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
Rare Release Events  
“Waterfall Methodology”  

Frequent Release Events  
“Agile Methodology”  

Effort Peaks  
High Risk  

Smoother Effort  
Less Risk
Going Agile - Breaking the monolith

Conway’s Law and Microservices

Dev Team
Automating through the Platform

Deploy

> cf push

App Deployed

Order Mgmt

Cloud Controller

BBS (message bus)

Router

Pivotal Cloud Foundry

Contract
Anatomy of a typical Integration Pipeline

Data → Source → Processing Step → Processing Step → Monolith → Processing Step → Processing Step → Destination → Data
Integration Pipeline

Source

Processing Step

Processing Step

Processing Step

Processing Step

Destination

SpringBoot app

Contract

SpringBoot app

Contract

SpringBoot app

Contract

SpringBoot app

Contract

SpringBoot app

Contract

SpringBoot app

Contract

Integration Pipeline
Spring Cloud Stream

Data Messaging Microservices

- SpringBoot app
  - Contract

Binding Abstraction Layer

Transport Options
- RabbitMQ
- kafka
- APACHE GEODE

Integration Pipeline
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

Sources
Destinations
Processors
HTTP
Spark
JSON

SpringBoot app
Contract

Pivotal Cloud Foundry

RabbitMQ
kafka

Greenplum Database
S3

AWS
openstack
Azure
vmware
@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

“Fast Data"

Normal

High Risk

Machine Learning

© Copyright 2014 Pivotal. All rights reserved.
Estimated number of clusters: 3

- **High Risk**
- **Low Risk**
- **Medium Risk**

- **Distance from home location**
- **Transaction value**

Clusters are color-coded and labeled according to risk level.
Machine Learning (Clustering)

Train

PMML

high risk transactions

ML Evaluation

enrich

λ

log

λ

PMML

filter

λ

log

+ Distance from home

Suspects

λ

Pivotal GemFire

Pivotal Greenplum

Analytics

Spark
Pivotal
A NEW PLATFORM FOR A NEW ERA