

Azure Serverless & Microservices Briefing

Ross.P.Smith@microsoft.com

Gabriel.Nepomuceno@microsoft.com

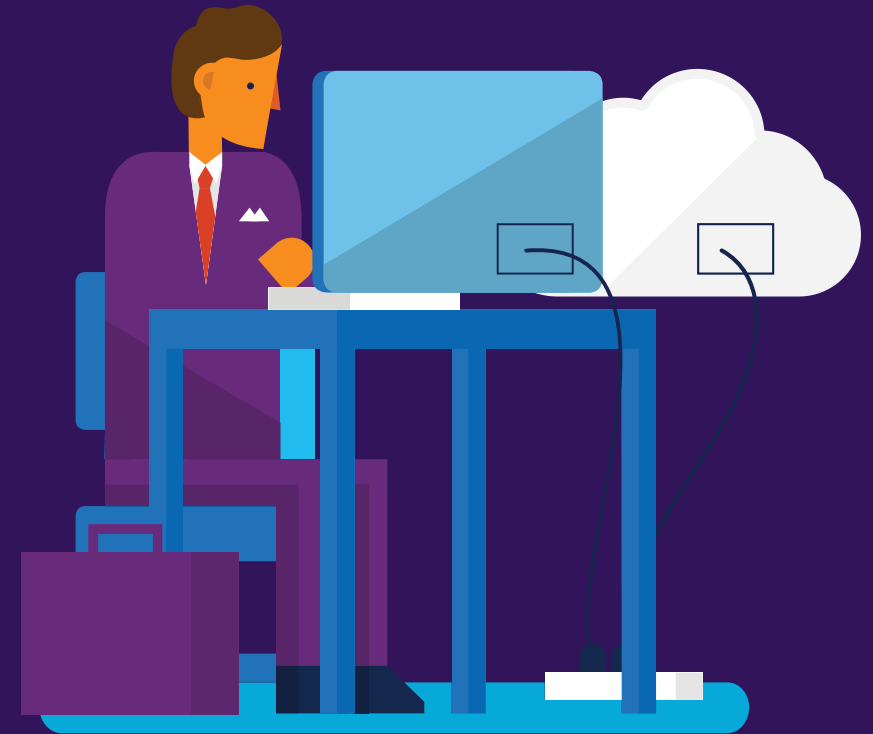
David.Gristwood@microsoft.com

Ben.Coleman@microsoft.com



<http://aka.ms/azureevent>

#Azure
#BuildWithAzure



vipazure@microsoft.com

Agenda

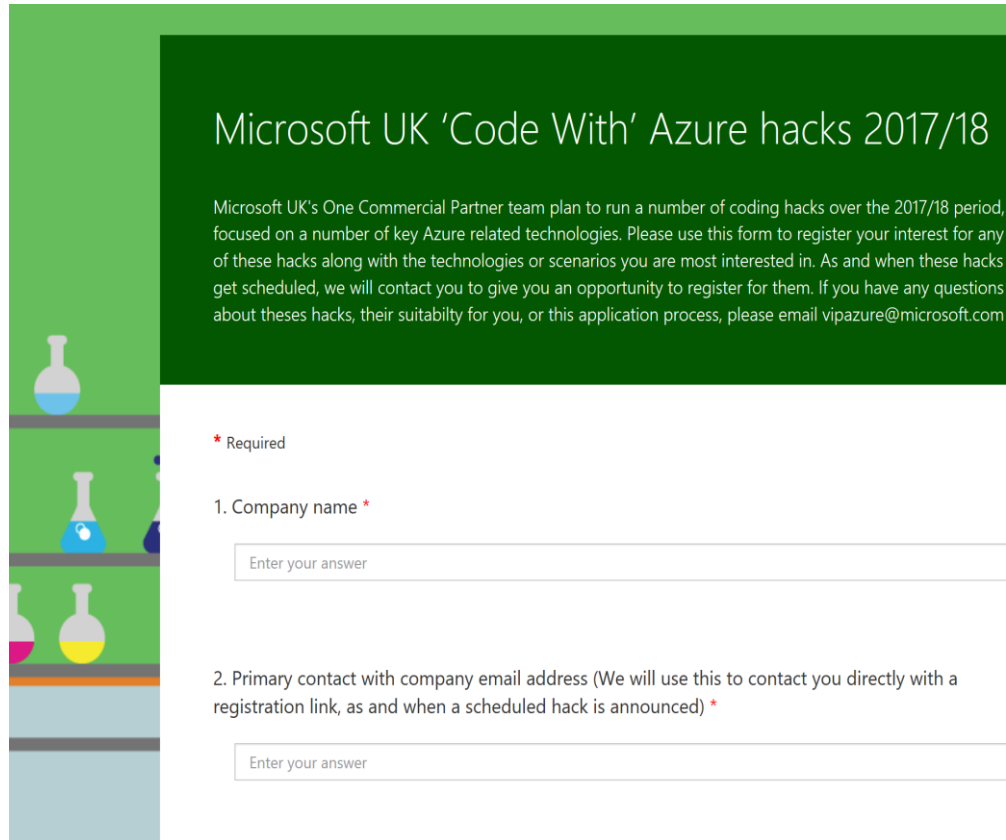
- Microservices architectures
- Serverless in the modern workplace
- Azure Functions
- *Lunch around 1pm*
- Build always-on, scalable, distributed apps
- Actor based computing
- *Finish around 4pm*



