CITRIX

CLOUD NATIVE
COMPUTING FOUNDATION

How to Choose the Right Proxy Architecture for Microservices-Based Application Delivery

Pankaj.Gupta @Citrix.com
Mikko.Disini @Citrix.com

August 27, 2019

As an active member of the Cloud Native Computing Foundation, Citrix is proud to present this webinar with the support of and in association with CNCF

CLOUD NATIVE COMPUTING FOUNDATION

CITRIX®

Agenda

- Importance of choosing the right architecture
- Quick recap of L4 & L7 load balancing
- 4 architecture options
- Deep dive into each architecture: 7 attributes
- Citrix solution at a glance

Your Presenters

Pankaj Gupta

Senior Director
Cloud Native Application Delivery @Citrix

A cloud native evangelist, Pankaj advises on product and go-to-market strategies for Citrix application delivery solutions.

Mikko Disini

Director
Cloud Native Application Delivery @Citrix

Mikko leads cloud native product management for Citrix ADC with a focus on production-grade application delivery solutions.

Challenges of Choosing the Right Proxy Architecture

How do you make the best decision for an existing or new business-critical application when you must consider:

- Each stakeholder has unique needs and evaluation criteria: e.g., developer, platform team, networking team, DevOps, SecOps, SRE, app owner
- Load balancing for north-south and east-west (inter microservices) traffic
- The tradeoff between benefits and complexity
- Architectures are complex
- Rapid pace of technology and open source innovation

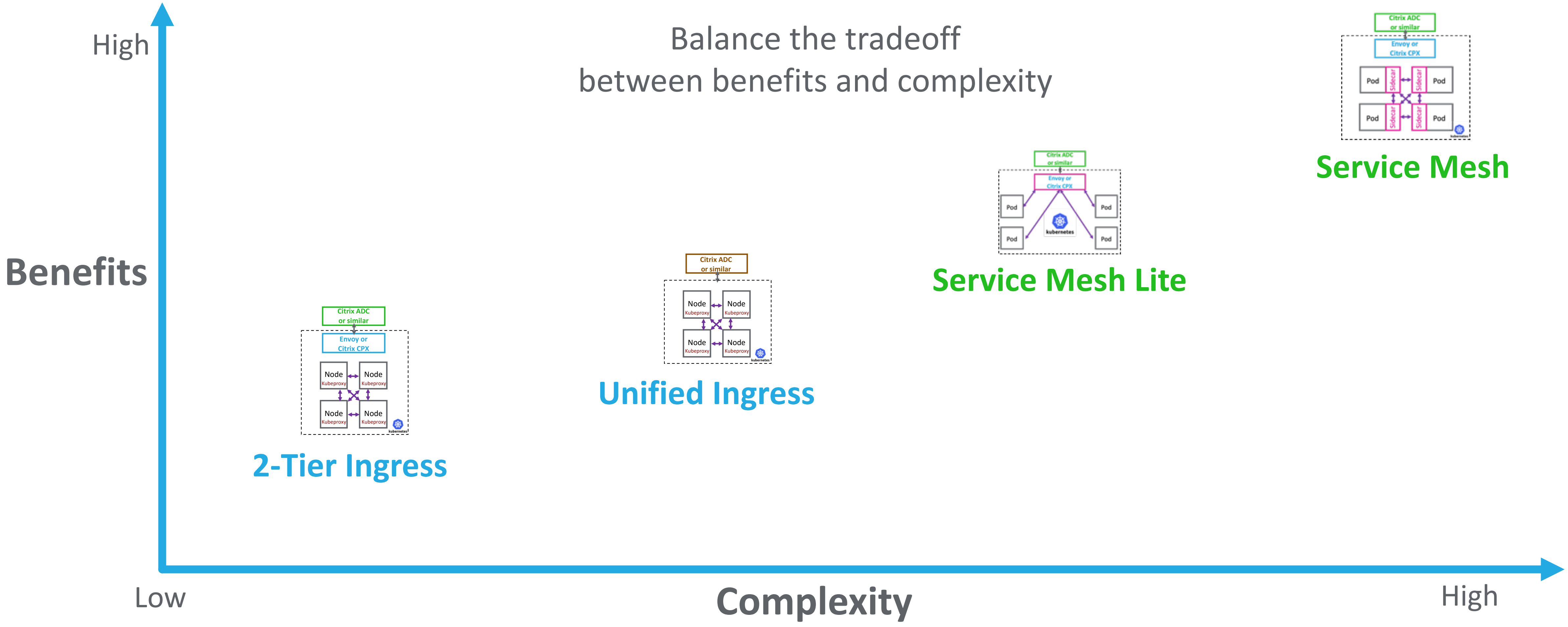
Recap: L4 vs L7 Load Balancing / Traffic Management

