



DAMIS

中国数据资产管理峰会

CHINA DATA ASSET MANAGEMENT SUMMIT

数据微服务化框架和进展

演讲人：周晖 (fzhou@pivotal.io)
Pivotal首席云计算架构师

目录

- **Pivotal和云原生/微服务理论**
- **Spring微服务**
- **Spring Cloud数据微服务**

Pivotal的业界大牛

Paul Mariz
Pivotal董事会主席，原微软Windows之父



James Waters
CloudFoundry开源技术联合创始人



Joshua McKenty
Openstack联合创始人



Eric Ries
<Lean Startup>作者，精益创业方法论倡导者



Kenny Bastani
Spring开源技术和NEO4J图数据主要技术贡献者之一



Rob Mee
Pivotal CEO，硅谷敏捷实践领导者



Benjamin Black
AWS云计算技术第一代框架设计领导人物



Andrew Shaffer
Puppet联合创始人，DevOps理念联合初创



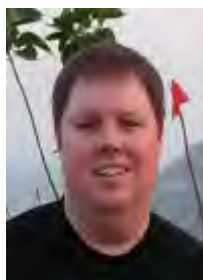
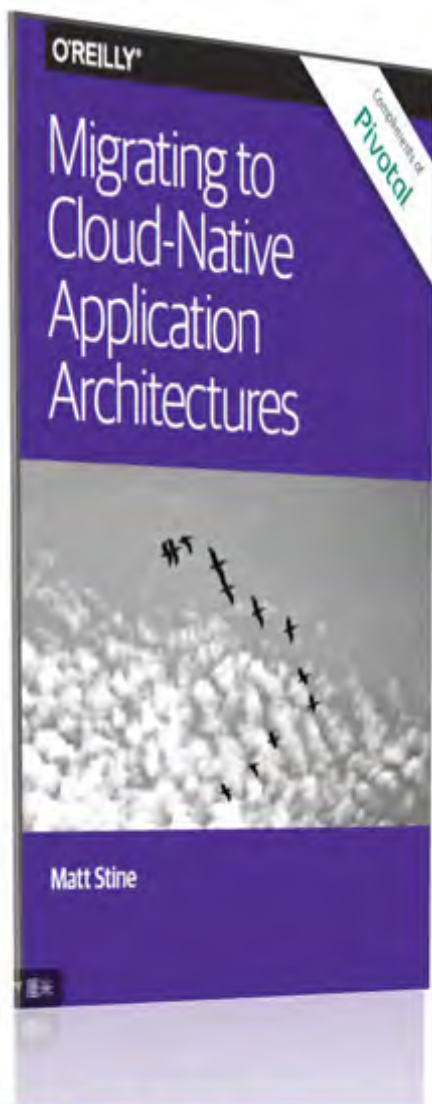
Matt Stine
Cloud Native理念初创



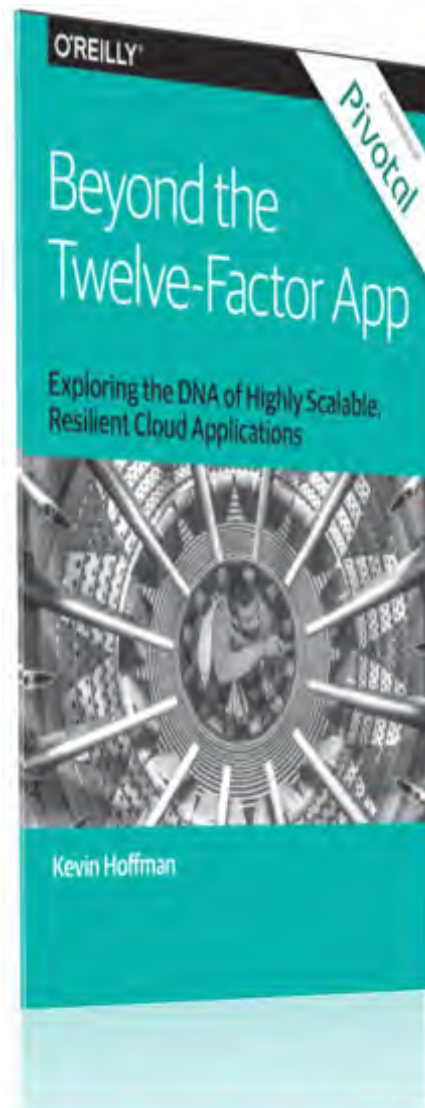
Mark Thomas
Tomcat开源软件主要技术贡献者之一



Pivotal的云原生微服务理论

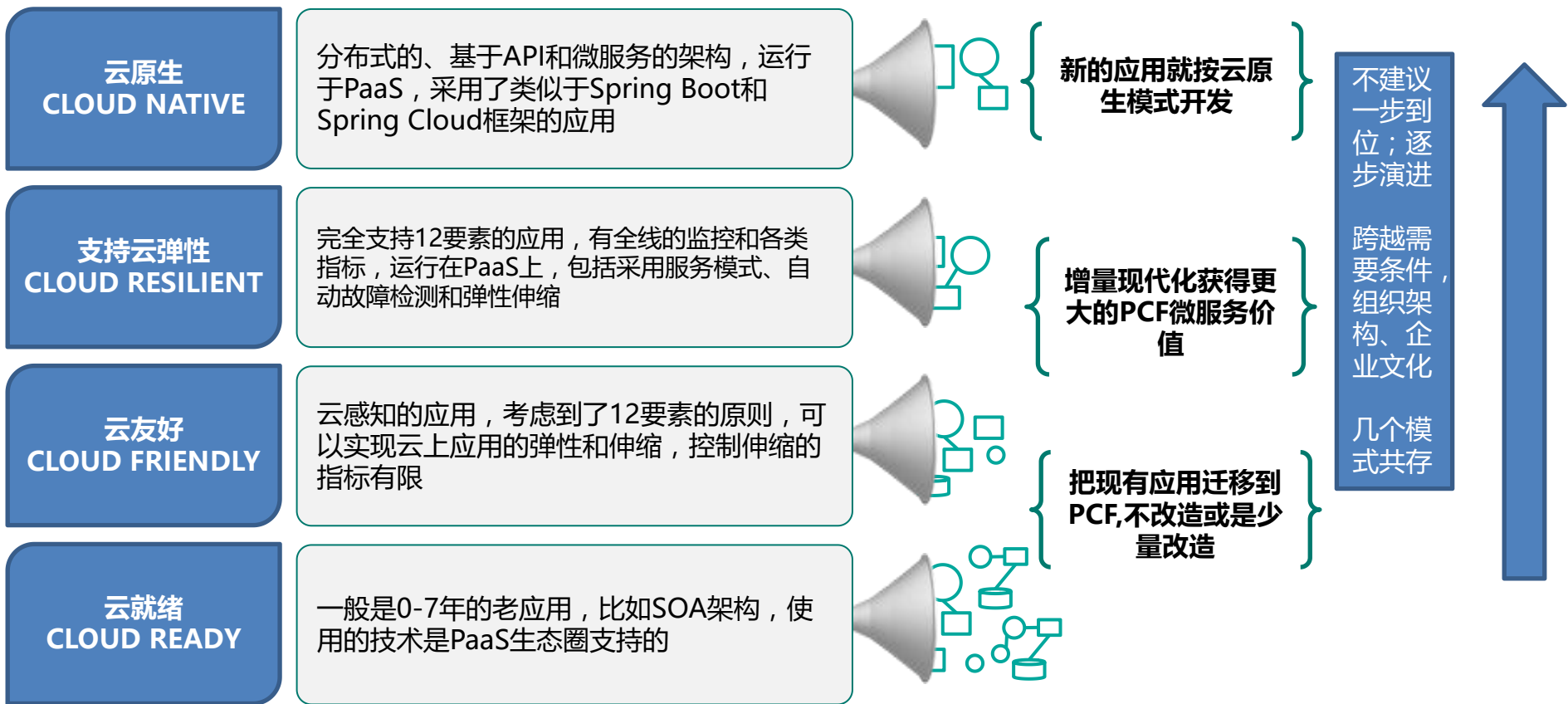


- <<Migrating to Cloud native>>的作者
- 2014年提出Cloud Native概念
- 2015年定义了云原生架构



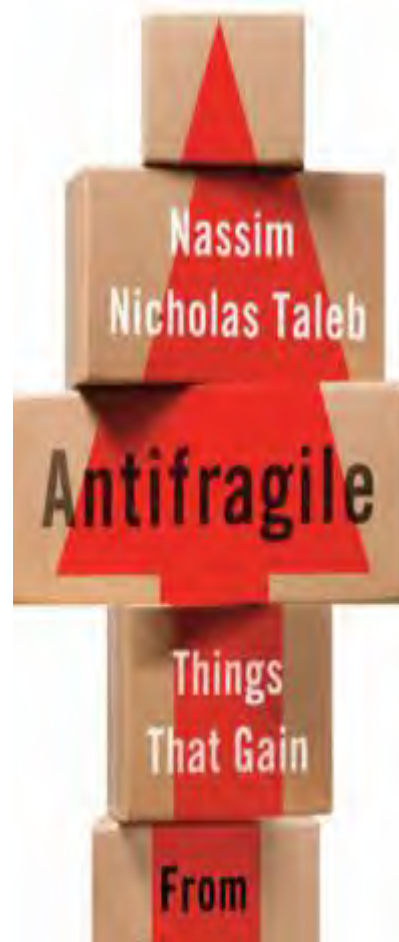
细化云原生应用和微服务架构的具体落地的技术原则和方法论

云应用成熟度模型

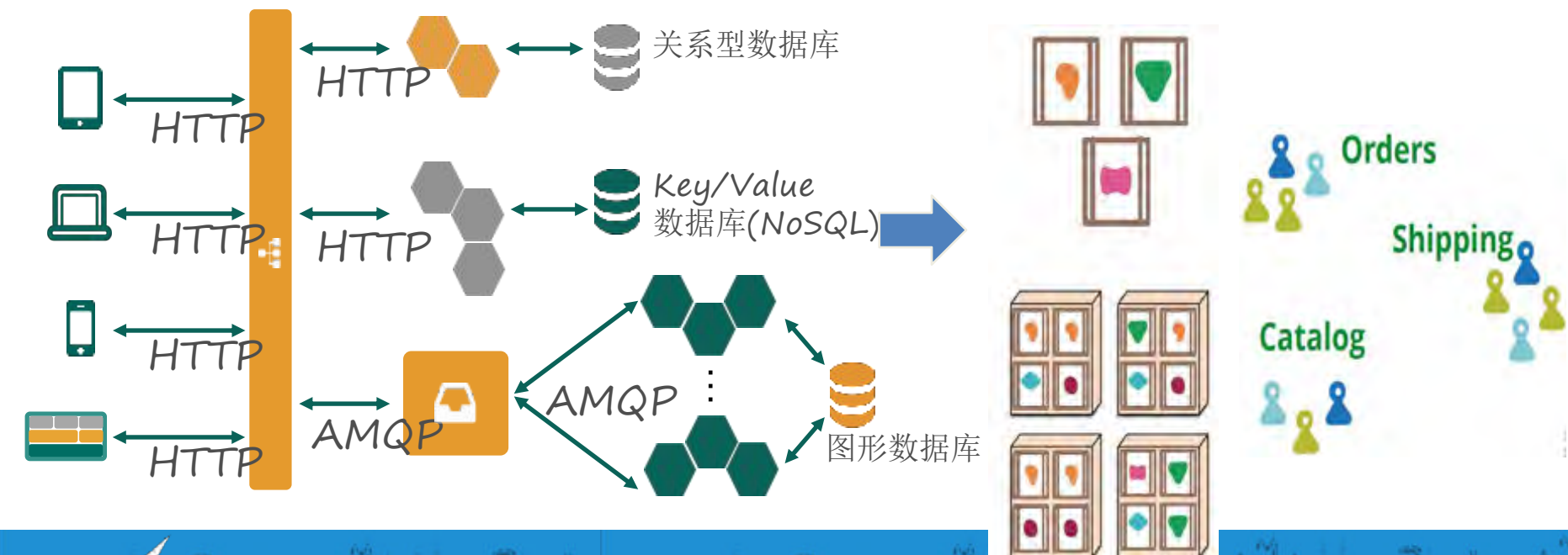
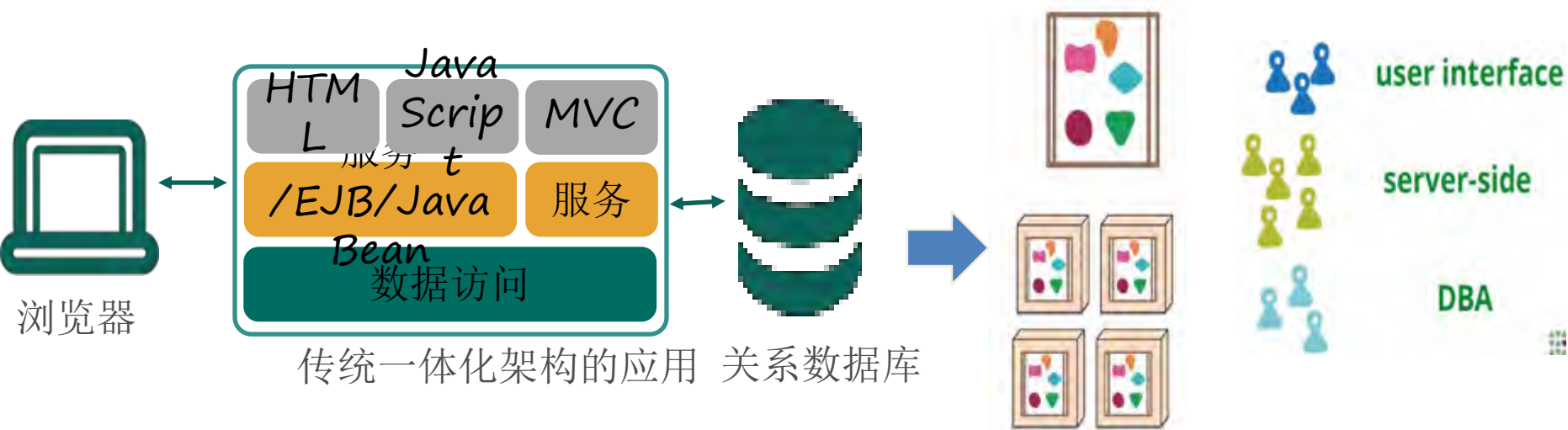


