SERVICE FABRIC

Hands-On

Simon Dale

“I'm a Technical Architect with BJSS currently working on Microsoft Azure, Service Fabric and Mixed Reality with the .NET tech stack. I’ve always had an interest for distributed computing, data and producing scalable, performant software.”

bjss

@simondale_
https://github.com/simondale
https://linkedin.com/in/simonjdale

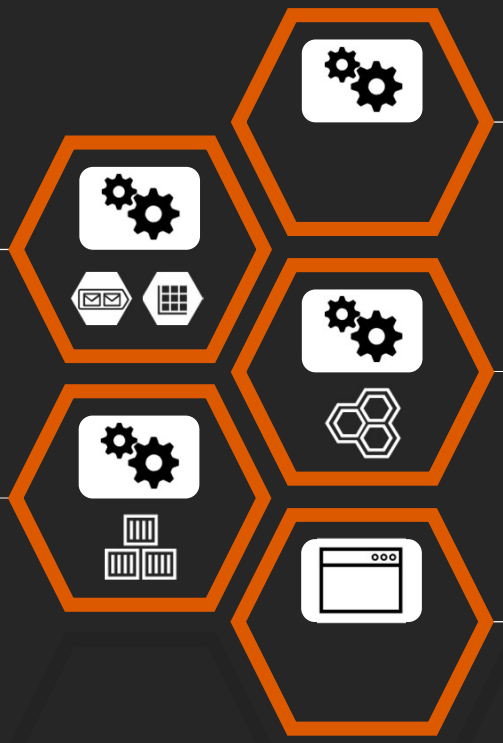
SERVICE FABRIC

- Distributed Systems Platform
- Scalable, Reliable Microservices and Containers
- Next Generation, Enterprise Class, Cloud Scale
- Application Platform Layer
- Runs on a cluster of machines
- Run anywhere

SERVICE APPROACH

State lives in compute tier
Low latency reads and writes
Partitions for horizontal scale
Data stores for analytics and DR

Stateful



Stateless

State lives within actor instances
Massively scalable, low overhead
Synchronised concurrency
Out of box messaging

