Azure Cost Optimization
Modernize & Save

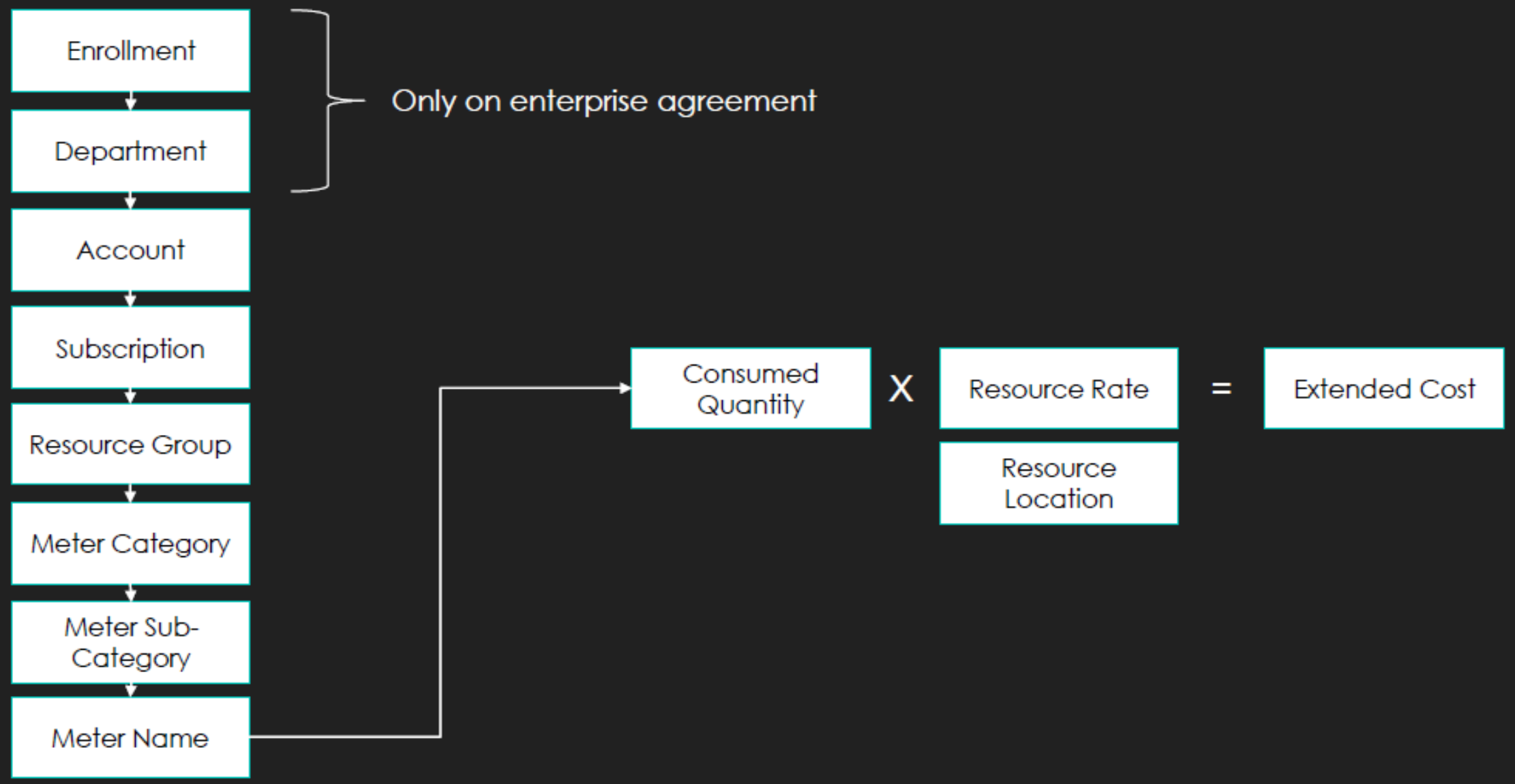
Matouš Rokos
Chief Cloud Solutions Architect
Atea

Microsoft

Agenda

- Organize
- How far am I?
- Size matters
- Technicalities

Tips: Tags can be attached



Tips: Tags can be attached

Enrollment	847455703
Department	Group IT
Account	TEAM Nettside
Subscription	Microsoft Azure Enterprise
Resource Group	default-sql-northeurope
Meter Category	Storage
Meter Sub-Category	Locally Redundant
Meter Name	Standard IO - Block Blob (GB)

Consumed Quantity: 122

X

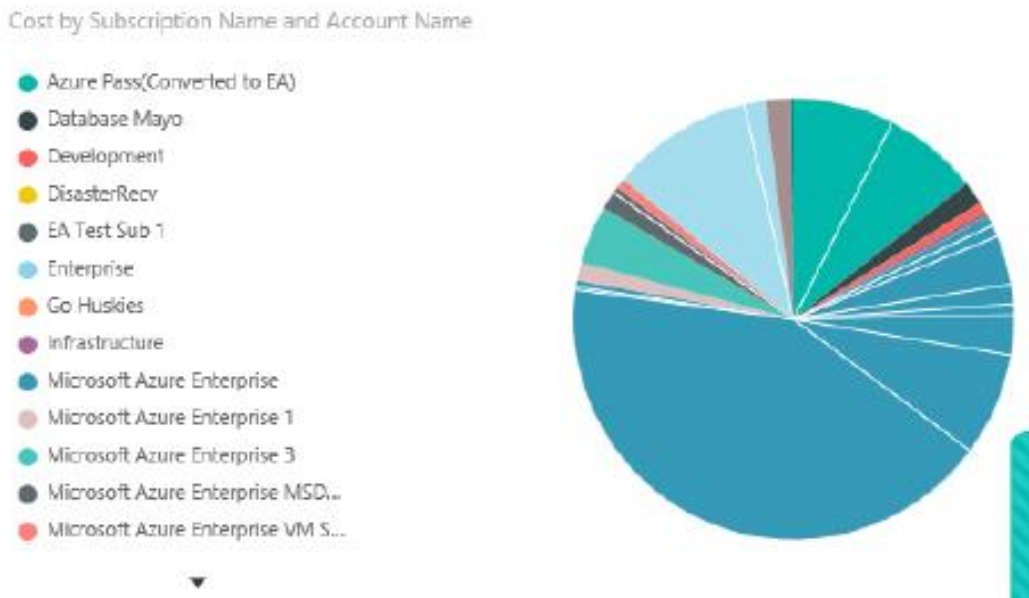
Resource Rate: 0.083065008185679

Resource Location: eunorth

=

Extended Cost: **NOK 10,133931**

Accounts & Subscriptions



Cost exploration: Azure EA PowerBI experience

Usage Trend By Services

Meter category

- App Services
- Business Analytics
- Cloud Services
- Data Management
- Data Services
- Mobile Services
- Networking
- Recovery Services
- Service Bus
- Storage
- Virtual Machines
- Virtual Network
- Visual Studio
- Websites
- Windows Azure - All Services
- Windows Azure Storage

Cost by Date and Meter Sub-Category

Meter Sub-Category

- A1 Cloud Services
- A1 VM (Windows)
- A2 Cloud Services
- A2 VM (Windows)
- A3 Cloud Services
- A5 VM (Windows)
- A6 VM (Windows)
- All
- Application Insights
- Backup
- BASIC.A0 VM (Windows)
- BASIC.A1 VM (Windows)
- BASIC.A2 VM (Windows)
- DS HDInsight
- Data Management

400 350 300 250 200 150 100 50 0

Nov 2015 Dec 2015 Jan 2016 Feb 2016 Mar 2016 Apr 2016

A0 Cloud Services -... Product	US South Central Meter Region	0.98 Consumed Quantity	Hours Unit Of Measure
A1 Cloud Services -... Product	AP East Meter Region	1.97 Consumed Quantity	Hours Unit Of Measure
A1 Cloud Services -... Product	EU West Meter Region	7.87 Consumed Quantity	Hours Unit Of Measure
A1 Cloud Services -... Product	US East Meter Region	0.98 Consumed Quantity	Hours Unit Of Measure
A1 Cloud Services -... Product	US South Central Meter Region	1.00 Consumed Quantity	Hours Unit Of Measure
A1 Cloud Services -... Product	US West Meter Region	10.75 Consumed Quantity	Hours Unit Of Measure

Month

- 2015-11
- 2015-12
- 2016-1
- 2016-2
- 2016-3
- 2016-4

Cost by Date and Product

35 30 25 20 15 10 5 0

Nov 2015 Dec 2015 Jan 2016 Feb 2016 Mar 2016 Apr 2016

Usage By Accounts&Subscriptions | Usage By Services

Management Dashboard

Default

Cost Entity Summary

7 Cost Entities
42 Accounts

Cost Over Time
Cost for last day

$26,063

Cost trend for last 30 days

Asset Controller
Number of Running Instances

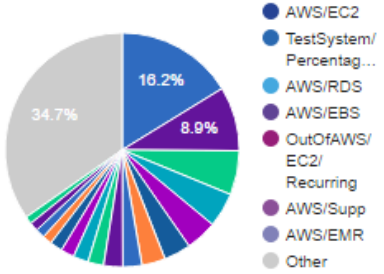
147 instances

Usage trend for last 30 days

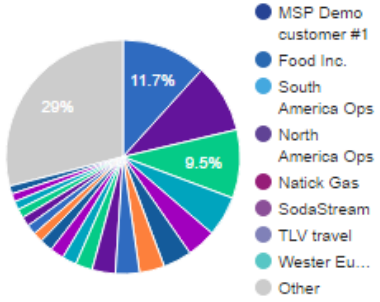
Unused RI Detector
Number of unused Reserved Instances

$1,241,959
631 RI Recommendations

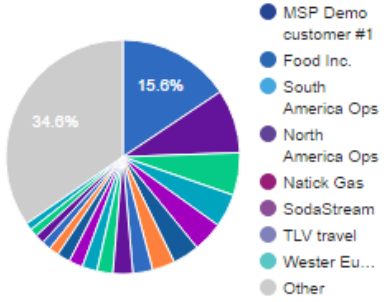
Cost – MSP View
Annual Projection



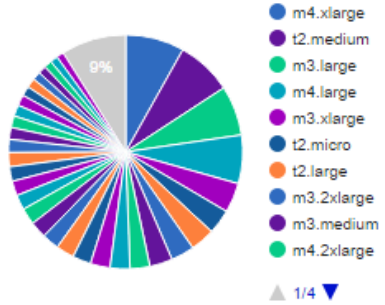
Profit Margin by Customer
Previous Month

