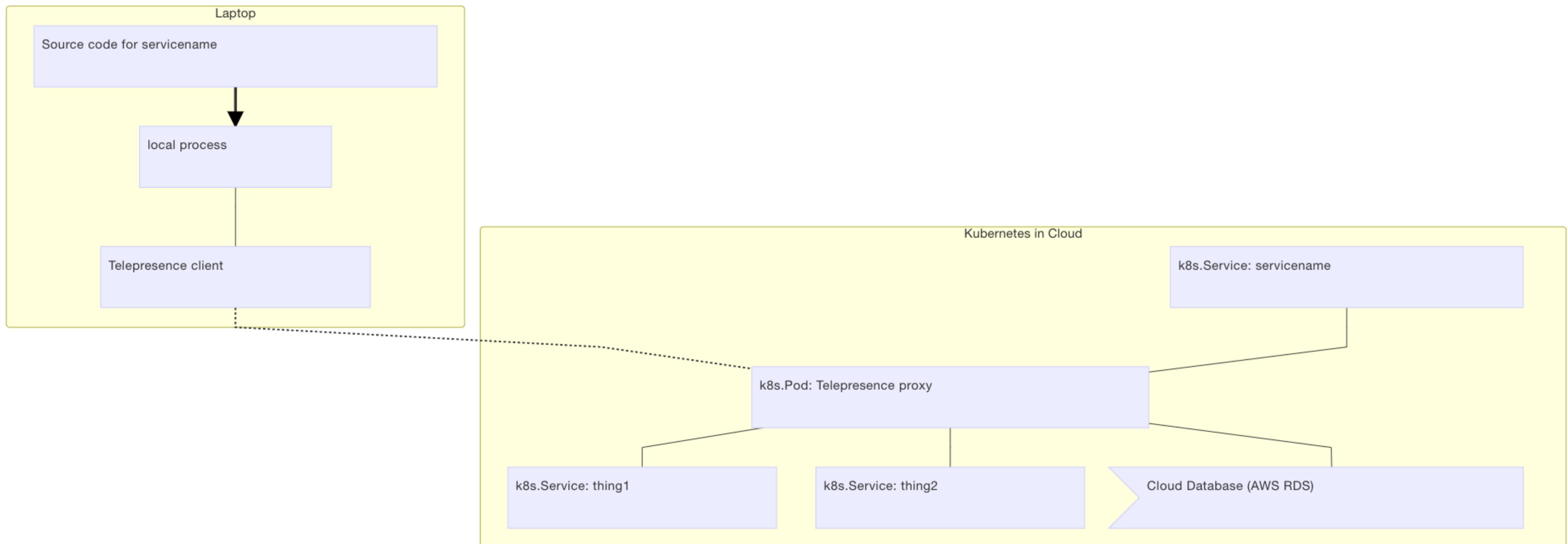
	Local	Cloud
Code	debugger attach to container in cloud	
Container		AKS
Functions		Azure Functions
Services		Azure