Other technical activities from our team

Code with hacks

<https://aka.ms/ukocpishack0817>



The image shows a registration form for Microsoft UK's 'Code With' Azure hacks 2017/18. The form is set against a green background with a vertical sidebar on the left containing illustrations of laboratory flasks. The main content area has a dark green header with the event title. Below the header, there is a paragraph of text explaining the event. The form consists of two numbered sections, each with a text input field and a label indicating required information.

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 these hacks, their suitability 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 Workshop - Serverless & Microservices Workshop

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 Microsoft's offering with talks and demos highlighting what to look for and what to avoid.

[Register for London October 10th](#)

Azure Workshop – GDPR, Security and Privacy features for cloud applications

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.

Microservices

Microservices

Architectural style that structures an application as a collection of loosely coupled services. In a microservices architecture, services should be fine-grained and the protocols should be lightweight.

The benefit of decomposing an application into different smaller services is that it improves modularity and makes the application easier to understand, develop and test.

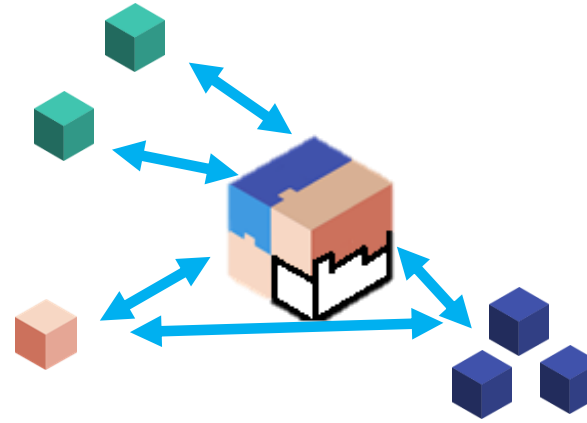
It also parallelizes development by enabling small autonomous teams to develop, deploy and scale their respective services independently.

It also allows the architecture of an individual service to emerge through continuous refactoring.

Microservices-based architectures enable continuous delivery and deployment.



