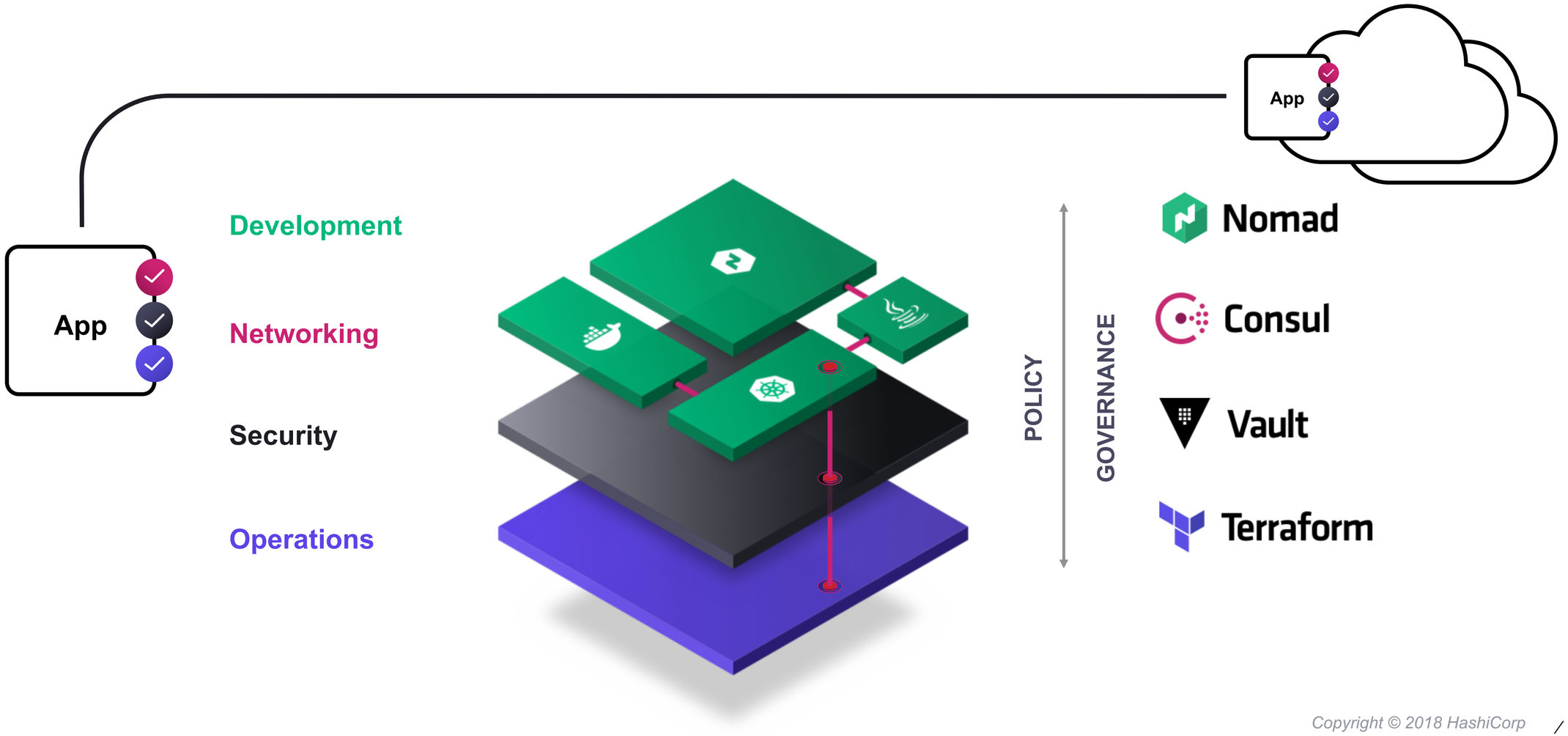
Consul Service Networking Made Easy

A multi-cloud service networking platform to connect and secure any service across any runtime platform and public or private cloud

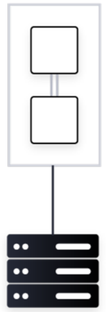
The Cloud Operating Model

A Common Cloud Operating Model to Accelerate Application Delivery

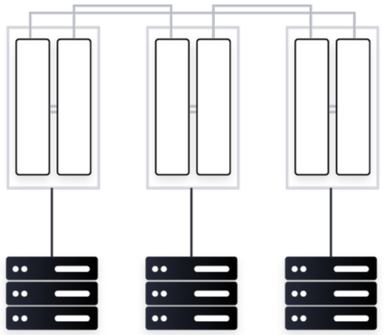


Shifting from Static to Dynamic Networking

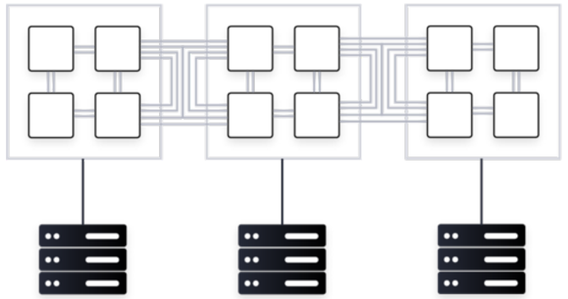
Market trend from monoliths to microservices



Single,
Physical Server



Dynamic Virtual
Machines



Smaller, Ephemeral
Containers

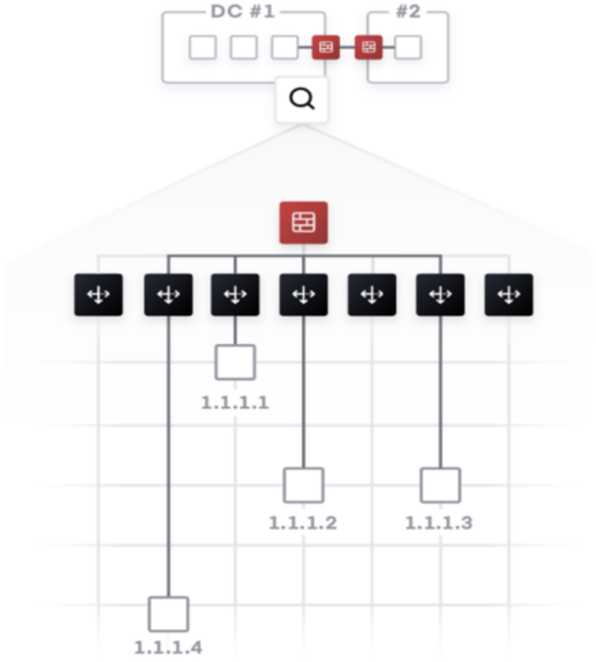
The shift **from static to dynamic networking**

