

# Microservices

An introduction.



## Agenda

1. The Monolith
2. Microservices
3. Microservices and Pivotal Cloud Foundry

## Monolith

A three tiered monolith.



## MONOLITH CHALLENGES



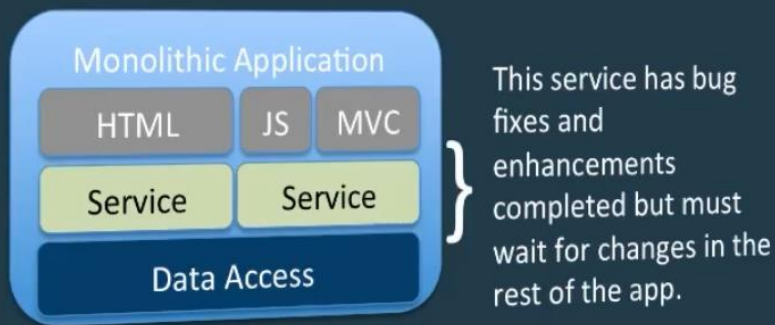
## Monolith Design Patterns

Traditional monolithic design patterns are not appropriate for the cloud.

Monoliths generally pass session state around, they also tend to write to the file system which is also not applicable in the cloud.

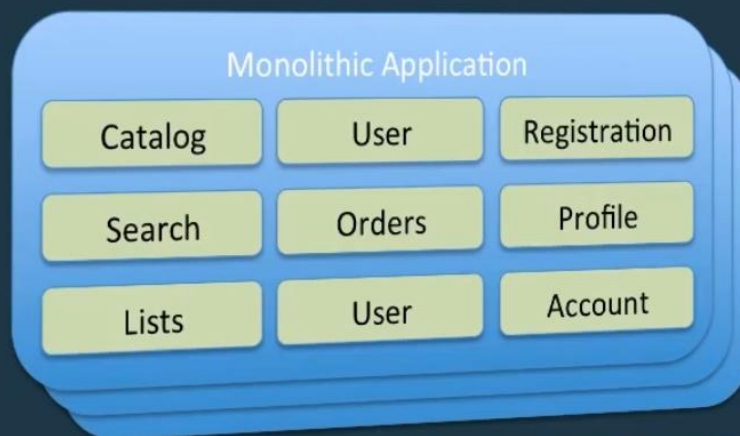
## Monolith Change Cycle

Monoliths couple change cycles together.



## Monolith Scaling

Monoliths services can't be scaled independently.



## Monolith Coordination

Too many developers in one code base.

## Monolith Knowledge

Developers struggle to understand a large codebase.

## Monolith Commitment

Long term commitment to the tech stack.

## Agenda

1. The Monolith
2. Microservices
3. Microservices and Pivotal Cloud Foundry

## The Importance of APIs

All teams will henceforth expose their data and functionality through service interfaces.

Jeff Bezos  
Amazon - 2002



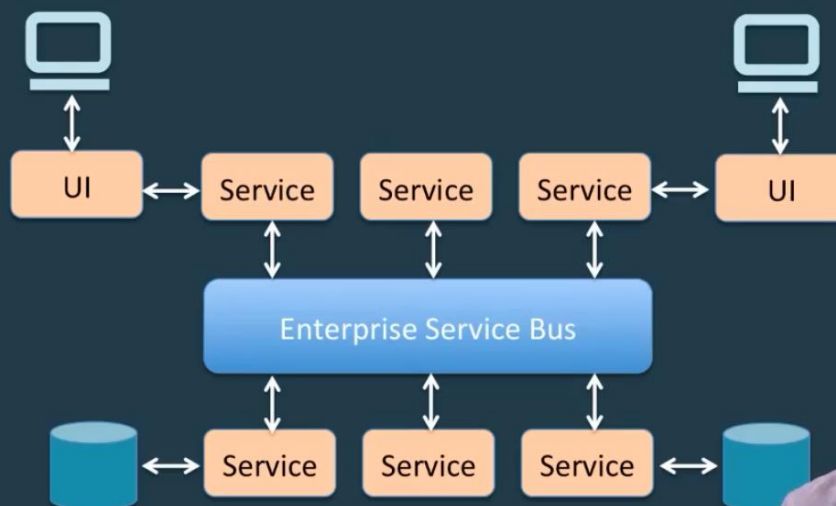
## Microservices Defined

Microservices are a loosely coupled Service-Oriented Architecture (SOA) with bounded contexts.

