

# Pivotal

A NEW PLATFORM FOR A NEW ERA

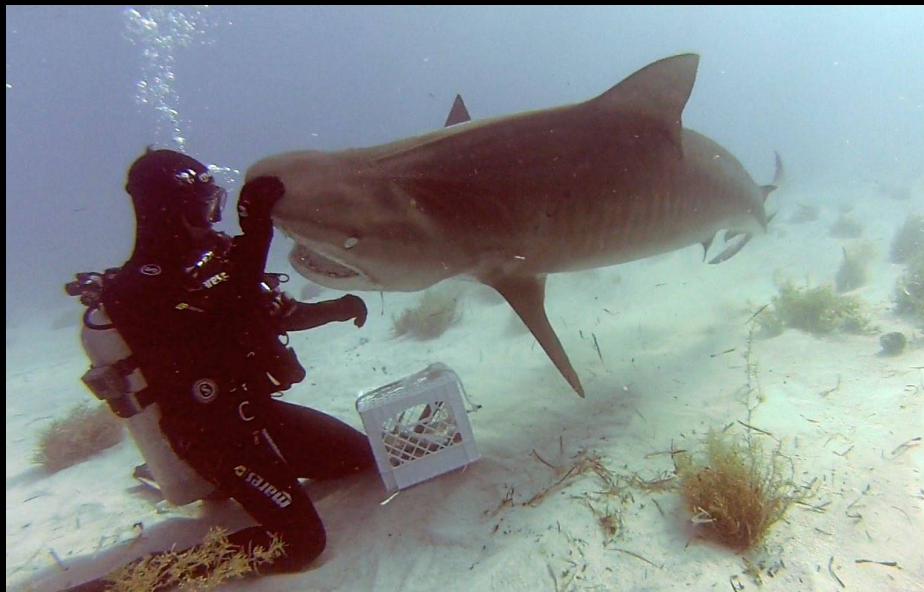
# Data Microservices with Spring

## Breaking the enterprise integration monolith

Fred Melo

 @fredmelo\_br

# About Me



Fred Melo



Mountain View, CA



@fredmelo\_br



<https://www.linkedin.com/in/melofred>

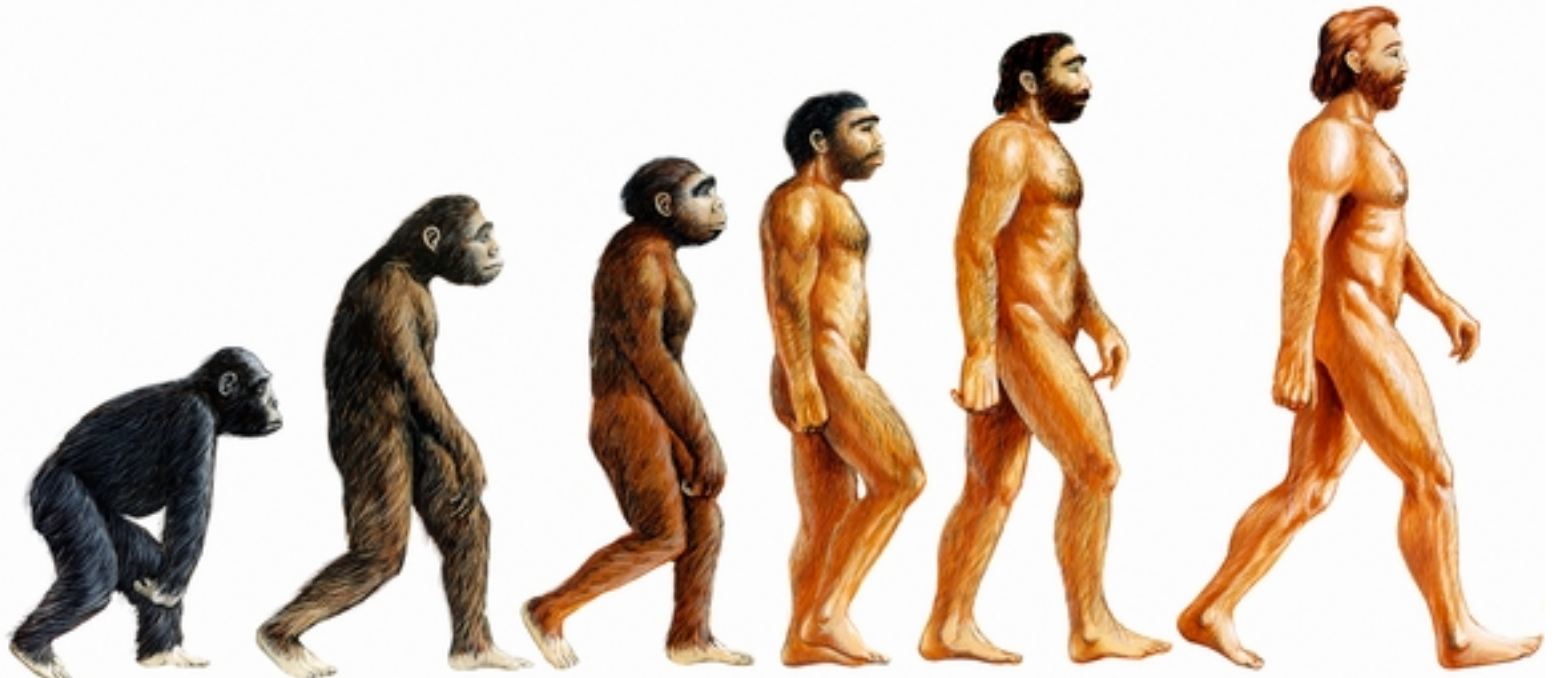
**Waterfall**

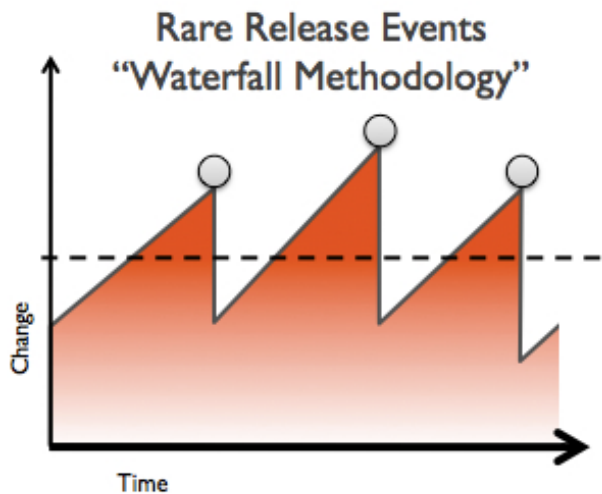
