



# Approach to Multi cloud

Myths, Reality, and way forward

Uli Homann  
Corp. VP & Distinguished Architect  
Cloud + AI engineering  
Microsoft Corp.

# Customer environments are evolving

## 100's–1,000's of apps



VMs



Databases



Containers



Serverless



## Diverse infrastructure



Datacenters



Hosters



Branch offices



OEM hardware



IoT devices



Edge

## Multi-cloud



Microsoft Azure



Google Cloud

# Motivation



## **Enterprise customer:**

Risk management – cloud dependencies are technology + hosting dependencies and therefore more impacting than either one alone

Regulatory requirement – ECB for example requires banks to be able to move out of a specific environment within a reasonable amount of time



## **ISV's: maximization of market impact**

# Definitions



Multi cloud or poly cloud: deploy solution/components across multiple clouds



Cloud agnostic: develop solutions that could potentially run on multiple cloud environments

# Approaches to consider



CLOUD-NATIVE



CLOUD-CONSISTENT



CLOUD-CONNECTED

# Key questions

---

What is your approach to multi-cloud?

---

Are you looking at cloud agnostic asset development?

---

What is the operations model you are looking for?

---

Which abstractions are you comfortable with?

---

What kind of dependencies are ok with you?

# Key themes

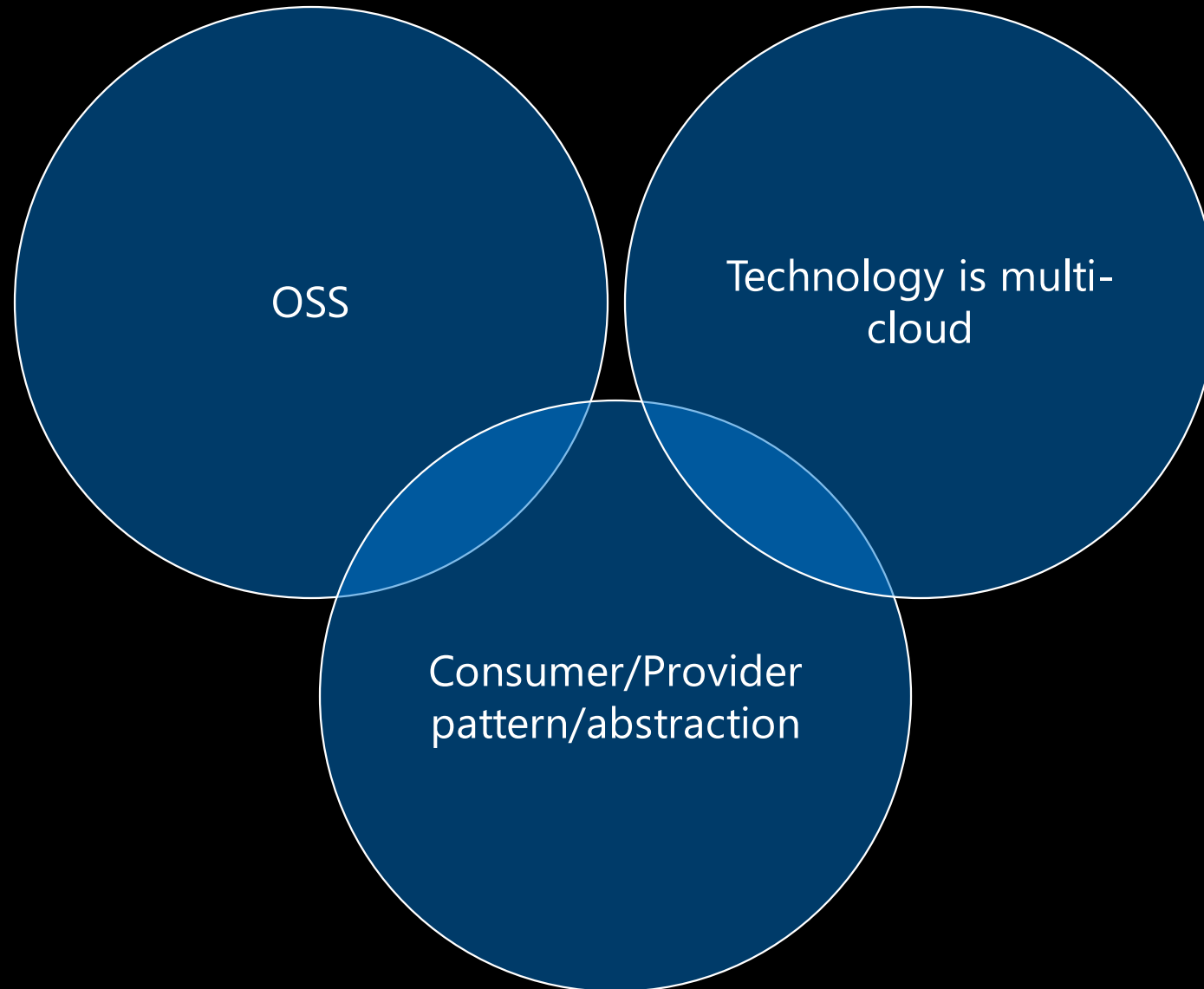
Cloud agnostic assets you care about are *developer* focused:

- Code
- Models
- Single process like CI/CD targeting multiple environments
- Key developer abstractions need to be multi-cloud (database, events, messages, etc.)

Pick a center of gravity for operations:

- Management
- Identity
- Security
- Logs
- Etc.

# Three ways to manage dependencies





Myths of  
Cloud  
agnostic  
applications





A large body of blue water with a dark silhouette of a person swimming in the distance. The water is bright blue with many small white reflections from the sun. The background is a dark, dense line of trees.

