SYNERGETICS GET IT RIGHT | Microsoft Partner
Silver Cloud Platform
Silver Learning

Microsoft

Cloud
Best Practices & Anti Patterns

What are the things the Cloud brings that makes it different?

What new things to we need to take into account when building for the Cloud?

What should we avoid when *going Cloudy*?

To boldly go...

The Cloud awakens…

The fundamental characteristics of IT in the post Cloud world are different!

There are new opportunities for you to learn to utilize and love!

Why are we looking to *"do"* this Cloud thing right?

- Throughput
- Latency
- Density
- Manageability
- Availability

Throughput

How to increase the number of operations
- transactions, service calls, … -
through the system and to reduce contention.

Latency

How to reduce the time we wait for and on, individual operations and service calls.

Density

How to enable us to run efficient software systems
on as few resources as possible
while maintaining high availability and reliability.

Manageability

How to understand the health and performance of your deployed services at scale.

Diagnostics, telemetry and other insights

Availability

How to reduce the impact of failures,
how to live in failure modes
and still give a great UX.

- Throughput
- Latency
- Density
- Manageability
- Availability

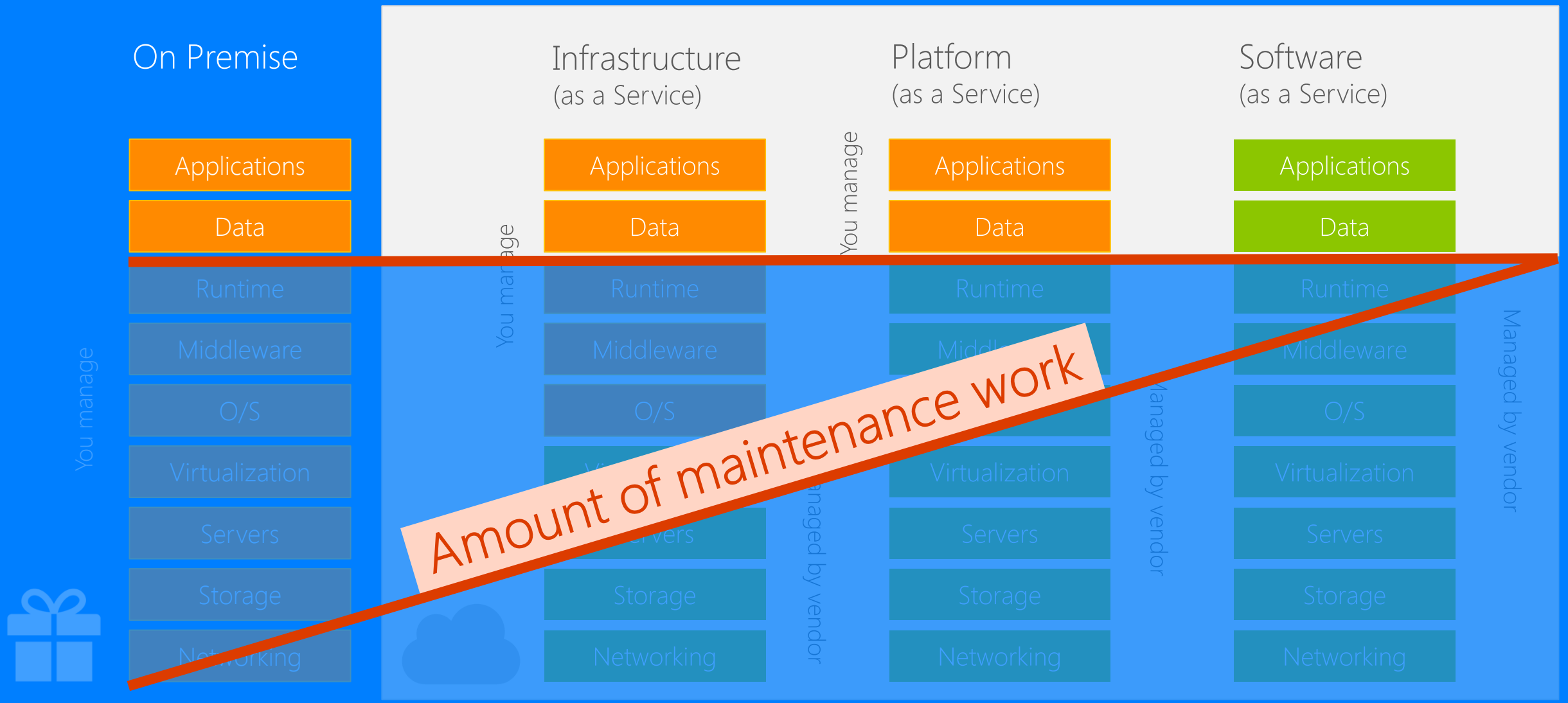
Migrating to the Cloud?!

"Lift and Shift"

vs.

TCO

Cloud Computing



"Microsoft believes that PaaS provides *the best* foundation for creating, running and managing custom applications."