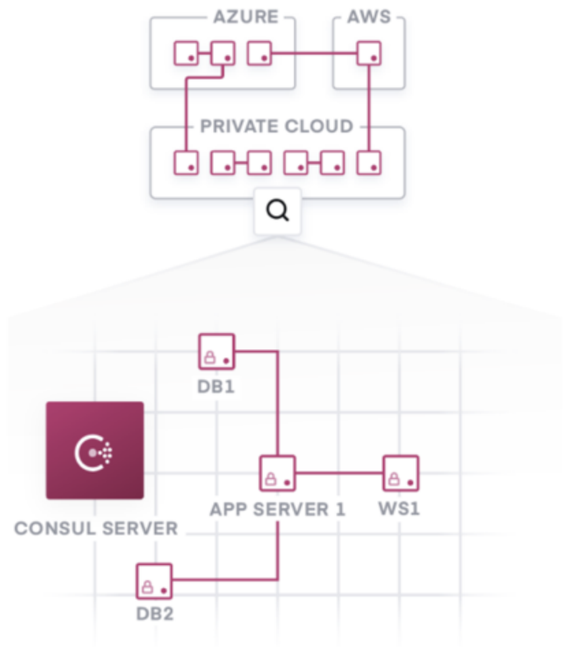
Static Infrastructure

Host-based networking

Dynamic Infrastructure

Service-based networking





The shift **from static to dynamic networking**

Static Infrastructure

Host-based networking

Private datacenters with static IPs, primarily north-south traffic, protected by perimeter security and coarse-grained network segments.

Traditional Approach

- Static connectivity between services
- A fleet of load balancers to route traffic
- Ticket driven processes to update network middleware
- Firewall rule sprawl to constrict access and Insecure flat network zones

Dynamic Infrastructure

Service-based networking

Multiple clouds and private datacenters with dynamic IPs, dominated by east-west traffic, no clear network perimeters.

Consul Approach

- Centralized registry to locate any service
- Services discovered and connected with centralized policies
- Network automated in service of applications
- Zero trust network enforced by identity-based security policies

Three Competencies of a Service Mesh

Networking

Dynamically locate any application or infrastructure service to simplify network connectivity

Observability

Increased visibility to gain insights into how services interact with and depend on each other at the runtime layer

Reliability

Building redundancy through automated failover and replication to minimize outages and downtimes