Myth

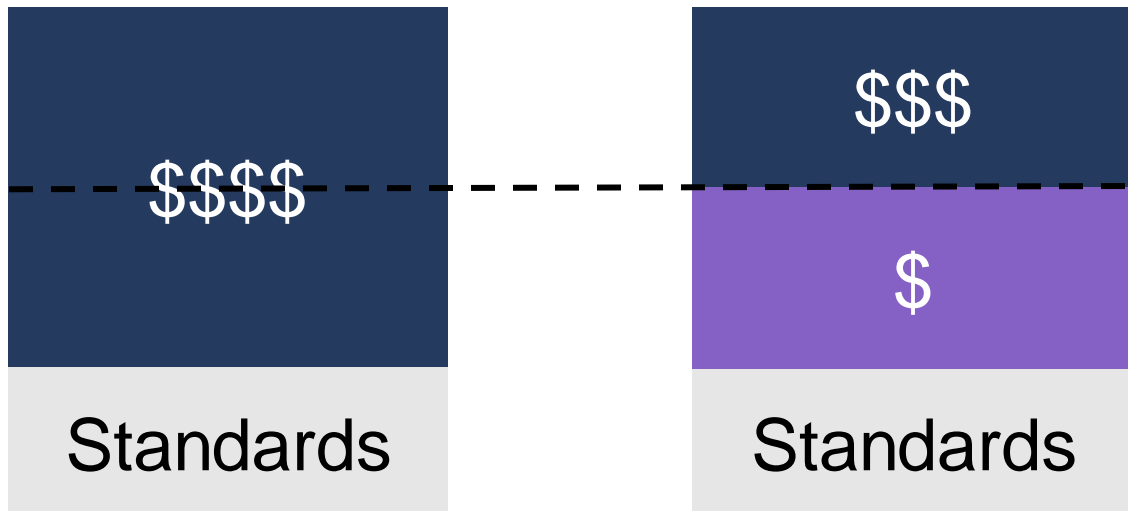
**If I only use the most basic  
cloud services, my  
application is portable**



More portability



Less portability



How much are you spending “below the value line” in this scenario?

Image credit: Flickr, [pshutterbug](#), CC 2.0





An underwater scene with a greenish-blue tint. In the center, a large, dark fish, possibly a shark or a large ray, swims horizontally. To the left, there are ancient stone ruins, including a large, ornate structure that looks like a throne or a ceremonial seat. The water is slightly murky, and there are other smaller fish visible in the background.