Evolution to Microservices



Monolith

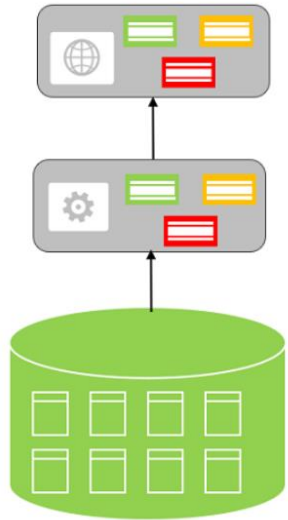
Client/Server

3-tier

Microservices

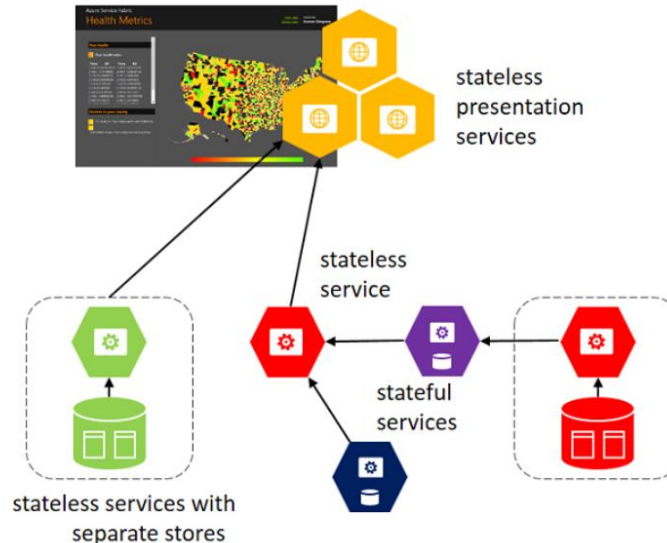
Microservices

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

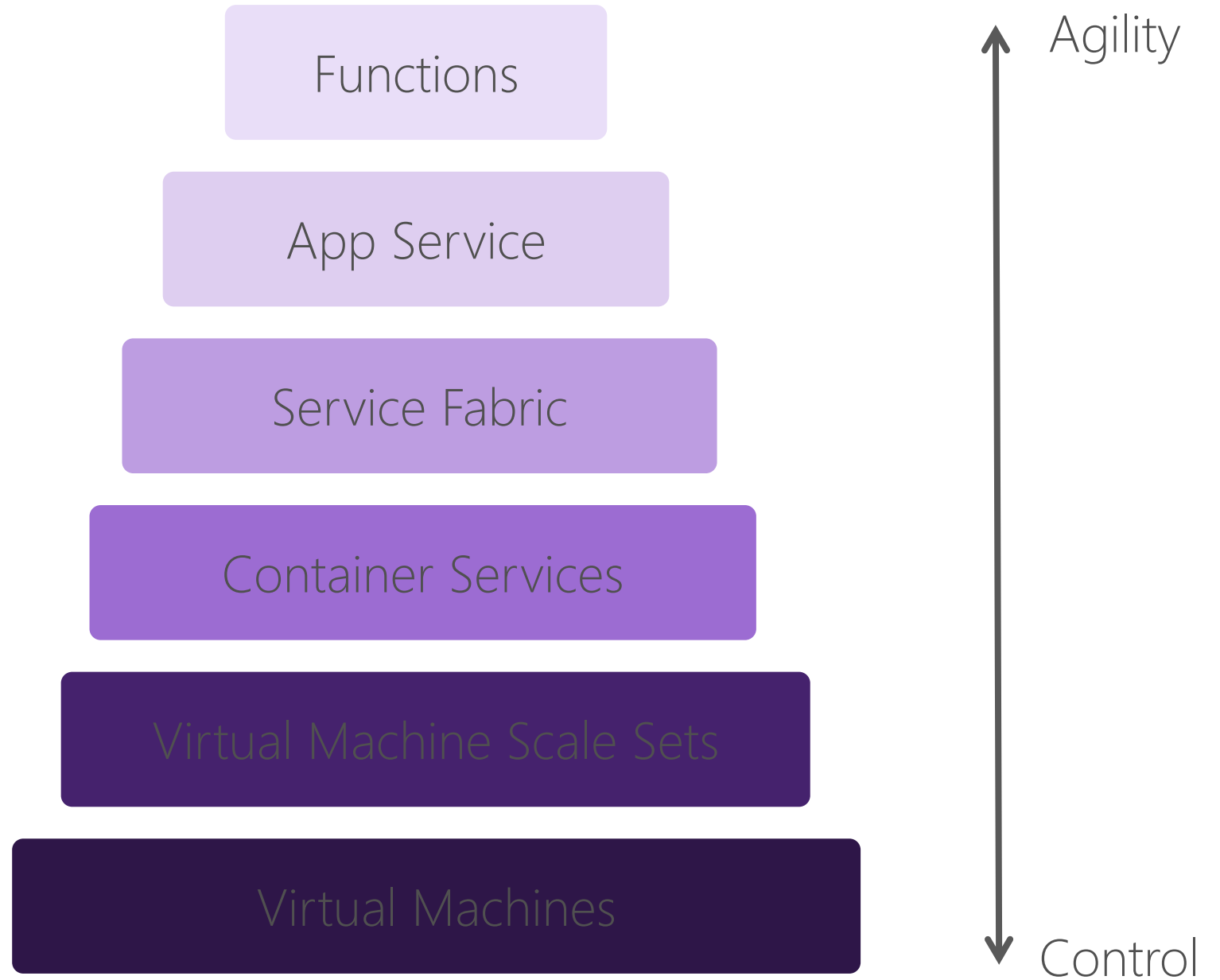
Challenges

- #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

aka.ms/MicroservicesEbook

Microservices & Serverless on Microsoft Azure

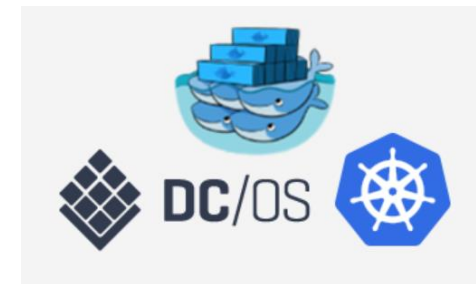
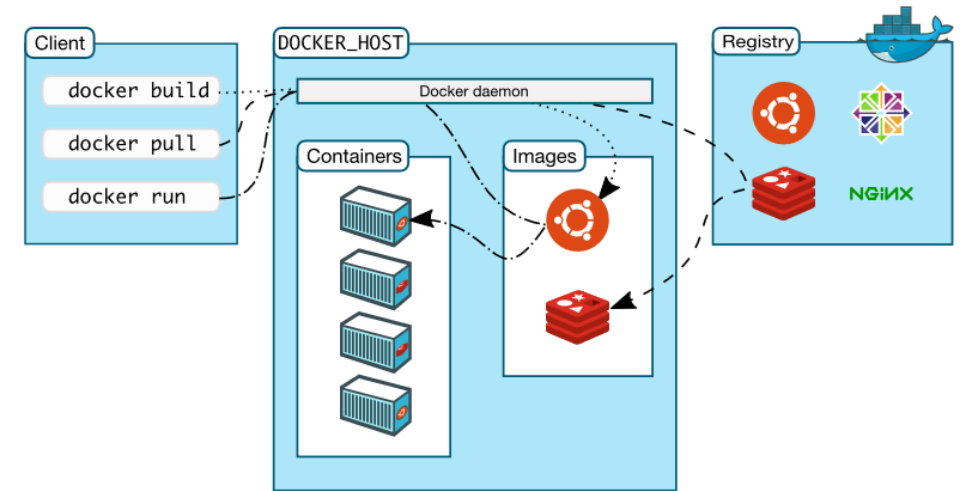
Code

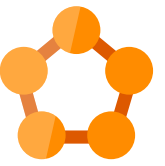