Demo

Demo

1. **Provision AKS Cluster Using Terraform**

2. Deploy Consul to cluster via Helm

3. Deploy Application to cluster

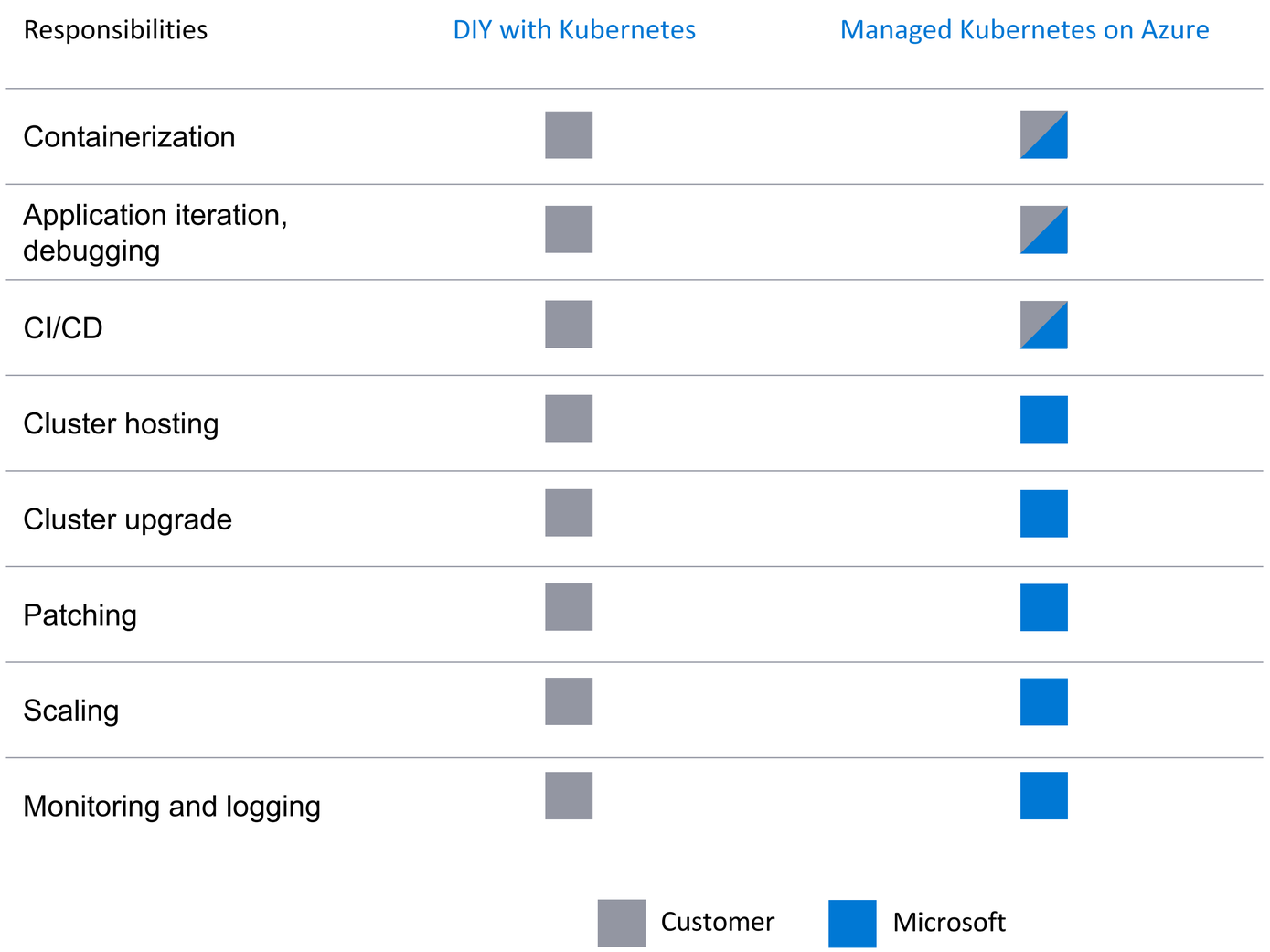
4. Use SMI to manage intentions

Kubernetes on Azure Overview

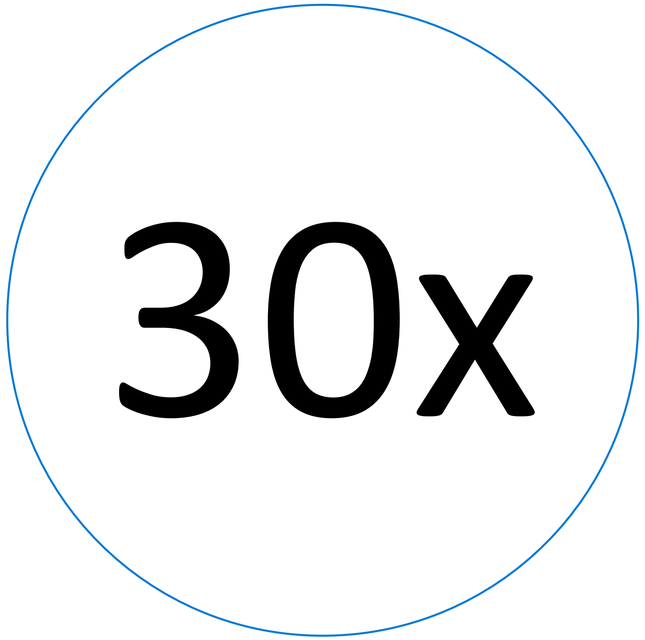
From infrastructure to innovation

Managed Kubernetes empowers you to achieve more

Focus on your containers and code, not the plumbing of them



Azure Kubernetes momentum



Azure Kubernetes Service
usage grew 30x since it was made
generally available in June 2018

Trusted by thousands of customers

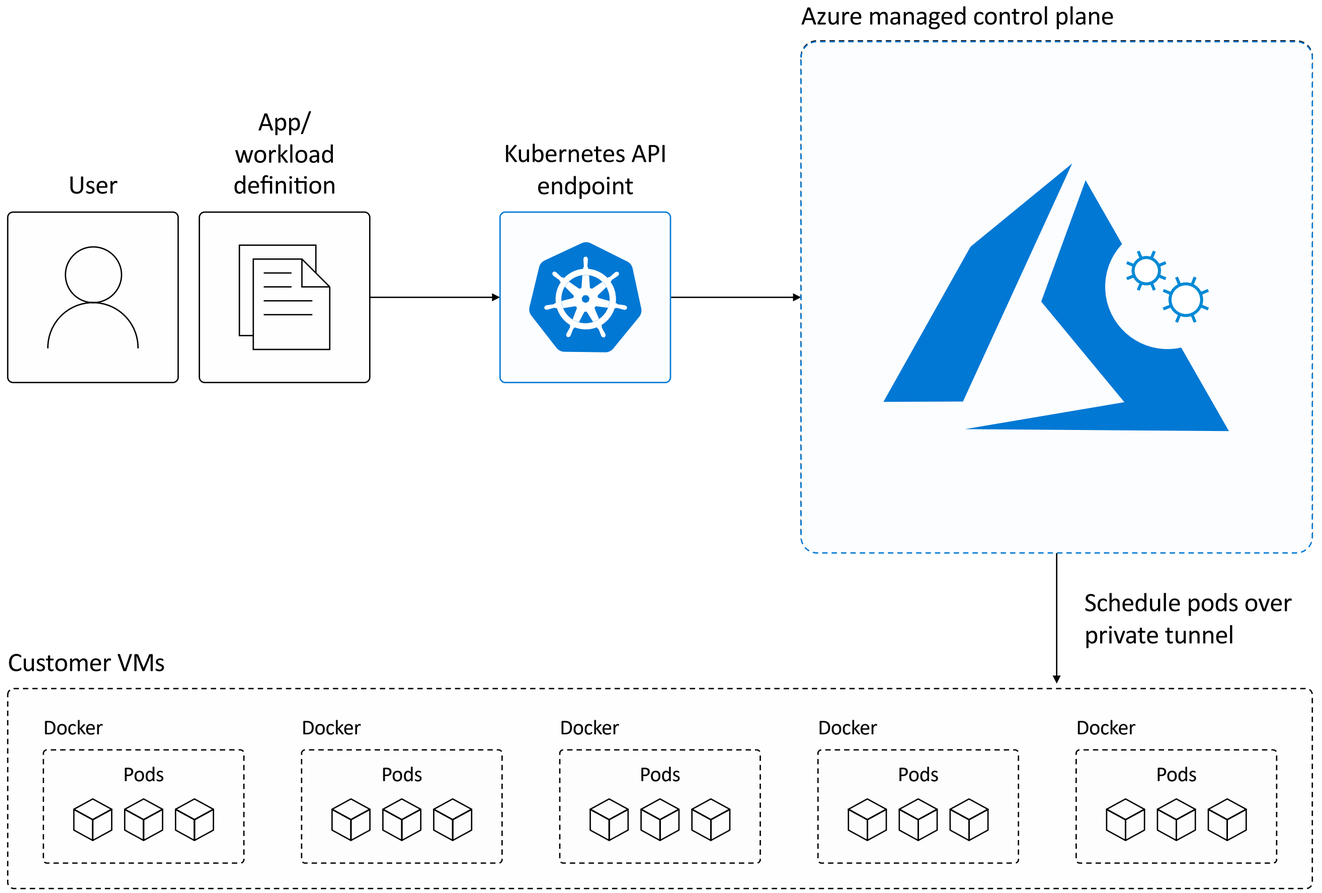
J.B. HUNT
Hafslund
MAERSK
AMBITENERGY
webjet.com.au
WhiteSource
falkonry
Aporeto
Altair
SOTETSU
xerox
SIEMENS Healthineers
varian
NAMOGOO
きりんカルテシステム株式会社
Tryg
BOSCH
ELASTACLOUD
Munich RE
welt hunger hilfe
FINASTRA
equinor
nyris