	L4	L7
Load Balancing	Basic load balancing • Based on IP address & port only HTTP/S blind No content (payload) rewrite & switching: Inability to change anything on the wire	Advanced load balancing • Based on URL – images, text, video • Based on client information – browser, OS, device, language Takes advantage of the HTTP/S packet info. Designed for apps of today & tomorrow Supports content rewriting: Apps with hard-coded URLs, mergers & acquisitions, publishing internal URLs, misconfigured apps, respond to malicious traffic. Can parse the payload and apply changes; allows making smarter content optimization and security decisions like app firewalling along with doing proxy
Session Persistence	Very limited: Only based on client IP address	Advanced session persistence for better user experience Can use cookies – identify users to provide persistent experience Better experience for stateful applications
Resource Monitoring	Health checking limited to Ping and TCP handshake only	Advanced customizable health checks Application-level visibility for better observability and load balancing decisions Enables circuit-breaking capabilities
App Security	Very limited due to just IP address and port visibility Lacks deep packet inspection	Advanced protection due to deep packet inspection Examples: web application firewall, L7 DoS protection, application stack vulnerabilities based on signature analysis, anomaly detection

Architecture Choices for Microservices-Based Applications

Move to cloud native at your pace

Balance the tradeoff
between benefits and complexity



2-Tier Ingress

Unified Ingress

Service Mesh Lite

Service Mesh

Diverse Stakeholders Have Unique Needs

DevOps

Faster release & deployment cycles
CI/CD & automation
Canary & progressive rollout

Developers

User experience
Troubleshooting
Microservice discovery & routing

Platform Team

Platform governance
Operational efficiency
Developer agility

SRE

Application availability
Observability
Incident response
Postmortems

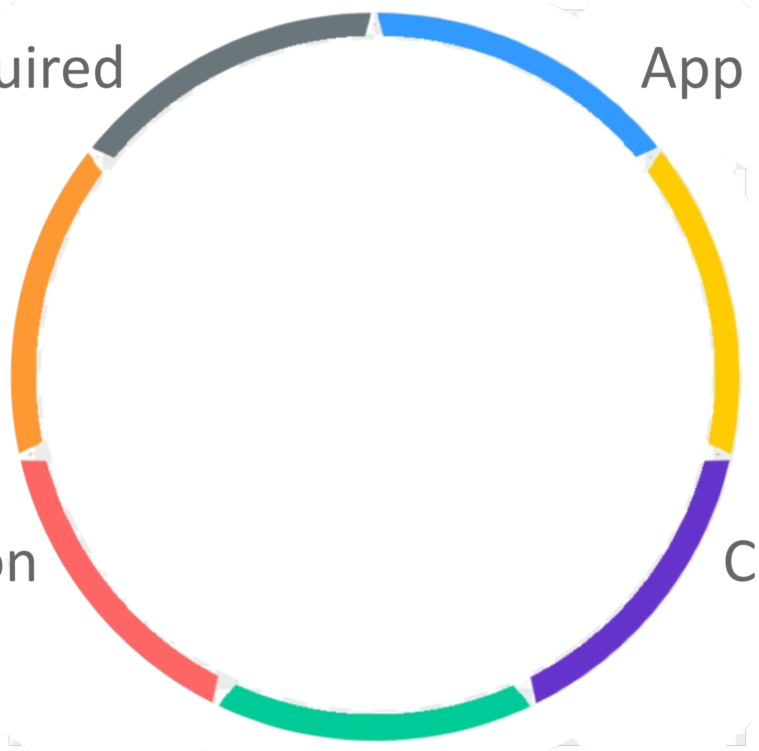
NetOps

Network policy & compliance
Manage, control & monitor network
Resources & capacity planning (visibility)