Azure Container Services



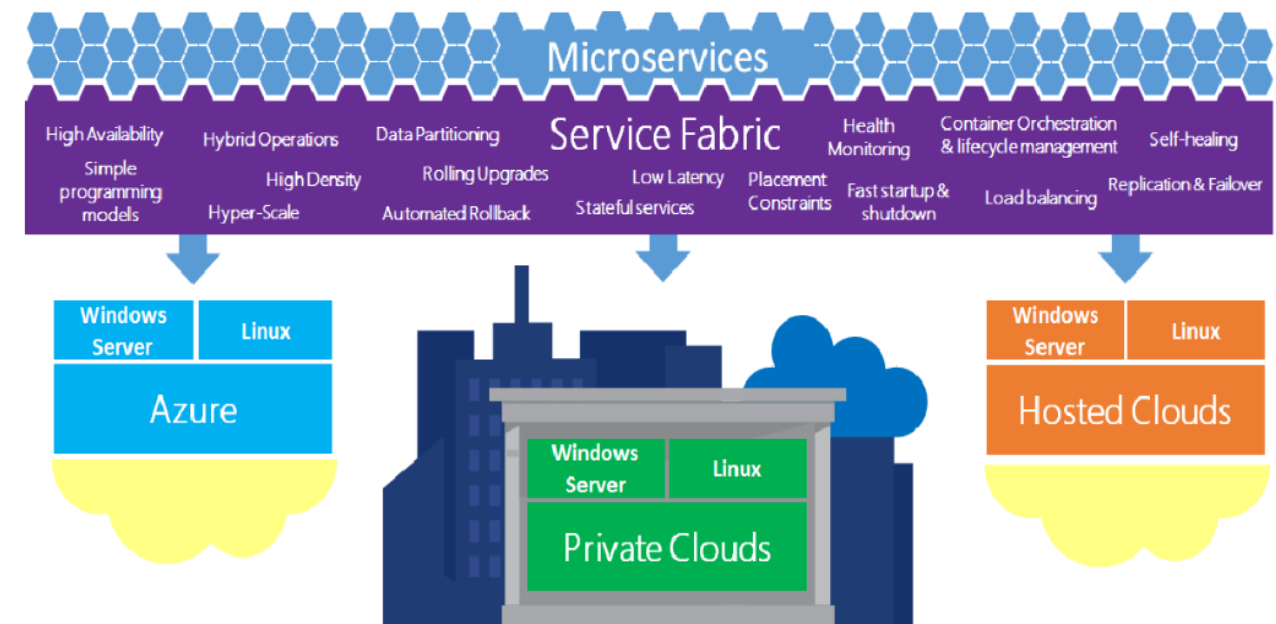
- Docker Container platform
 - Linux, Windows Server 2016
- Running applications in lightweight, isolated units
 - File system, registry, processes, etc
 - Apps unaware share the host OS
- Scale and orchestrate using DC/OS, Docker Swarm, or Kubernetes
- Azure Container Instances (preview)
 - Isolated containers for simple applications, task automation, and build jobs
- Azure Container 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 App Services