Dated November 2018

Manage Kubernetes with ease

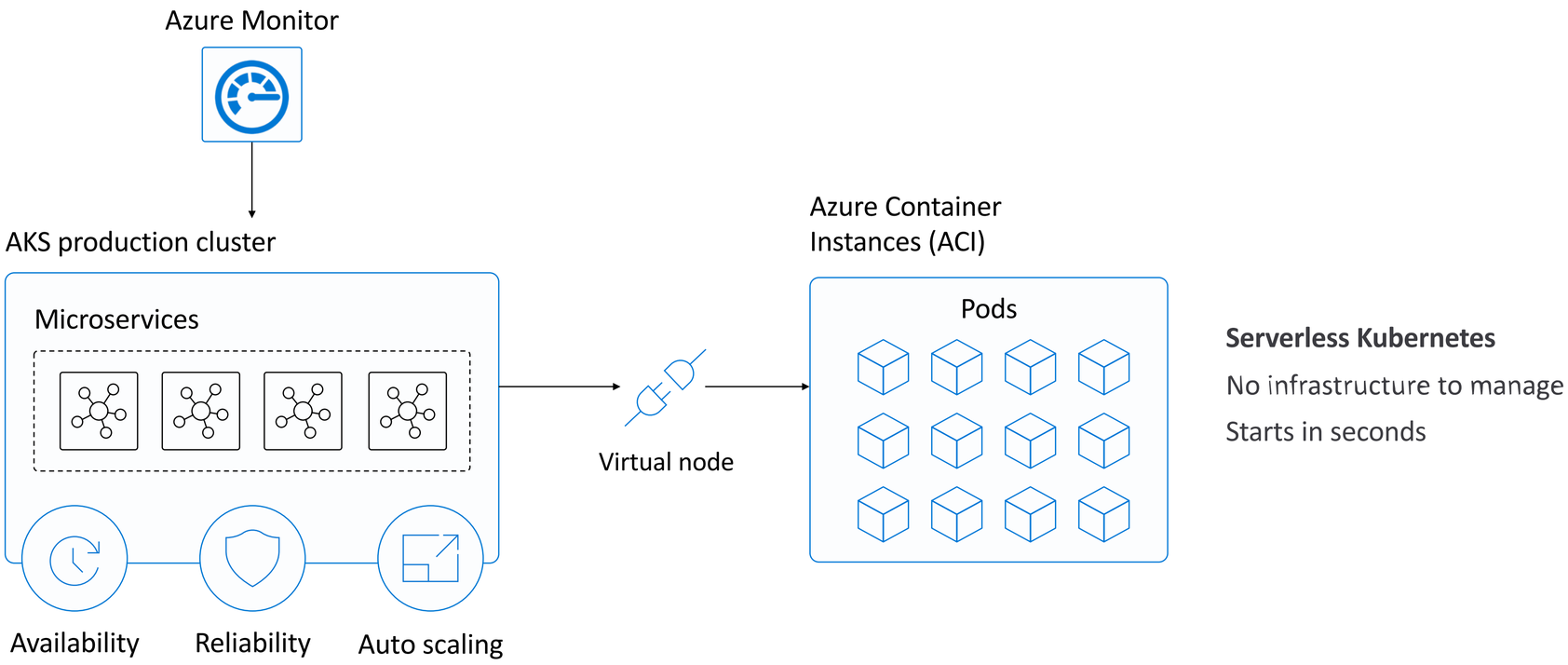
Infrastructure automation

- Automated provisioning, upgrades, patches
- High reliability, availability
- Easy, secure cluster scaling
- Self-healing
- API server monitoring
- At no charge



Manage Kubernetes with ease

Highly available, reliable service with serverless scaling



Serverless Kubernetes

No infrastructure to manage

Starts in seconds

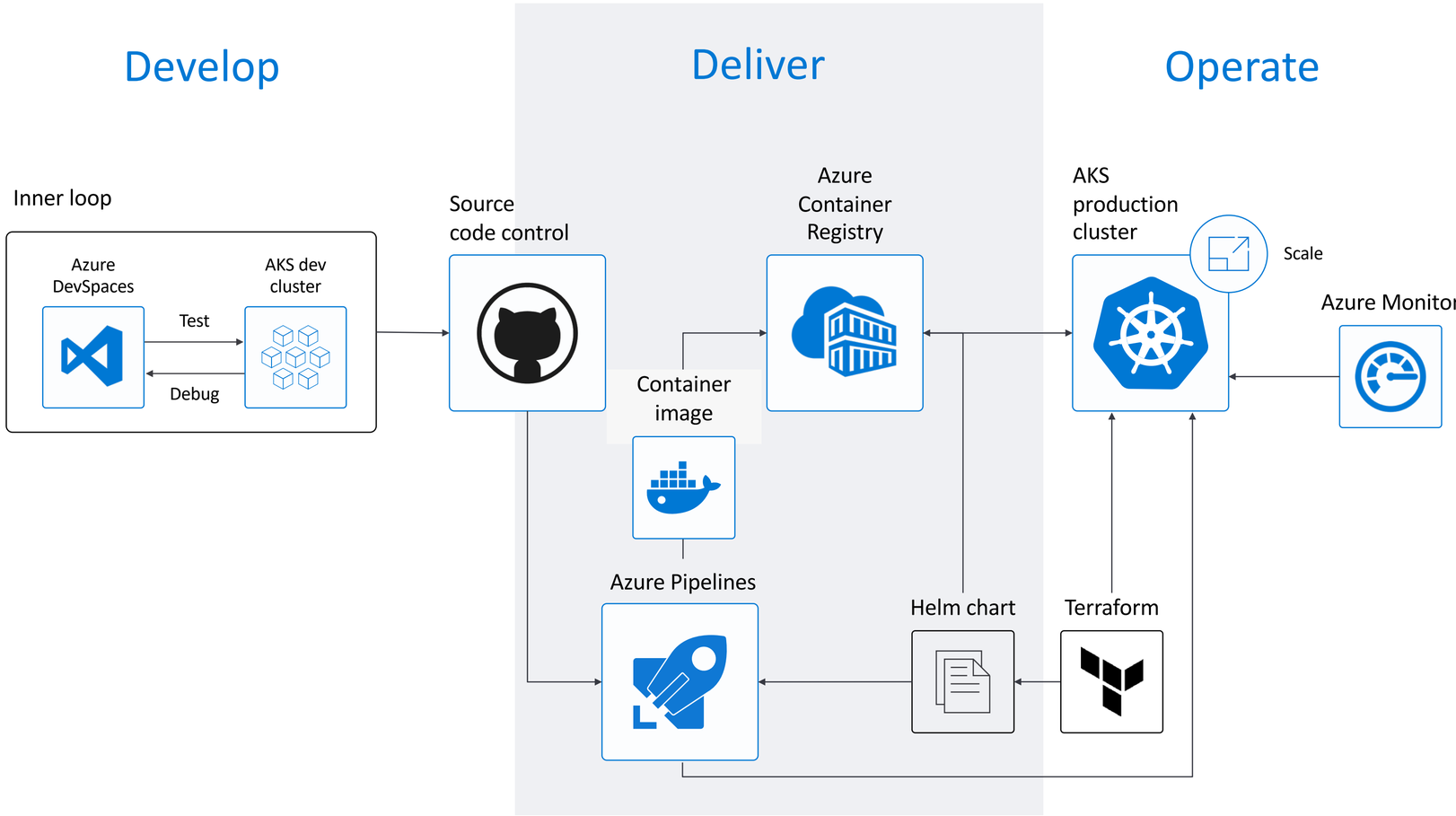
Accelerate containerized development

Kubernetes and DevOps better together

Develop
- Native containers and Kubernetes support in IDE
- Remote debugging and iteration for multi-containers
- Effective code merge
- Automatic containerization