DevSecOps

Application & infrastructure security
Container security & API gateways
Automation

7 Key Attributes to Evaluate



IT Skill Sets Required

App Security

Istio: Unified Control Plane

Observability

Open Source Tools Integration

Continuous Deployment

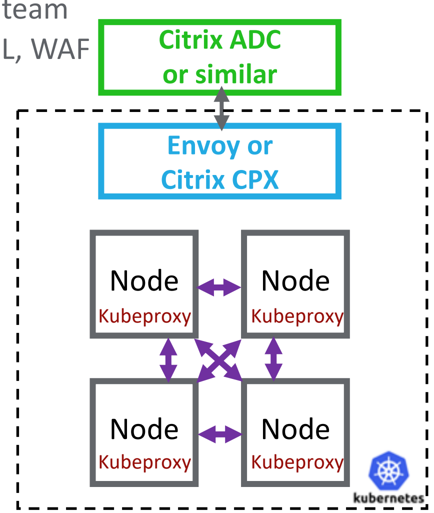
Scale and Performance

2-Tier Ingress: Simplest and Quickest to Production

For both cloud native novices & experts

2-Tier Ingress

- Managed by networking team
- N-S L4 load balancing, SSL, WAF
- Citrix ADC or similar



- Managed by platform team
- N-S L7 load balancing
- Envoy, Citrix CPX, or similar

North-South App Traffic LB

Green ADC for L4 LB for cloud native; L4-7 LB for monolith apps
Blue ADC for L7 LB and faster change of pace

East-West App Traffic LB

Basic layer 4 load balancing (round robin) by Kubeproxy

App Security	N-S: Excellent protection by green ADC E-W: None; need network policy/segmentation, e.g., Project Calico
Observability	N-S: Excellent, green & blue ADC sees all traffic E-W: Very limited telemetry
Continuous Deployment	N-S: Excellent; advanced traffic control by ADC E-W: Lacks due to Kubeproxy limitations
Scale Performance	N-S: Good for scale out E-W: Use IPVS mode; Iptables mode lacks scalability
Open Source Tools Support	N-S: Excellent; e.g., Prometheus, Spinnaker, EFK E-W: Limited due to Kubeproxy limitations
Istio: Unified Control Plane	N-S: Support via Istio-enabled ADCs E-W: Kubeproxy is not Istio enabled
IT Skill Set Required	Minimal training for platform & networking teams Both teams can move at their own speed

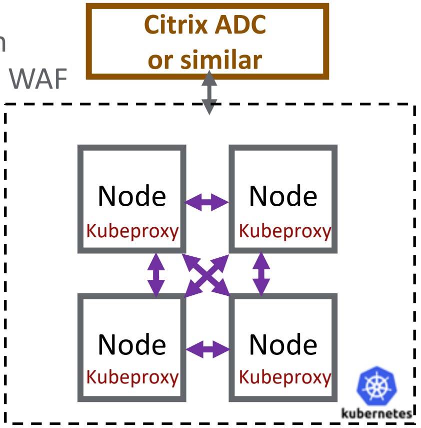
LB = Load Balancing ADC = Application Delivery Controllers

Unified Ingress: Simple for Network-Savvy Platform Teams

Reduce 1 ADC tier and 1 hop latency, suitable for internal apps with option to add WAF/SSL and external apps later