Myth

**If it's open source, I can port  
it wherever I need to**



# Along the spectrum Innovation

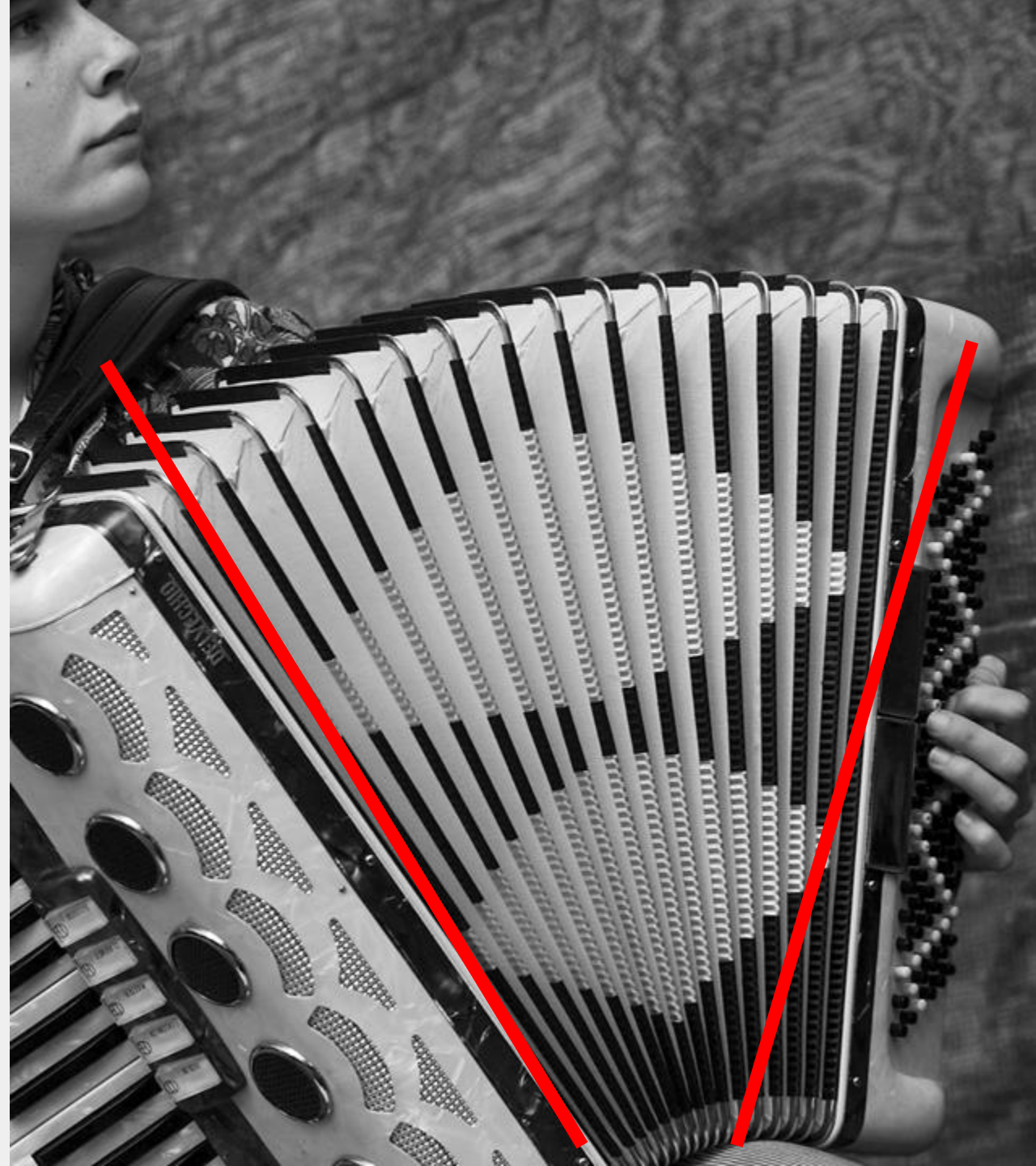
**Discoveries**

**Community**

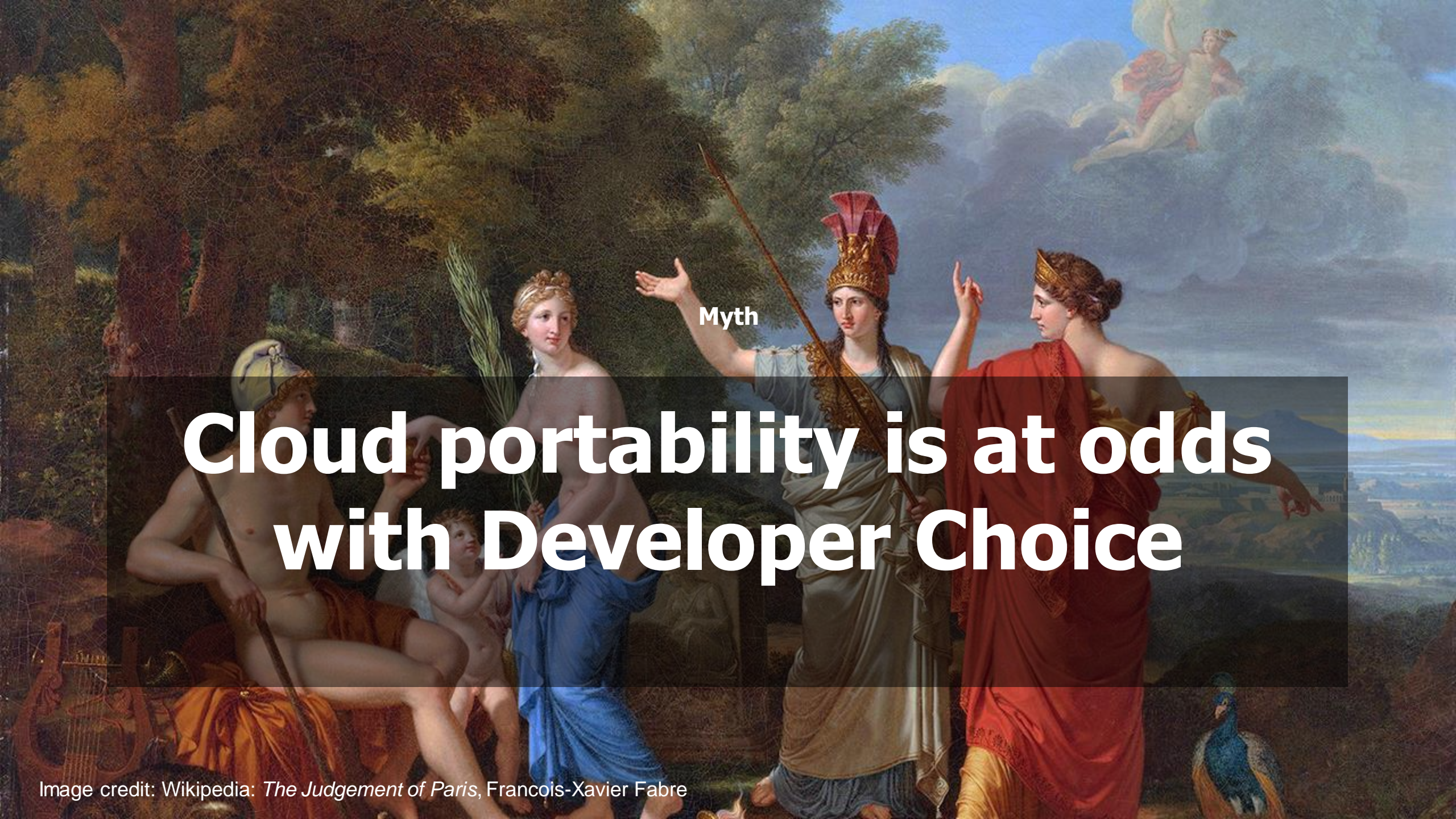
**De facto standards**

**De jure standards**

**Standards**







Myth