Deliver
- CI/CD pipeline with automated tasks in a few clicks
- Pre-configured canary deployment strategy
- In depth build and delivery process review and integration testing
- Private registry with Helm support

Operate
- Out-of-box control plane telemetry, log aggregation, and container health
- Declarative resource management
- Auto scaling



CONSUL ADOPTION / / PHASE TWO

Demo

Demo

1. ~~Provision AKS Cluster Using Terraform~~

2. **Deploy Consul to cluster via Helm**

3. Deploy Application to cluster

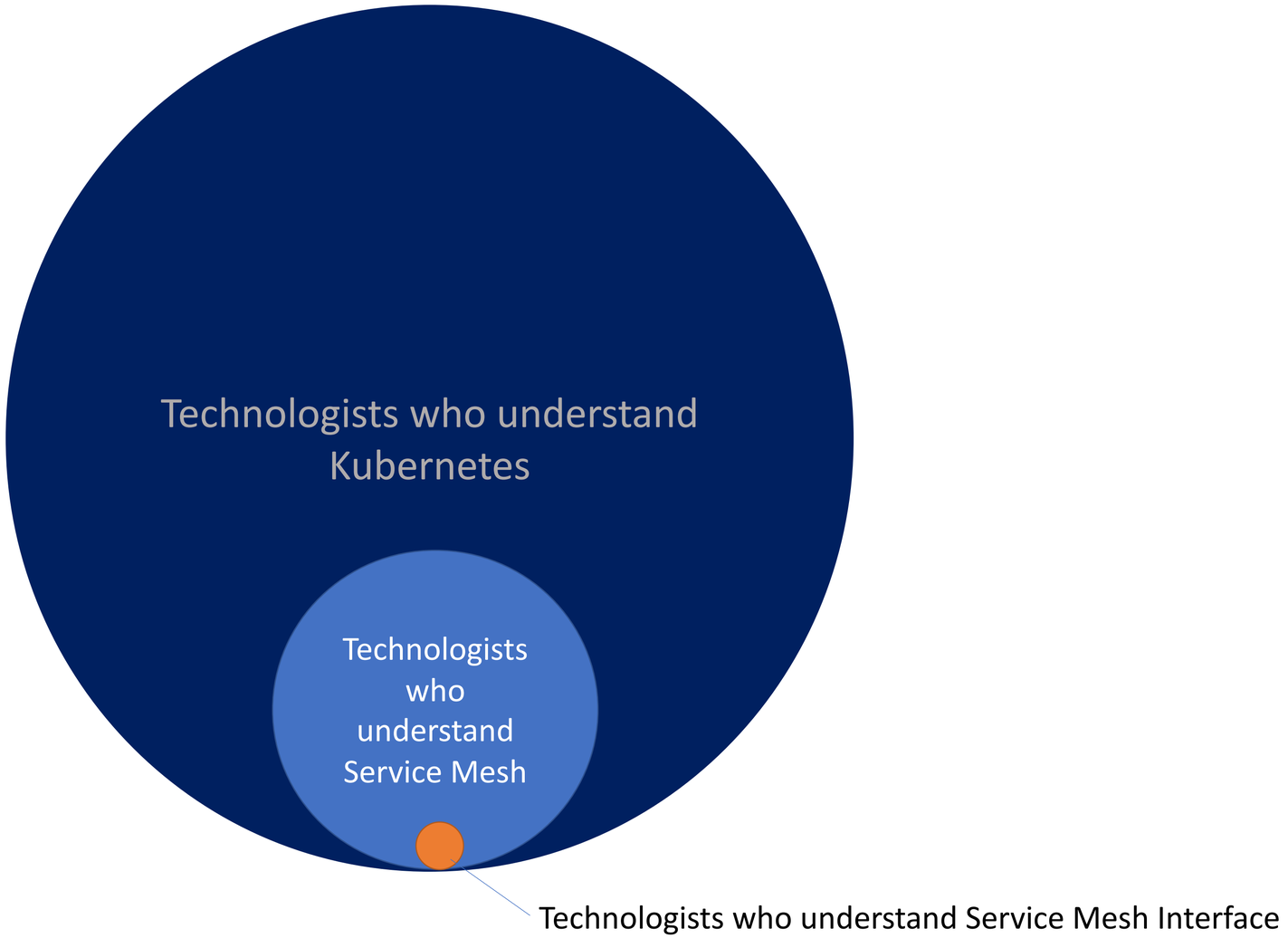
4. Use SMI to manage intentions

Demo

1. ~~Provision AKS Cluster Using Terraform~~
2. ~~Deploy Consul to cluster via Helm~~
3. **Deploy Application to cluster**
4. Use SMI to manage intentions

Microsoft Service Mesh Interface (SMI)

Towards Service Mesh Interface



But that shouldn’t really be the case

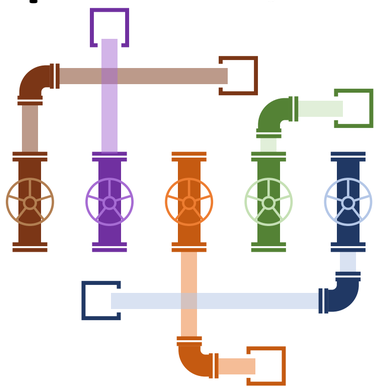
Technologists who understand Kubernetes expect a set of networking functionality available to them

Services should have an identity and should be able to communicate securely

Communication between services should be monitored and logged

Traffic patterns between services should be configurable, i.e., newer service versions could be receiving less traffic, etc.