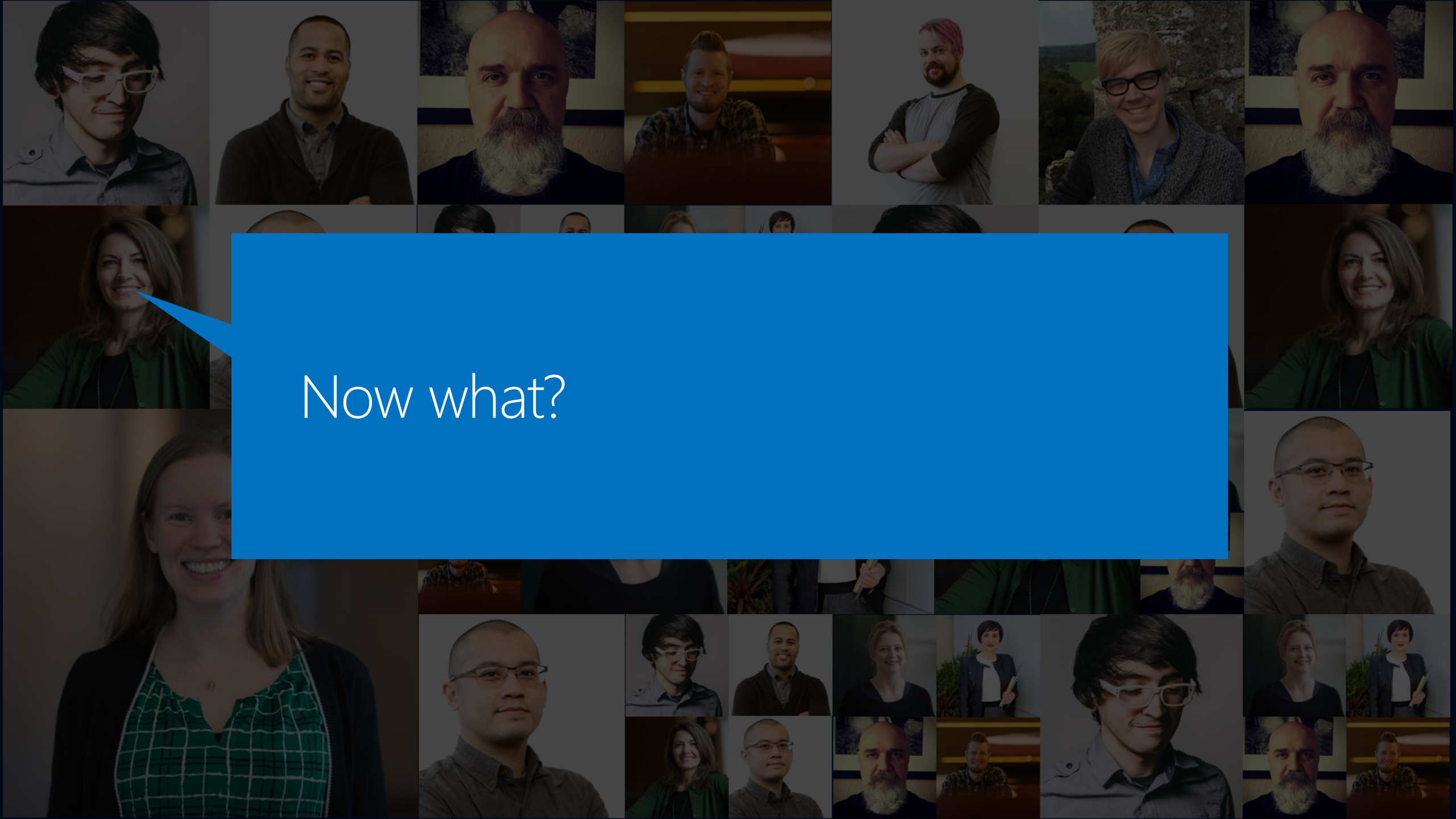


Cost – Customer View
Annual Projection



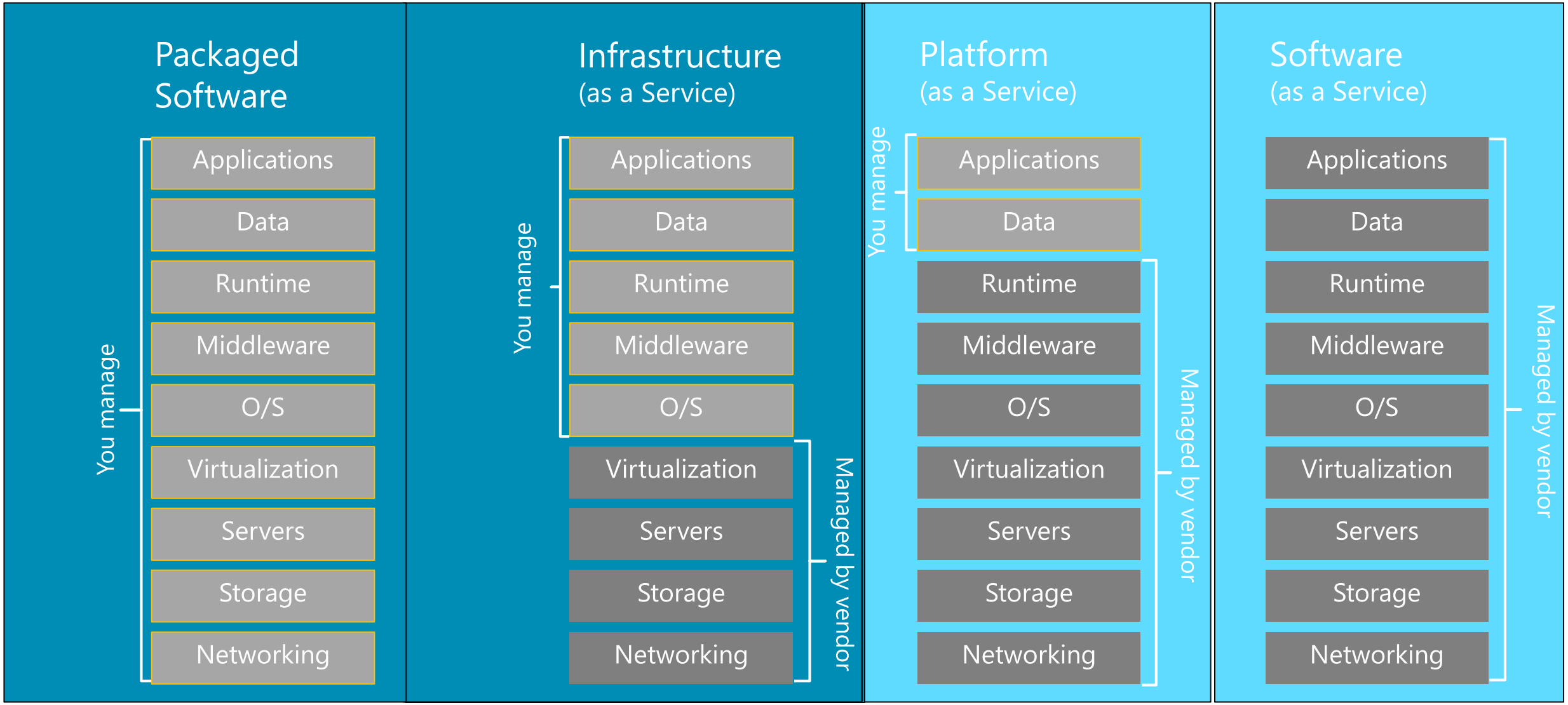
Potential Savings
Instance Type Pricing Recommendations





Now what?

Where do you see yourself?




VOLGA-DNEPR
Putzmeister
Putzmeister
© AP


RENA
MONROVIA

Breaking up the monolith

Into Microservices


Lea & Devin
Leafcutter Studio
Berkeley, CA
VIA AIR MAIL
SPECIAL DELIVERY
Wonderful You
Wherever You Are
On the Road
Or Cozy at Home

The Trouble With Monoliths

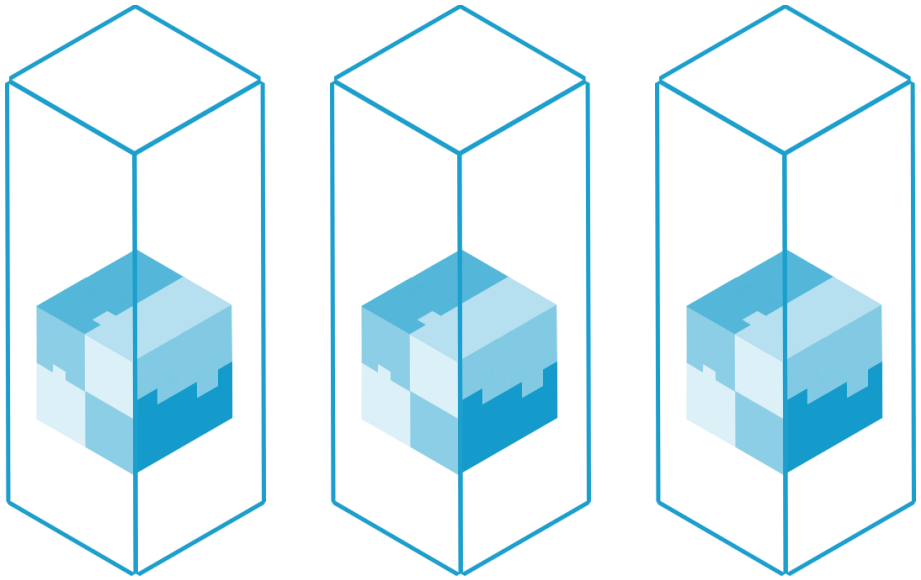
Tightly coupled components

All components updated together

Not agile, time to market suffers

Scale by cloning entire apps

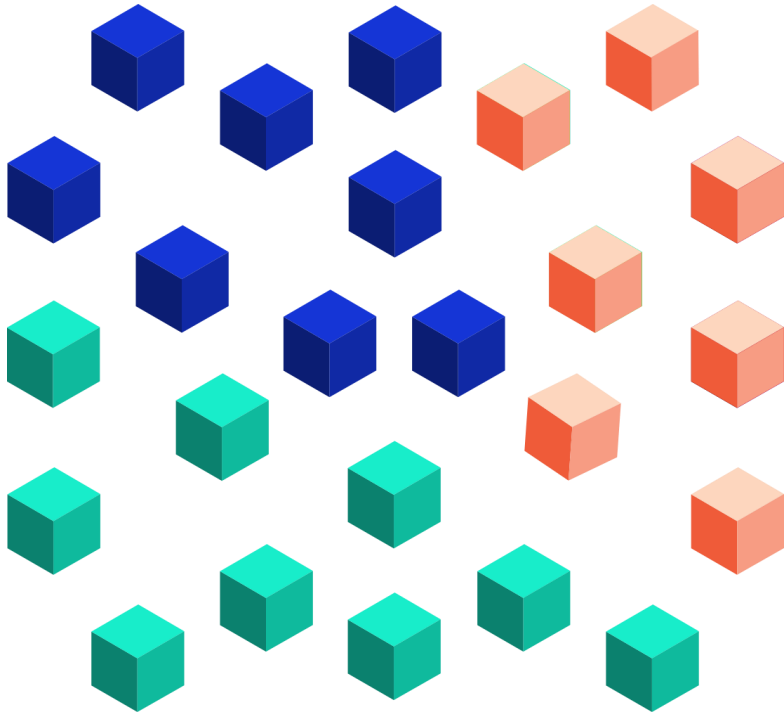
All components scaled similarly → expensive



Microservices

Do one thing well

- Manage independent code and state
- Are generally developed by a small cross-functional team
- Are built with task-appropriate languages/frameworks



Are loosely coupled

- Communicate over well-defined interfaces/protocols
- Have unique names (URI) that can be resolved
- Are independently updated
- Are independently scaled

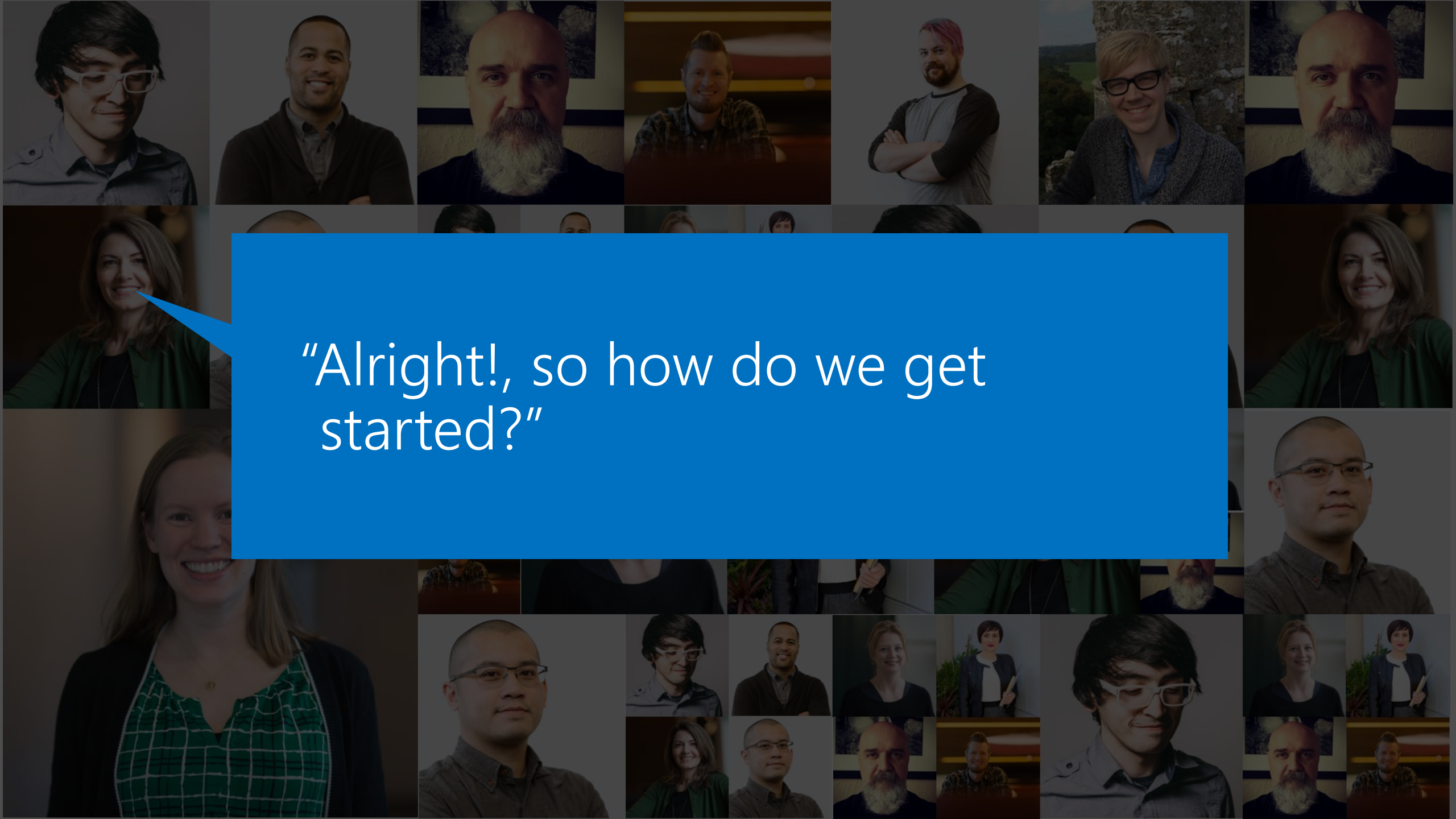
Why?

