2017

Global Azure BOOTCAMP

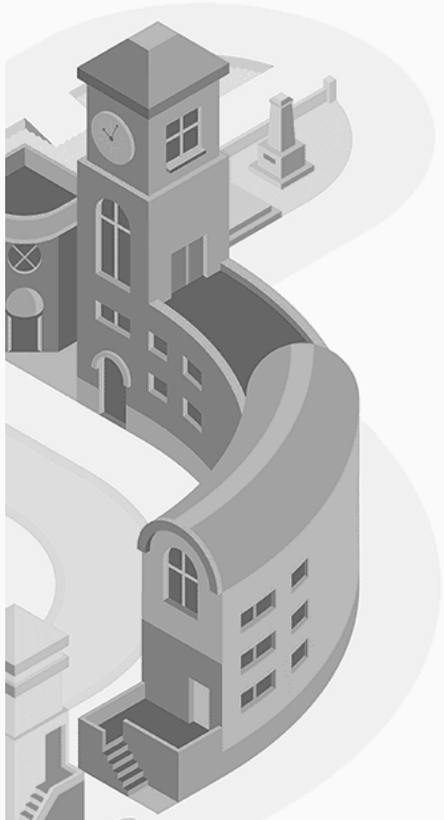
autoexec.gr

SQLschool.gr

dotNETZone.gr
η δύναμη των ελλήνων προγραμματιστών

2017 Global Azure BOOTCAMP

Designing for scalability and high availability on Microsoft Azure



Vaggelis Kappas
Premier Field Engineer - Microsoft
http://autoexec.gr | vaggeliskappas

autoexec.gr

Session objectives and takeaways

Session objective(s):

1. Design highly available applications on the Azure platform

2. Understand best practices and common pitfalls to avoid

3. Learn the inner workings of Azure and how to best work around hidden limitations

Note: *This session focuses mostly on the design of Cloud-based apps using IaaS.*

High availability

What does HA mean to you?

What are the business requirements?

Different requirements mean a different architecture

Resiliency

Best For:

Backup	Disaster recovery	High availability
Data deletion	Protection for unplanned failures	Mission critical apps
Data corruption	Don’t want to re-architect for HA	New apps
Legal, governance & compliance	Don’t want the cost of HA	Localized failures
	Large scale failures	

Mean time between failure (MTBF)

Metric used to identify reliability

Component failures

Cost of delivering cloud infrastructure

On-premises vs cloud

Hardware failure is inevitable

Designing for the cloud

No single point of failure

State stays at the edges of your app stack

Loosely couple your components

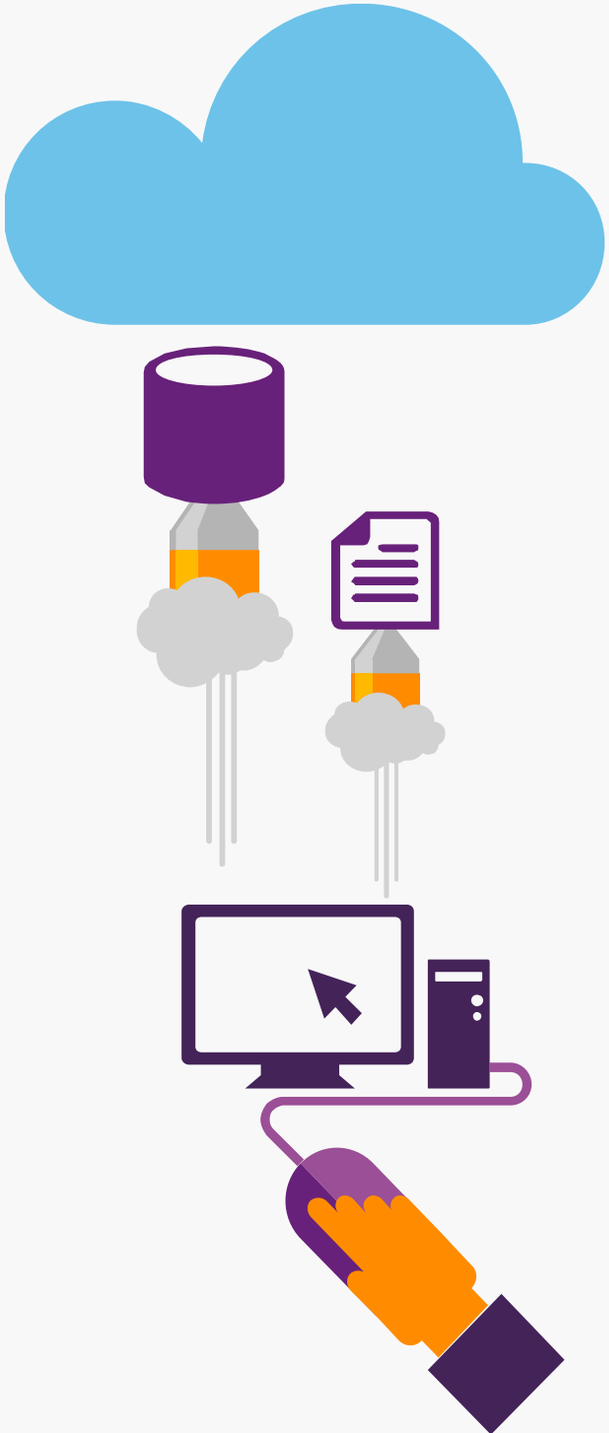
Build for scale

Process centrally, deliver locally

Automate

Trust and verify

Scale out, not up



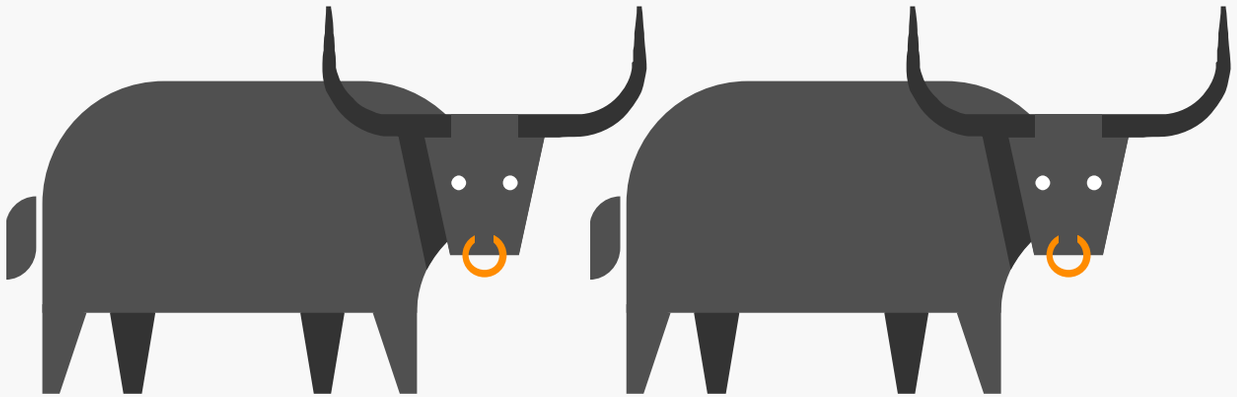
Scaling vertical (up) vs horizontal (out)

