HEXAWARE

Microservices with Service Fabric

Madan Sudarman

Azure Boot Camp 2018 - Hexaware, Chennai

Agenda

HEXAWARE

- Microservices Concepts
- Azure Service Fabric Overview
- Example: Customer Solution
- Getting Started

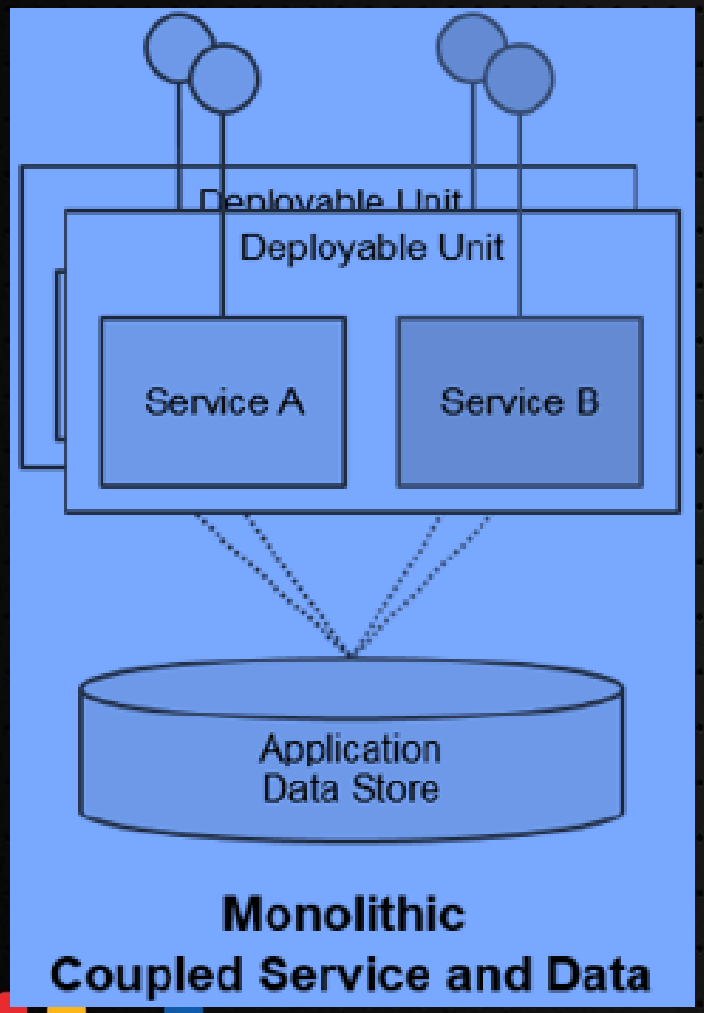
Microservices concepts

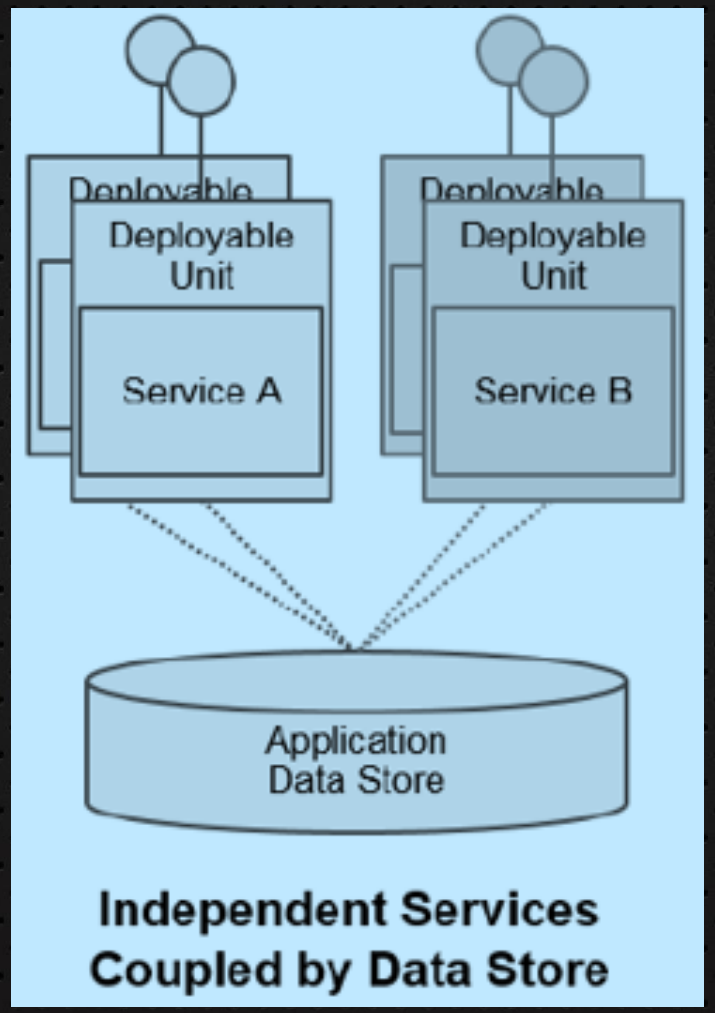
What is a Microservice?

