

基于微服务的 企业应用架构设计范式

SPEAKER

朱 江



关注InfoQ官方信息
及时获取QCon软件开发者
大会演讲视频信息

ArchSummit
全球架构师峰会 2016

[北京站] 2016年12月2日-3日
咨询热线: 010-89880682

QCon
全球软件开发大会

[北京站] 2017年4月16日-18日
咨询热线: 010-64738142



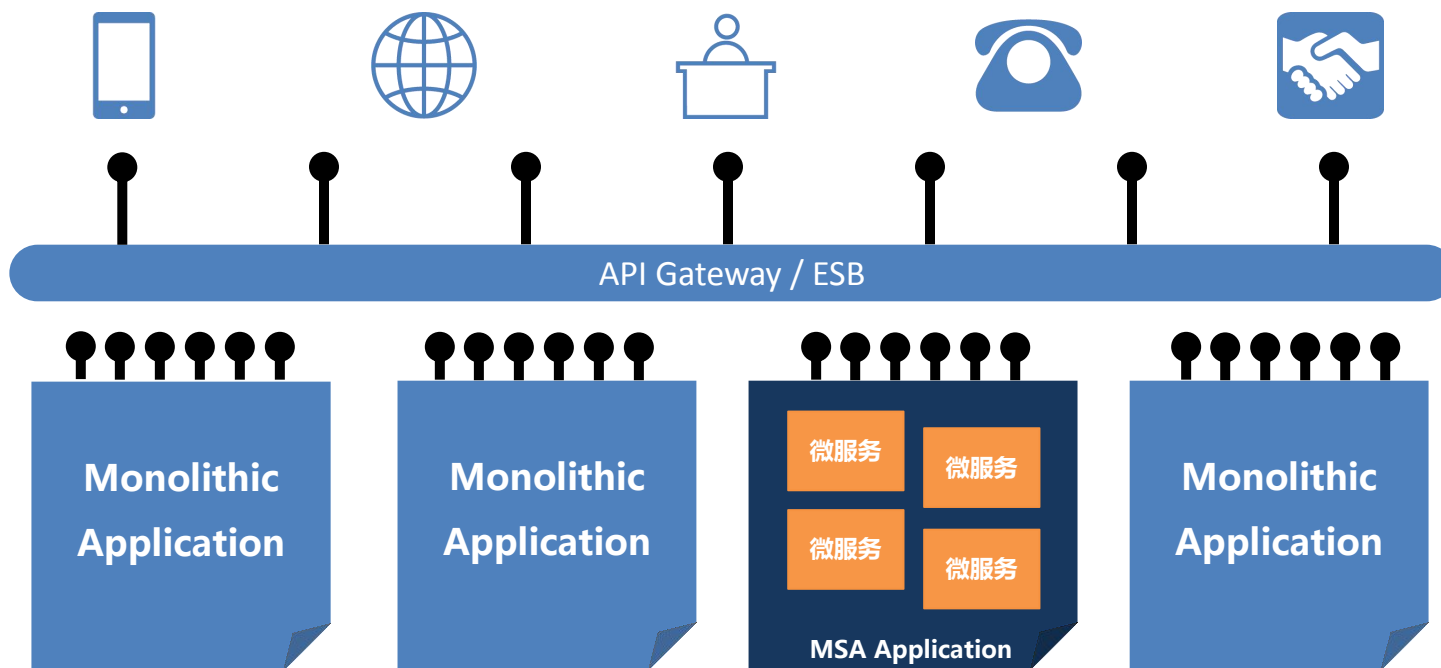
朱江

中国

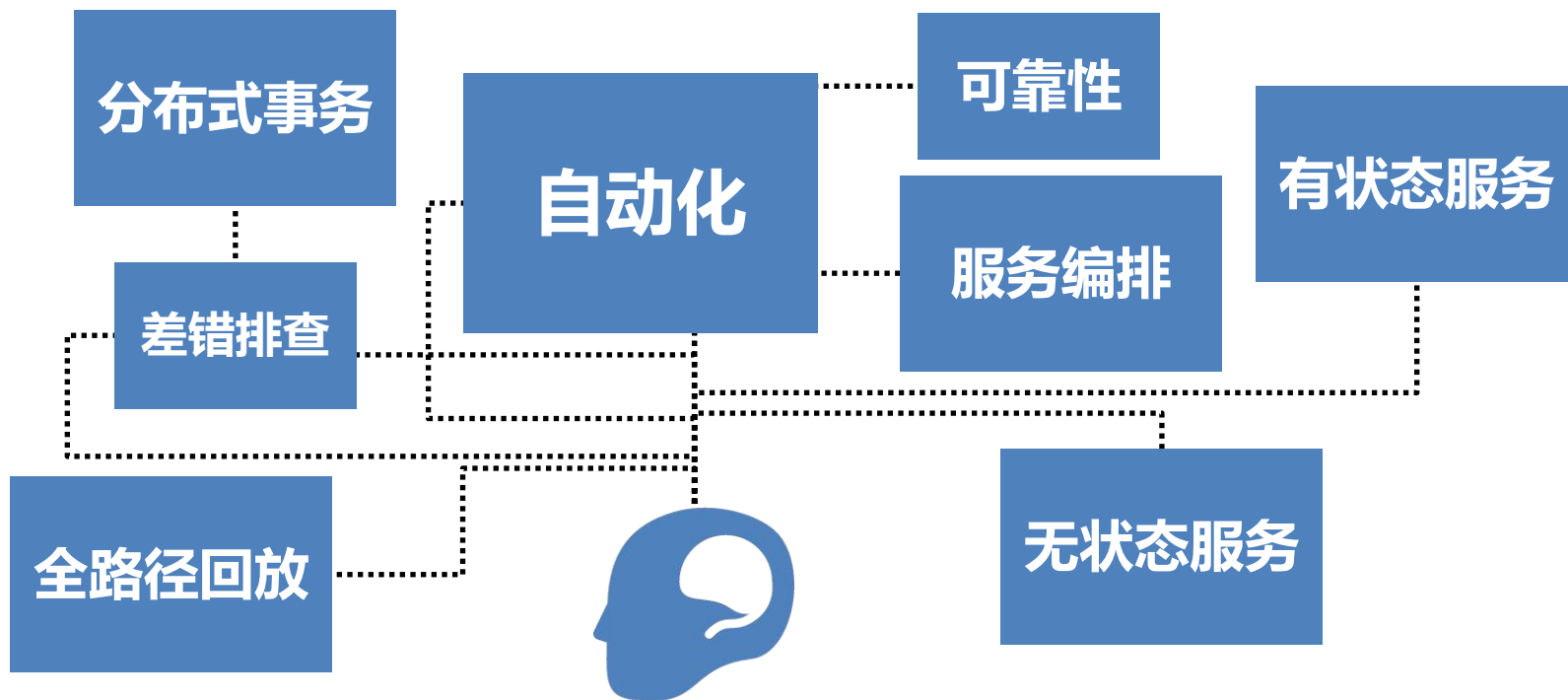


扫一扫上面的二维码图案，加我微信

EGO EXTRA GEEKS' ORGANIZATION
NETWORKS



实施微服务的技术困难与挑战



实施微服务架构的两大误区

1.混淆远程服务与本地服务的区别

2.试图将远程服务与本地服务的调用透明化

微服务应用架构八大设计范式

一

同步记录
业务流水

二

流水号的
GAIR模式

三

元数据驱动的
服务定义

四

同步模式
异步化

五

进程间服务
无状态

六

保证最终
数据一致性

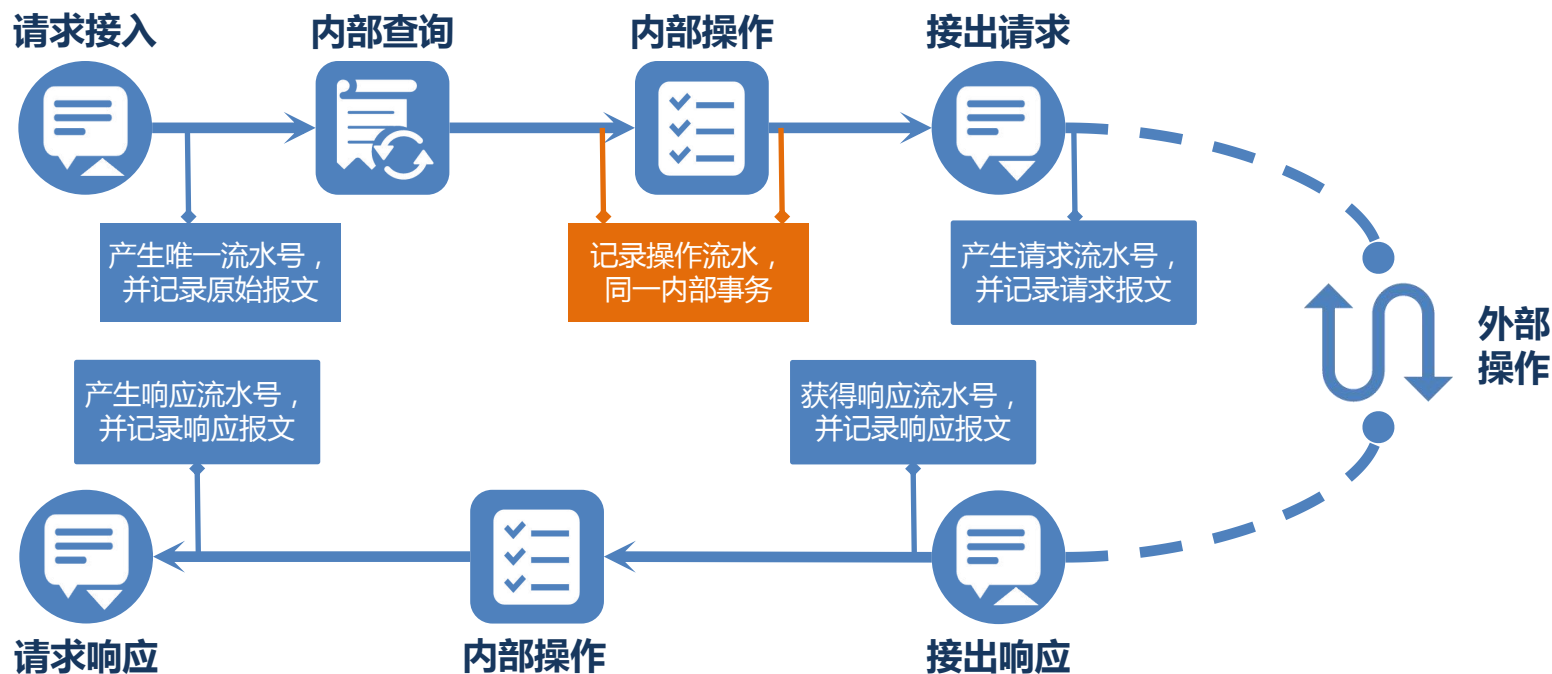
