Service Fabric Deep Dive

Christoph Schittko

Azure Technical Architect
GBB Cloud App Dev

Microsoft

Agenda

- Service Fabric – A quick intro
- Container orchestration at hyper-scale
- Any Cloud, any OS
- Data-aware container orchestrator

Service Fabric – A Quick Intro

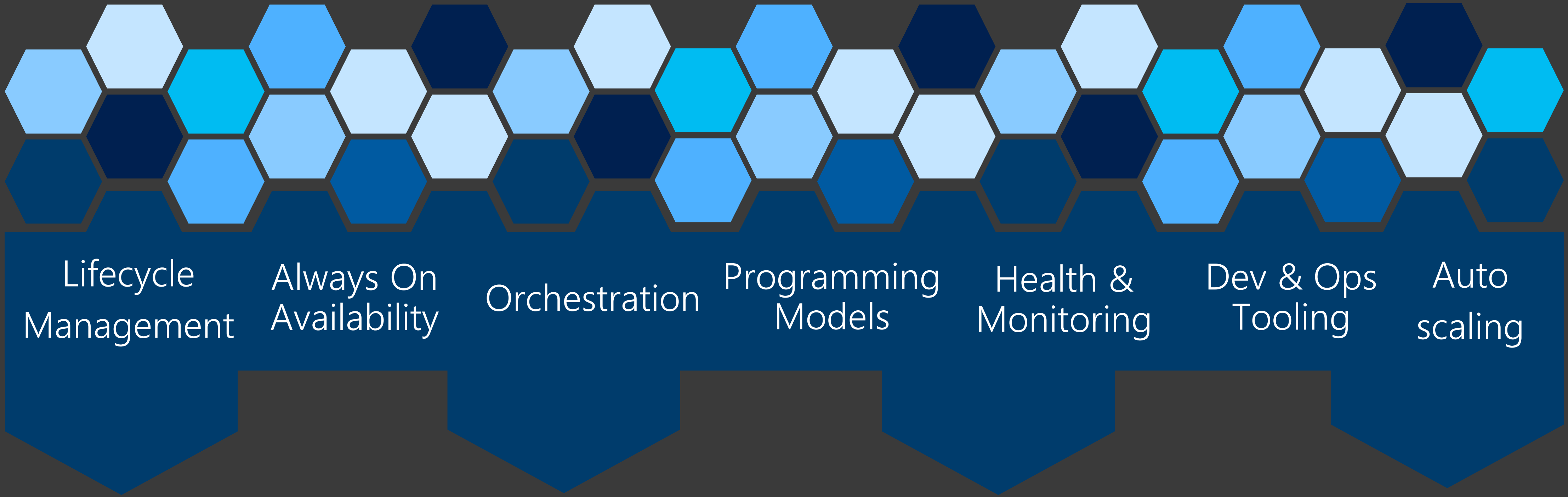
Service Fabric offers an E2E integrated solution

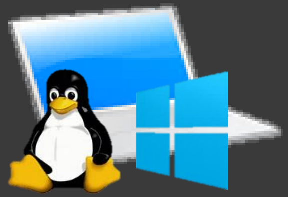
Rolling Upgrades
Availability Guarantees
Scale Out Architecture
Resource Governance
Density
Packaging & Deployment
Policy Enforcement
Granular Versioning
Stateful Workloads
Leader Election

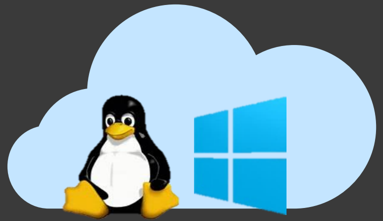
Service Fabric

Azure Service Fabric

Any OS, Any Cloud

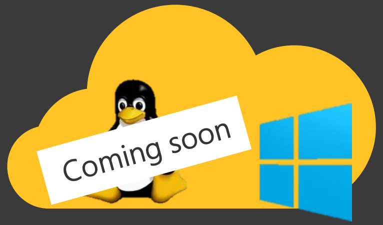



Dev Box


Azure

On-Premises Data Centers

Azure Stack
Coming soon



Other Clouds

What is Service Fabric?

- Clustering – Create a pool of resources

What is Service Fabric?

- Clustering – Create a pool of resources
- Environment abstraction – Run anywhere

What is Service Fabric?

- Clustering – Create a pool of resources
- Environment abstraction – Run anywhere
- Always-on availability – Auto detect and handle failure in seconds

What is Service Fabric?

- Clustering – Create a pool of resources
- Environment abstraction – Run anywhere
- Always-on availability – Auto detect and handle failure in seconds
- Orchestration – Manage resources and placement

What is Service Fabric?

- Clustering – Create a pool of resources
- Environment abstraction – Run anywhere
- Always-on availability – Auto detect and handle failure in seconds
- Orchestration – Manage resources and placement
- Management – Rolling upgrades with health monitoring

What is Service Fabric?

- Clustering – Create a pool of resources
- Environment abstraction – Run anywhere
- Always-on availability – Auto detect and handle failure in seconds
- Orchestration – Manage resources and placement
- Management – Rolling upgrades with health monitoring
- Programming models – Microservices application platform

What is Service Fabric?

- Clustering – Create a pool of resources
- Environment abstraction – Run anywhere
- Always-on availability – Auto detect and handle failure in seconds
- Orchestration – Manage resources and placement
- Management – Rolling upgrades with health monitoring
- Programming models – Microservices application platform
- Reliability & Latency – Support for stateful workloads

What is Service Fabric?

- Clustering – Create a pool of resources
- Environment abstraction – Run anywhere
- Always-on availability – Auto detect and handle failure in seconds
- Orchestration – Manage resources and placement
- Management – Rolling upgrades with health monitoring
- Programming models – Microservices application platform
- Reliability & Latency – Support for stateful workloads
- Data-aware – Meets all your data needs

What is Service Fabric?

- Clustering – Create a pool of resources
- Environment abstraction – Run anywhere
- Always-on availability – Auto detect and handle failure in seconds
- Orchestration – Manage resources and placement
- Management – Rolling upgrades with health monitoring
- Programming models – Microservices application platform
- Reliability & Latency – Support for stateful workloads
- Data-aware – Meets all your data needs
- E2E tooling – IDEs, dev box, chaos testing, monitoring

Services Powered by Service Fabric

SQL Database
2.0 million DBs

Document DB
Billions transactions/day

IoT Hub
10 of Ks devices &
millions of messages

Event Hubs
20bn events/day

Skype

Cortana

Intune

Dynamics

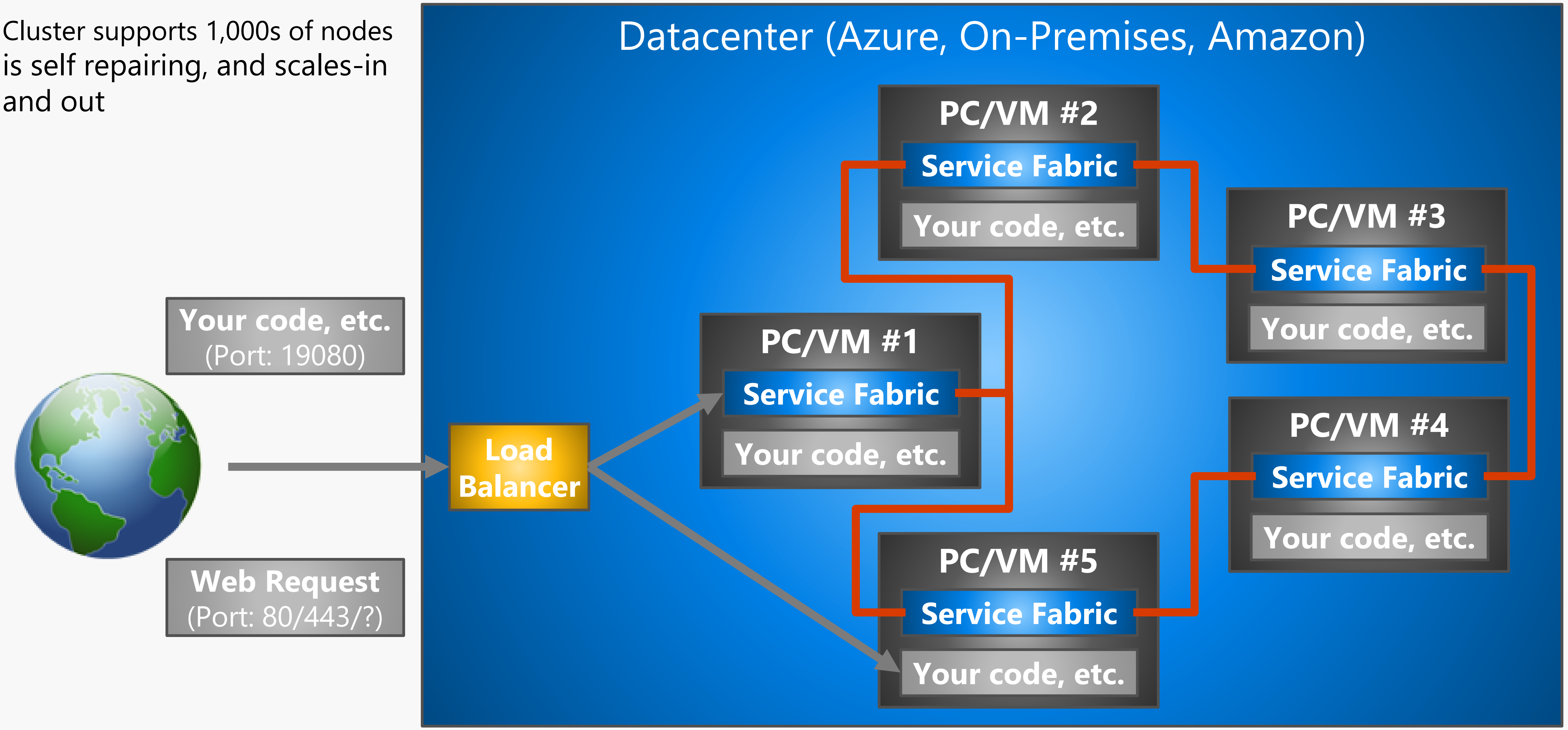
Power BI

Service Fabric Cluster

Cluster Service	Description
Cluster Resource Manager	Cluster and resource management
Failover Manager	Rebalances service instances as nodes come/go
Naming	Registry mapping service instances → endpoints
Fault Analysis	Let’s you inject faults to test your services
Image Store	Contains your app packages
Upgrade	Upgrades Service Fabric on nodes (Azure only)
DNS (new)	Used for containers in conjuction with the naming service for service discovery

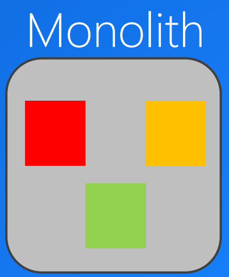
Service Fabric Cluster in Azure

Cluster supports 1,000s of nodes is self repairing, and scales-in and out

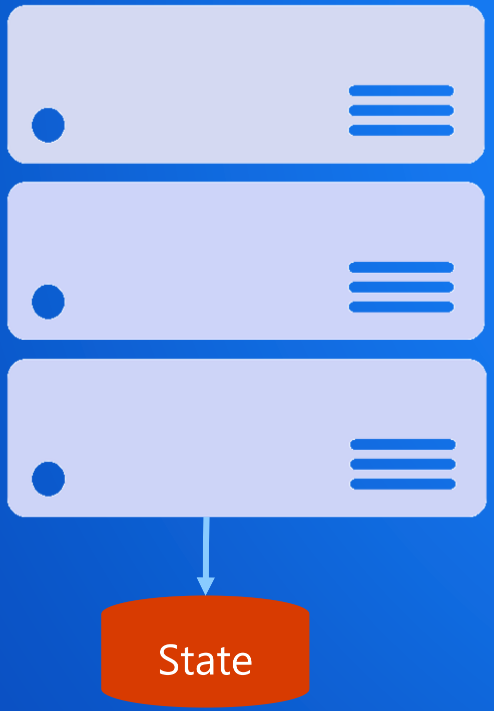


Monolithic application approach

- A monolithic application has most of its functionality within a single process that is commonly componentized with libraries.