Unified Ingress

- Managed by network-savvy platform/infrastructure team
- N-S L4-7 load balancing, SSL, WAF



North-South App Traffic LB

Brown ADC for L4-7 load balancing for cloud native & monolith apps

East-West App Traffic LB

Basic layer 4 load balancing (round robin) by Kubeproxy

App Security	N-S: Excellent protection by brown ADC E-W: None; Need network policy/segmentation, e.g., Project Calico
Observability	N-S: Excellent; brown ADC sees all traffic E-W: Very limited telemetry
Continuous Deployment	N-S: Excellent; advanced traffic control by ADC E-W: Lacks due to Kubeproxy limitations
Scale Performance	N-S: Good for scale out E-W: Use IPVS mode; Iptables mode lacks scalability
Open Source Tools Support	N-S: Excellent; e.g., Prometheus, Spinnaker, EFK E-W: Limited due to Kubeproxy limitations
Istio: Unified Control Plane	N-S: Support via Istio-enabled ADCs E-W: Kubeproxy is not Istio enabled
IT Skill Set Required	*Platform/infrastructure team needs to be network savvy*

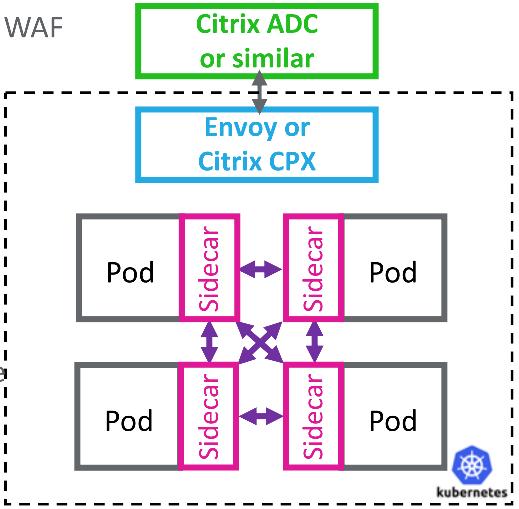
LB = Load Balancing ADC = Application Delivery Controllers

Service Mesh: Best Observability & Security but Complex

Very secure traffic among microservices, fine-grained traffic management, offload some app functions to sidecar

Service Mesh

- Managed by networking team
- N-S L4 load balancing, SSL, WAF
- Citrix ADC or similar



- Managed by platform team
- N-S L7 LB
- Envoy, Citrix CPX, or similar

- Sidecars managed by platform team
- E-W load balancing & more
- Envoy or Citrix CPX as sidecar

North-South App Traffic LB

Green ADC for L4 LB for cloud native; L4-7 LB for monolith apps; Blue ADC for L7 LB and faster change of pace

East-West App Traffic LB

Sidecars for E-W advanced load balancing. Pods communicate via sidecars

App Security	N-S: Excellent protection by green ADC E-W: *Excellent protection by sidecar, policy, rate control, auth, mTLS, API & layer 7 attack protection*
Observability	N-S: Excellent; green & blue ADCs see all traffic E-W: *Excellent; as sidecar sees all the traffic*
Continuous Deployment	N-S: Excellent; advanced traffic control by ADCs E-W: *Excellent; advanced traffic control by sidecar*
Scale Performance	N-S: Good for scale out E-W: *Distributed architecture scalability, sidecar-quality dependent, adds 2-hop latency, more CPU/memory*
Open Source Tools Support	N-S: Excellent; e.g., Prometheus, Spinnaker, EFK E-W: *Excellent; e.g., Prometheus, Spinnaker, EFK*
Istio-Unified Control Plane	N-S: Support via Istio-enabled ADC E-W: *Support via Istio APIs, Istio Mixer bottlenecks.*
IT Skill Set Required	Steep learning curve for platform & networking teams