Higher density - reduce cost

Scale the things that needs scaling

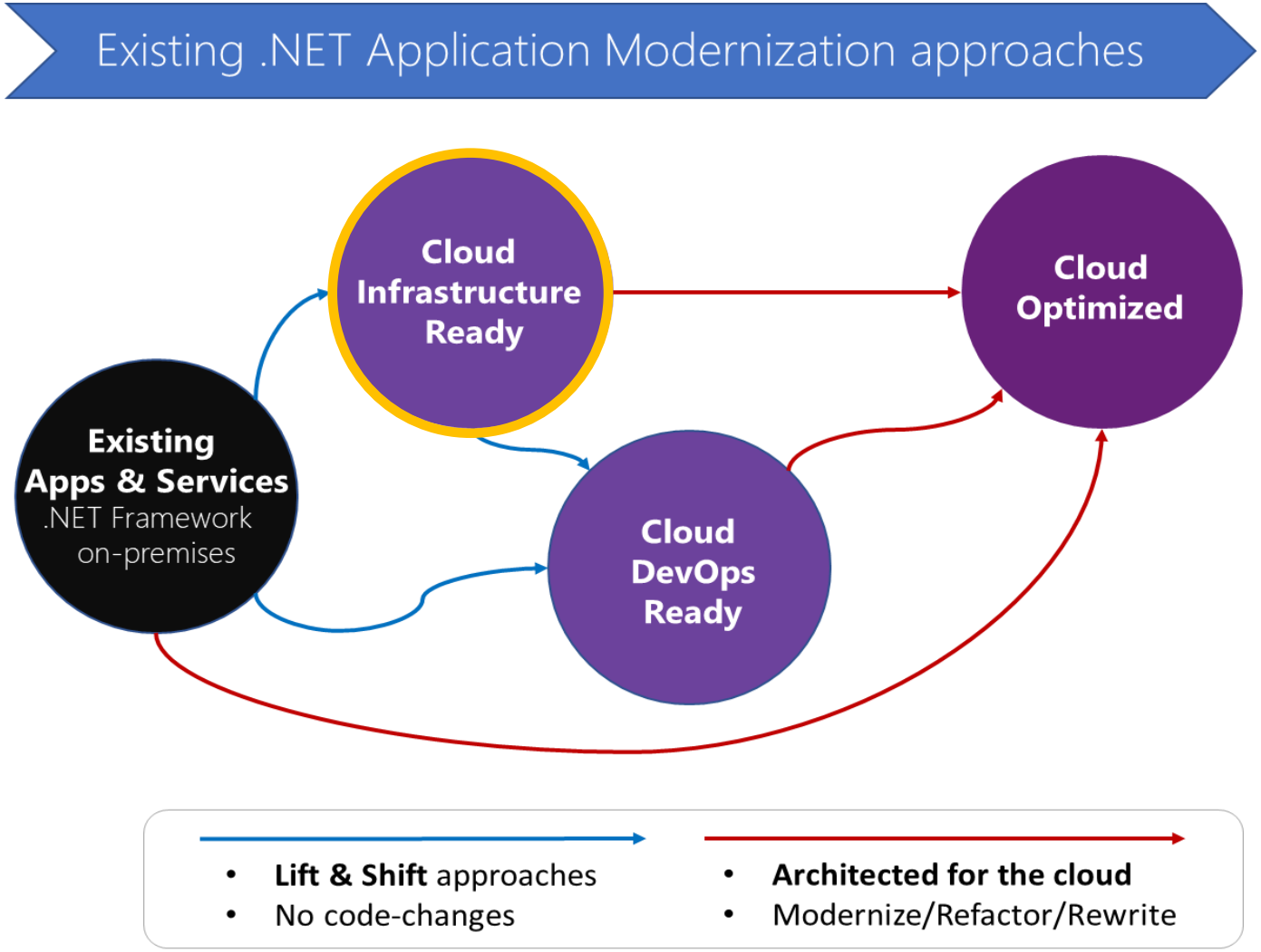
Deliver more and faster

Deploy features independent from each other



"Alright!, so how do we get started?"

Cloud Maturity Model



1. Cloud Infrastructure ready

Simply **Rehost** your **on-premise application to IaaS on Azure**

PROS

- ✓ No re-architect or new code
- ✓ Least effort for quick migration
- ✓ Supported on the least common denominator on Azure

Existing Apps
.NET Web apps (on-prem.)

On-Premises

Using:

Servers or VMs on-premises

SQL Server, Oracle, MySQL, etc.

Monolithic Architecture

Lift & Shift
.NET Web apps (Cloud)

IaaS Cloud

Using:

Virtual Machines (VM)

Azure SQL DB Managed Instance

Monolithic Architecture

1. Cloud Infrastructure ready

Simply **Rehost** your **on-premise application to IaaS on Azure**

PROS

- ✓ No re-architect or new code
- ✓ Least effort for quick migration
- ✓ Supported on the least common denominator on Azure

CONS

- × Smaller Cloud Value
- × Manual Patching, Upgrades
- × No Automated App Scaling and High Availability

Existing Apps
.NET Web apps (on-prem.)

On-Premises

Using:
Servers or VMs on-premises
SQL Server, Oracle, MySQL, etc.

Monolithic Architecture

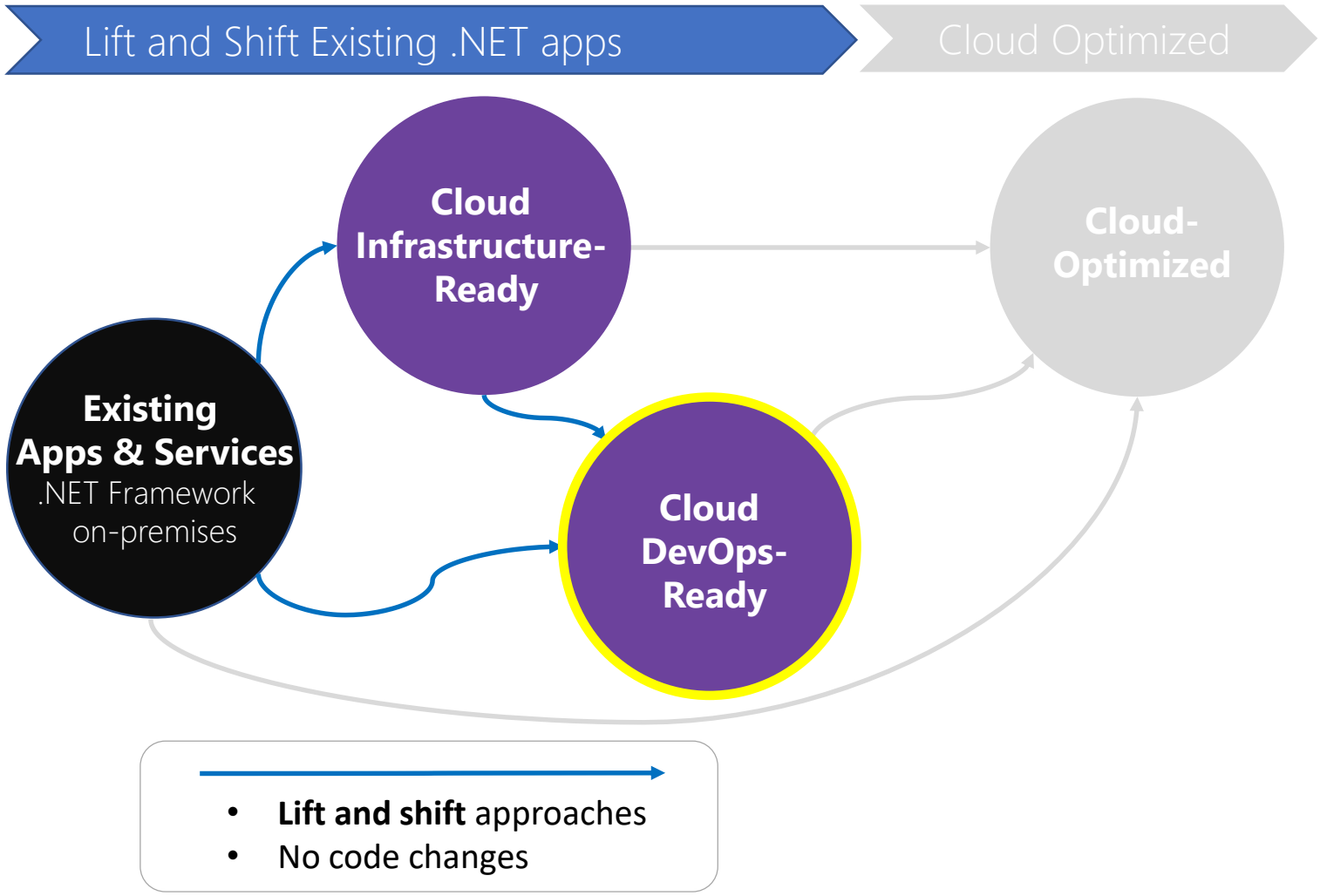
Lift & Shift
.NET Web apps (Cloud)