云原生应用的特征

- 符合十二要素
- 微服务
- 自服务的敏捷基础设施
- 基于API的协作
- Antifragility (反脆弱性)



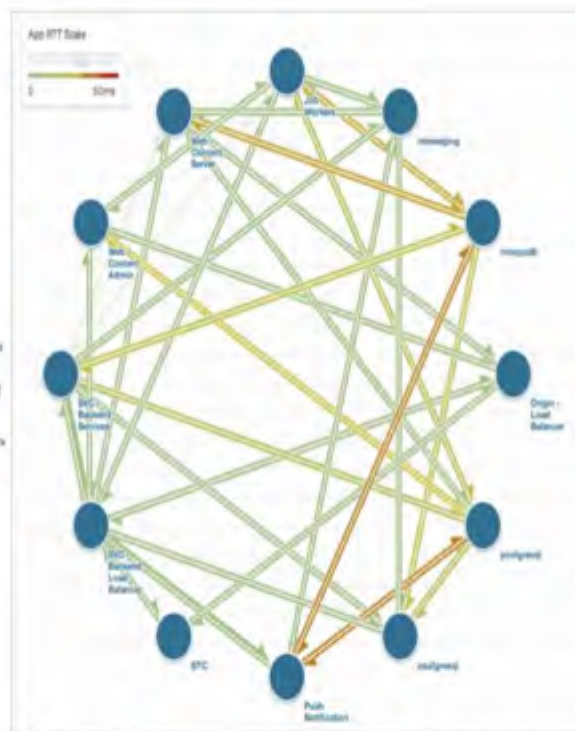
传统巨石应用和云原生微服务应用的概念模型



云原生应用的微服务是这样的 不强调分层架构



Netflix



Gilt Groupe (12 of 450)



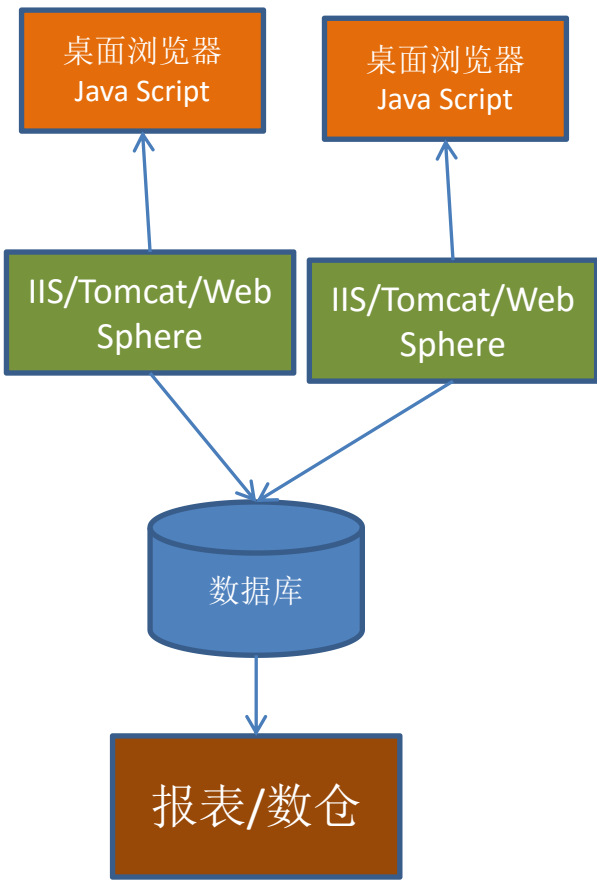
Twitter



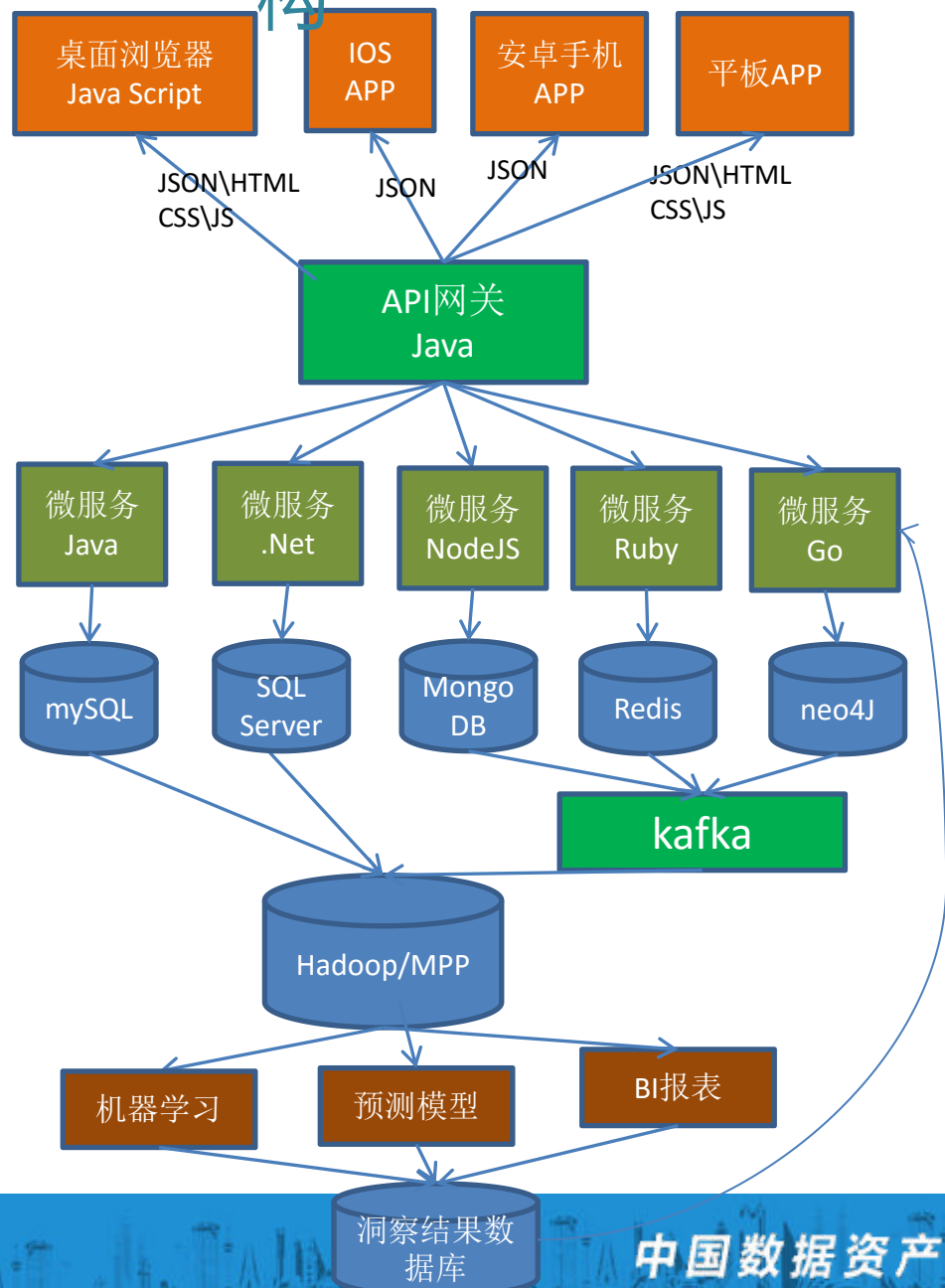
目录

- Pivotal和云原生/微服务理论
- Spring微服务
- Spring Cloud数据微服务

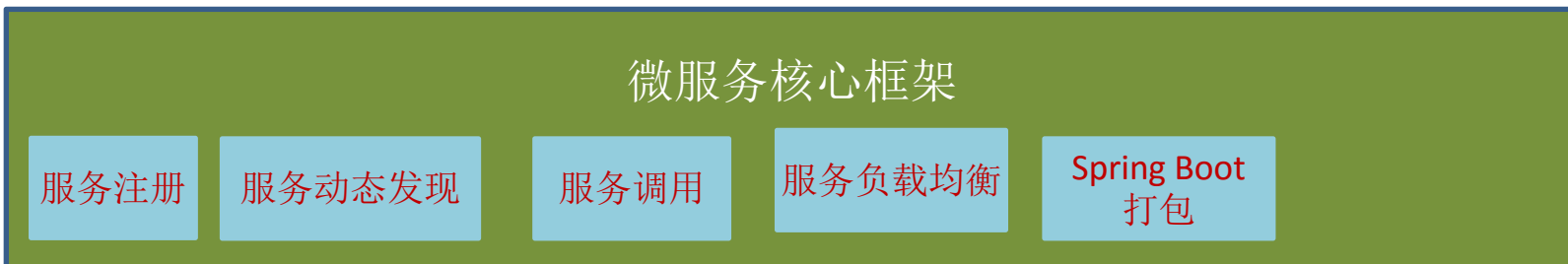
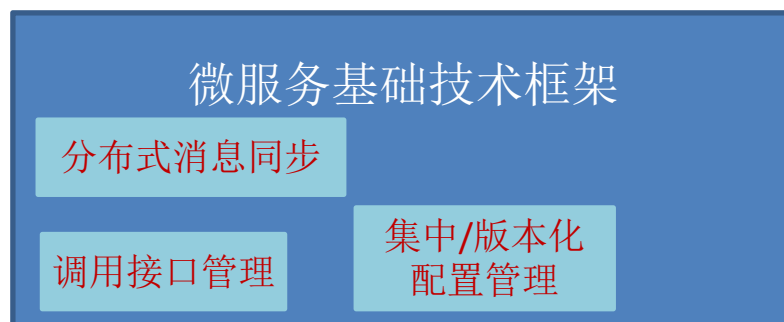
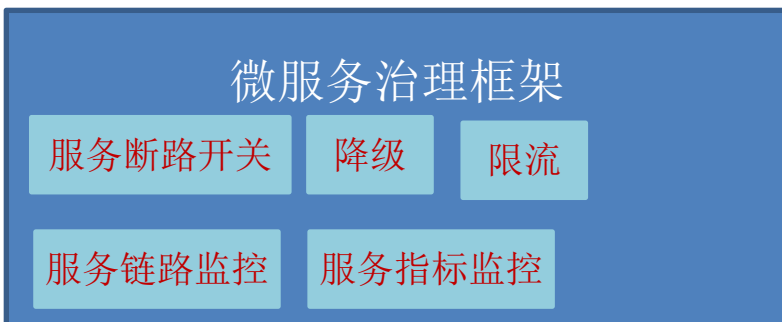
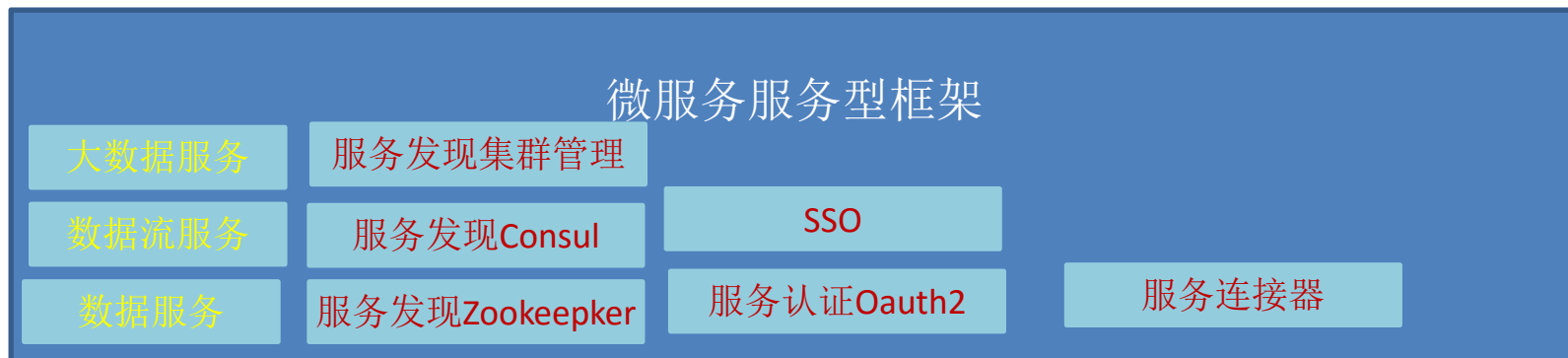
传统架构



