Azure Serverless & Microservices Briefing

Jan 18th 2018

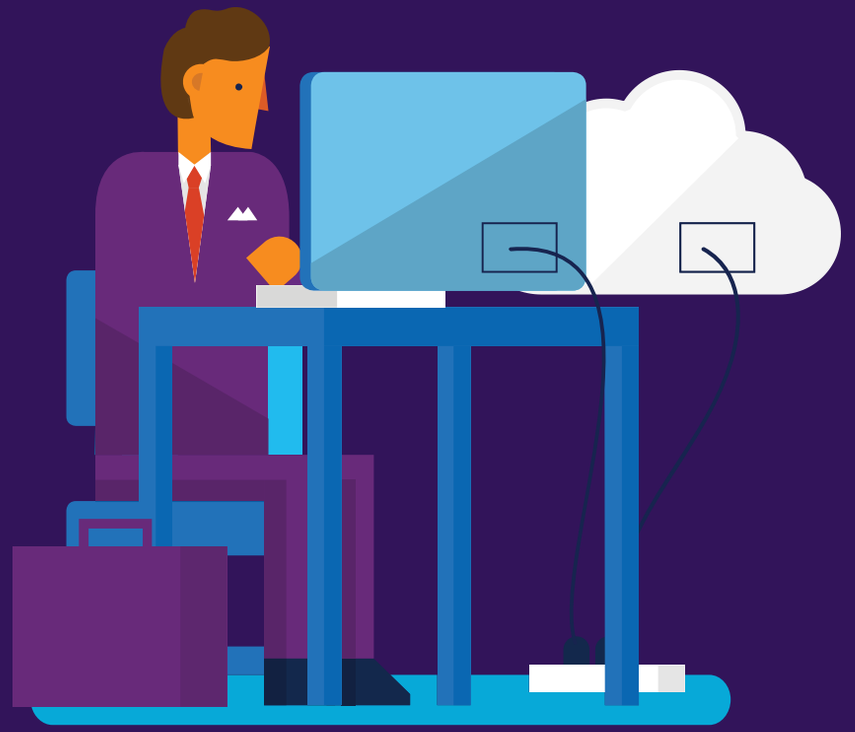
Ross.P.Smith@microsoft.com
David.Gristwood@microsoft.com
Ben.Coleman@microsoft.com

Siddhartha.Chaturvedi@microsoft.com

Microsoft

http://aka.ms/azureevent

#Azure
#BuildWithAzure



vipazure@microsoft.com

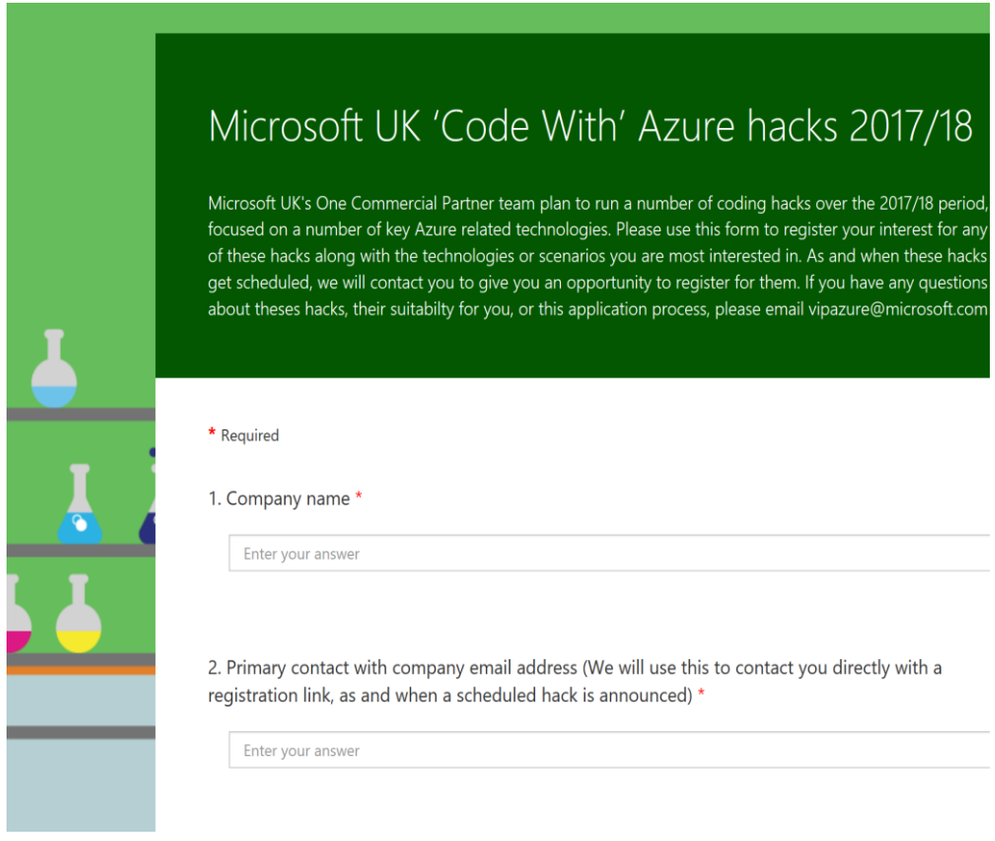
Agenda

- Microservices & Serverless overview
- Smilr – our demo app for the day
- Smilr & Node, ComosDB and Azure PaaS
- Containers overview
- Smilr & Service Fabric
- *(Breaking for Lunch around 1pm)*
- Smilr & Kubernetes
- Smilr & Azure Functions
- Q&A
- *Finish around 4pm*

