2017

Global Azure BOOTCAMP

autoexec.gr

SQLschool.gr

dotNETZone.gr
η δύναμη των ελλήνων προγραμματιστών

Global Azure Bootcamp

One whole day devoted to Cloud and Microsoft Azure

GAB Athens 2017 - Agenda

2017 Global Azure BOOTCAMP

- **Azure Service Fabric: Microservices Architecture made simple** (*Paris Polyzos*)
- **Build Container as a Service (CaaS) solutions with Azure Container Service** (Kostas Pantos)
- **Azure Application Insights** (*Dimosthenis Stellakis*)
- **WebApp on Linux Service, First Look** (*speaker: George Capnias*)
- **Hybrid IT – Hybrid Datacenter** (*Vaggelis Kapsalakis*)
- **Conversations as a Platform: Bots and AI powered by Cloud** (*Sophie Chanialaki*)
- **Azure Mobile Apps and Xamarin: From zero to hero** (Nasos Loukas)
- **Design for scalability and High Availability on Microsoft Azure** (Vaggelis Kappas)
- **Azure IoT - IoT Hub, Device Management and IoT SDK's** (Vasilis Aivalis)
- **Azure SQL Data Warehouse** (Antonis Chatzipavlis)
- **Azure Active Directory: 5 reasons to implement it today!** (Chris Spanougakis)
- **Data Science with Azure Machine Learning and R** (Christos Charmatzis)
- **Protect your data with a modern backup, archive and disaster recovery solution** Pantelis Apostolidis

Lab and Giveaway

GAB Athens 2017

Sponsors

2017 Global Azure BOOTCAMP

Service Fabric
Microservices Architecture made simple

Paris Polyzos
Software Engineer @ Stoiximan
Microsoft Azure MVP

2017 Global Azure BOOTCAMP

Senior Software Engineer

@ Stoiximan

github.com/ppolyzos

ppolyzos.com

AzureHeads.gr

Paris Polyzos

@ppolyzos

Microsoft CERTIFIED Professional

AzureHeads

MVP Microsoft Most Valuable Professional

Agenda

- Traditional Architectures & Monolithic Applications
- Microservices
- Azure Service Fabric
- Azure Service Fabric Cluster
- Application & Programming Model
- Demos

What does Cloud mean?

On Demand
Cloud Services are available on demand. Whether that is VM, File Storage or Data Processing

Distributed
Cloud Services are distributed across many fault zones to ensure high uptime

Horizontal Scale
Scale out instead of scale up

Built for Failure
Because of the transient nature of resources in a cloud environment services will eventually fail

Not a Place
Not only Azure or AWS

What is a Microservice?

Simply put, it's a self contained, domain specific, service

What is a Service?