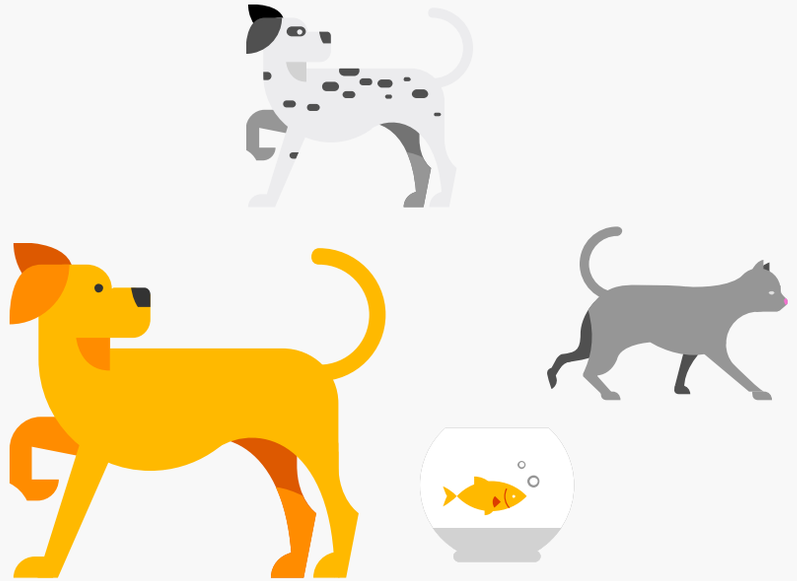
Vertical

Horizontal

Cost

VMSS and autoscale





Good candidates for Azure

Built versus bought software

Software designed for cloud

Applications designed around a single-state datastore

What can go wrong?

Compute

Networking

Storage

All 3

Compute

Availability Sets

Governs fault and update domains

Tied to a role in your application

Required for 99.95% SLA from Microsoft

Provides fault domains and upgrade domains

Availability Sets (upgrade domain)

Fault Domain is single point of failure

Upgrade domain used for patching

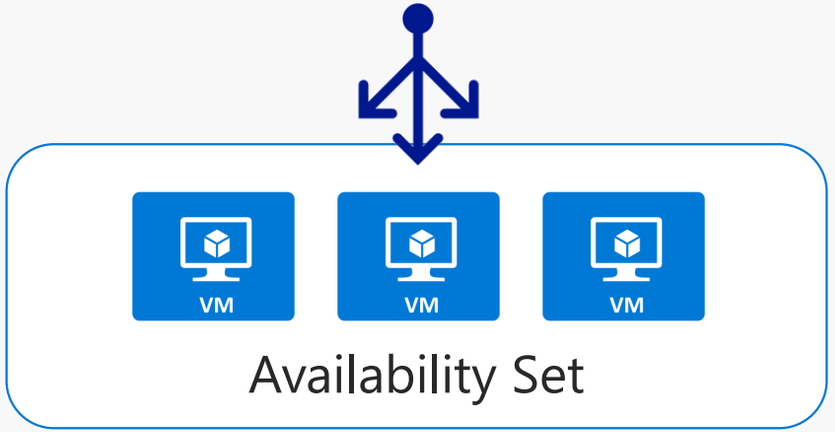
ASM (2 FDs)

VM	Update Domain	Fault Domain
VM1	0	0
VM2	1	1
VM3	2	0
VM4	3	1
VM5	4	0

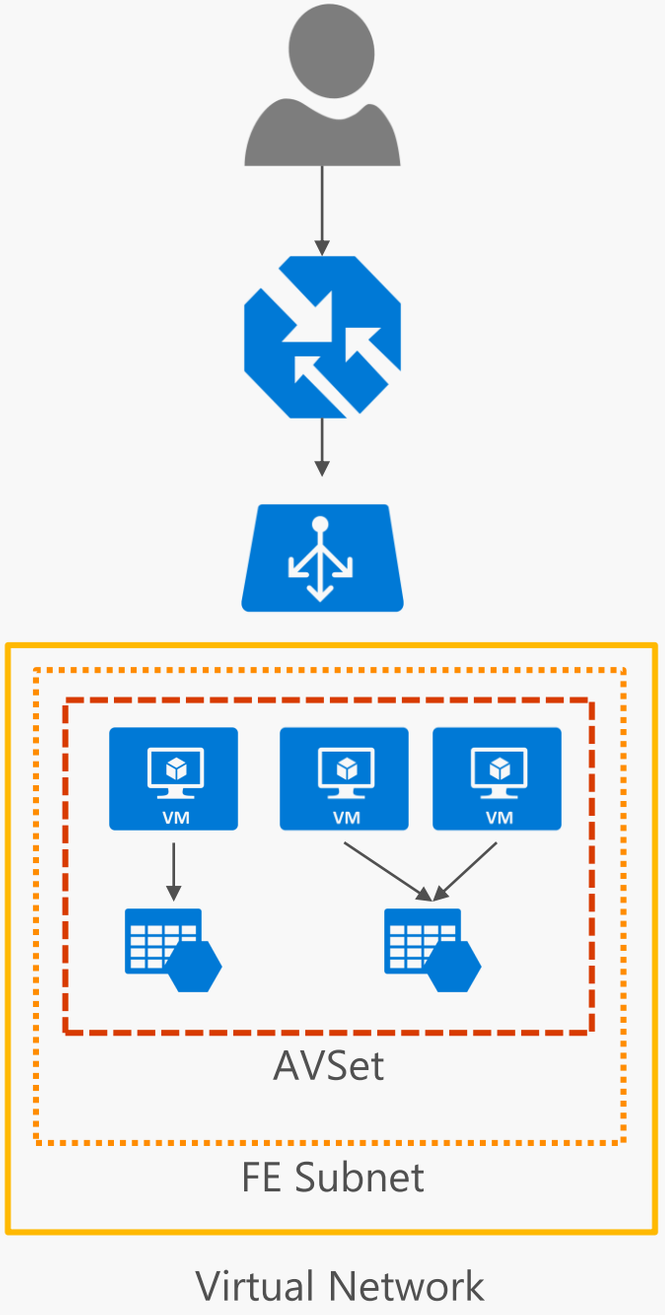
ARM (3 FDs)