Other technical activities from our team

Code with hacks

https://aka.ms/ukocpisvhack0817



Microsoft UK 'Code With' Azure hacks 2017/18

Microsoft UK's One Commercial Partner team plan to run a number of coding hacks over the 2017/18 period, focused on a number of key Azure related technologies. Please use this form to register your interest for any of these hacks along with the technologies or scenarios you are most interested in. As and when these hacks get scheduled, we will contact you to give you an opportunity to register for them. If you have any questions about theses hacks, their suitabilty for you, or this application process, please email vipazure@microsoft.com

* Required

1. Company name *

Enter your answer

2. Primary contact with company email address (We will use this to contact you directly with a registration link, as and when a scheduled hack is announced) *

Enter your answer

Workshops

https://aka.ms/AzureEventList

Azure Serverless & Microservices Workshop – London October 10th 2017

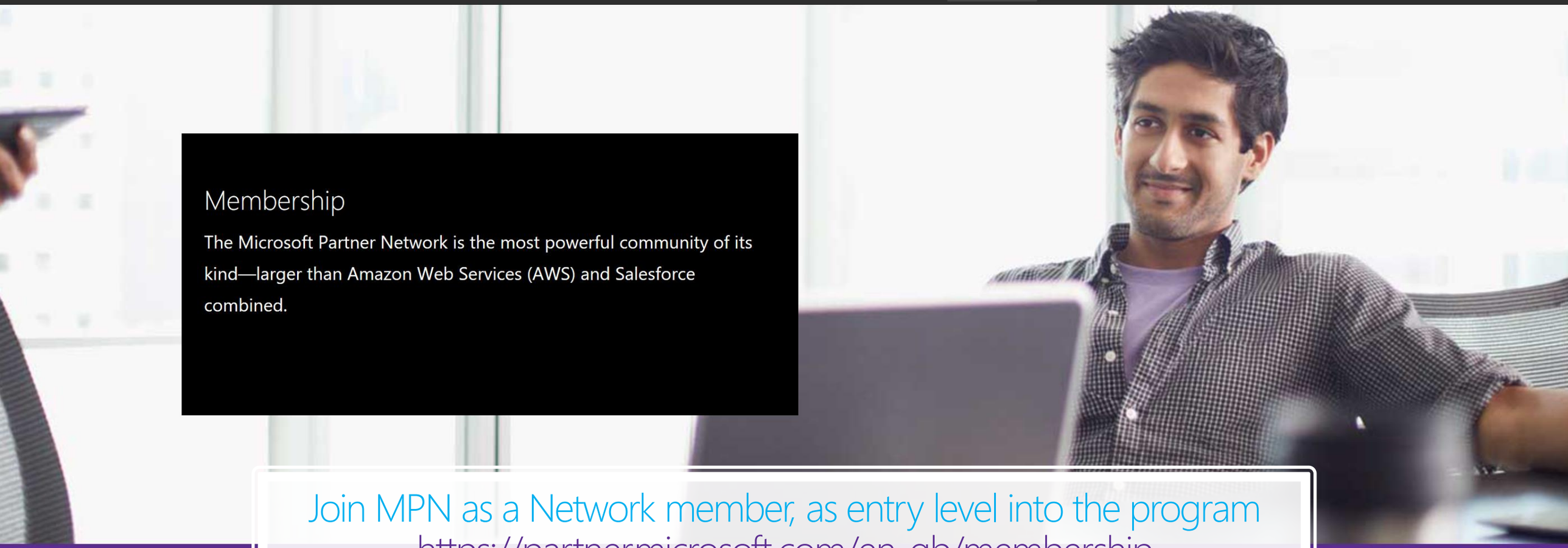
What if you could spend all your time building and deploying great apps, and none of your time managing servers? Server less computing lets you do just that, because the infrastructure you need to run and scale your apps is managed for you. Alternatively, are you using a monolith architecture at work and keen to learn how you can decompose it into discreet microservices? Our technical experts will walk through Mi-crosoft's offering with talks and demos highlighting what to look for and what to avoid.

Register for London October 10th

GDPR, Security and Privacy features for cloud applications – London, October 17th 2017

Join us for a day in which we explore Microsoft Azure and the services and features related to security, privacy, governance and GDPR. In this workshop we will explore how to secure your investments, infrastructure, data and applications whilst covering topics such as security patterns, access control, identity, networking, hybrid configurations, databases and threat detection.