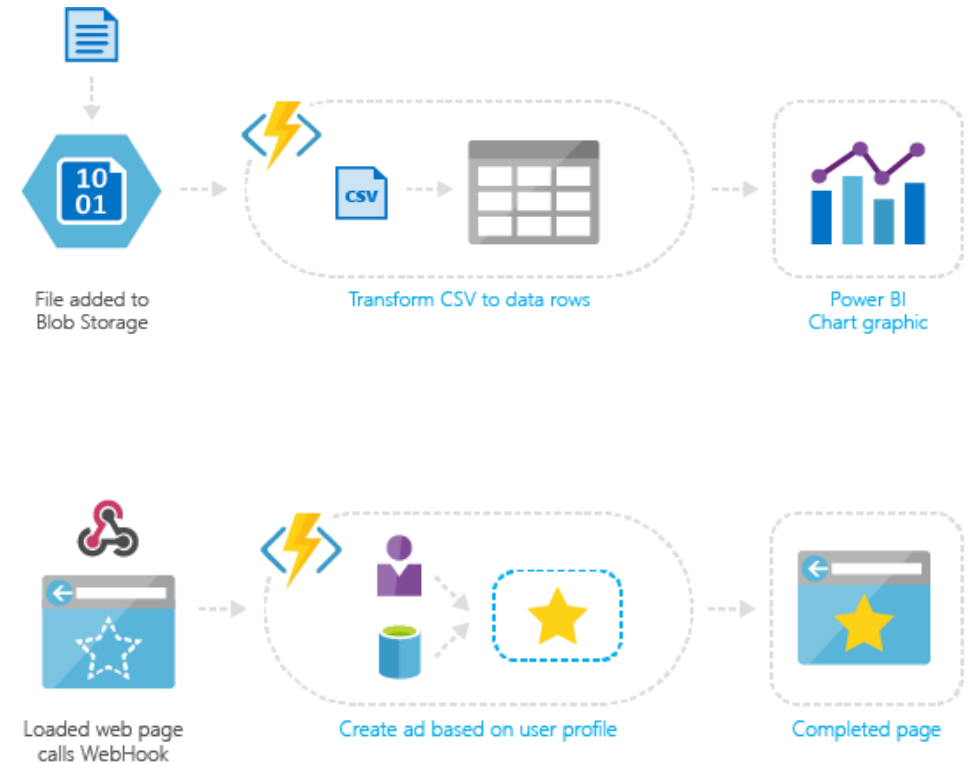
- Managed compute platform optimized for hosting websites and web applications
 - Multiple languages & frameworks
 - DevOps optimization for CI/CD
 - Global scale with high availability
 - Optional fully isolated and dedicated environment
- App Service Plans represent physical resources available to your apps
 - Instance count, instance size, tiering (Free, Shared, Basic, Standard*, Premium*, Isolated*)
- “Best choice for most web apps”
- Native isolation for Windows
- Web App for Containers for Linux