VM	Update Domain	Fault Domain
VM1	0	0
VM2	1	1
VM3	2	2
VM4	3	0
VM5	4	1
VM6	5	2

Deployment scale

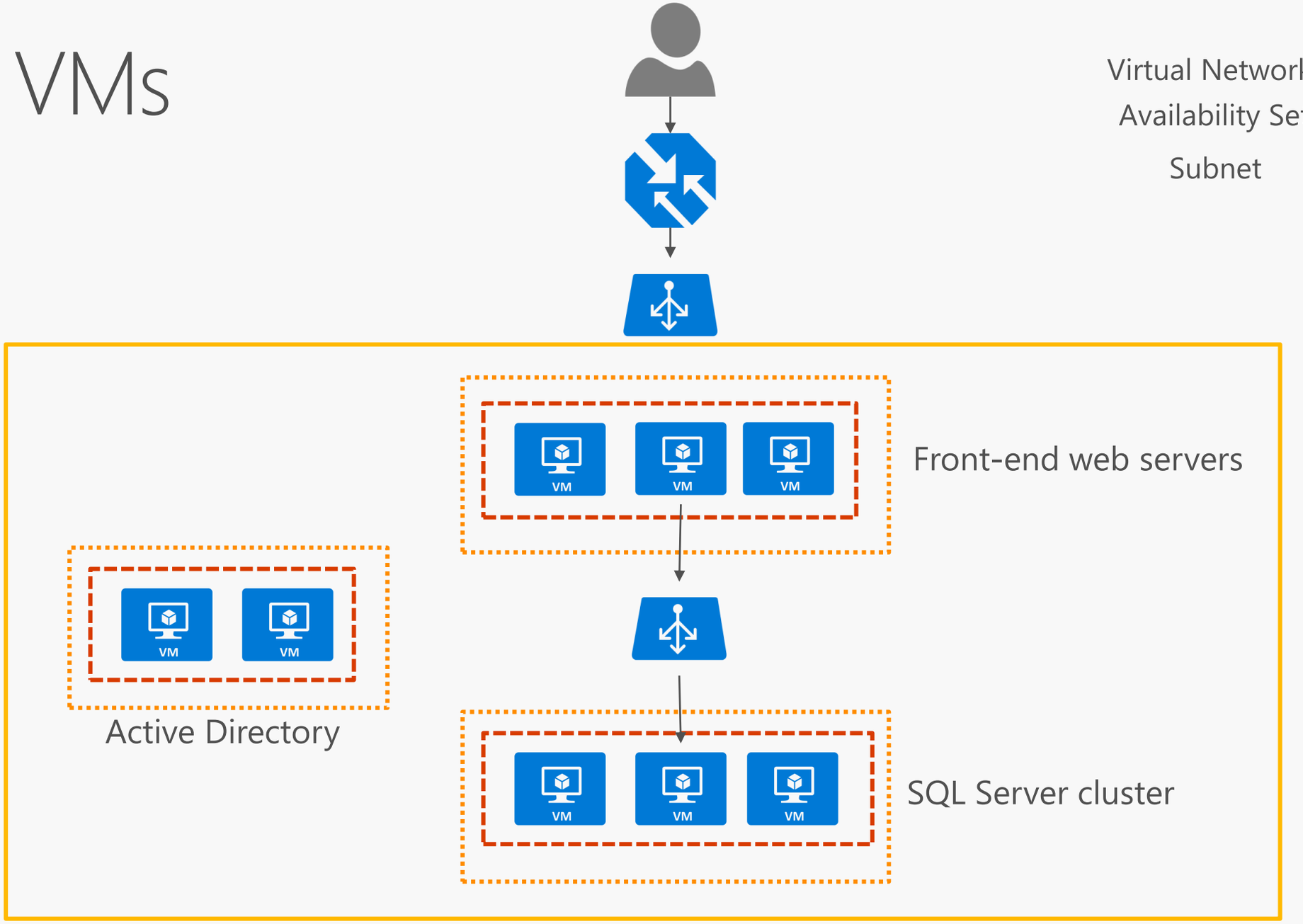


Azure VMs



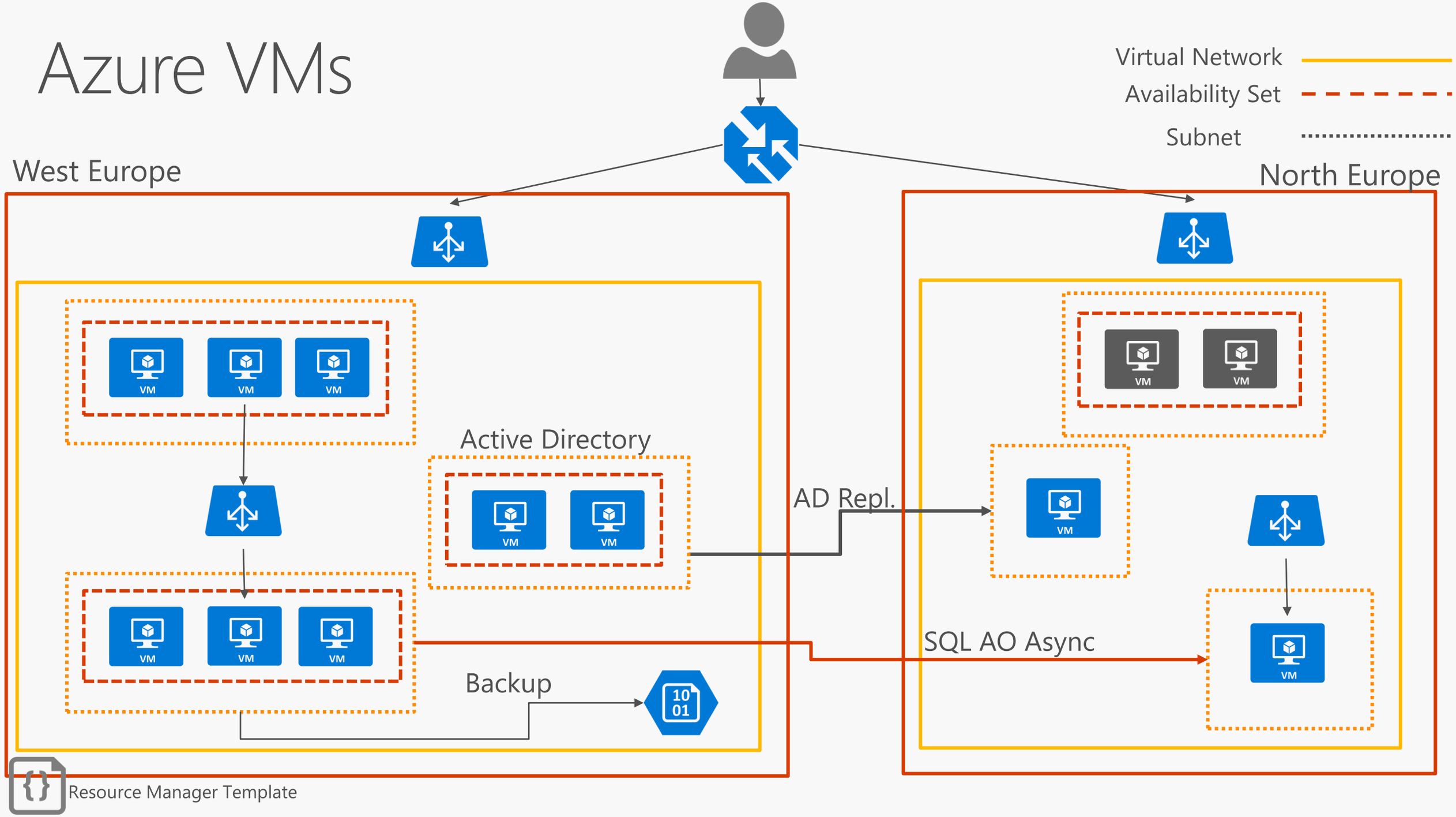
Resource Manager Template

Azure VMs



Resource Manager Template

Azure VMs


Virtual Network
Availability Set
Subnet
West Europe
North Europe
VM
Active Directory
AD Repl.
SQL AO Async
Backup
10
01
Resource Manager Template

