Geekbang. 极客邦科技 InfoQ

QCon 全球软件开发大会【北京站】2016

个性化定制与批量交付的解决之道

华为基于元数据的企业应用平台简介

Intemational Software Development Conference

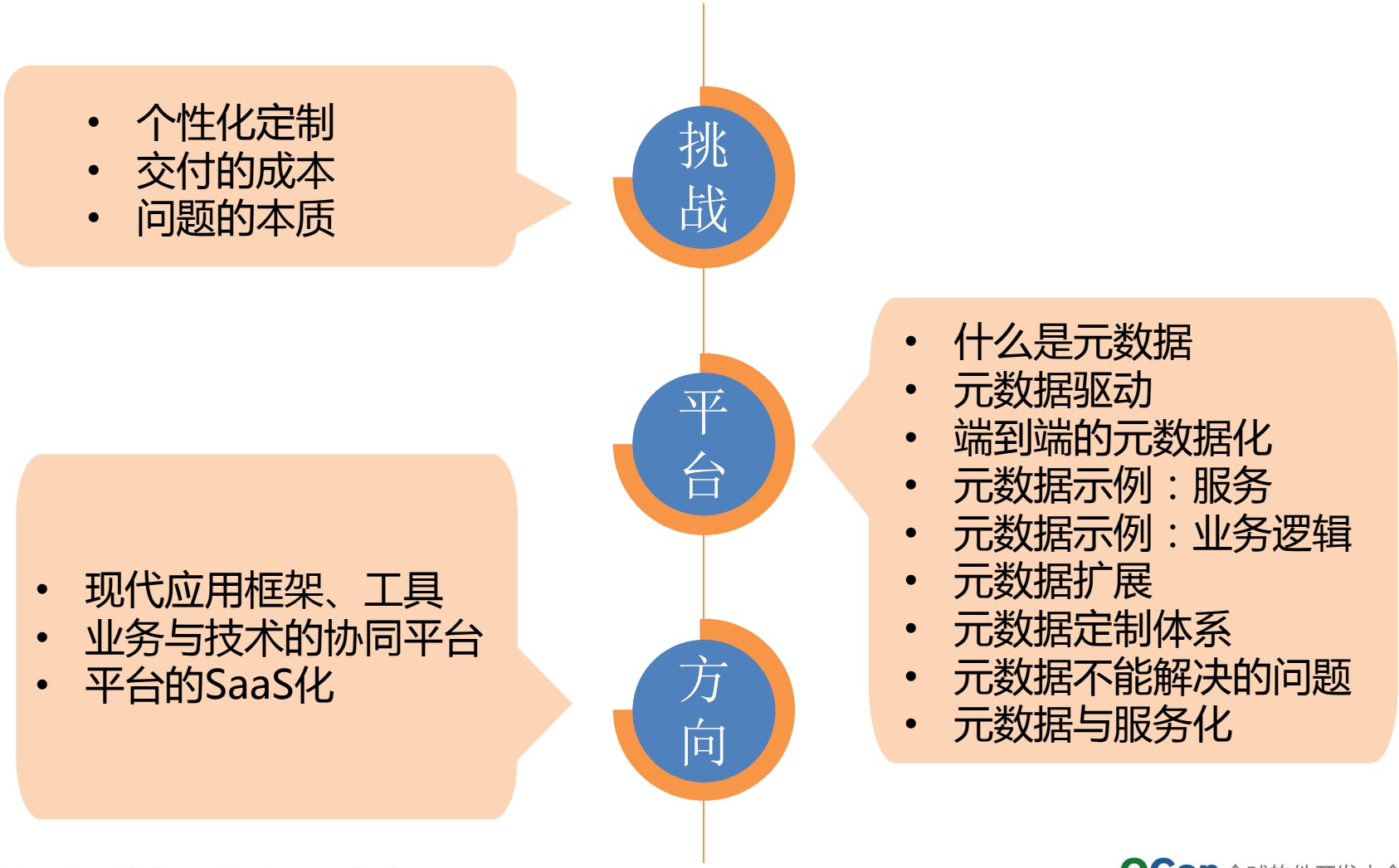
自我简介

- 赵 永
- 主任架构师
- 华为-电信软件
- 基本情况

先后在东软、普元等公司工作，一直从事企业平台相关的架构设计和研发，包括BPM、ESB、开发平台、云平台等等

2015年加入华为，开始了基于元数据的企业平台的研发。对于软件提供商而言，除了面对高性能、架构设计等的挑战外，还面临如何为不同的客户提供有差异化的软件系统，并能够尽量节省成本地按时交付的困难。

大纲


挑战
个性化定制
交付的成本
问题的本质
平台
什么是元数据
元数据驱动
端到端的元数据化
元数据示例：服务
元数据示例：业务逻辑
元数据扩展
元数据定制体系
元数据不能解决的问题
元数据与服务化
方向
现代应用框架、工具
业务与技术的协同平台
平台的SaaS化