bit.ly/ITasaService – Microsoft 2010

Migrate to PaaS

Microservices

The new Black of Platform as a Service

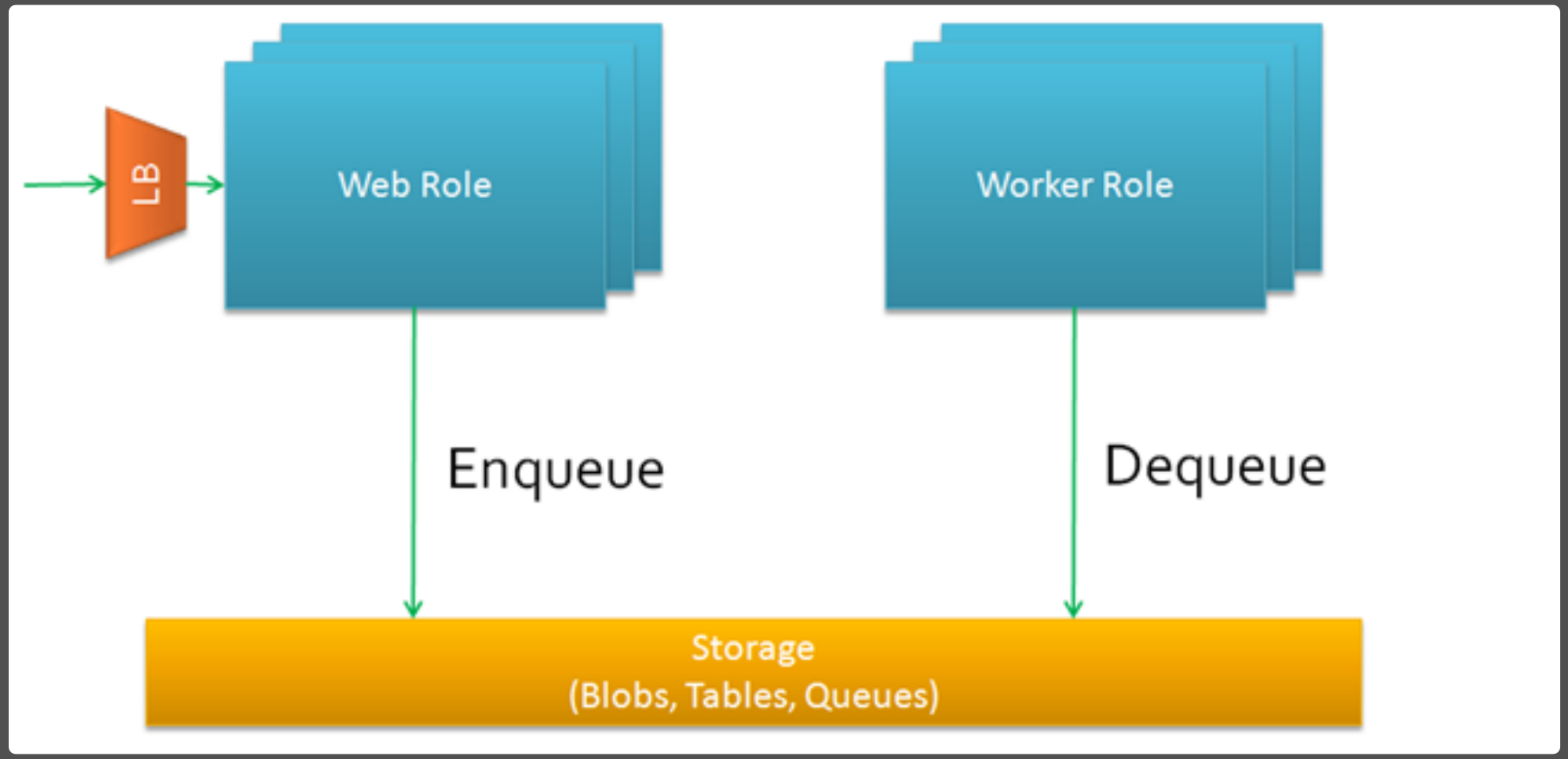
Best Practice: Servers

Don’t treat your servers like ~~pets~~!

Treat them like *cattle*!

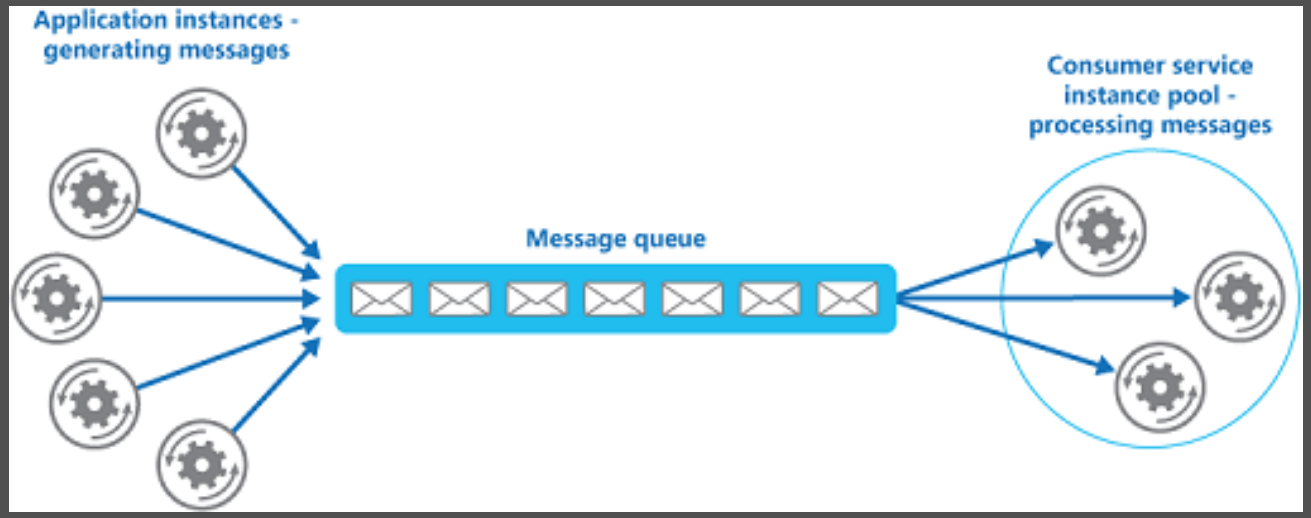
Queueing

Queueing is important!

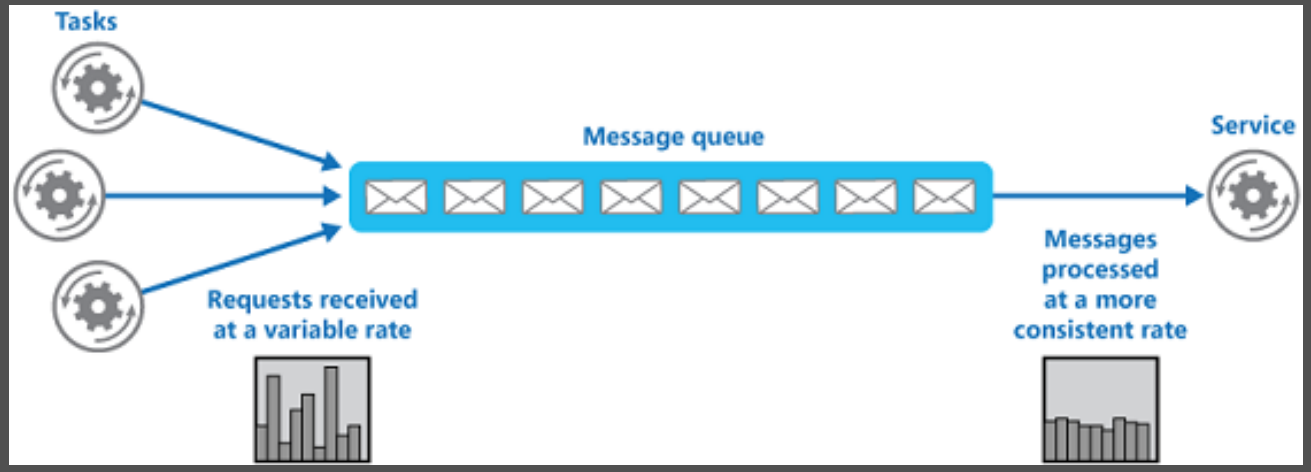


Solves reliability, availability and scalability!