**Agile**

**Continuous Lean Integration**

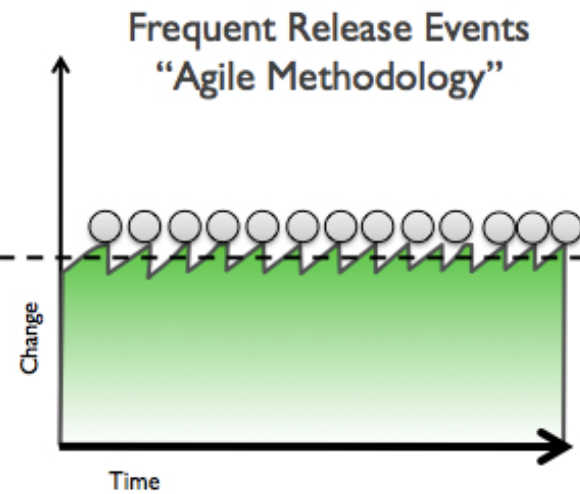
**Continuous Delivery**

**Continuous Deployment**





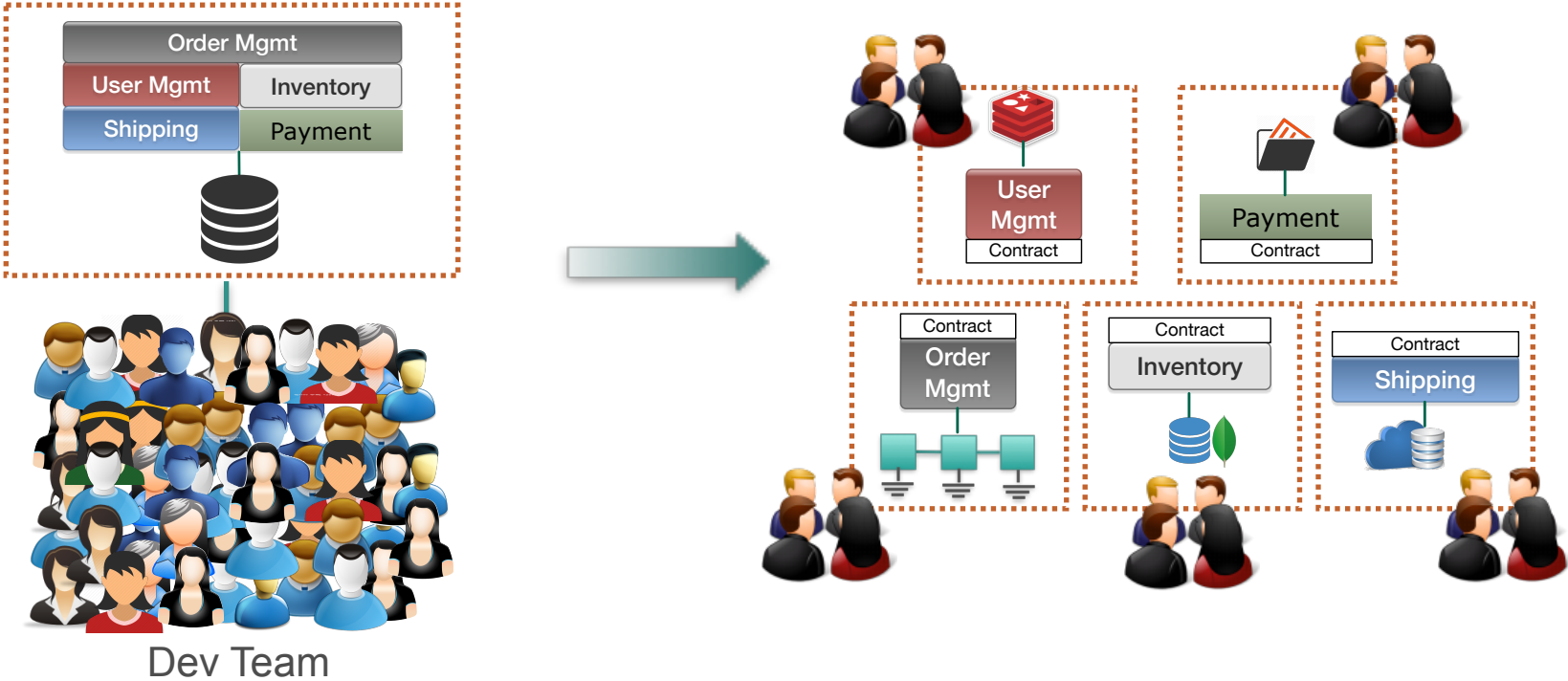
Effort Peaks  
High Risk



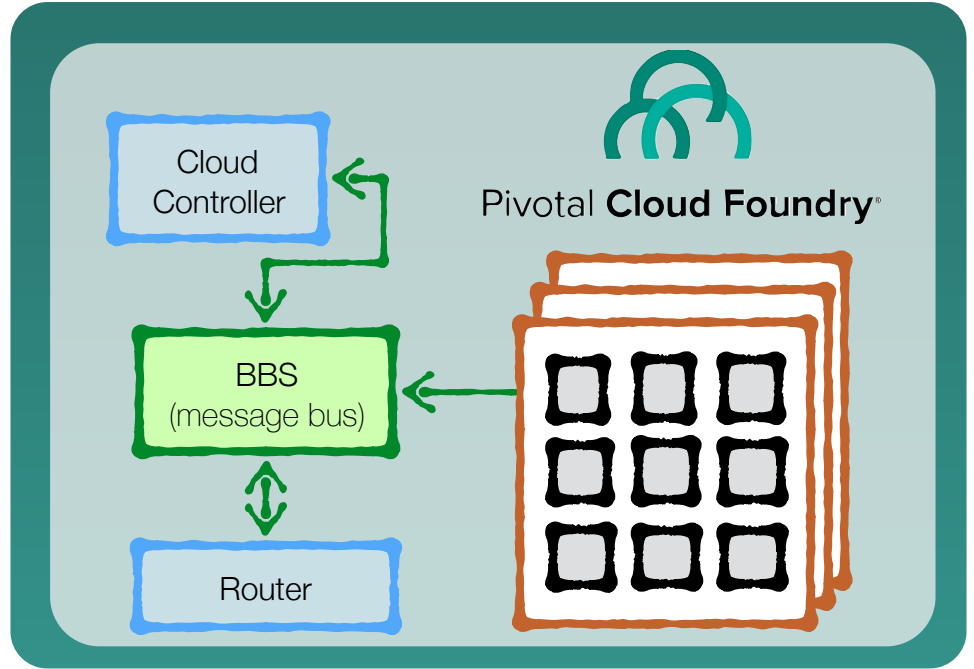
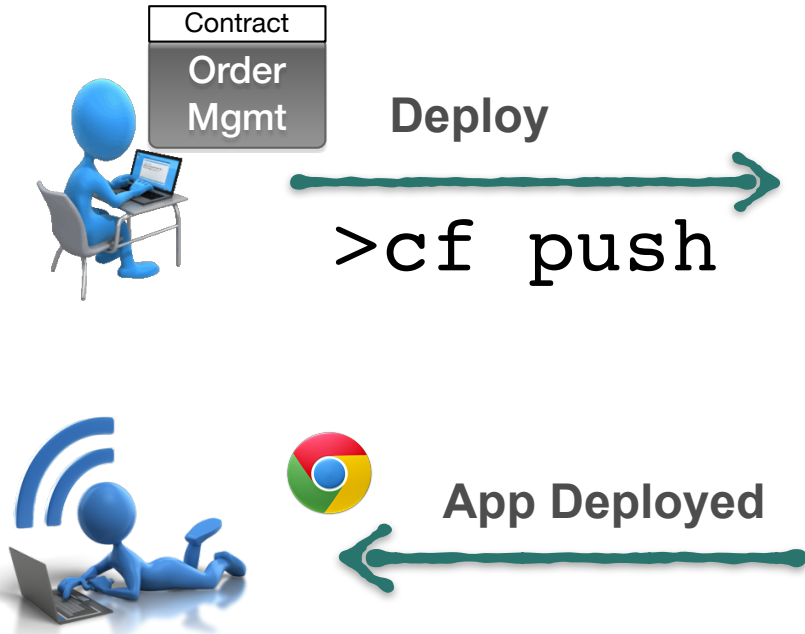
Smoother Effort  
Less Risk

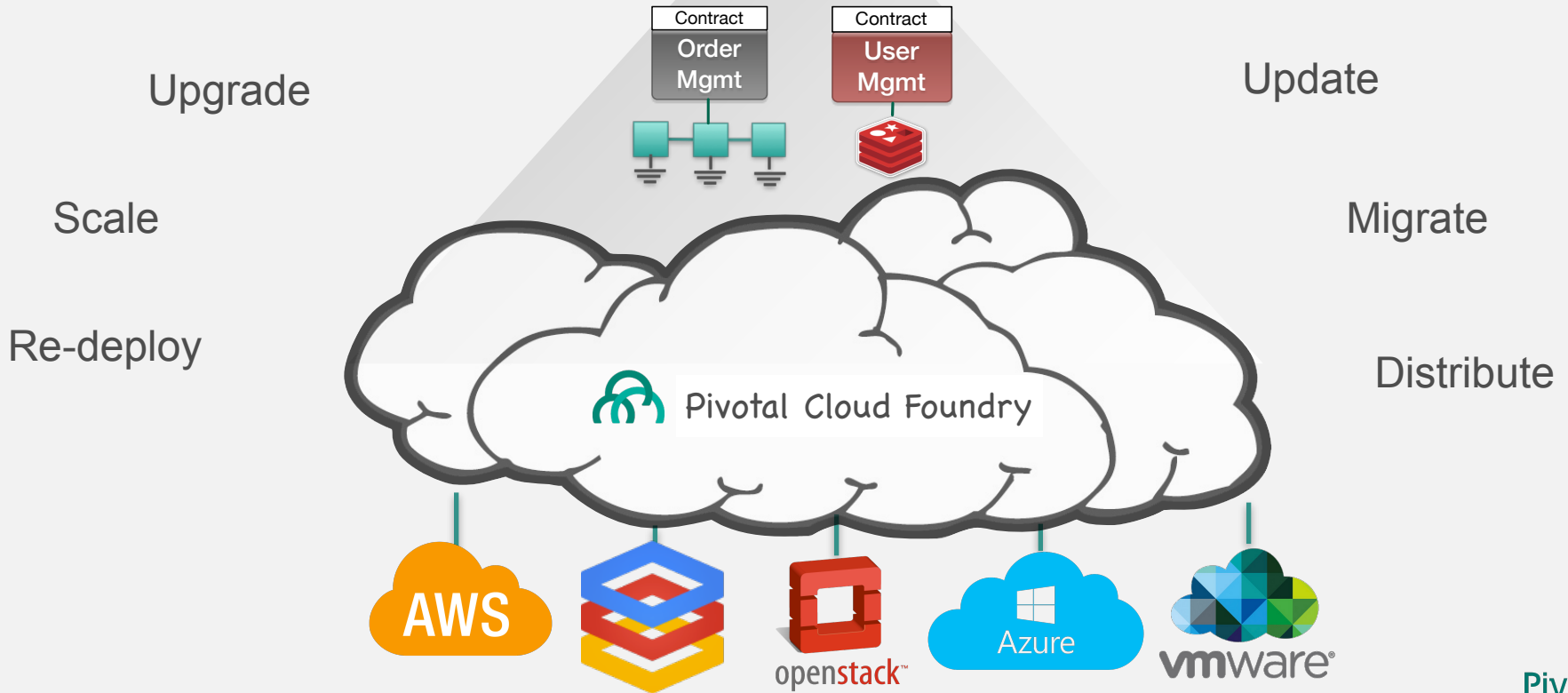
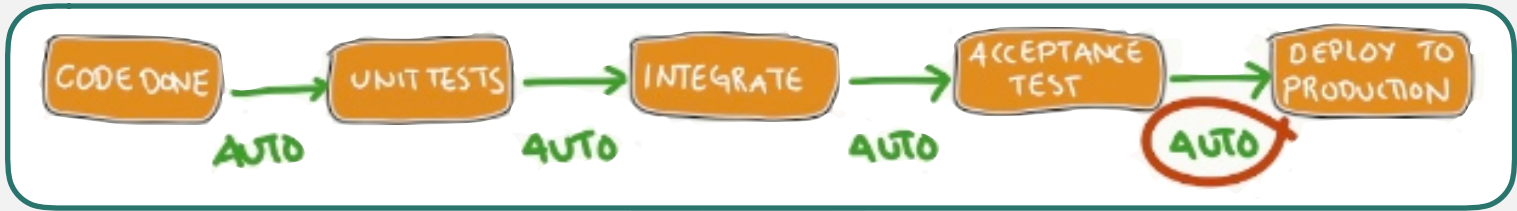
# Going Agile - Breaking the monolith