个性化定制的挑战

不同的**功能**需求

各运营商系统多种多样，功能自然也不同；即便相同的领域，功能要求也千差万别

如何用一套平台构建出功能多变的应用

不同的**架构**需求

不同的运营商对系统的架构有着不同理解和偏好

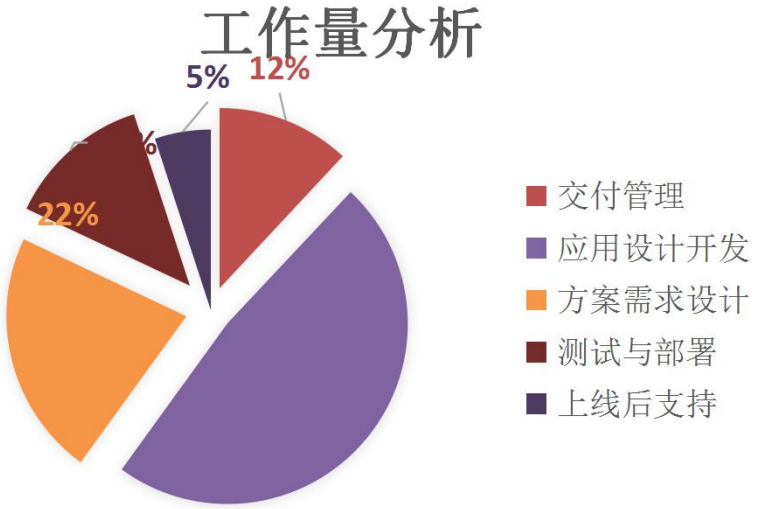
如何用一套平台适应不同的架构需求

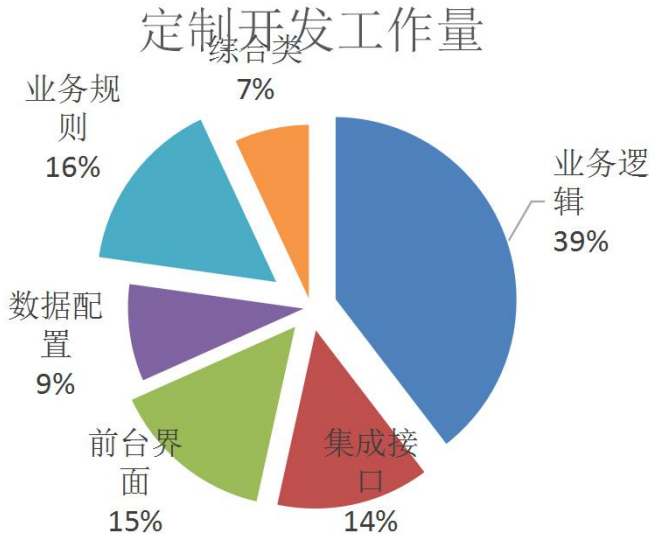
不同的**套件**需求

不同的套件针对不同的业务域，业务的复杂程度不同，对应用的构建方式也不同

如何用一套平台适应不同业务域的构建要求

交付成本的挑战





1. 基线与定制未分离，可基线化的定制需求占比40%以上
2. 90%以上的业务逻辑都是硬编码定制实现，缺少抽象
3. 前台界面定制占比15%
   常用表单都是硬编码实现
4. 接口定制占比14%
   影响交付效率和周期
5. 数据配置9%
   工具方面欠缺

问题的本质是什么

系统可变性

VS

个性化定制

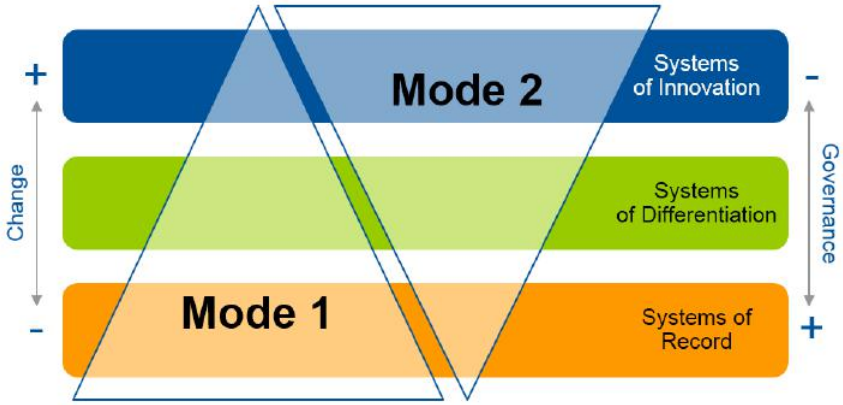
- 系统可变性的3个方面
  - 系统架构的可变性
  - 组件服务的可变性
  - 技术的可变性