166.35 GBP/MONTH (ESTIMATED)		332.71 GBP/MONTH (ESTIMATED)		665.41 GBP/MONTH (ESTIMATED)	
S1 Standard		S2 Standard		S3 Standard	
1 Core		2 Core		4 Core	
1.75 GB RAM		3.5 GB RAM		7 GB RAM	
50 GB Storage		50 GB Storage		50 GB Storage	
Custom domains / SSL SNI Incl & IP SSL Support		Custom domains / SSL SNI Incl & IP SSL Support		Custom domains / SSL SNI Incl & IP SSL Support	
Up to 10 instances Auto scale		Up to 10 instances Auto scale		Up to 10 instances Auto scale	
Daily Backup		Daily Backup		Daily Backup	
5 slots Web app staging		5 slots Web app staging		5 slots Web app staging	
Traffic Manager Geo availability		Traffic Manager Geo availability		Traffic Manager Geo availability	
55.45 GBP/MONTH (ESTIMATED)		110.90 GBP/MONTH (ESTIMATED)		221.80 GBP/MONTH (ESTIMATED)	
B1 Basic		B2 Basic		B3 Basic	
1 Core		2 Core		4 Core	
1.75 GB RAM		3.5 GB RAM		7 GB RAM	
10 GB Storage		10 GB Storage		10 GB Storage	
Custom domains		Custom domains		Custom domains	
SSL Support SNI SSL Included		SSL Support SNI SSL Included		SSL Support SNI SSL Included	
Up to 3 instances Manual scale		Up to 3 instances Manual scale		Up to 3 instances Manual scale	
41.59		83.18		166.35	

Azure Functions



- Serverless computing
 - Serverless consumption plan
- Built on App Service WebJobs
- 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"



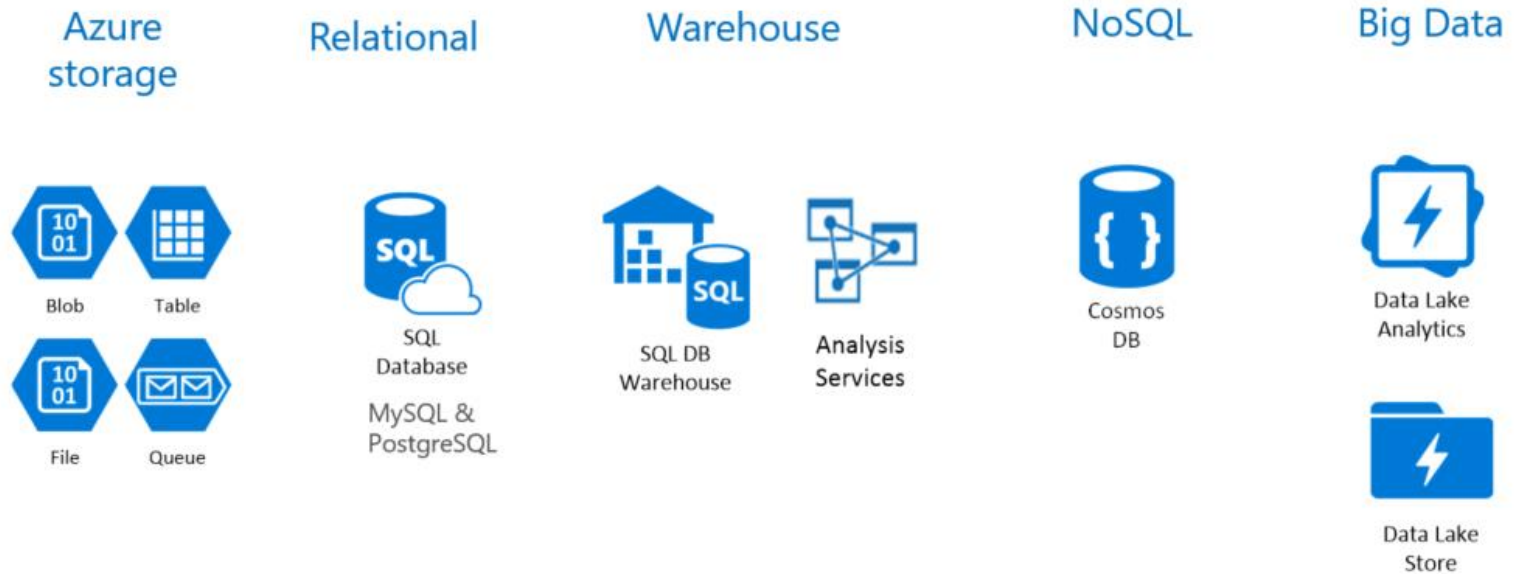
Independent,
telemetry, insights, etc.