七

用编排实现微
服务组合

八

业务配置
集中管理

一、采用同步方式记录业务流水



二、流水号设计的GAIR模式

G

Global_ID , 全局流水号 , 需保证全局唯一 , 用于标识一次完整的端到端交易请求

A

Answer_ID , 响应流水号 , 业务域内唯一 , 用于标识一次响应

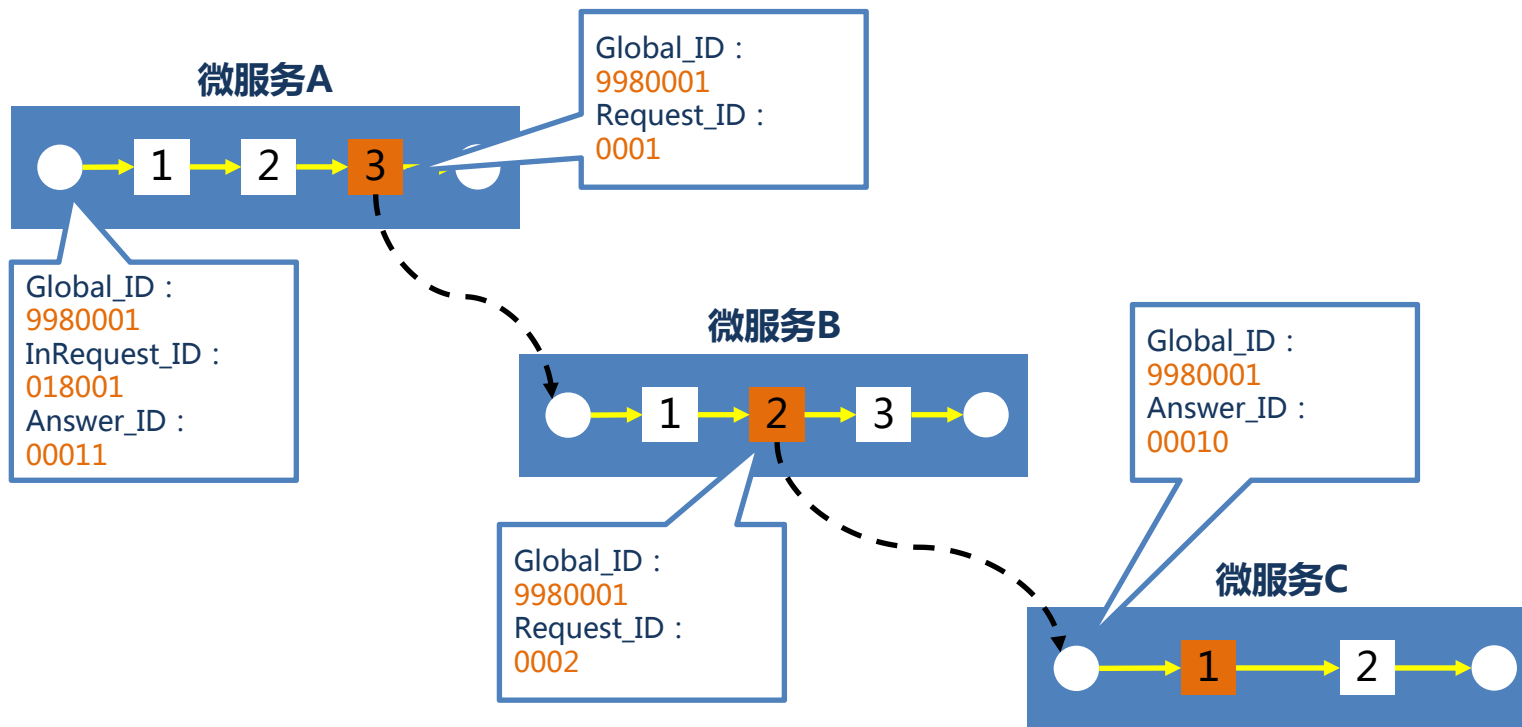
I

InRequest_ID , 前台流水号 , 业务域内唯一 , 用于标识一次接收到的交易请求

R

Request_ID , 请求流水号 , 业务域内唯一 , 用于标识一次自身发起的交易请求

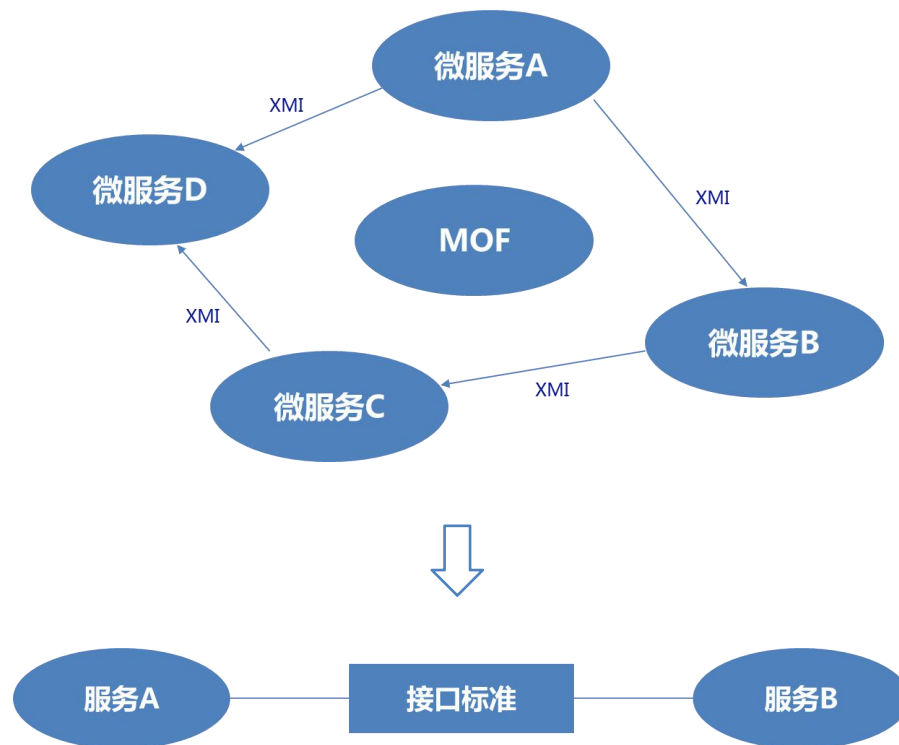
流水号设计的GAIR模式



三、元数据驱动的微服务定义

机器可读，
全面自动化的前提条件

标准统一，
打通应用交付各个环节



元数据驱动的微服务定义

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://
  www.primeton.com/btp/cfg" xsi:schemaLocation="http://www.primeton.com/btp/cfg