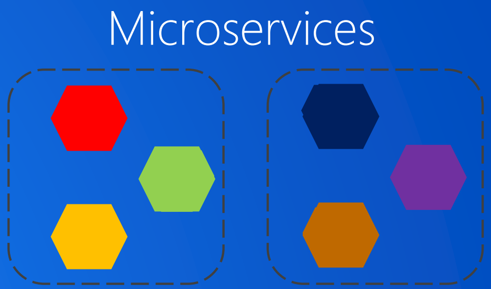


- Scales by cloning the app on multiple servers/VMs/Containers

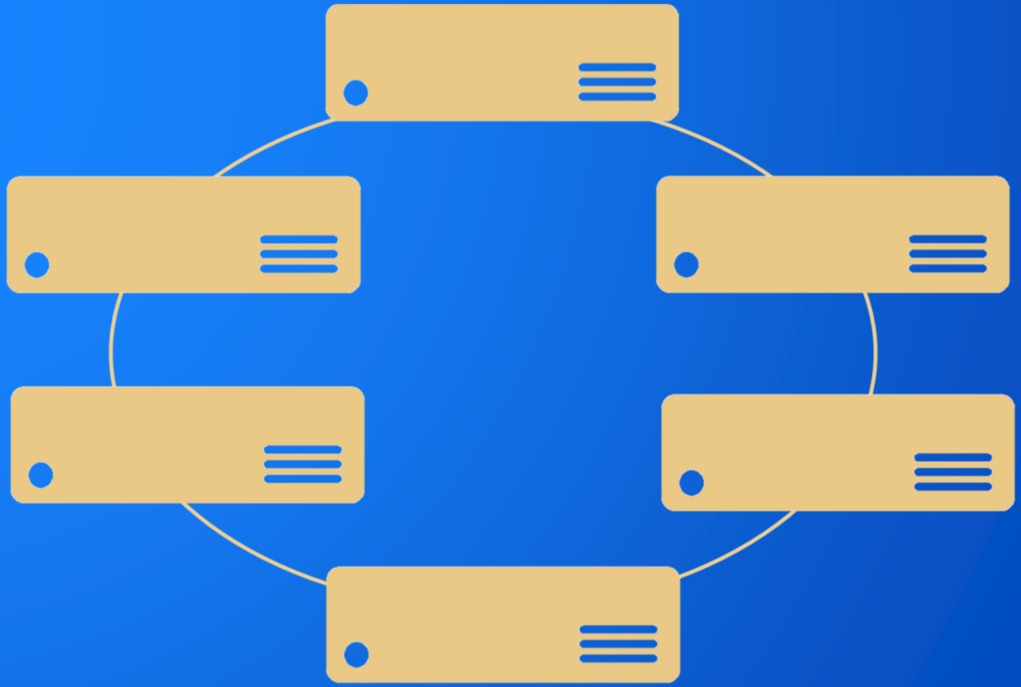


Microservices application approach

- A microservice application separates functionality into separate smaller services.



- Scales out by deploying each service independently creating instances of these services across servers/VMs/containers