Virtual Machine Scale Sets (VMSS)

A set of VMs which are identical (cattle)

Deployed with configuration management techniques

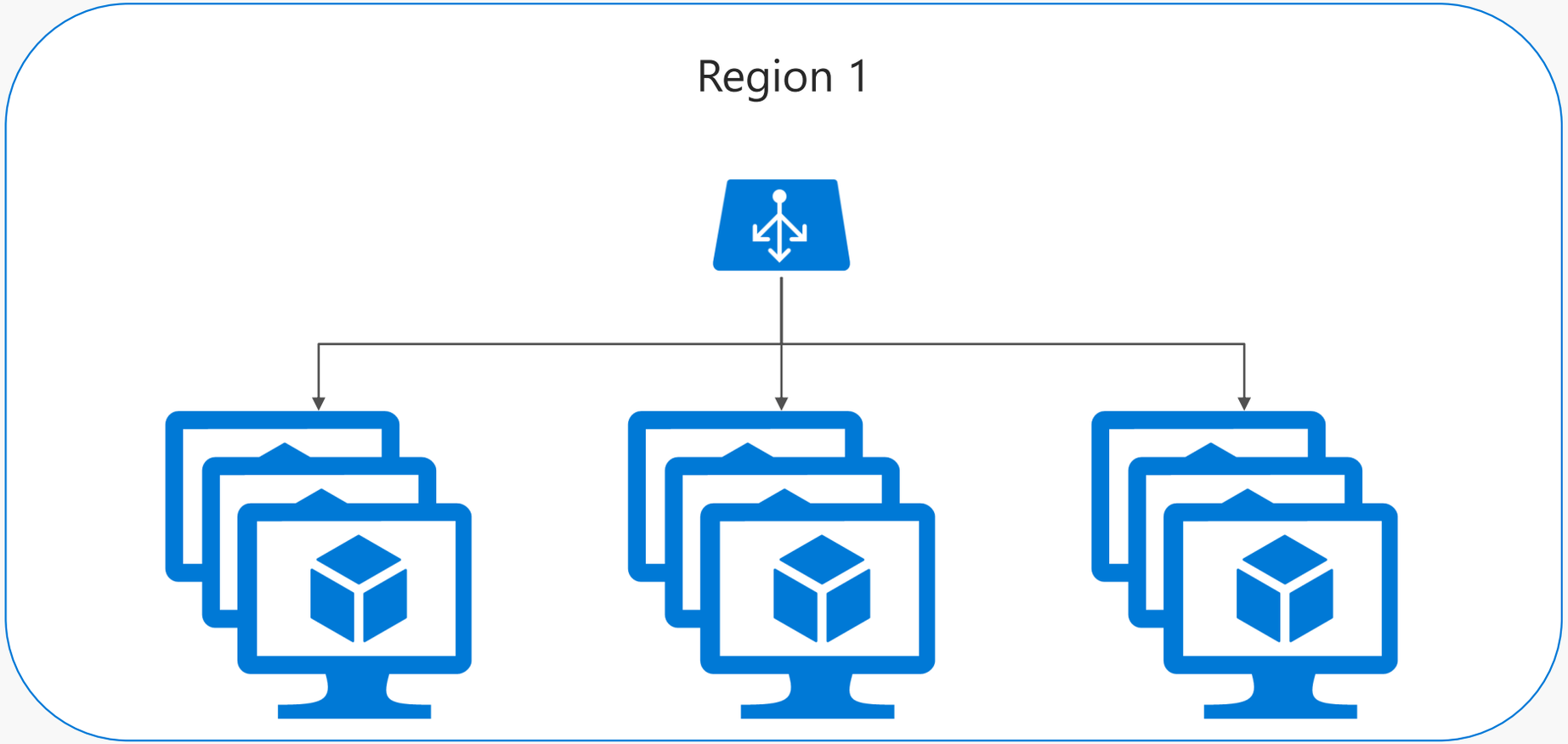
Can be auto-scaled

Based off a common base image

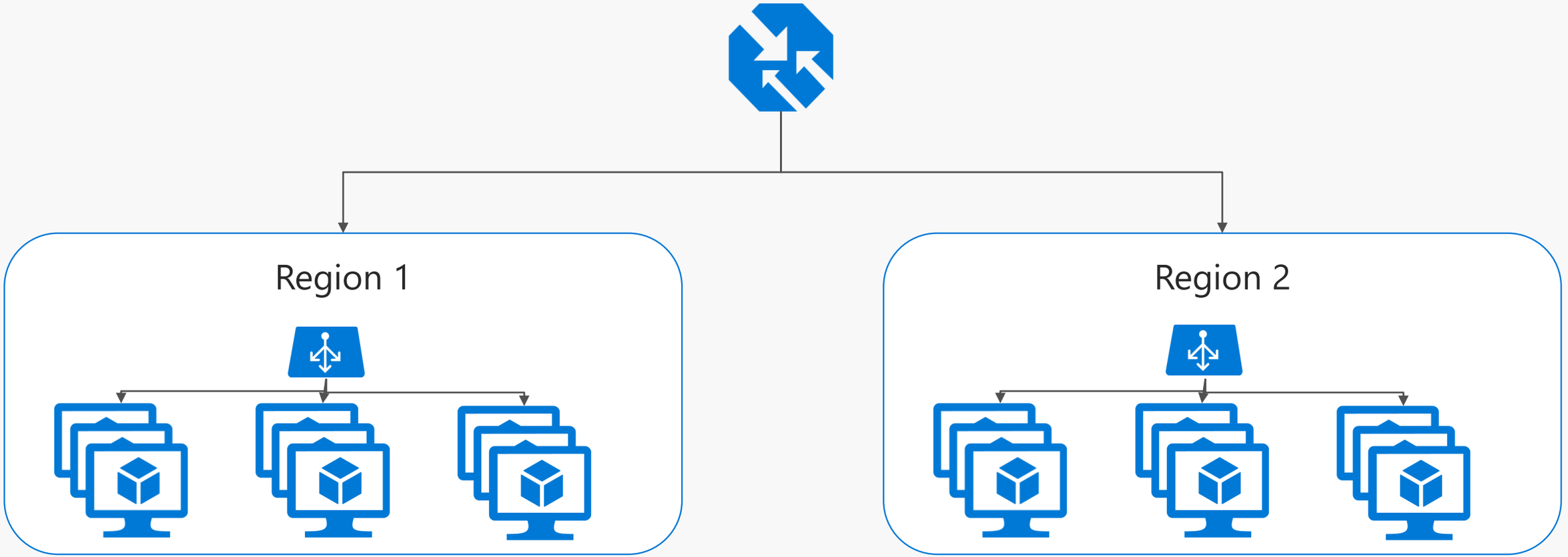
Portal / ARM template

Configuration management

Deployment scale - VMSS



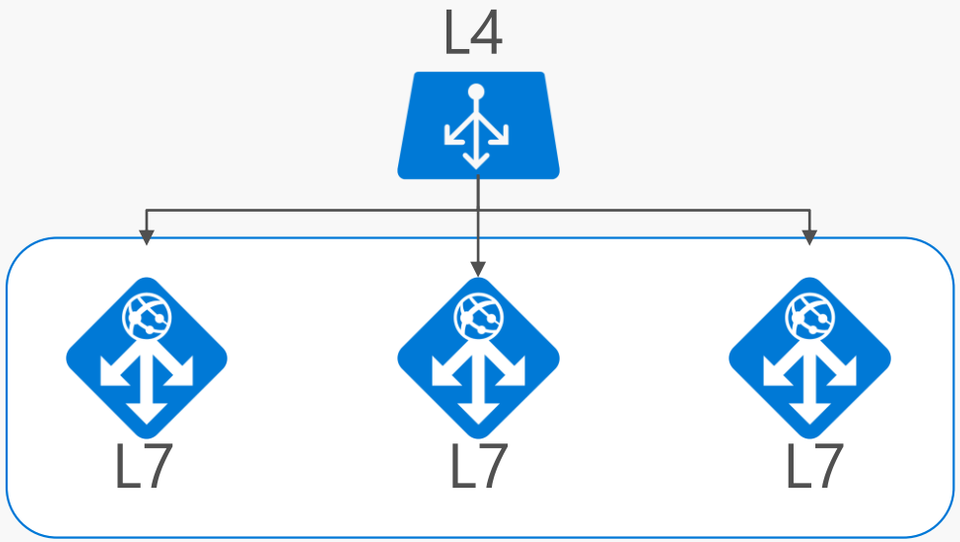
Deployment scale – multi-region



Network

Layer 4 vs layer 7 Load balancers

Type	Azure Load Balancer	Application Gateway
Protocols	UDP/TCP	HTTP, HTTPS, and WebSockets
Load balancing mode	5-tuple(source IP, source port, destination IP, destination port, protocol type)	Round Robin Routing based on URL
SSL offloading	Not supported	Supported



Demo
Stages of availability

What do you do if a region fails?

a) Use a Load Balancer to shift to additional resources

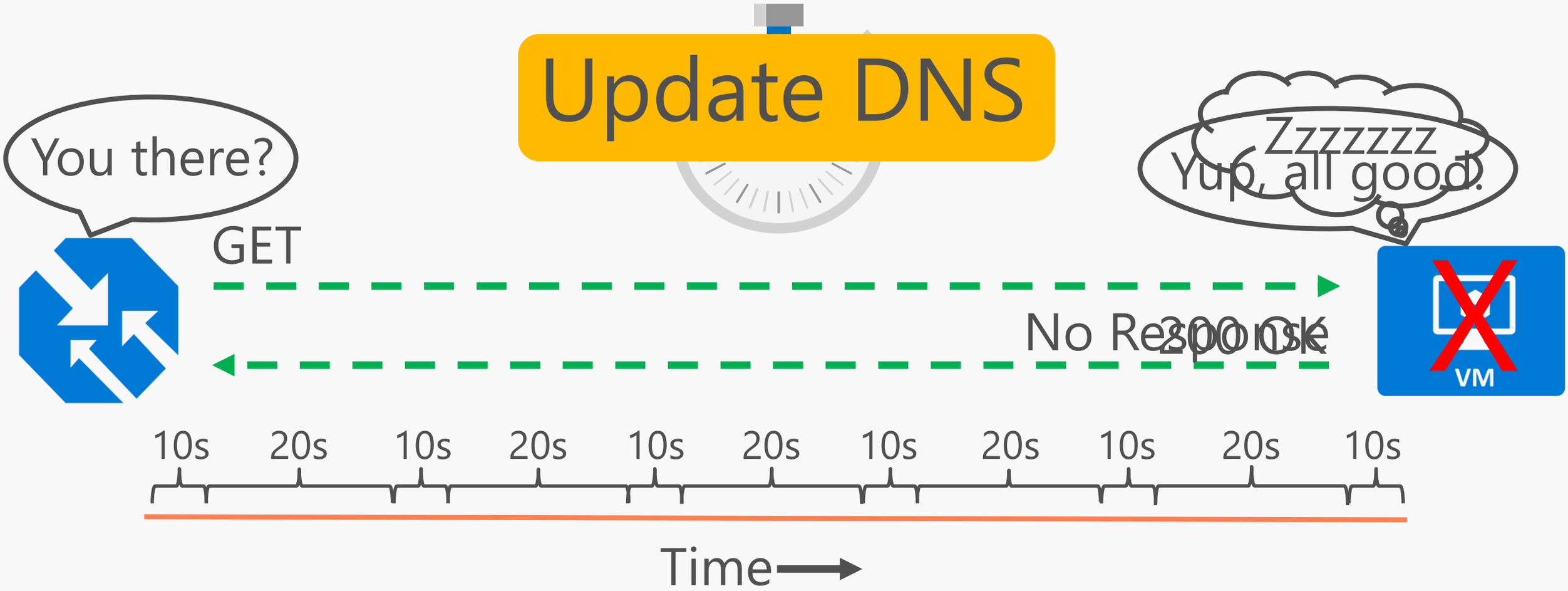
b) Use Azure DNS to point to a paired region

c) Use Traffic Manager to change regions

d) Initiate your failover runbook in Azure Site Recovery

e) Complain to @Azure on Twitter

How Traffic Manager handles endpoint loss



Traffic Manager

DNS-based load balancing and failover

Health probe of endpoints to determine availability

Impact of Time To Live (TTL)

Multiple performance policies

- Performance
- Priority
- Weighted

Storage