# Telepresence

# Telepresence



# Telepresence



# Developer Workflows

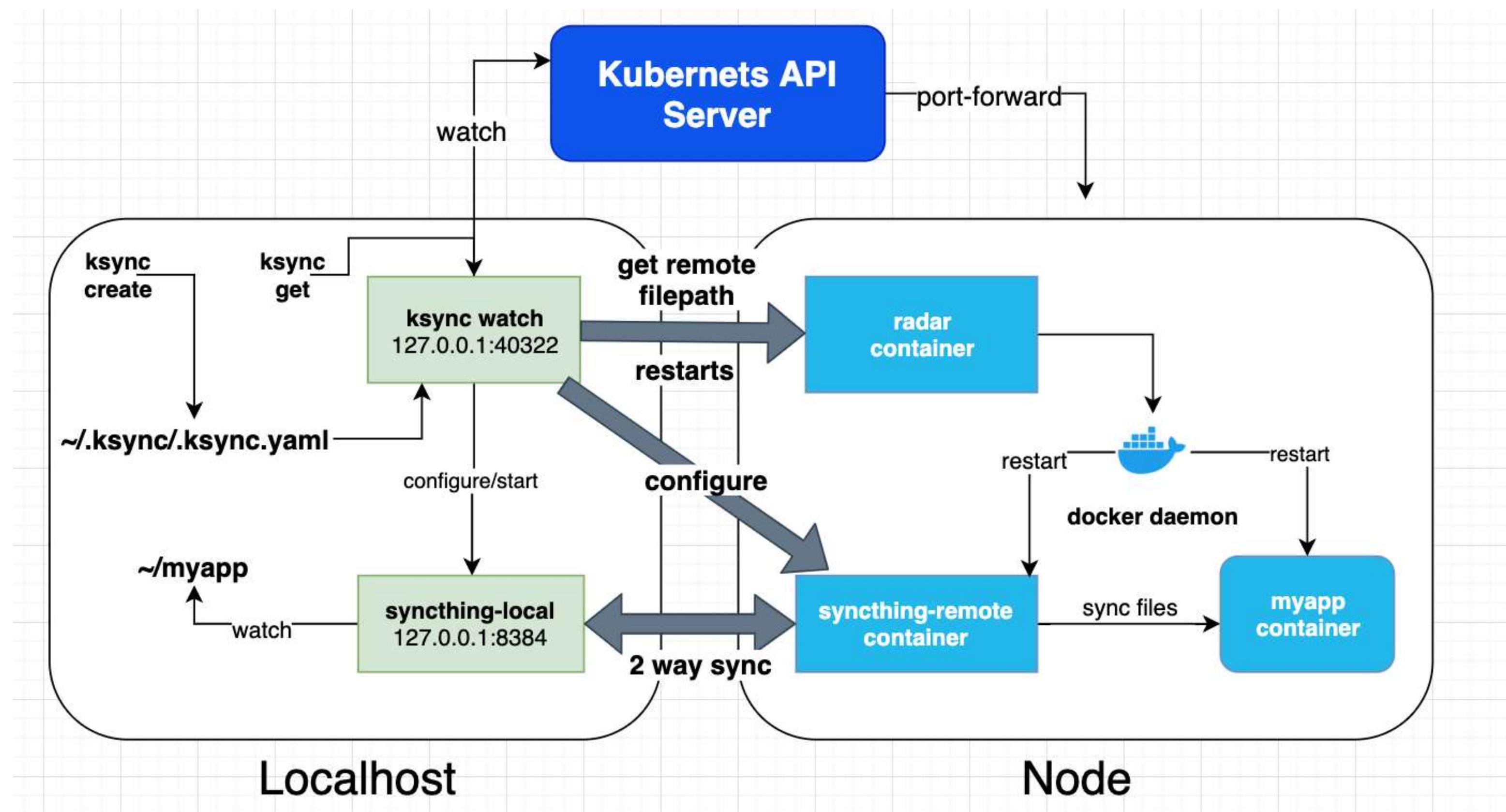
	Local	Cloud
Code	local code	
Container	local process proxied inside aks	AKS
Functions		Azure Functions
Services		Azure

# Ksync

If you've been wanting to do something like `docker run -v /foo:/bar` with Kubernetes, ksync is for you!