# Cloud portability is at odds with Developer Choice



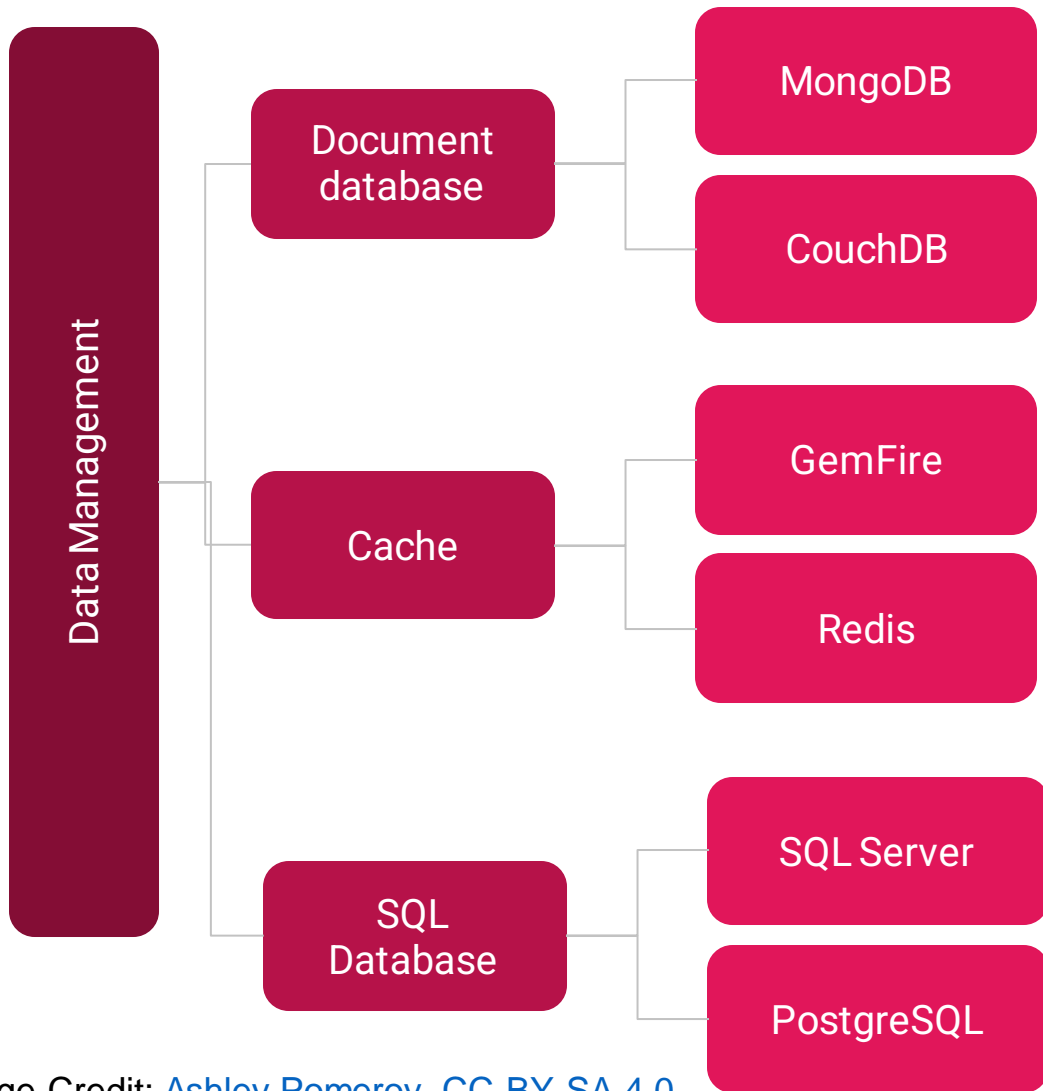


Image Credit: [Ashley Pomeroy, CC-BY-SA-4.0](#)







Myth

**Every Additional Developer  
Choice You Support Adds  
Proportionate Ops Overhead**



API/Protocol

Engine



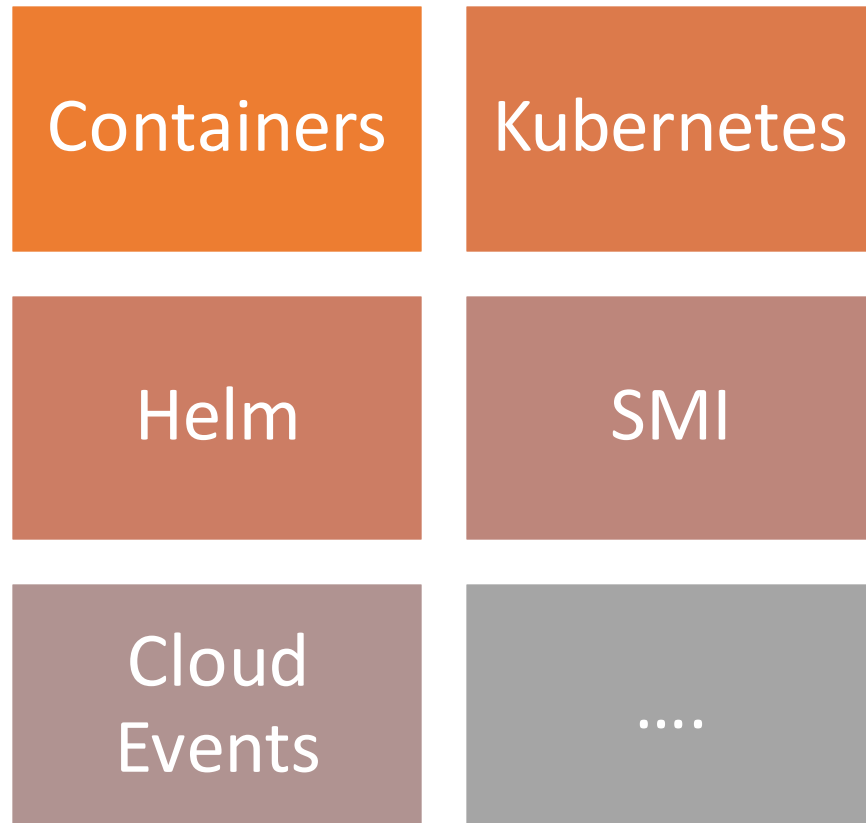
MongoDB

CosmosDB



How do I build applications in  
this world?