Storage

Types of storage

- **Locally redundant (LRS)** - 3 local copies strongly consistent
- ~~**Zone redundant (ZRS)** – 3 copies across 2 or 3 facilities~~
- ~~**Geo-redundant (GRS)** – 3 local copies strongly consistent, 3 remote copies eventually consistent~~
- **Read access geo-redundant (RA-GRS)** – Same as GRS but with read-only access of eventually consistent data

Provides tables, blobs, queues, and files

They provide:

- Isolation
- Replication
- Region order recovery
- Sequential updates
- Data residency

Primary	Secondary
North Central US	South Central US
South Central US	North Central US
East US	West US
West US	East US
West US 2	West Central US
West Central US	West US 2
US East 2	Central US
Central US	US East 2
North Europe	West Europe
West Europe	North Europe
South East Asia	East Asia
Germany Northeast	Germany Central
Germany Central	Germany Northeast

Primary	Secondary
East Asia	South East Asia
East China	North China
North China	East China
Japan East	Japan West
Japan West	Japan East
Brazil South	South Central US
Australia East	Australia Southeast
Australia Southeast	Australia East
Canada Central	Canada East
Canada East	Canada Central
UK South	UK West
UK West	UK South

https://docs.microsoft.com/en-us/azure/best-practices-availability-paired-regions

Storage – hidden point of failure?