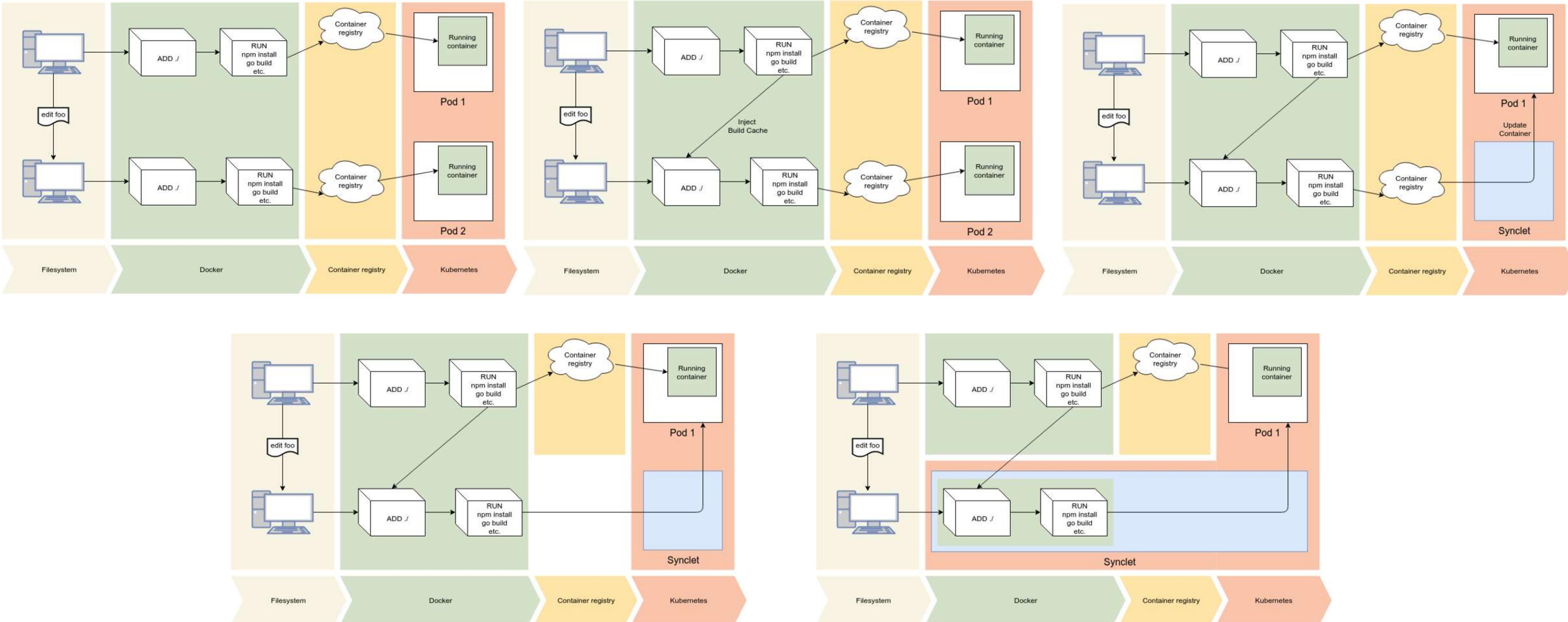
Using ksync is as simple as:

1. `ksync create --pod=my-pod local_directory remote_directory` to configure a folder you'd like to sync between your local system and a specific container running on the cluster.
2. `ksync watch` to connect to the cluster and start updating the container.
3. Use your favorite editor, like [Atom](#) or [Sublime Text](#) to modify the application. It will auto-reload for you remotely, in seconds.



# Tilt

# Tilt



<https://medium.com/windmill-engineering/how-tilt-updates-kubernetes-in-seconds-not-minutes-28ddffe2d79f>

# Tilt

## Tiltfile Walkthrough

This Tiltfile is going to start out looking like any other. We're first going to grab the YAML that defines the Kubernetes service.

```
k8s_yaml('serve.yaml')
```

Next we tell Tilt about how to build the Docker image. We also use `live_update` to provide the lightning-fast reload times that frontend developers expect.

```
docker_build('tilt-frontend-demo', '.',  
    live_update=[  
        # Map the local source code into the container under /src  
        sync('.', '/src'),  
    ])  
)
```

This is fast, but has a bug: when you change `package.json`, the dependencies don't get updated. Let's use the fall back feature of Live Update to fix that:

```
docker_build('tilt-frontend-demo', '.',  
    live_update=[  
        # when package.json changes, we need to do a full build  
        fall_back_on('package.json', 'package-lock.json'),  
        # Map the local source code into the container under /src  
        sync('.', '/src'),  
    ])
```

Now we're cruising! Updates that don't require a `bundle update` zoom by in less than a second.