Mode 1

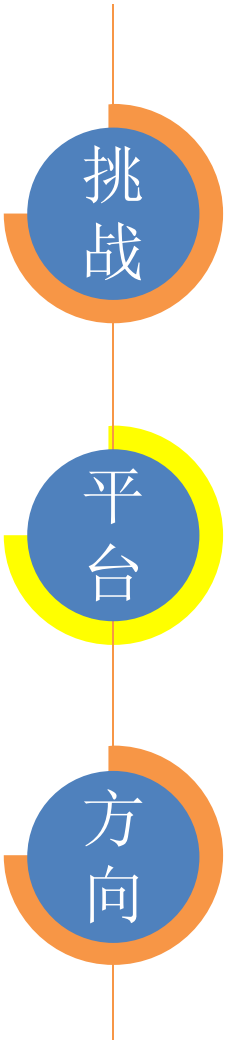
Sequential: stability and reliability

Mode 2

Exploratory: agility and flexibility



大纲



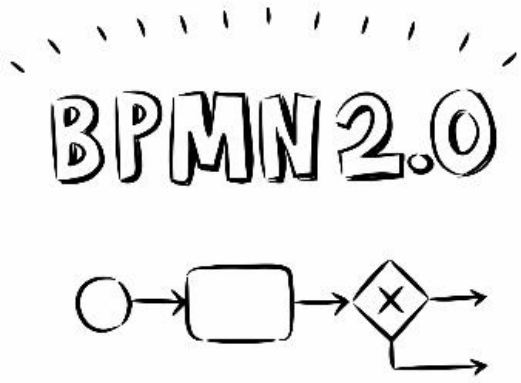
- 个性化定制
- 交付的成本
- 问题的本质

- 什么是元数据
- 元数据驱动
- 端到端的元数据化
- 元数据示例：服务
- 元数据示例：业务逻辑
- 元数据扩展
- 元数据定制体系
- 元数据不能解决的问题
- 元数据与服务化

- 现代应用框架、工具
- 业务与技术的协同平台
- 平台的SaaS化

什么是元数据

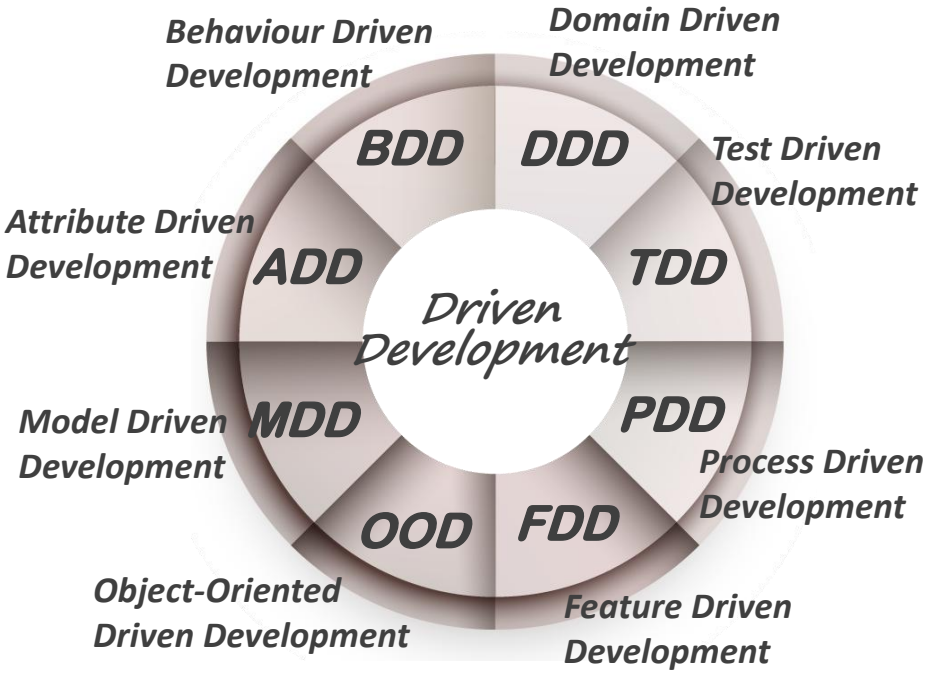
参考牛津词典中的定义：**A set of data that describes and gives information about other data.**



