携程在企业架构方面的探索

刘剑@携程研发中心•软件架构部

目录

相关背景•携程旅行网

携程旅行网

1999年成立，总部设在中国上海，在北京、广州、深圳、成都、杭州、南京、重庆、厦门、青岛、武汉、沈阳、三亚、丽江、南通、香港等16个城市设有分支机构。现有员工 12000名。

是国内领先的综合性旅行服务公司，提供酒店预订、机票预订、旅游度假、商旅管理、美食订餐及旅游资讯等全方位旅行服务。

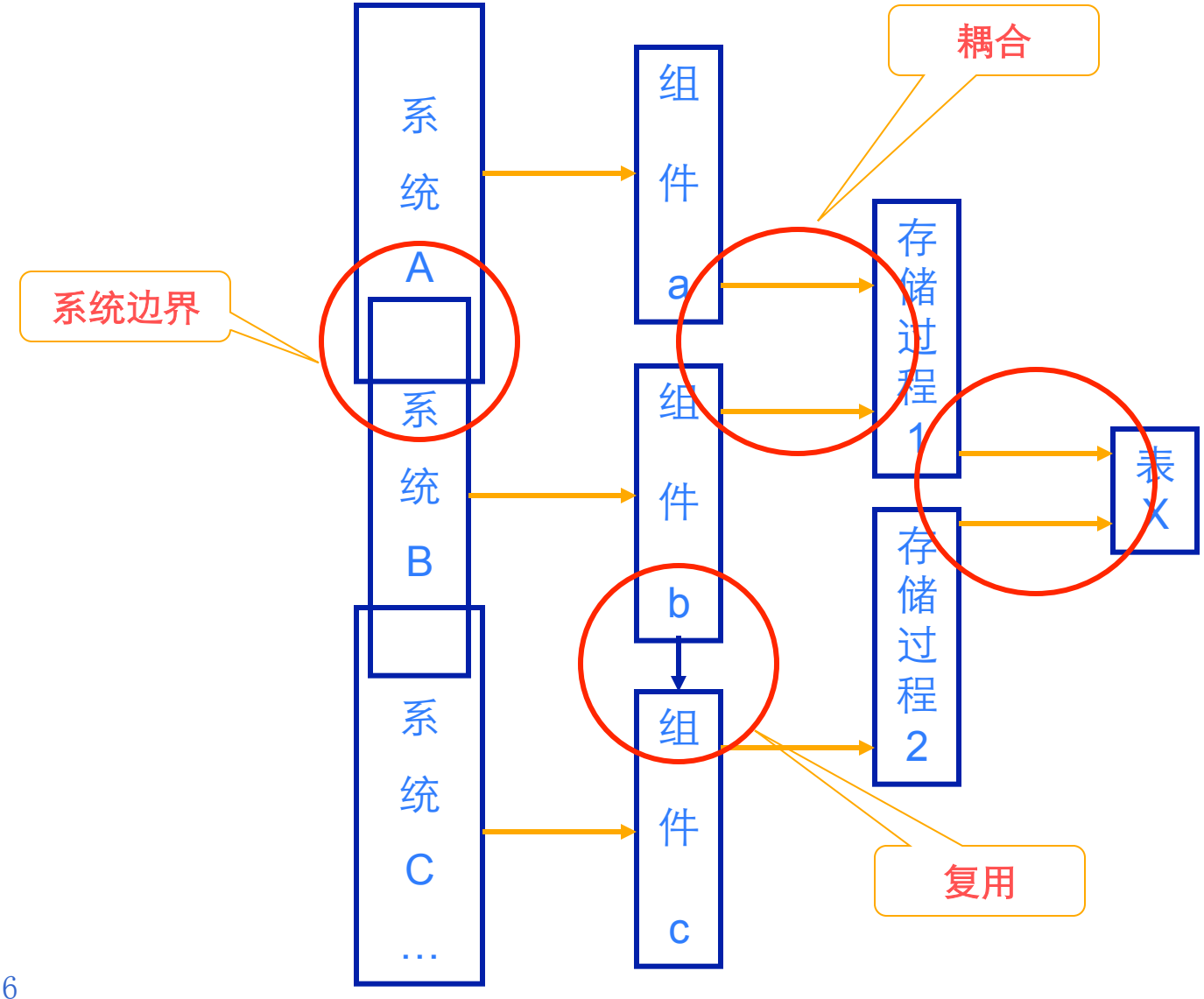
相关背景•业务特点

方面	特点
业务特色	➢核心价值在于整合资源，提供服务 ➢存在众多的各类资源及代理合作方 ➢服务资源(合作方产品)不受控，基本无库存
业务流程	➢业务流程复杂，要求灵活 ➢存在许多特殊的人工处理环节
业务变化	➢业务流程变化 ➢产品功能变化

相关背景•系统特点

方面	问题	举例
系统庞杂	经过十多年的快速发展，内部系统越来越多，彼此间交互和依赖越来越复杂	22个产品，100个子系统，459个应用，203个数据库
系统划分	系统边界模糊、职责不清、耦合过深，导致系统难以维护和扩展	机票结算子系统 订单与支付
业务流程	流程过于分散以及流程的硬编码导致对流程变化的支持缓慢	出票、出保、配送
数据分布	多系统间的数据的不合理的集中与分散，在系统越来越复杂的情况下，严重影响系统的性能与可扩展能力	集中：庞大的订单表 分散：支付方式、证件类型
开发模式	多种技术并存的情况下，开发模式的选择，直接关系到开发效率及运维效率	各种各样的系统交互方式 JOBAPP/JOB/SSIS/ Replication/WCF/ Remoting/WS