  common.xsd" author="sundy" category="message" create-date="2014-12-08 17:01:13"
  version="7.0.0.0">
3   <message-definition bitmap-format="Binary" bitmap-length="0" bitmap-type="
  unionpay" category="json" filter-null="false" id="BBB_DeleteRES" name="
  ResponseMessage" namespace="BankXT_BasicBusiness.BBB_DeleteRES" orderly="false"
  xml-pack-simple-node="false" xml-pack-statement="true">
4     <message-item align="right" bitmap-field-filled-mode="left" bitmap-field-length
      ="0" bitmap-field-type="fixedLength" bitmap-lenfield-length="0"
      bitmap-lenfield-type="string" bitmap-position="0" dict-id="anyString"
      display-name="" field-type="normal" is-recheck="false" is-required="false"
      is-secure="false" length="0" modify-value="0" name="RespCode" pad-char="0x20"
      seqno="0" validate-rule="" value-mode="normal" xml-field-type="VALUE">
5       <ext-property/>
6       <description></description>
7     </message-item>
8     <message-item align="right" bitmap-field-filled-mode="left" bitmap-field-length
      ="0" bitmap-field-type="fixedLength" bitmap-lenfield-length="0"
      bitmap-lenfield-type="string" bitmap-position="0" dict-id="anyString"
      display-name="" field-type="normal" is-recheck="false" is-required="false"
      is-secure="false" length="0" modify-value="0" name="RespMsg" pad-char="0x20"
      seqno="1" validate-rule="" value-mode="normal" xml-field-type="VALUE">
9       <ext-property/>
10      <description></description>
11    </message-item>
12  </message-definition>
13 </configuration>
```

结构定义

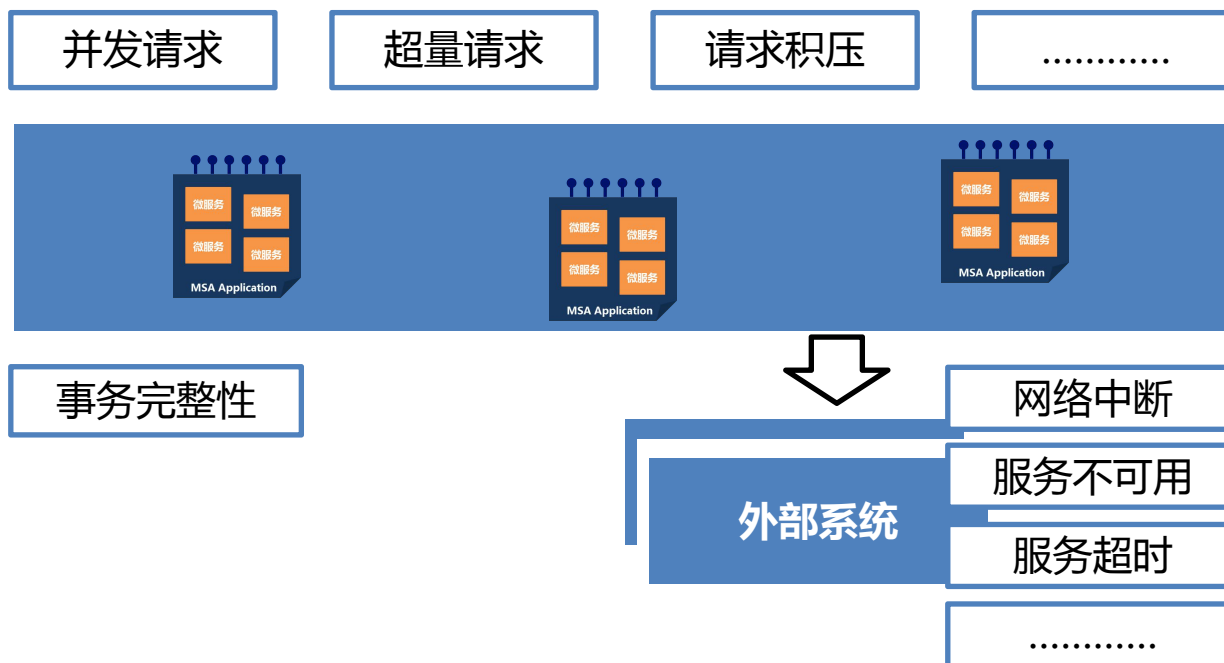
数据标准

校验规则

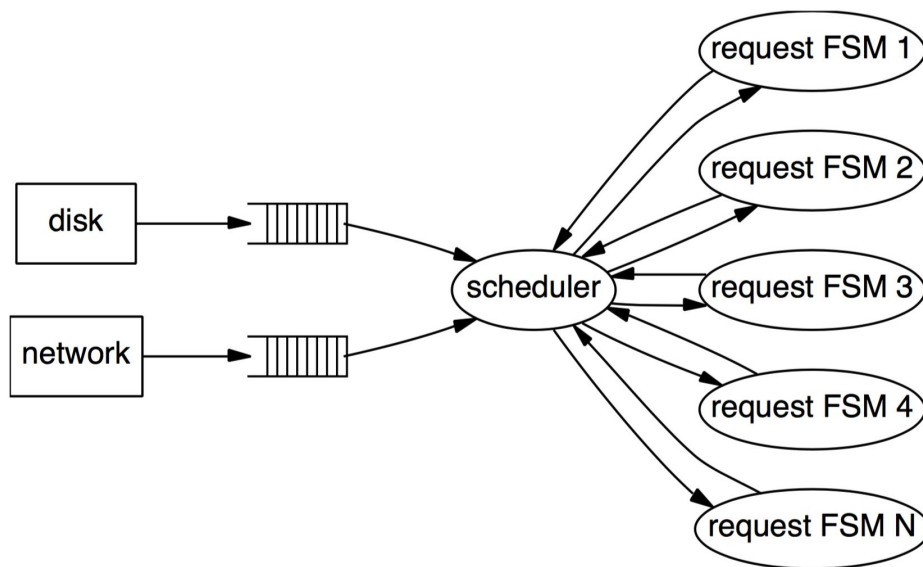
值转换规则

四、同步模式异步化

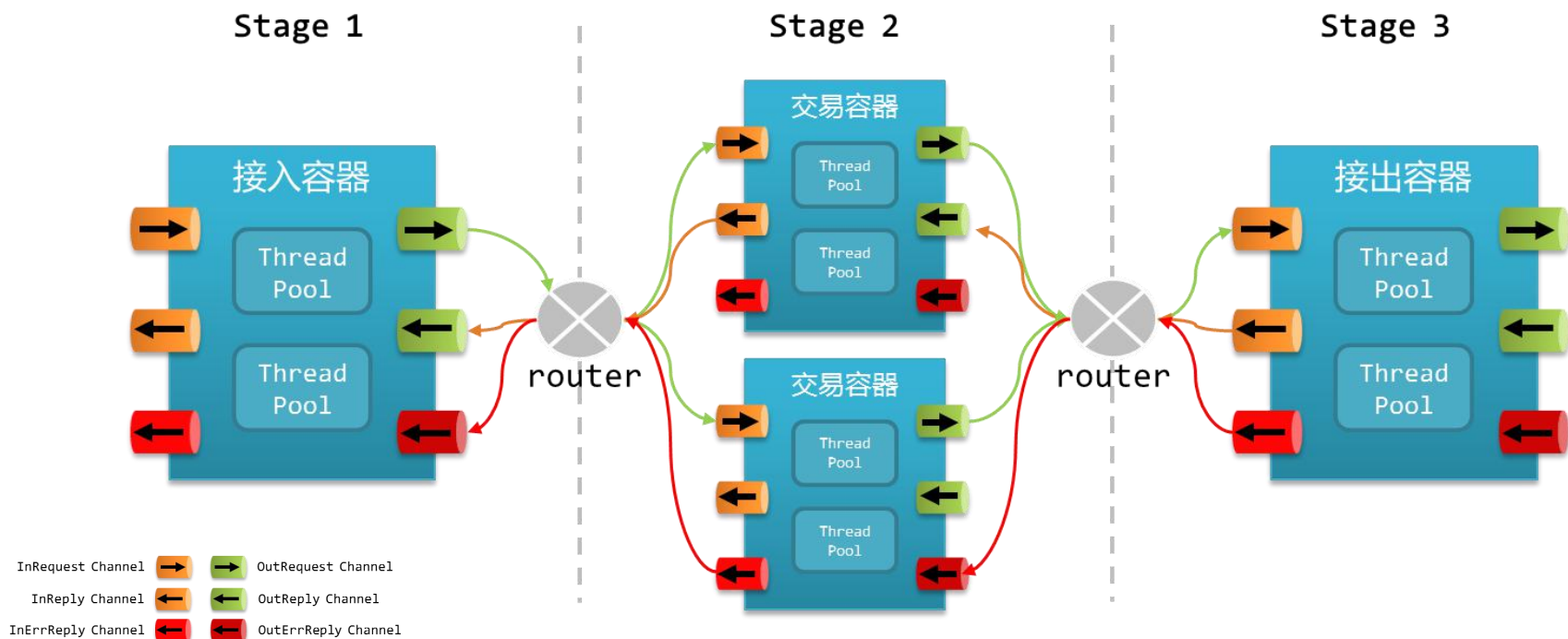