Register for London October 17th



Membership

The Microsoft Partner Network is the most powerful community of its kind—larger than Amazon Web Services (AWS) and Salesforce combined.

Join MPN as a Network member, as entry level into the program
https://partner.microsoft.com/en-gb/membership

You want to grow your business. We know how.

Partnering with Microsoft pays off.

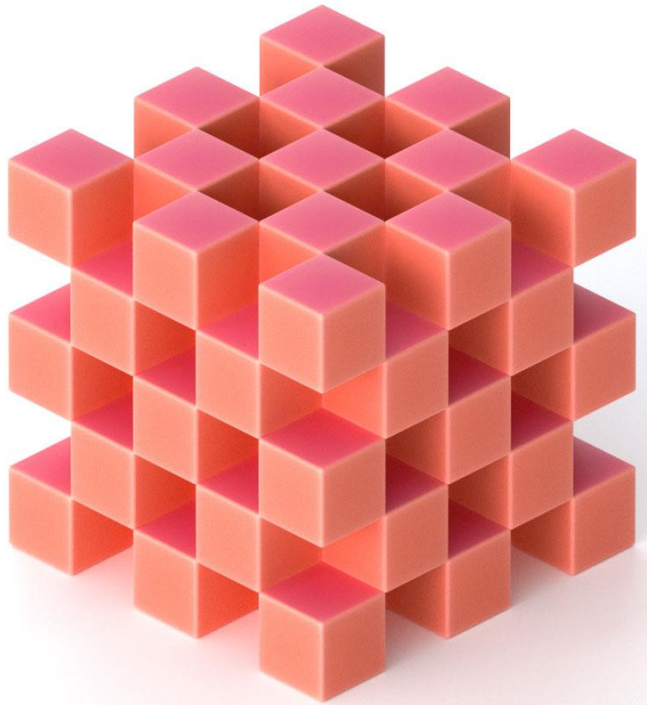
Partner sign in >

Introduction to Microservices

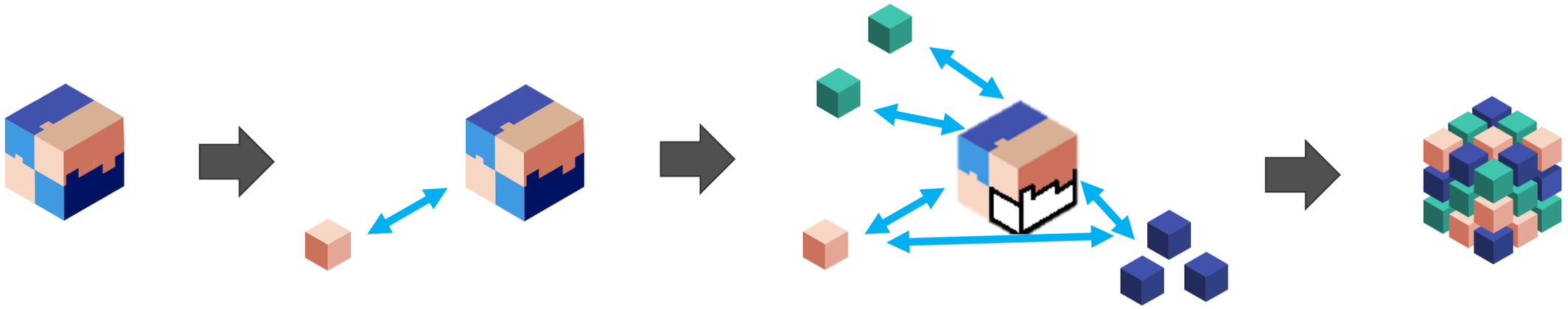
MICROSERVICES IS AN ARCHITECTURAL DESIGN PATTERN

Loosely coupled collection of **small**, **autonomous** services.

Each service is **self-contained** and should implement a **single** business capability.



Evolution to Microservices



Monolith

Client/Server

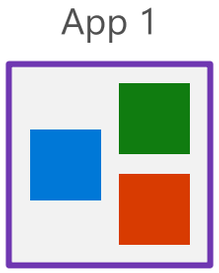
3-tier

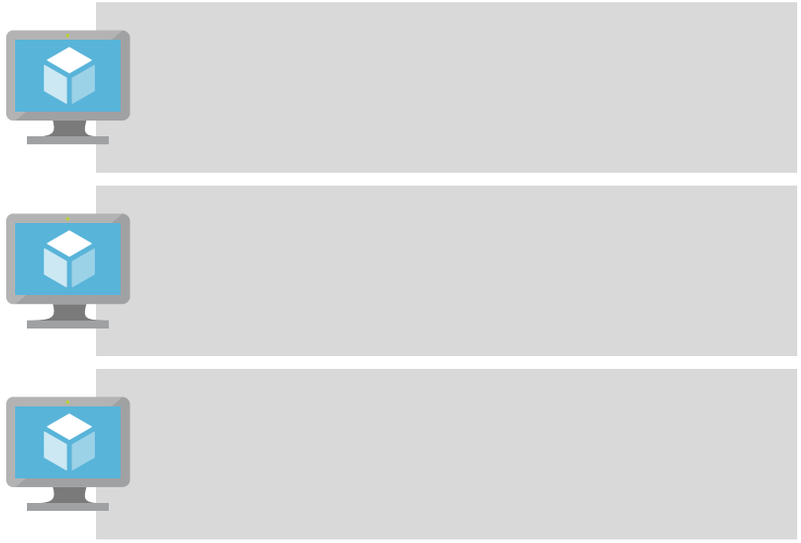
Microservices

Architecture and Deployment

Traditional Application

- Has its functionality within a few processes that are componentized with layers and libraries.
- Scales by deploying the whole app on multiple servers or VMs