Adrian Cockcroft



## Traditional ESB / SOA



## Orchestration vs Choreography



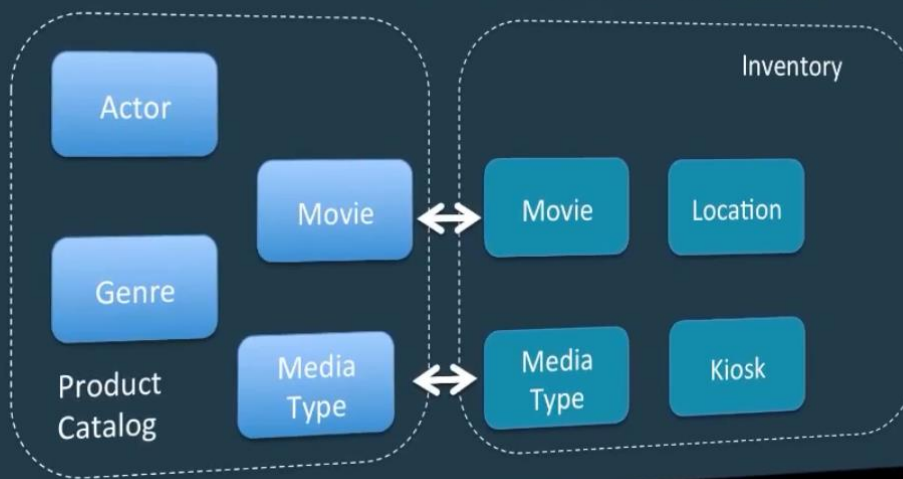
## Unix Pipes and Filters

A microservice has a single responsibility.

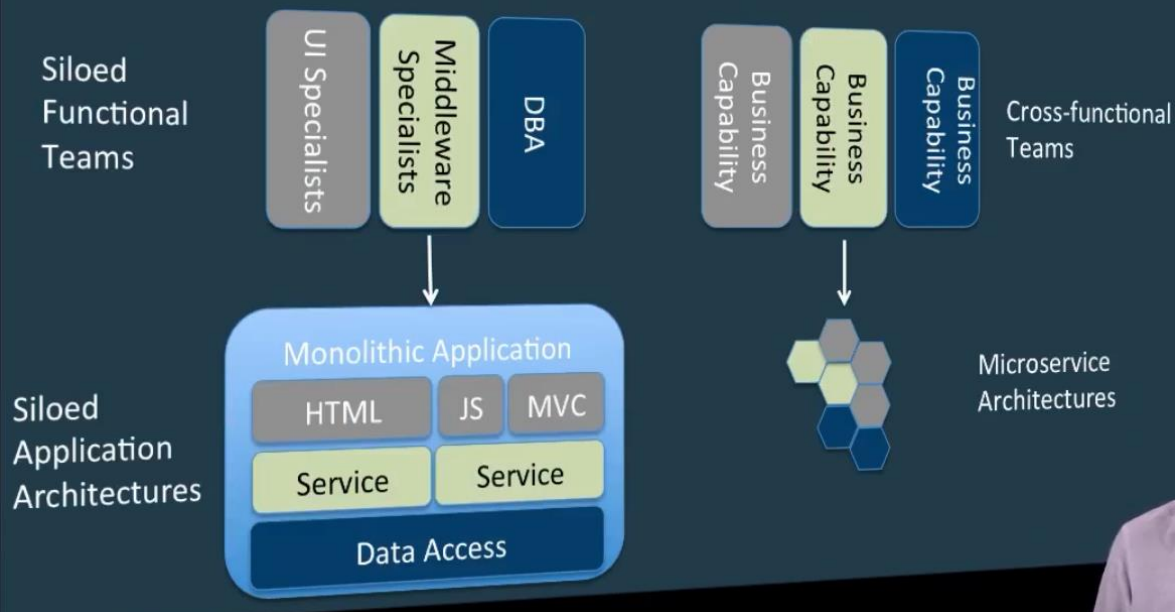
```
cut -d" " -f1 < access.log | sort | uniq -c | sort -rn | less
```

## Bounded Context

Microservices use a bounded context.