如何在**不确定**的环境中交付**确定**的服务？



传统基于事件的异步模式



分段事件驱动（SEDA）模式



五、进程间服务无状态



什么是状态？

如果一个数据需要被多个服务共享，才能完成一笔交易，则这个数据被称为状态。

依赖这个数据的服务被称为有状态服务，反之称为无状态服务

如何通过无状态的技术手段实现有状态的业务？

四种手段

发起方传递状态

粘滞+复制

状态共享-缓存

状态共享-持久



时间窗口

性能

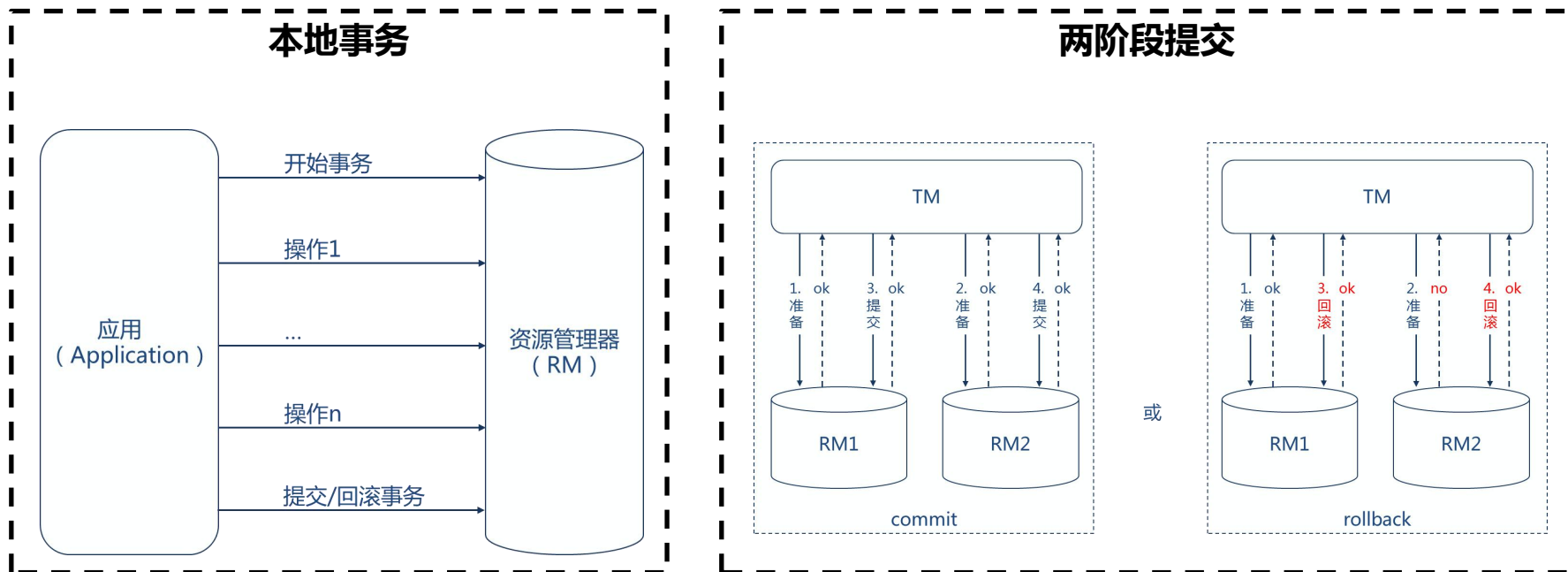
可靠性

安全性

四个维度

六、保证最终数据一致性

传统的事务处理方式无法适应微服务架构

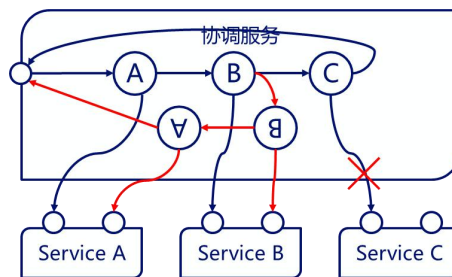
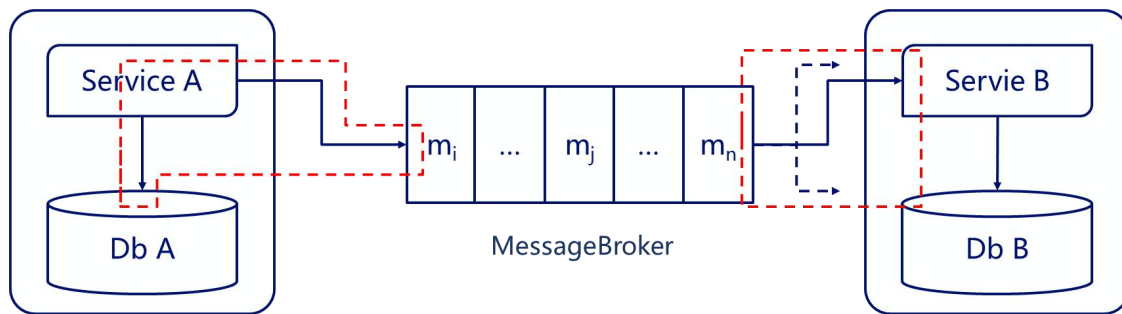


保证最终数据一致性

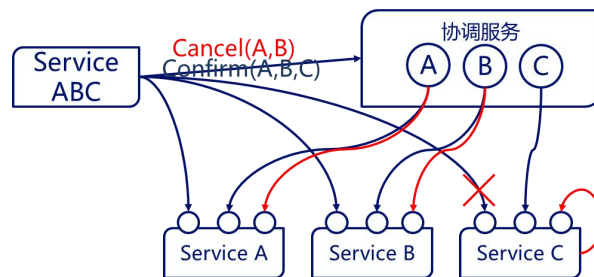
最终一致性

是指系统中的所有数据副本
经过**一定时间**后，
最终能够达到一致的状态

可靠事件模式



补偿模式



TCC模式

七、用编排实现微服务组合

配置化编排