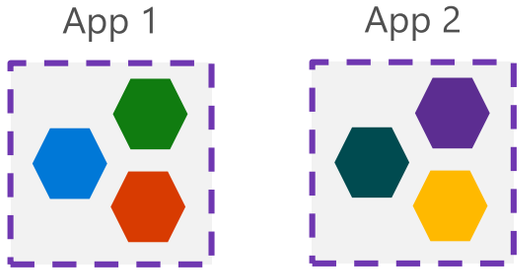


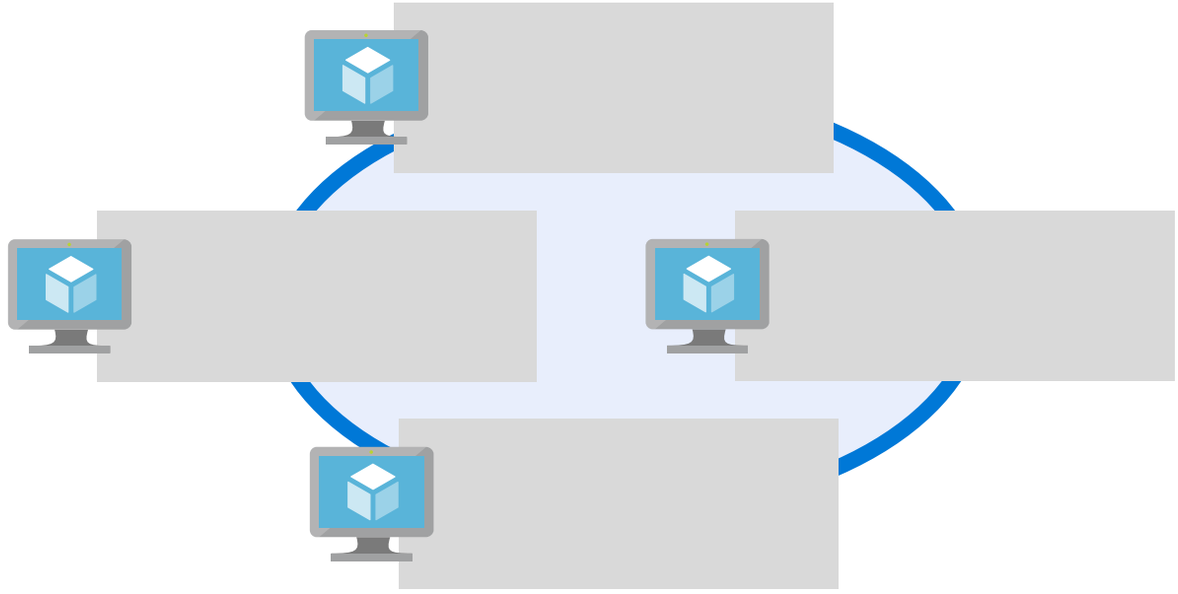


- **Course grained scaling**
- **Deploy entire app stack each time**
- **Difficult resource optimization**

Microservices Application

- Application functionality segregated into separate smaller services.
- Scaled by deploying services independently with multiple instances across VM clusters

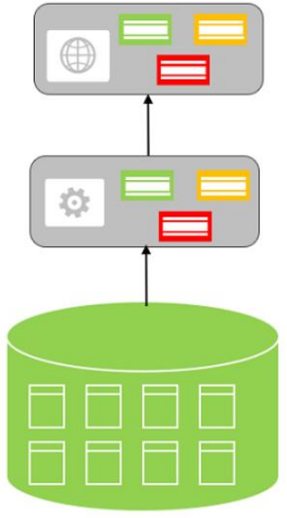




- **Fine grained scaling**
- **Deploy individual services as needed**

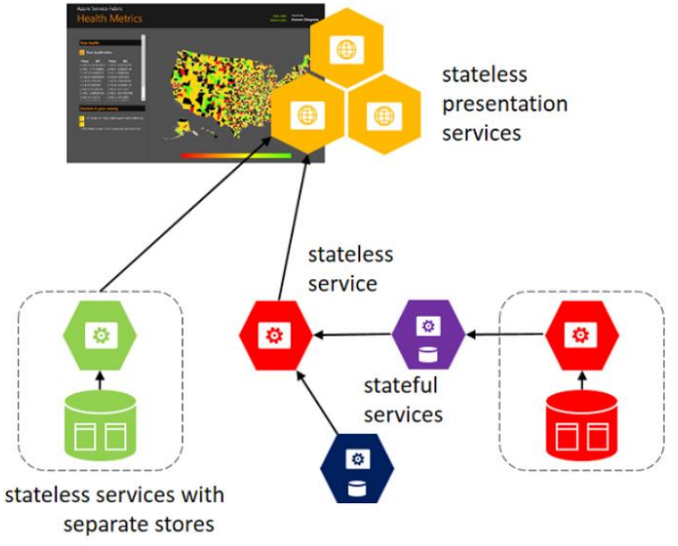
Microservices Challenges

State in Monolithic approach



The monolithic approach has a single database and tiers of specific technologies.