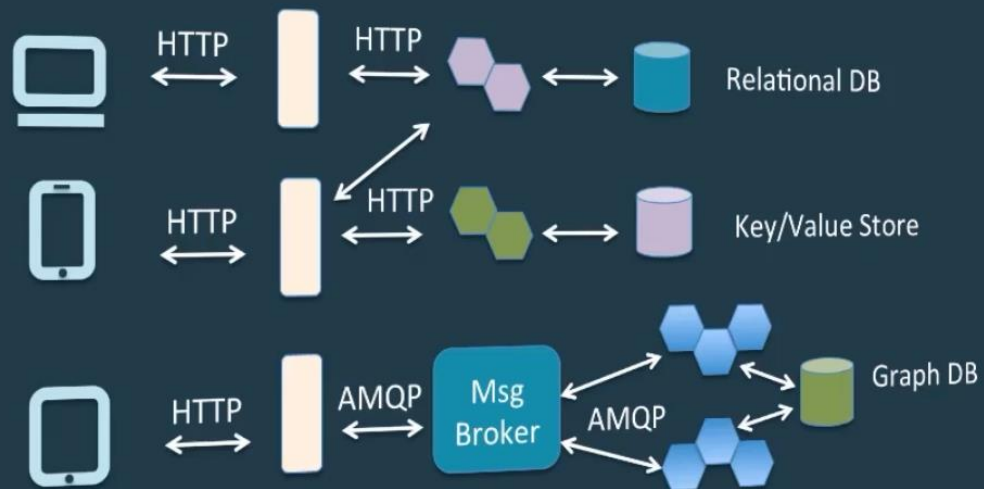


## Organize Around Business Capabilities



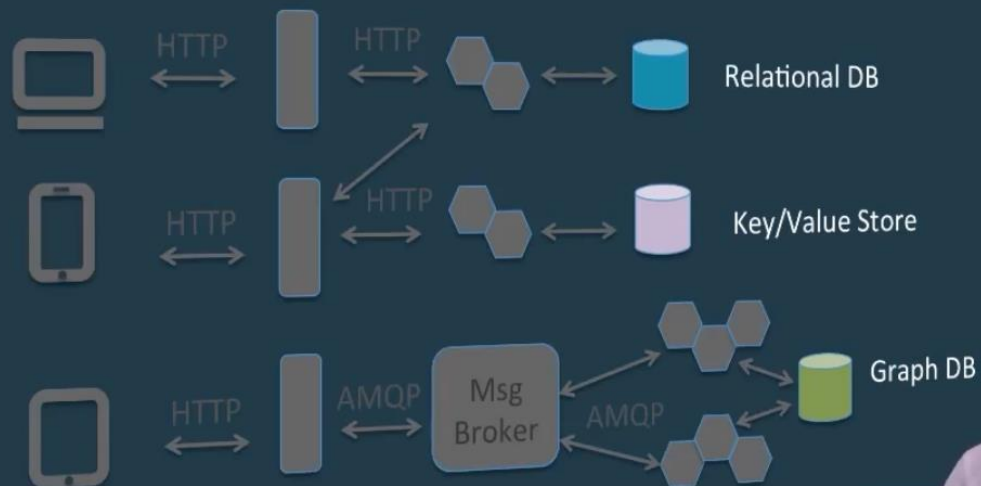


## Microservice Architecture (Simplified)



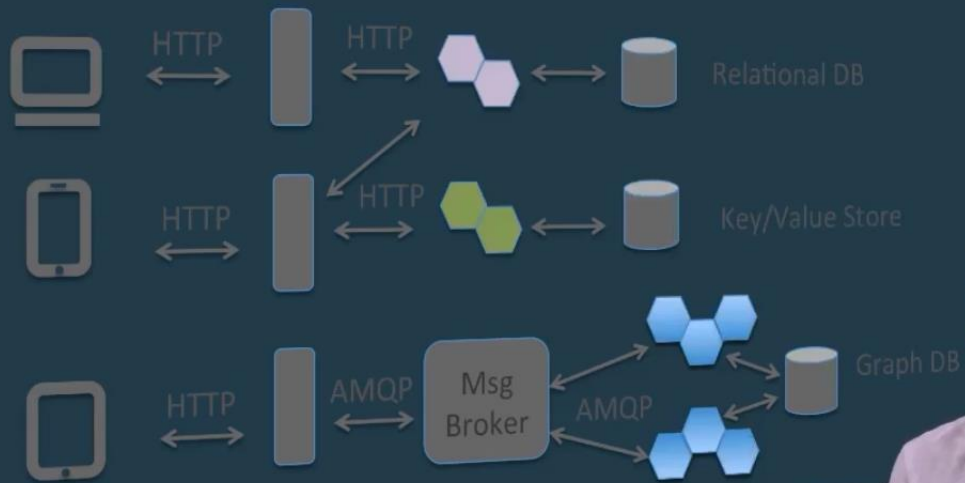
## Polyglot Persistence

Freedom to pick the persistence solution.