```
<?xml version="1.0" encoding="UTF-8"?>
<process:tBusinessLogic xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:model="http://com.primeton.emf.core" xmlns:process="http://primeton.com/emf/core/process" name="BBB_OverviewSearch.process_btx" detailDescription="" demonstration="" urls="" gridVisibility="false" rulerVisibility="true" snapToGeometry="true" productVersion="7.0.0.0">
  <nodes xsi:type="process:tStart" id="start0" name="Start" displayName="开始" type="start">
    <sourceConnections xsi:type="process:tlink" id="link6" name="link6" displayName="连接线" isDefault="true" type="transition">
      <sourceNode>start0</sourceNode>
      <targetNode>invokePojo0</targetNode>
      <process:transitionCondition/>
    </sourceConnections>
    <nodeLabel>start0label</nodeLabel>
  </nodes>
  <nodes xsi:type="process:tEnd" id="end0" name="End" displayName="结束" type="end">
    <targetConnections>link9</targetConnections>
    <targetConnections>link10</targetConnections>
    <targetConnections>link11</targetConnections>
    <nodeLabel>end0label</nodeLabel>
    <process:returns>
      <process:return id="0" language="BankXT_BasicBusiness.BBB_OverviewSearchRES.BBB_OverviewSearchRES" name="outMessage" type="query" valueType="DataObject">outMessage</process:return>
    </process:returns>
  </nodes>
</topRuler/>
<leftRuler/>
<process:info author="sundy" createTime="2015-01-04 14:34:36" date="2015-01-04Z" description="" name="BBB_OverviewSearch" version="7.0.0.0"/>
<process:variables>
  <process:variable description="" historyStateLocation="client" isArray="false" name="pageNo" primitiveType="Int"/>