IaaS Cloud

Using:
Virtual Machines (VM)
SQL DB Managed Instance

Monolithic Architecture

Modernization Maturity Model

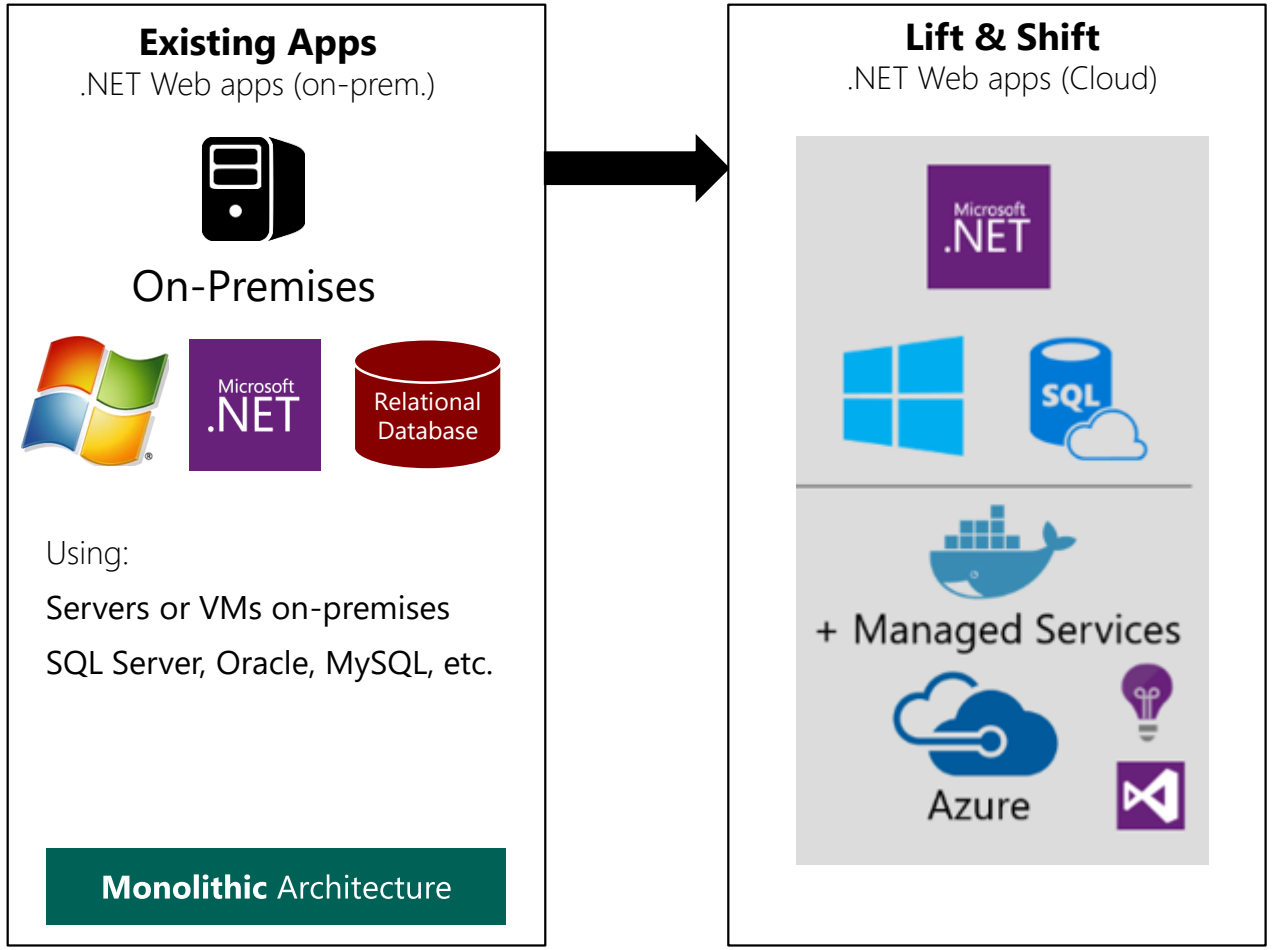


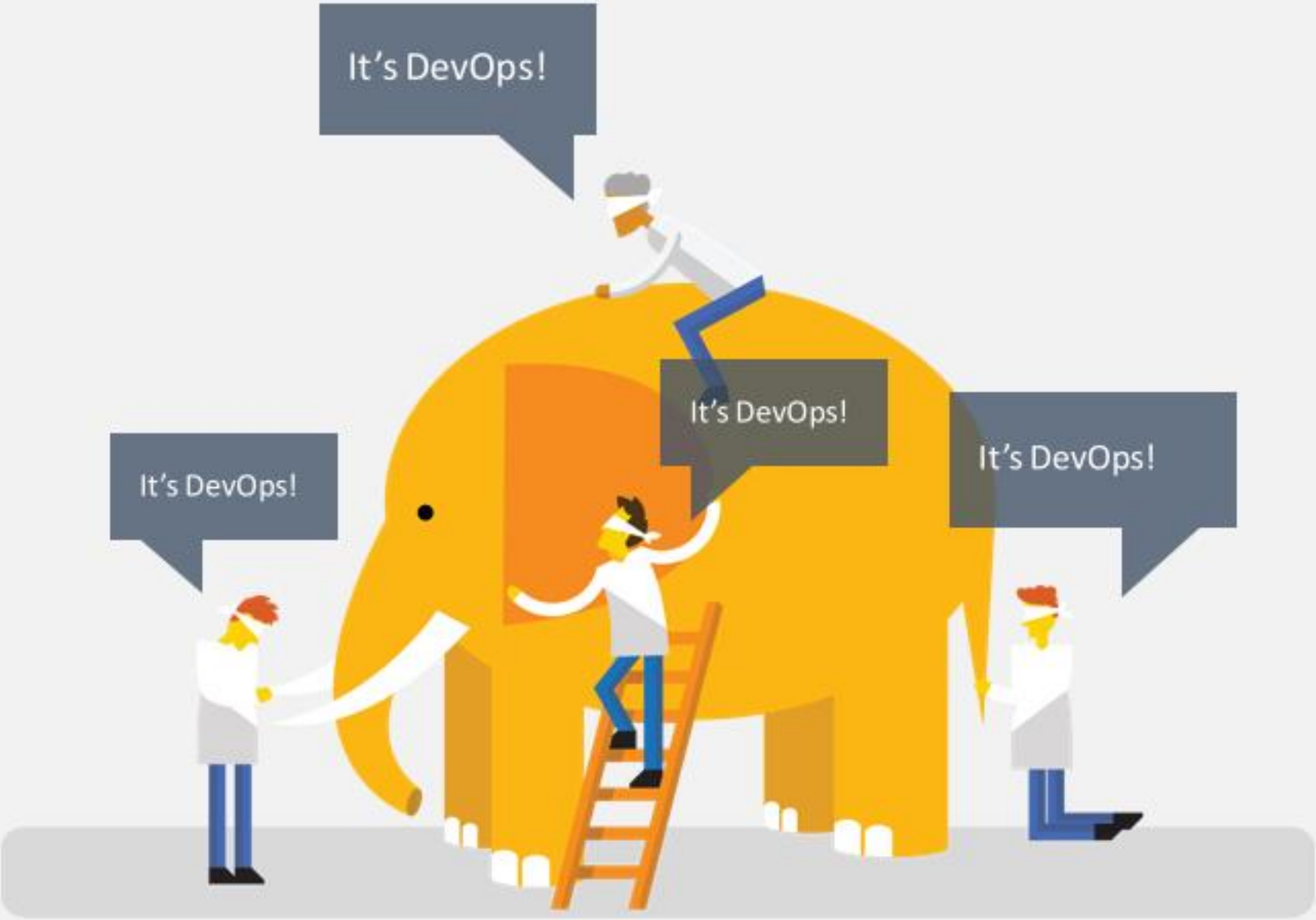
2. Cloud DevOps ready

Get more Cloud benefit by **Containerizing** your app with **Windows Server Docker Containers** and deploying them to Azure cloud or on-premises.

PROS

- ✓ No re-architect or new code
- ✓ Increased density & lower deployment cost
- ✓ Improved productivity and DevOps agility
- ✓ Portability of apps and dependencies
- ✓ High availability and Orchestration with ACS/K8 and Service Fabric




It's DevOps!
It's DevOps!
It's DevOps!
It's DevOps!


WORKED FINE IN DEV
OPS PROBLEM NOW
memegenerator.net

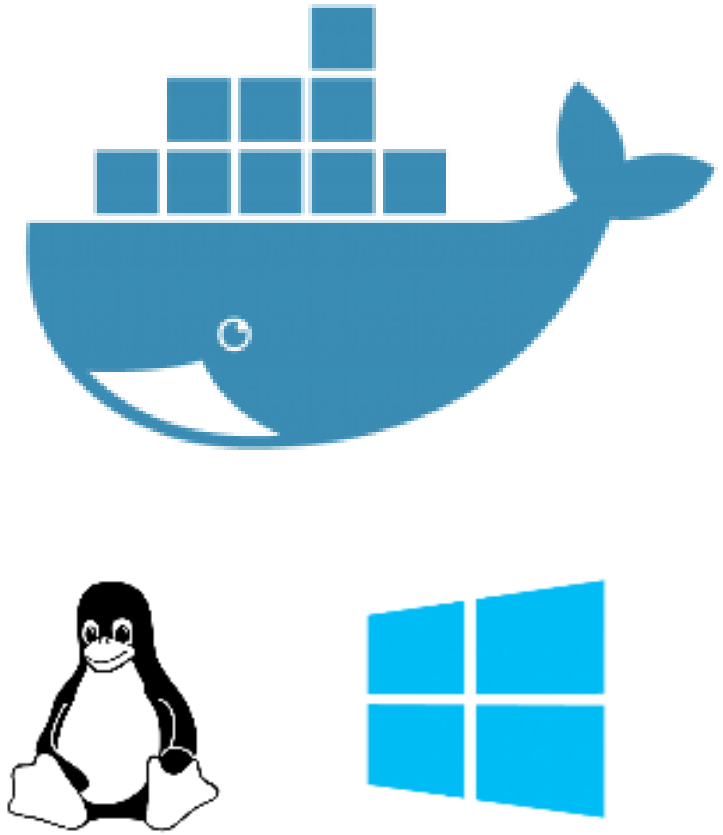
Docker Containers

- Docker helps automating the deployment of applications as portable, self-sufficient containers that can run on any cloud or on-premises.

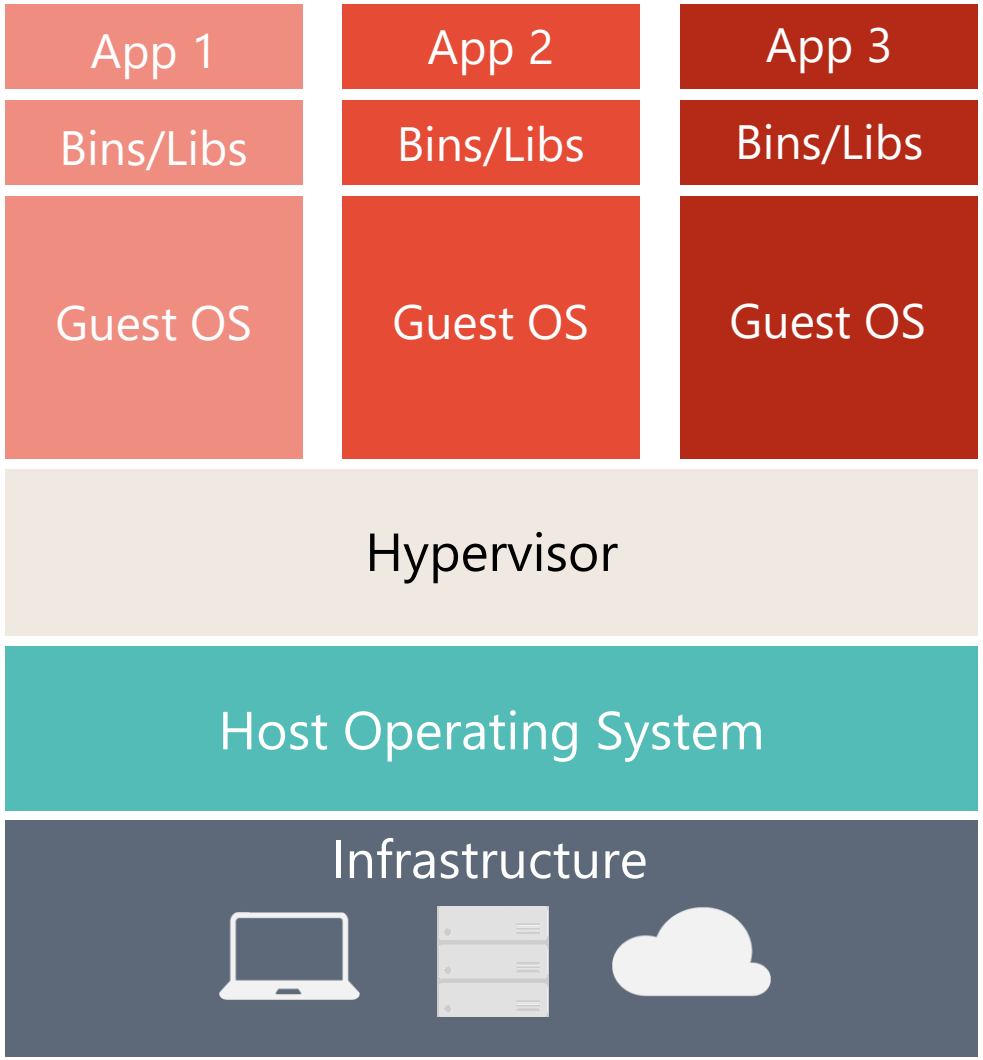
No more:
"It works in my dev machine!...
Why not in production?"

Now it is:
"If it works in Docker, it works in production"

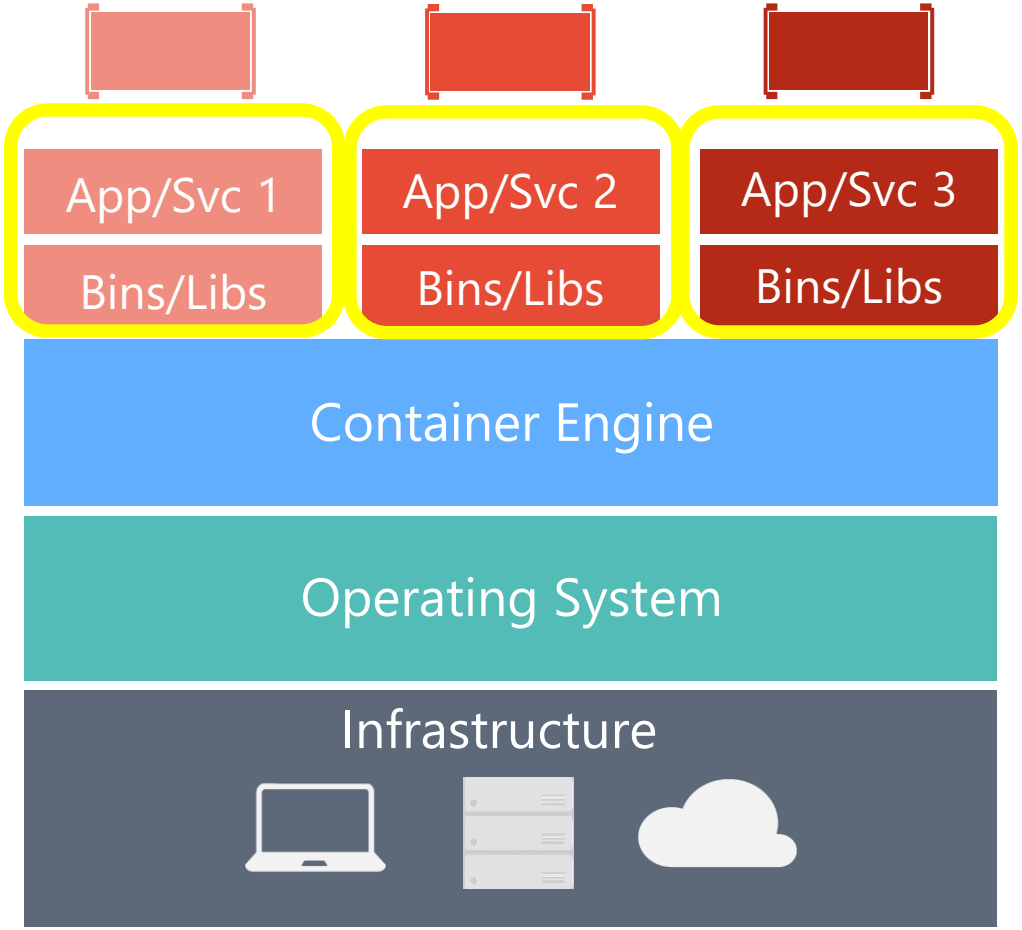
Keywords about WHY Docker?
- ***Dependencies (self-sufficient)***
- ***Deployment***