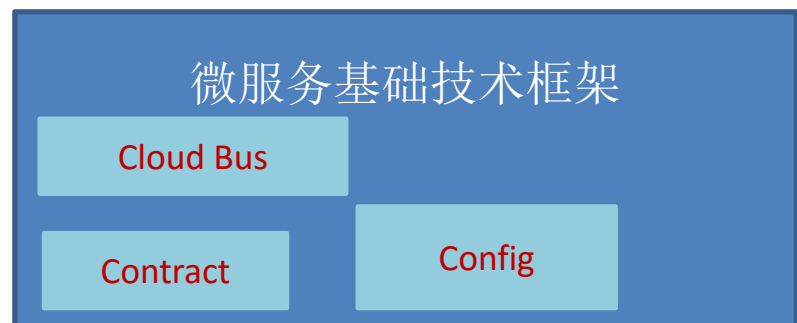
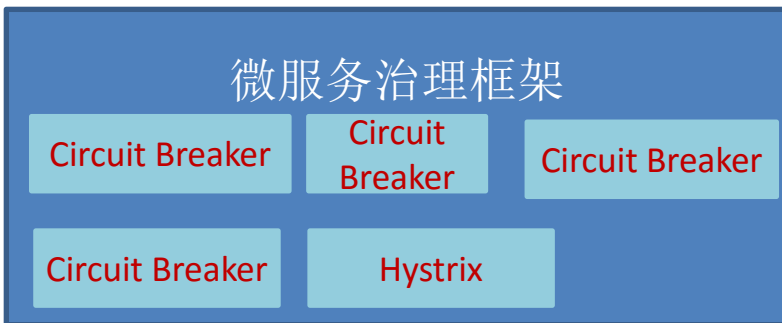
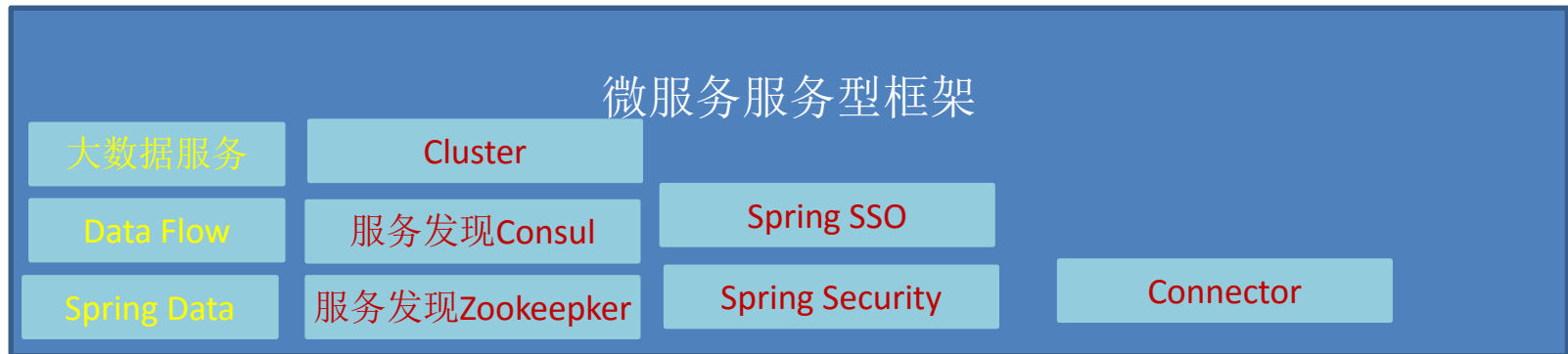
微服务架构



微服务框架



微服务框架和Spring微服务的对应



互联网应用架构--微服务渐成趋势

1月
1000万

THE WALL STREET JOURNAL THE CIO REPORT

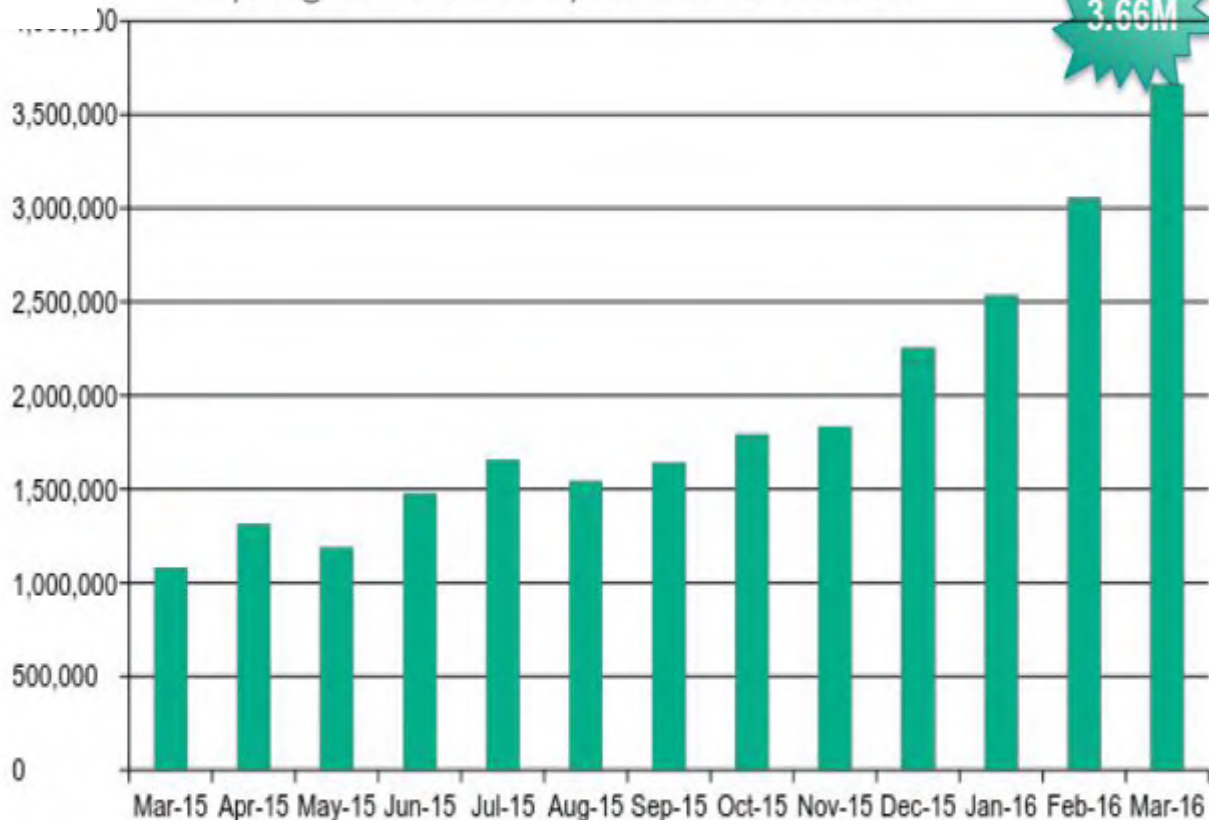
Innovation or Die: The Rise of Microservices
Oct 5, 2015