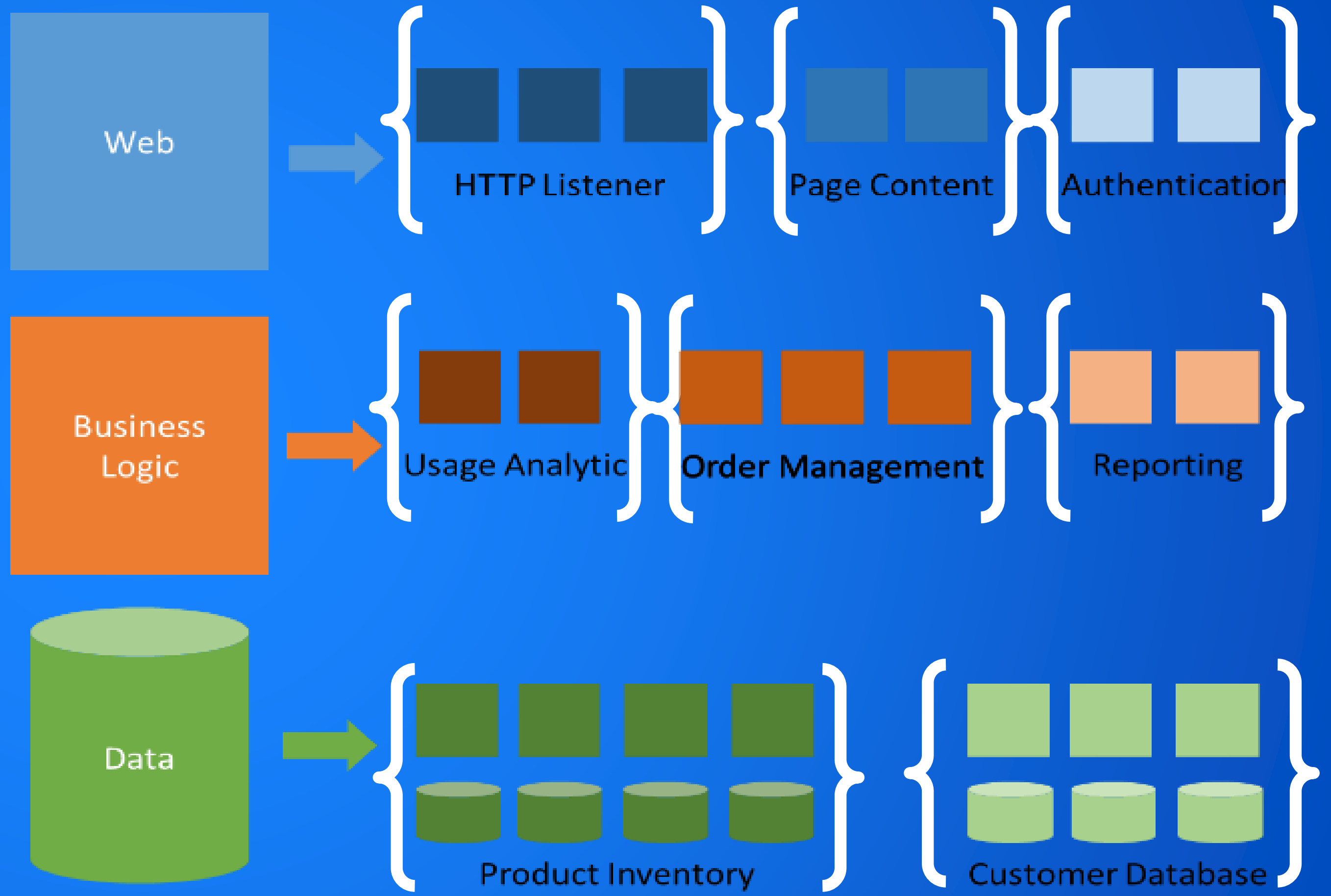
Modernization with microservices

Individually built and deployed

Small, independent services

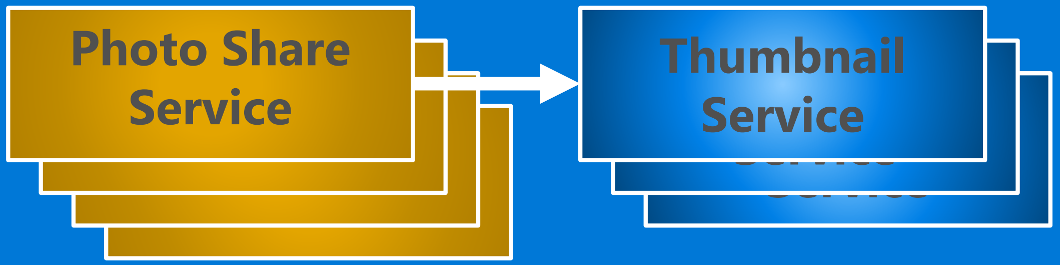
Integrate using published API

Fine-grained, loosely coupled

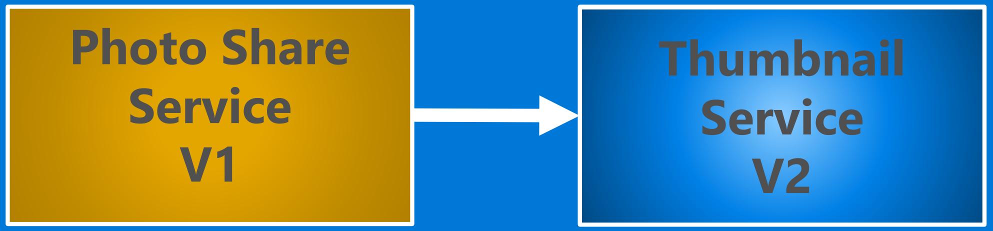


Microservice architecture benefits

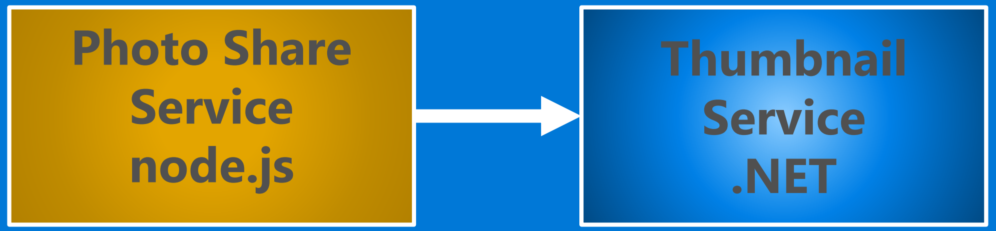
Scale Independently



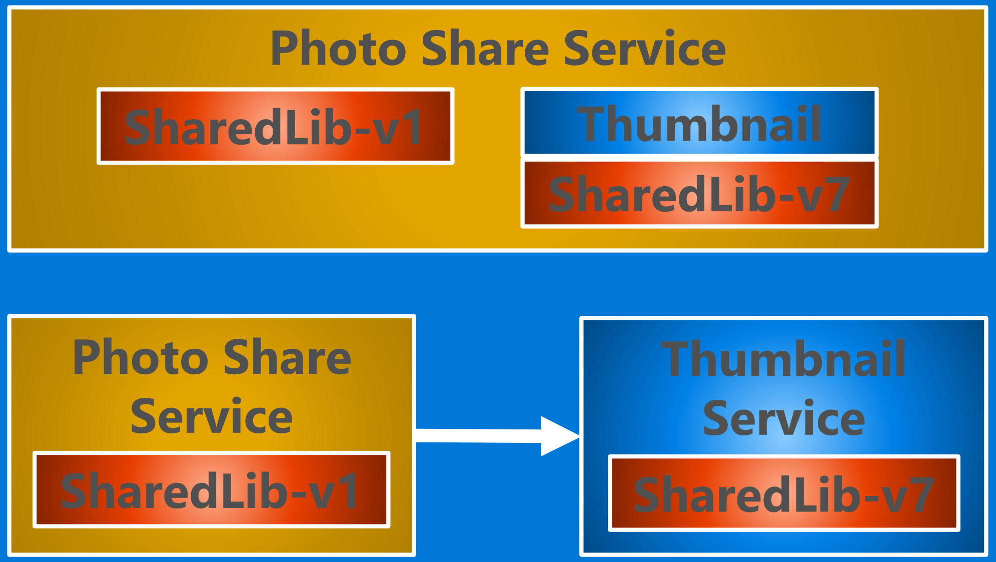
Independent Deployments



Different Technology Stacks



Conflicting Dependencies



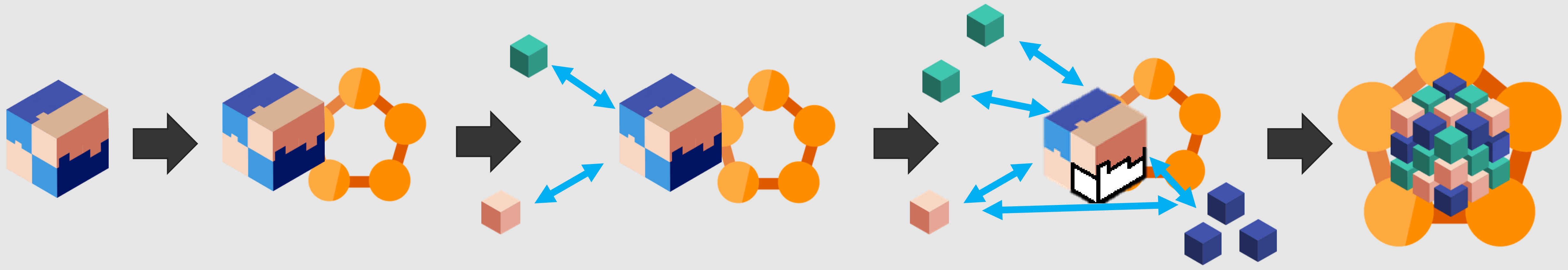
Benefits of microservices

- Enable continuous innovation through independent deployments
- Allow technology diversity
- Built by smaller focused teams
- Improved scale and resource utilization per service
- Provide fault isolation
- No downtime upgrades

What's the downside?

- More services means more network communication
  - Decreases overall performance due to network hops & (de)serialization
  - Requires more failure (timeout) recovery code
- Hard to test in isolation without dependent services
- Hard to debug/monitor across services
- New service versions must support old & new API contracts simultaneously
- Devops persona trade short-term pain for long-term gain

Migrating a traditional application to microservices

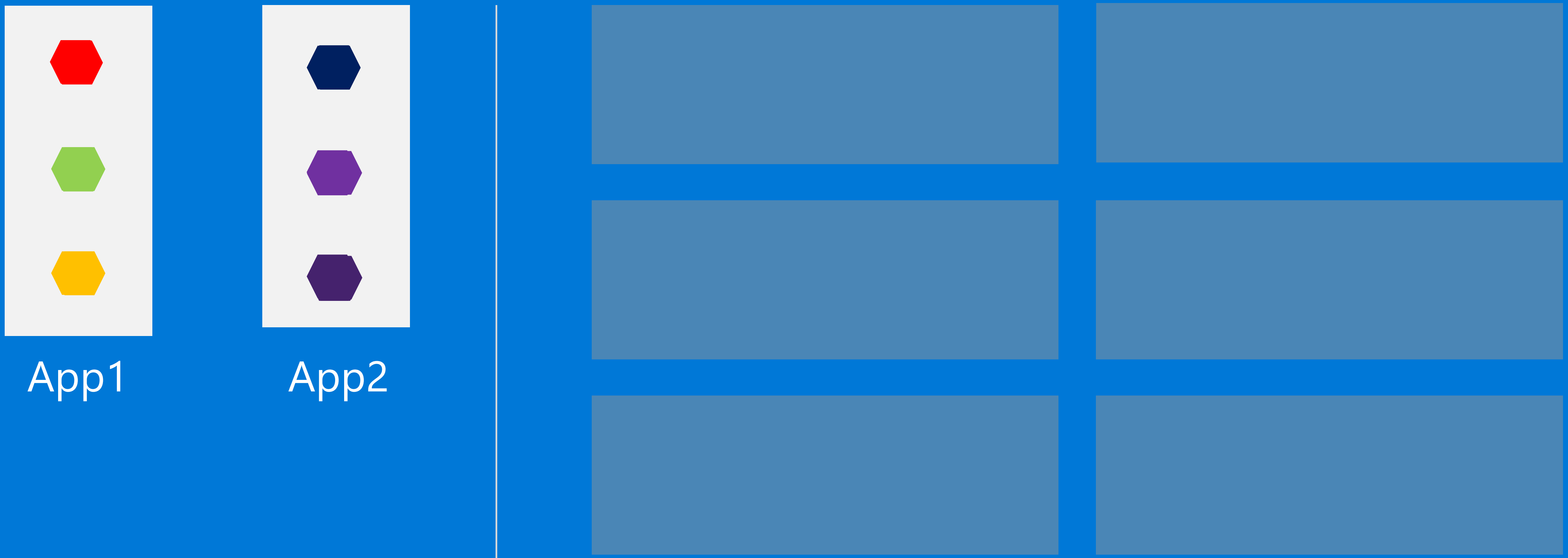


1) Traditional app

...You can stop at any stage

MICROSOFT CONFIDENTIAL – INTERNAL ONLY

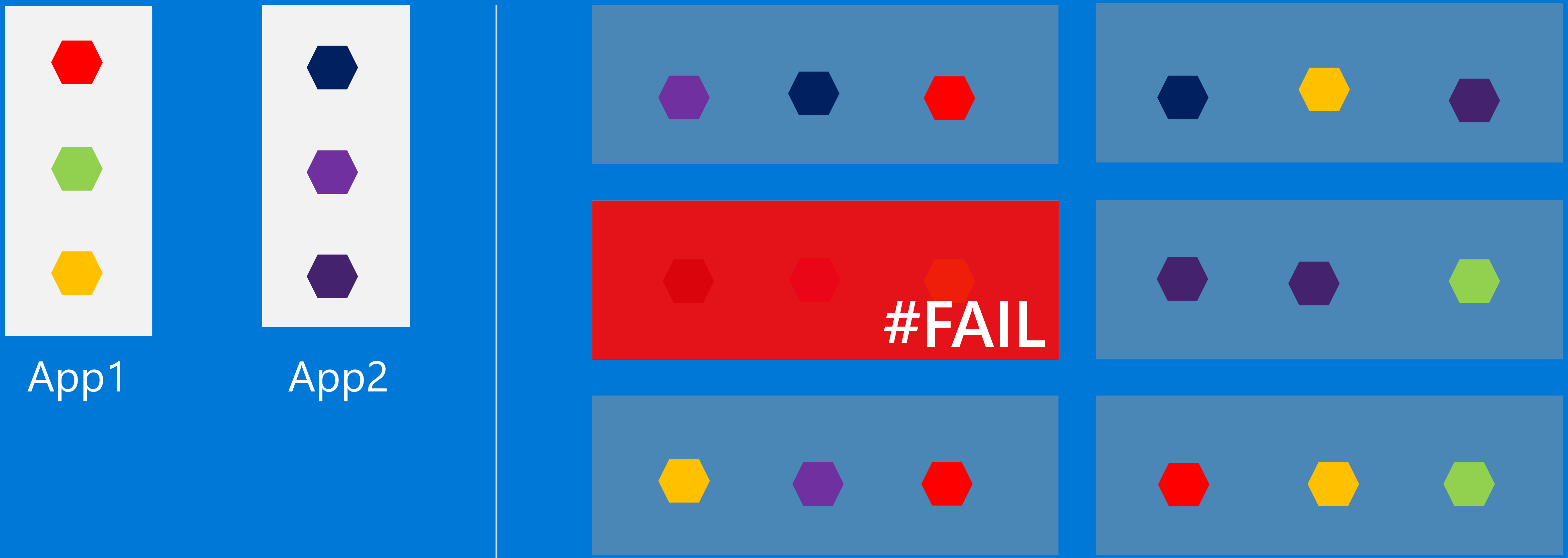
Service Fabric microservices



App Type Packages

Service Fabric Cluster VMs

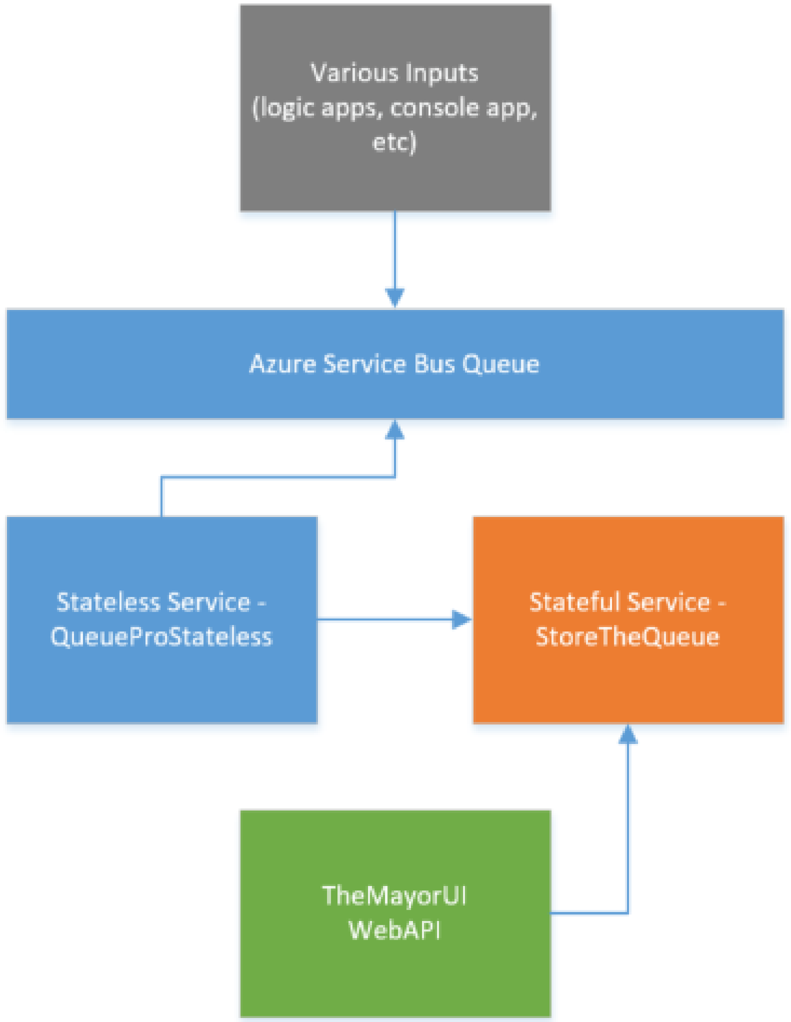
Machine failures



App Type Packages

Service Fabric Cluster VMs

Demo


Various Inputs (logic apps, console app, etc)
Azure Service Bus Queue
Stateless Service - QueueProStateless
Stateful Service - StoreTheQueue
TheMayorUI WebAPI