Simply put, **something** that does a body of work

Has an input, performs some body of work and produces an output

Traditional 3-Tier Architecture

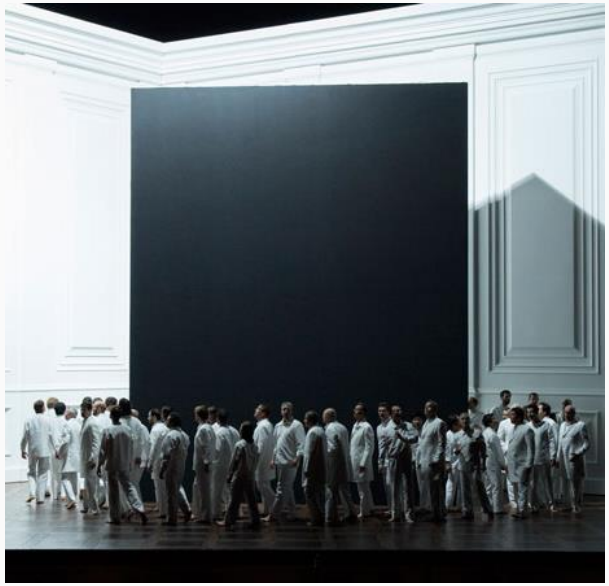
Monolithic Application

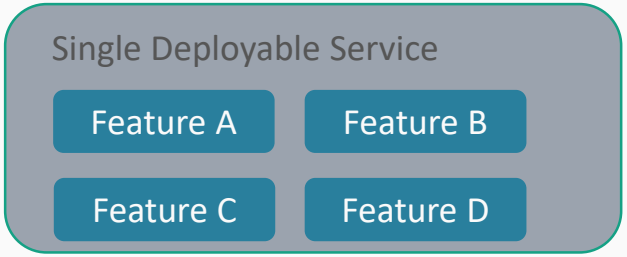
Advantages

Single Deployable
Deploy and manage one application

High Initial Agility
Few things to wire up at the beginning

Broad Changes Made Easily
Change a big part of the source code, all in one place





Challenges

Inconsistent State
Feature A is ready to go but other features are not ready yet

Testing
Test entire surface area of the application

Scale
Cannot scale the areas of the application that create “hot zones” independently

Microservices: Different Paradigm

"I don't need to know everything, I just need to know where to find it when I need it"

Albert Einstein

Focused NOT on **features** but on a given business or technical **domain** in which they operate.

What is a Business Domain?

An area of the business that tends to use similar terminology to describe a business/technical need

Customers
Customer management and information retrieval

Payments
Customer financial information and payments

Authentication
User authentication into the system

Logging
Focus on the logging of events

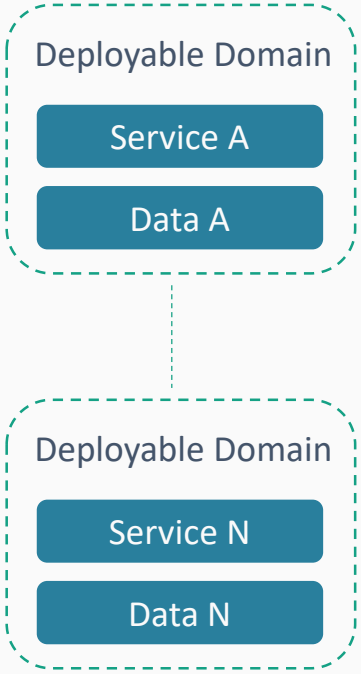
Microservice Application

Advantages

Physically Decoupled
If one service fails it does not bring the whole system down

Testing
You only have to test one domain

Scale
Easily scale a specific application's portion



Challenges

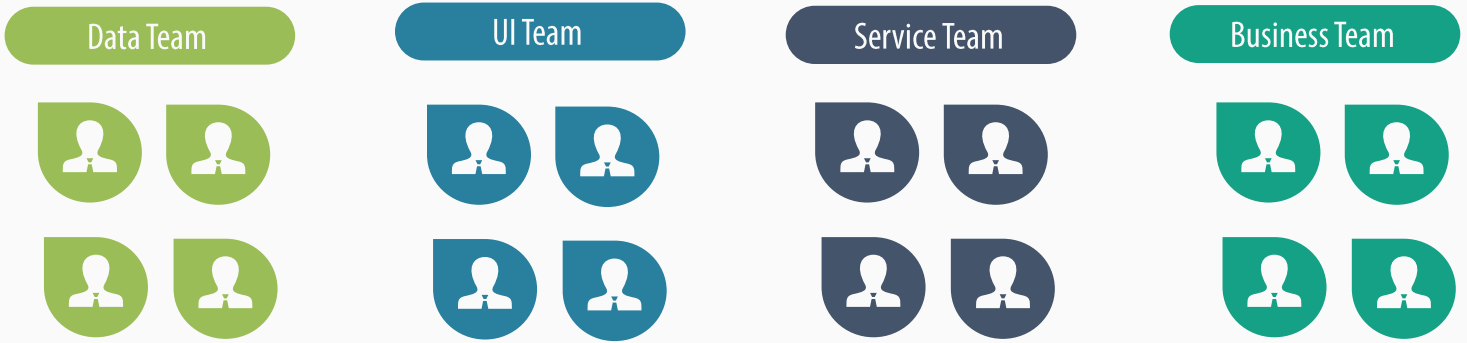
Culture Shift
Mentality shift within the organization