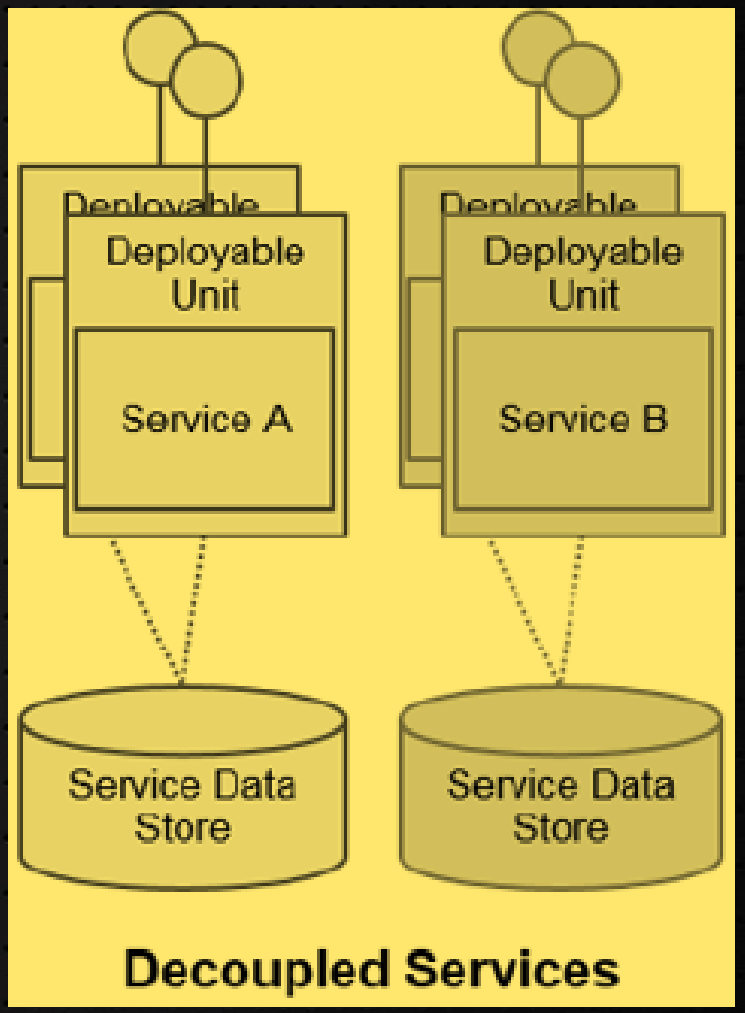
HEXAWARE

- Encapsulates a scenario
  - Single responsibility
  - Bounded context
- Contain code plus state that is independently versioned, deployed, and scaled
- Can be written in any language and framework
- Interact with other Microservices over well defined interfaces and protocols such as http (have a unique name - URL that can be resolved)
- Remains consistent and available in the presence of failures

Evolution







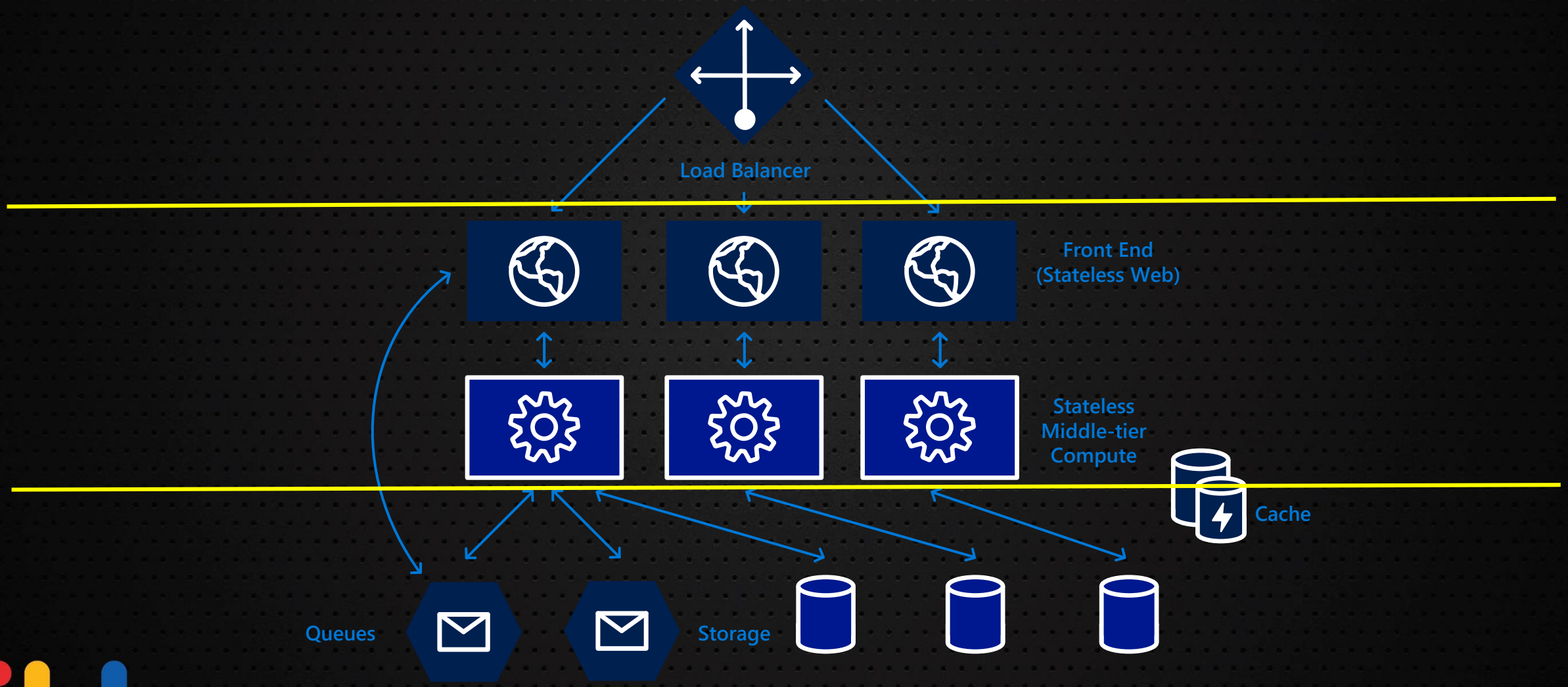
Credits: Gartner

Types of Microservices - Service Fabric perspective

HEXAWARE

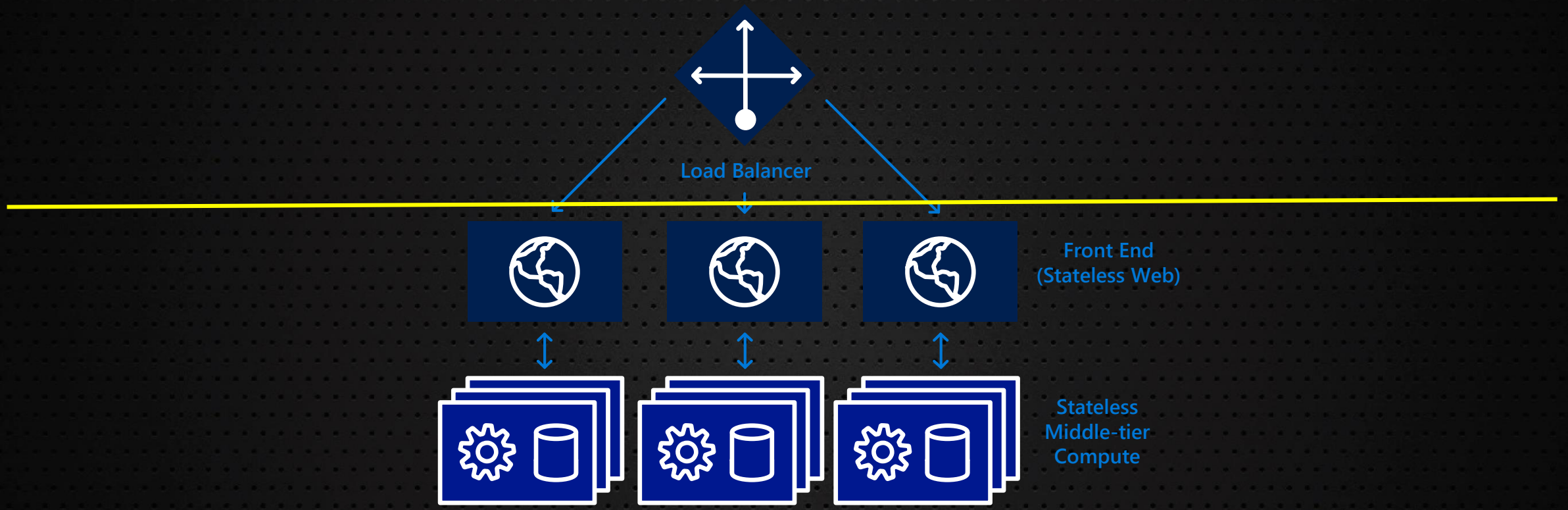
- Stateless microservice
  - Has either no state or it can be retrieved from an external store
  - There can be N instances
  - e.g. web frontends, protocol gateways, Azure Cloud Services etc.
- Stateful microservice
  - Maintain hard, authoritative state
  - N consistent copies achieved through replication and local persistence
  - e.g. database, documents, workflow, user profile, shopping cart etc.