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](#)

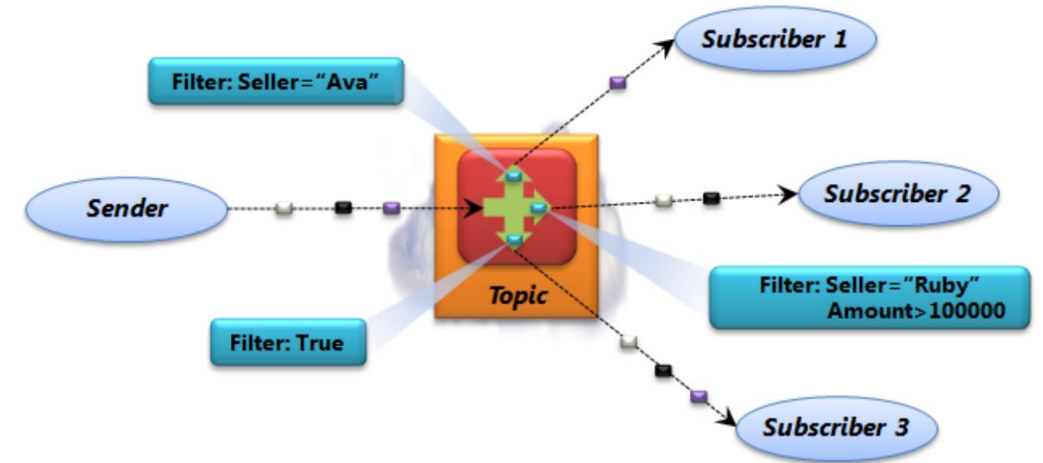
Events

- 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 Service Bus

- Reliable information delivery service
 - Duplicate detection, time-based expiration, batching, etc
 - Separate Send and Listen access
- Brokered messaging support:
 - Queues – single consumer
 - Topics & subscriptions – multiple consumers
- Premium tier addresses common requests around scale, performance and availability
 - Dedicated resources in form of messaging unit

Premium	Standard
High throughput	Variable throughput
Predictable performance	Variable latency
Fixed pricing	Pay as you go variable pricing
Ability to scale workload up and down	N/A
Message size up to 1 MB	Message size up to 256 KB

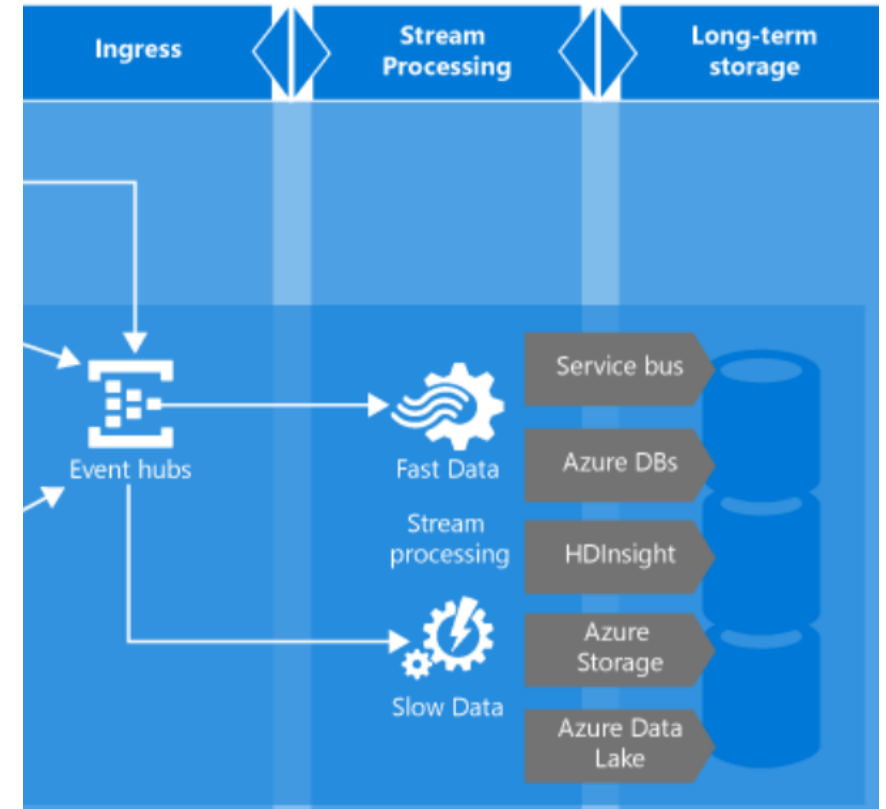


Best Practices for performance improvements:

- Use AMQP and SBMP over HTTP
 - Use Asynchronous operations on queues
 - Client-side batching
 - Use partitioned queues or topics
- docs.microsoft.com/azure/service-bus-messaging/service-bus-performance-improvements