  <process:variable description="" historyStateLocation="client" isArray="false" name="pageSize" primitiveType="Int"/>
  <process:variable anyType="commonj.sdo.DataObject" description="" historyStateLocation="client" isArray="false" name="loopRecord"/>
  <process:variable anyType="commonj.sdo.DataObject" description="" historyStateLocation="client" isArray="true" name="wtRecords"/>
</process:variables>
</process:tBusinessLogic>
```

图形化编排

用编排实现微服务组合

ifttt.com

IFTTT

Search



My Recipes

Browse

Channels

zhujiang001

Recipe ID 38742858

[Back to My Recipes](#)



New pull request for the
[Primeton-Ltd/Primetonkb](#)
repository

Send an email from [zhujiang001@gmail.com](#)

Recipe Title

If new pull request for the Primeton-Ltd/Primetonkb repository, then send an email from [zhujiang001@gmail.com](#)

use '#' to add tags

Turn off

Publish

Check now

Log

Delete

created June 28, 2016
never run

Trigger

New pull request for a specific repository

This Trigger fires every time a new pull request is opened for a repository you specify.



Repository name

Primeton-Ltd/Primetonkb

e.g. IFTTT/JazzHands

Action

Send an email

This Action will send an email to up to five recipients from your Gmail account.



To address

[zhujiang@primeton.com](#)

Accepts up to five email addresses, comma-separated



Subject

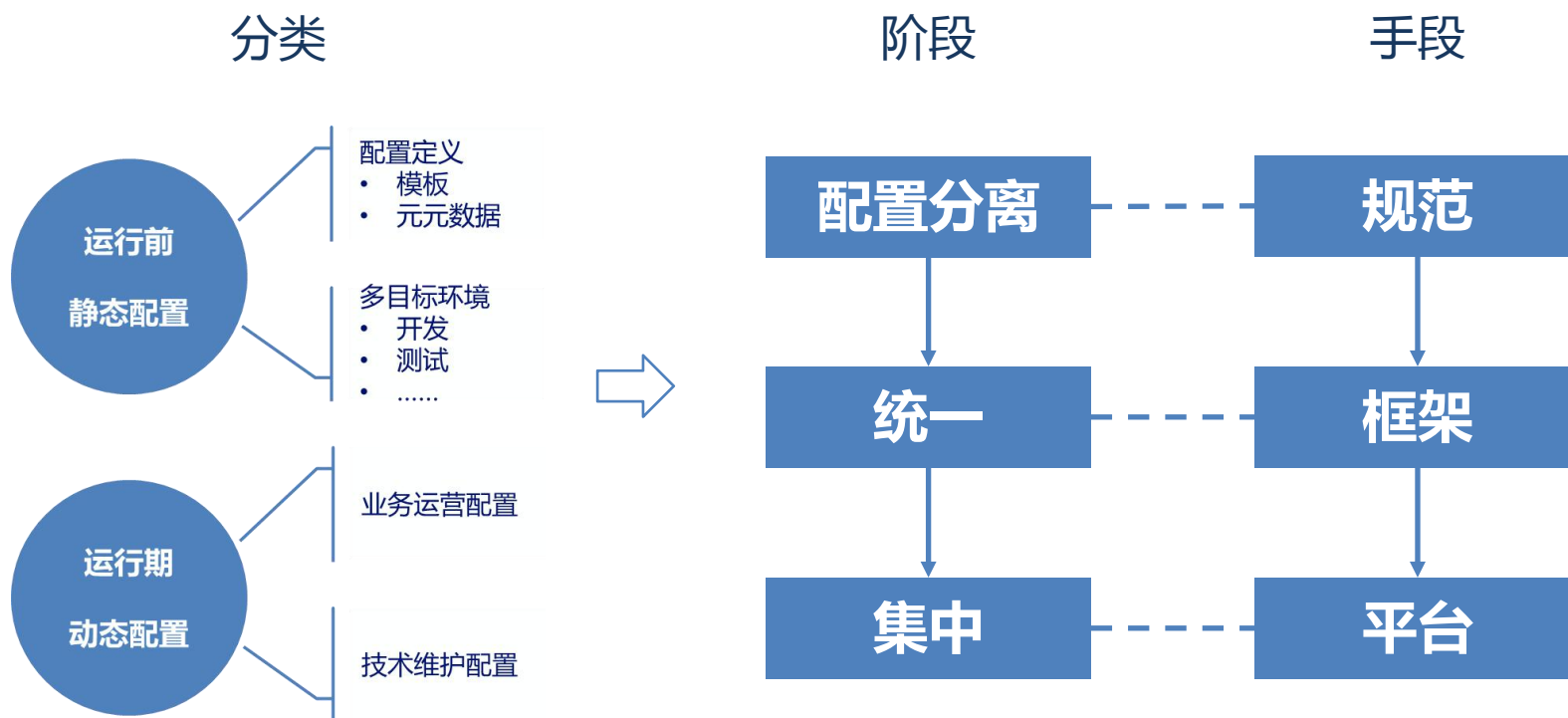
Pull request [[RepositoryName](#)] [PullRequestTitle](#)

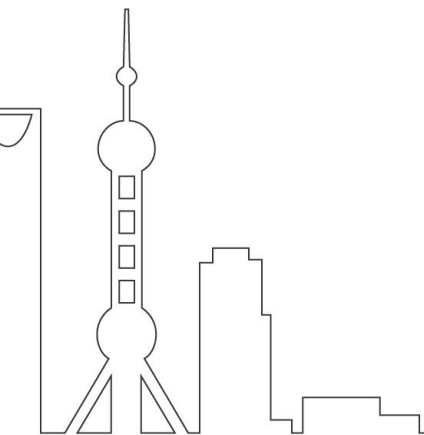


Body

```
<b> PullRequestTitle </b><br>
By AuthorUsername <br><br>
PullRequestBody <br><br>
CreatedAt <br>
via GitHub PullRequestURL
```

八、业务配置集中管理





Thanks!

International Software Development Conference