Reliable Actors

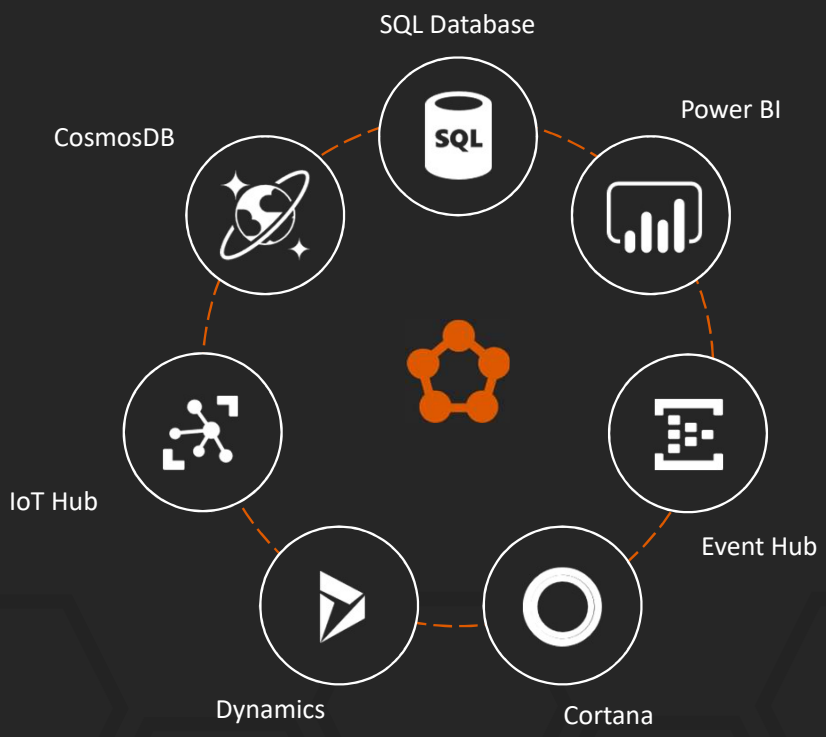
Containers

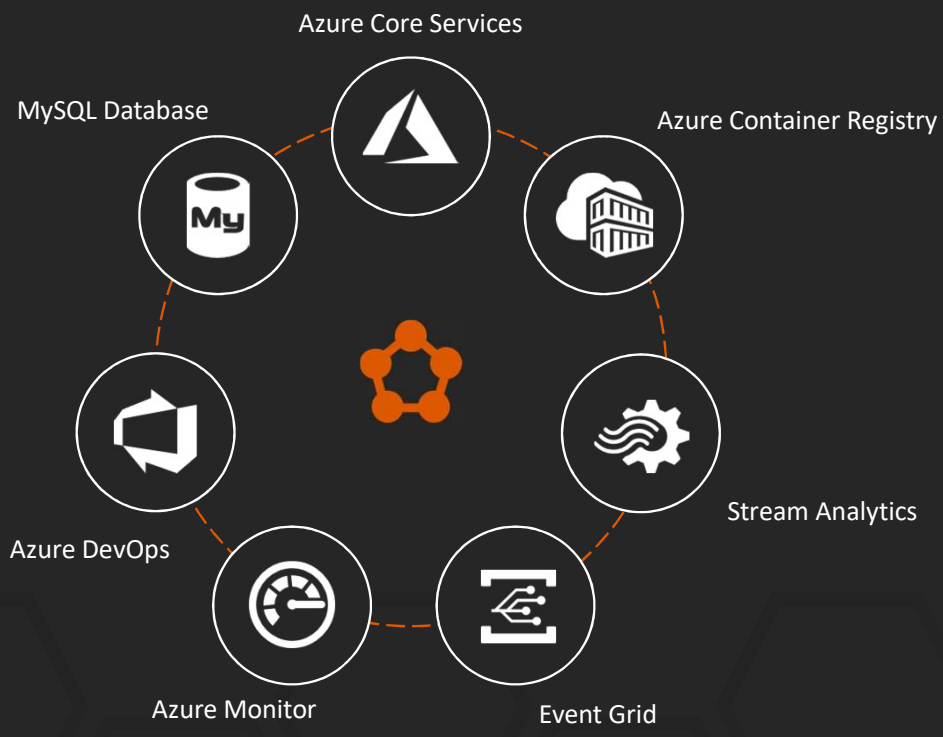
Encapsulate key dependencies
Reuse existing estate
Additional isolation from host
Benefit from placement and scale

Reuse existing applications
Lightweight process execution
Maximise compute density
Benefit from placement and scale