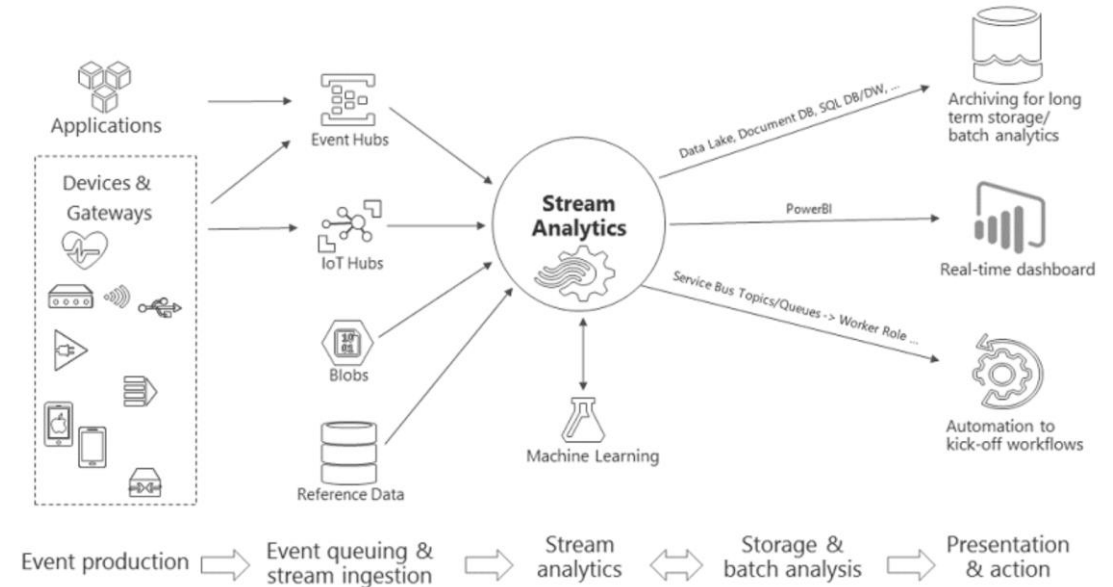
Azure Event Hubs

- “Event Ingestor”
 - Accepts and stores event data
 - Makes data available for fast “pull” retrieval
- Log millions of events per second in near real time
 - Low latency, elastic scale
 - Configurable time retention, read data using publish-subscribe semantics
 - Partition is used to store ordered sequence of events
 - Multiple subscribers via consumer group
- Scale through Throughput Units (TUs)
 - Single TU entitles you to 1MB/second or 1000 events/second ingress and 2MB/second or 2000 events/second egress
 - Auto-scale up

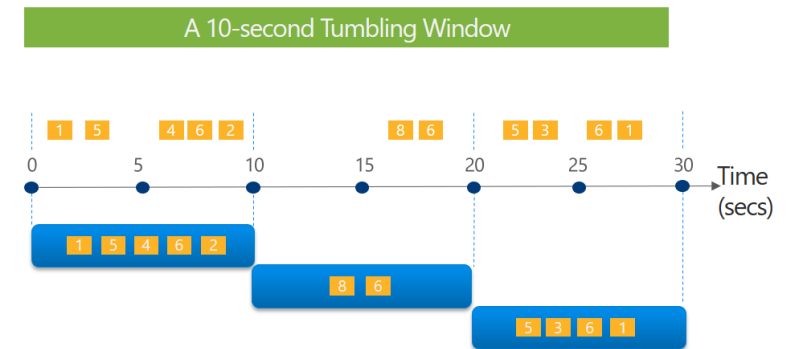


Stream Analytics

- Massively parallel Complex Event Processing pipeline
 - Integration into source and destination
 - Support for partition in event hubs, blob, etc
- Declarative SQL like Stream Analytics query language
- Streaming units (SUs) represent resources & computing to execute an Stream Analytics job
 - Blended measure of CPU, memory and read/write rates
 - Corresponds to around 1 MB/sec throughput



Tell me the count of tweets per time zone every 10 seconds



```
SELECT TimeZone, COUNT(*) AS Count
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY TimeZone, TumblingWindow(second,10)
```

Azure Event Grid (Preview)

- Route events from any source to any destination
 - Scale dynamically
 - Near-real-time event delivery
 - Uses publish/subscribe model
- Scenarios:
 - Serverless architecture
 - Application integration
 - Ops automation