## Conway's Law and Microservices

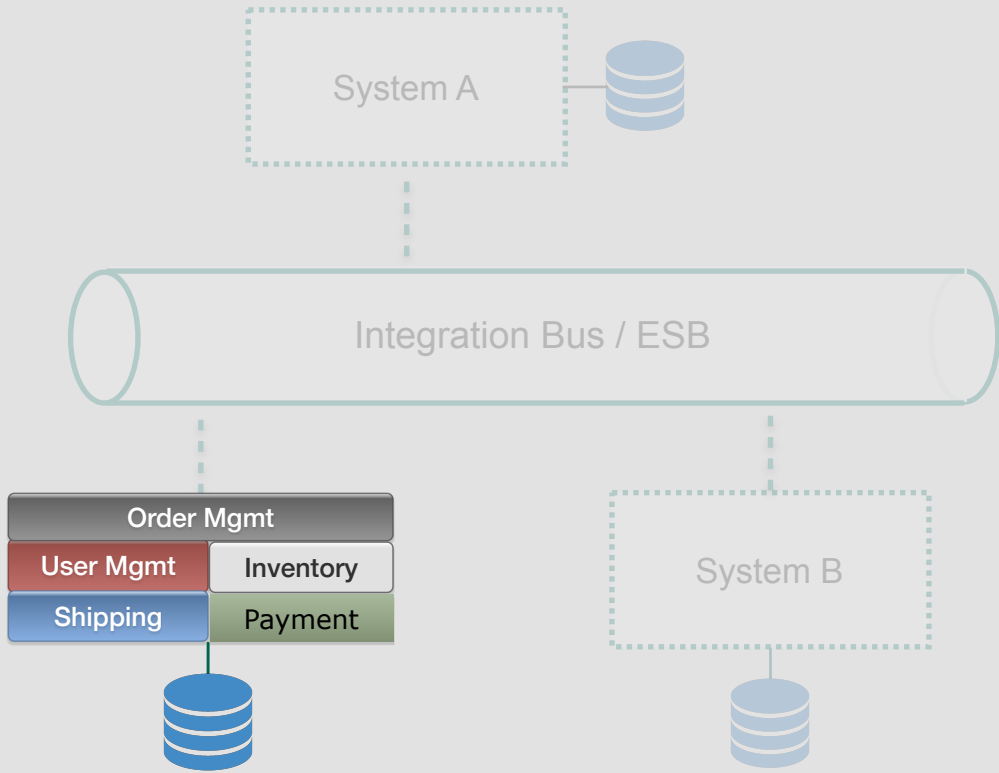


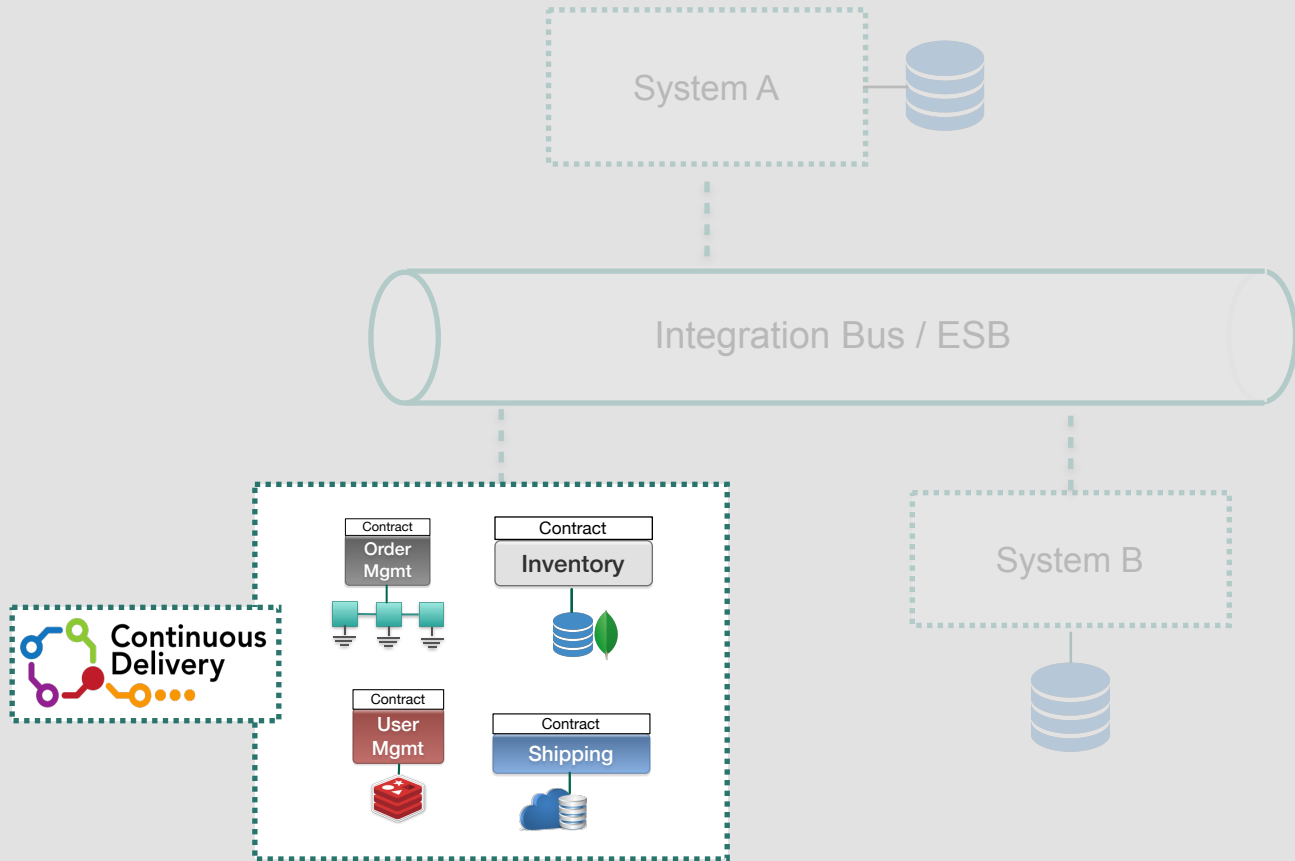
# Automating through the Platform

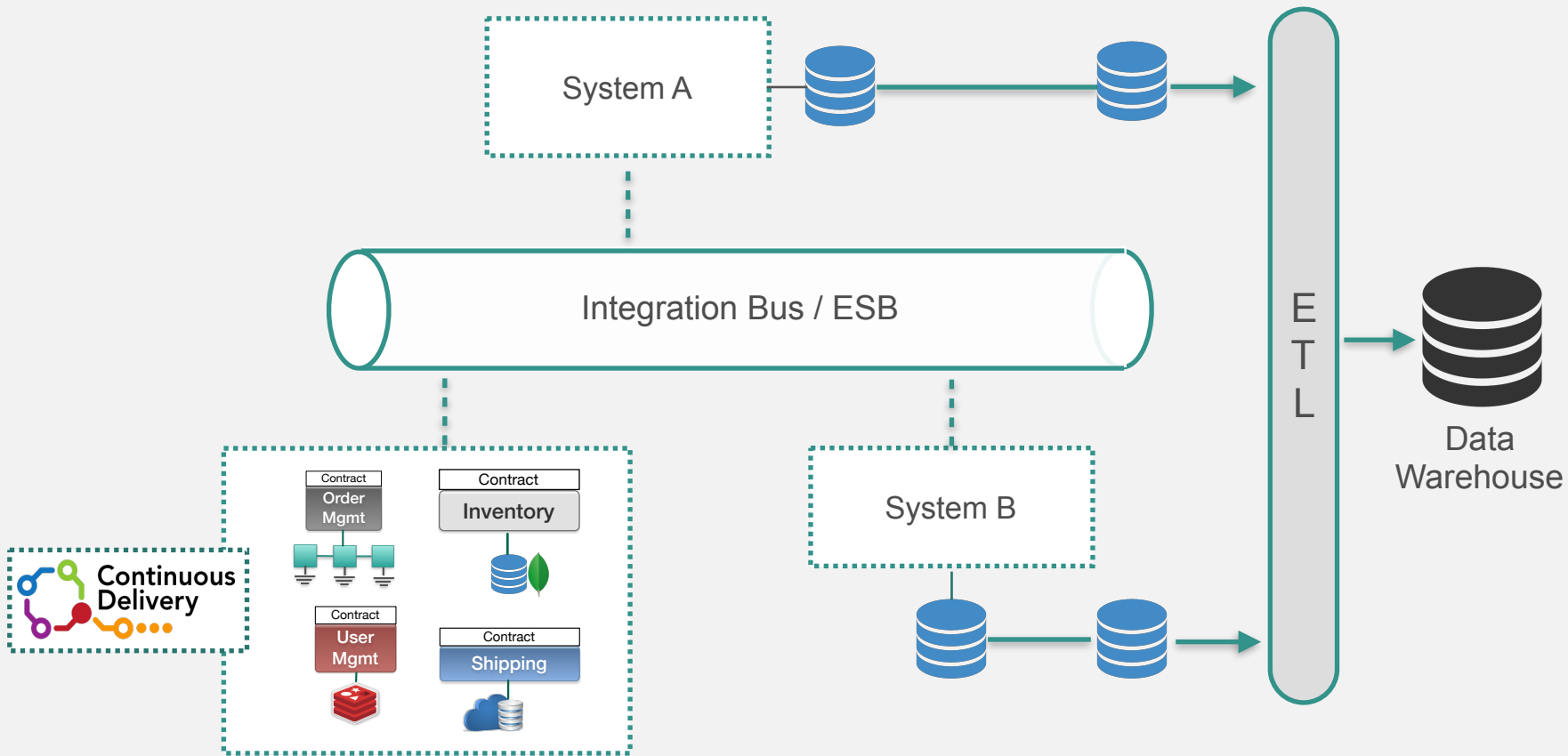


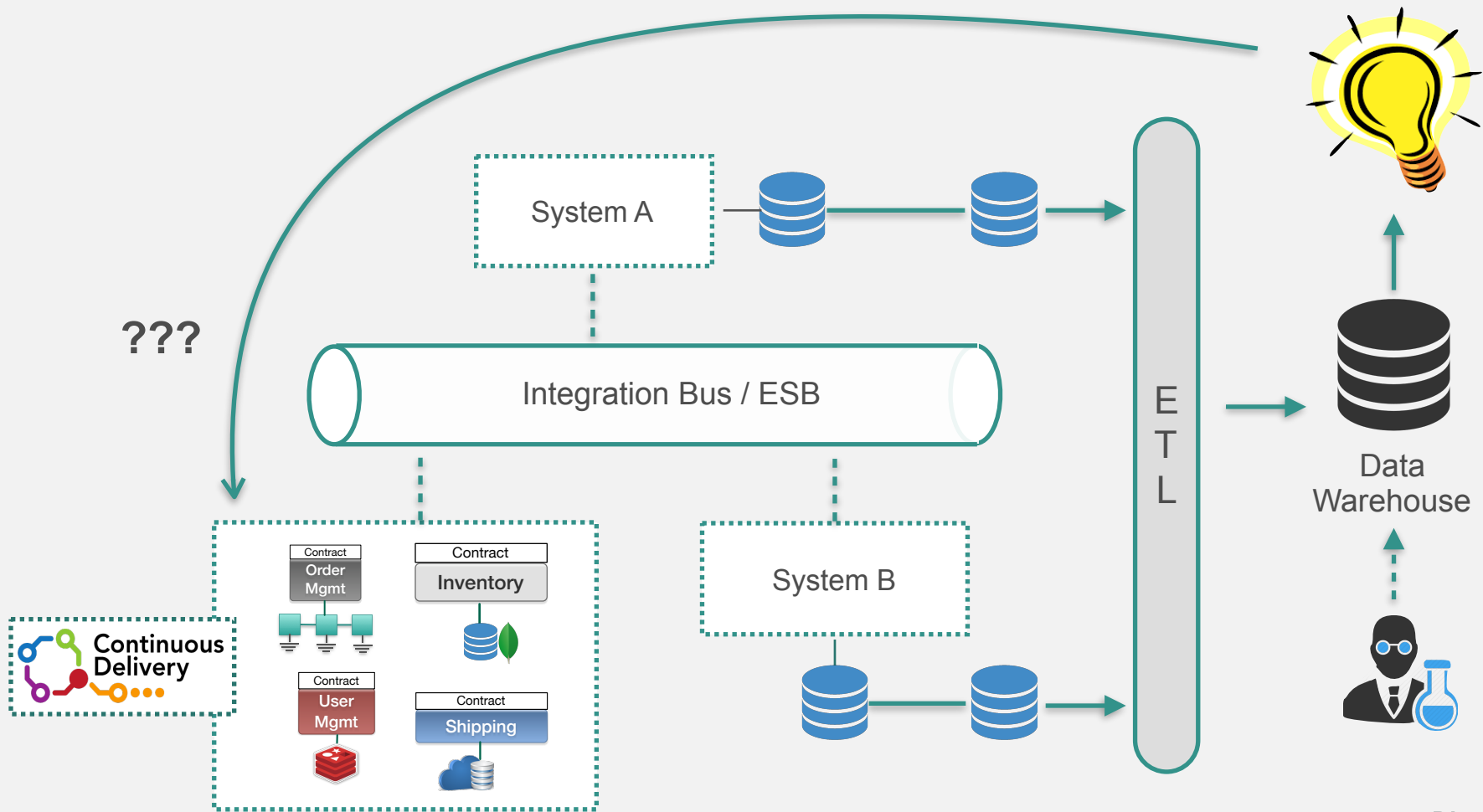