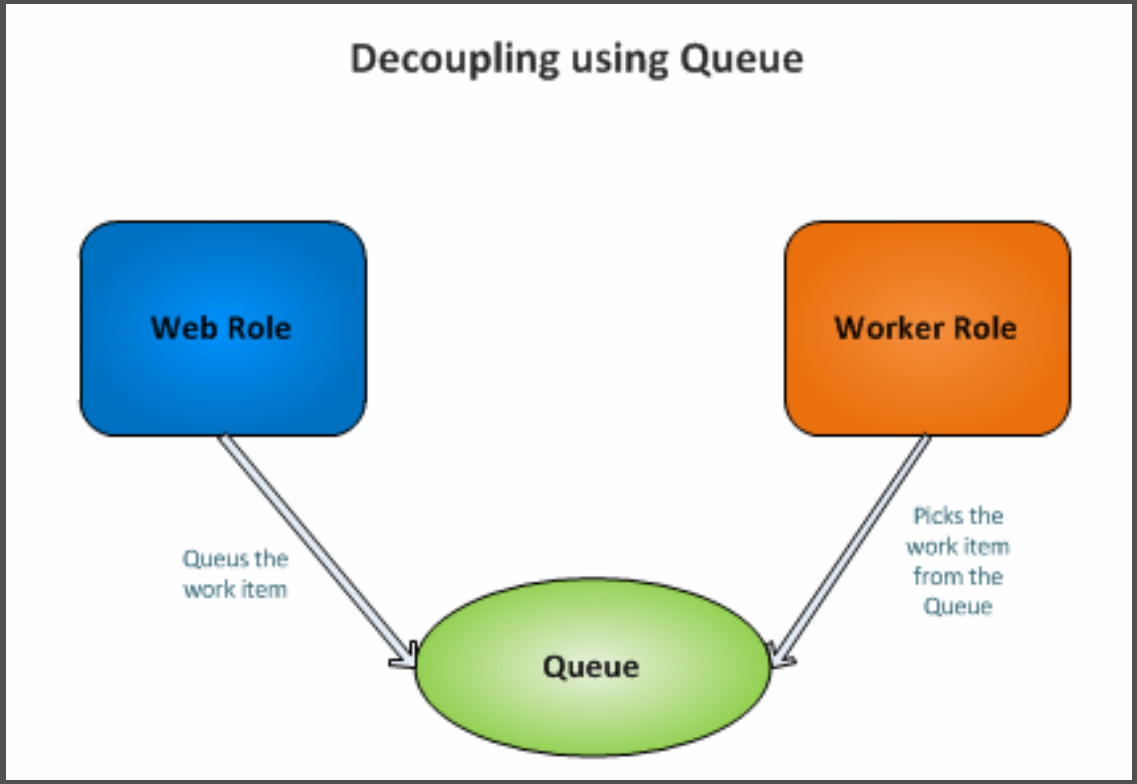
Competing Consumers Pattern



Queue-based Load Leveling Pattern



Queues bring decoupling



Scalability

Scaling Options

Scale Up / Vertical Scaling

Increase resource capacity within existing node

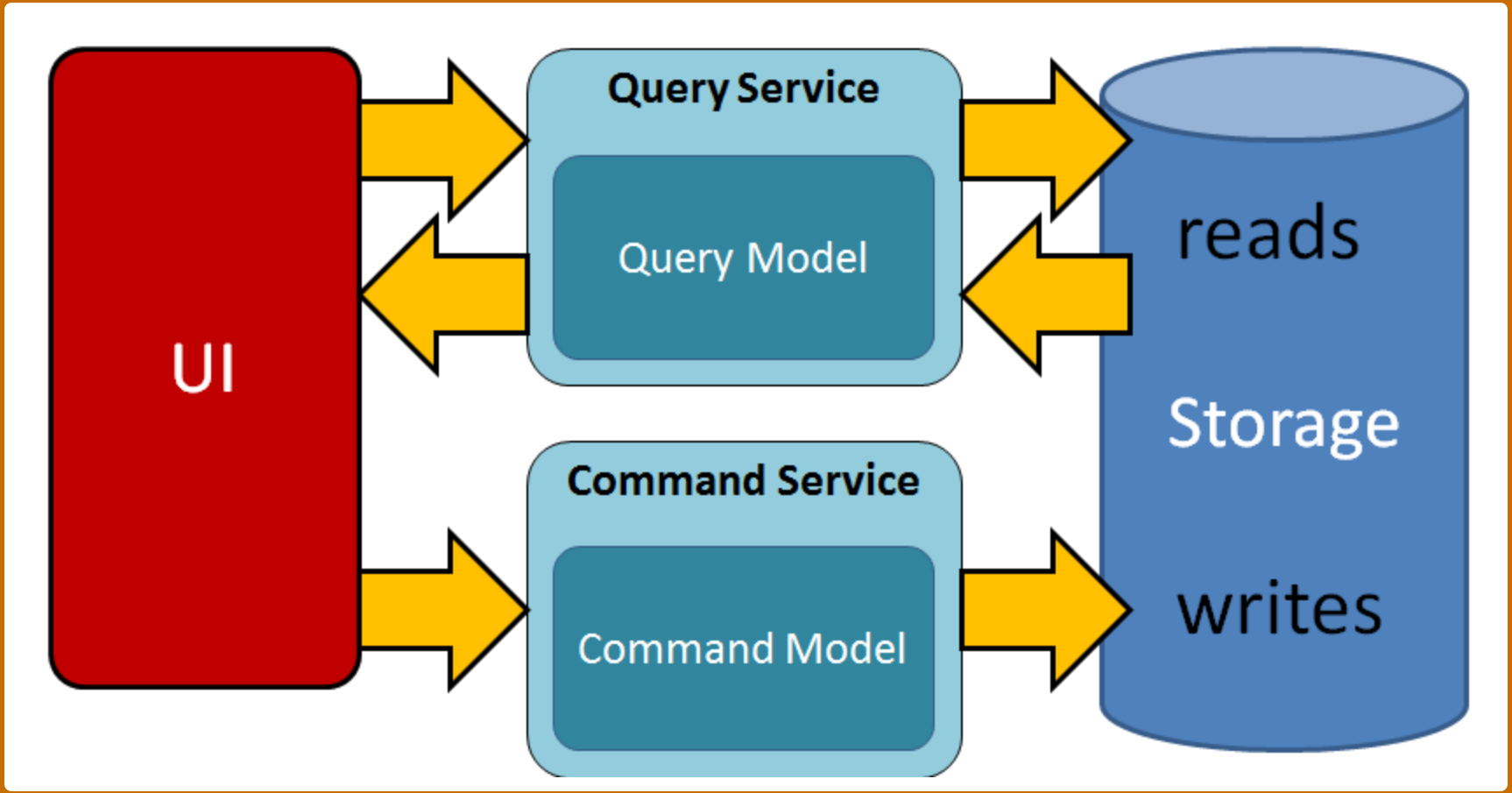
Scale Out / Horizontal Scaling

Increase resource capacity by adding nodes

Optimize for reads

Learn to live with stale data

Command and Query Responsibility Segregation (CQRS)



The *"teenage sex"* of Software Architecture!

Illustration: David Pallmann

Use queues to move long running tasks to the background

Use async programming models

Make sure you instrument your code to learn what's going on in production!

Almost Flying in the Clouds
(anti-pattern)

Wile E. Coyote by Chuck Jones © Looney Tunes and Merrie Melodies

Building applications today
is becoming less about actually building services
and more about bolting them together!

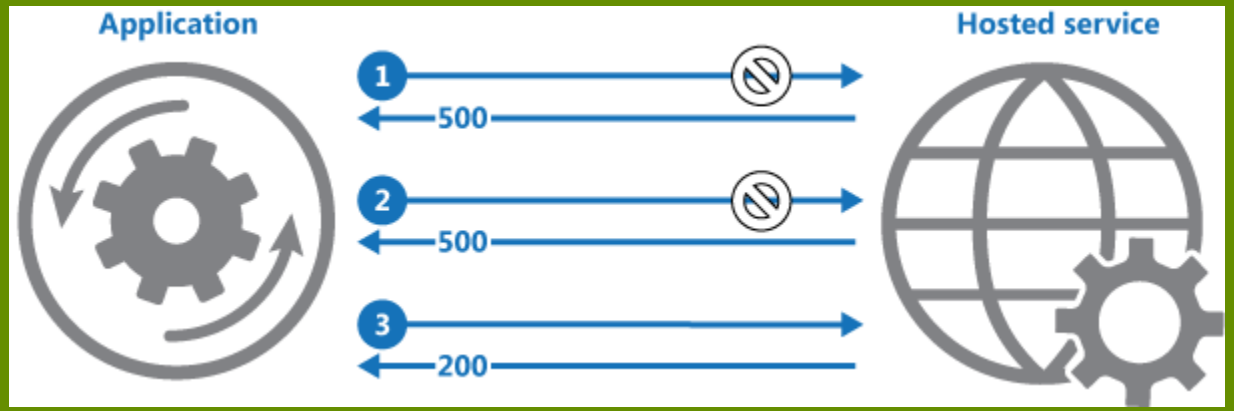
Chris Auld - Intergen

Don’t build – rent!
Spread your risk
High availability

You must code asynchronously, defensively, use patterns like the queue load leveling pattern and handle sub-system failures!

Understand that you must async/await!

Understand that you must retry!



Understand that you must code for contingencies!

Measure then cut!

Make sure you start to measure!

(How else will you know if you improve?)

Test Early/Fail Fast

Implement a skeleton solution and set it up for *automated* deployment to Azure!