<https://github.com/windmilleng/tilt-frontend-demo>

# Developer Workflows

Code

Container

Functions

Services

Local

local changes  
synced to aks

Cloud

AKS

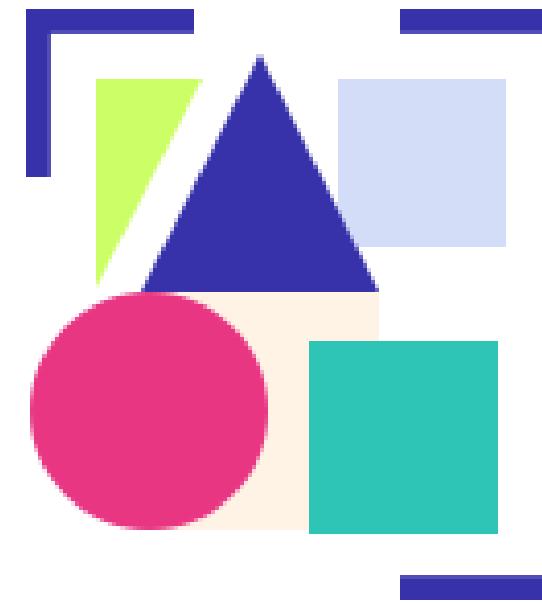
Azure Functions

Azure

# CNAB

# Cloud Native Application Bundle

## Spec for packaging distributed apps



### **CNAB: package distributed apps**

CNABs facilitate the bundling,  
installing and managing of  
container-native apps — and their  
coupled services

# Duffle

Install and manage distributed app bundles



## **Duffle: install & manage distributed app bundles**

Simple CLI to interact with CNAB,  
for use with your clouds and  
services of choice

# porter

A friendlier cloud installer



## Install your app and its baggage

Bundle up not just the app, but everything it needs to run in the cloud

## Build bundles smarter, not harder

Use mixins for common tools and clouds, and depend on existing bundles.

## Surprise! It does package management too

Package and version your bundle, then distribute it for others to use.

# Microsoft at #GIDS19



Today 14:30: Maheshkumar R  
Architect your Solution using  
Virtual Nodes to Auto-Scale your  
Application on Kubernetes (AKS)

Tomorrow 11:30 Shashank Barsin  
Traceable and Safe Deployments to  
Kubernetes using CI/CD Pipelines

Booth game & CosmosDB  
demos

We're hiring: <https://aka.ms/ca-india>

# Build Viewing Parties

May 6-8 9 cities in India

Night parties

[events.microsoft.com](https://events.microsoft.com)

# Ignite | The Tour

May 22nd, Mumbai

The graphic features a large, semi-transparent hexagonal grid pattern in light blue and white against a dark background. In the bottom right corner, there is a photograph of the Taj Mahal Palace & Tower in Mumbai, showing its iconic red domes and white facade across a body of water with several boats.

**Microsoft Ignite | The Tour**  
**Mumbai**

Bombay Convention & Exhibition Centre  
May 22-23, 2019

Learn new ways to code, optimize your cloud infrastructure, and modernize your organization with deep technical training.

[Register for free >](#)