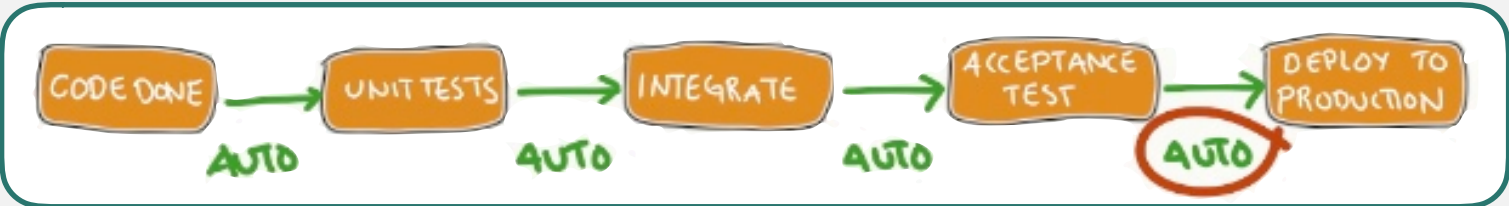




## Anatomy of a typical Integration Pipeline



Monolith



Upgrade

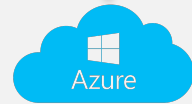
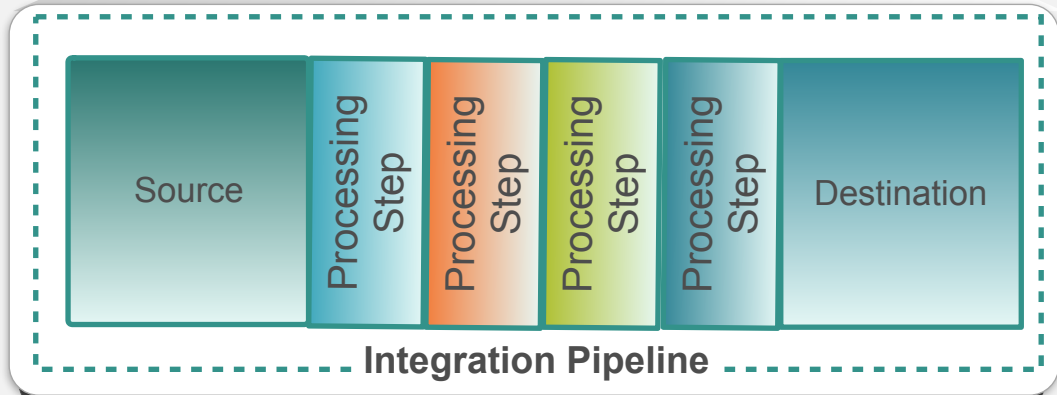
Scale