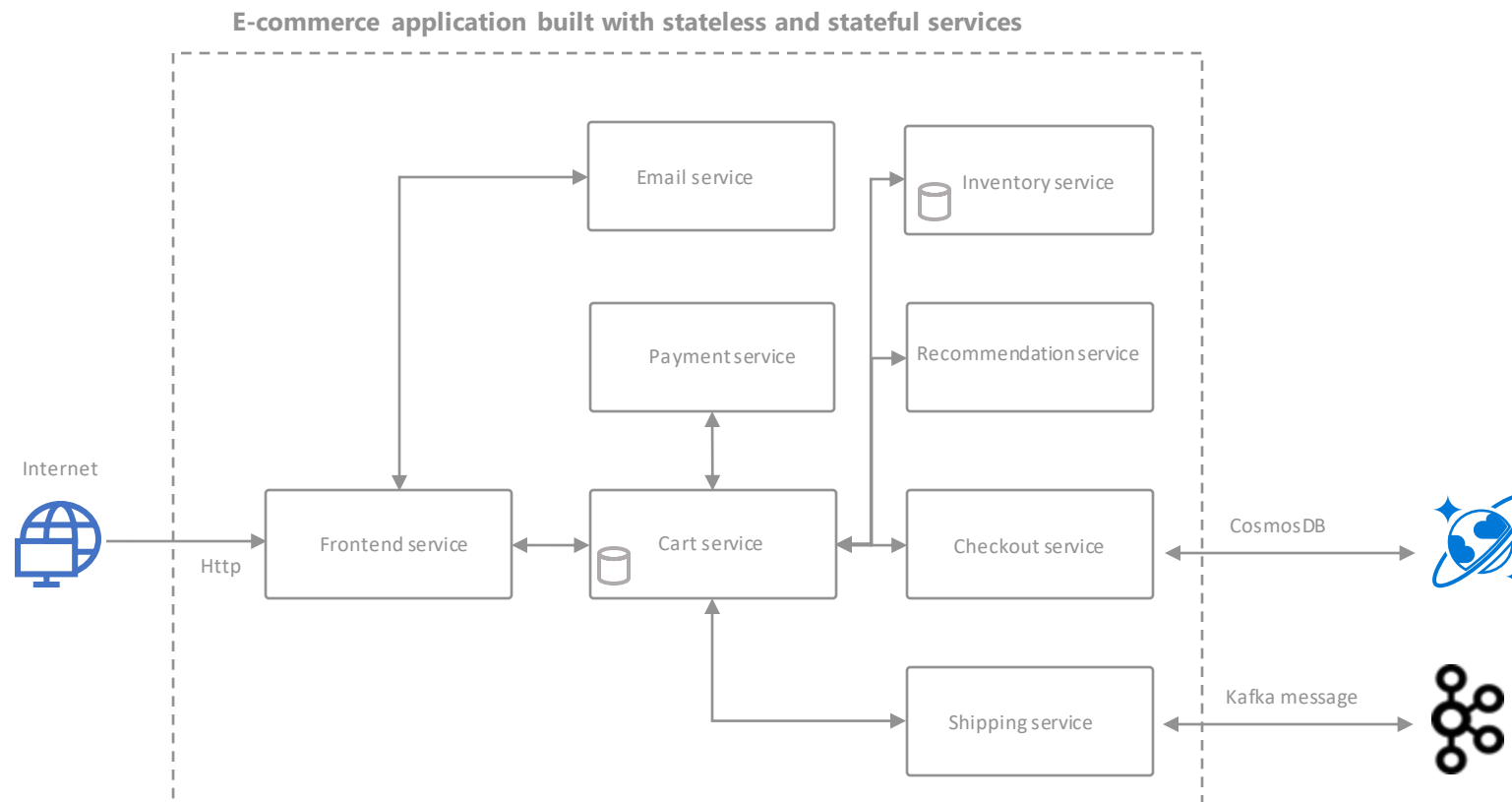
# CNCF is the foundation



Add -  
A programming model

# What is a programming model?

- The way developers write their application that interacts with other services and data stores.
- Increasingly polyglot, with microservice architecture



# Dapr: Build apps using any language with any framework

Microservice application

Services written in

Any code or framework...



node.js

python™

.NET Core



Functions



HTTP/gRPC APIs

Service-to-service invocation

State management

Publish and subscribe

Resource bindings & triggers

Actors

Distributed tracing

Extensible...