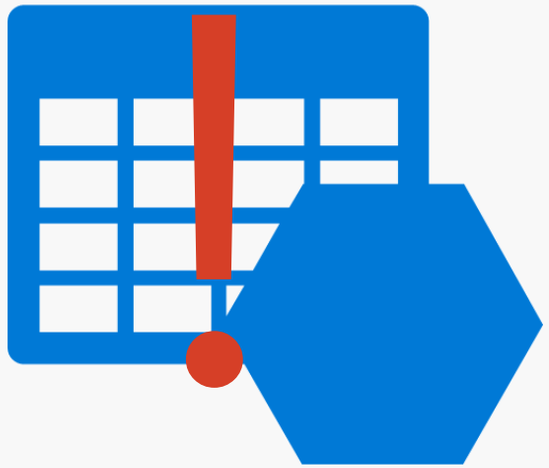
Standard vs Premium

Disks live in a storage account

A storage account exists on a storage stamp

A storage stamp is a single point of failure

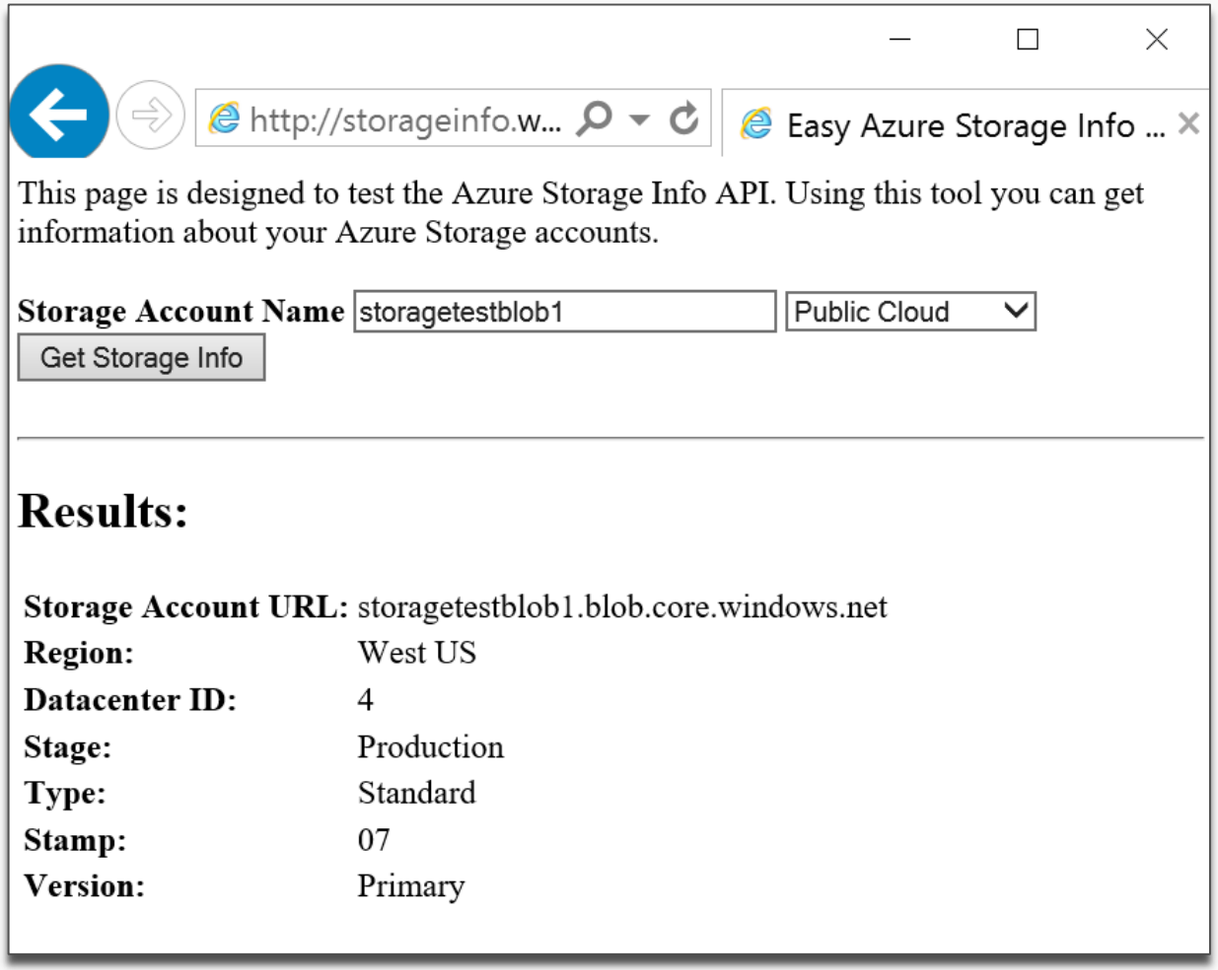
Storage outage can span fault domains



Demo:
Storage account information tool

Storage account information

http://aka.ms/StorageAccountInfo*



POST to (Yes, it's an Azure Function):

https://storageinfo.azurewebsites.net/api/StorageInfo

Post Content:

```
{"storageUrl" : "storagetestblob1.blob.core.windows.net"}
```

Response (in JSON):

```
{
  "Region":"West US",
  "Datacenter":"4",
  "Stage":"Production",
  "Type":"Standard",
  "Stamp":"07",
  "Version":"Primary"
}
```