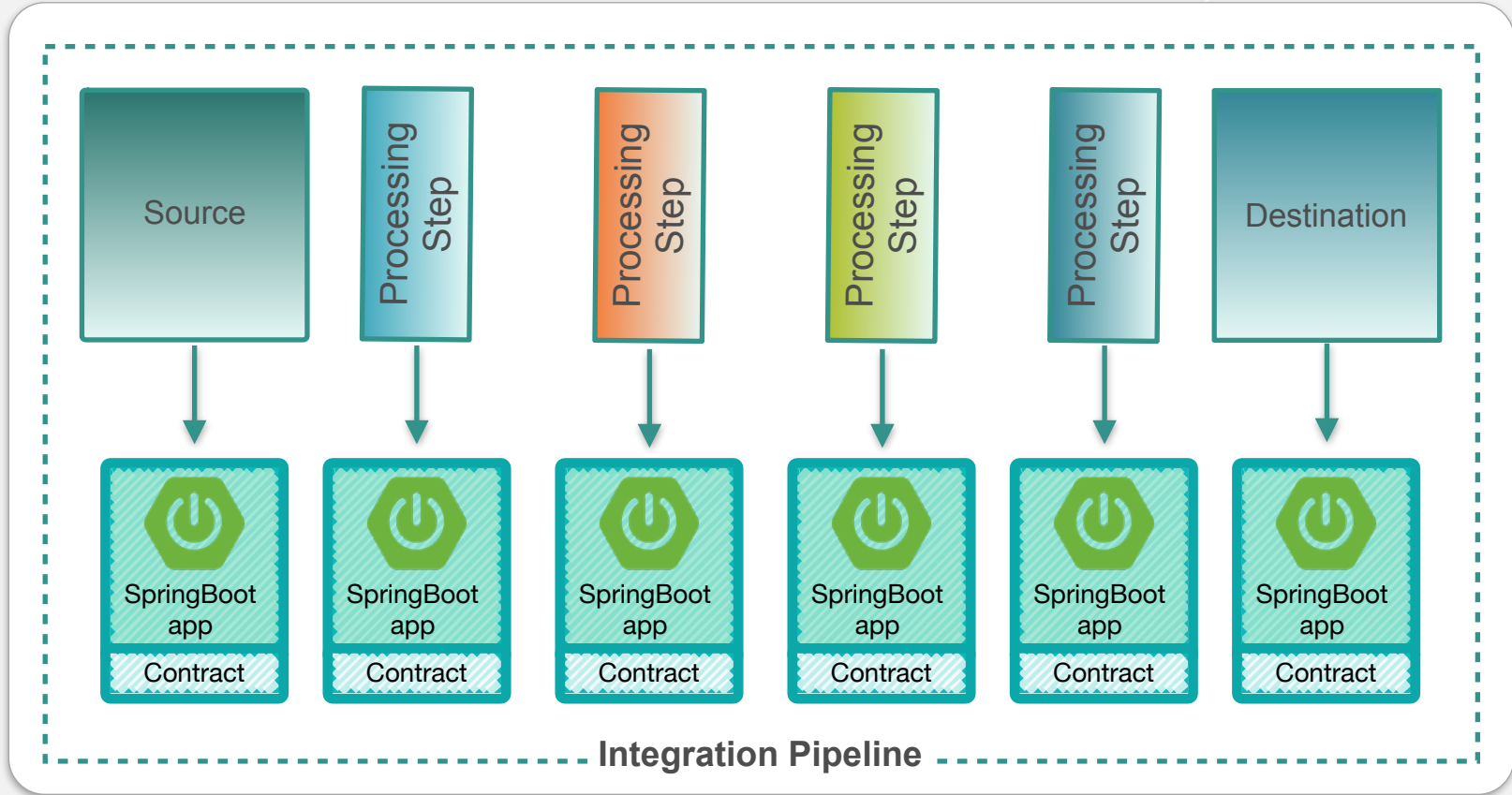
Re-deploy

Update

Migrate

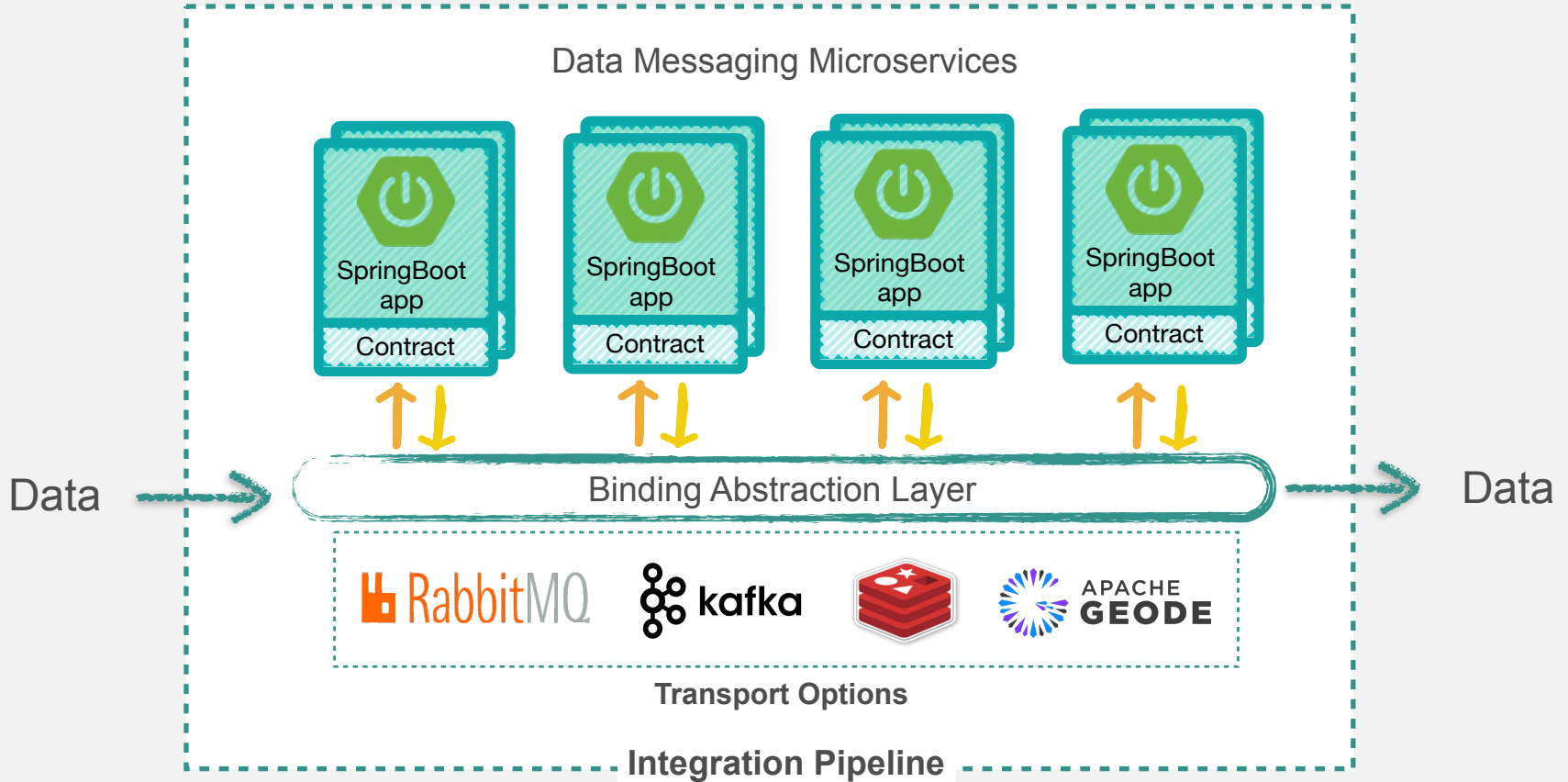
Distribute





Integration Pipeline

# Spring Cloud Stream





# Spring Cloud DataFlow

Orchestrate composable Data Microservices on modern runtimes

Infrastructure  
Transparency

Transport  
Transparency

Data Pipeline  
Visual Design

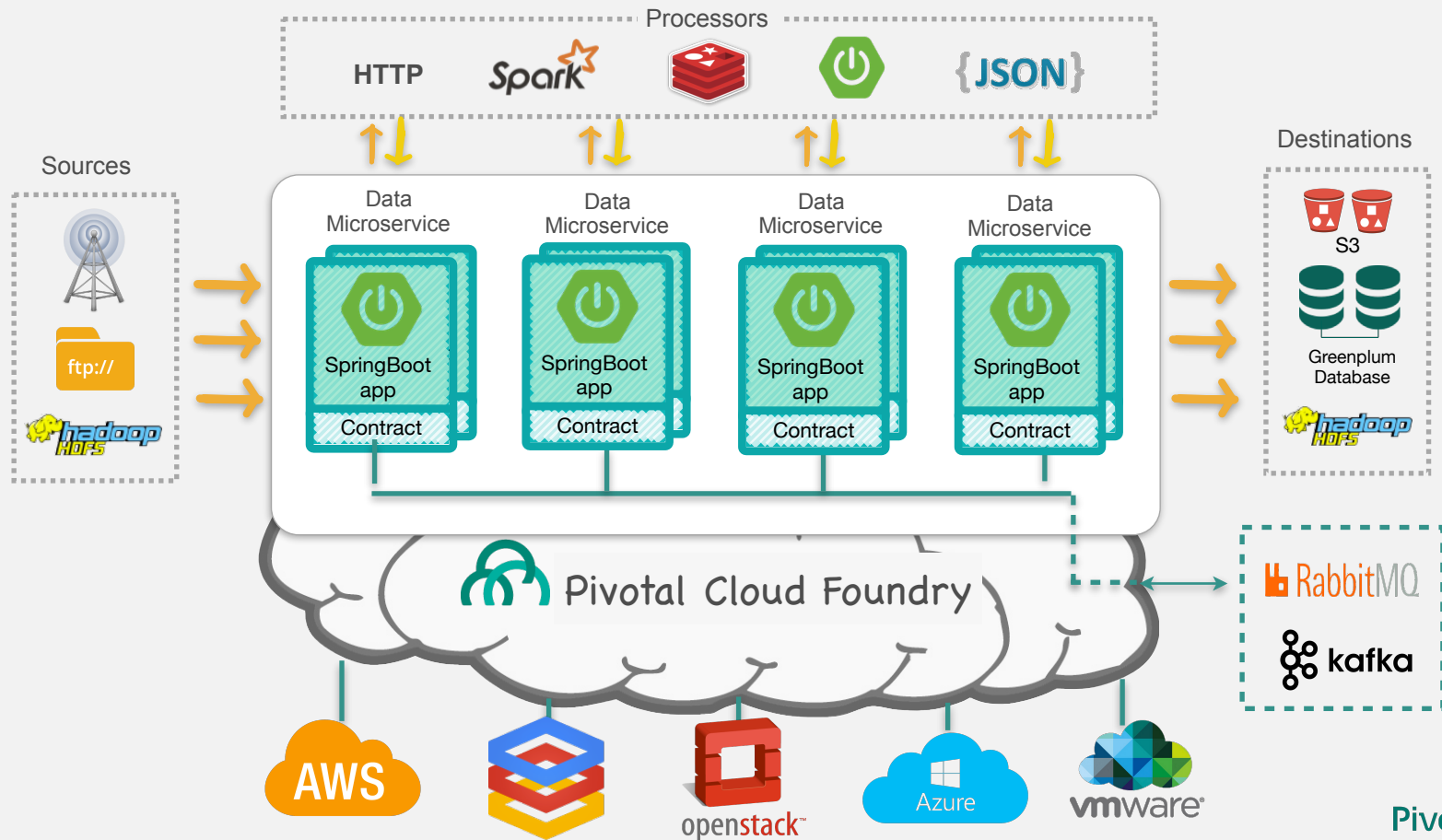
Integrated  
Metrics

Integrated  
Monitoring

Aggregated  
Logging

Auto Healing

Auto Scaling



Infrastructure  
Transparency

Transport  
Transparency

Data Pipeline  
Visual Design

Integrated  
Metrics

Integrated  
Monitoring

Aggregated  
Logging

Auto Healing

Auto Scaling

CODE DONE

```
@EnableBinding(Processor.class)  
public class EnrichProcessor {
```

```
@ServiceActivator(inputChannel = Processor.INPUT, outputChannel = Processor.OUTPUT)  
public Object process(Message<?> message) throws Exception {
```

```
SpelExpressionParser spel = new SpelExpressionParser();
```