MICROSOFT CONFIDENTIAL – INTERNAL ONLY

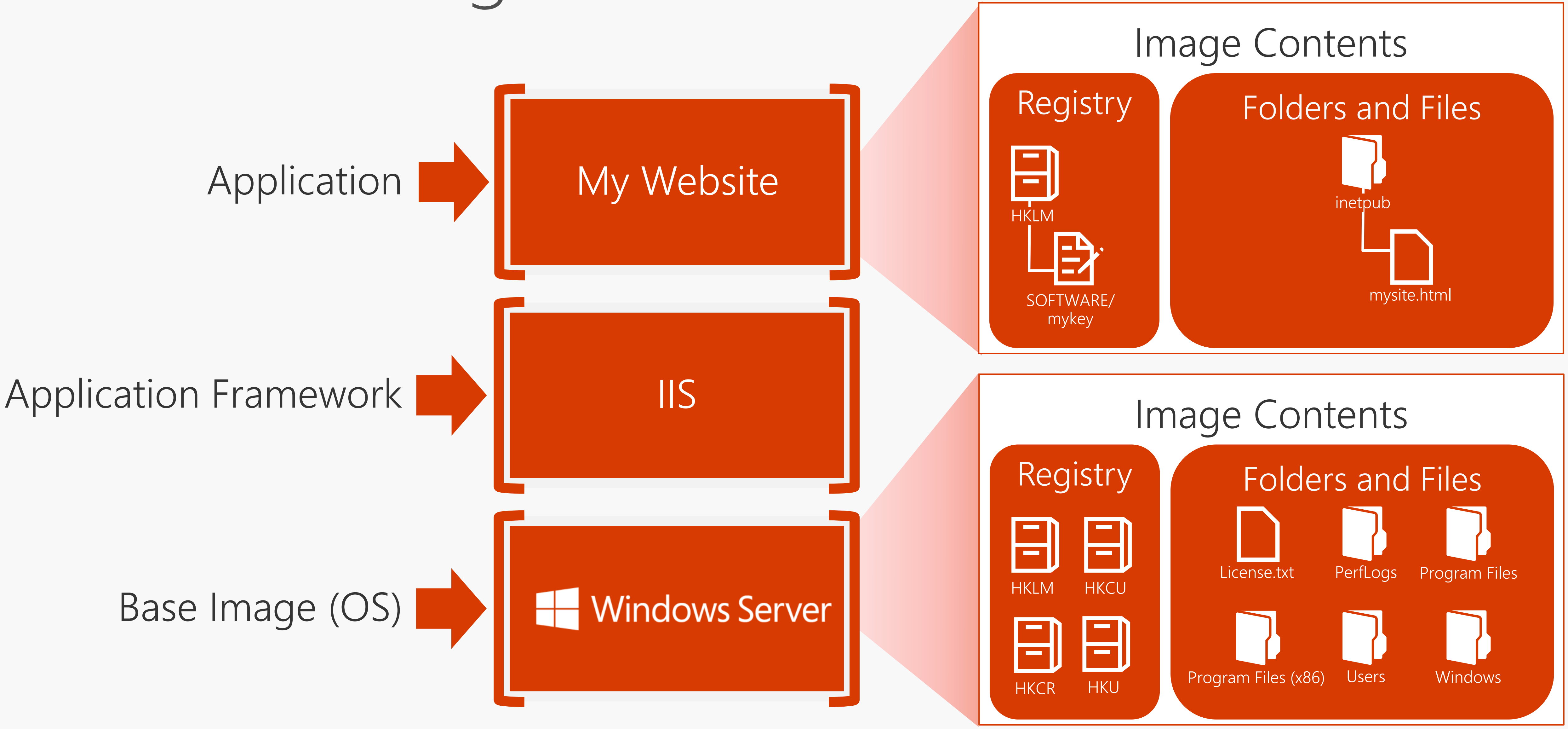
Service Fabric and Containers

- Image deployment and activation
  - Support for authenticating with private registry
  - Environment variables to provide inputs to the container
- Volume driver support
  - Mounting of persistent volumes drives
- Networking
  - Bridge network: Mapping of container ports to dynamic ports on host machine
  - Registration of container endpoints with the Naming Service for communicating between containers
  - DNS service within cluster to resolve container endpoints
- Resource governance
  - Apply policy on containers for resource constraints and use them during placement. CPU, memory, I/O
  - Process constraints: Extending constraints and governance to processes
- Windows containers with Hyper-V isolation
- Preview: Support for Docker Compose