- 不同数据的元数据，内容是不一样的，元数据具有领域特征
- 华为企业平台（Digital Studio）使用体系化的元数据用来呈现应用的各种配置和整个脚本

端到端体系化	应用的可控性
从UI到数据，从功能到性能的所有方面	通过元数据的方式，可以控制到应用的所有环节

什么是元数据驱动的开发



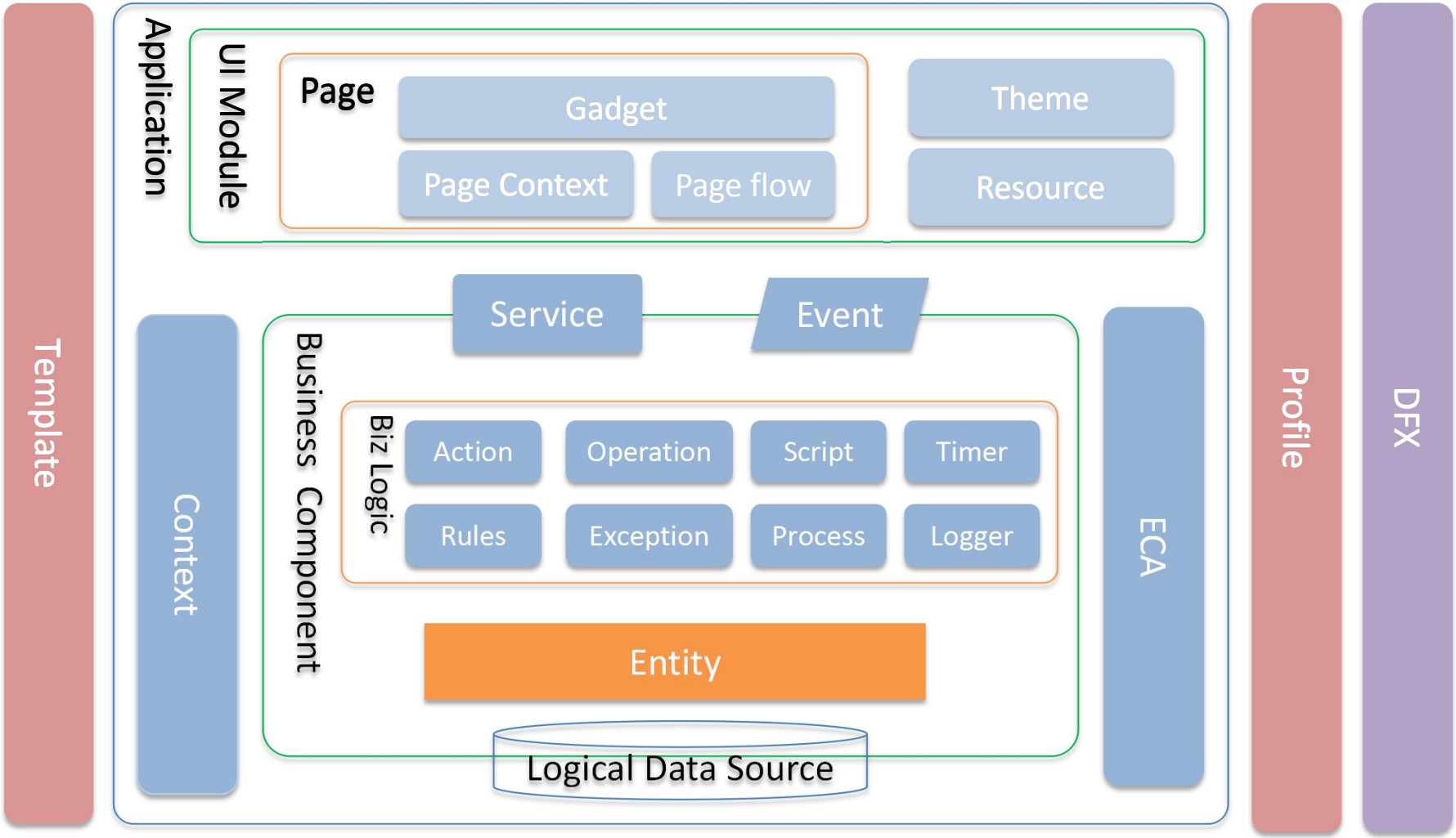
以元数据对象为中心进行业务应用的设计

基于业务对象声明式开发整个业务应用

围绕元数据对象进行业务应用的全生命周期管理

元数据驱动开发是一个**体系**，核心在于对应用的元数据抽象，并建立基于元数据的全工具链支撑。

端到端元数据化


Template
Application
UI Module
Page
Gadget
Page Context
Page flow
Theme
Resource
Service
Event
Business Component
Biz Logic
Action
Operation
Script
Timer
Rules
Exception
Process
Logger
Entity
Logical Data Source
Context
ECA
Profile
DFX

元数据示例：服务

```
<business-service name="Business" >

    <operation name="checking">
        <in-parameters>
            <parameter name="param1" type="entity:PARAM1"/>
            <parameter name="param2" type="entity:PARAM2"/>
        </in-parameters>
        <out-parameter name="outParam" type="entity:PARAM3"/>

        <actions>
            <call-script location="/scipts/checking.script"/>
            <call-business-operation name="" out-field="boOut">
                <parameter-mapping parameter-name="boParam1" from="param1.person"/>
                <parameter-mapping parameter-name="boParam2" from="param1.name"/>
            </call-business-operation>
        </actions>

    </operation>

</business-service>
```

```
<service-trigger name="" service="Business#checking" when="pre-service">
    <condition>
        <expression></expression>
    </condition>
    <actions>
        <if condition="param1.person.age &gt 5 "/>
            <!--do some business logic-->
            <else-if/>
        </if>
    </actions>
</service-trigger>
```