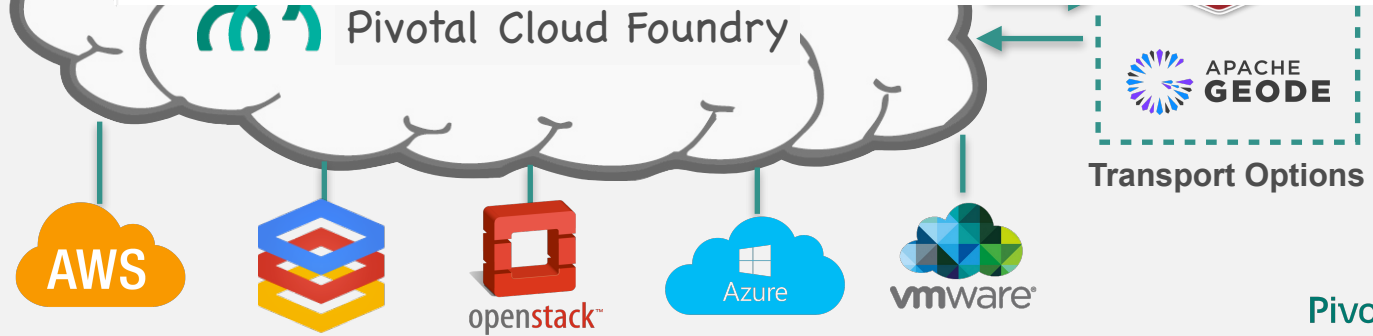
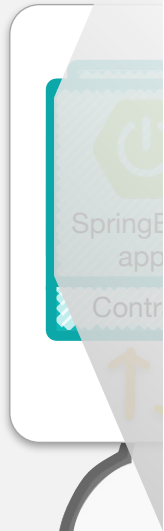
```
String account = spel.parseExpression("payload.accountId").getValue(evaluationContext);
```

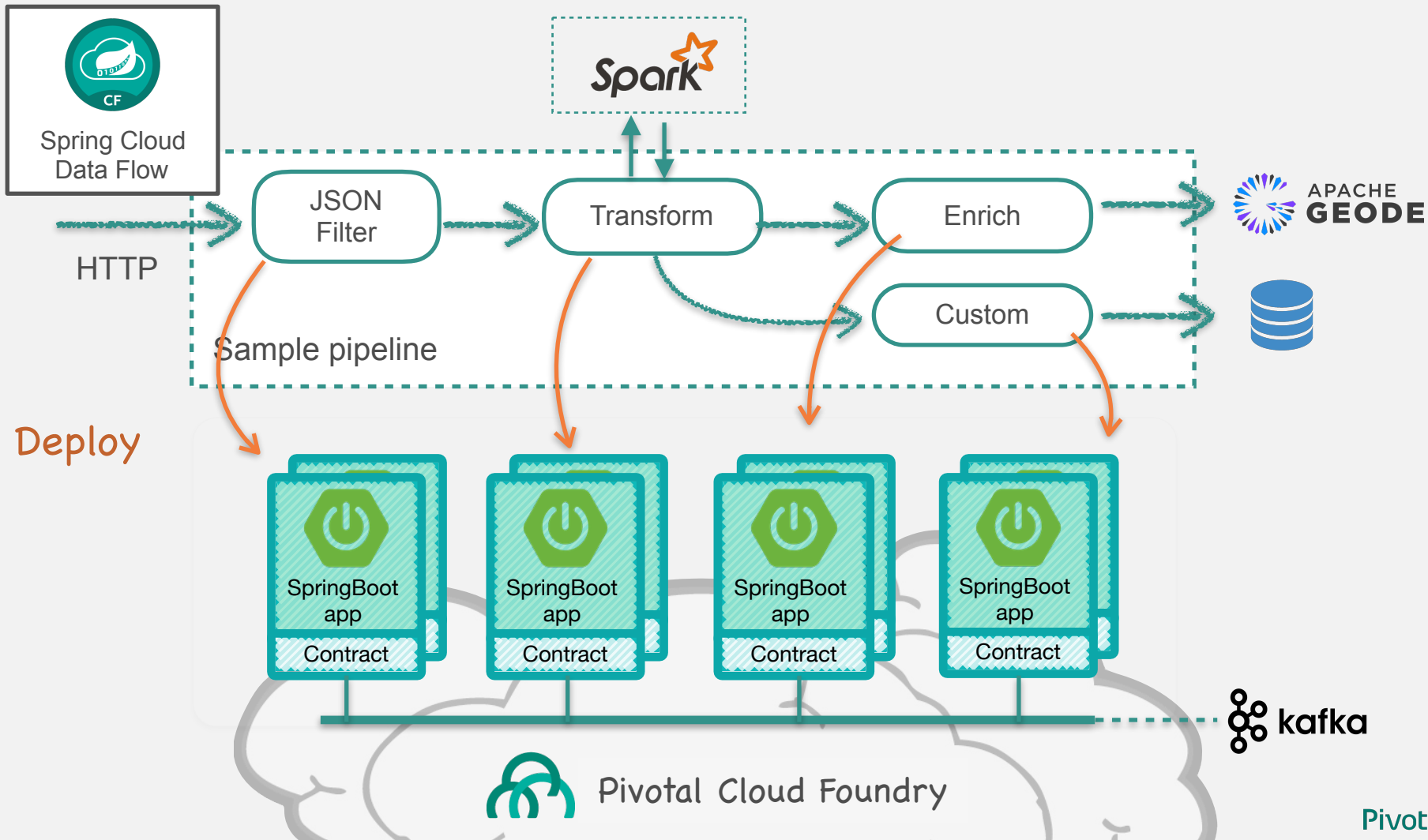
```
String device = spel.parseExpression("payload.deviceId").getValue(evaluationContext);
```

```
String location = redis.opsForValue().get("device::"+device);
```

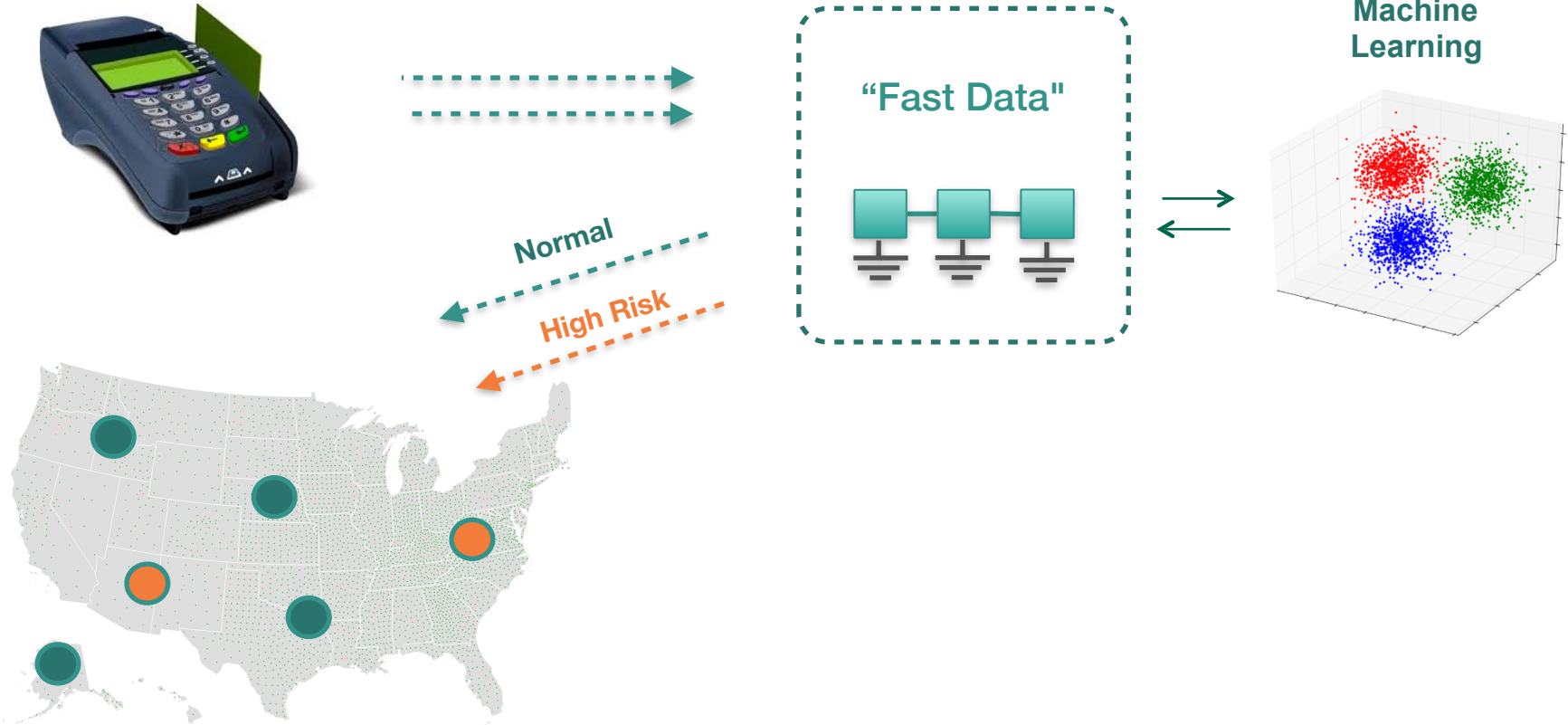
```
String home = redis.opsForValue().get("home::"+account);
```

```
double homeLat=0 ,  
       homeLong=0,  
       locLat=0,  
       locLong=0,  
       distance=0;
```