Historically, this was achieved via smart endpoints, dumb pipes

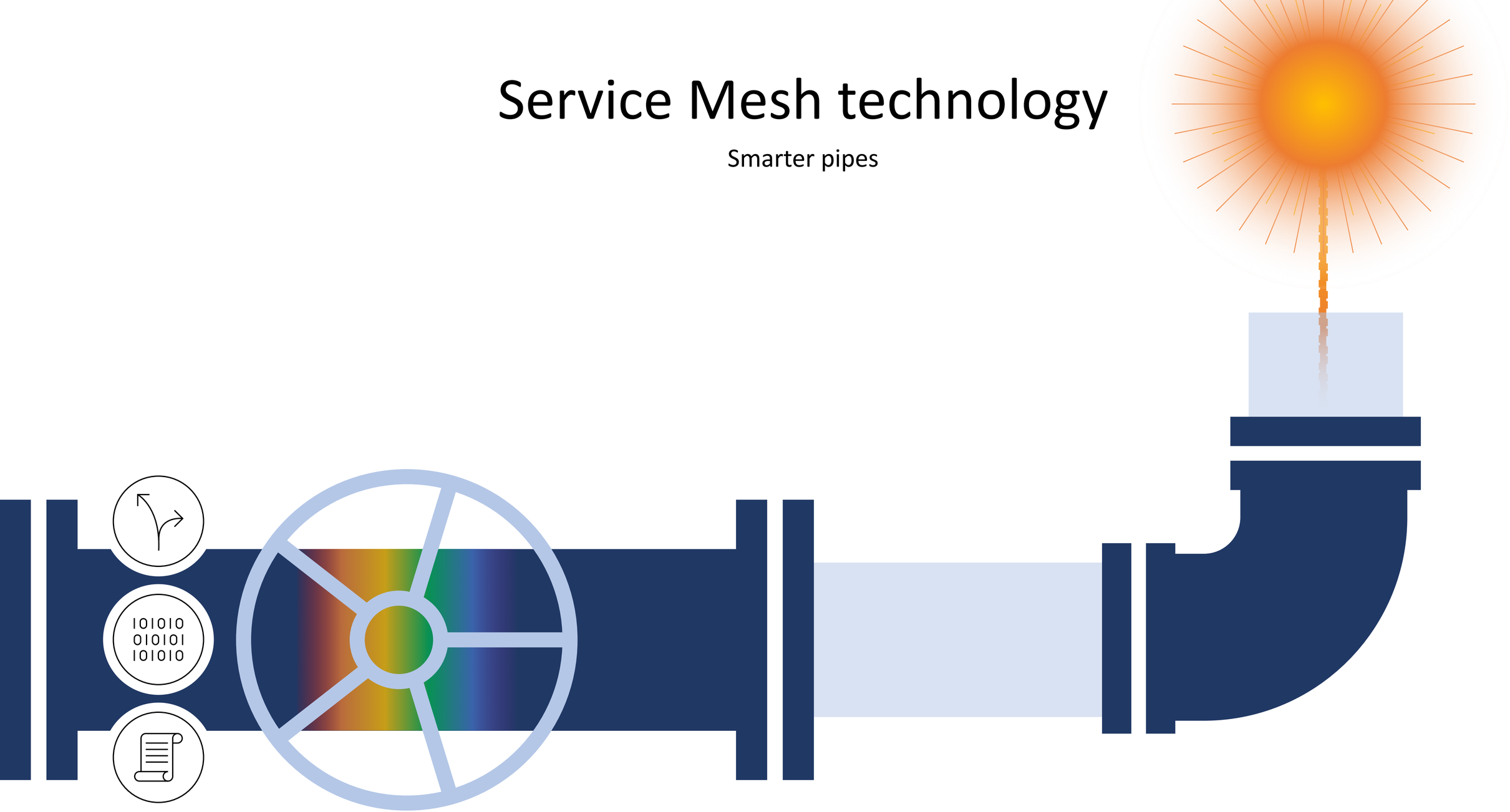


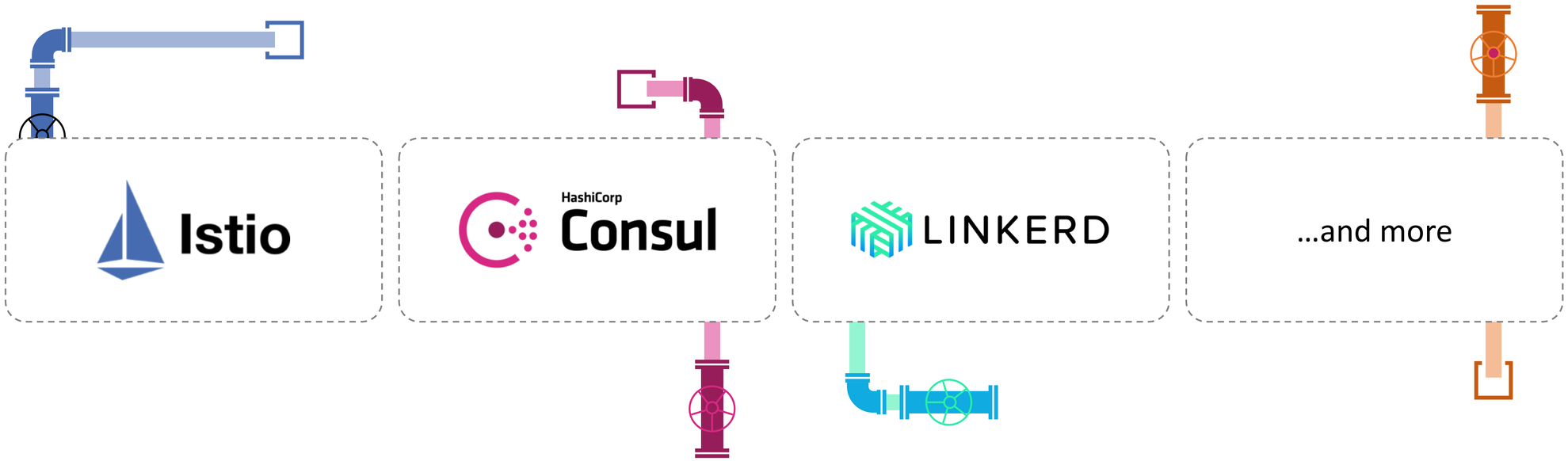
This has worked for the past 25 years

But with so many endpoints today, how do you manage

Service Mesh technology

Smarter pipes




Istio
HashiCorp
Consul
LINKERD
...and more

Service Mesh Interface (SMI) for Kubernetes

In partnership with

Service Mesh Interface (SMI) for Kubernetes

A Kubernetes interface that provides traffic routing, traffic telemetry, and traffic policy