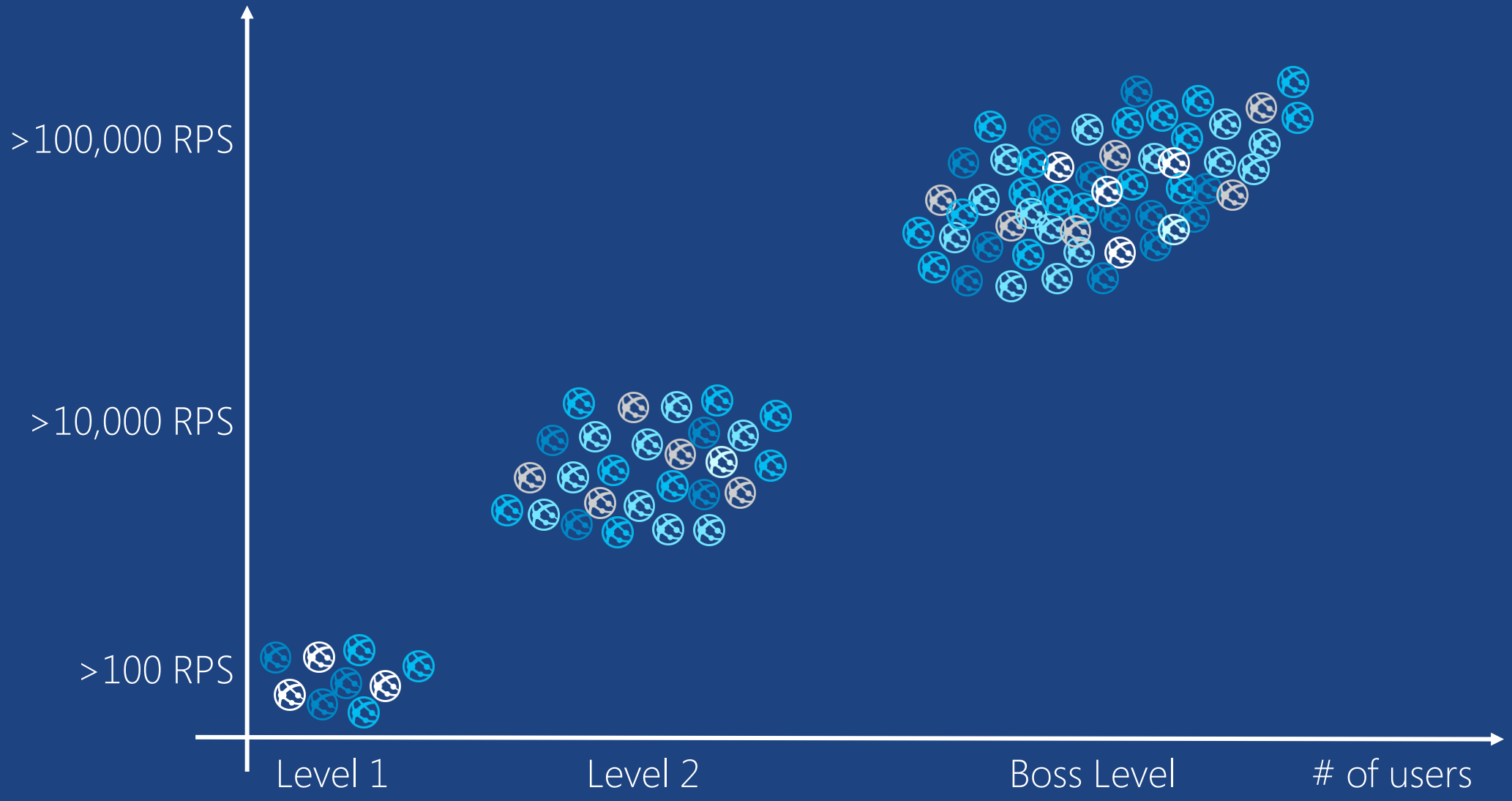
Horseless Carriage

Continuous Delivery

Architecting for Scale

A web app's journey towards scalability

A web app's journey towards scalability



Architecting for Scale

Level 1: Beginner

Create websites in less than a minute

Use your existing tools and knowledge

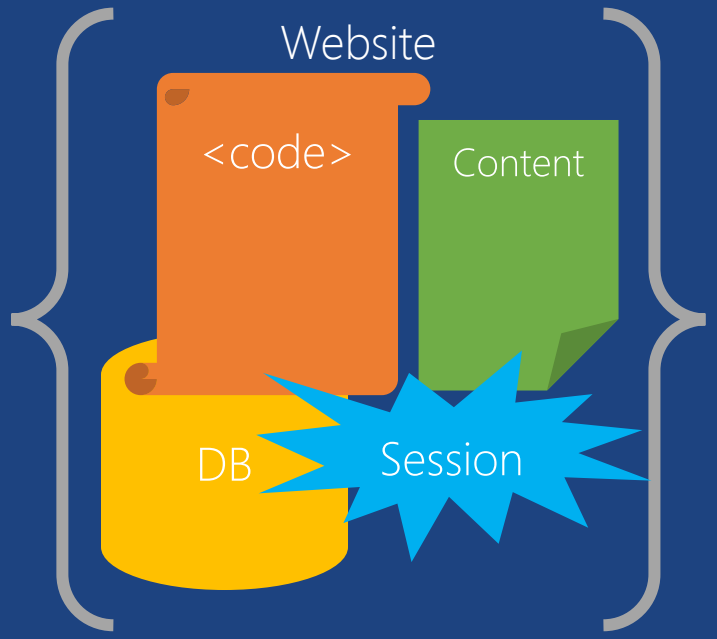
Integrate with other services you're already using

Supports a variety of stacks

Demo

Our “little” Application

Level 1 – Photo Gallery (untouched)



Database stored on local disk

Session state stored in local memory

Images stored on local disk

Stateul Application

Level 1 – Photo Gallery (untouched)

Scale Testing Results:

Scale Count	Scale Size	Test Duration	Max User Count	Avg Pages/Sec	Avg Page Time (sec)	Avg RPS (Visual Studio)	WebSites RPS	Failed Tests	Total Tests	Failed Tests (%)
1	MED	15	2	**0.27**	**1.16**	**1.77**	1	0	77	0.0
1	MED	15	20	1.9	6.5	14.2	18	0	559	0.0
1	MED	15	50	**3.48**	**10.4**	**39.5**	45	1	980	0.1
1	MED	15	80	**5.58**	**12.5**	**51.5**	55	0	1639	0
1	MED	15	120	**4.95**	**22.5**	**47.8**	50	19	1446	**1.3**