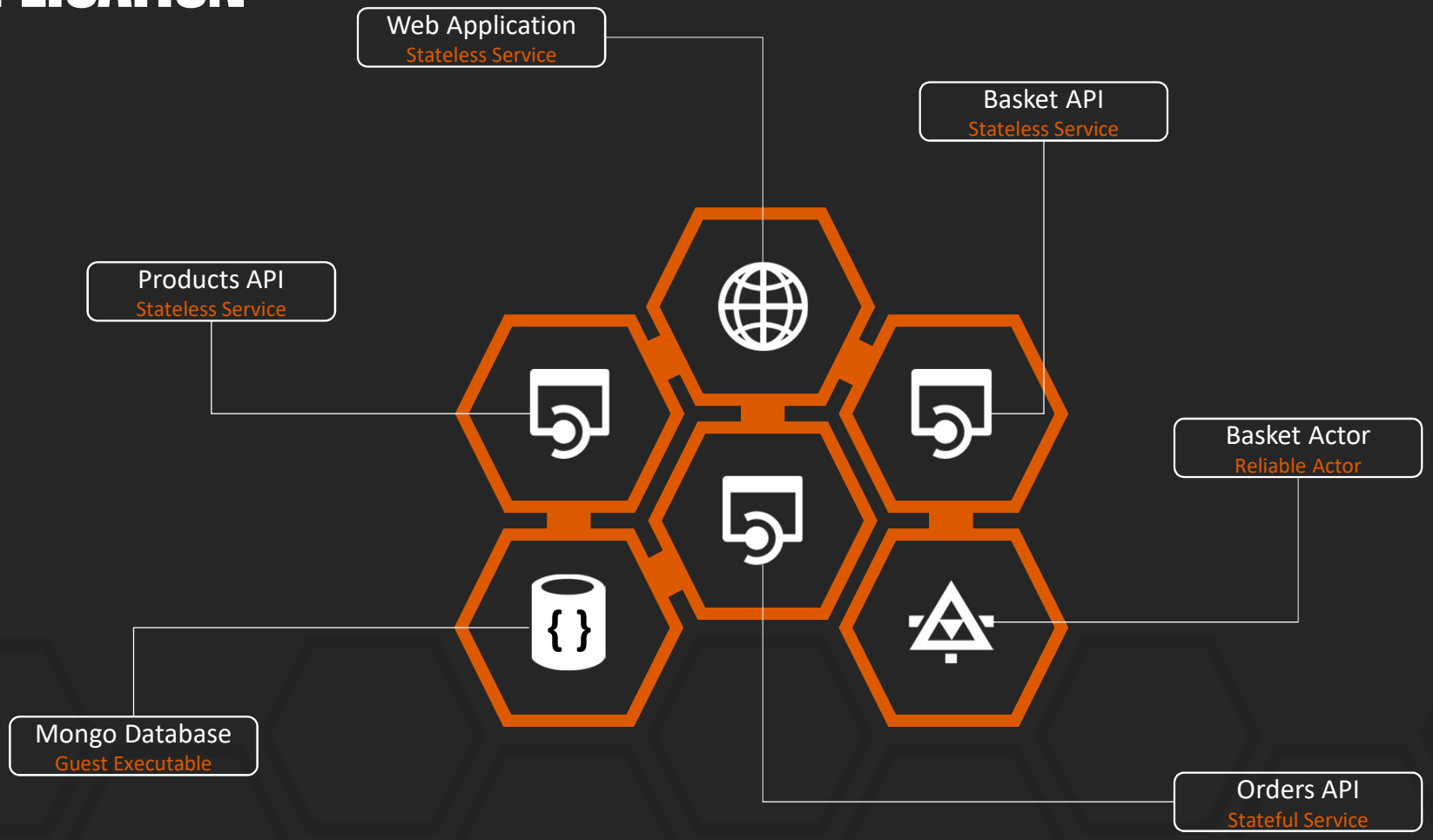
Guest EXE

POWERING AZURE SERVICES

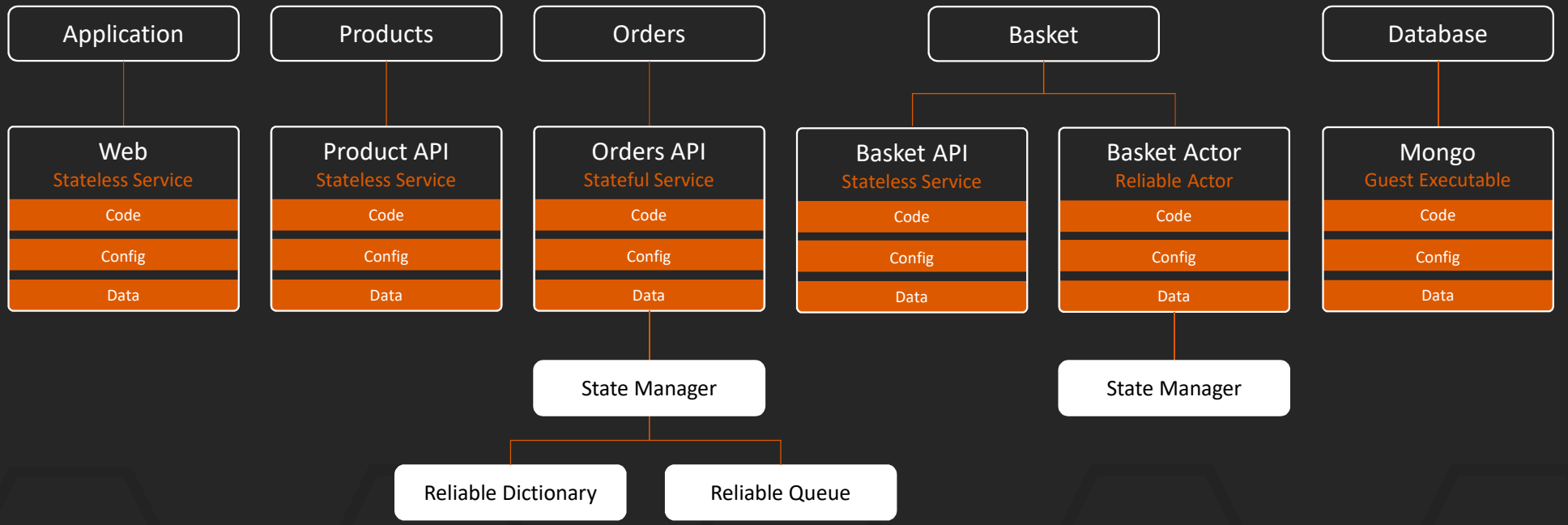




APPLICATION

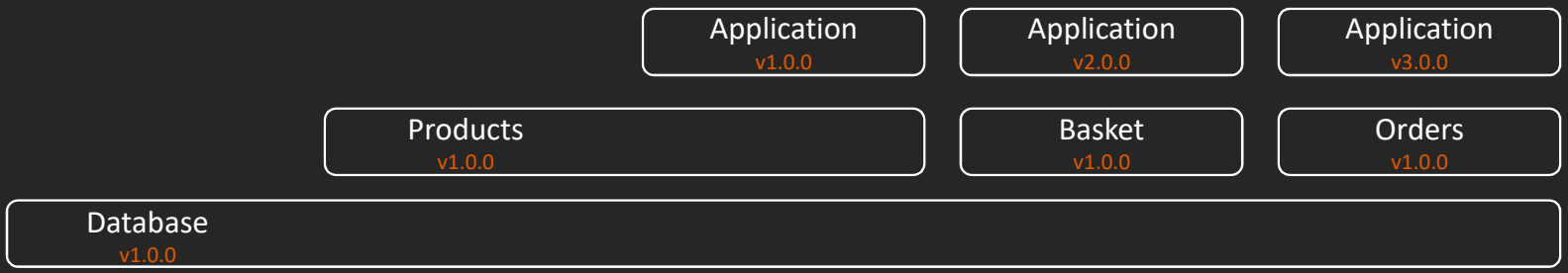


DEPLOYMENT

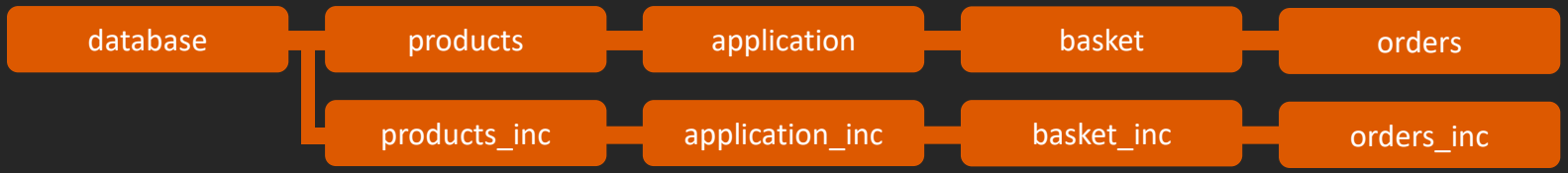

Application
Products
Orders
Basket
Database
Web
Stateless Service
Code
Config
Data
Product API
Stateless Service
Code
Config
Data
Orders API
Stateful Service
Code
Config
Data
Basket API
Stateless Service
Code
Config
Data
Basket Actor
Reliable Actor
Code
Config
Data
Mongo
Guest Executable
Code
Config
Data
State Manager
State Manager
Reliable Dictionary
Reliable Queue

WORKSHOP

Service Fabric



Branches



PREREQUISITES

- Visual Studio Code or Community Edition
  https://visualstudio.microsoft.com
  https://code.visualstudio.com
- Service Fabric SDK
  https://www.microsoft.com/web/handlers/webpi.ashx?command=getinstallerredirect&appid=MicrosoftAzure-ServiceFabric-CoreSDK
- Azure Subscription
  https://portal.azure.com
- Azure DevOps Account
  https://dev.azure.com

DATABASE GUEST EXECUTABLE

```
$ git checkout database
```

- MongoDB
- Guest executable (mongod)
- Test deployment using client (mongo)
- Roadmap to replace this service with Azure PaaS (CosmosDB)

STATELESS PRODUCTS WEB API

```
$ git checkout products
```

- ASP.NET MVC WebAPI
- Single controller (Products)
- MongoDB repository for CRUD operations
- Settings containing connection details
- Test using curl or PowerShell

STATELESS APPLICATION

```
$ git checkout application
```

- ASP.NET MVC Web Front-End
- Implement a single controller for Products