Standardized

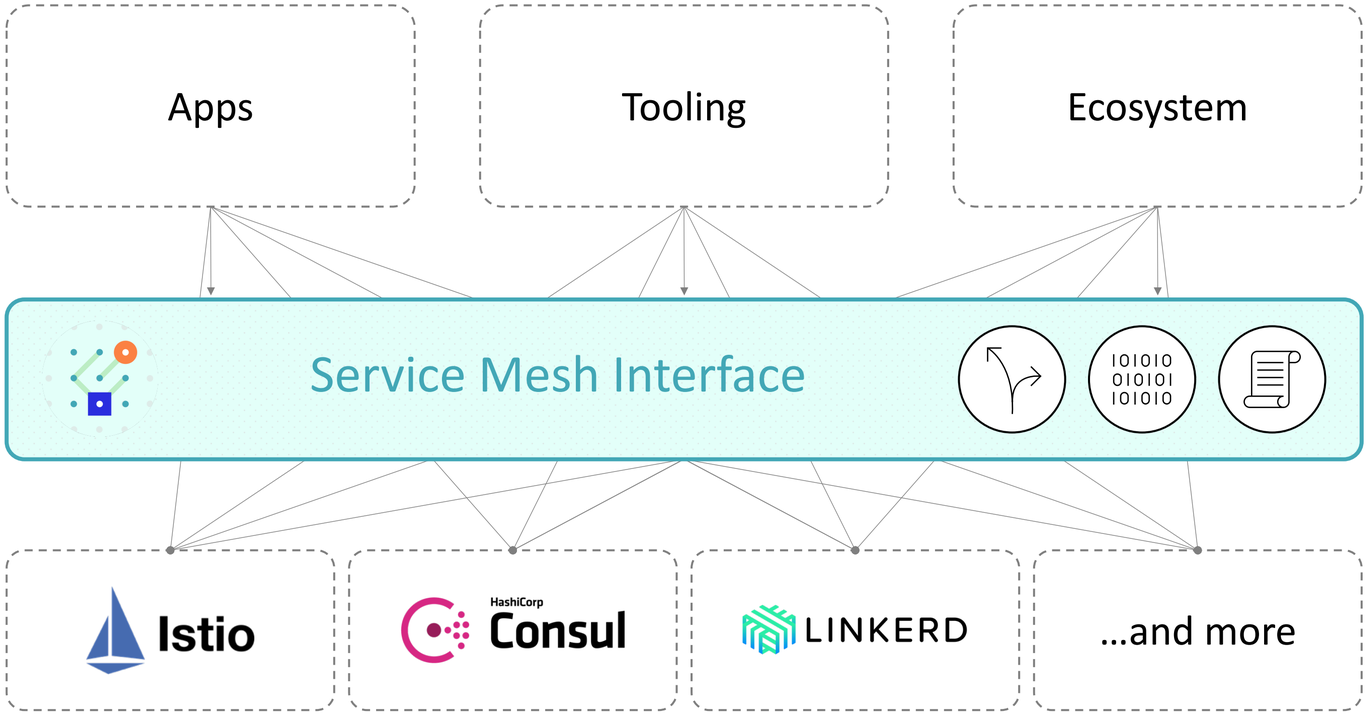
Standard interface for
service mesh on Kubernetes

Simplified

Basic feature set to address most
common scenarios

Extensible

Support for new features as they
become widely available



Microsoft Azure

Service Mesh Interface

Technologists who understand Kubernetes expect a set of networking functionality available to them, and now they can get that functionality through SMI

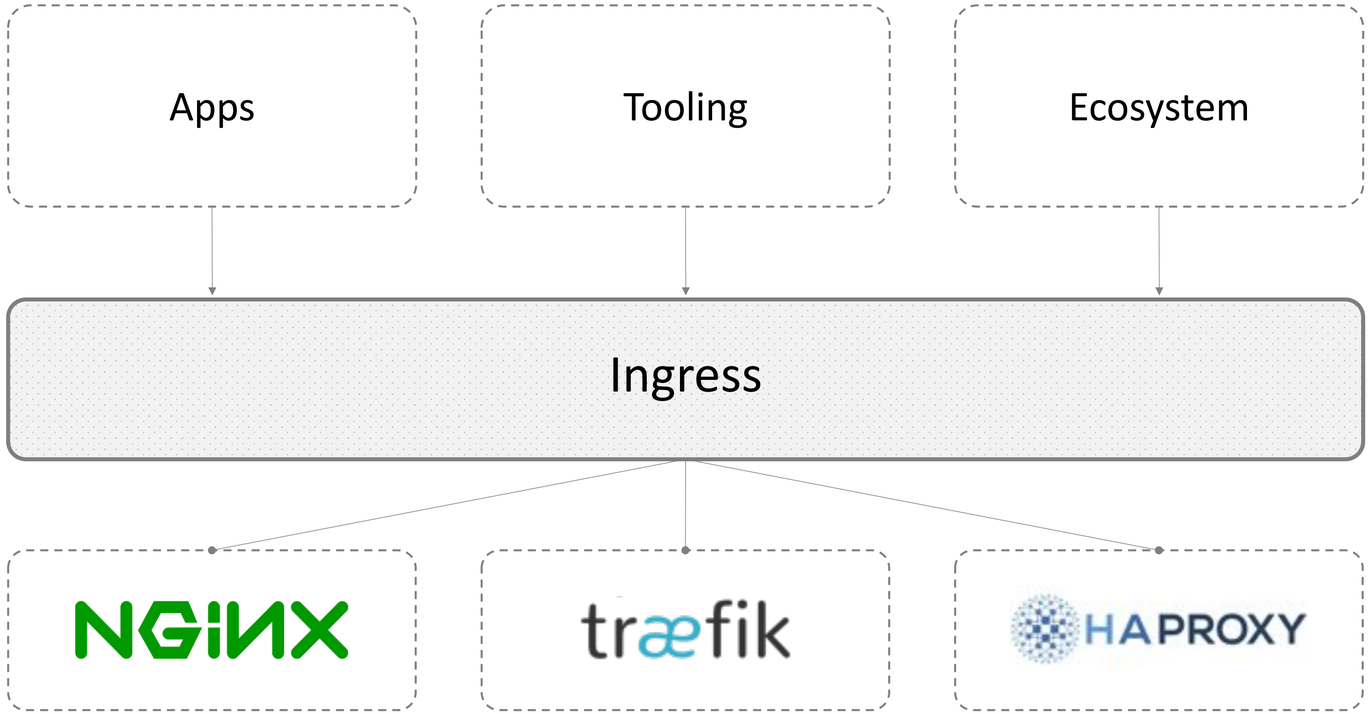
Services should have an identity and should be able to communicate securely

Communication between services should be monitored and logged

Traffic patterns between services should be configurable, i.e., newer service versions could be receiving less traffic, etc.

This isn’t a new concept

If the SMI concept sounds familiar, that’s because it is



So, why SMI?

Because it’s faster, simpler, and friendlier

Get started quickly

Simpler is better

Ecosystem friendly

CONSUL ADOPTION / / PHASE TWO

Demo

Demo

1. ~~Provision AKS Cluster Using Terraform~~
2. ~~Deploy Consul to cluster via Helm~~
3. ~~Deploy Application to cluster~~
4. **Use SMI to manage intentions**

New Features
L7 Observability, Routing, and Mesh Gateways

Feature: **L7 Observability**

CHALLENGE SOLUTION RESULTS

Debugging and isolating issues in a microservices environment can be challenging

A primary microservices challenge is trying to understand how individual services being independently deployed are interacting with each other.