Spring Boot每月有1000万
的下载量

是微服务中最流行的框
架

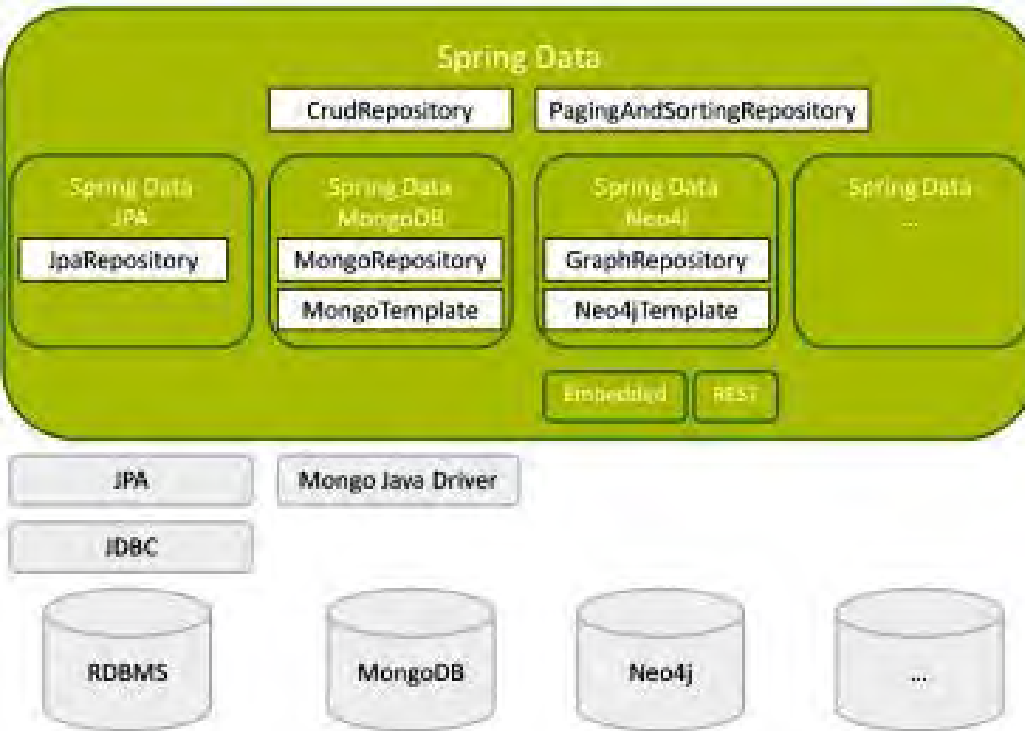
Spring Boot: Monthly Maven downloads



Source: oss.sonatype.org, via Pivotal

Spring Boot is the most popular ices
developer technology in the world

数据微服务化—Spring Data



Spring Data 对数据访问进行抽象，便于数据服务化，包括非关系数据库、Map-Reduce 框架、云数据服务等等；另外也包含对关系数据库的访问支持。

Spring Data 包含多个子项目：

Commons - 提供共享的基础框架，适合各个子项目使用，支持跨数据库持久化

Hadoop - 基于 Spring 的 Hadoop 作业配置和一个 POJO 编程模型的 MapReduce 作业

Key-Value - 集成了 [Redis](#) 和 [Riak](#)，提供多个常用场景下的简单封装

[CouchDB](#) 和 [MongoDB](#) 并提供基本的配置映射和资料库支持

Graph - 集成 [Neo4j](#) 提供强大的基于 POJO 的编程模型

Graph Roo AddOn - Roo support for Neo4j

JDBC Extensions - 支持 Oracle RAD、高级队列和高级数据类型

JPA - 简化创建 JPA 数据访问层和跨存储的持久层功能

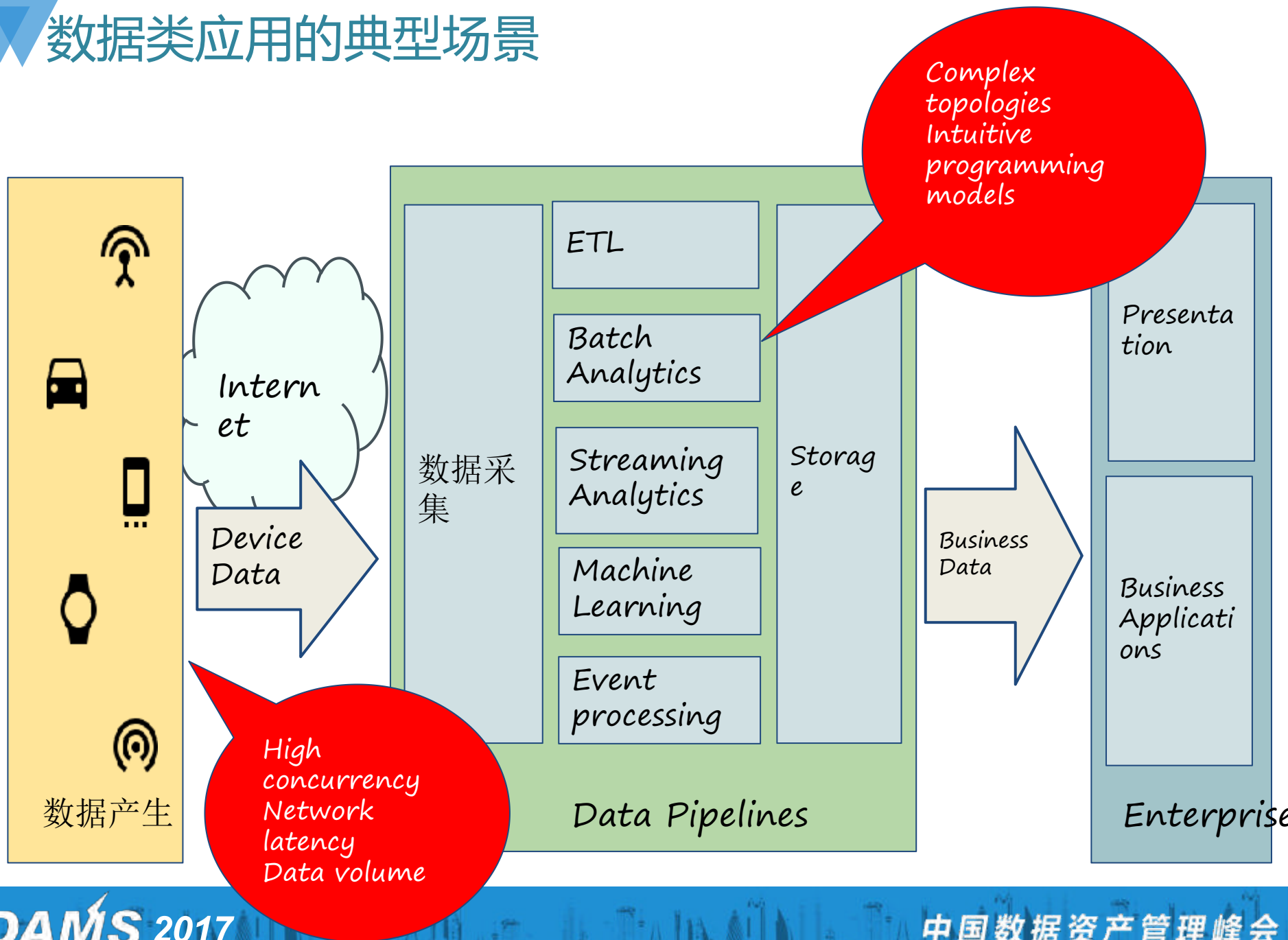
Mapping - 基于 Grails 的提供对象映射框架，支持不同的数据库