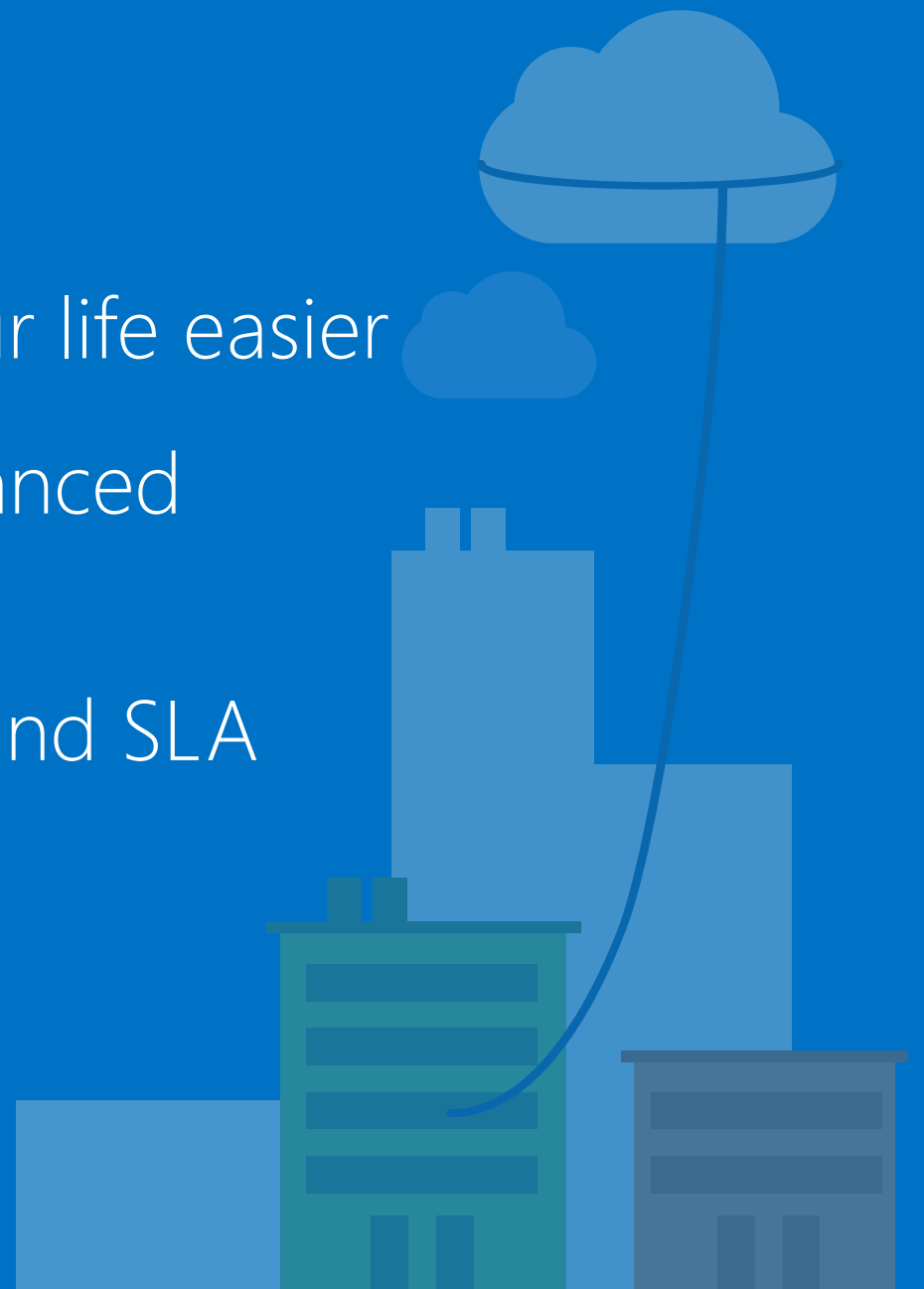
Architecting for Scale

Level 2: Expert

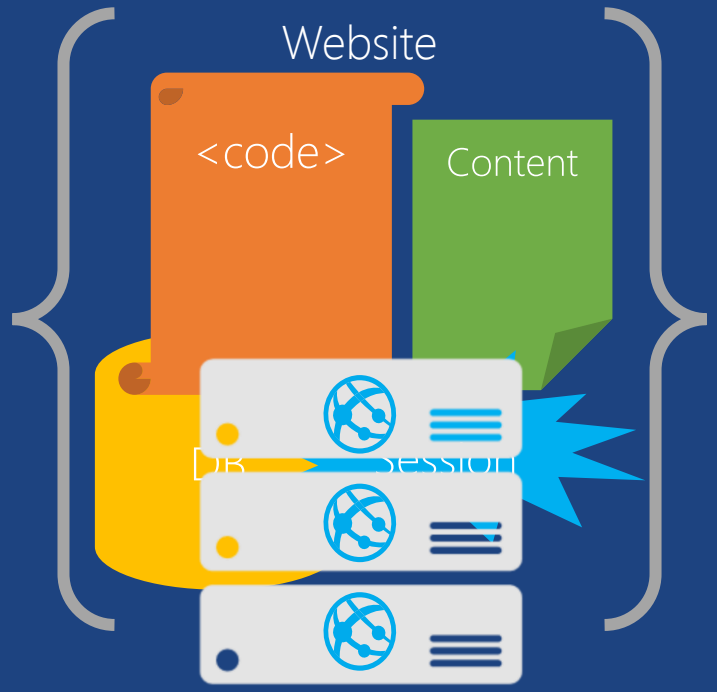
A rich set of functionality that will make your life easier

Features that are easy to use, even for advanced operations

Save money while running at higher scale and SLA



Level 2: Horizontally Scalable Photo Gallery



Demo

Our Application is growing up!

Level 2: Horizontally Scalable Photo Gallery – Strategy Summary

Move database to cloud
(Azure SQL Database)

Move site content to blob
(Azure Storage)