Any cloud or edge infrastructure

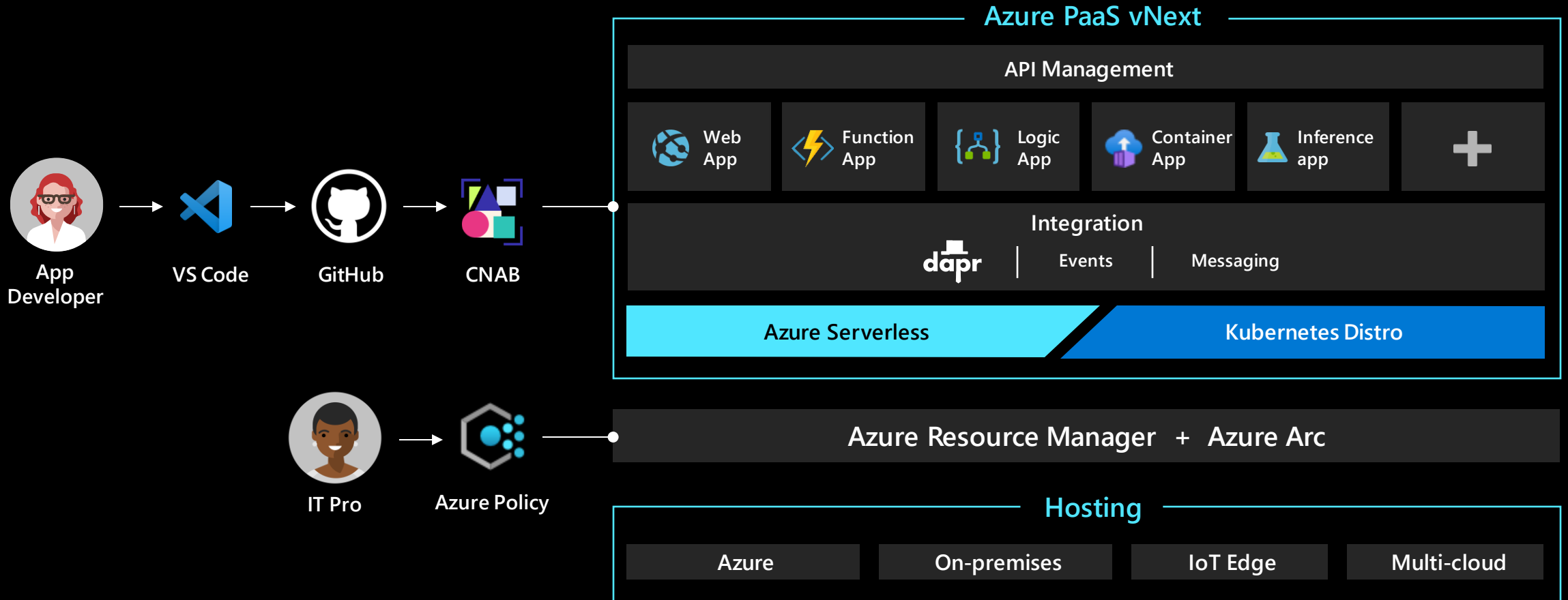


What about gaps such as flow  
architecture?