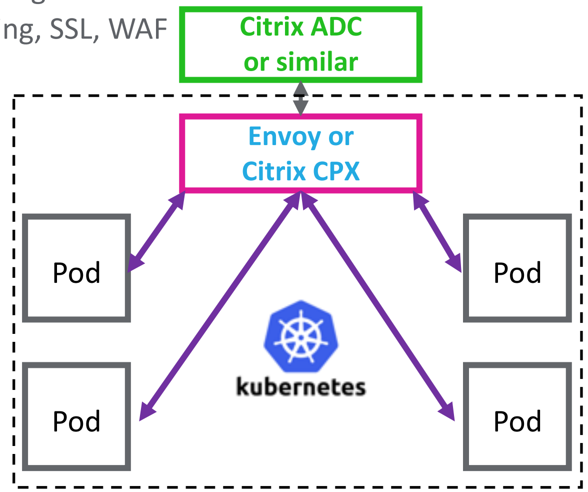
LB = Load Balancing ADC = Application Delivery Controllers

Service Mesh Lite: Service Mesh-Like Benefits & Simpler

Secure traffic among microservices, optional encryption by app, fine-grained traffic management, observability

Service Mesh Lite

- Managed by networking team
- N-S L4-7 load balancing, SSL, WAF
- Citrix ADC or similar



- Managed by platform team
- E-W LB & more
- Envoy, Citrix CPX, or similar

North-South App Traffic LB

Green ADC for L4-7 LB & security for cloud native & monolith apps

East-West App Traffic LB

Purple ADC for E-W advanced load balancing

App Security	N-S: Excellent protection by green ADC E-W: Excellent protection by purple ADC, *optional mTLS*
Observability	N-S: Excellent; green ADC sees all traffic E-W: Excellent; purple ADC sees all traffic
Continuous Deployment	N-S: Excellent; advanced traffic control by ADCs E-W: Excellent; adv. traffic control by purple ADC
Scale Performance	N-S: Good for scale out E-W: *Highly scalable, adds 1-hop latency*
Open Source Tools Support	N-S: Excellent; e.g., Prometheus, Spinnaker, EFK E-W: Excellent; e.g., Prometheus, Spinnaker, EFK
Istio-Unified Control Plane	N-S: Support via Istio-enabled ADC E-W: Support via Istio APIs, Istio Mixer bottlenecks
IT Skill Set Required	*Minimal training for platform & networking teams* *Easy transition from 2-Tier ingress architecture*

LB = Load Balancing ADC = Application Delivery Controllers

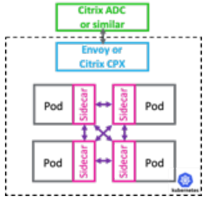
What Will Be Your Architecture Choice?

Move to cloud native at your pace

Balance the tradeoff
between benefits and complexity

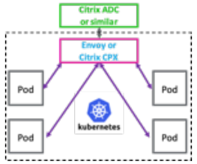
High

Benefits



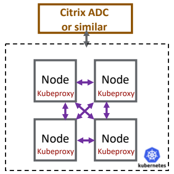
Service Mesh

Best observability &
security but complex



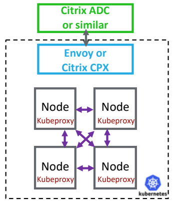
Service Mesh Lite

Service mesh-like
benefits but simpler



Unified Ingress

Simple for network-savvy
platform teams



2-Tier Ingress

Simplest and quickest way
to production

Low

Complexity

High

Citrix Cloud Native Solution Principles

A comprehensive solution addresses all stakeholder needs:

Architecture Flexibility

Move to cloud native at your pace: ingress, service mesh, Istio

Works With Your Environment & Tools

Get apps to production fast with Kubernetes platform & CNCF tools

Performance & Scale

Support large clusters & very dynamic microservices

App & API Security

Extend integrated security to microservices

Actionable Insights

Gain visibility & troubleshoot problems faster

Production-Grade Solution at the Speed of Business

Broadest Open Source Tools & Platforms Integration

Get your apps to production fast with out-of-the-box integration with your preferred open source tools