ElasticSearch—提供了对ElasticSearch的访问

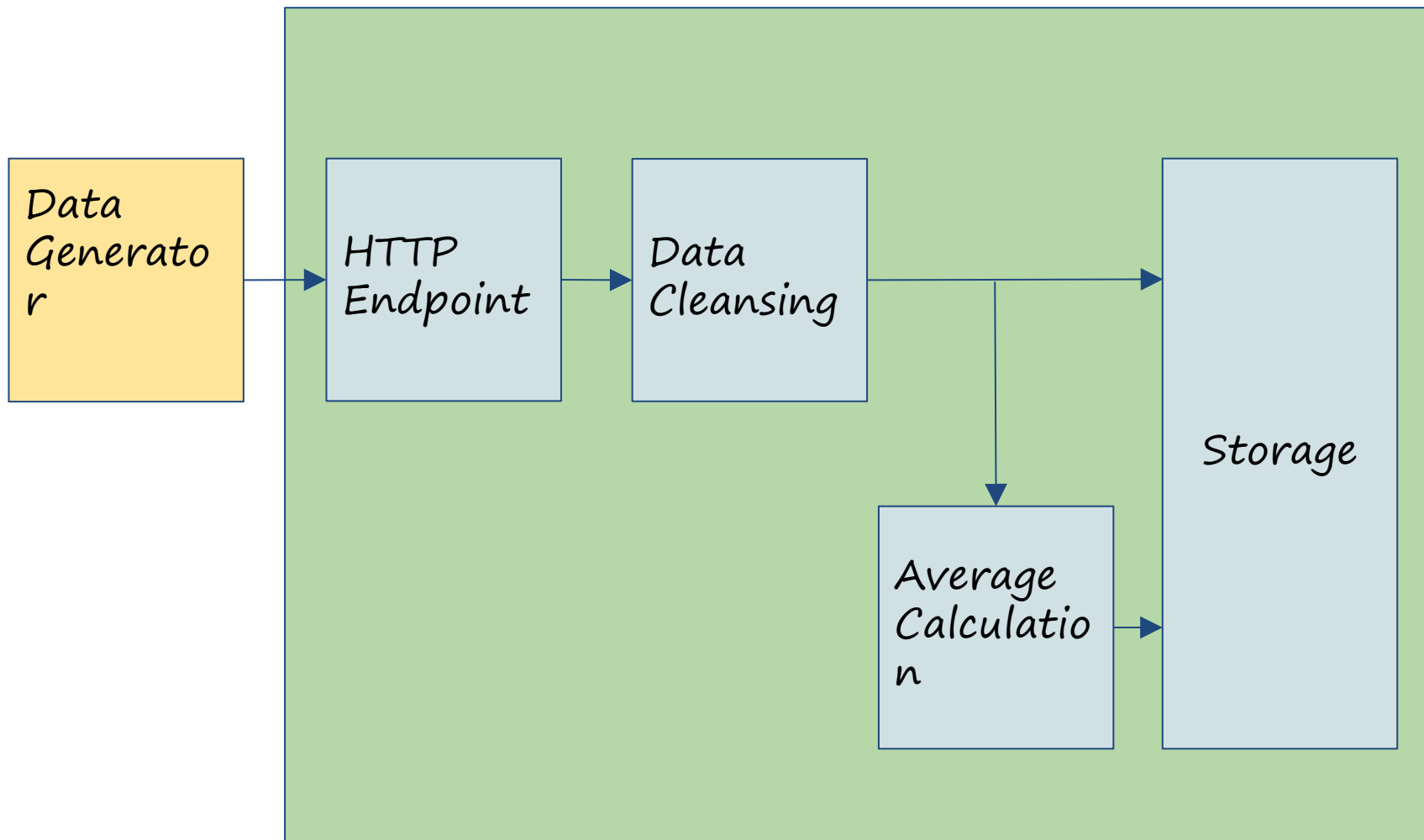
目录

- Pivotal和云原生/微服务理论
- Spring微服务
- Spring Cloud数据微服务

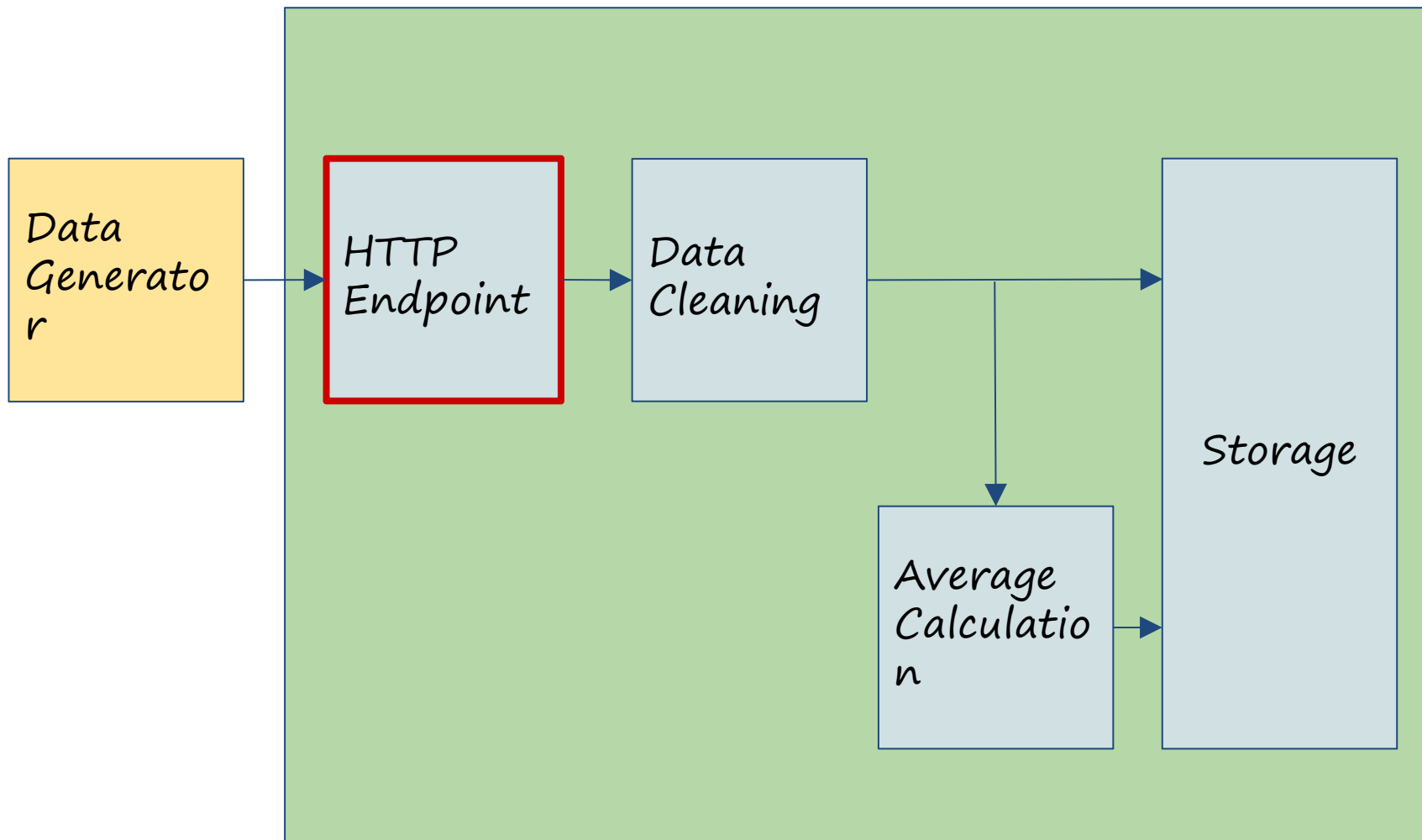
数据类应用的典型场景

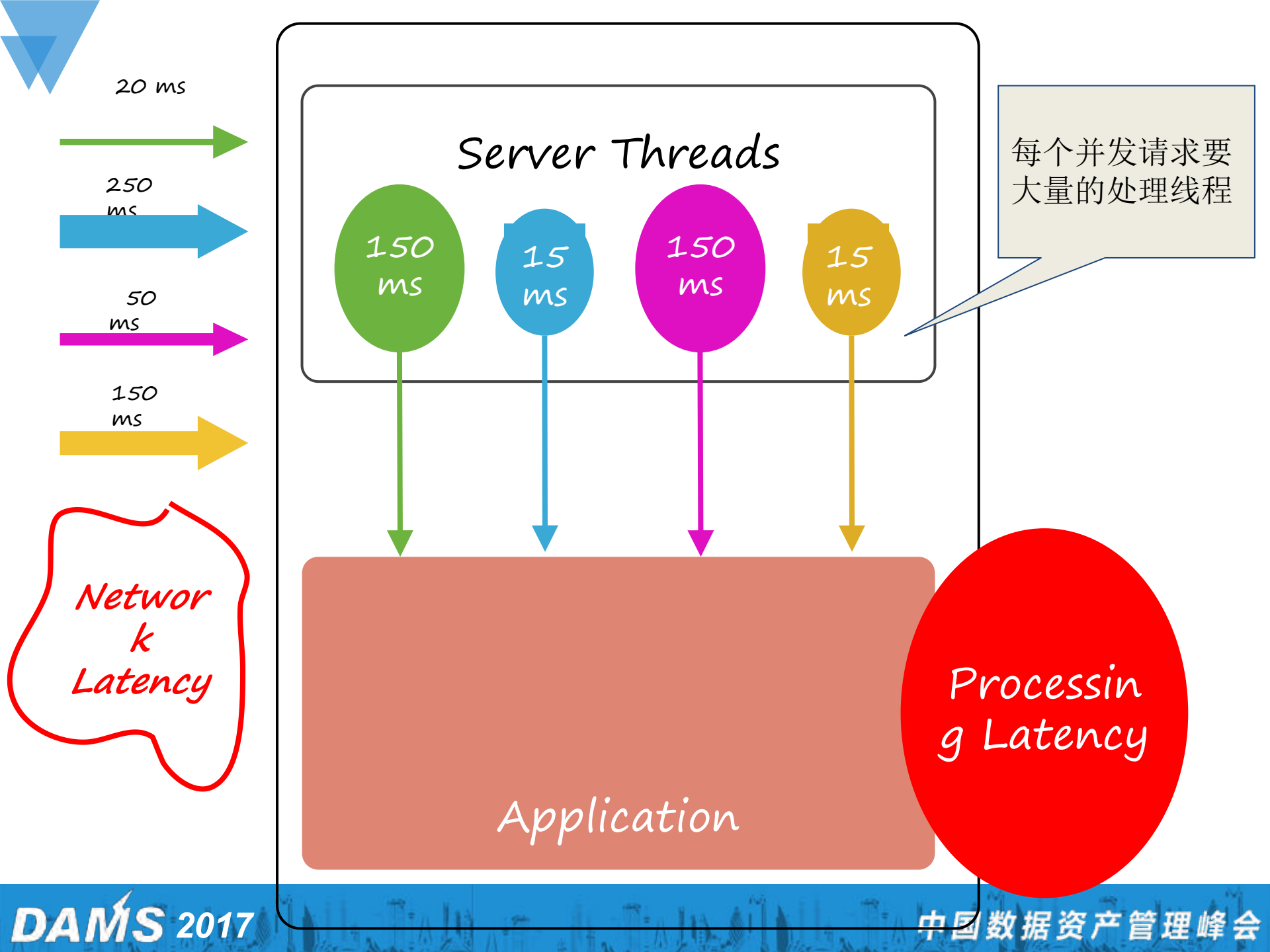


以一个典型的数据应用为例来说Spring数据微服务



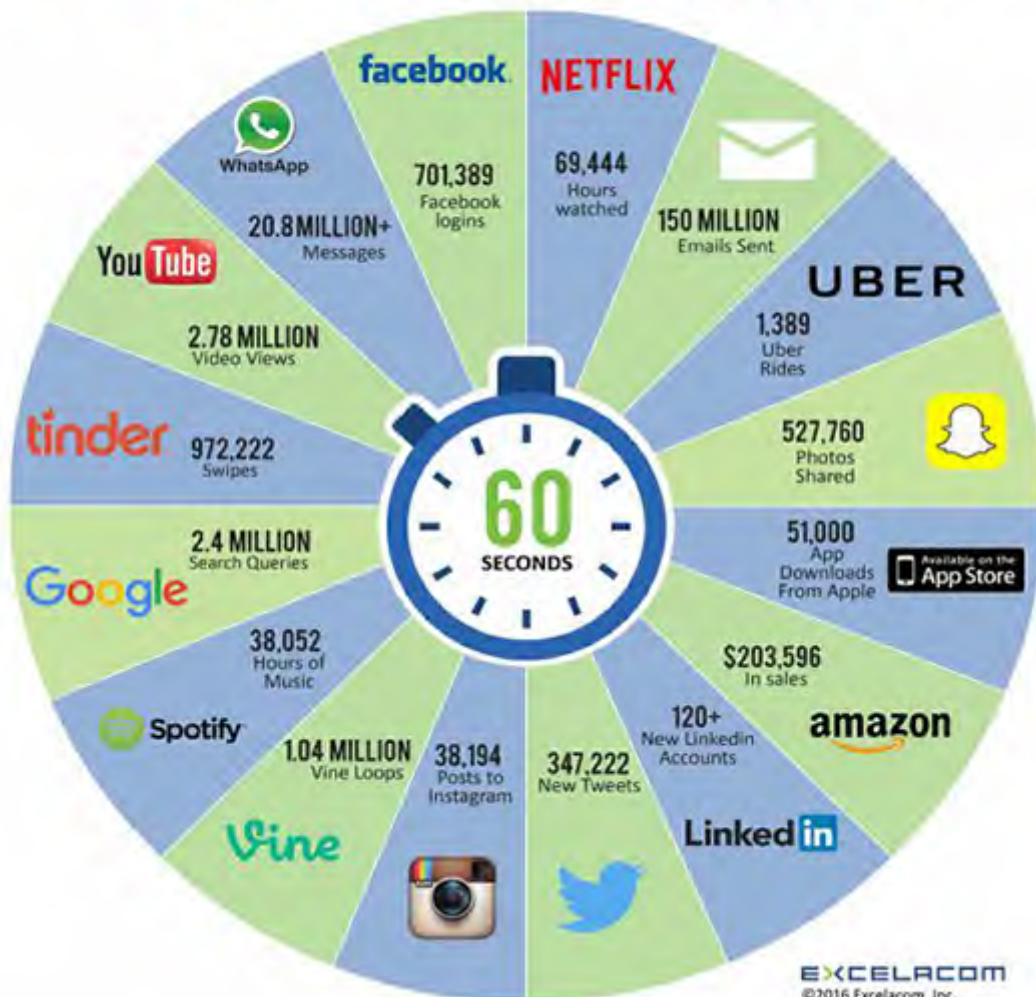
如何接受/采集数据



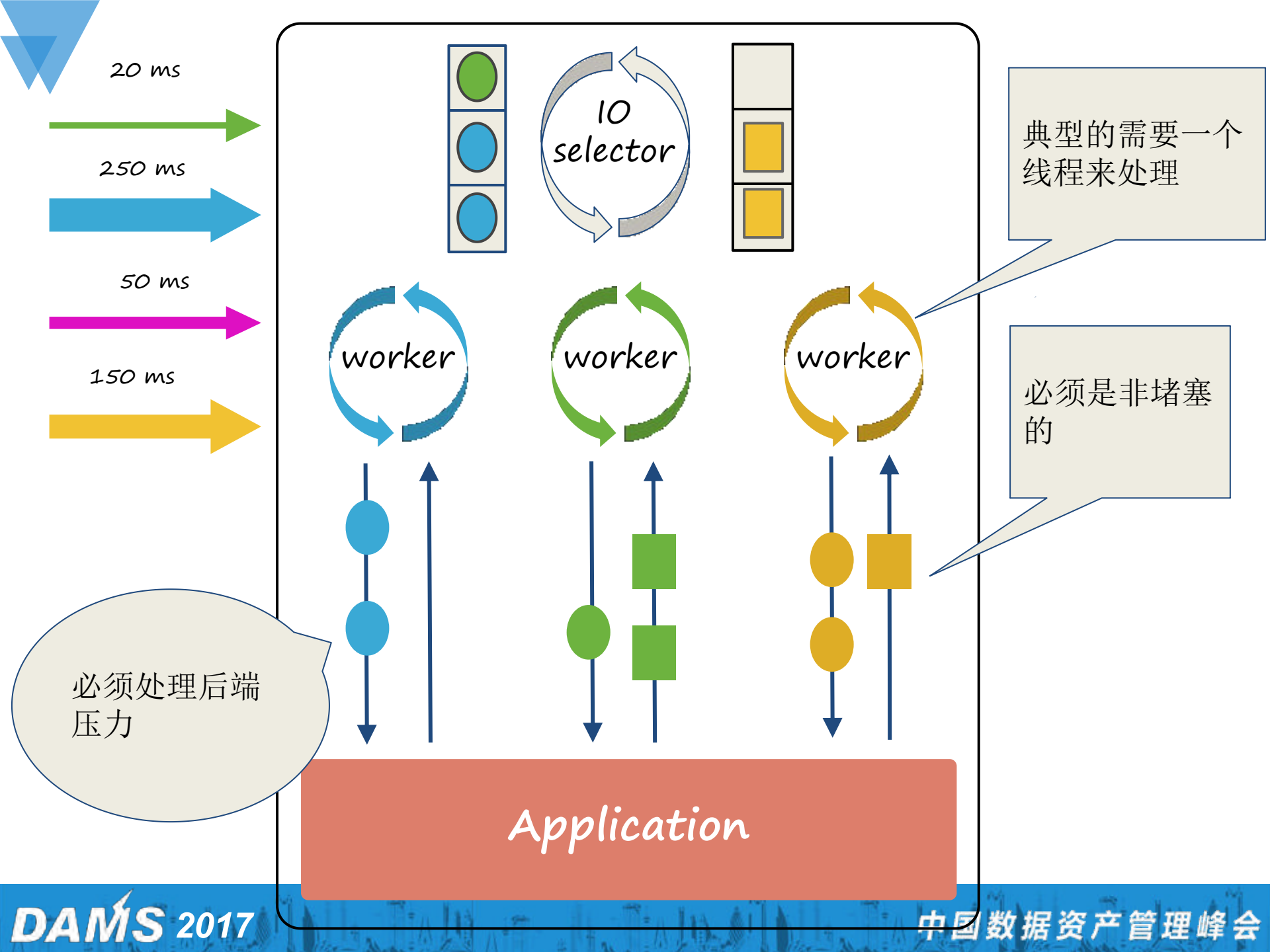


互联网的一分钟有多少数据发生

2016 What happens in an INTERNET MINUTE?



EXCELACOM
©2016 Excelacom, Inc.



典型的需要一个线程来处理

必须是非堵塞的

必须处理后端压力

Application

响应式编程(Reactor)项目

- 响应式和非堵塞机制
- 基于流的响应式 (也支持JDK 9)
 - 后端压力非堵塞式互操作标准
- 基于Java 8的API提供响应式编程的API
 - Functional programming model: map(), flatMap(), groupBy(), window()
 - Composability
- 针对TCP, Netty, Aeron, Kafka等扩展
- 主要的响应式编程机制
 - **Spring 5**, Spring Data, Spring Cloud Stream,...