---



# PaaS vNext approach

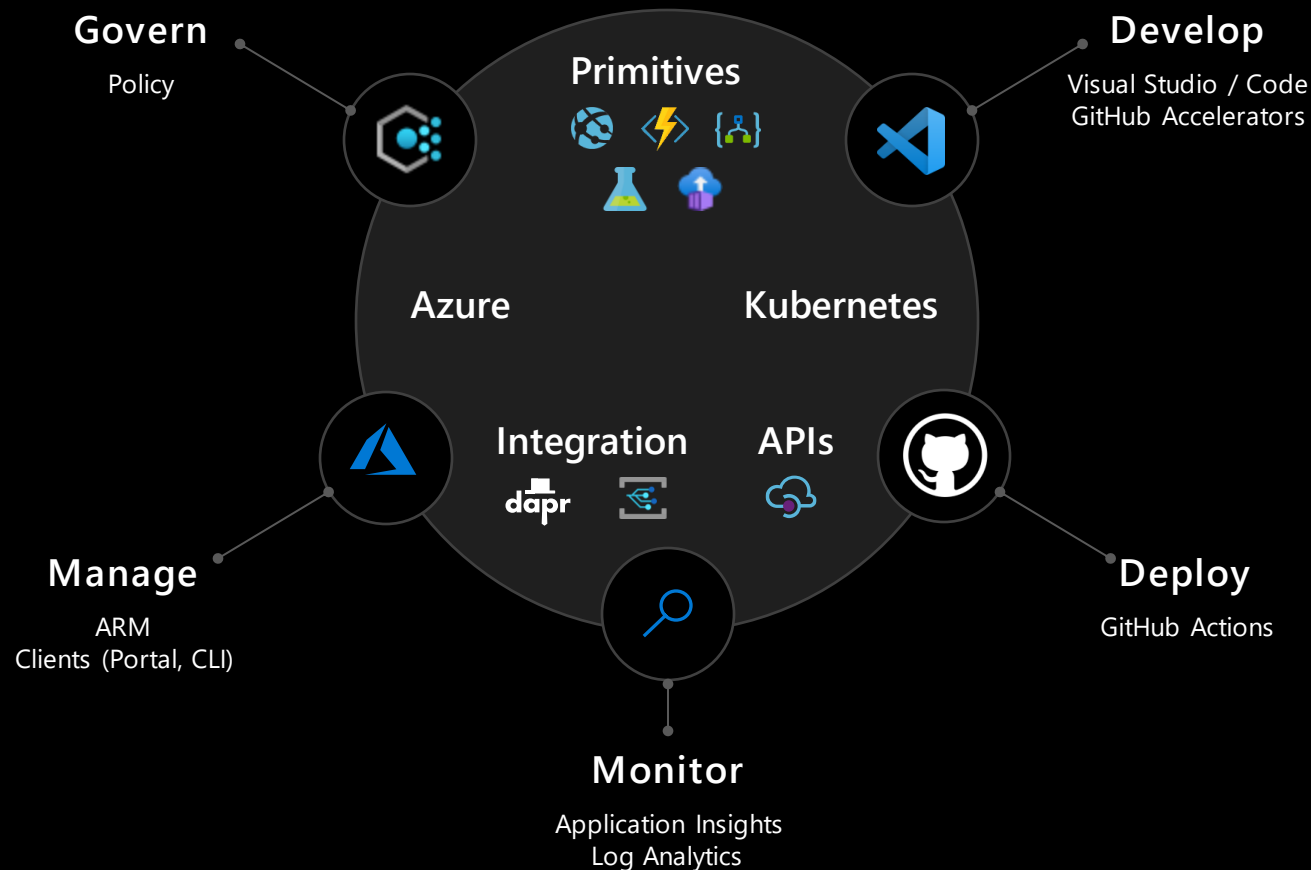


# Arc-enabled developer services

Primitives for specific use case

Consistency across developer and operator lifecycle

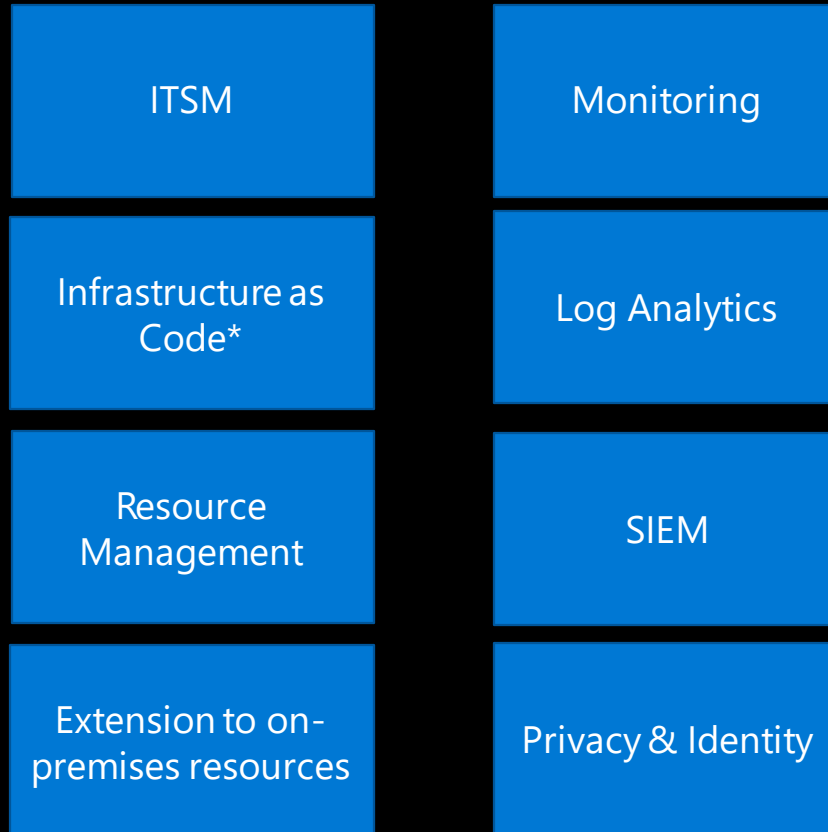
Integrates with common services



How do I operate these clouds  
then?

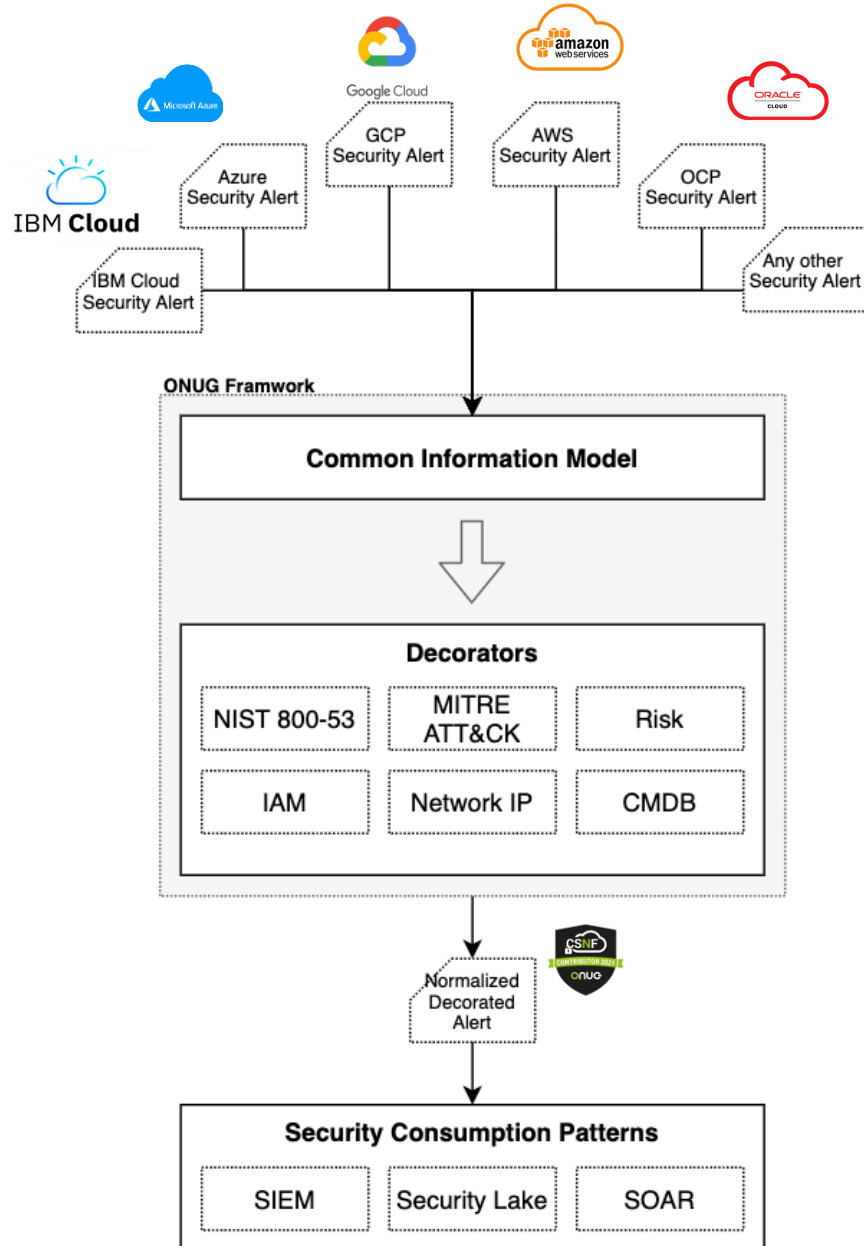
---

# End-to-end multi-cloud management (not exhaustive)



\* consider using a formal source code management environment for your infrastructure as code repository