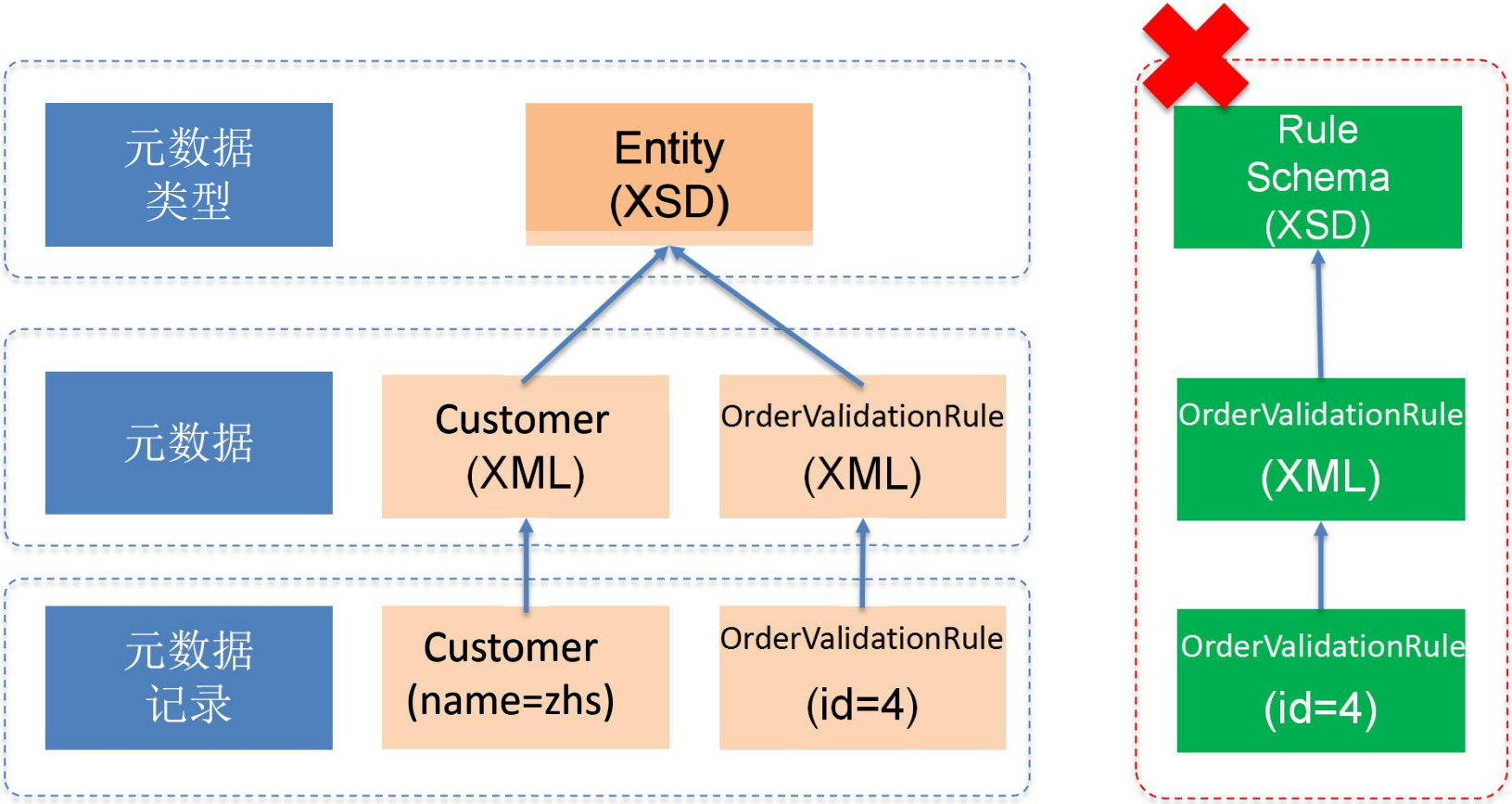
元数据示例：业务框架元数据

```
<om:request-validation id="BizCommonValidate" name="通用开户业务报文校验" version="1.0"
    xmlns:om="http://www.huawei.com/bes/om/schema/core">

    <om:validate-rule priority="1" id="bizCommonValidation">
        <!--业务校验逻辑定义-->
    </om:validate-rule>

</om:request-validation>
```

```
<om:request-validation id="groupInstallValidation" name="集团开户业务报文校验" version="1.0"
    extends="BizCommonValidate"
    xmlns:om="http://www.huawei.com/bes/om/schema/core">

    <om:validate-rule priority="2" id="groupInstallValidation1">
        <!--业务校验逻辑定义-->
    </om:validate-rule>
</om:requ
```

```
<om:request-validation id="personInstallValidation" name="个人开户业务报文校验" version="1.0"
    extends="BizCommonValidate"
    xmlns:om="http://www.huawei.com/bes/om/schema/core">

    <om:validate-rule priority="2" id="personInstallValidation1">
        <!--业务校验逻辑定义-->
    </om:validate-rule>

</om:request-validation>
```