Stateless microservices pattern



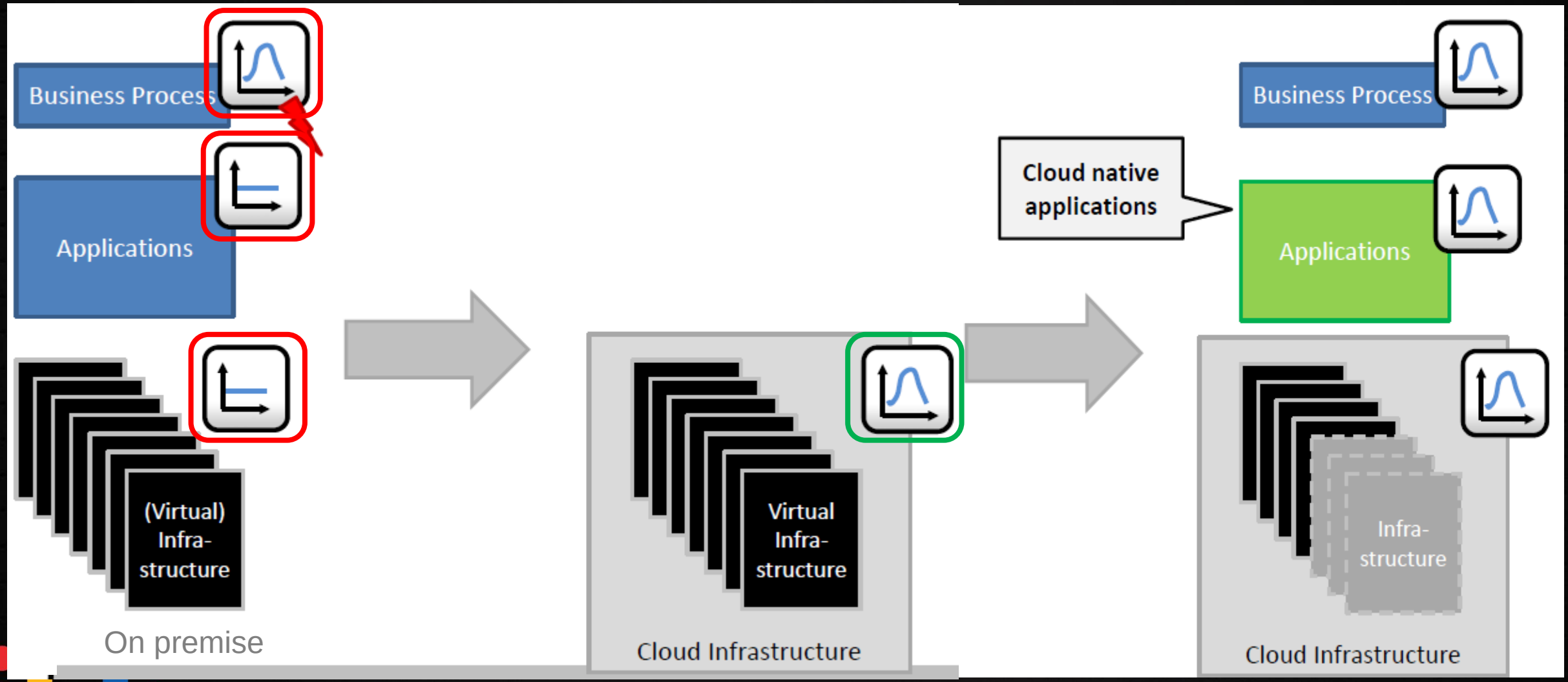
Credits: Microsoft Azure

Stateful microservices pattern



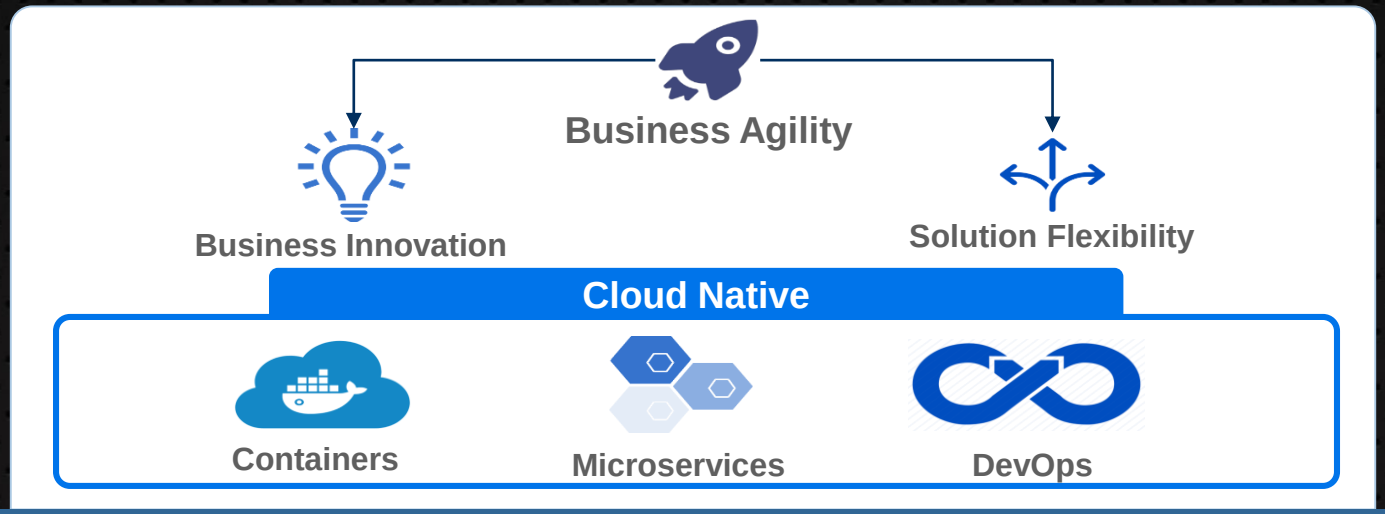
Credits: Microsoft Azure

When to consider Microservices



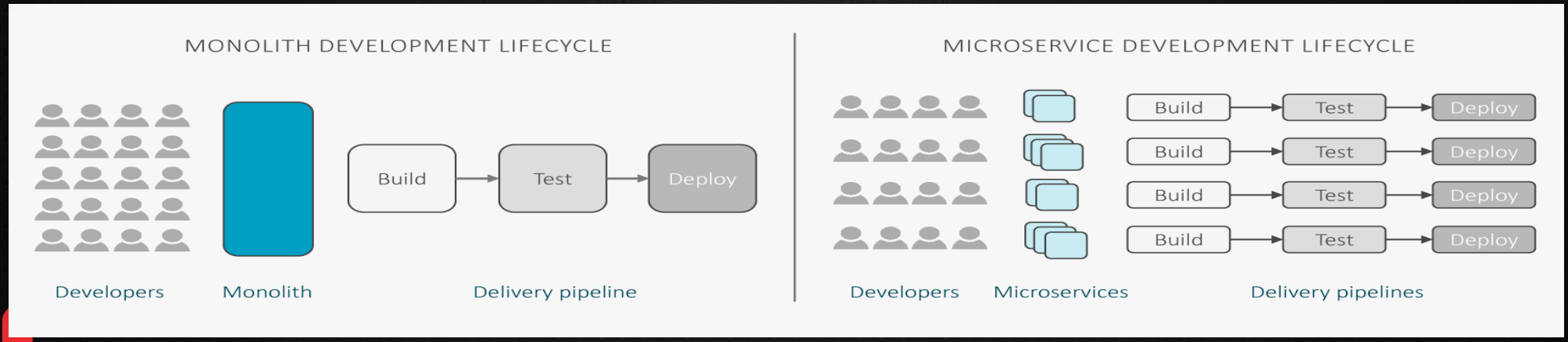
Credits: cloudcomputingpatterns.org

Cloud Native for Business Agility



Microservices architecture helps to continuously innovate by providing flexibility

Easier development & Testing | **Easy to upgrade** | **Agility & Continuous Delivery**



Azure Service Fabric - Overview

Cloud application development challenge

Balancing the needs of Business

- **Developer productivity** is essential for **business agility**
- Yet, keeping data and **apps stable**; and **cost optimal** is critical
- While you continue to **perform** to deliver value to your customers
- And **digitally transform** your business to **innovate**

Credits: Microsoft Azure

Azure Service Fabric

Build, deploy, and operate applications, using any OS, at any scale, on any cloud

Build

Build new or transform existing applications

Deploy

Deploy any code at any scale using tools you know

Operate

Run and secure services reliably at any scale

Credits: Microsoft Azure

To build, deploy, and operate…

…containers or microservices on any OS on any cloud

Programming Models

Dev & Ops Tooling

Orchestration

Lifecycle Management

Health & Monitoring

Always On Availability

Auto Scaling

Dev machine

Azure

On-premises infrastructure

Other clouds

Credits: Microsoft Azure

Build: data-aware microservices

Programming Models

Dev & Ops Tooling

Orchestration

Lifecycle Management

Health & Monitoring

Always On Availability

Auto Scaling

Reliable Actors

Use familiar tools: Visual Studio + Team Services for .NET or Jenkins + Yeomen for Java