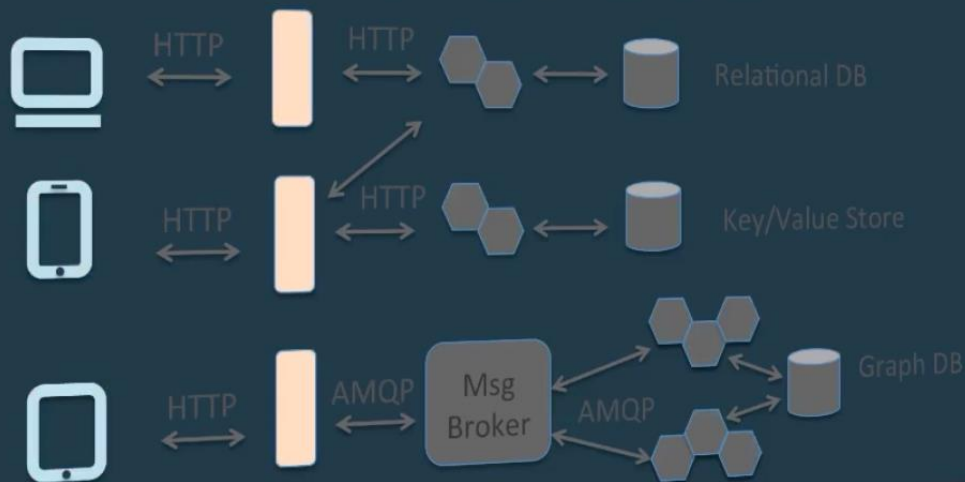
## Polyglot Apps

### Choice of language when developing apps.



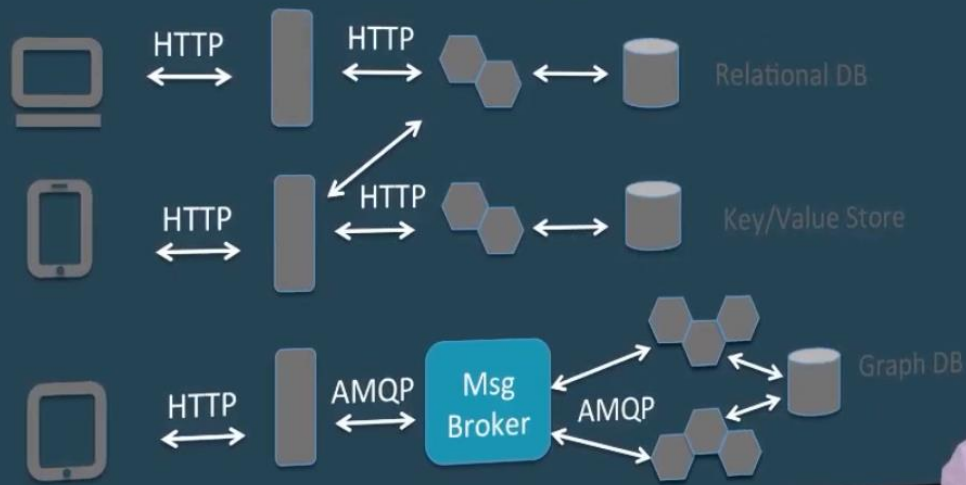
## API Gateway

### Device specific gateways.



## Cloud Protocols

Use cloud friendly protocols.



## MICROSERVICE BENEFITS



### Microservice Change Cycle

Change cycles are decoupled. Enabling frequent deploys.

### Microservice Scale

Allow for efficient scaling.





## Microservice Knowledge

Developers learn a smaller codebase faster.

## Microservice Coordination

Fewer developers in each code base.

## Agenda

1. The Monolith
2. Microservices
3. Microservices and Pivotal Cloud Foundry

## Microservice Challenges

It's been said that microservices have some challenges too!

<http://martinfowler.com/bliki/MicroservicePrerequisites.html>

<http://highscalability.com/blog/2014/4/8/microservices-not-a-free-lunch.html>

Significant Operations Overhead

Microservices have significant operations overhead.

Agreed, but this is mitigated with PCF.

Significant Operations Overhead

Consider: Dynamic Routing

Scaling

Monitoring

Services

Health Mgmt

Buildpacks



PCF offers Buildpacks to enable you bundle your app, provision them and deploy them quickly, PCF also gives you logs to enable monitoring using the **\$ cf logs** or **\$ cf events** commands. PCF also provides the **\$ cf scale** command and has very simple dynamic routing as soon as you bring an application up and running.

Substantial DevOps Skills Required

Substantial DevOps skills are required to run microservices.

Agreed. This is a good thing.



Substantial DevOps Skills Required

Consider: Polyglot Persistence via  
Service Brokers

Space Parity & Immutable  
Infrastructure

Buildpacks

Health Mgmt



## Microservices

Recap