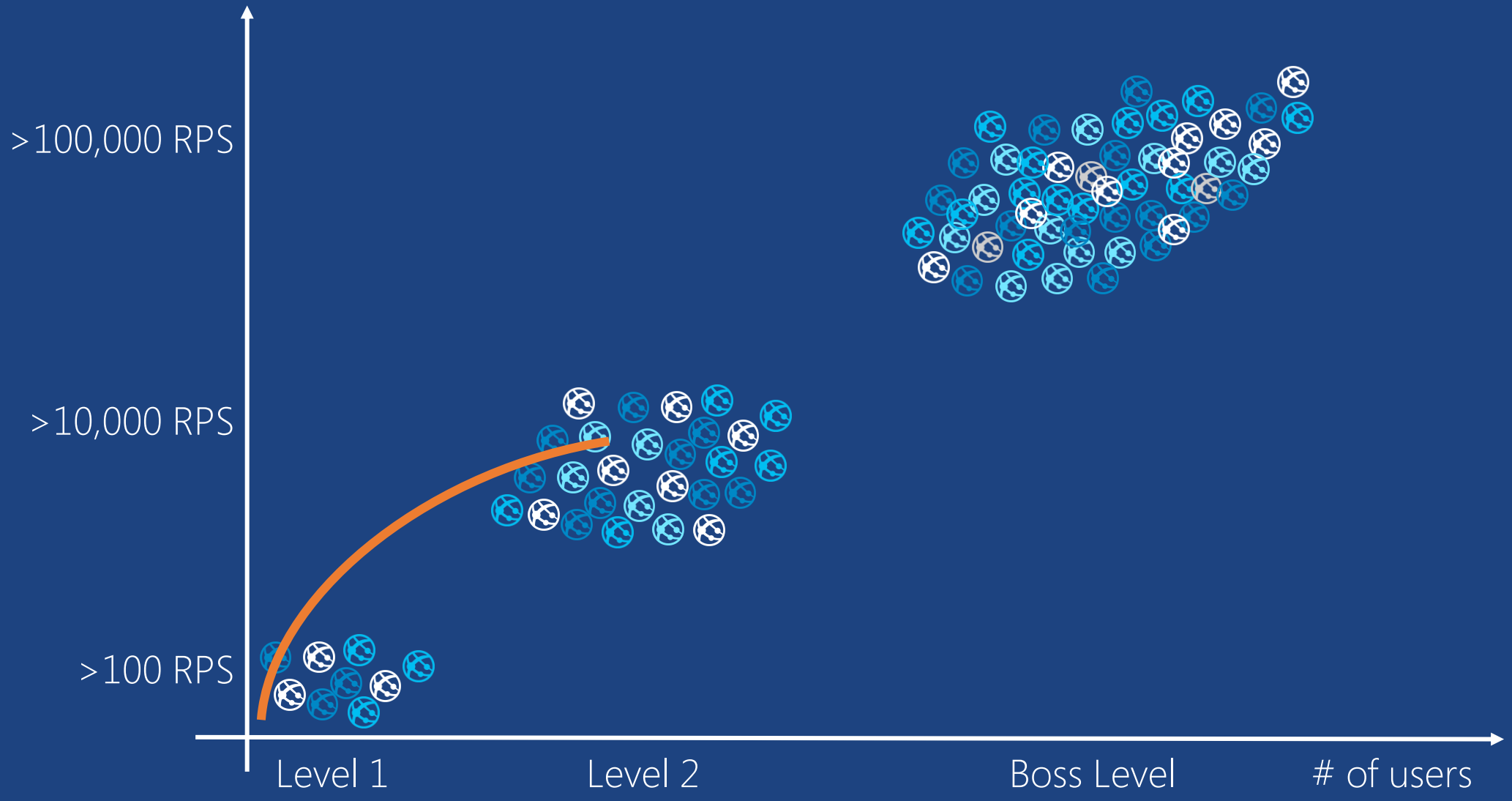
Make site stateless
Non-Sticky Session

Horizontal scale
Web Tier

Scale Testing Results

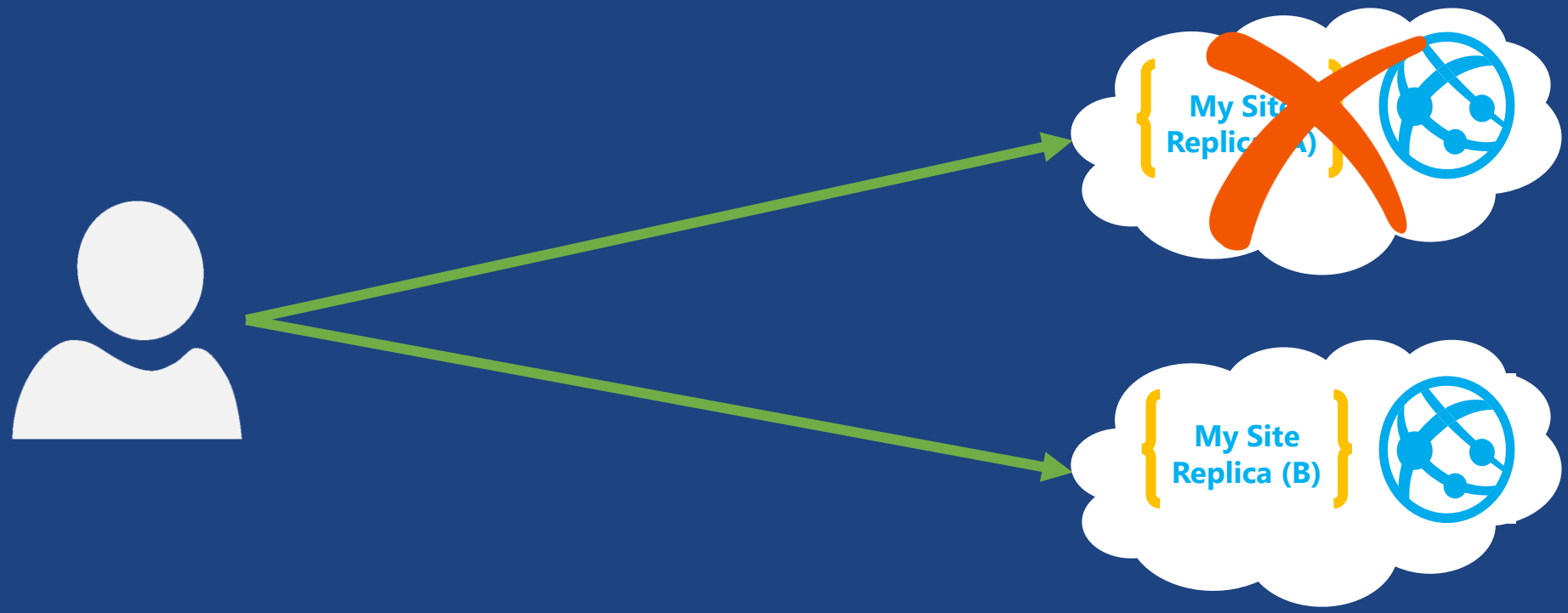
Scale Count	Scale Size	Test Duration	Max User Count	Avg Pages/Sec	Avg Page Time (sec)	Avg RPS (Visual Studio)	WAWS RPS	Failed Tests	Total Tests	Failed Tests (%)
1	LARGE	15	20	**2.91**	**0.13**	23.4	21	0	849	0.0
1	LARGE	15	100	**14.4**	**0.15**	232	77	0	4,247	0.0
1	LARGE	15	200	**29.2**	**0.14**	966	155	0	8,563	0.0
1	LARGE	15	300	**43.6**	**0.24**	2,535	231	0	12,839	0.0
1	LARGE	15	1,000	**141**	**0.67**	8,135	735	0	20,591	0.0
3	LARGE	20	1,500	**198**	**1.37**	19,855	1297	1	32,763	0.0
3	LARGE	25	2,000	242	**2.12**	24,896	1547	**870**	53,496	**1.6**

A web app's journey towards scalability



Level 3 – Globally Scalable Photo Gallery

When bad things happen to good datacenters



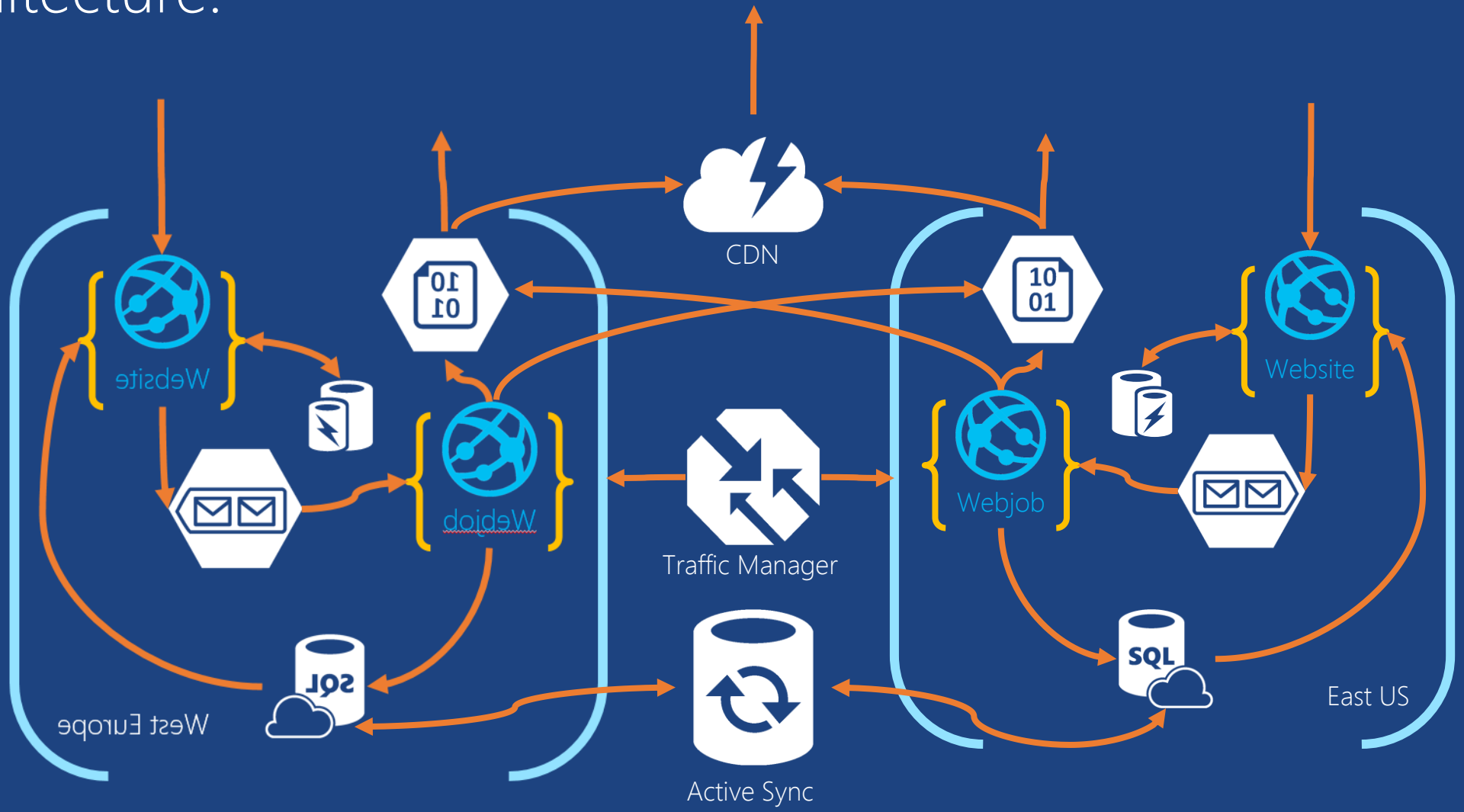
Level 3 – Globally Scalable Photo Gallery