**This tool is not endorsed, supported, or in any way officially endorsed by Microsoft. Use at your own risk.*

Send feedback to aglick@microsoft.com

Common architectures

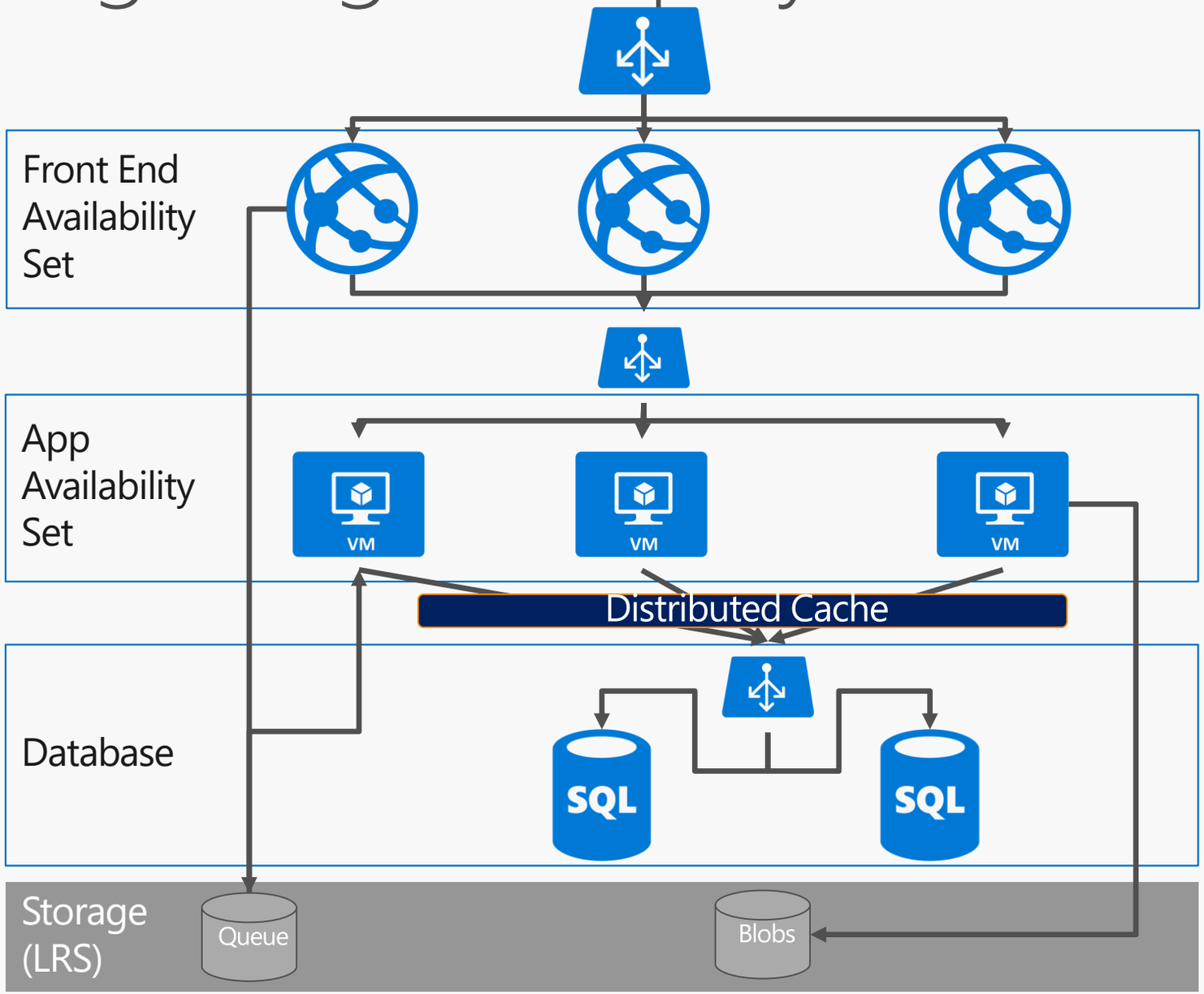
Consistency models

Strong consistency

Session consistency

Eventual consistency

Common single-region deployment- RDBMS



Multi-region deployment

Active/Passive

Only one region online with DR failover

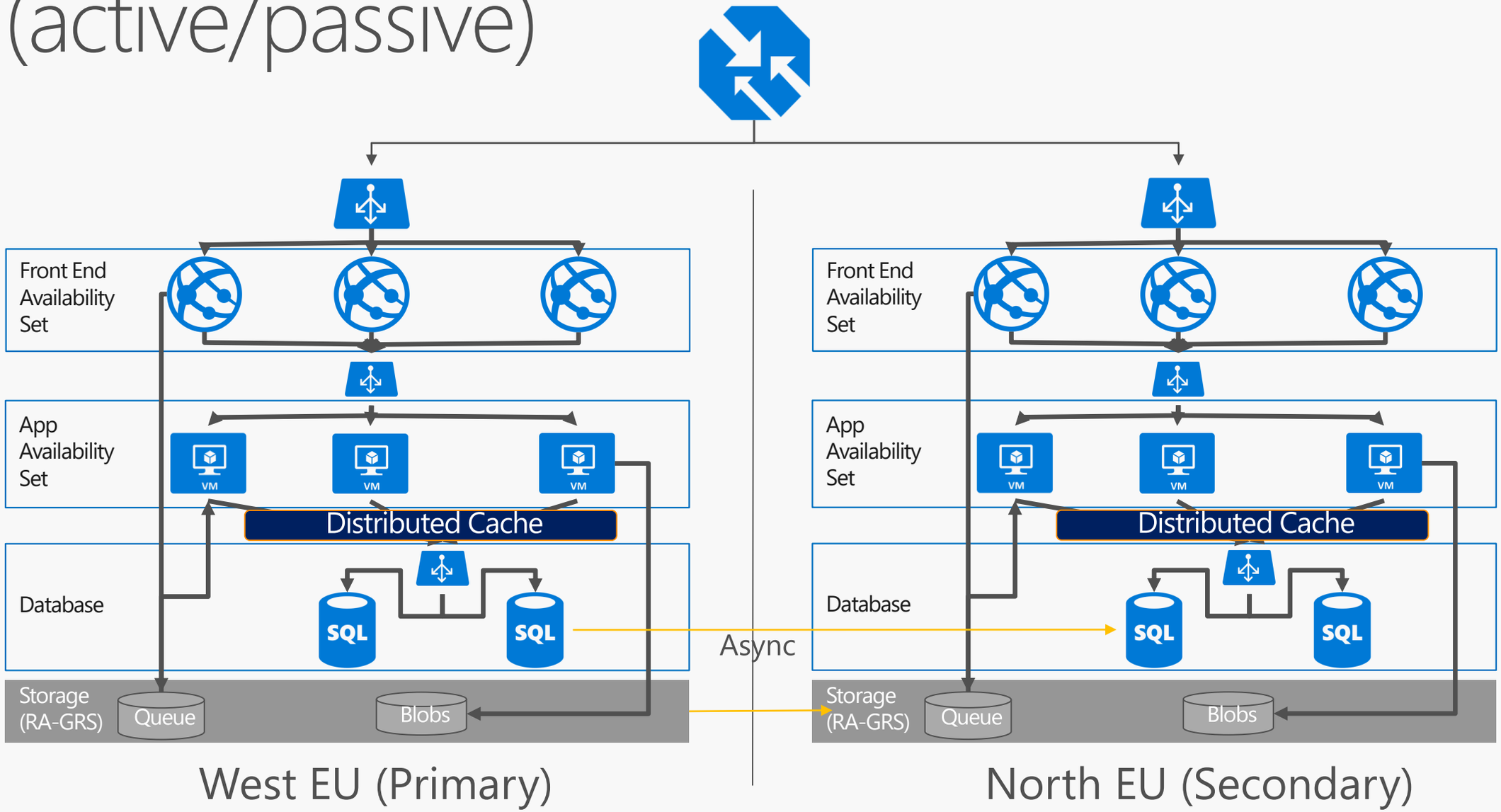
Active/Active

Round robin (all regions accept traffic)