Large Footprint
More moving parts to keep track of.
Embrace automation

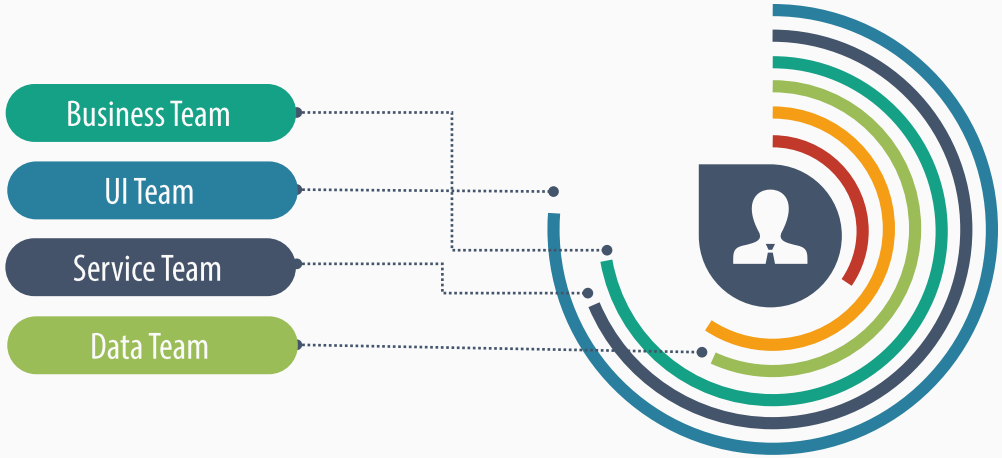
Slower Initial Agility
Lots of things to wire up