- Support CRUD operations
- Link to Product API microservice

BASKET RELIABLE ACTORS

```
$ git checkout basket
```

- ASP.NET MVC WebAPI
- Service Fabric Reliable Actor
- Calls to API request Actor Proxy and issue requests
- Reliable actors maintain state linked to user session
- Expiry timeout for actors
- Update Web Application to add items to Basket

STATEFUL ORDER SERVICE

```
$ git checkout order-processing
```

- ASP.NET MVC WebAPI
- Stateful Service
- Support competing consumers through Reliable Queue
- Persist order details and statistics to MongoDB
- Maintain current statistics in Reliable Dictionary

DEPLOYMENT SCRIPTS

```
$ git checkout deployment
```

- Stand-Alone Cluster Support
- PowerShell Script
- Multiple Applications

DEPLOY TO AZURE

```
$ az sf create
```

- Stand-Alone Cluster Support
- PowerShell Script
- Multiple Applications

AZURE DEVOPS INTEGRATION

Login to https://dev.azure.com
Create a new team project
Ensure git source control provider

Create a repository
Import existing code from github
https://github.com/simondale/service-fabric-workshop

Create build pipeline
Select Service Fabric Build template
Add task for each application in repository
Enable continuous integration

Create release pipeline for the build
Select Service Fabric Deploy template
Add task for each artefact in drop
Configure variables and parameter overrides

LOOKING BACK

- What/why/how of Service Fabric
- Service types and deployment
- Built a sample application
- Pipelines with Azure DevOps
- Continuous integration