Moving towards a global presence

Gaps	Goals	Benefits
Web app is ‘local’ to a single region Images stored in ‘local’ data center Database in ‘local’ data center	Active / Active replication across regions	Bigger Scale Lower Latency Higher Uptime and DR

Level 3 – Globally Scalable Photo Gallery

Architecture:



Level 3 – Globally Scalable Photo Gallery

Strategy:

Add Traffic Manager

Controlled and synchronized deployment with Site Slots

Asynchronous data processing with WebJobs

Copy images to remote regions

Use queue CQRS* to update DB

* Command Query Responsibility Segregation adds latency and app complexity

Level 3 – Globally Scalable Photo Gallery

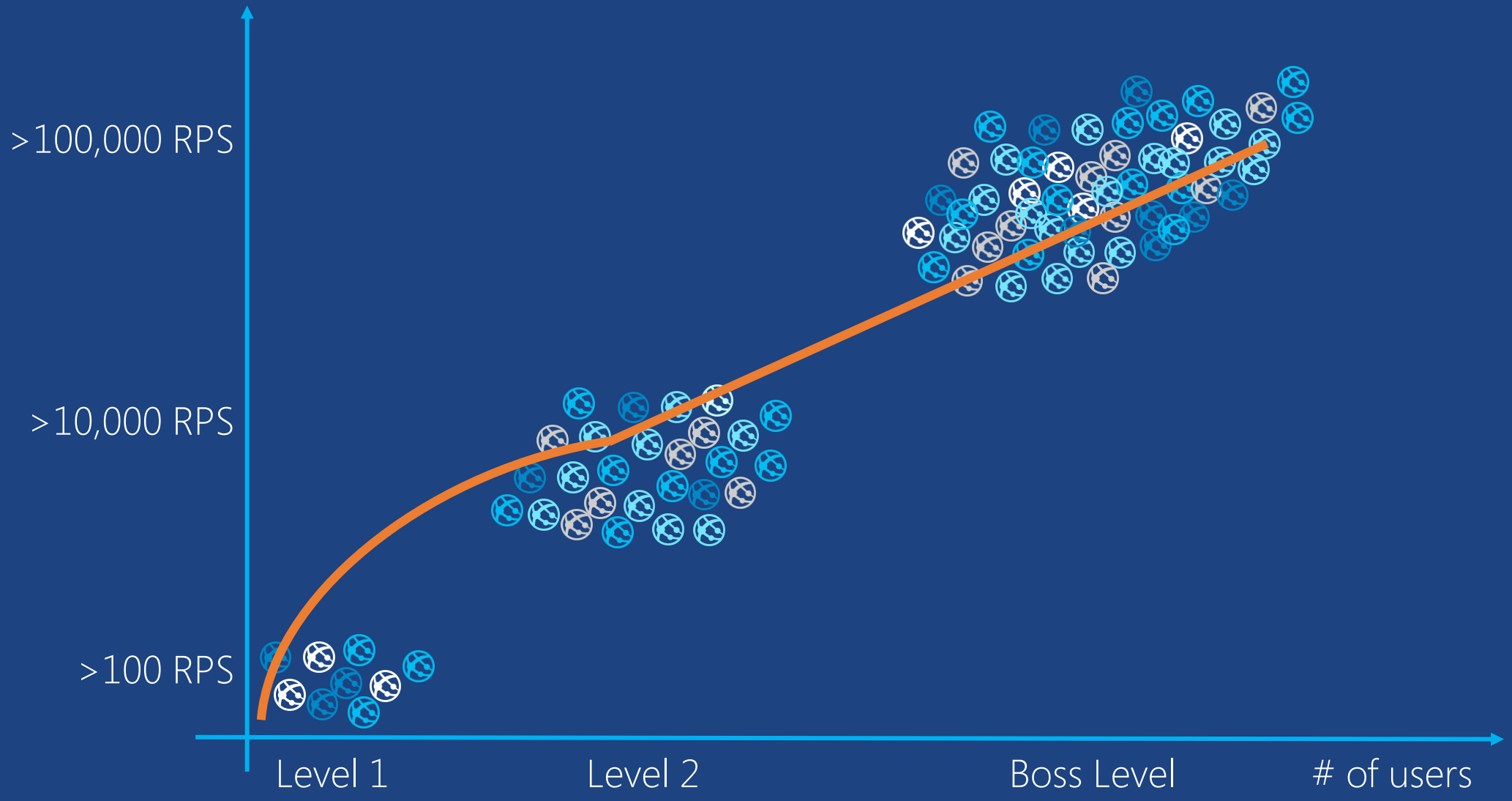
Words of caution:

Using a queue to sync DB access leads to eventual consistency

Syncing database works for ‘most’ scenarios

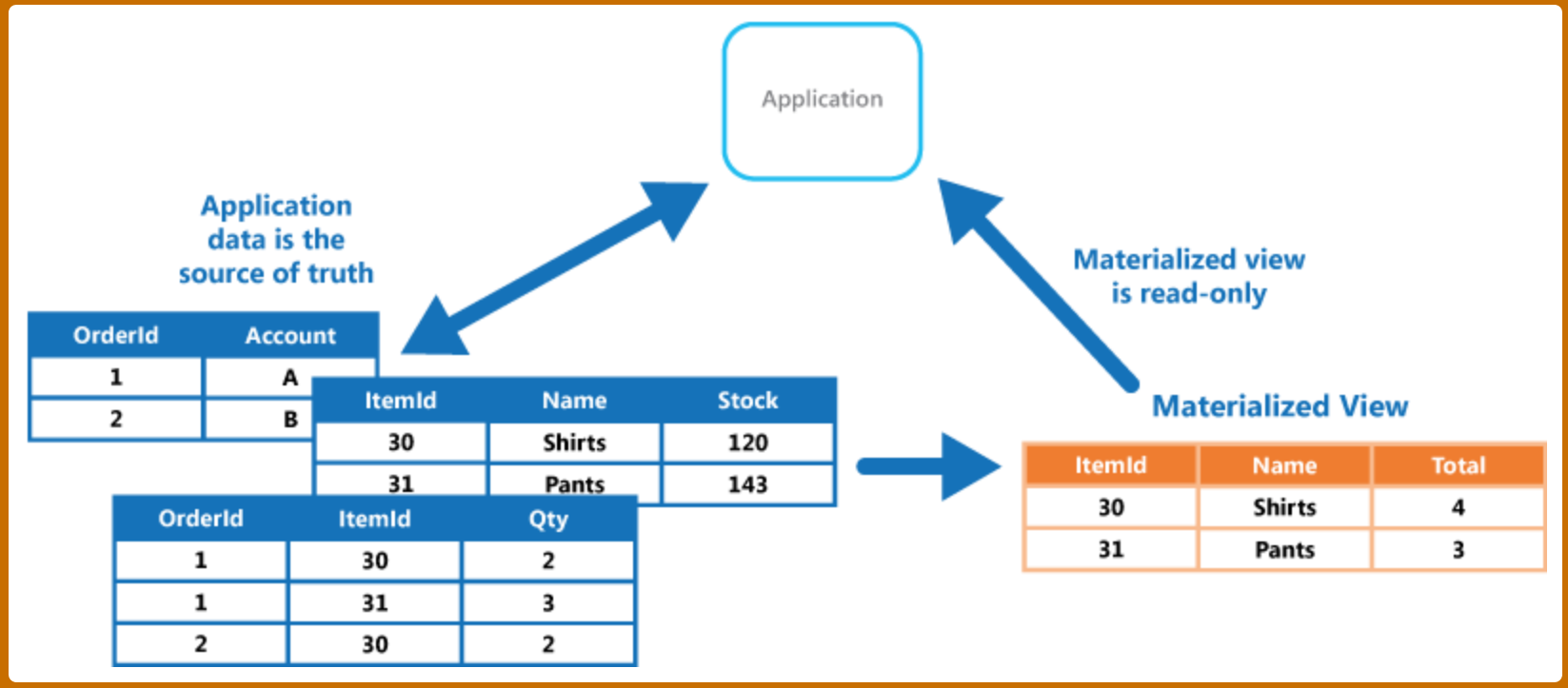
Active / Active state is app dependent

A web app's journey towards scalability

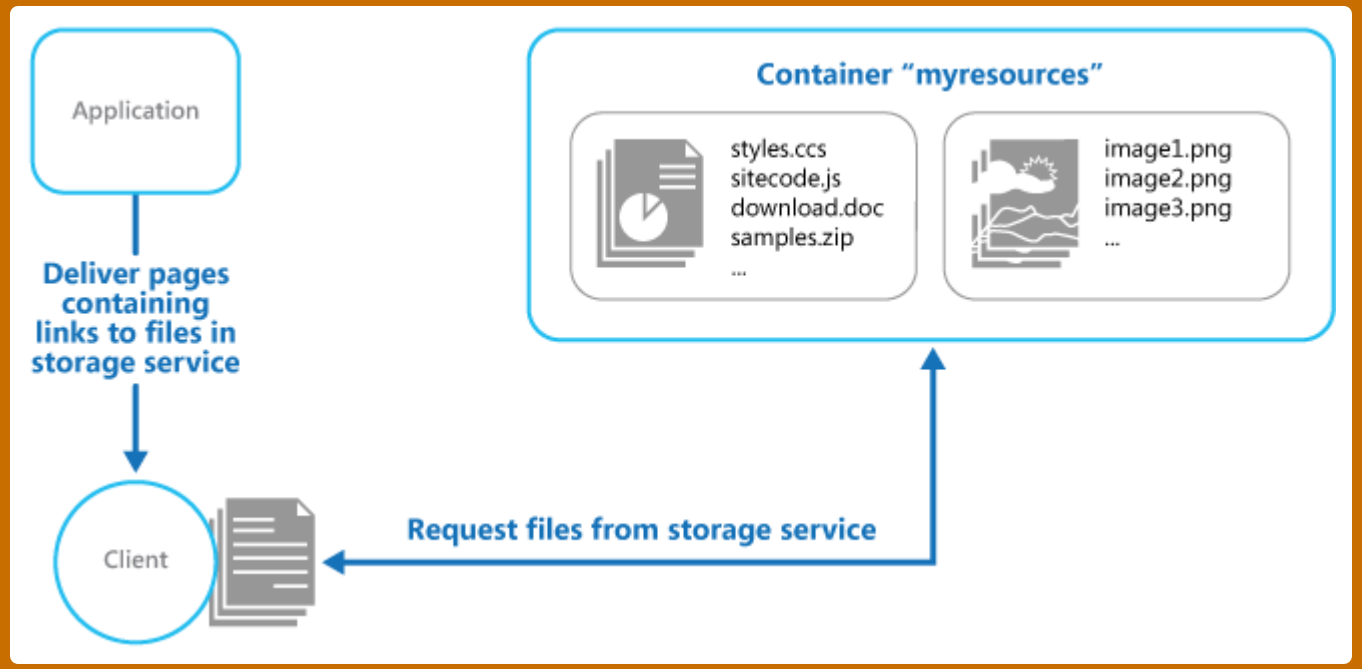


Azure Storage

Materialized View Pattern

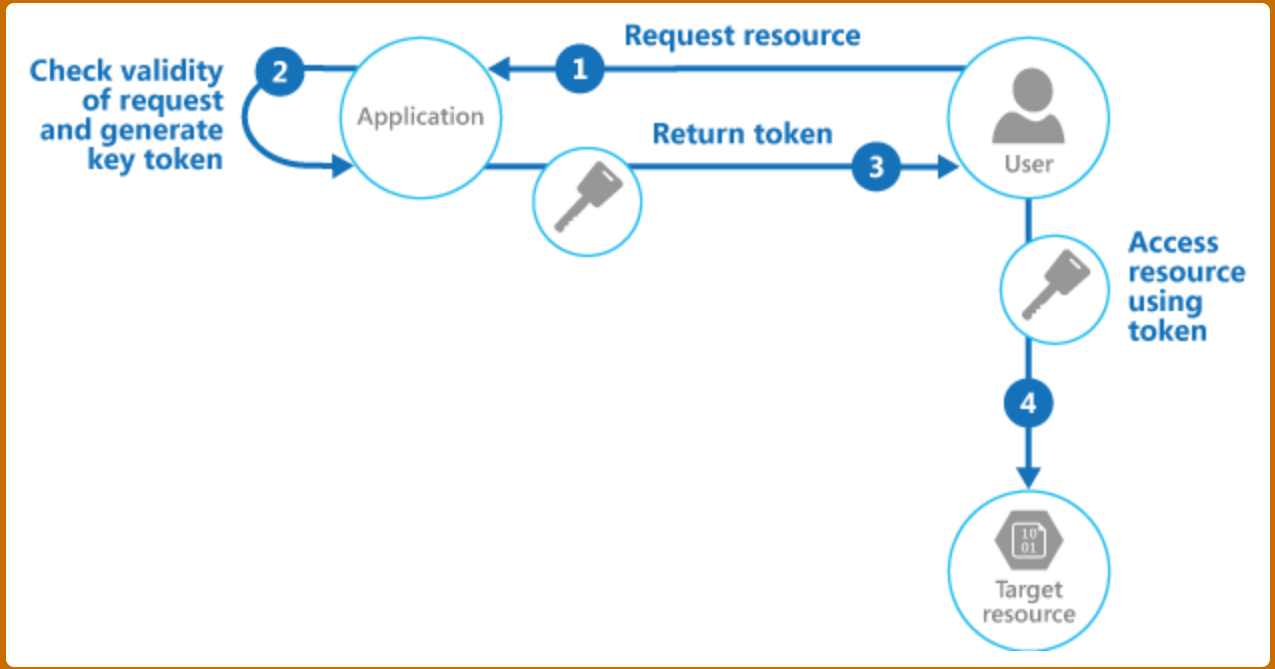


Static Content Hosting Pattern



CDN or Replication to multiple Storage Accounts

Valet Key Pattern



Demo

Valet Key Pattern

Best Practices Storage

Lean on Storage for static content hosting.

Use the Materialized Views Pattern.

Grant client access using the Valet Key Pattern.

Patterns & Practices: Cloud Design Patterns

bit.ly/PPCloudPatterns

Services are bolted together – not built!

Automate deployments, environments, scaling – life!

We are moving to a Microservices/PaaS world!