Lack of visibility to get insights into the way services interact with and depend on each other at runtime makes it difficult to understand the performance bottlenecks and isolate any issues or incidents

Feature: **L7 Observability**

CHALLENGE **SOLUTION** RESULTS

L7 Observability

Consul Connect lets users configure the observability features in Envoy proxy.

Observability makes it easier to see what is happening when services interact with each other.

Envoy also provides a simple way to get the same high-level metrics for all services.

Feature: **L7 Observability**

CHALLENGE **SOLUTION** RESULTS

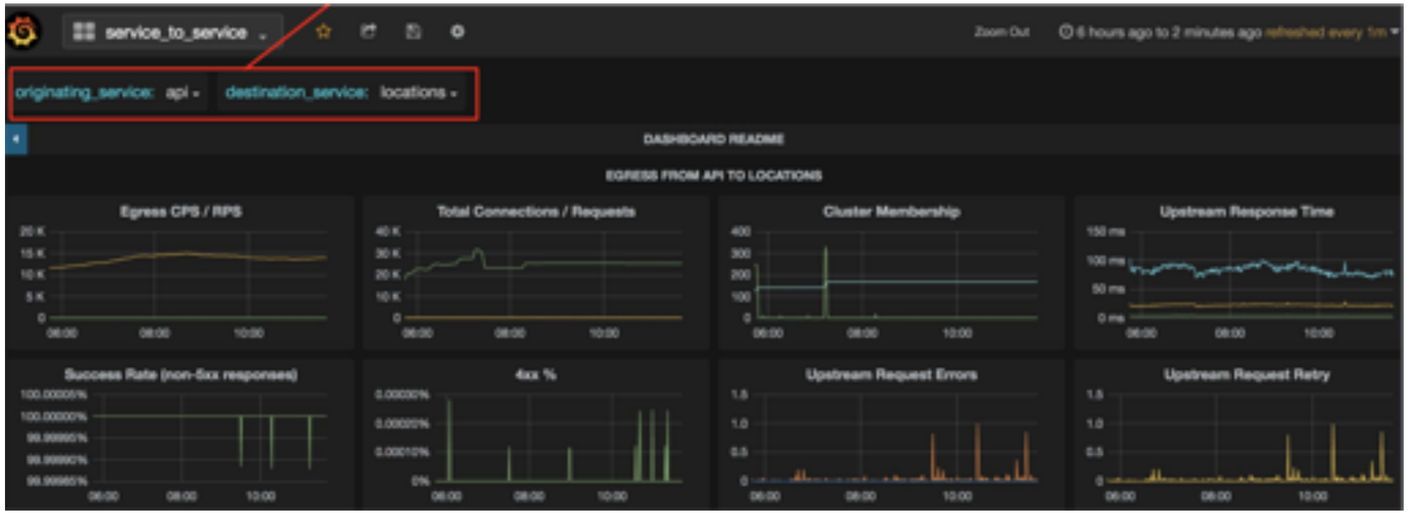
L7 Observability

Observability features include:

Tracing: Helps with understanding the application behavior when interacting with upstream and downstream services

Metrics: Gathers telemetry data from across the mesh and provide high-level application information

Logging: Provides an immutable record of discrete events that helps in better fault isolation



Feature: **L7 Observability**

CHALLENGE

SOLUTION

RESULTS

L7 Observability

Observability is achieved by integration with third party tools :

Tracing: Jaeger, Zipkin, DataDog

Metrics: Grafana, Prometheus, etc

Logging: Splunk, ELK, Fluentd

Feature: **L7 Routing**

CHALLENGE SOLUTION RESULTS

Consul Connect only supports L4 proxy

Consul Connect can route traffic and set policies based on layer 4 headers,

It does not support routing and redirecting requests based on L7 path, content type, runtime values, etc.

Lack of L7 filtering capabilities also prevents it from enforcing fine-grained security policies based on rich HTTP headers

Feature: **L7 Routing**

CHALLENGE SOLUTION RESULTS

L7 Routing and Policies

Ability to steer traffic based on header for canary testing, gradual deployment roll outs and canary releases

Feature: **Mesh Gateways**

CHALLENGE SOLUTION RESULTS

Cross service connections across multi-K8s clusters or multi- clouds is challenging

Consul expects all the services to be on a flat network and routable

Consul cannot solve service discovery and connectivity use cases in overlay networks

Connecting services across multi-clouds requires expensive IPsec or MPLS connectivity

Feature: **Mesh Gateways**

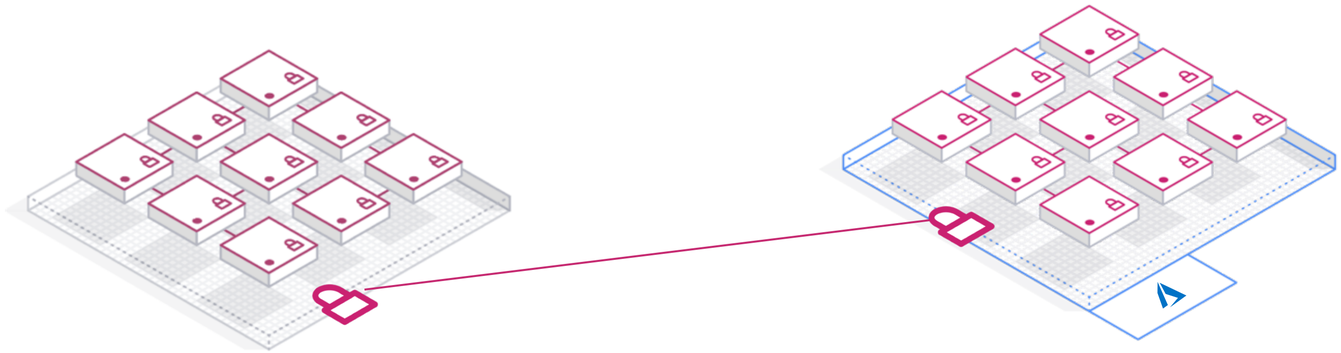
CHALLENGE **SOLUTION** RESULTS

Network Gateways

Network gateway, built upon Envoy, will sit on the public internet and accept L4 traffic with mTLS

Network gateways will perform NAT and route the traffic to correct endpoint on the private network

All the services need not be exposed on public network for cross cloud service communication



Thank you

hello@hashicorp.com

www.hashicorp.com