State in Microservices approach

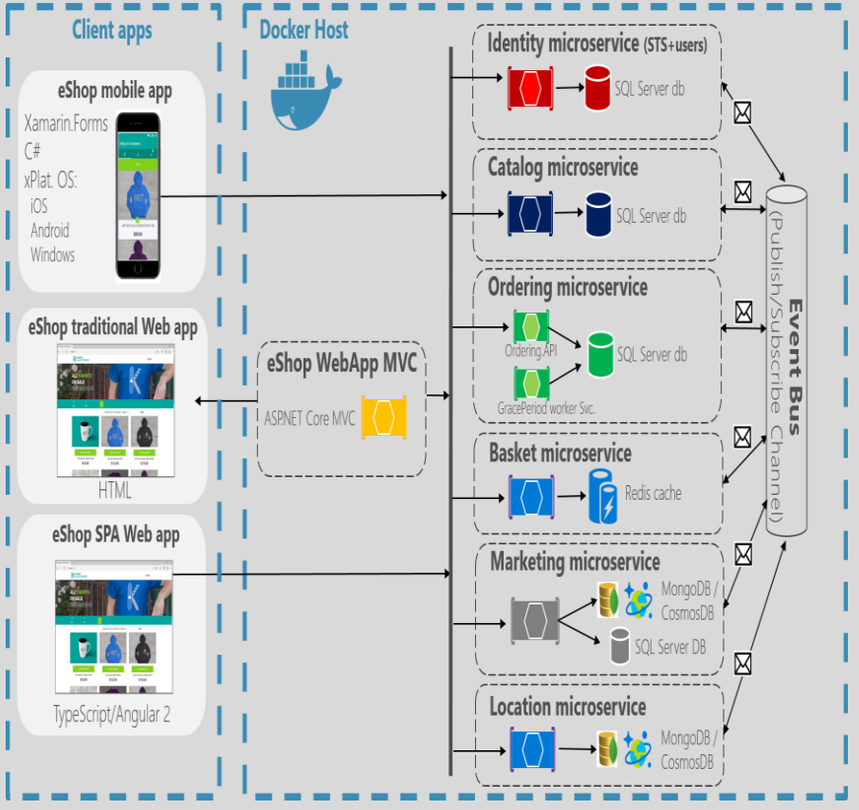


The microservices approach has a graph of interconnected microservices where state is typically scoped to the microservice and various technologies

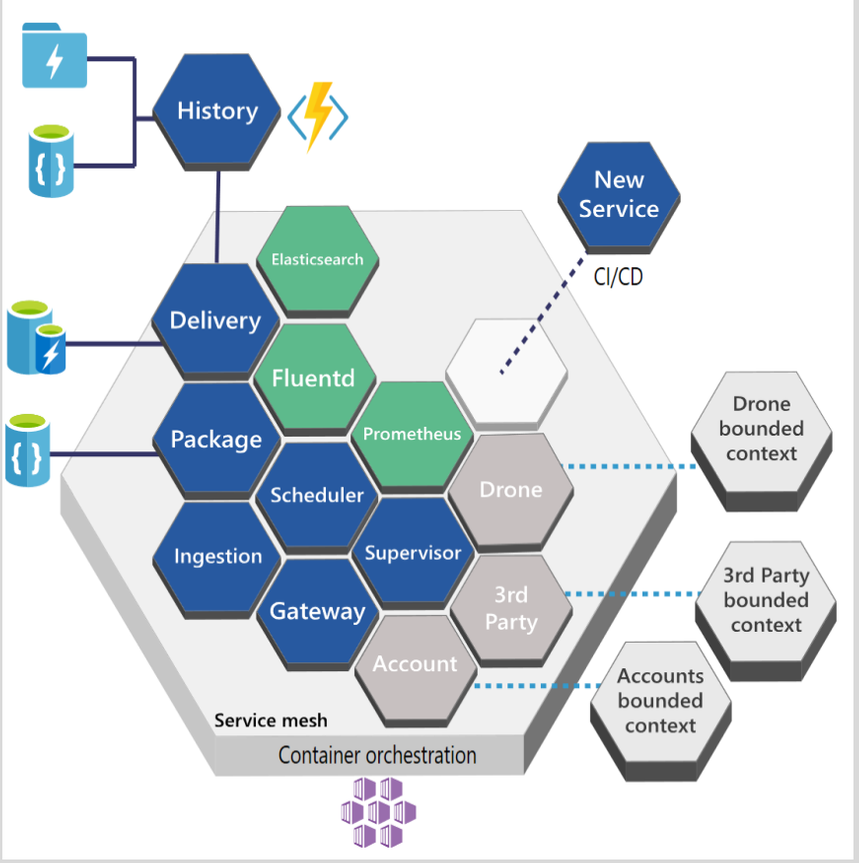
- #1: How to define the boundaries of each microservice
  - "user" could be in CRM, a customer, logged on account, etc
- #2: How to create queries that retrieve data from several microservices
  - API Gateway, CQRS with query/reads tables, big data repository
- #3: How to achieve consistency across multiple microservices
  - CAP theorem
- #4: How to design communication across microservice boundaries
  - Blocking, chaining, coupling, etc