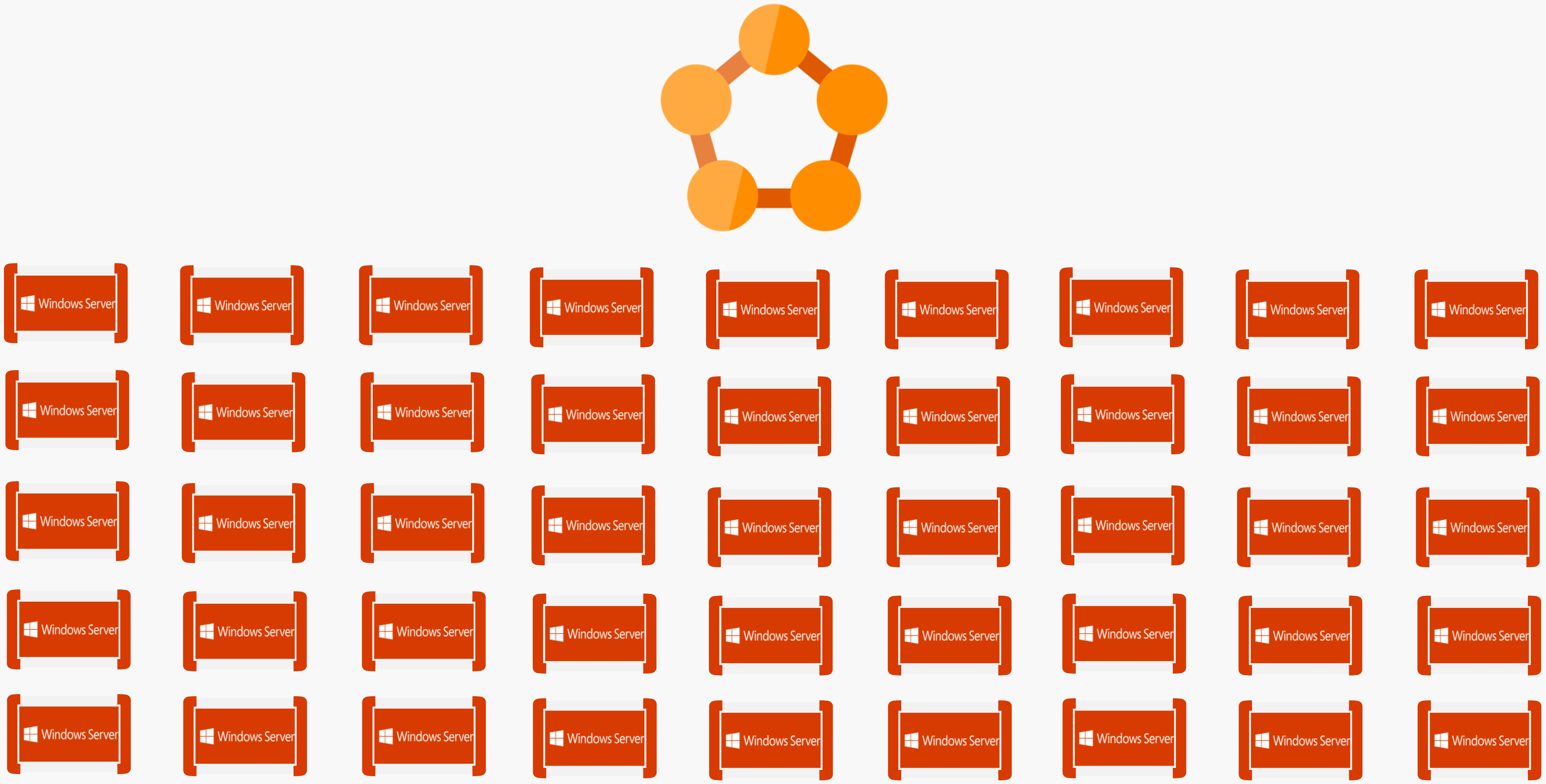
Dv3
New generation of D family

Ev3
High memory

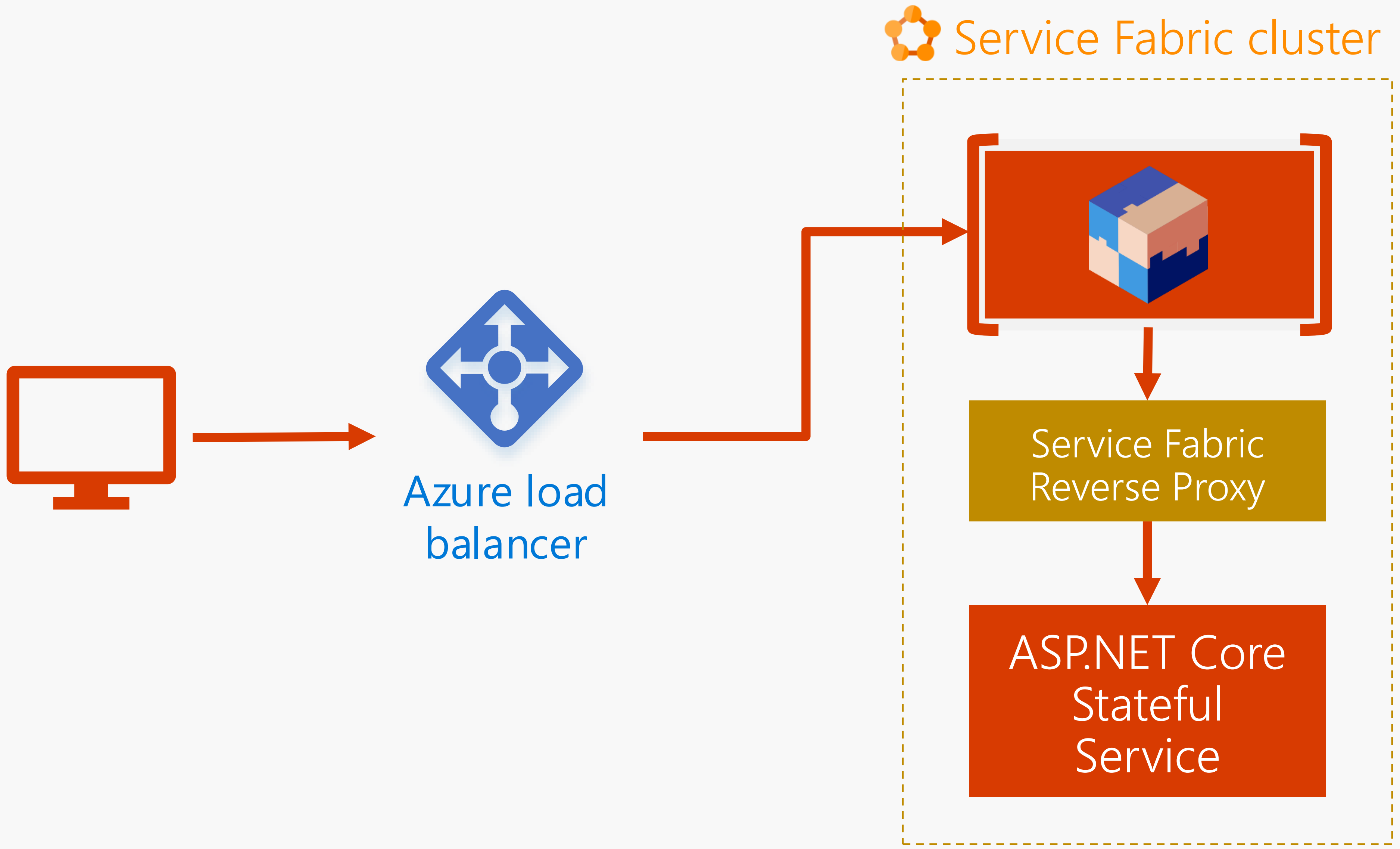
Container images



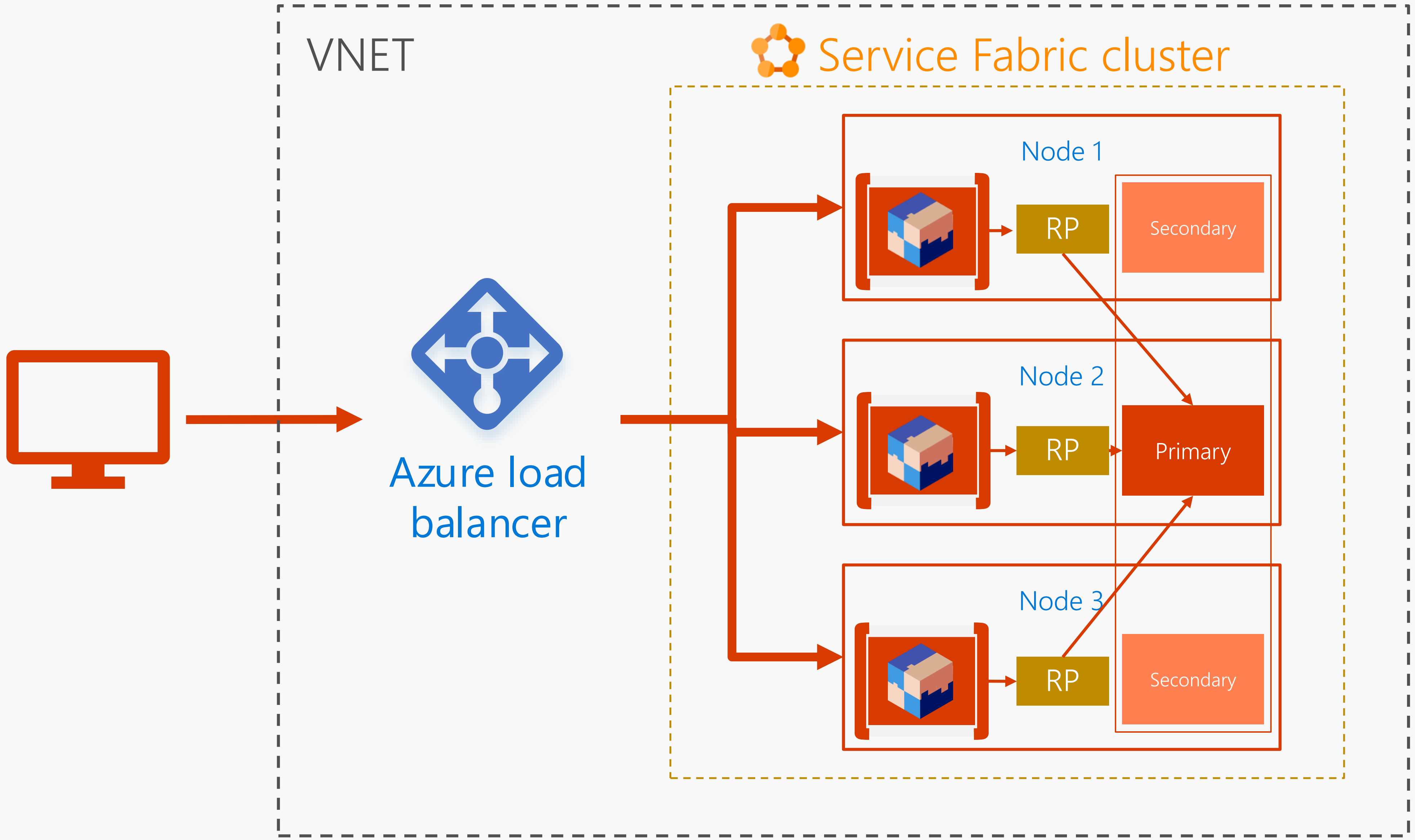
Service Fabric = Windows Container Orchestrator