Virtual Machines compared to Containers



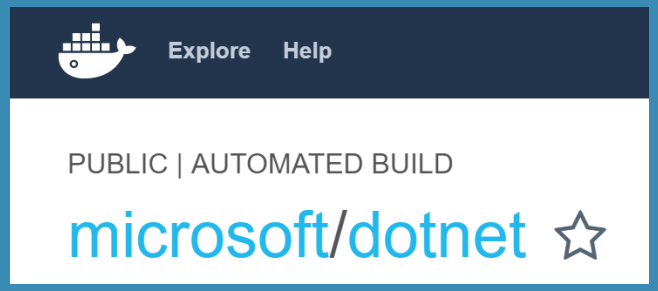
+ Network isolation at container level
+ Resources constraints at container level



Docker and .NET

- **.NET Framework** images
  Windows Server Core
- **.NET Core** Docker images
  xPlat. (Linux & Windows Nano Server)

See at Docker Hub



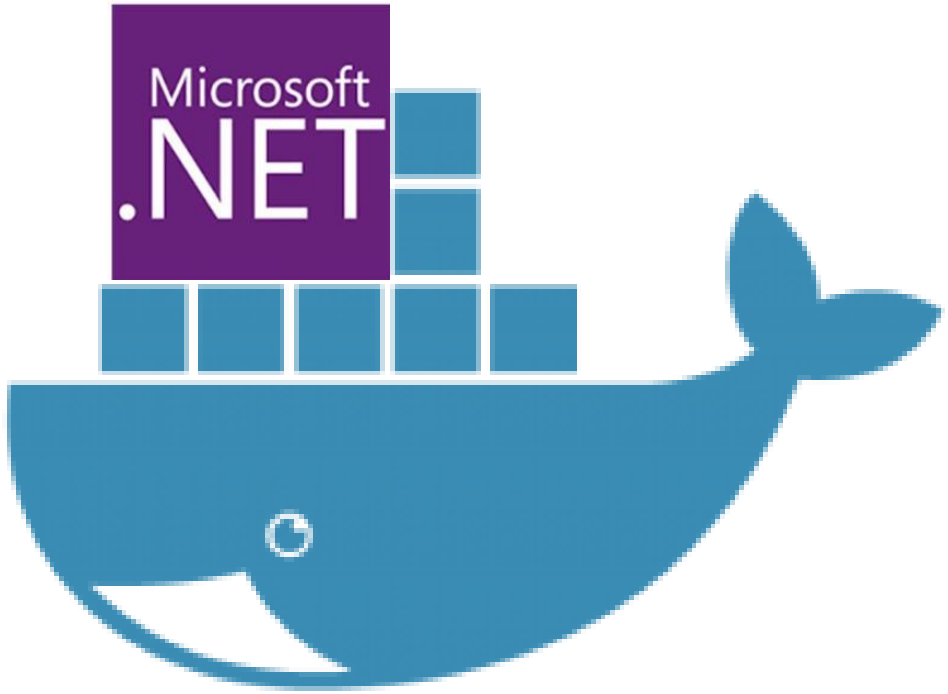
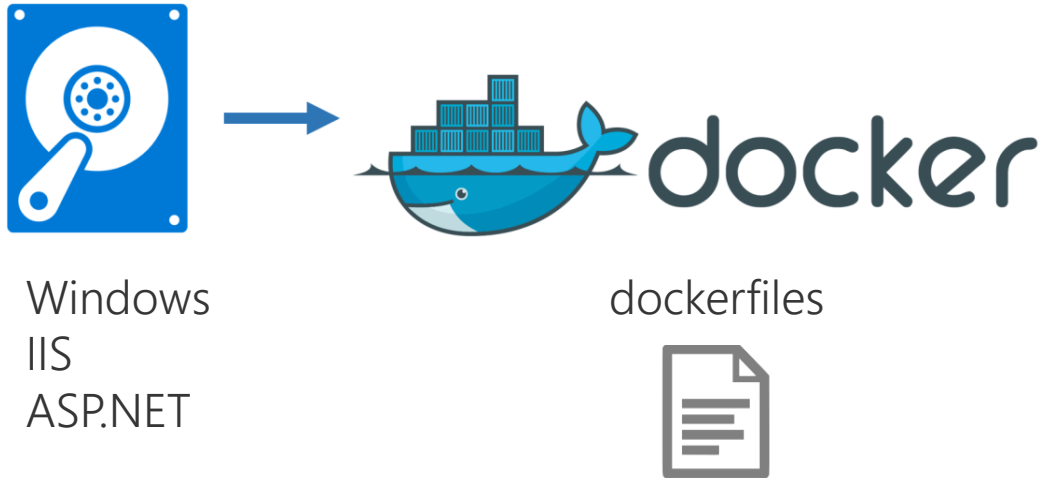


Image2Docker tool

- Ports existing Windows application workloads to Docker
- **IIS and ASP.NET apps**
  Extract ASP.NET websites config/dependencies from a VM or server
- Generates **dockerfiles** for Windows Docker images, based on analysis of existing Windows machines.
- Open Source community tool, powered by Docker (the company)



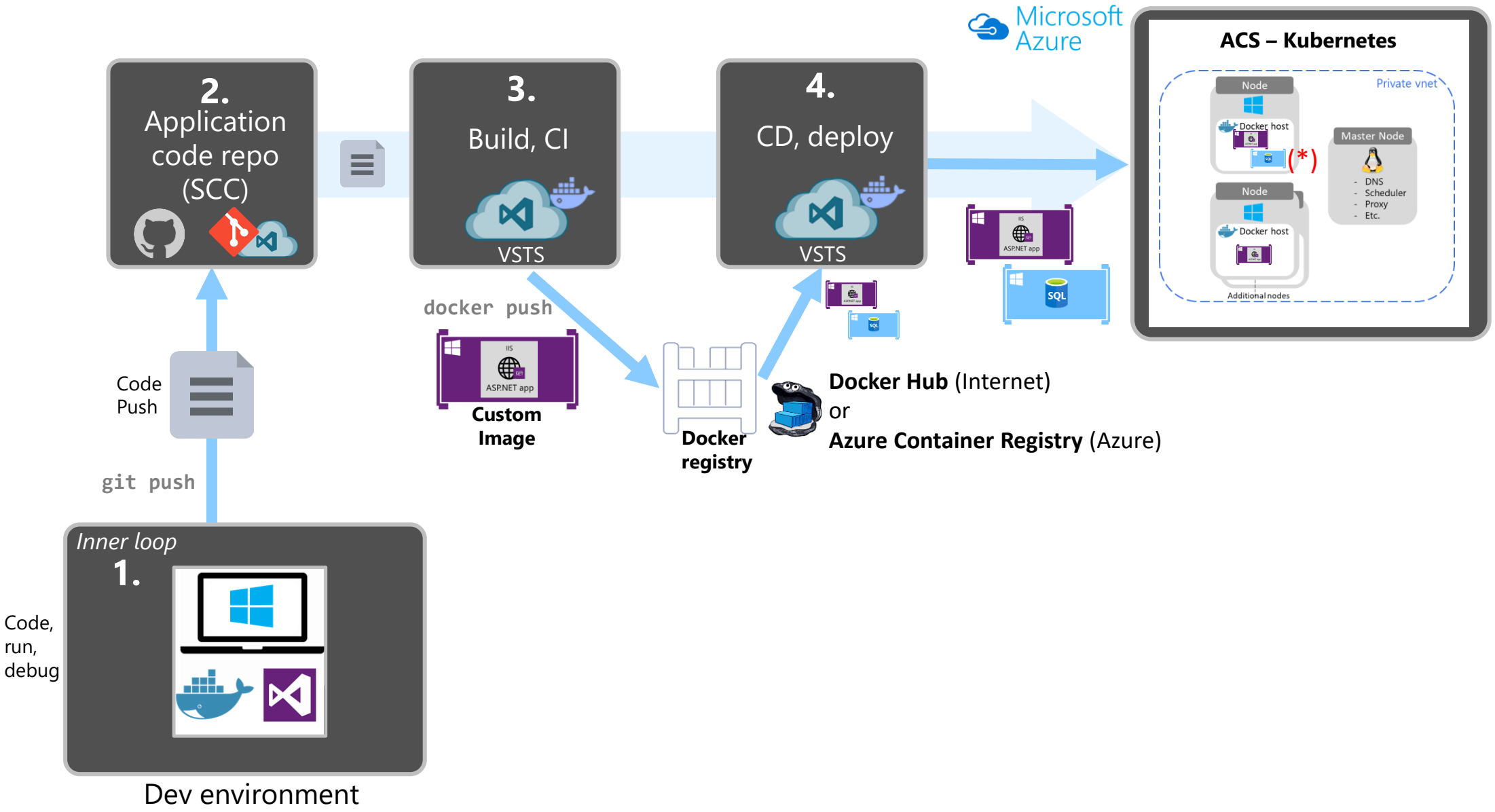
Choosing Orchestrators in Azure

Azure Product	Orchestrator	Description	Good for	Common workloads
Azure Container Service	**Kubernetes**	Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts	Production-ready & Windows/Linux ecosystem	Microservices based on containers
	Mesosphere DC/OS	As a datacenter operating system, DC/OS is itself a distributed system, a cluster manager and a container platform	Production-ready & Linux ecosystem	Microservices based on containers
	Docker Swarm	Docker Swarm is a clustering and scheduling tool for Docker containers. With Swarm, IT administrators and developers can establish and manage a cluster of Docker nodes as a single virtual system	Production-ready & Linux ecosystem	Microservices based on containers
Azure Service Fabric	**Service Fabric**	*Azure Service Fabric* is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices	Production-ready & Linux ecosystem	a) Stateful svc & Actors b) Microservices based on plain processes c) Microservices based on containers

Choosing Orchestrators in Azure

Azure Product	Orchestrator	Description	Good for	Common workloads
Azure Container Service	**Kubernetes**	Kubernetes is an open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts	Production-ready & Windows/Linux ecosystem	Microservices based on containers
	Mesosphere DC/OS	As a datacenter operating system, DC/OS is itself a distributed system, a cluster manager and a container platform	Production-ready & Linux ecosystem	Microservices based on containers
	Docker Swarm	Docker Swarm is a clustering and scheduling tool for Docker containers. With Swarm, IT administrators and developers can establish and manage a cluster of Docker nodes as a single virtual system	Production-ready & Linux ecosystem	Microservices based on containers
Azure Service Fabric	**Service Fabric**	*Azure Service Fabric* is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices	Production-ready & Linux ecosystem	a) Stateful svc & Actors b) Microservices based on plain processes c) Microservices based on containers

Scenario: Deploy to Kubernetes through CI/CD pipelines



2. CONS in Cloud DevOps ready

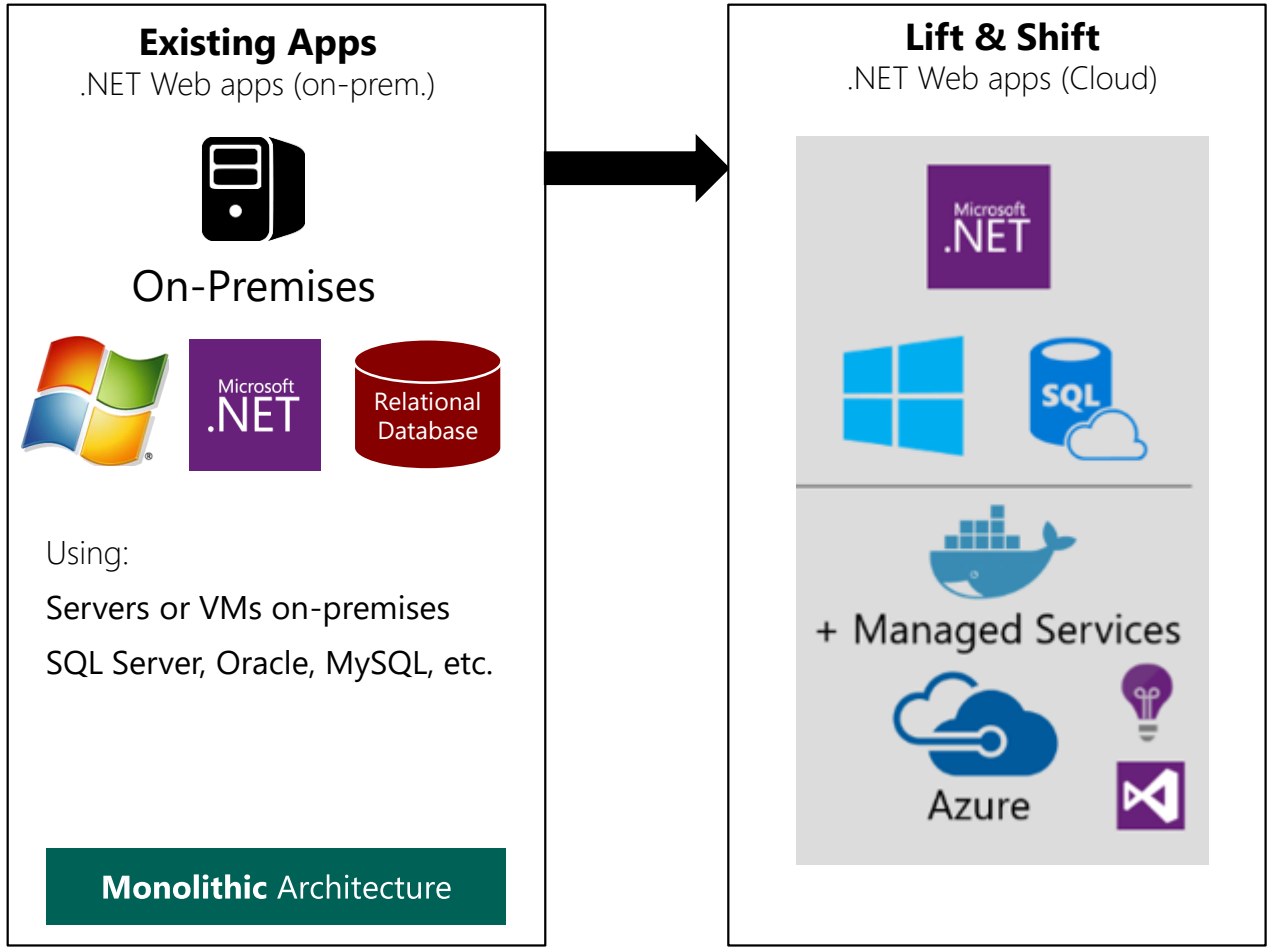
Get more Cloud benefit by Containerizing your app with Windows Server Docker Containers and deploying them to Azure using production orchestration