Google kubernetes | Amazon EKS | Azure Kubernetes Service | RED HAT OPENSHIFT

Prometheus
Monitoring
CLOUD NATIVE COMPUTING FOUNDATION

Grafana
Data Visualization, Custom Dashboards

HELM
Kubernetes Package Manager
CLOUD NATIVE COMPUTING FOUNDATION

gRPC
Universal RPC Framework
CLOUD NATIVE COMPUTING FOUNDATION

Spinnaker
Multi-cloud Continuous Delivery, Canary

Istio
Control Plane

elasticsearch
Log Collection, Storage, Search

fluentd
Data Collector For Unified Logging Layer
CLOUD NATIVE COMPUTING FOUNDATION

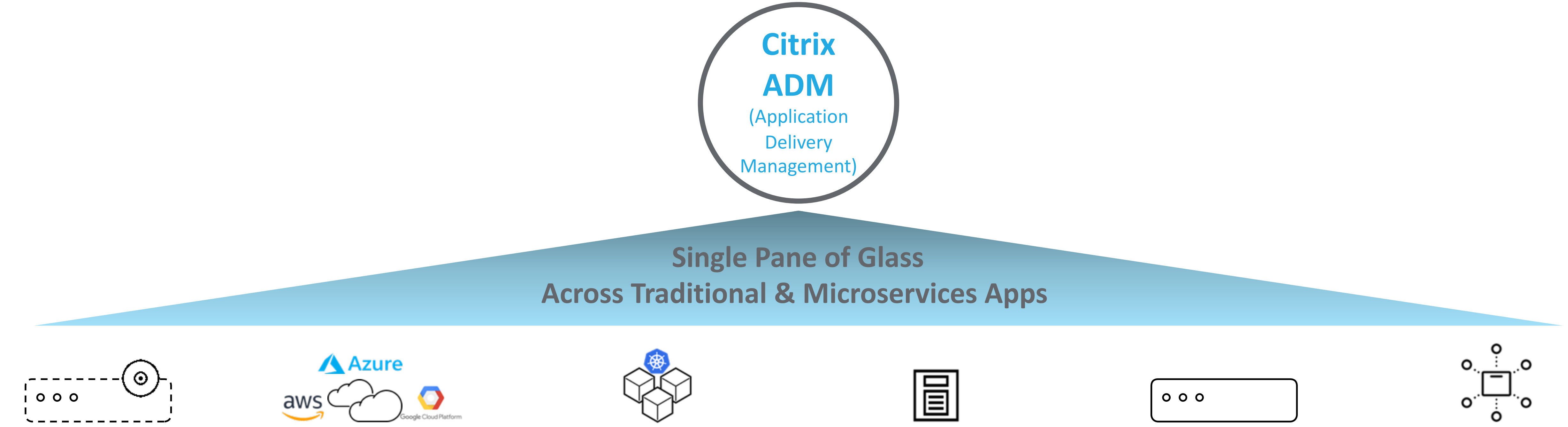
kibana
Query UI, Alerting

CNI
Linux Container Network Interface
CLOUD NATIVE COMPUTING FOUNDATION

ZIPKIN
Distributed Tracing For Latency Issues

Citrix ADC Portfolio: Built for Hybrid Multi-Cloud

Most Comprehensive, Feature-Rich & Software-Centric Application Delivery Solution



Citrix ADM
(Application Delivery Management)

Single Pane of Glass
Across Traditional & Microservices Apps

Virtual Appliances
DC & Private Cloud, VPX

In Public Cloud
VPX for AWS, Azure, GCP

Containers
Public & Private Cloud, CPX

Bare Metal
Private & Public Cloud, BLX

Hardware Appliances
DC & Private Cloud, MPX

Multi-Tenant
MSP & Private Cloud, SDX

Common Software Code Base
across portfolio for
Operational Consistency and Innovation Velocity

Thank you!

github.com/citrix
www.citrix.com/networking/microservices

CLOUD NATIVE COMPUTING FOUNDATION

CiTRIX®