Gateway to your application



Gateway to your application: stateless service



Service Fabric + API Management

Service Fabric

+

Azure API Management

Service discovery and routing

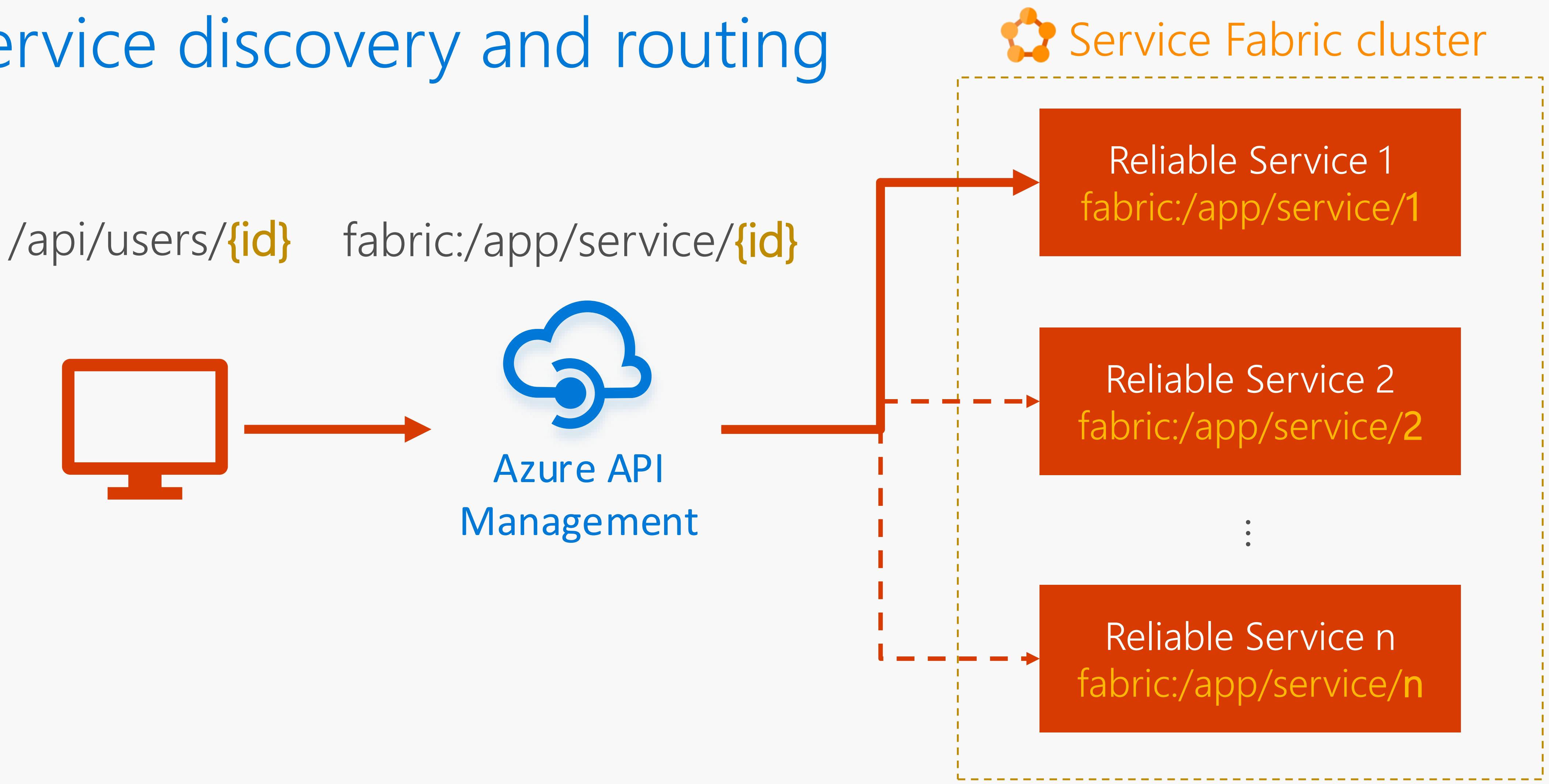
Partition resolution

Replica selection

Resolve and retry policies

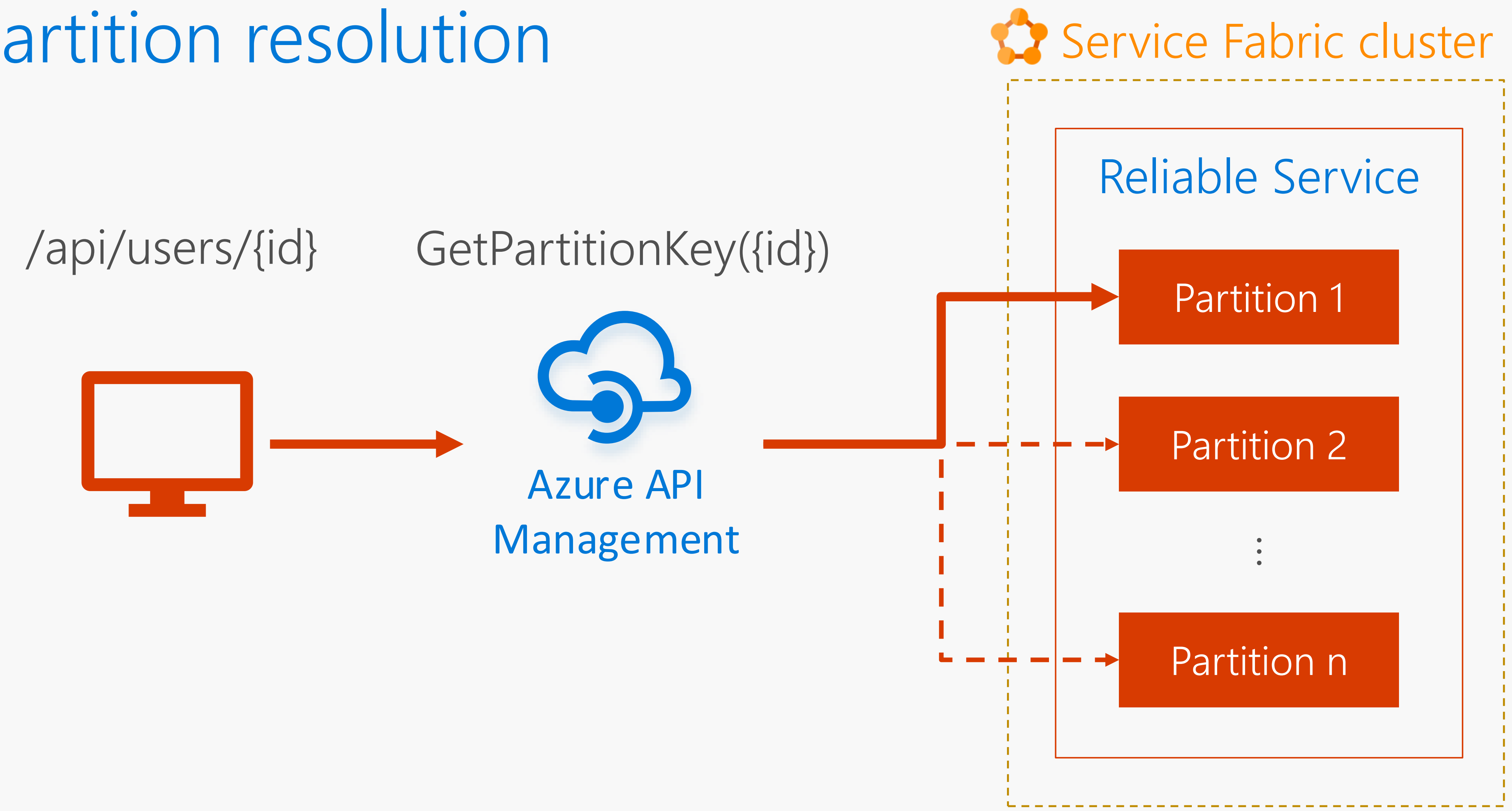
Gateway to your application: API Management