PROS

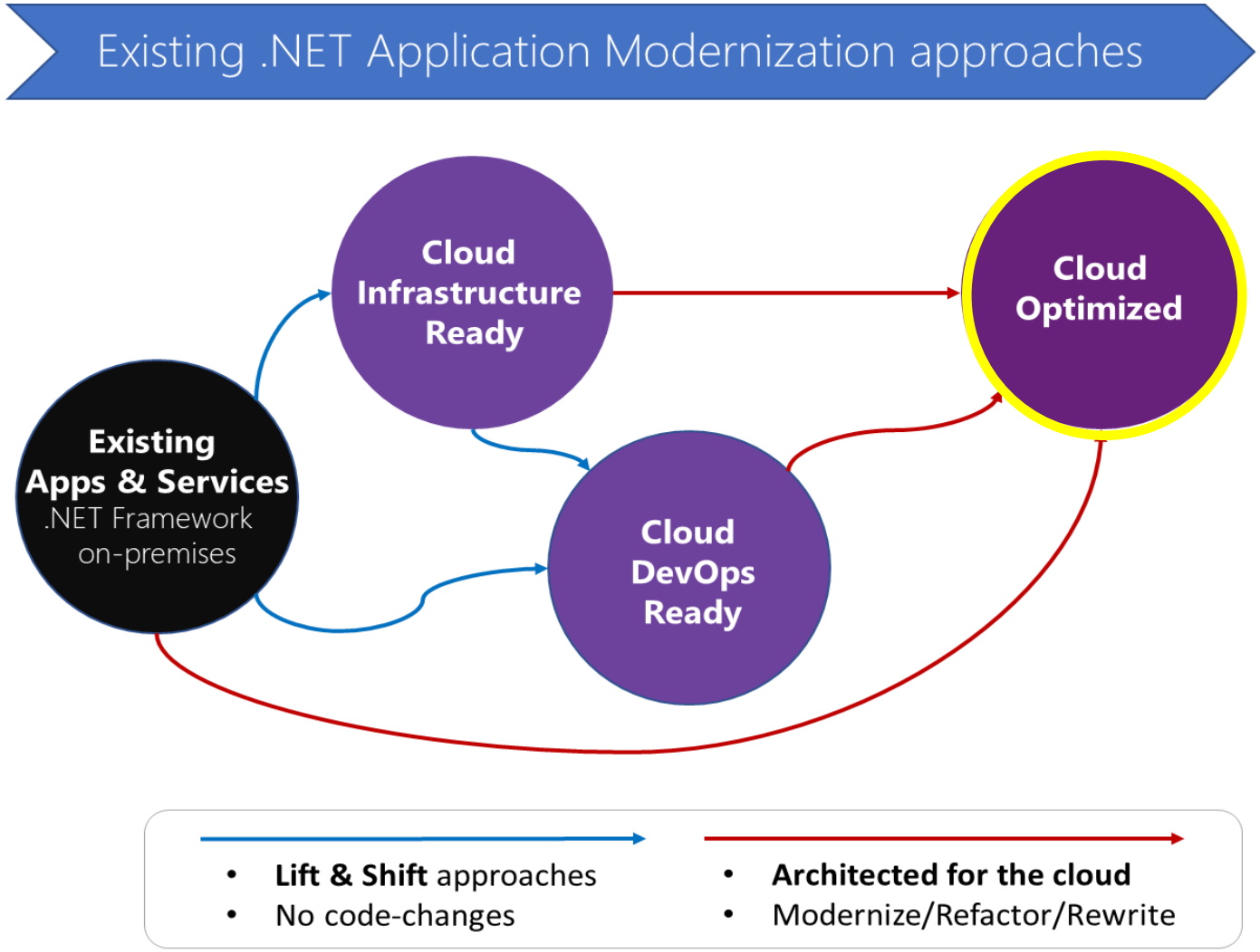
- ✓ No re-architect or new code
- ✓ Increased density & lower deployment cost
- ✓ Improved productivity and DevOps agility
- ✓ Portability of apps and dependencies
- ✓ High availability and Orchestration with ACS/K8 and Service Fabric

CONS

- × Containerization is an additional step in the learning curve



Cloud Maturity Model

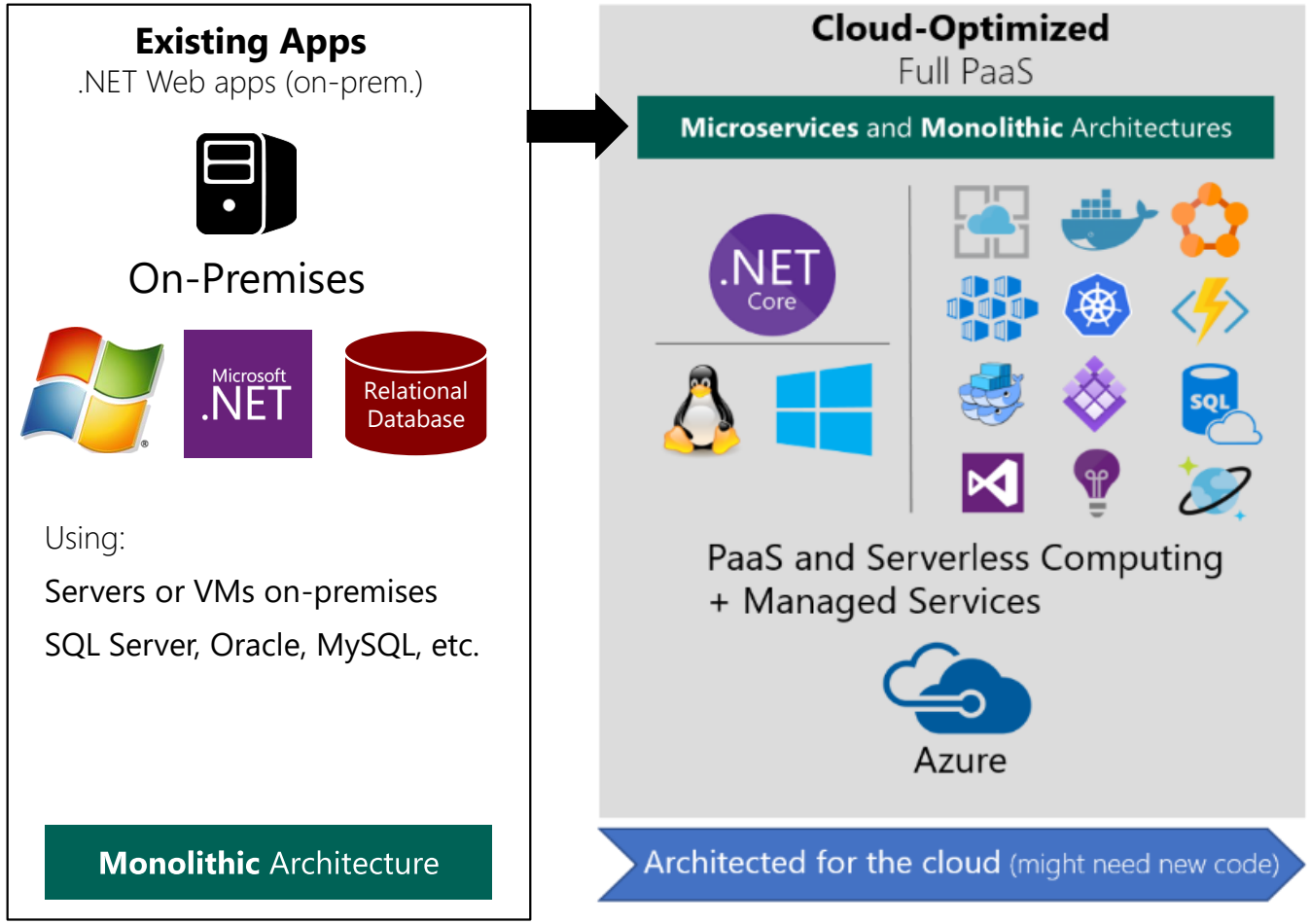


3. Going to Cloud-Optimized (Full PaaS)

Extend your apps with new services based upon Server less computing, Microservices architecture and PaaS services (AppService) to fully exploit the advantages of the cloud.

PROS

- ✓ Optimized for long term agility



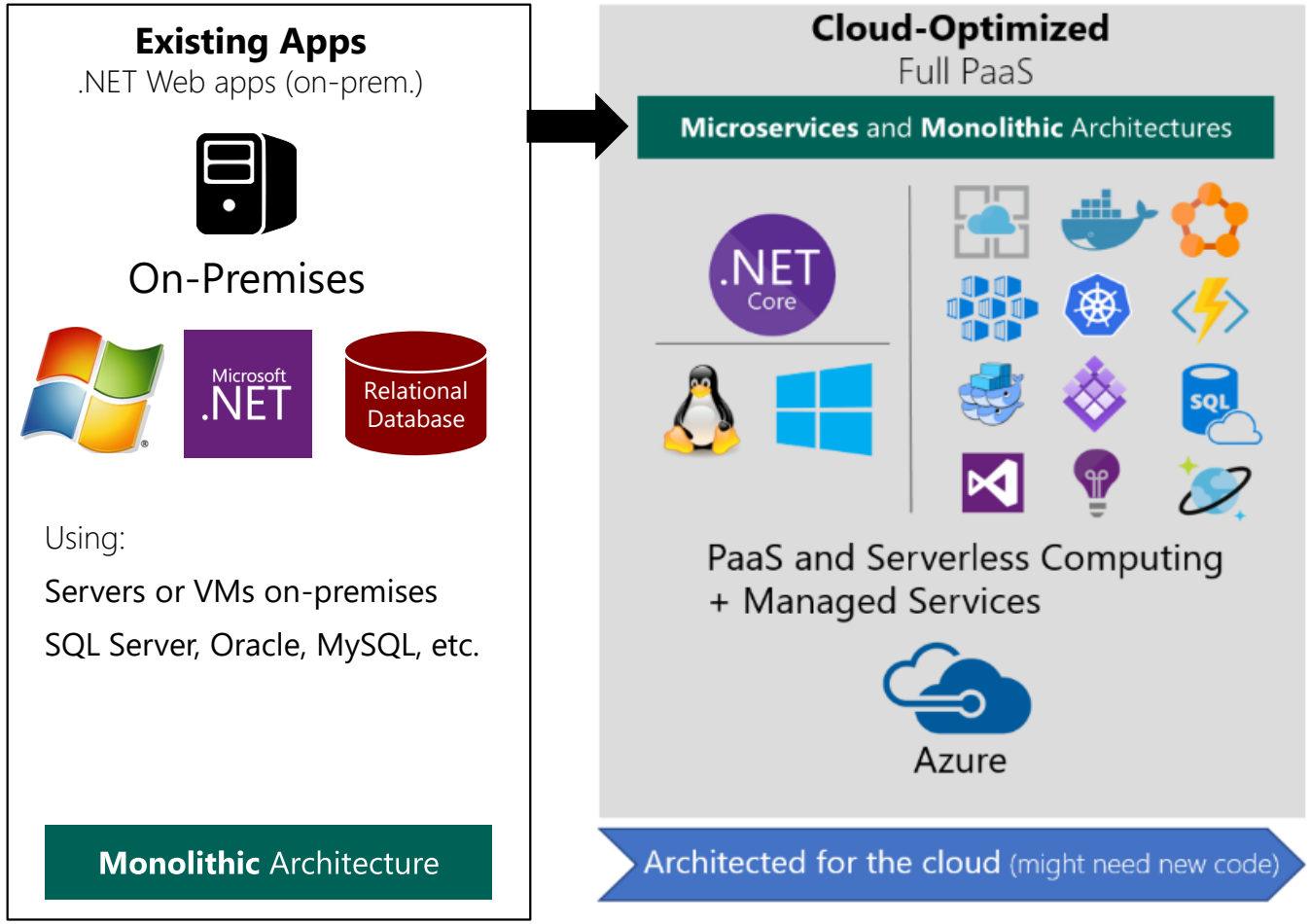
Wrap-up

3. Going to Cloud-Optimized (Full PaaS)

Extend your apps with new services based upon Server less computing, Microservices architecture and PaaS services (AppService) to fully exploit the advantages of the cloud.