Reliable Services

Manage state reliability without a database, lowering latency

Guest Executables

Run existing code and orchestrate life cycle using service fabric

Containers

Orchestrate your Windows Server or Linux containers reliably at scale

.NET or Java ...

Built-in ASP.NET core integration; work with VS and VSTS or Eclipse and Jenkins

Credits: Microsoft Azure

Deploy: any code on any OS

Programming Models

Dev & Ops Tooling

Orchestration

Lifecycle Management

Health & Monitoring

Always On Availability

Auto Scaling

CI/CD

Maximize uptime and scalability with isolated compute threads running concurrently

Docker Compose

Orchestrate existing container applications natively

Automate

Deploy or remove applications using PowerShell, CLI, Visual Studio, and other APIs

Rolling upgrades

Upgrade non-disruptively and roll-back in case of failures, automate with PowerShell

Monitor and diagnose

Generate, aggregate, and analyze events with built-in tooling and integration with Azure services

Credits: Microsoft Azure

Operate: on any cloud at any scale

Programming Models

Dev & Ops Tooling

Orchestration

Lifecycle Management

Health & Monitoring

Always On Availability

Auto Scaling

Use familiar tools

Such as Splunk, OMS, ELK, or AppInsights to gain deep insights or monitor application health

Use controlled chaos

Test graceful and ungraceful failure scenarios

Recover gracefully

Recover from node or service failure gracefully; replicate data automatically

Secure at scale

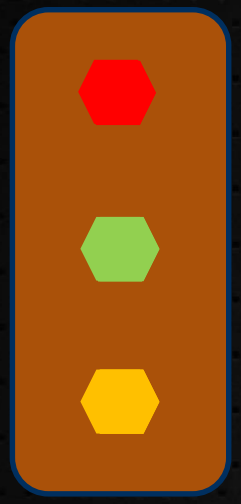
Secure node-to-node communication and user access using built-in capabilities

Scale programmatically

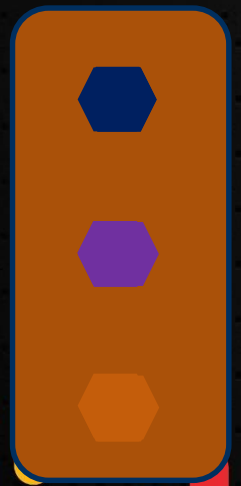
Use PowerShell, CLI, or APIs to scale programmatically achieving very high densities