Service discovery and routing

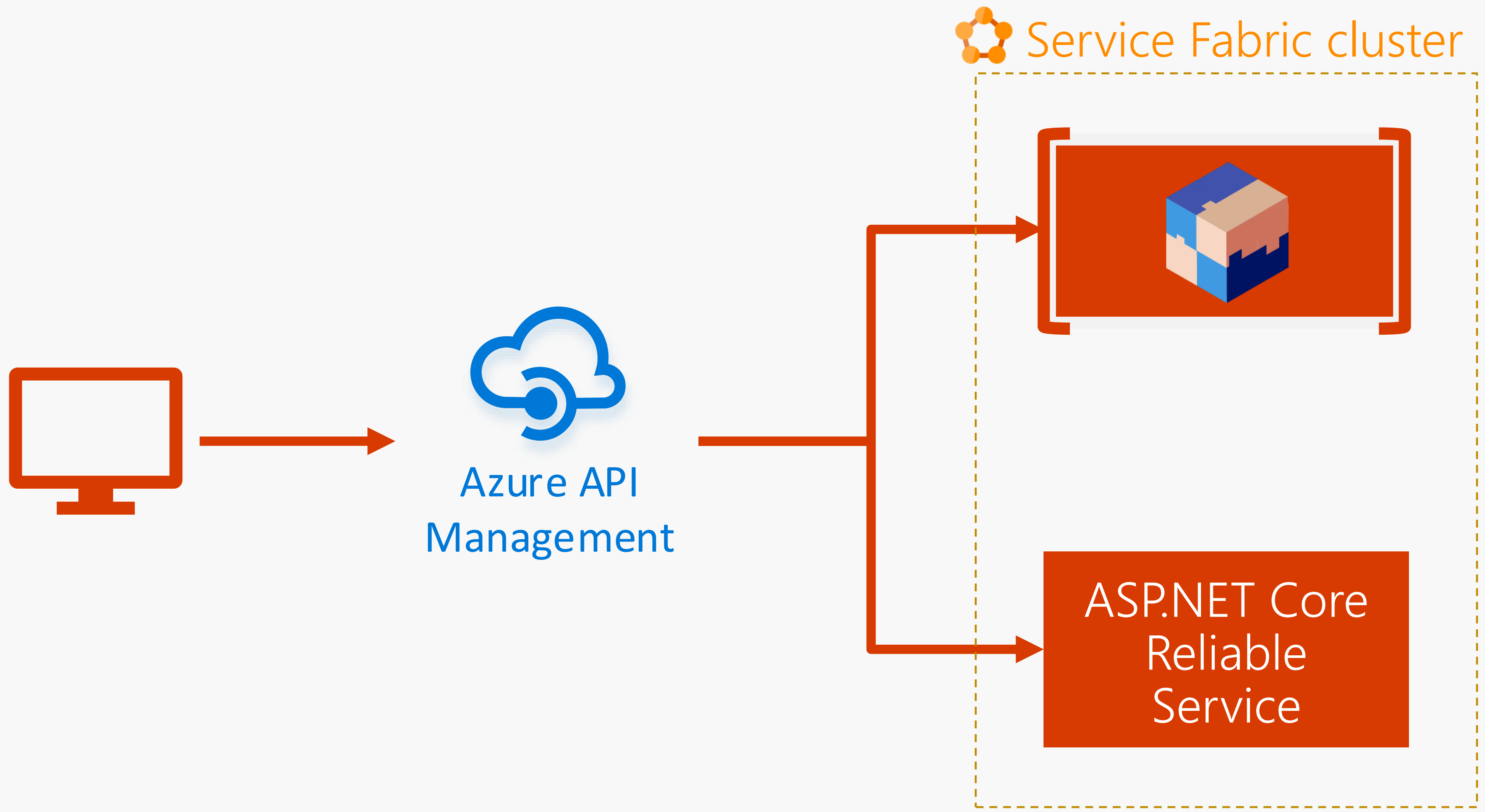


Gateway to your application: API Management

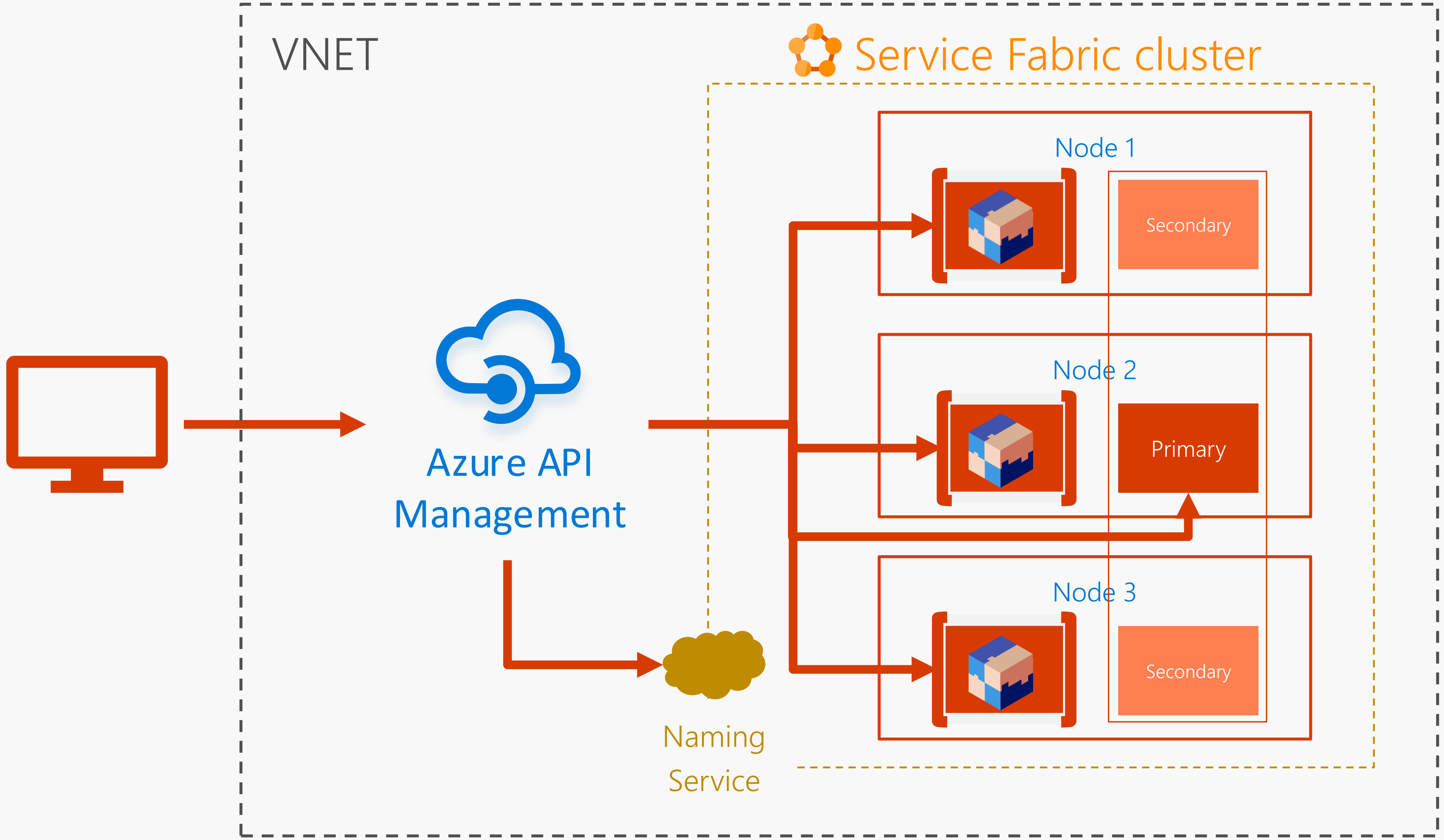
Partition resolution



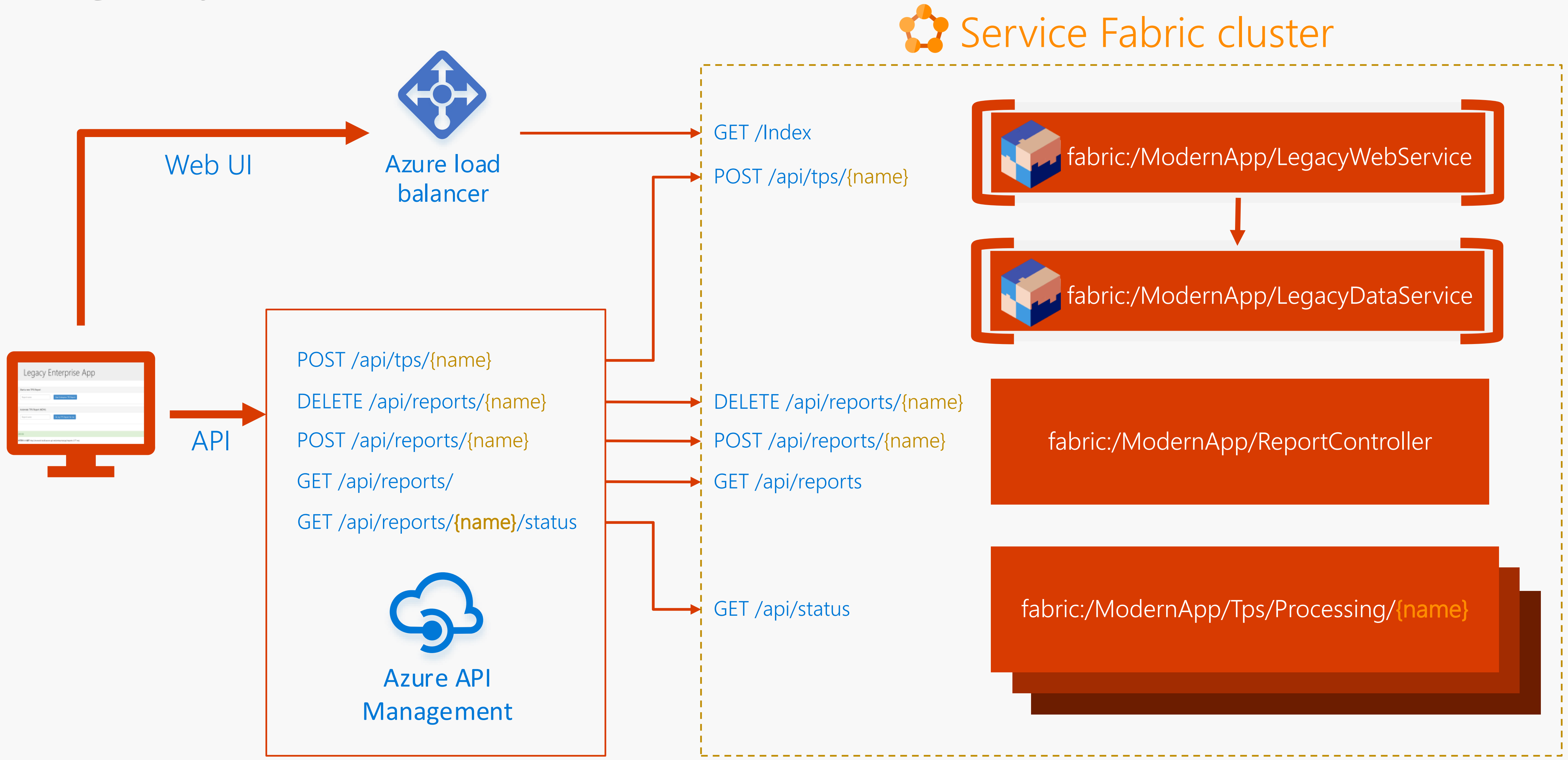
Gateway to your application: API Management



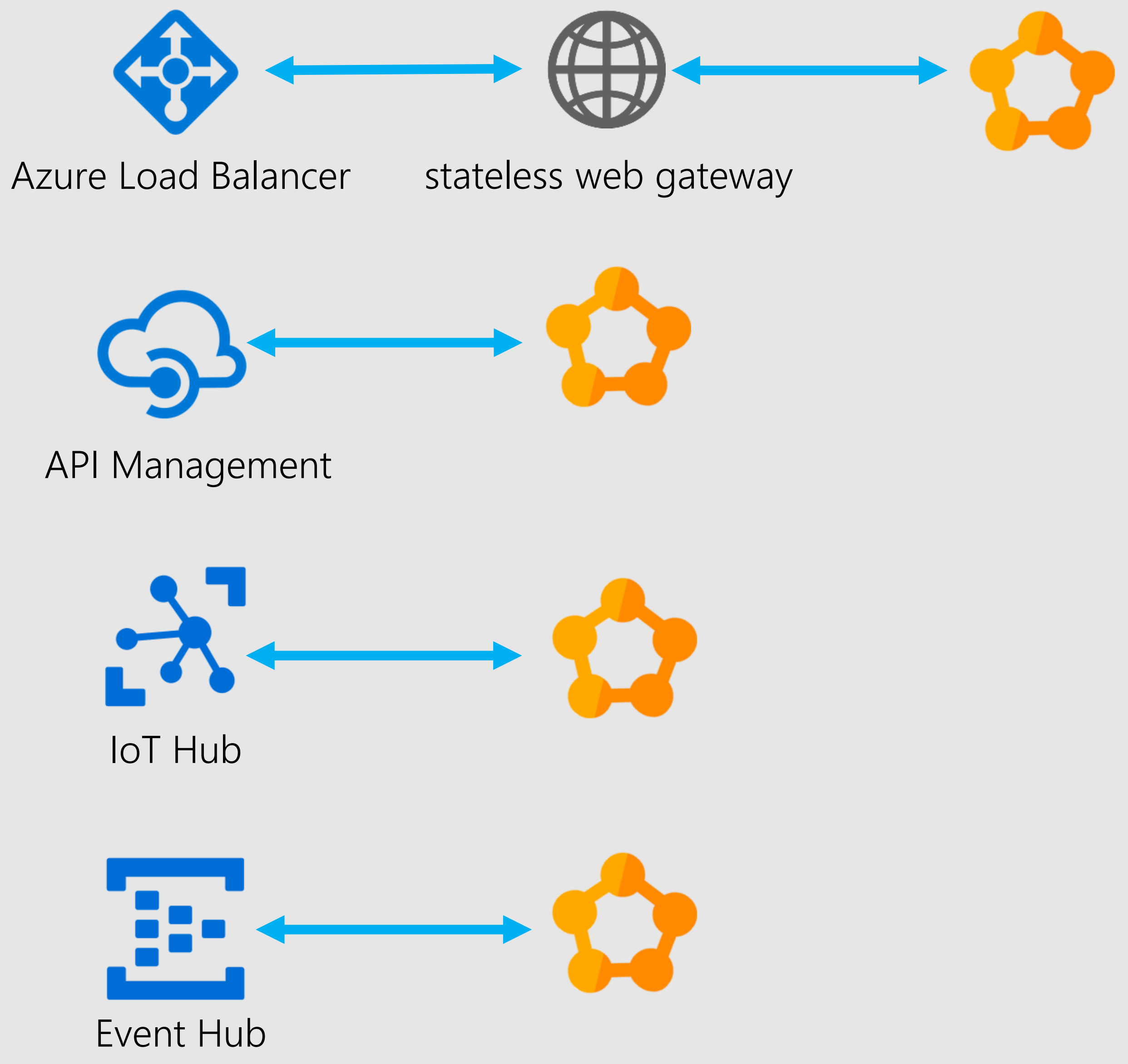
Gateway to your application: API Management