元数据的扩展体系-示意


元数据
类型
Entity
(XSD)
元数据
Customer
(XML)
OrderValidationRule
(XML)
元数据
记录
Customer
(name=zhs)
OrderValidationRule
(id=4)
Rule
Schema
(XSD)
OrderValidationRule
(XML)
OrderValidationRule
(id=4)

元数据的扩展体系

元数据扩展体系：元数据的自我管理和描述体系。元数据的扩展机制，保证了不同的元数据具有相同的扩展、兼容性检查、打包、热部署、资料生成、治理等机制。

业务元数据
应用元数据

应用元数据描述了平台技术层面的元数据
业务元数据是基于应用元数据的扩展能力定义的业务逻辑

自生长机制

使用同一套模型来定义其它的元数据结构，保证同源，为统一元数据生命周期管理机制的基石

基础元数据

定义基础的数据、逻辑、组件的表示方式，也称为元数据类型

基于元数据的定制开发体系

元数据仓库
元数据的统一管理、分析
全球元数据仓库、局点本地仓库
5

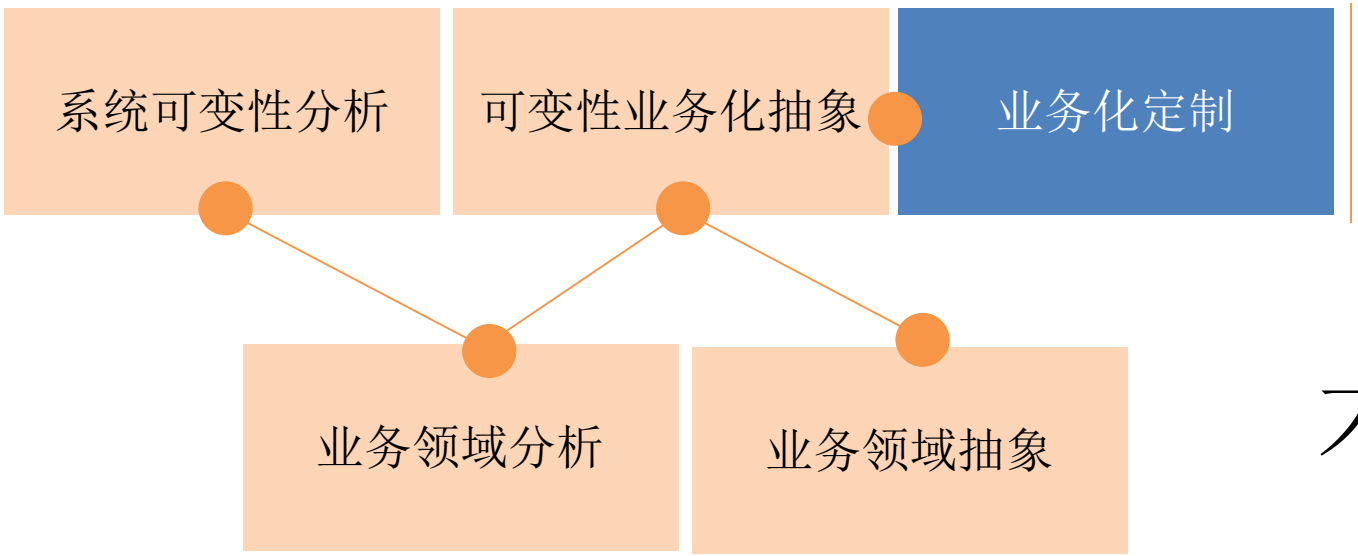
元数据引擎
完全基于元数据的运行引擎