Credits: Microsoft Azure

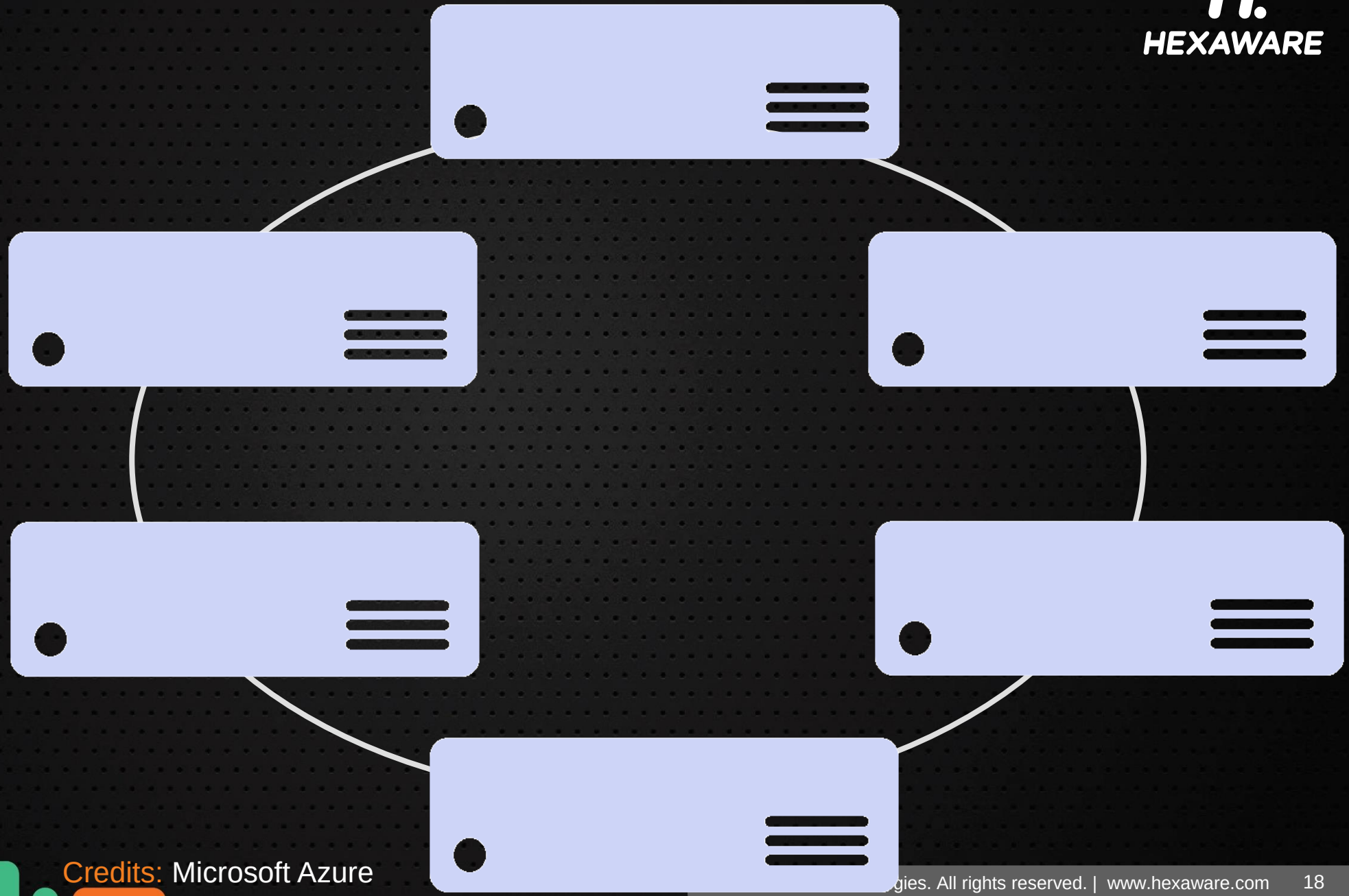
Service Fabric cluster with microservices