Spring WebFlux in Spring 5

@Controller, @RequestMapping

Router Functions

spring-webmvc

spring-webflux

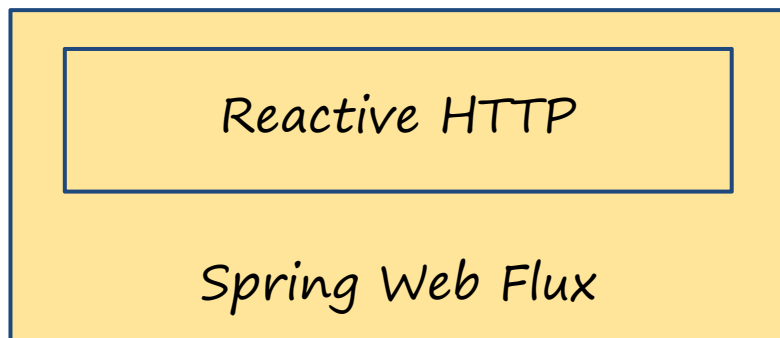
Servlet API

HTTP / Reactive Streams

Servlet Container

Tomcat, Jetty, Netty, Undertow

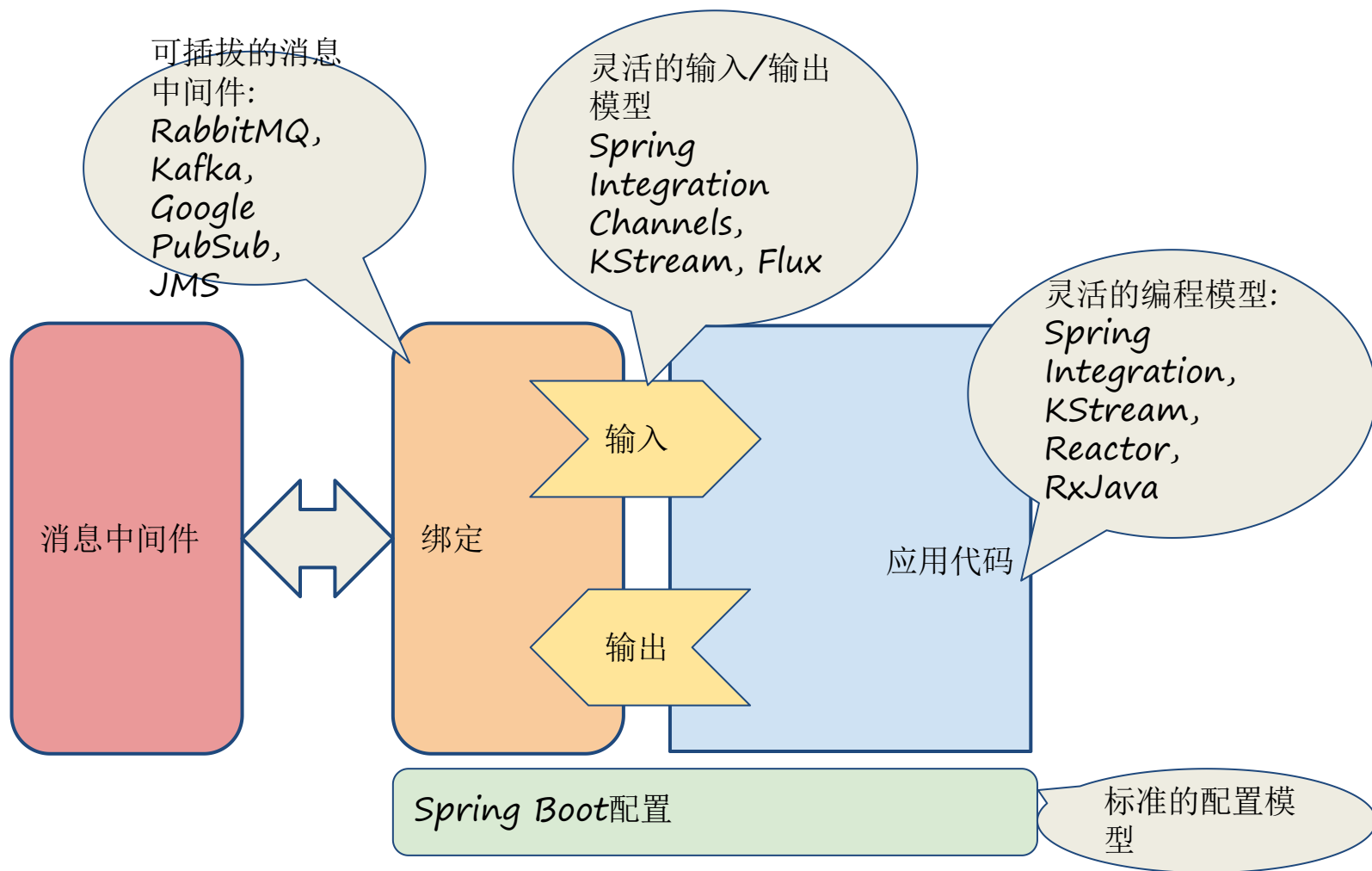
绑定HTTP终点



Spring Cloud Stream

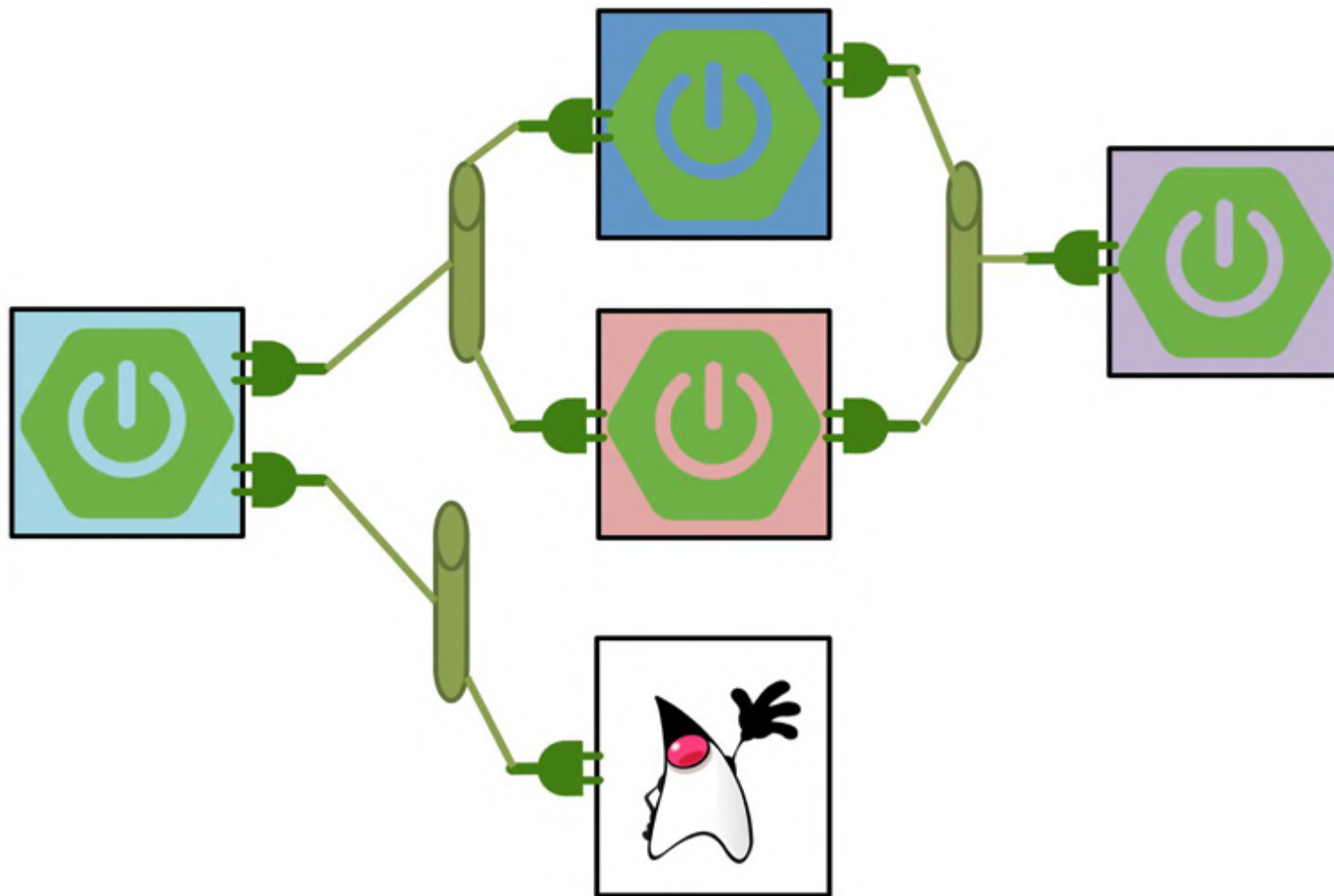
- 事件驱动的微服务框架
- 中间件只作为一个工具
- 可选的部署环境
- 基于以下Spring的产品构建
 - Spring Boot
 - Spring Integration - binder implementations, programming model
 - Reactor - Reactive API

Spring Cloud Stream的模型





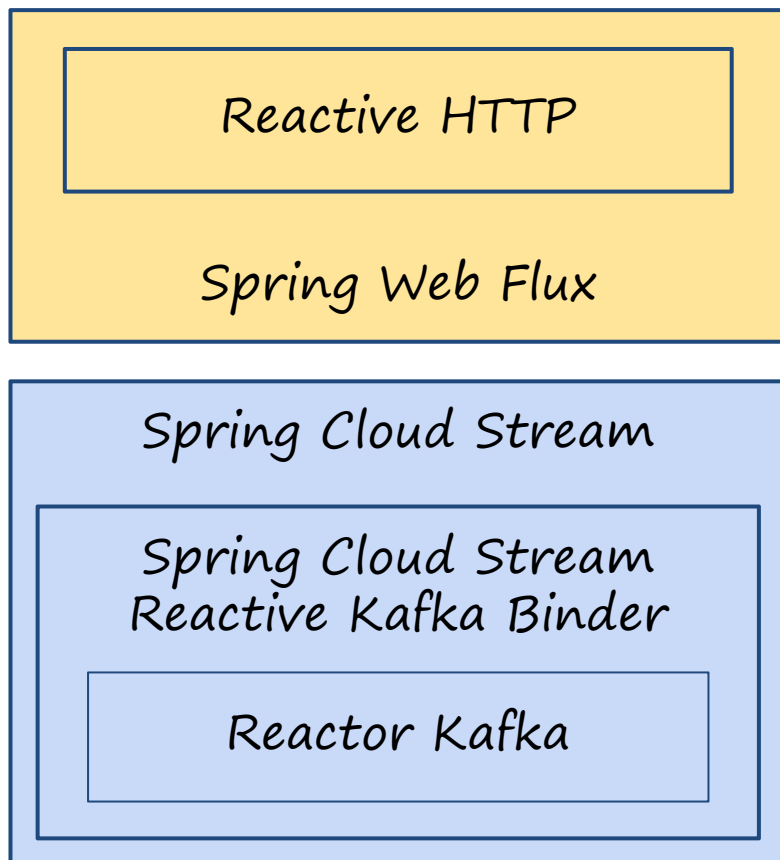
Spring Cloud Stream的编排模型



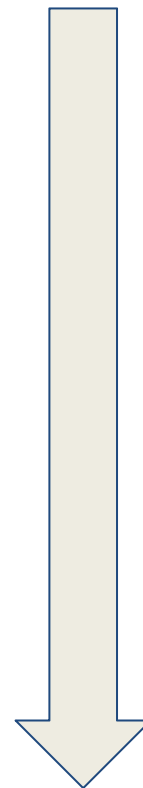
Spring Cloud Stream主要功能

- 持久的发布订阅消息机制
 - For easily creating complex topologies
- 消费者组
 - 支持弹性伸缩的多实例运行
- 可定义的数据分区
 - 在消费这实例上把相关数据共置
- 内容协商
 - 灵活的，自描述的序列化/反序列化策略