**Some open community initiatives**



## Today:

CSP's all provide semantically equivalent elements, but elbow grease is required to get them into a common format suitable for downstream processing

## Common Information Model:

We are asking CSP's provide alerts using an open and common set of elements that will remove ambiguity and allow enterprise Cloud customers to process security alerts from all CSP's using a common data pipeline

## Decorator:

Alerts received through batch or stream processing will then be decorated or 'enriched' providing the the customer with the ability to provide additional context in order to speed decision making and automation across large data streams

## Decorated Alert Consumption:

Decorated alerts are received and processed by the Cloud customer to answer questions and report on trends that are most important to them. Deep insights can be applied to identify anomalous activity, find indicators of compromise, or update risk scores across all Cloud provider workloads

# Delivering upon operations – current options:

- Build your own – service now, Splunk, data dog, etc.

Vendor  
dependency:

---

VMWare Tanzu  
Mission control

---

IBM Satellite

---

Google Anthos

---

Azure Arc

---



## Single control plane with **Azure Arc**

**Azure Arc enabled infrastructure**  
Connect and operate hybrid resources  
as native Azure resources

**Azure Arc enabled services**  
Deploy and run Azure services outside of  
Azure while still operating it from Azure

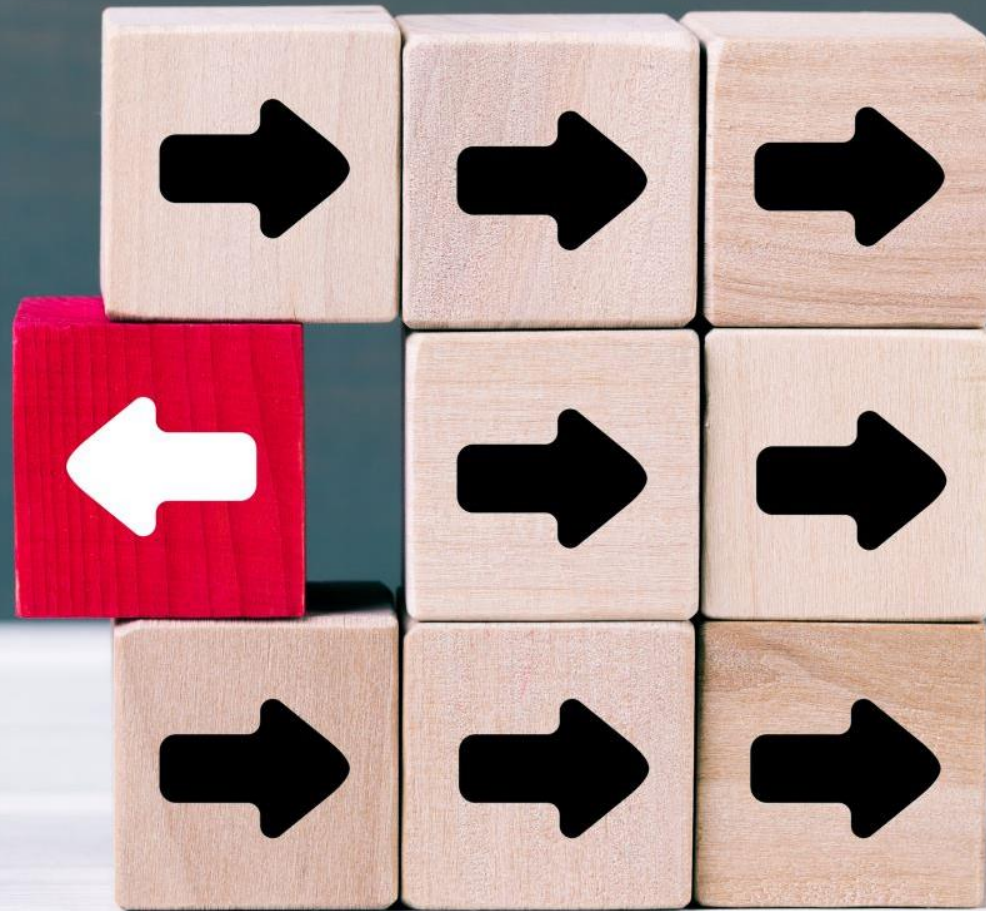


  
Multi-cloud

  
Datacenter

  
Edge





**End-to-end management approach**

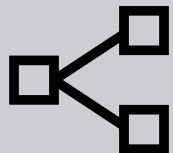
# Proven practices (few but important)



*"Static"* configuration such as VPC/VNET or similar structures should be cloud-native per environment using pre-designed IP per cloud environment



Daily operations tools need to be able to cover multi-cloud scenarios



What is your center of gravity for operations?

Identity? Security? Monitoring?  
Backup? Logging?

# End-to-end multi-cloud management

ITSM	ServiceNow or equivalent	Monitoring	De-Facto standards such as Prometheus Azure Monitor (x-platform   x-environment) DataDog or equivalent
Infrastructure as Code*	TerraForm or equivalent For DevOps scenarios consider using technologies such as <a href="#">GitHub Actions</a>	Log Analytics	Splunk or similar Azure LogAnalytics (x-platform   x-environment)
Resource Management	Cloud-native resource management API, e.g. ARM for Azure	SIEM	Qradar and equivalent (not built for cloud) Azure Sentinel (x-platform   x-environment)
Extension to on-premises resources	Azure Arc or equivalent	Privacy & Identity	Standards (Oauth, etc.) Azure Active Directory, Okta, or equivalent

\* consider using a formal source code management environment for your infrastructure as code repository

# Things to worry about...



Testing for compatibility across cloud providers



Management



Billing



Costs



Skills across cloud