Microservices by Domain

Team Agility



Microservices by Domain

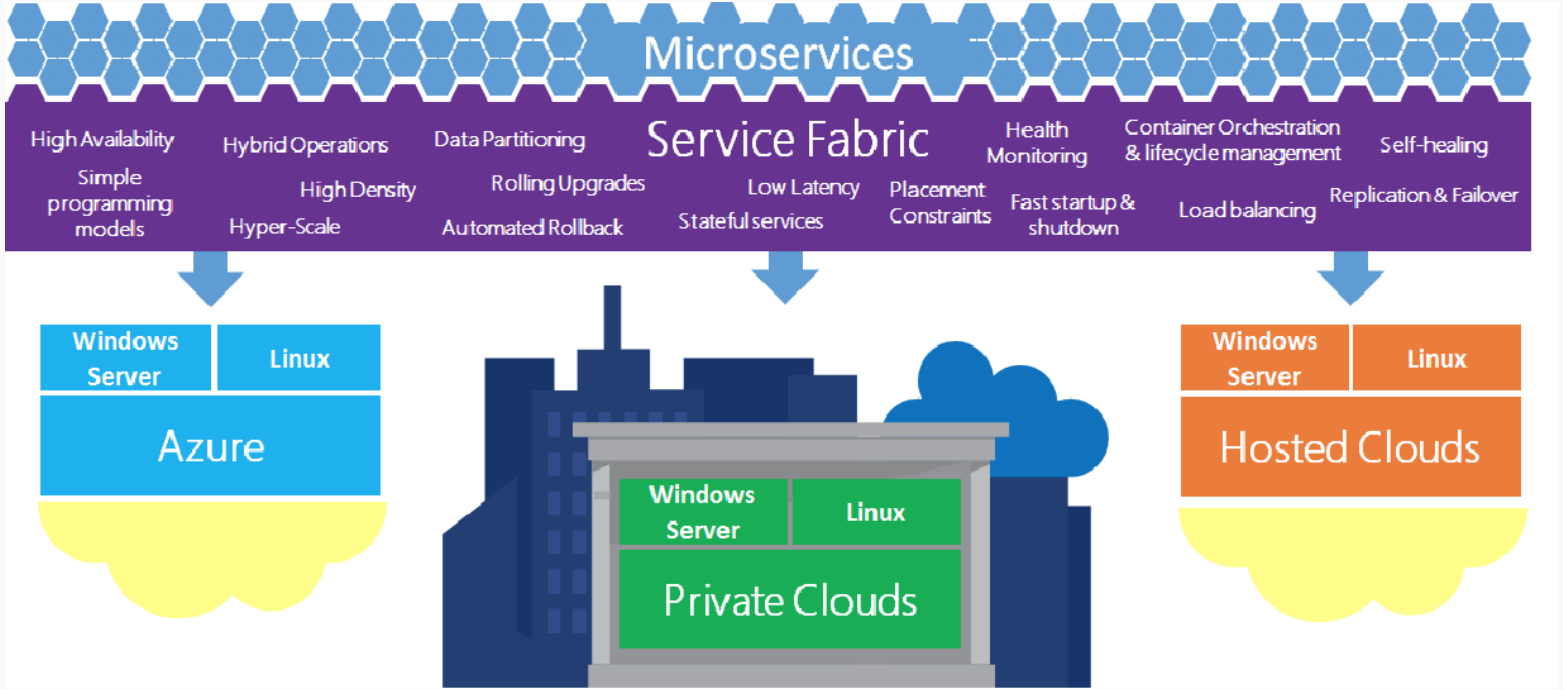


End-to-end ownership of a team