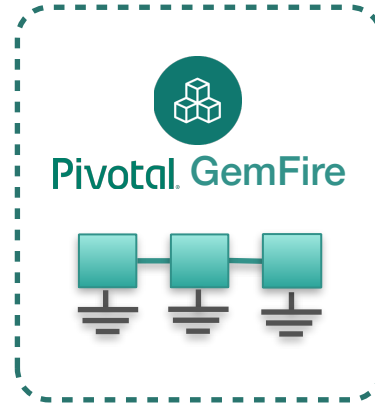


# DEMO

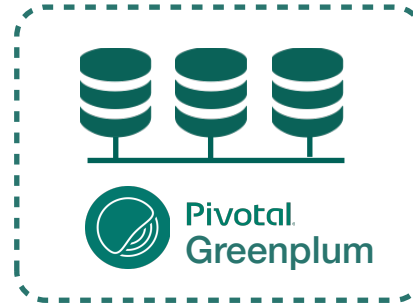
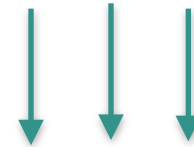
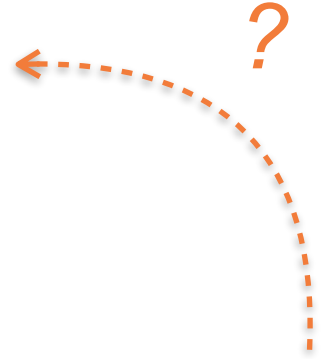
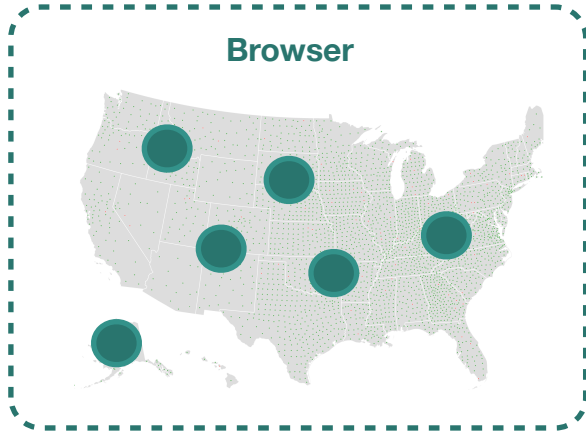




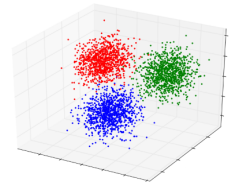
Rest / HTTP



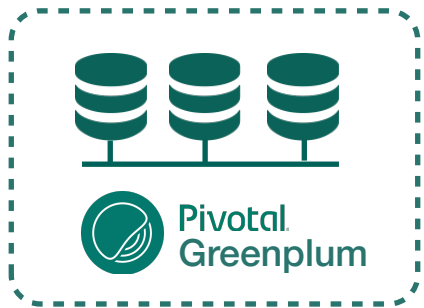
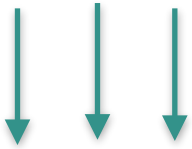
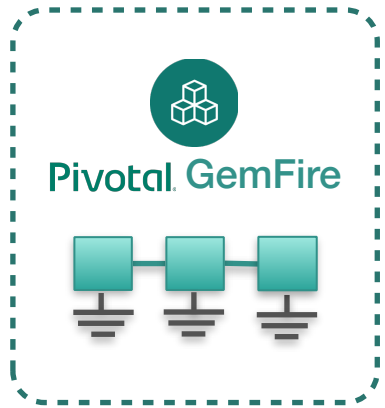
Transactions



Machine Learning

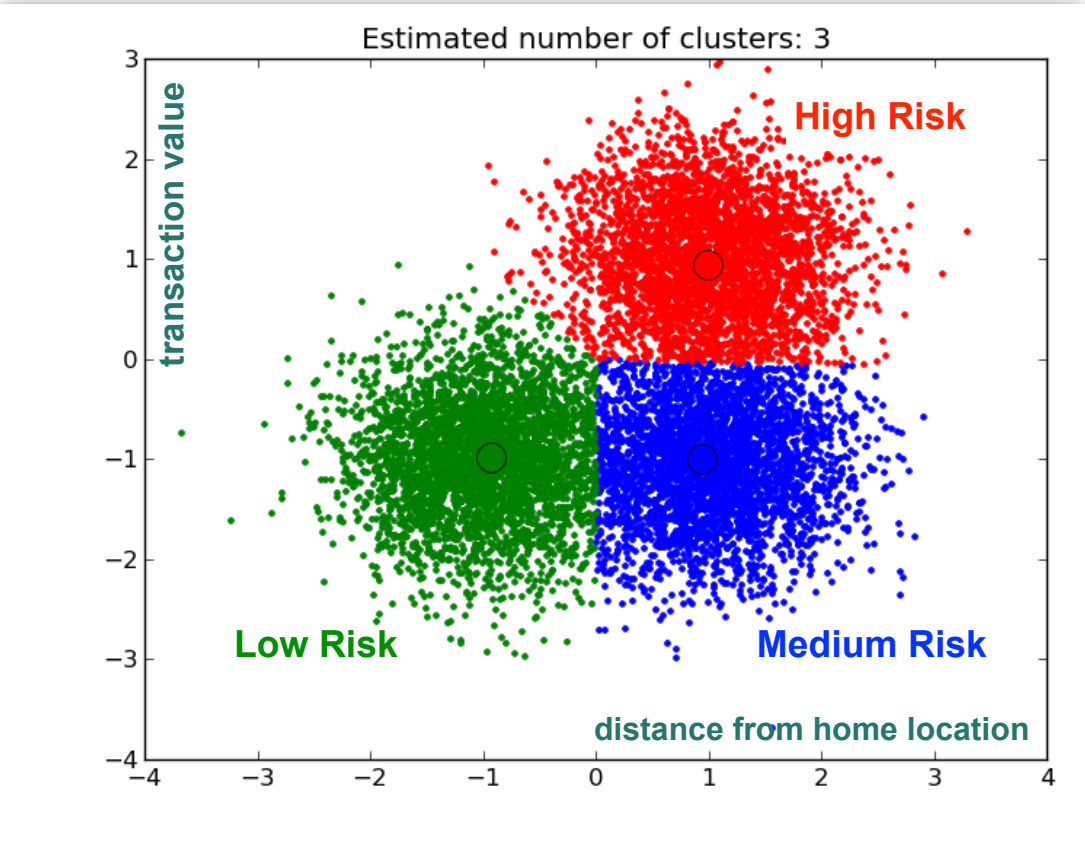
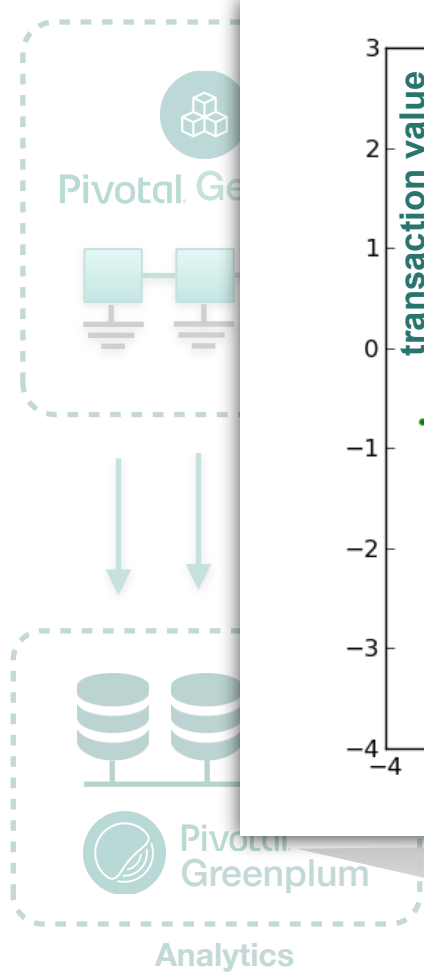