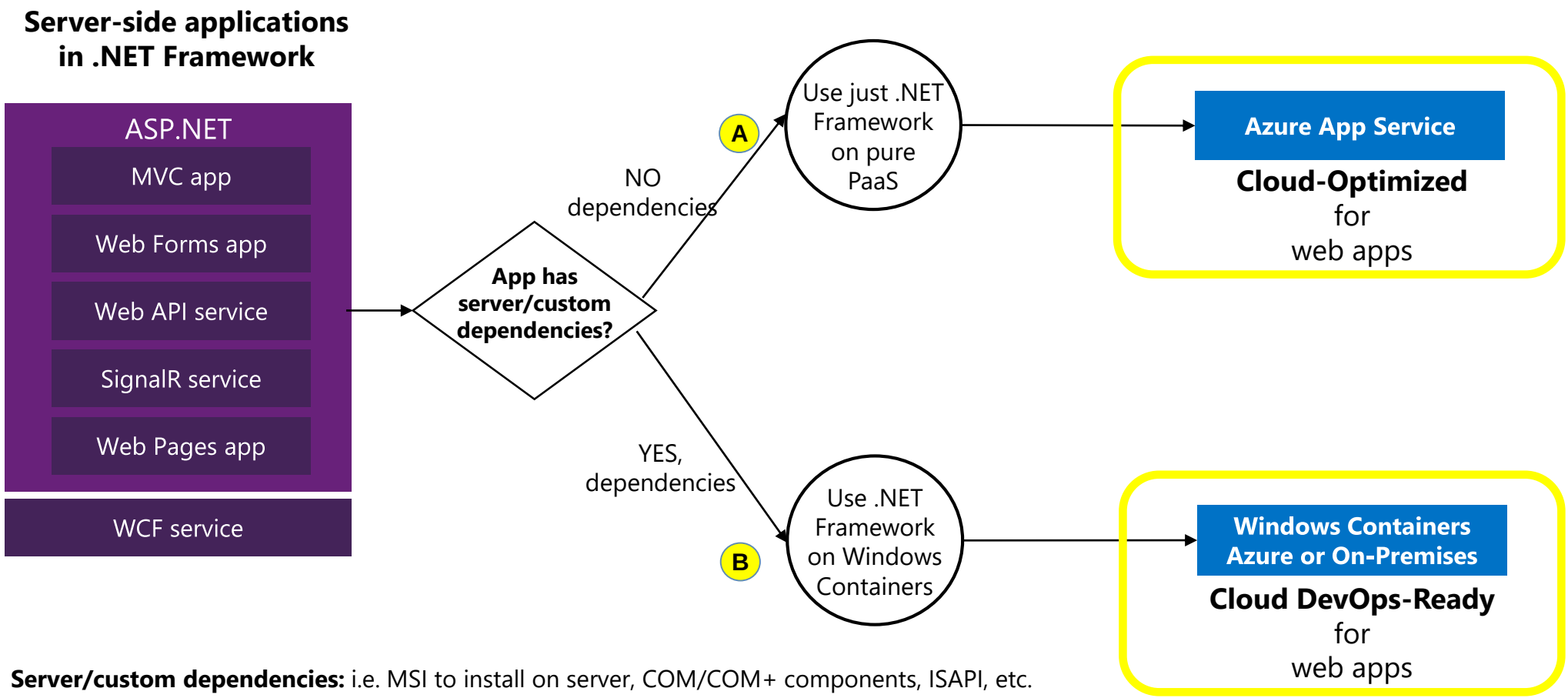
PROS

- ✓ Optimized for long term agility
- ✓ Optimized for scale and high availability
- ✓ Modern Architecture with Microservices and Cloud Native technologies



Wrap-up

When to use Azure App Service? (PaaS for Web Apps)



Server/custom dependencies: i.e. MSI to install on server, COM/COM+ components, ISAPI, etc.

3. Going to Cloud-Optimized (Full PaaS)

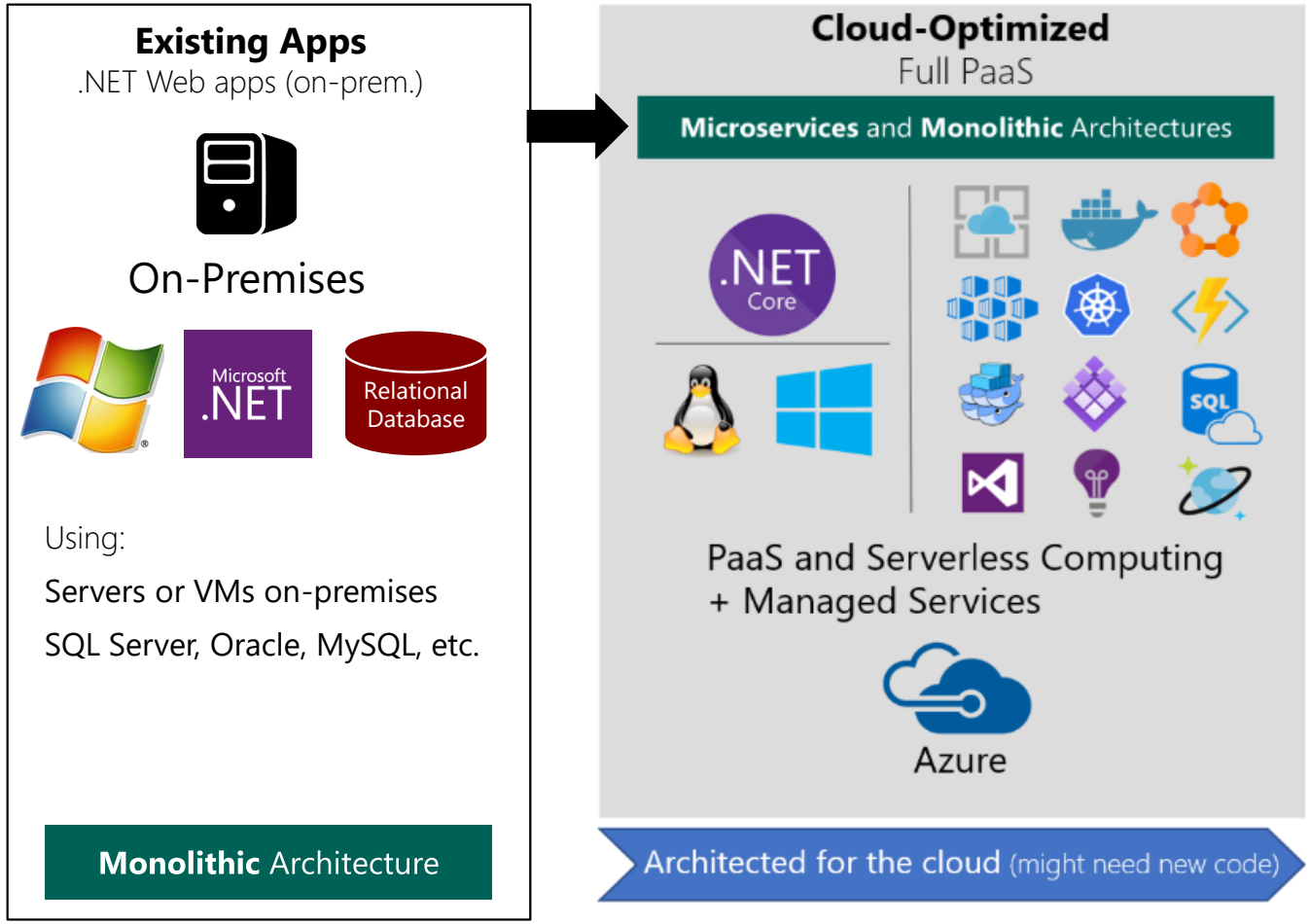
Extend your apps with new services based upon Server less computing, Microservices architecture and PaaS services (AppService) to fully exploit the advantages of the cloud.

PROS

- ✓ Optimized for long term agility
- ✓ Optimized for scale and high availability
- ✓ Modern Architecture with Microservices and Cloud Native technologies