Reference Architectures and Guides

aka.ms/MicroservicesArchitecture



docs.microsoft.com/azure/architecture/microservices/

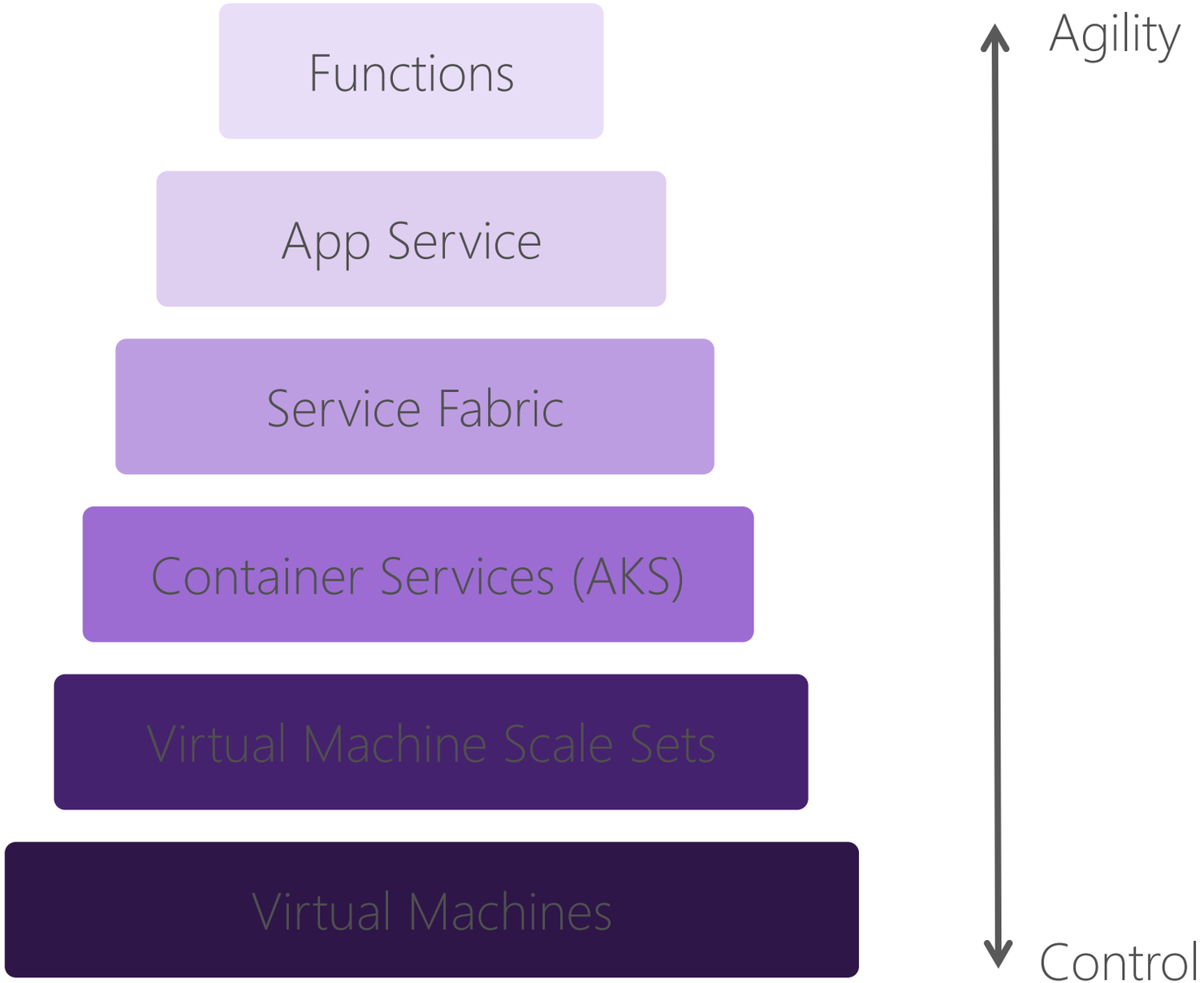


Free eBook

aka.ms/microservicesebook

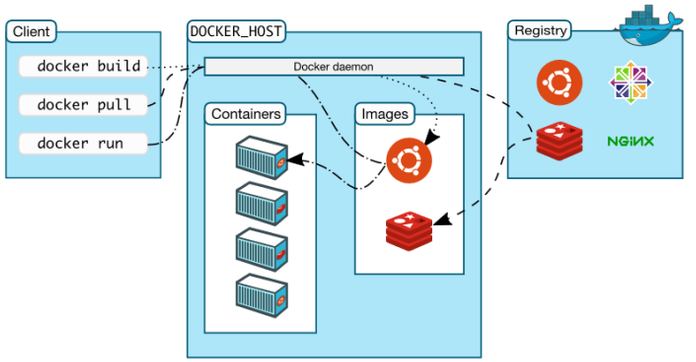
Microservices on Microsoft Azure

Code


Functions
App Service
Service Fabric
Container Services (AKS)
Virtual Machine Scale Sets
Virtual Machines
Agility
Control

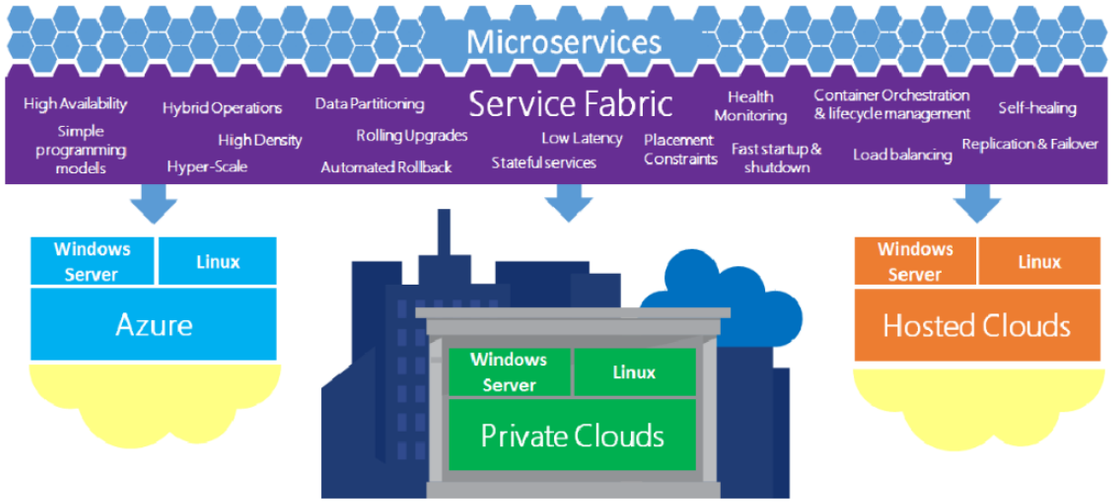
Azure Container Services (AKS)

- Fully managed Kubernetes container orchestration service
  - or choose other orchestrators
- **Azure Container Instances**