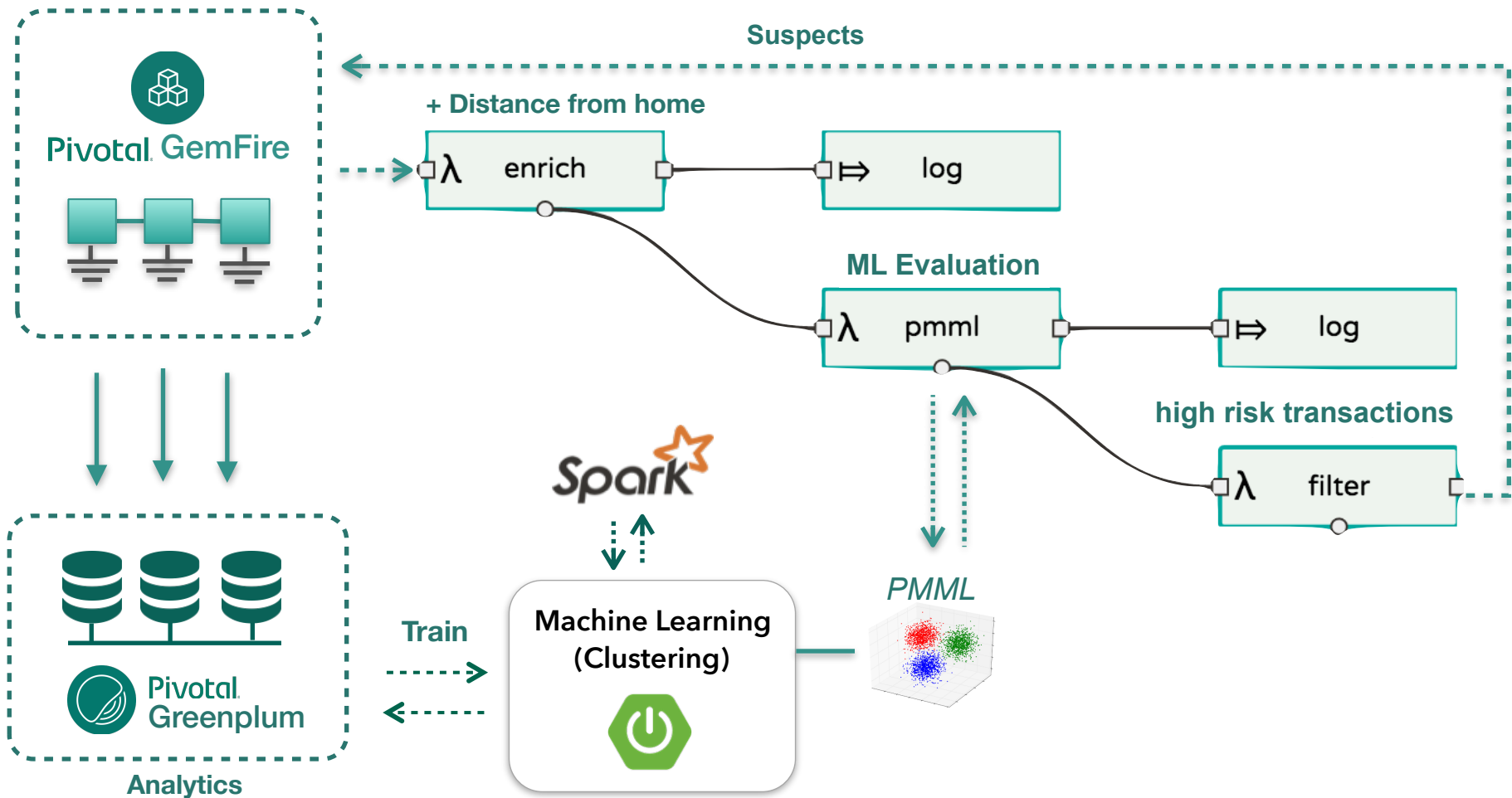
Analytics



Analytics



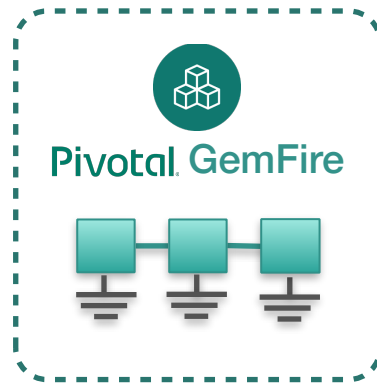




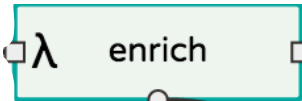




Rest / HTTP



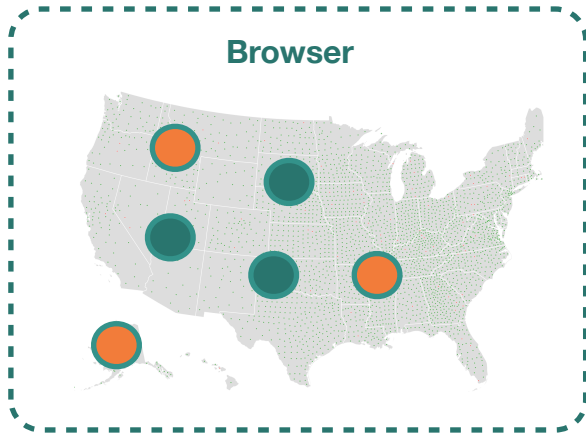
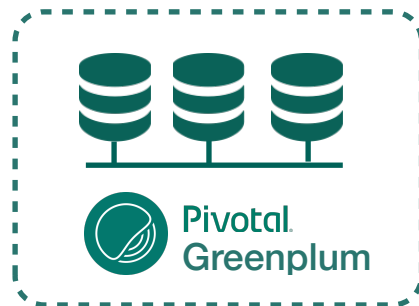
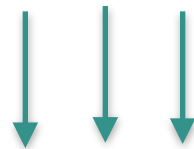
+ Distance from home

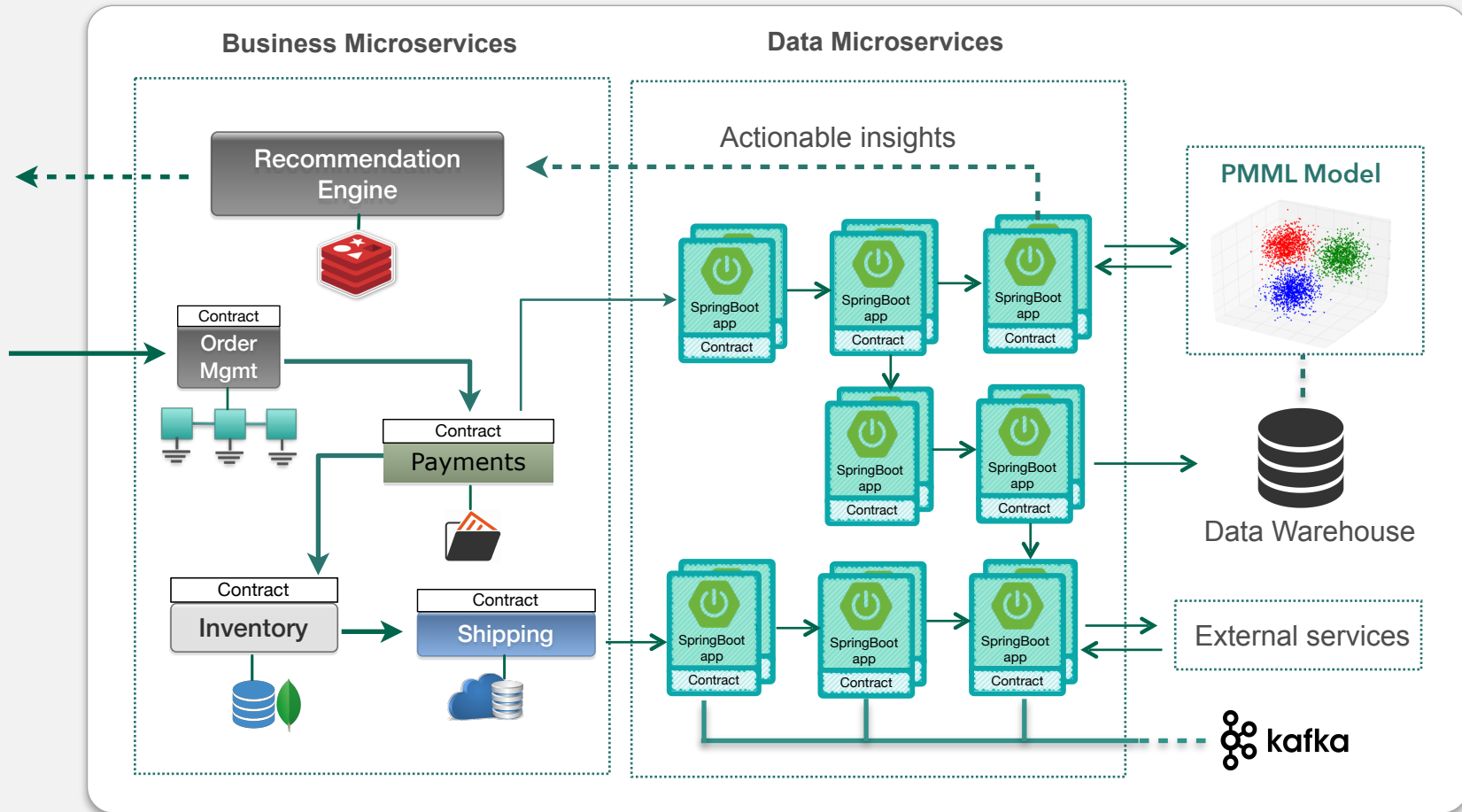


Transactions



High Risk Alerts





# Pivotal

A NEW PLATFORM FOR A NEW ERA