高性能、安全、故障定位
4

定制工具
端到端的元数据定制工具
开发、测试、部署、运行、监控、运维、资料
3

定制机制
基线定制能力分离
定制能力固化到元数据上
2

元数据
技术元数据
业务元数据.
1

元数据不能解决的问题



业务化的快速定制，快速上线是最大目标

不能解决

利用基于元数据的声明式开发，将可变性进行业务化封装、屏蔽底层技术，使开发人员更多的关注业务本身，从而更快地进行全球局点的个性化定制。

元数据体系平台的难点

业务与技术的协同

以业务的视角进行定制开发，以技术的视角进行基线开发

模块内元数据贯通

组件内数据与逻辑相贯通，在限制范围内修改，不相互影响

贯通与协同

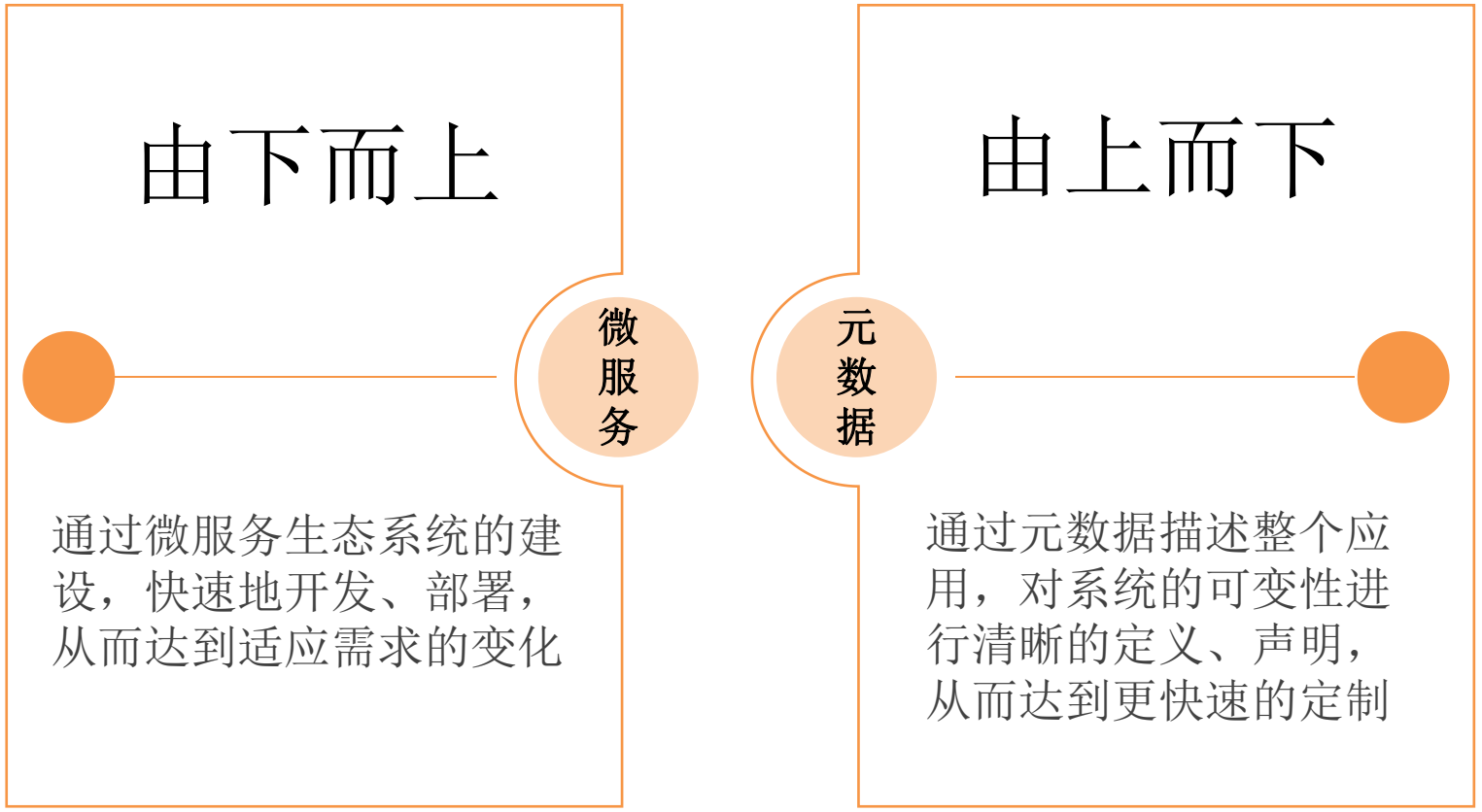
前后台贯通

以业务对象为中心，前后台贯通并响

模块与模块之间贯通

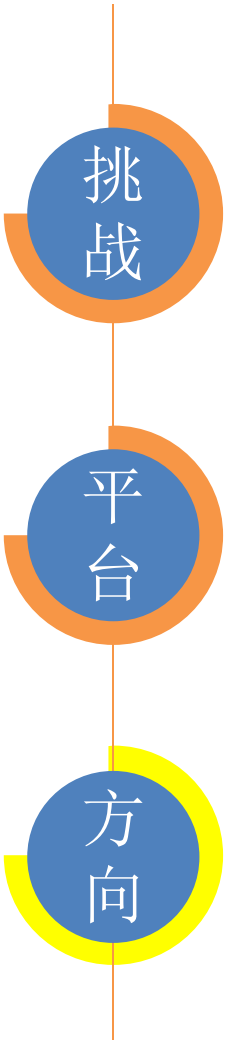
组件自治，元数据描述组件之间的引用，服务调用组件贯通

元数据体系与微服务架构



面向的场景不同，采用的方式不同，都适应了系统的快速变化。

大纲



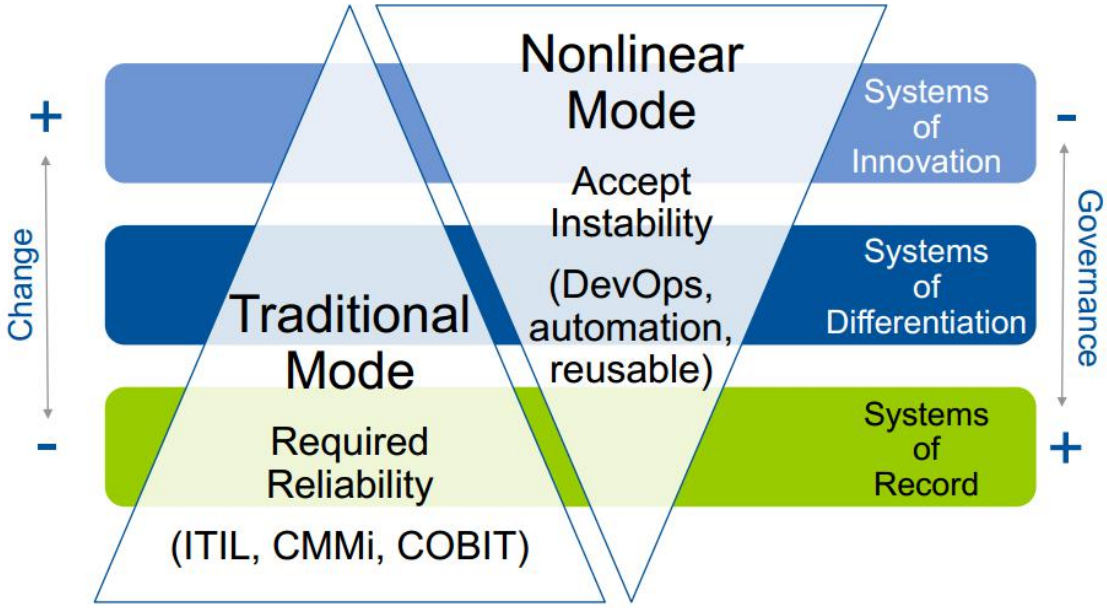
- 个性化定制