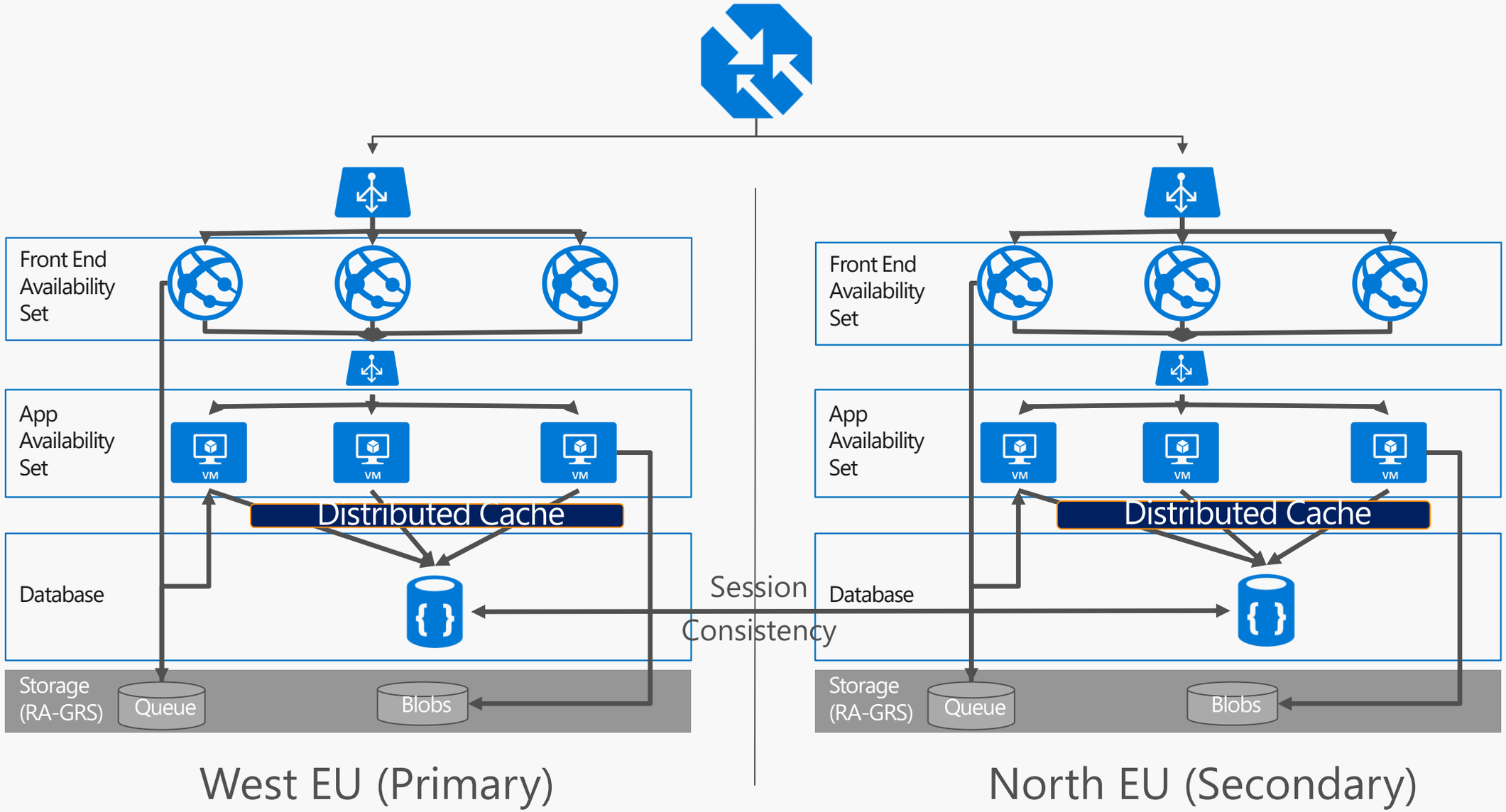
Active performance

Many regions, user is routed to nearest

Multi-region deployment for OLTP (active/passive)



Multi-region deployment (active/active)



Using a CDN for availability

Protection from DDoS

Greater availability and scale for your origin

Other benefits:

Lower latency content delivery

Faster data delivery

Lower data usage

Origin obfuscation

Demo:
Azure CDN

Chaos Engineering - Fault Injection

- Chaos Monkey
- Chaos Dingo
- Tests for single point of failure
- Automated architecture testing
- Simulates failure

Session objectives and takeaways

Session objective(s):

1. Design highly available applications on the Azure platform

2. Understand best practices and common pitfalls to avoid

3. Learn the inner workings of Azure and how to best work around hidden limitations

Note: *This session focuses mostly on the design of Cloud-based apps using IaaS.*

Resources

Azure High Availability

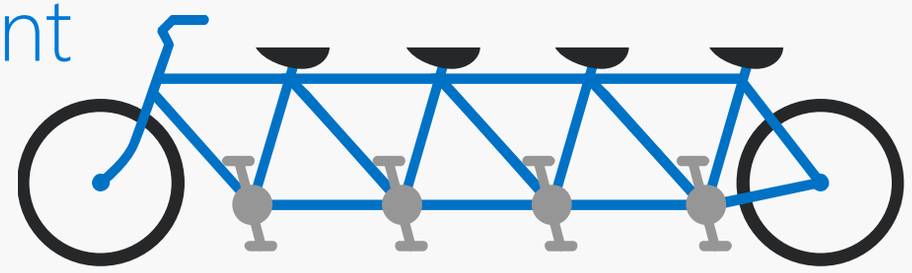
High Availability Checklist – https://aka.ms/AzureHaChecklist

Designing for High Availability – https://azure.microsoft.com/en-us/documentation/articles/resiliency-high-availability-azure-applications/

Overview of HA and DR - https://azure.microsoft.com/en-us/documentation/articles/resiliency-disaster-recovery-high-availability-azure-applications/

High availability checklist

- [x] Use Traffic Manager
- [x] Avoid single VMs
- [x] Use load balancers in front of web-facing VMs
- [x] Put your stateless servers in Availability Sets
- [x] Use VMSS for your stateless server scaling
- [x] Use Premium Storage for your production VMs
- [x] Use internal load balancers (or queues) between tiers
- [x] Distribute your database
- [x] Use caches
- [x] Contact support before a high scale event
- [x] Store static assets in Blob Storage
- [x] Use a CDN in front of your static assets



https://aka.ms/cc9cf1

Thank You