App1

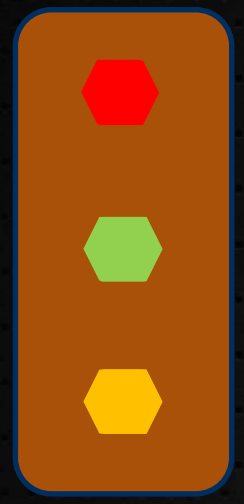


App2

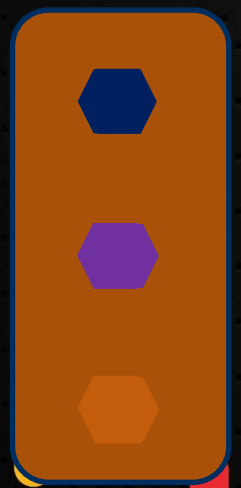


Credits: Microsoft Azure

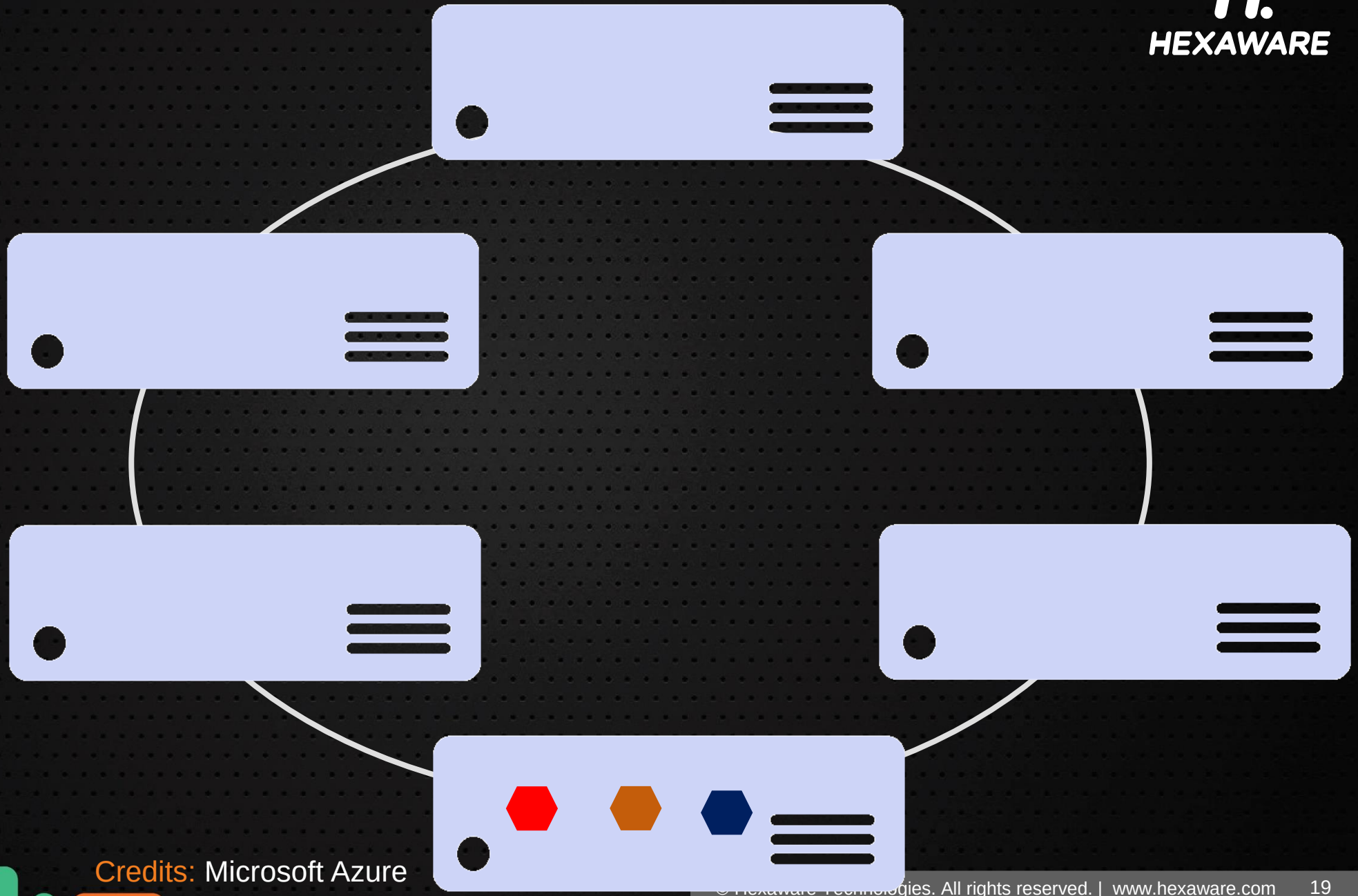
Handling machine failures



App1



App2



Credits: Microsoft Azure

Example:
Customer Solution

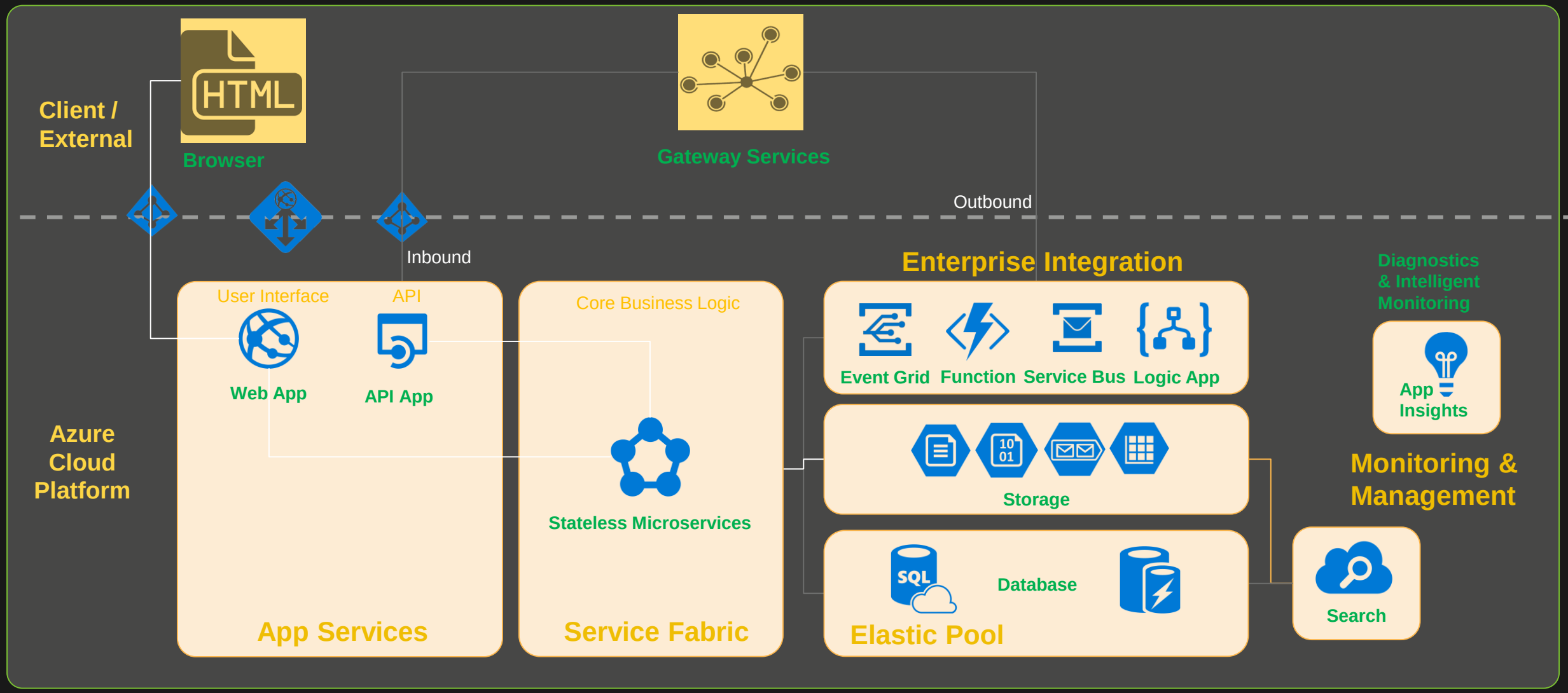
Proposed Solution & Key Design Principles

Proposed Solution

*fully developed, custom solution hosted in **Azure** leveraging **Service Fabric** as **Cloud Native Platform***

- Agile and robust delivery using Microservices
- Loosely coupled independent subsystems with clear purpose and scope
- Minimal technical diversity for ease of development & support (Microsoft Technology Stack)
- Data protection by Design By Default