Highly concurrent deployment scenarios

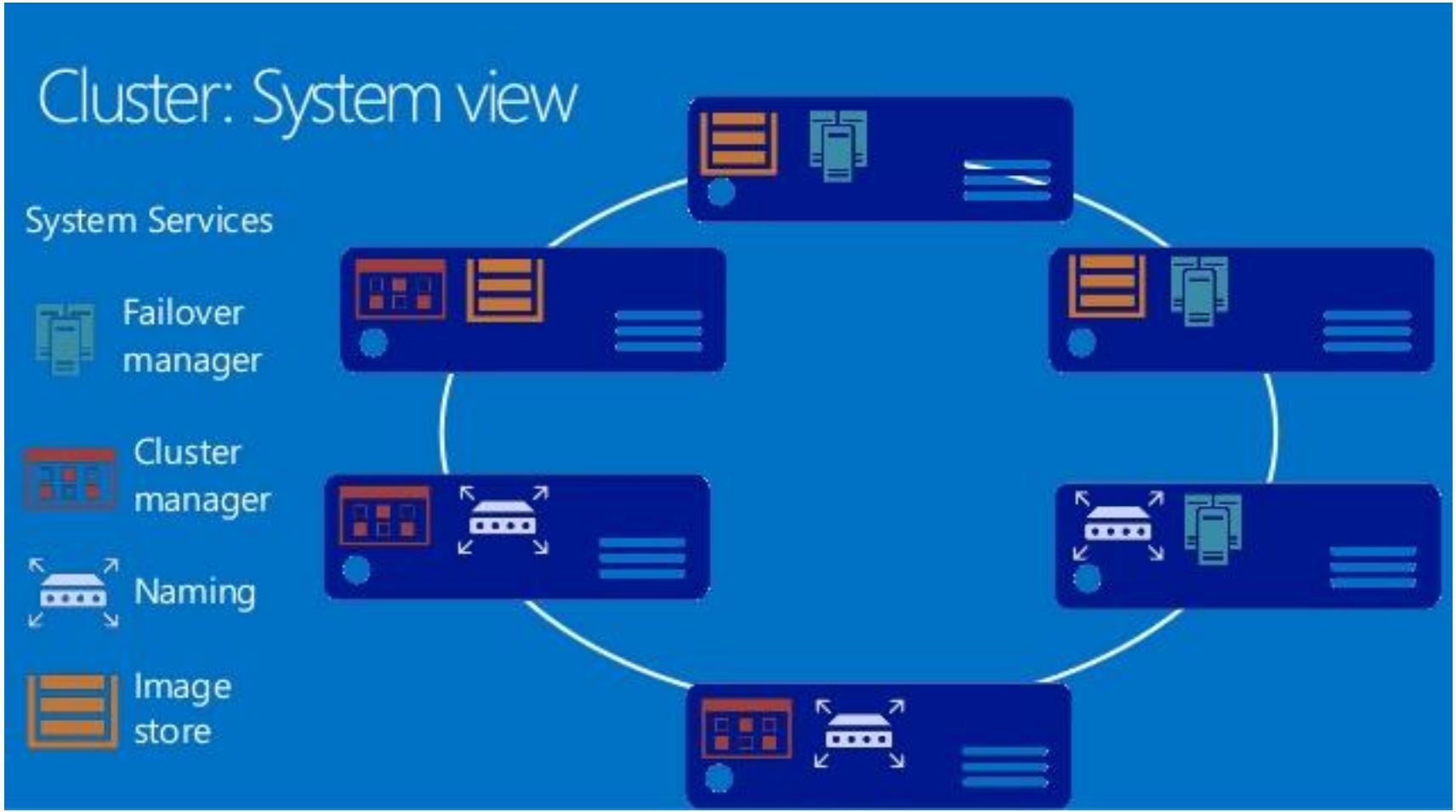
Better isolation
Teams care about what is happening within their domain