BUILD THE INTELLIGENT FUTURE

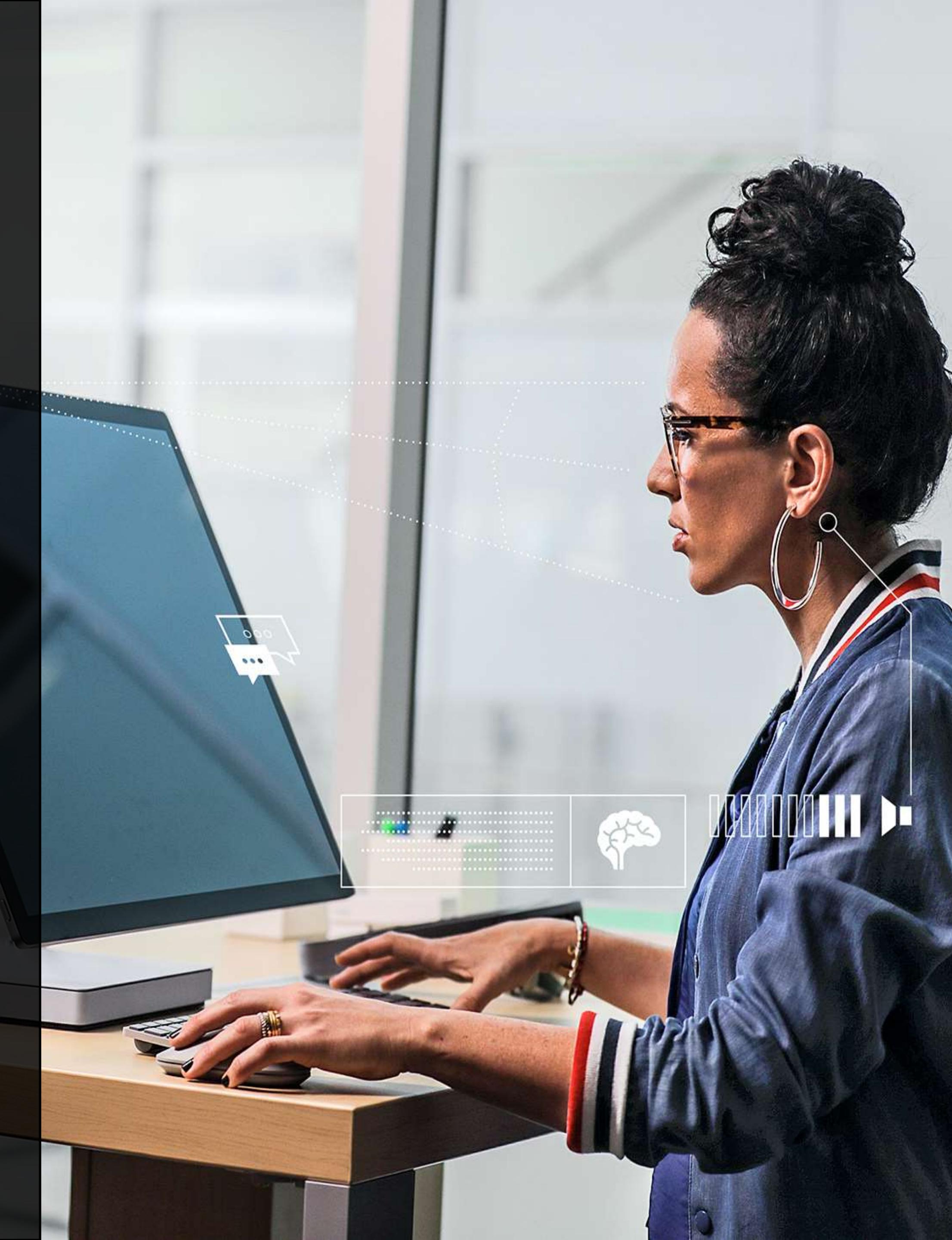
Announcing

# Microsoft “Week of AI”

VIRTUAL WORKSHOP SERIES

27th May – 31st May, 2019  
In India Timezone

To know more & register, visit:  
[aka.ms/WeekofAI](http://aka.ms/WeekofAI)



# Resources

Create a free Azure Account: <https://aka.ms/pat/account>

## Containers

docs: <https://aka.ms/pat/container>

learn: <https://aka.ms/pat/learn/container>

## Functions

docs: <https://aka.ms/pat/functions>

learn: <https://aka.ms/pat/learn/functions>

We're hiring: <https://aka.ms/ca-india>

# Thank you

@bridgetkromhout and @brendandburns for some of the slides  
@bcantrill for history of containerization slide



The End

PANORAMA

MADE IN HOLLYWOOD, U.S.A.

@chanezon

<https://aka.ms/pat/account>