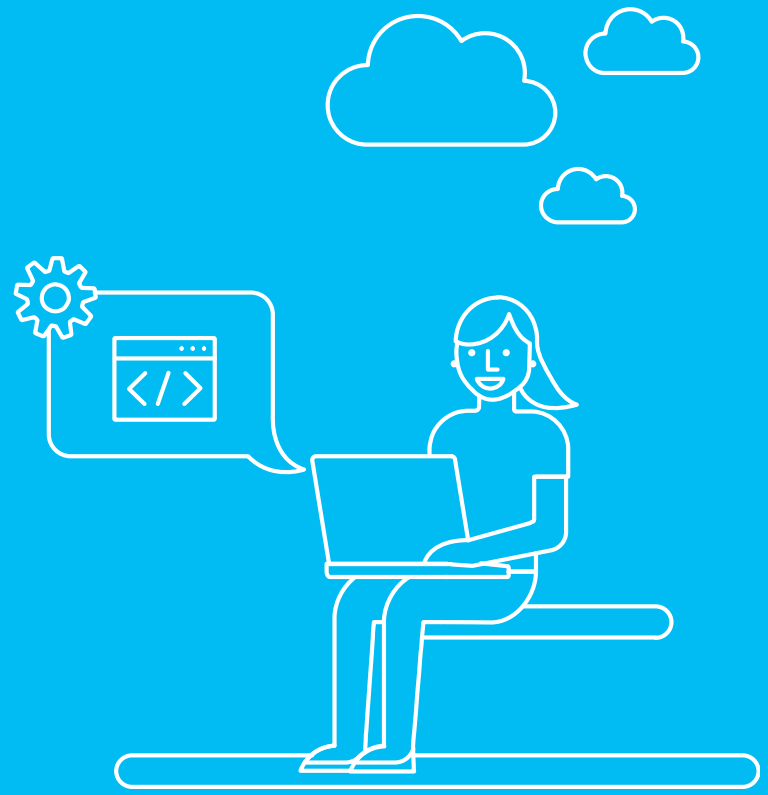
Solution Overview



Getting Started

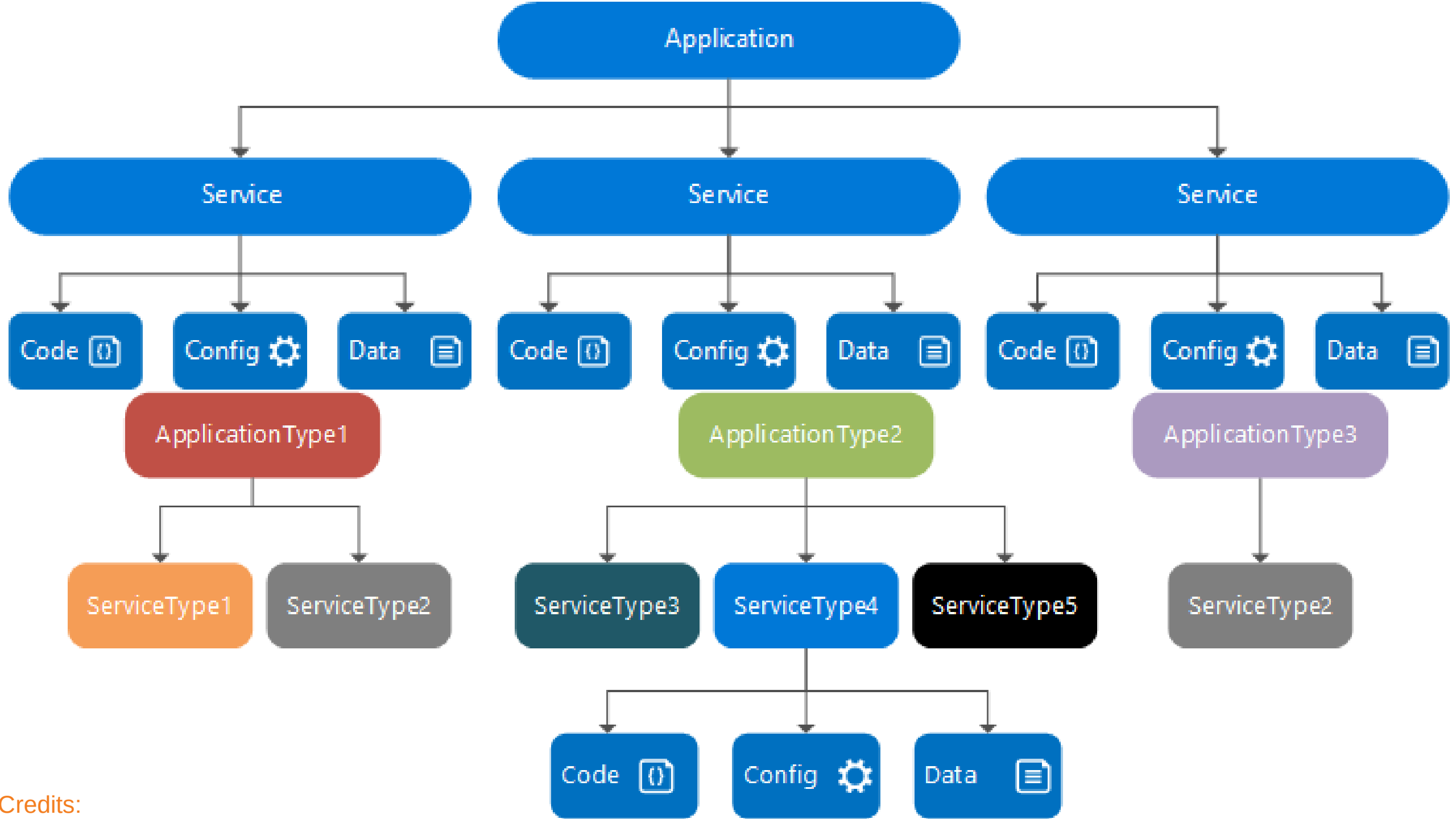
Awesome for developers

- Use .NET, Java or more
  - Stateless and Stateful services
  - ASP.NET core integration
- Open source programming models
  - Reliable services and actor programming model
- Visual studio integration
  - Use single node cluster for dev/test purposes
- Consistent experience across environments
  - Run the same environment on dev box as in production