Service Fabric

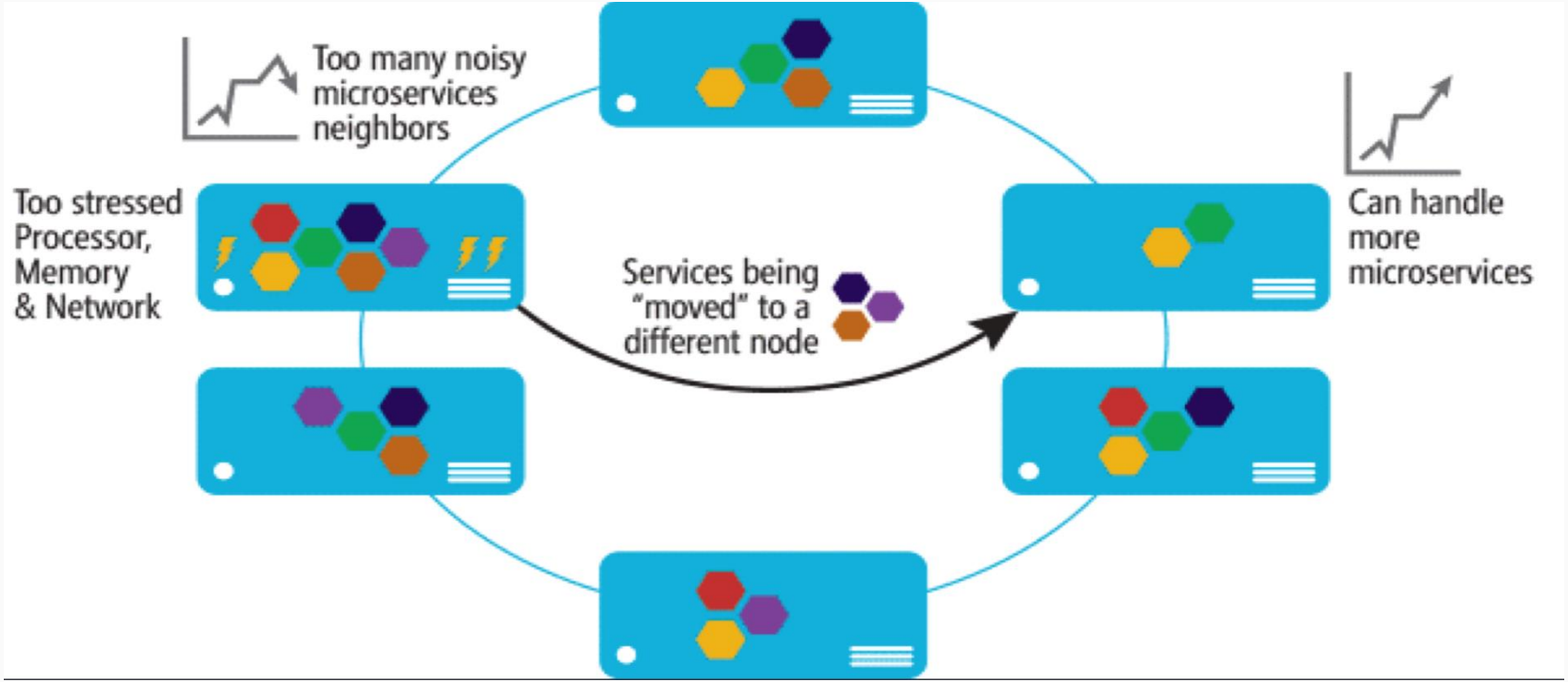


Service Fabric enables you to build and manage scalable and reliable applications composed of microservices that run at very high density on a shared pool of machines which is referred to as a cluster.

Service Fabric Cluster



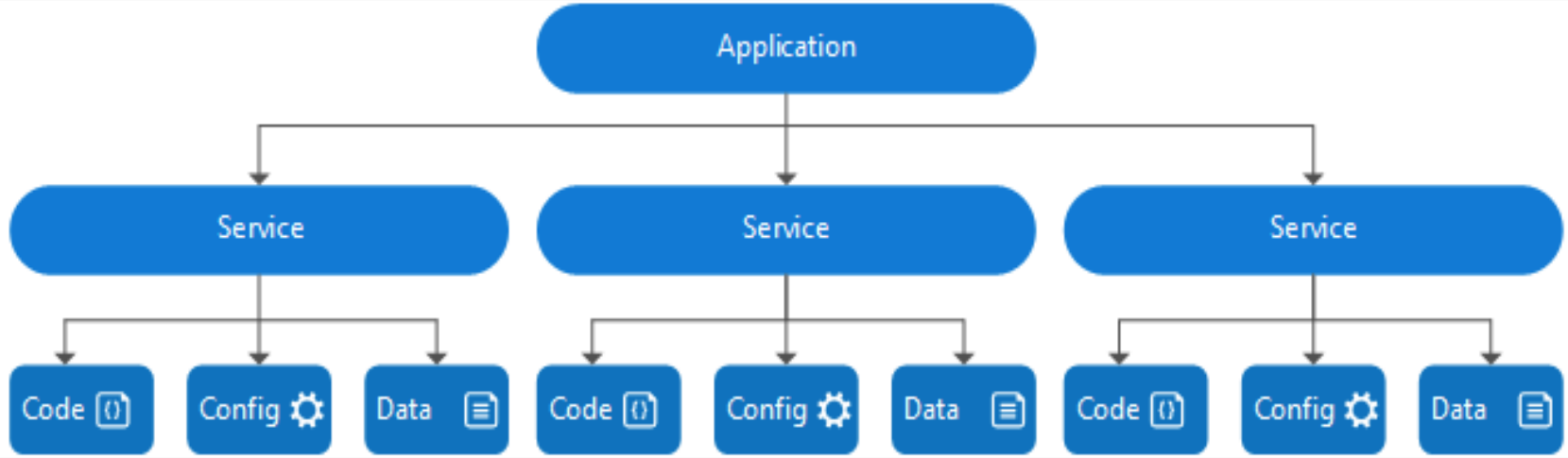
Service Fabric



Service Fabric – Capabilities

- Application Deployment Services
  - Rolling update with rollback
  - Strong Versioning
  - Side-by-side support
- Name service for discovery of applications
- High density
- Load balancing and placement constraints
- Consistent state replication framework
- Reliable distributed key/value store, collections and queues

Service Fabric – Application Model



Service Fabric – Programming Model

Guest Executable
Any executable, written in any language. It can be packaged in an application and hosted alongside other services

Stateless Reliable Services
State is persisted to external storage, such as Azure DB or Azure Storage

Stateful Reliable Services
Reliability of state through replication and local persistence

Reliable Actors
On top of Reliable Services. This framework implements the Virtual Actor pattern. Independent units of compute and state with single-threaded execution

Docker & Windows Containers (preview)
Future improvements to networking, resources constraints, security, diagnostics, volume drivers, and tooling support in Visual Studio

Service Fabric

Thank You

Evaluation Survey < https://aka.ms/cc9cf1 >

Service Fabric – Node Processes