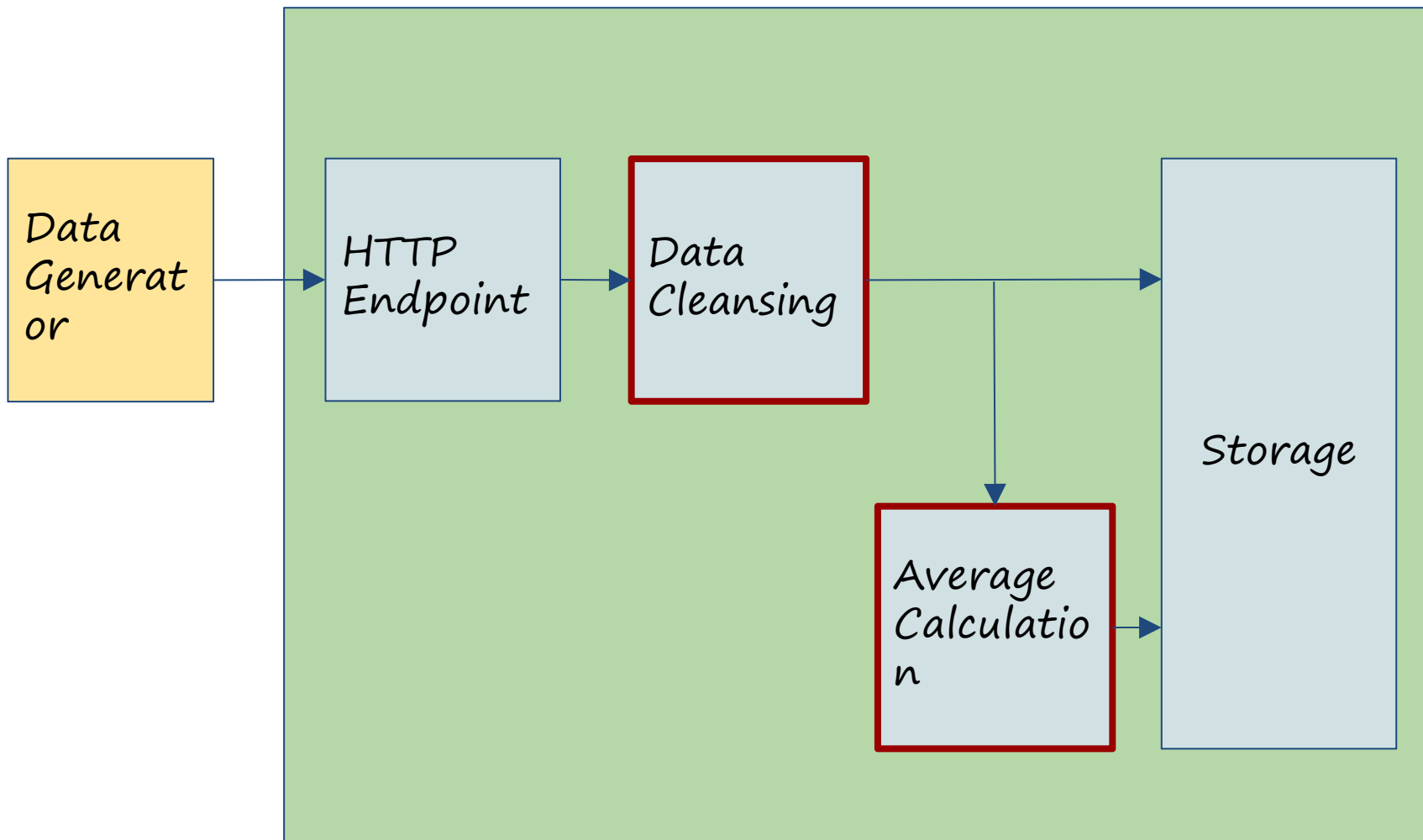
例：构建HTTP终点



端到端的响应式



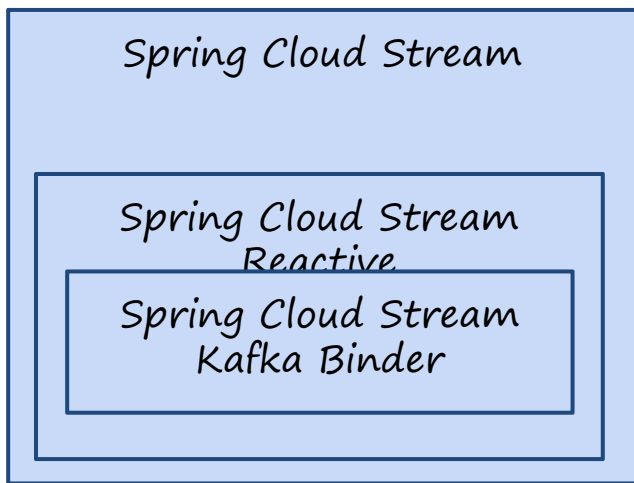
数据流例子—环节2 如何处理数据



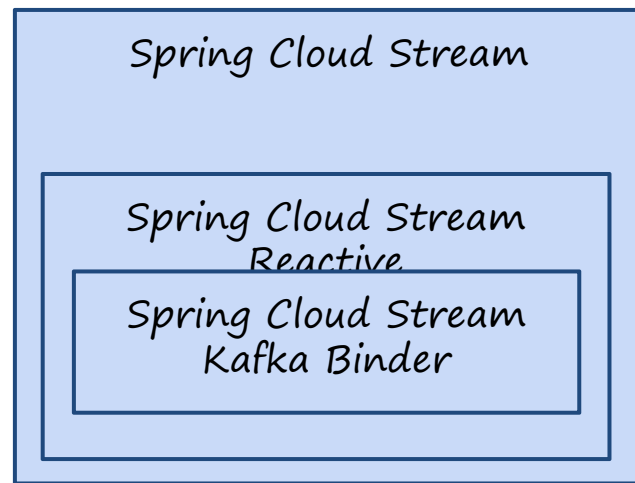
流处理的函数式编程

- 和Web终点的目标不一样
 - 不再太关注外部客户端、网络时延、资源利用率、后端压力等
- 流处理和事件处理不一样
 - 事件处理是事件模型，事件消息一般是相互独立处理的.
 - 流处理：更关注于消息组、处理顺序等.
- 功能编程
 - 主要是数据流的操作.
- Spring Cloud Stream的灵活性更易于采用
 - 针对传统消息机制的响应式编程适配
 - 全原生的响应式适配要求全响应式堆栈支持

构建处理管道



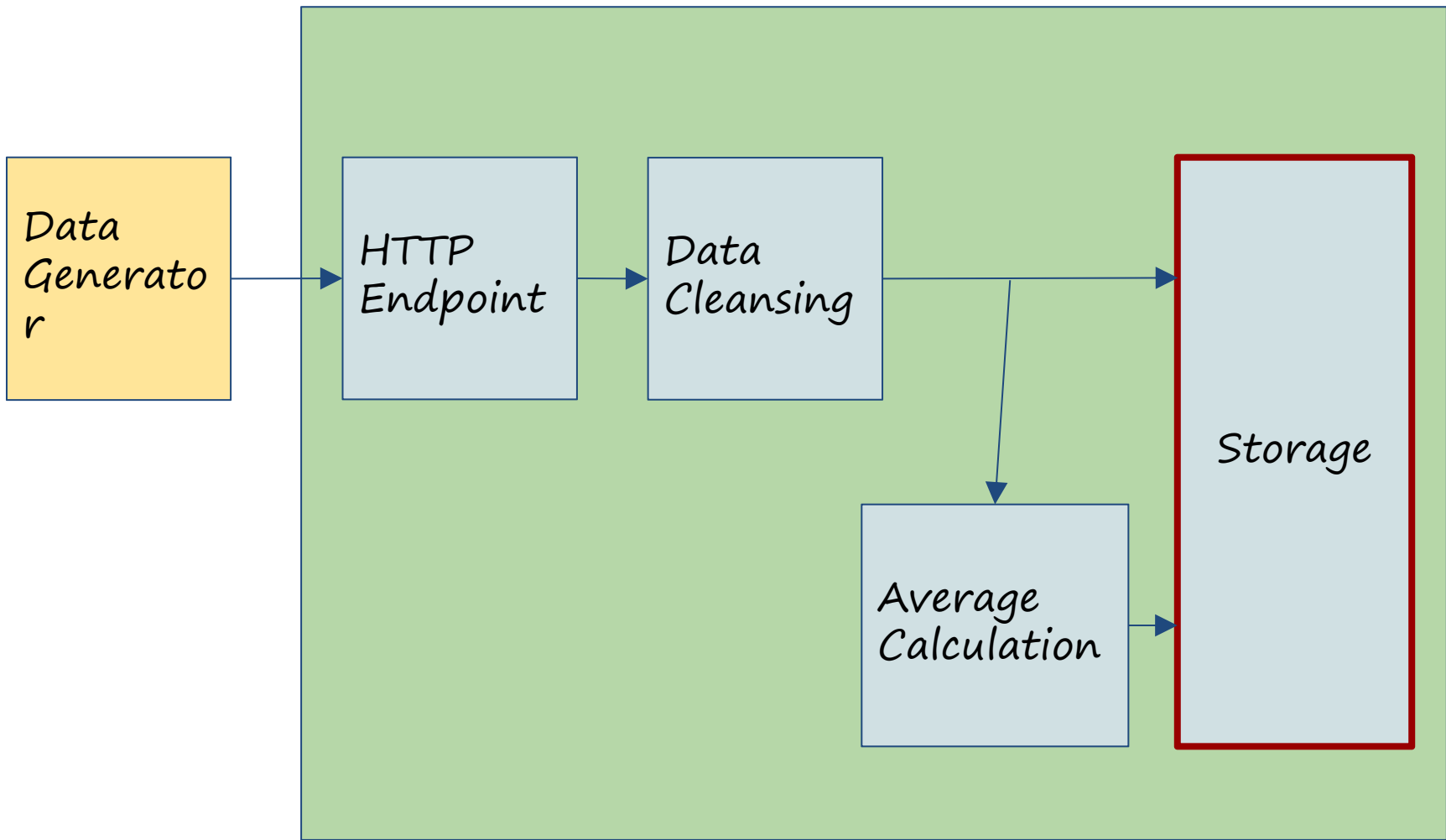
数据转换



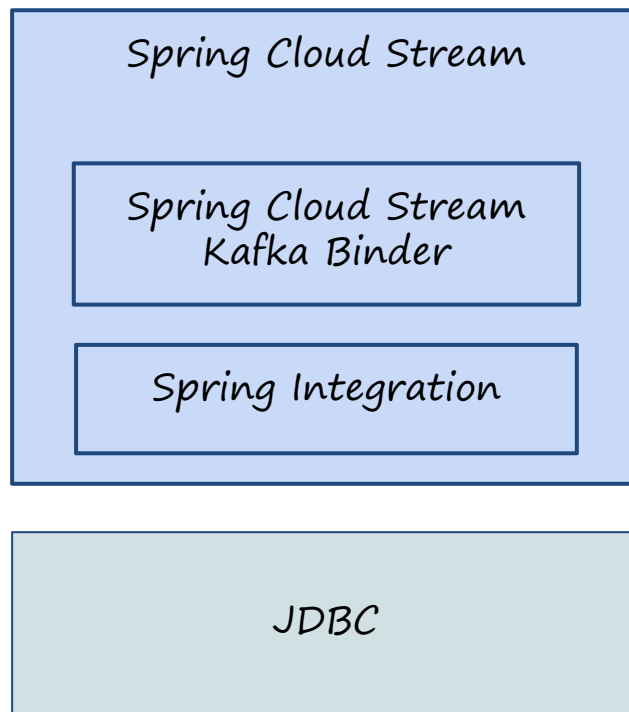
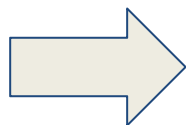
平均值计算



数据流例子—环节3 如何存储数据



构建JDBC落地点



JDBC Sink

Spring Cloud Stream:从命令式到反应式

定义式

Application

Message Channels

Spring Integration Binder
(RabbitMQ, Kafka, JMS,
Google PubSub)

响应式 函数编程
非响应式消息

Application

Reactive
Programming Model

Spring Cloud Stream Reactive
Adapter

Message Channels

Spring Integration Binder
RabbitMQ, Kafka, JMS,
Google PubSub)