  - Isolated containers for simple applications, task automation, and build jobs
- **Azure Container Registry**
  - storing and managing container images for Azure deployments in a central registry



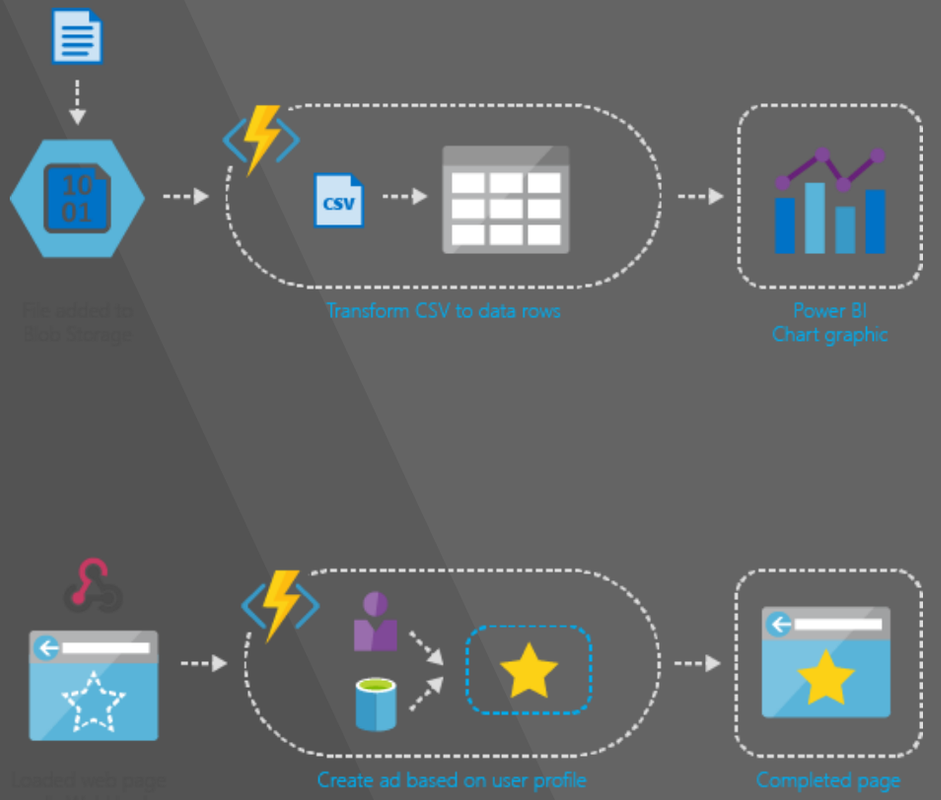
Azure Service Fabric

- Powers many Microsoft services
  - SQL Database, Azure DB, Cortana, Power BI, Intune, IoT Hub,
- Runs Windows and Linux, on-premise and cloud and developer PCs
- Scale from a few to thousands of servers
- Stateless and stateful programming models
- Comprehensive runtime and lifecycle management capabilities
- Container deployment and orchestration