- 交付的成本
- 问题的本质

- 什么是元数据
- 元数据驱动
- 端到端的元数据化
- 元数据示例：服务
- 元数据示例：业务逻辑
- 元数据扩展
- 元数据定制体系
- 元数据不能解决的问题
- 元数据与服务化

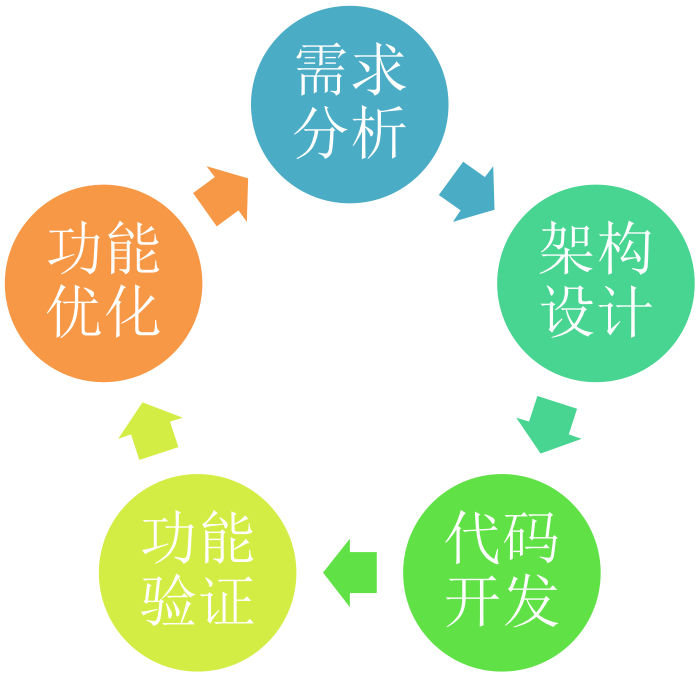
- 现代应用框架、工具
- 业务与技术的协同平台
- 平台的SaaS化

现代应用框架、服务和工具

- aPaaS存在两个分支（High-productivity和High-control），分别支撑Bimodal的两种IT模式
- 两种aPaaS具备相同的基础云特性，弹性伸缩、多租、自动管控、高可靠等
- 两种模式具有融合的趋势



技术与业务协同的创新平台



业务编排 × 原型验证

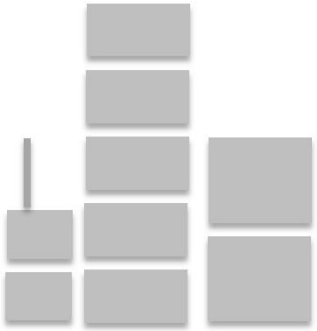
1. 业务与技术的一致性视图
   业务人员直接参与设计
2. 强大的数字资产管理能力，
   提供众多的专业行业包
3. 基于资产的业务、IT编排能力
   快速导入需求、快速原型验证

SaaS化企业平台

多租户

云定制

云运行



本地平台

Geekbang 极客邦科技 InfoQ

THANKS!