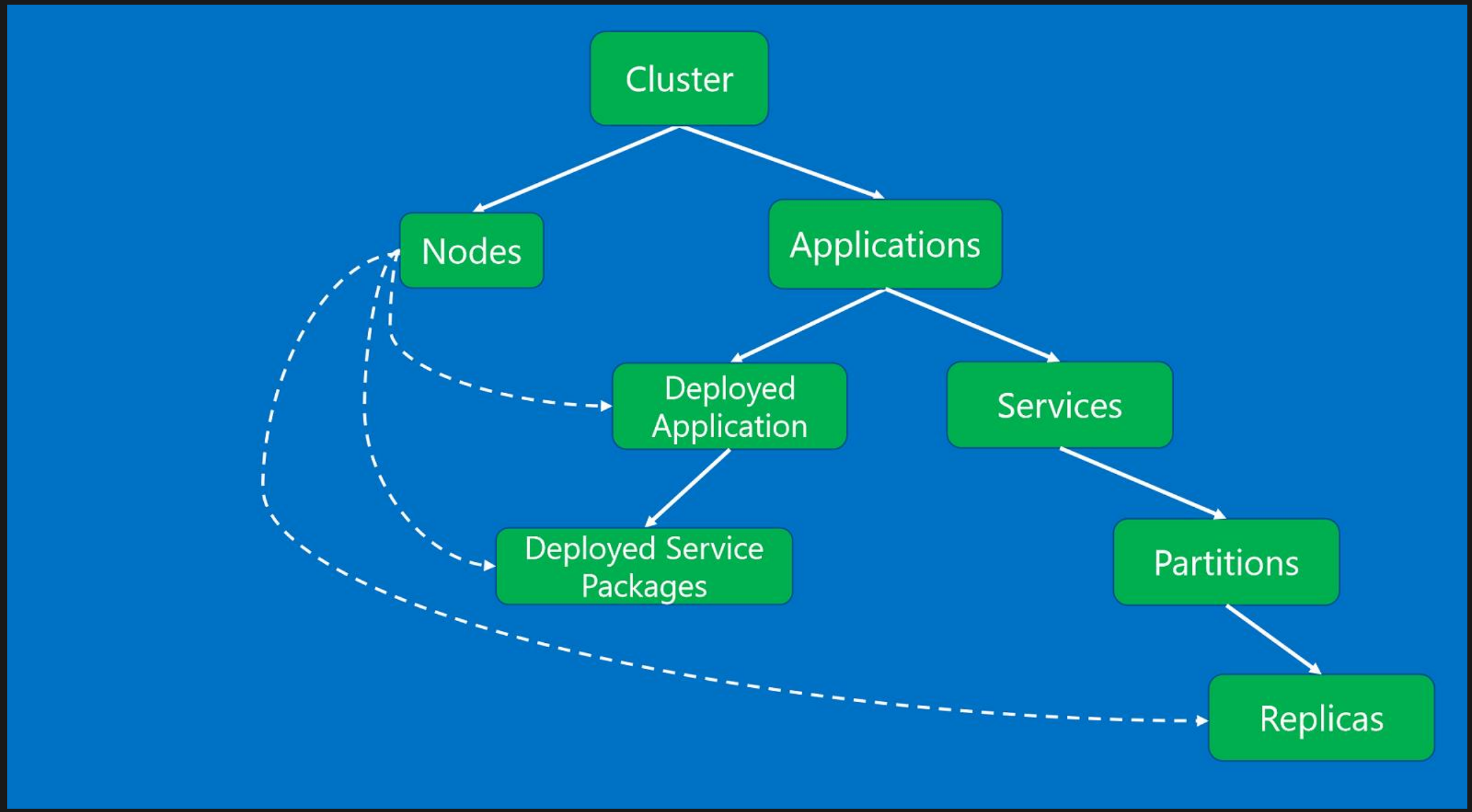


Credits: Microsoft Azure

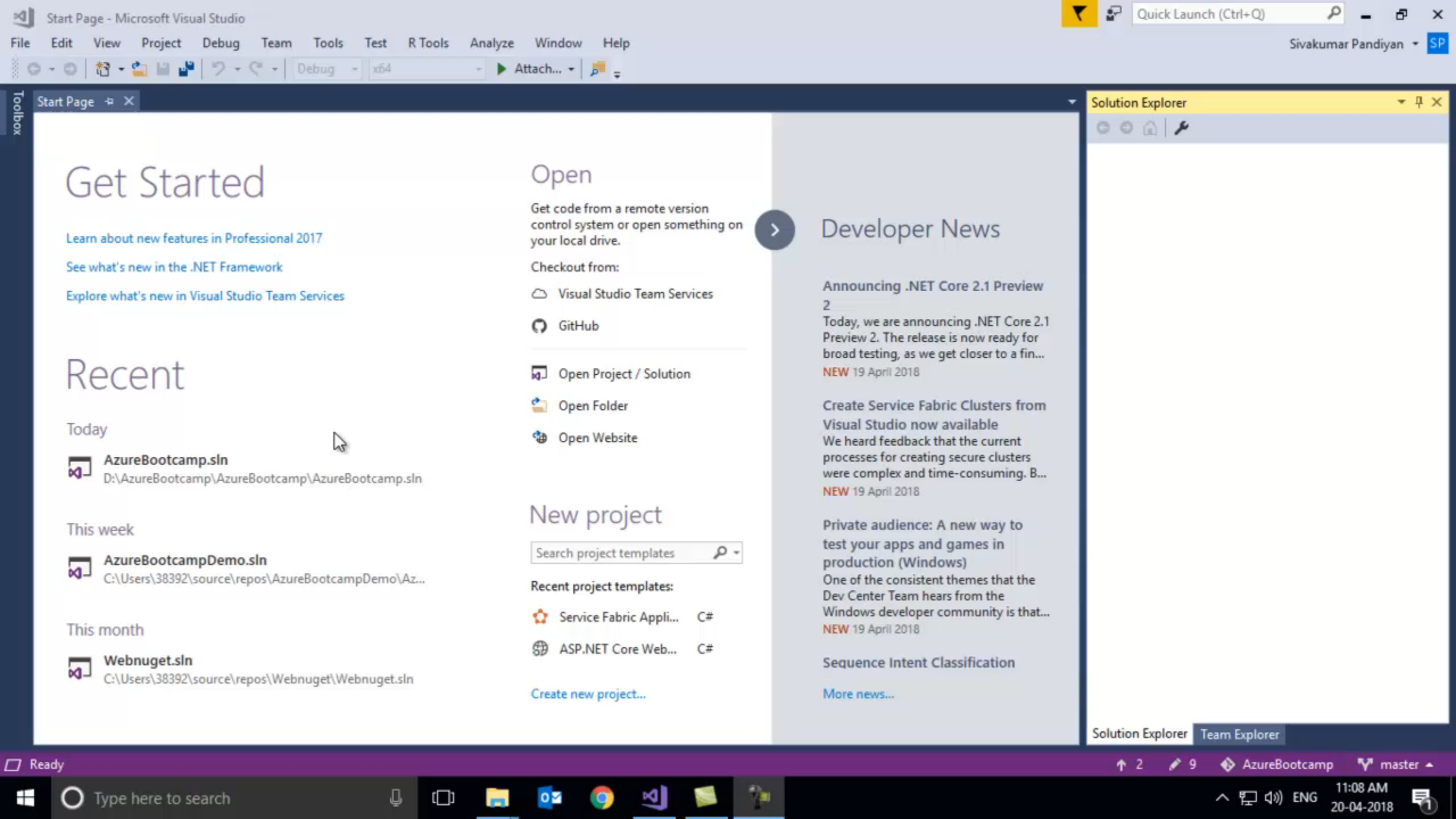
Demo


Application
Service
Service
Service
Code
Config
Data
Code
Config
Data
Code
Config
Data
ApplicationType1
ApplicationType2
ApplicationType3
ServiceType1
ServiceType2
ServiceType3
ServiceType4
ServiceType5
ServiceType2
Code
Config
Data

Credits:



Credits: Microsoft Azure


Start Page - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Debug Team Tools Test R Tools Analyze Window Help
Sivakumar Pandiyan
Debug
x64
Attach...
Toolbox
Start Page
Solution Explorer
Get Started
Learn about new features in Professional 2017
See what's new in the .NET Framework
Explore what's new in Visual Studio Team Services
Recent
Today
AzureBootcamp.sln
D:\AzureBootcamp\AzureBootcamp\AzureBootcamp.sln
This week
AzureBootcampDemo.sln
C:\Users\38392\source\repos\AzureBootcampDemo\Az...
This month
Webnuget.sln
C:\Users\38392\source\repos\Webnuget\Webnuget.sln
Open
Get code from a remote version control system or open something on your local drive.
Checkout from:
Visual Studio Team Services
GitHub
Open Project / Solution
Open Folder
Open Website
New project
Search project templates
Recent project templates:
Service Fabric Appli... C#
ASP.NET Core Web... C#
Create new project...
Developer News
Announcing .NET Core 2.1 Preview 2
Today, we are announcing .NET Core 2.1 Preview 2. The release is now ready for broad testing, as we get closer to a fin...
NEW 19 April 2018
Create Service Fabric Clusters from Visual Studio now available
We heard feedback that the current processes for creating secure clusters were complex and time-consuming. B...
NEW 19 April 2018
Private audience: A new way to test your apps and games in production (Windows)
One of the consistent themes that the Dev Center Team hears from the Windows developer community is that...
NEW 19 April 2018
Sequence Intent Classification
More news...
Solution Explorer
Team Explorer
Ready
2
9
AzureBootcamp
master
Type here to search
ENG
11:08 AM
20-04-2018

Key Takeaway

Key Takeaways

- Microservices are key for scalable and evolving applications
- Service Fabric is a platform for building applications with a microservices design approach

https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-get-started

Q & A

Thank you