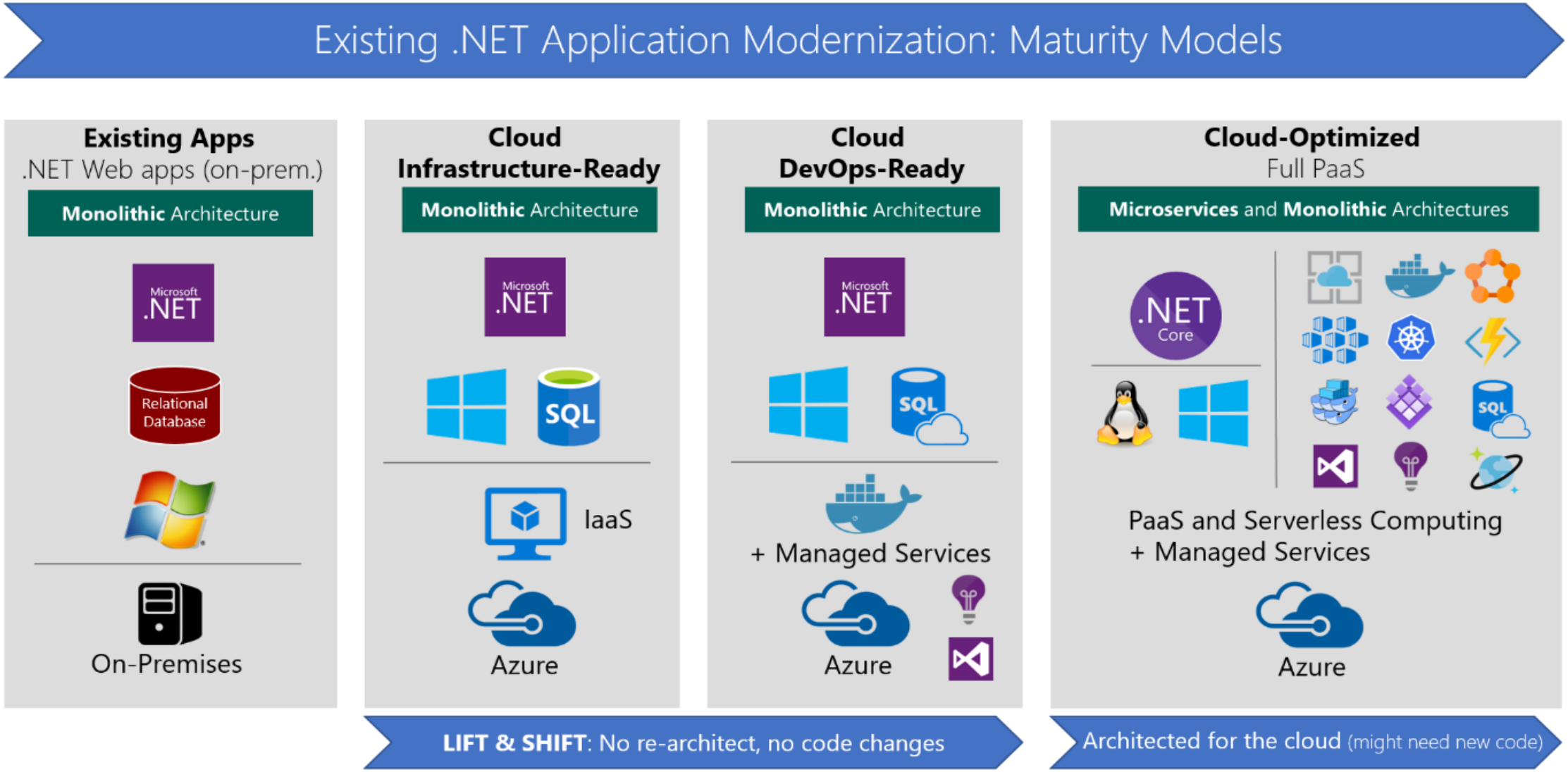
CONS

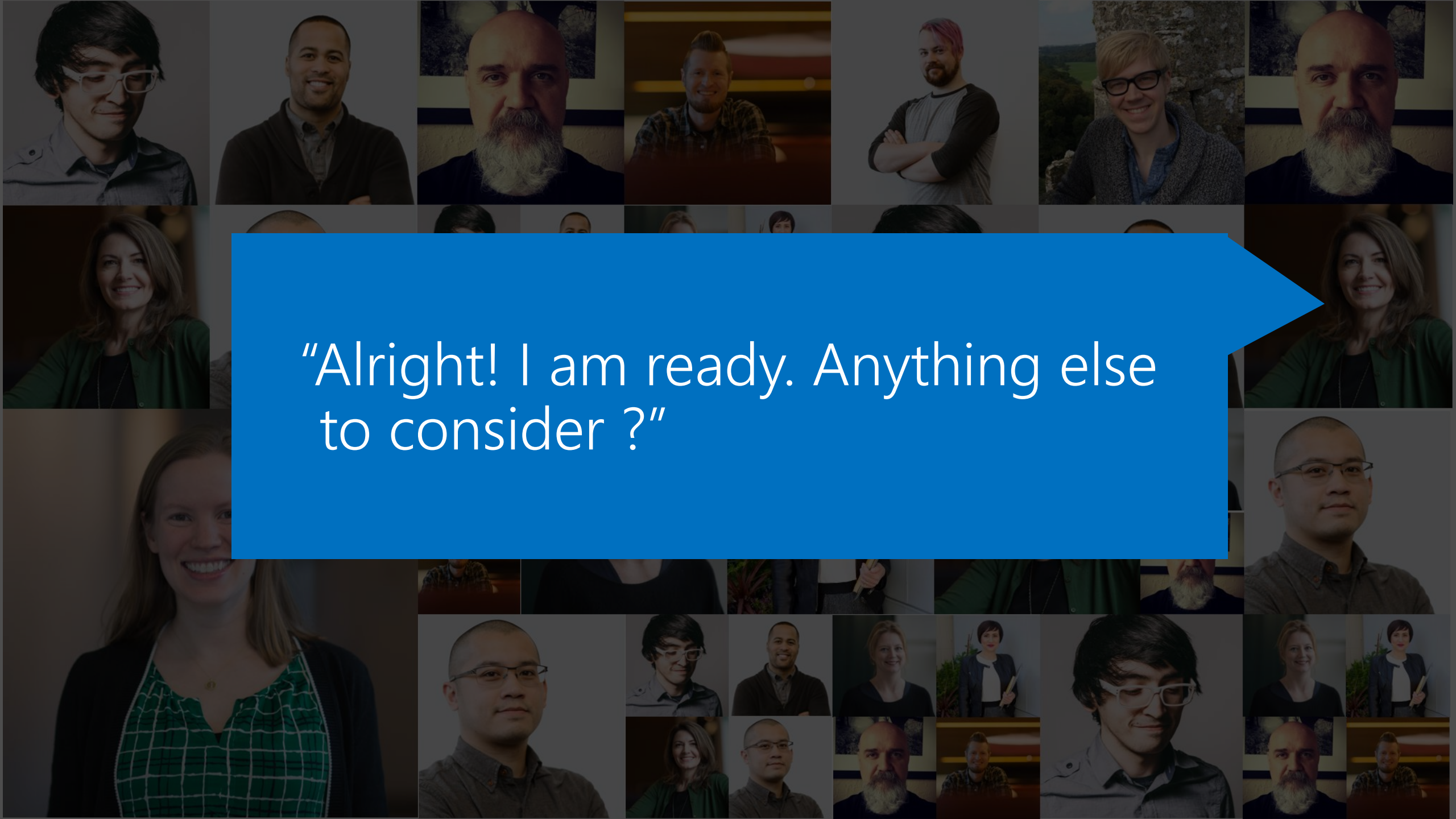
- × Requires significant code refactoring or rewriting (increased time and budget)



Wrap-up

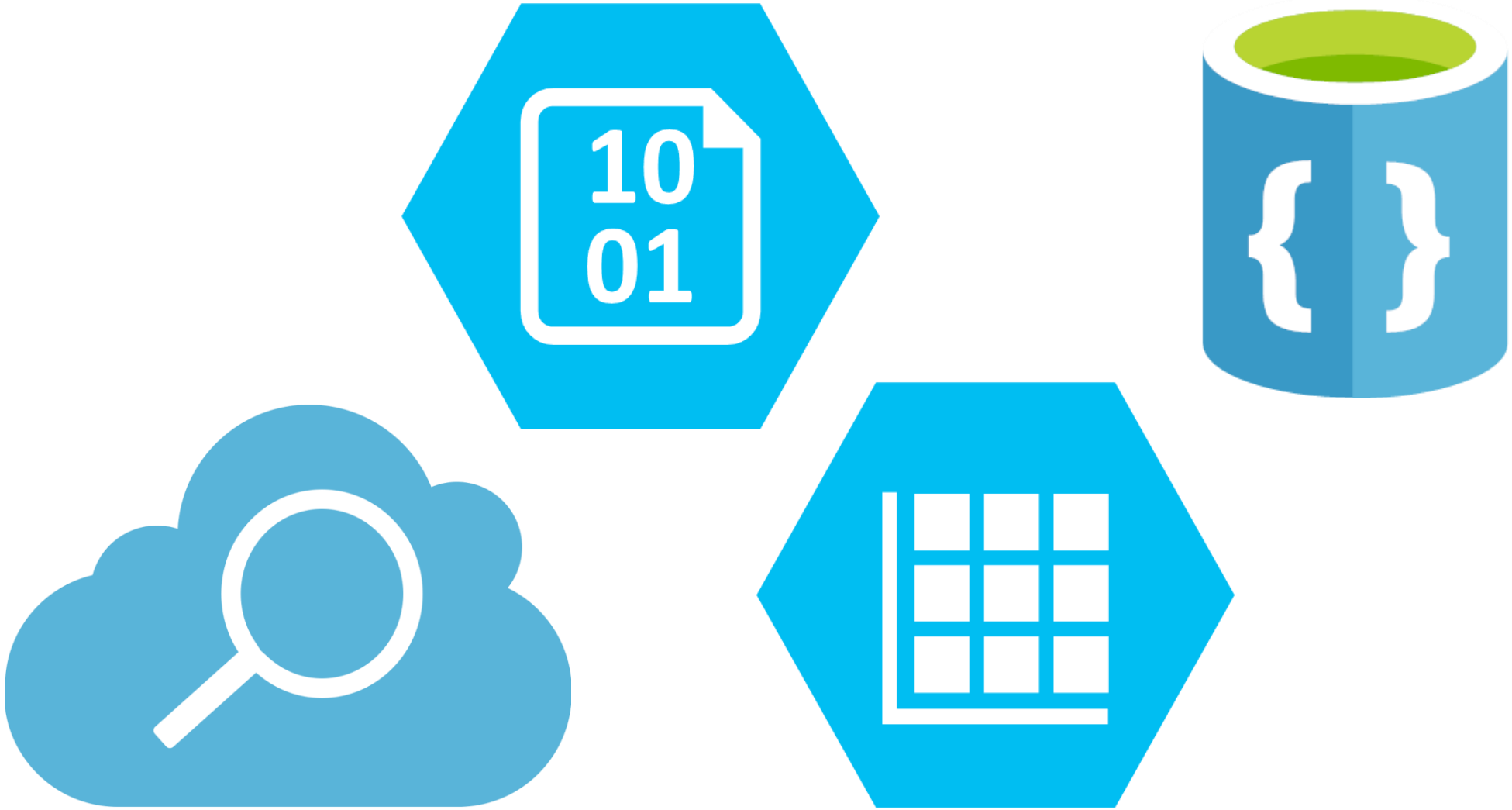
Modernization Maturity Model



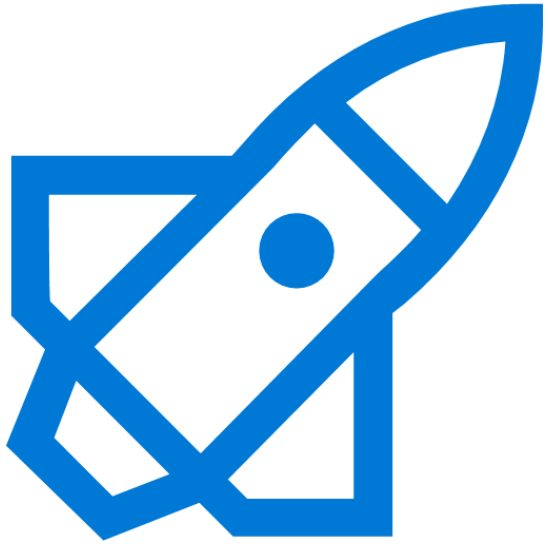


“Alright! I am ready. Anything else to consider ?”

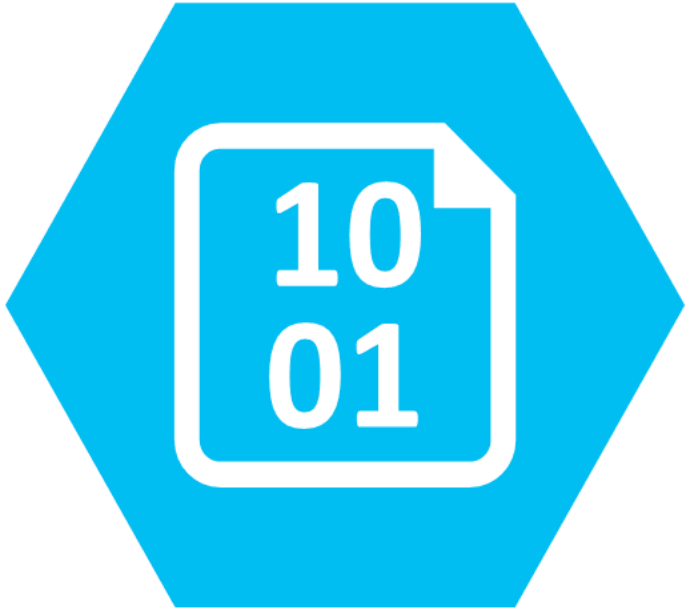
SQL is not the only option

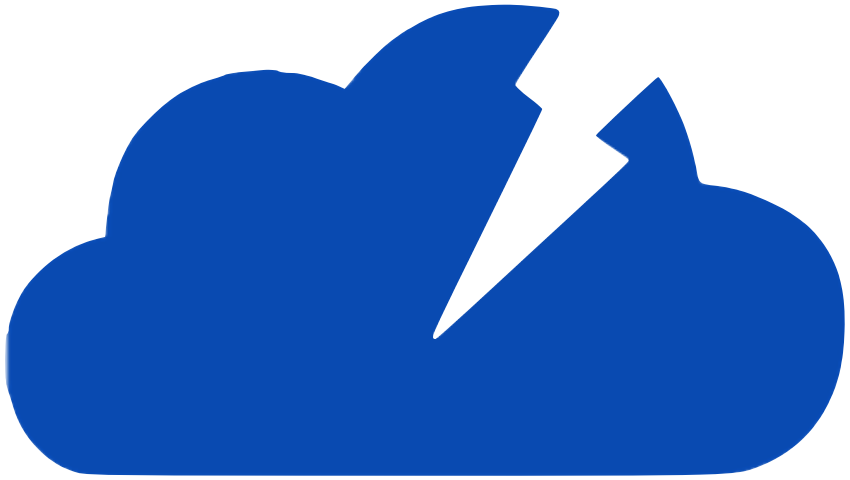


Do you need to build everything?



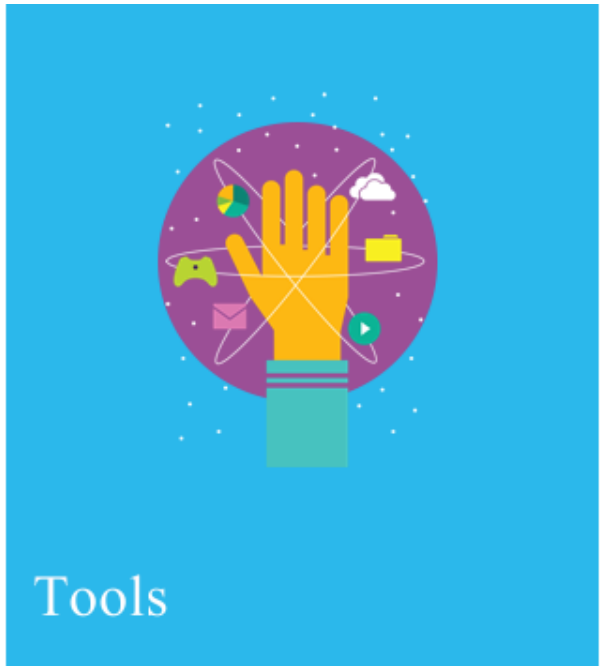
Do you need a web server?





Add Business Value

DevOps



Thanks

Most Important?

We can help you out!

matous.rokos@atea.no