全响应式堆栈

Application

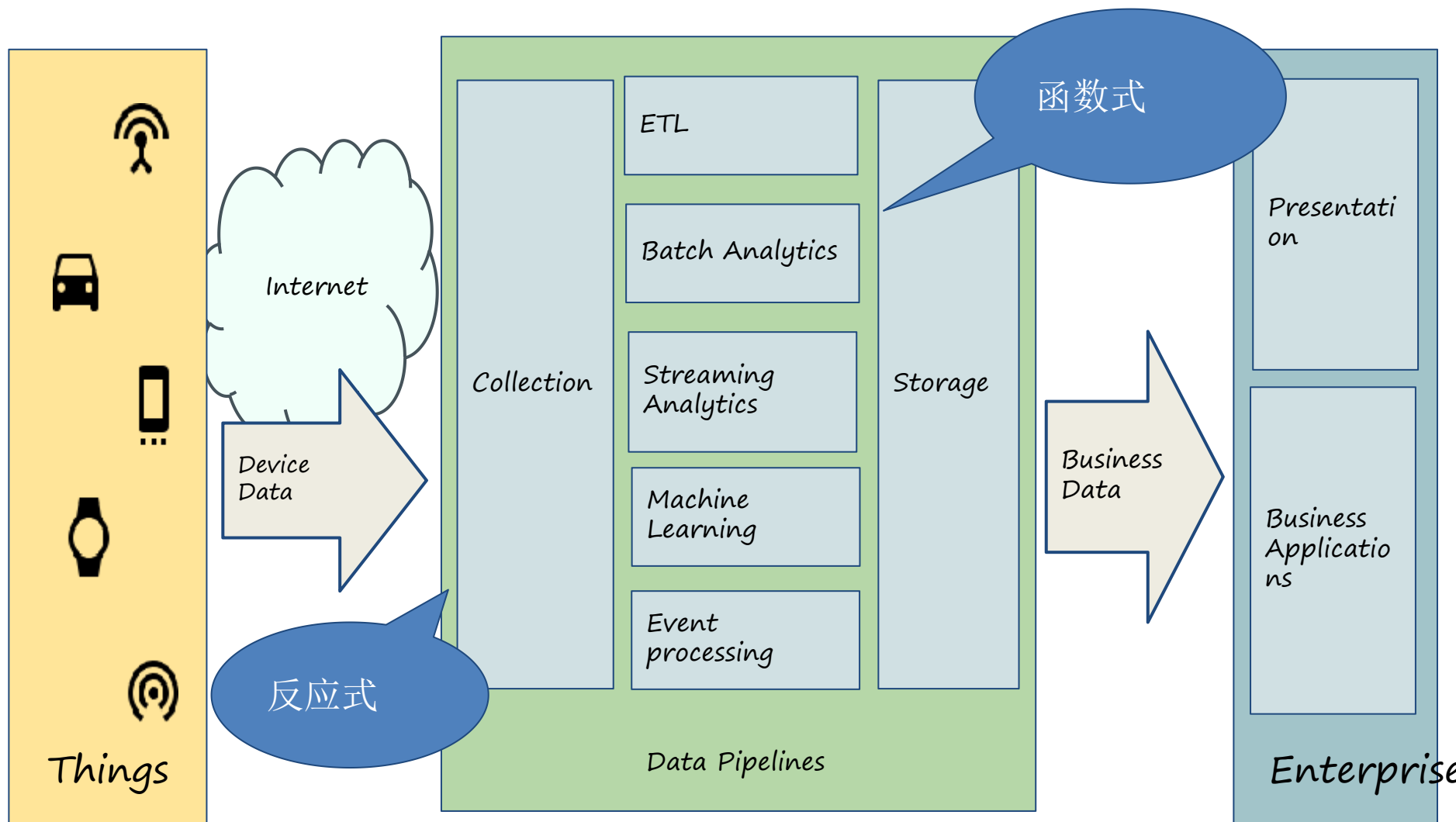
Reactive
Programming Model

Reactive API (Reactor,
RxJava)

Reactive Streams Binder
(>1.2)

Reactive Streams
Integration (Kafka)

总体架构图



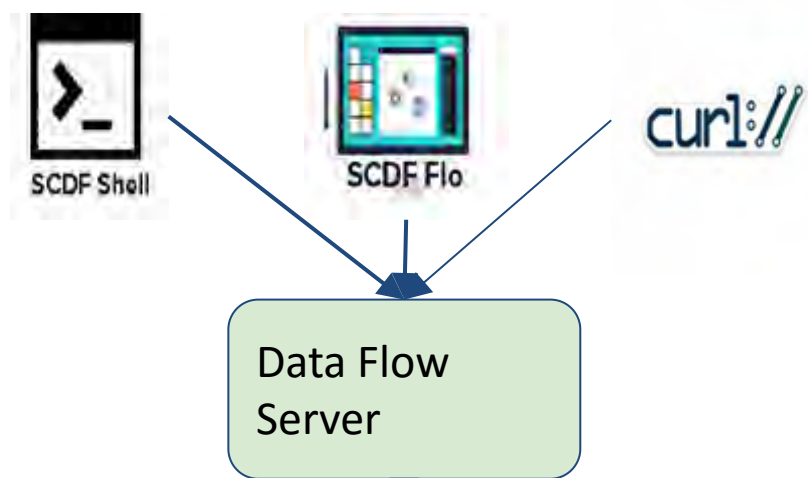
微服务的数据管道

- 专注于数据处理的、独立的、生产级的应用
- 通过消息中间件实现“轻量级”机制的通讯



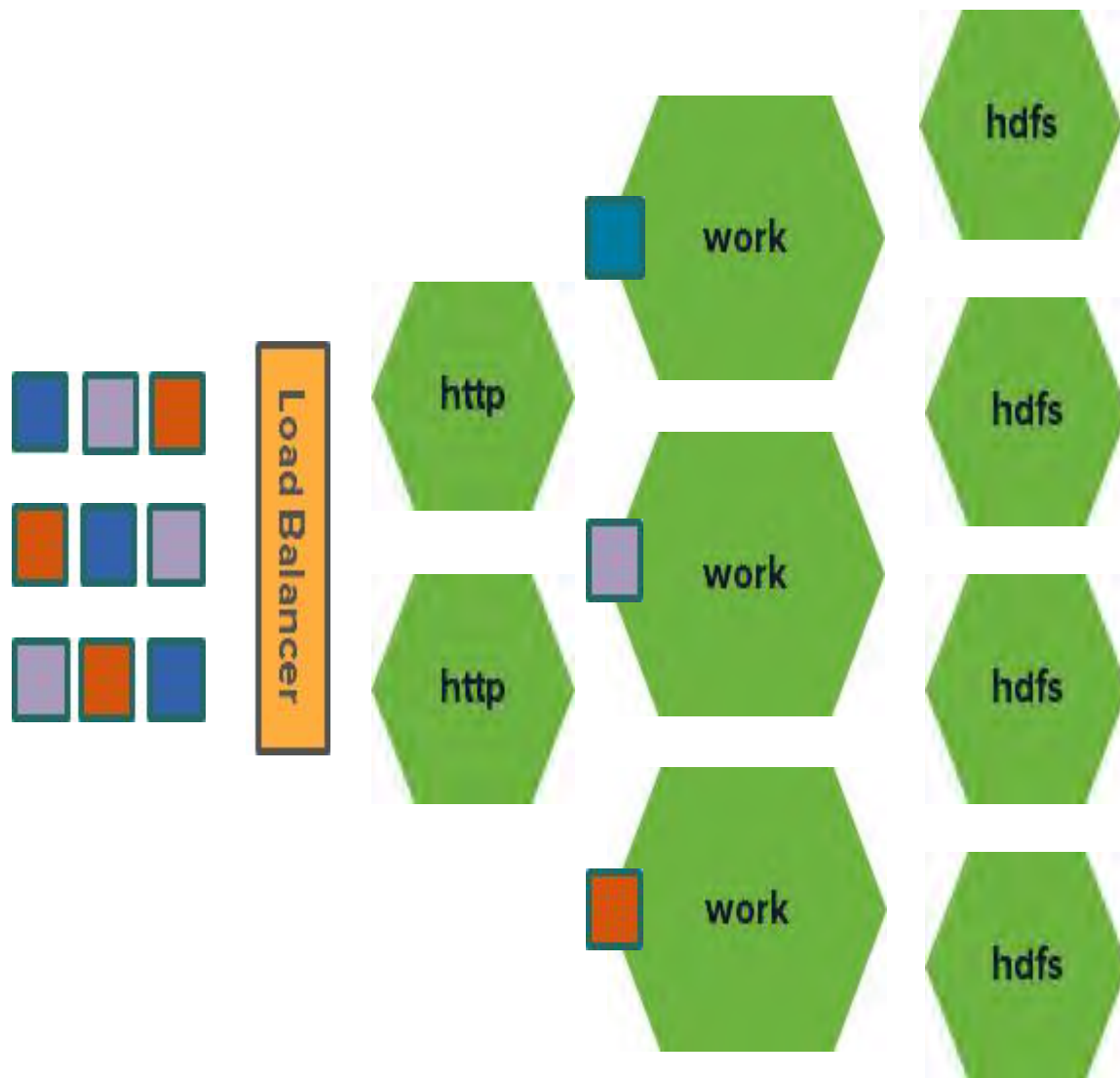
Spring Cloud Data Flow的云环境部署

数据微服务应用的编排，并且运行在现代的运行时上(容器)
应用场景：数据处理、流处理、批处理





Deployment清单模式

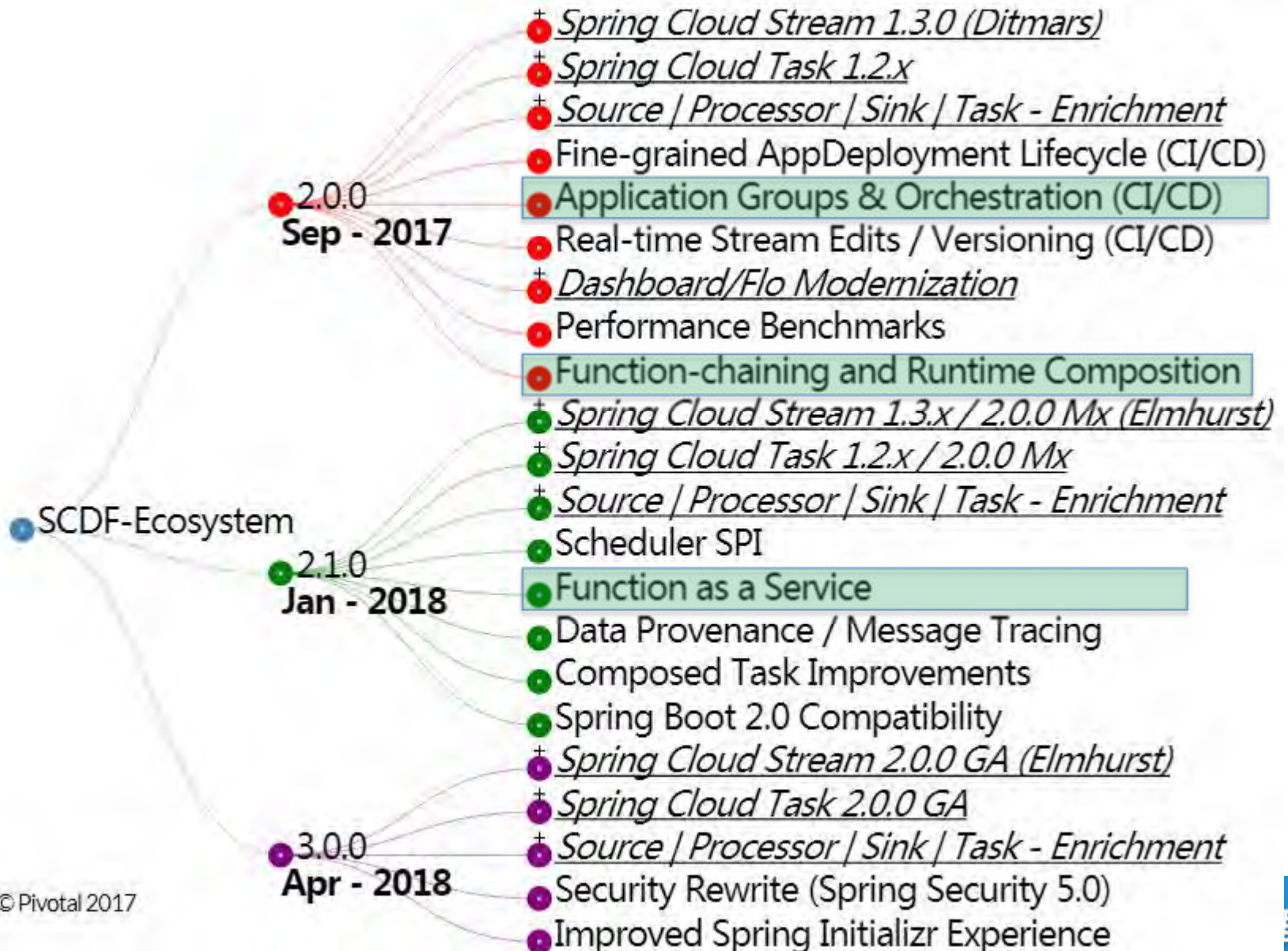


SCDF的现有处理节点

Sources Processors Sinks Tasks

mqtt[1.3]	tensorlow[1.3]	mqtt[1.3]	hdfsjdbc[1.3]
hdfs[1.3]	tensorflow-twitter[1.3]	rich-gauge[1.3]	s3jdbc[1.3]
twittersearch[1.3]	tensorflow-image-recognition[1.3]	google pub/sub[1.4]	filepollhdfs[1.4]
google pub/sub[1.4]	py-script[1.3]	hbase[1.4]	gpload[1.4]
kinesis[1.4]	py-http[1.3]	solr[1.4]	hdfsmongodb[1.4]
netty-tcp[1.4]	kstream[1.3]	kinesis[1.4]	ftphdfs[1.4]
google-services[1.4]	(improvement) pmml[1.4]	elasticsearch[1.4]	
azure-services[1.4]	r-pr[1.4]		
coap[1.4]			

发展规划





- 问？答！



DAMS

中国数据资产管理峰会

CHINA DATA ASSET MANAGEMENT SUMMIT

THANK YOU !