Azure Functions

- Serverless computing
  - Built on App Service WebJobs
  - Serverless consumption plan
- Build “nano-services” as individual functions
- Event driven model
  - Timer, http hook, queues, storage, etc
  - Easy way to implement load levelling
  - No upper limit on compute
- Easy to test, tune and deploy



Data

"human data"

Transactional integrity, operational information, etc.

"machine data"

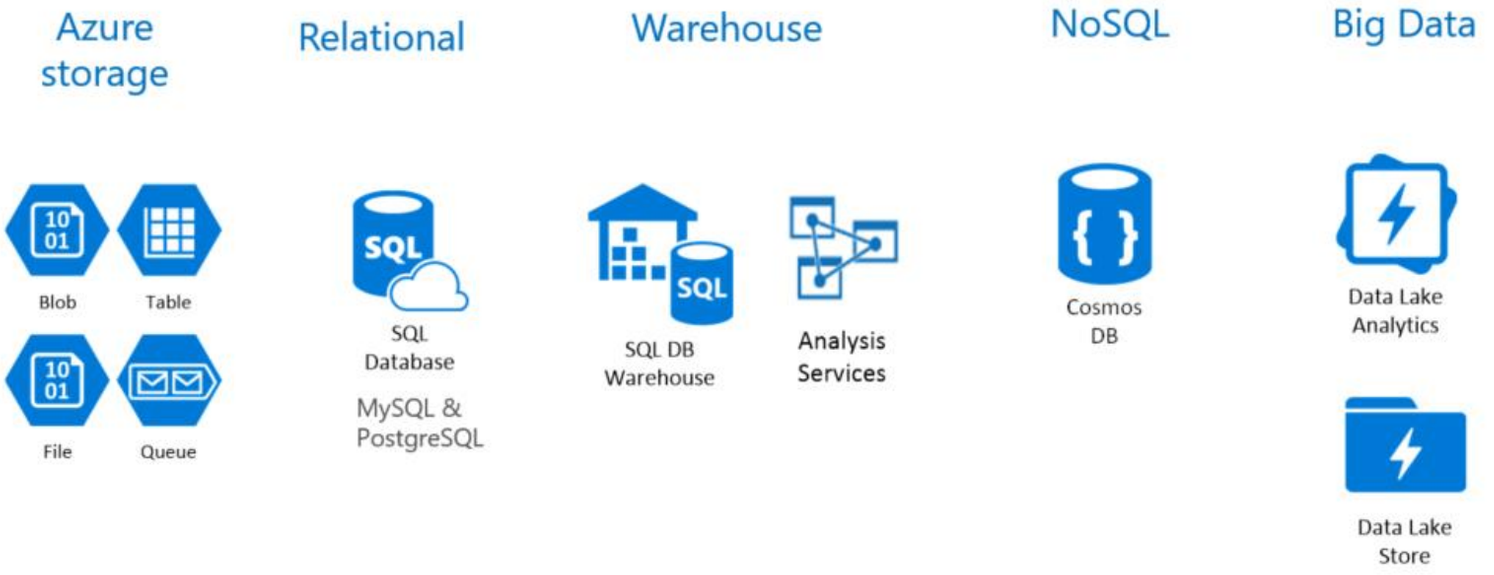
Independent, telemetry, insights, etc.

Polyglot Persistence

Different databases are designed to solve different problems. Using a single database engine for all of the requirements usually leads to non-optimal solutions

e.g.:

- *User session*
- *Catalogue data*
- *Product search*
- *Shopping cart*
- *Orders database*
- *Analytics*
- *Reporting, ….*



Events and Messages

Messages and Events

Messages

- Typically carry information needed for a step in a defined workflow
- May express inherent monetary value or commands to performs actions
  - Consider Azure Service Bus or Azure Queues

Events

- Don't generally convey publisher intent, other than to inform

1. "Business logic activity" carried out by publishing application
   - Something has happened in system X that may be of interest elsewhere
   - Consider Azure Event Grid or Logic Apps
2. Informational data points from continuously published stream: IoT, etc
   - Logic often related to changes in pattern (such as sensor temperature rising) rather than individual data points
   - "Complex Event Processing" model
   - Consider Azure Event Hubs / IoT Hubs

azure.microsoft.com/blog/events-data-points-and-messages-choosing-the-right-azure-messaging-service-for-your-data