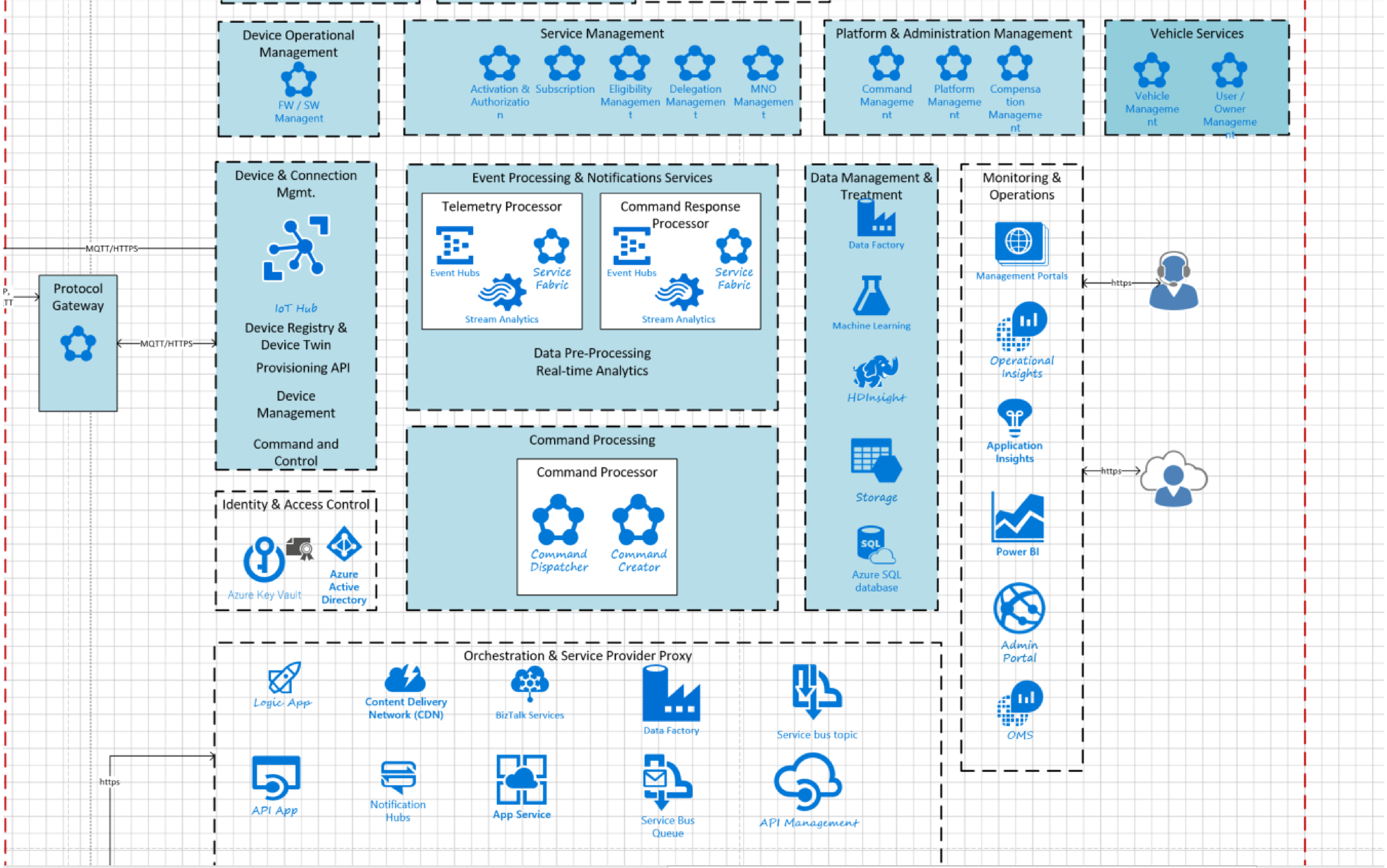
Legacy application: Modernized

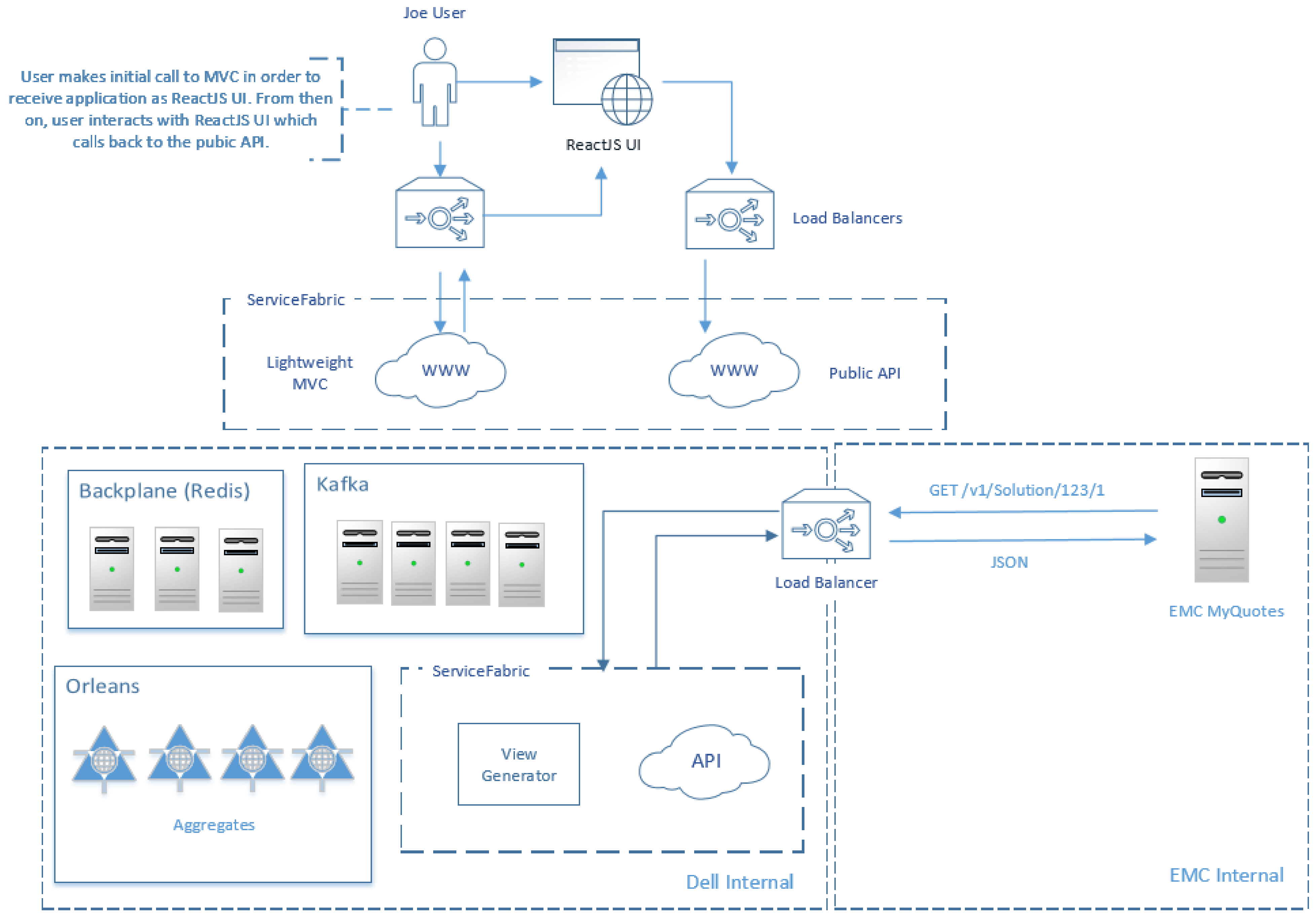


Common gateways for Service Fabric

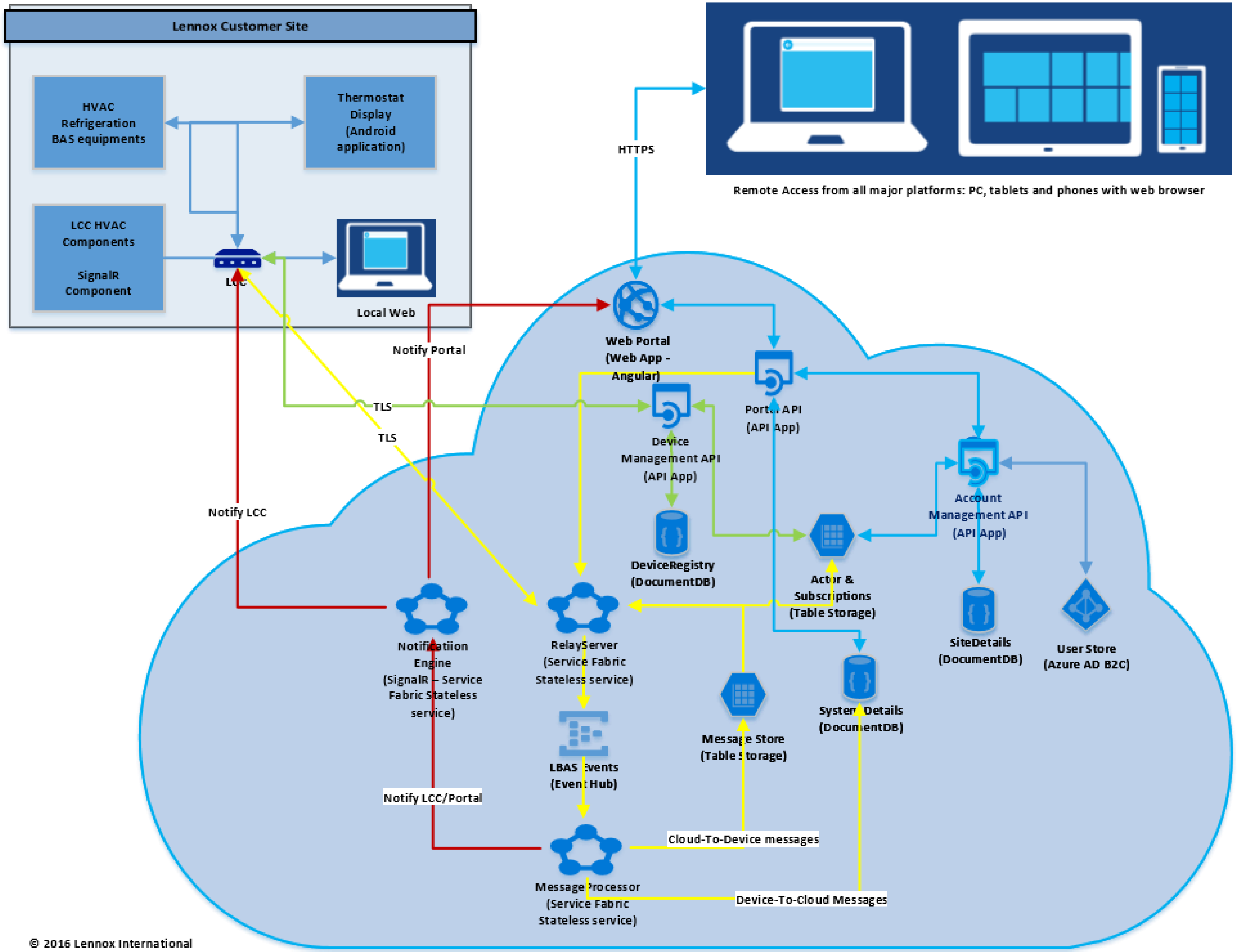


MICROSOFT CONFIDENTIAL – INTERNAL ONLY


Device Operational Management
FW / SW Managent
Service Management
Activation & Authorizatio n
Subscription
Eligibility Managemen t
Delegation Managemen t
MNO Managemen t
Platform & Administration Management
Command Manageme nt
Platform Manageme nt
Compensa tion Manageme nt
Vehicle Services
Vehicle Manageme nt
User / Owner Manageme nt
Device & Connection Mgmt.
IoT Hub
Device Registry & Device Twin
Provisioning API
Device Management
Command and Control
MQTT/HTTPS
Protocol Gateway
MQTT/HTTPS
Event Processing & Notifications Services
Telemetry Processor
Event Hubs
Service Fabric
Stream Analytics
Command Response Processor
Event Hubs
Service Fabric
Stream Analytics
Data Pre-Processing
Real-time Analytics
Data Management & Treatment
Data Factory
Machine Learning
HDInsight
Storage
Azure SQL database
Monitoring & Operations
Management Portals
Operational Insights
Application Insights
Power BI
Admin Portal
OMS
https
https
Command Processing
Command Processor
Command Dispatcher
Command Creator
Identity & Access Control
Azure Key Vault
Azure Active Directory
Orchestration & Service Provider Proxy
Logic App
Content Delivery Network (CDN)
BizTalk Services
Data Factory
Service bus topic
API App
Notification Hubs
App Service
Service Bus Queue
API Management
https


Joe User
User makes initial call to MVC in order to receive application as ReactJS UI. From then on, user interacts with ReactJS UI which calls back to the pubic API.
ReactJS UI
Load Balancers
ServiceFabric
Lightweight MVC
WWW
WWW
Public API
Backplane (Redis)
Kafka
Load Balancer
GET /v1/Solution/123/1
JSON
EMC MyQuotes
Orleans
Aggregates
ServiceFabric
View Generator
API
Dell Internal
EMC Internal

iCON Core Architecture


Lennox Customer Site
HVAC Refrigeration BAS equipments
Thermostat Display (Android application)
LCC HVAC Components
SignalR Component
LCC
Local Web
Remote Access from all major platforms: PC, tablets and phones with web browser
HTTPS
Notify Portal
TLS
TLS
Notify LCC
Web Portal (Web App - Angular)
Portal API (API App)
Device Management API (API App)
DeviceRegistry (DocumentDB)
Actor & Subscriptions (Table Storage)
Account Management API (API App)
SiteDetails (DocumentDB)
User Store (Azure AD B2C)
Notificatiion Engine (SignalR – Service Fabric Stateless service)
RelayServer (Service Fabric Stateless service)
LBAS Events (Event Hub)
Message Store (Table Storage)
System Details (DocumentDB)
Notify LCC/Portal
Cloud-To-Device messages
Message Processor (Service Fabric Stateless service)
Device-To-Cloud Messages

© 2016 Lennox International Inc.