相关背景•系统特点



相关背景•技术特点

方面	技术
操作系统	WinServer2008 CentOS5.6
Web服务器	IIS6
开发平台	VS6:ASP VS2005/2010:ASP.NET,C# 其他开发工具:C++Builder,Eclipse,Dephi6
数据库	SQLServer2008
开源/框架	Entl2.0/Entl5.0/MVC/ORM… Memcache Lucense

相关背景•期望目标

方面	目标
业务响应	降低系统耦合，提升对变化的响应速度 改善系统架构，更好地支持业务扩展 建立流程引擎，更灵活地支持业务流程变化
系统质量	合理地增加或减少系统间交互及补偿，提升系统性能、稳定性
开发效率	合理划分系统，通过系统职责的分离，对开发组的职责进行合理分配，同时建立更完整的公共平台、基础框架、基础类库，提高开发效率
运维水平	完善配置、监控、预警、日志系统，提升系统运维配置效率及排查问题的速度

目录

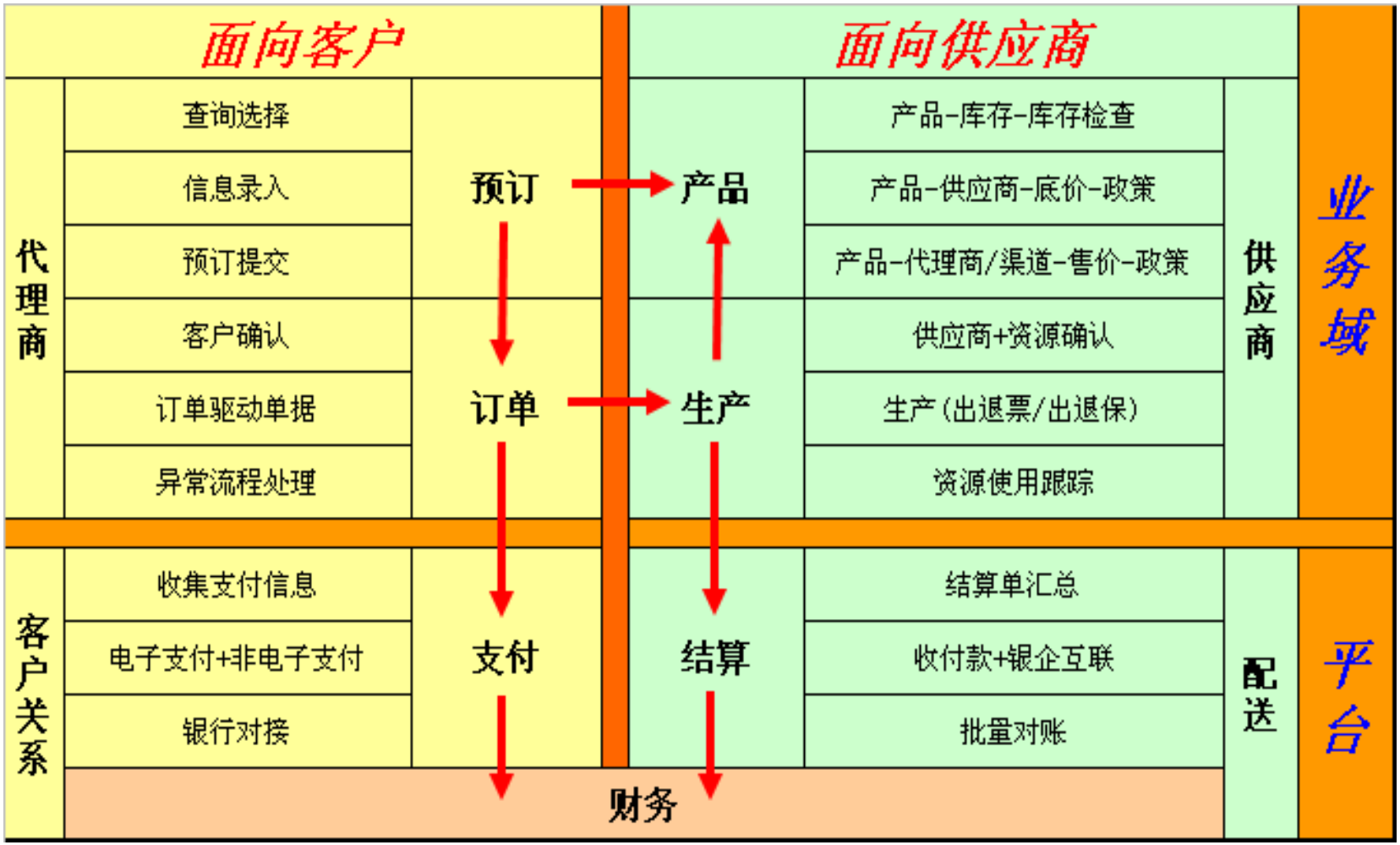
架构规划

架构视角	规划重点
业务架构	业务模型 流程模型
应用架构	子系统划分、服务识别与设计 应用系统交互方式、应用组合 应用与应用、数据、部署矩阵
数据架构	数据在系统之间的分布 系统间数据交换的方式 主数据管理与数据迁移
技术架构	技术标准/基础框架/基础组件 基础平台/管理平台/治理平台/监控平台

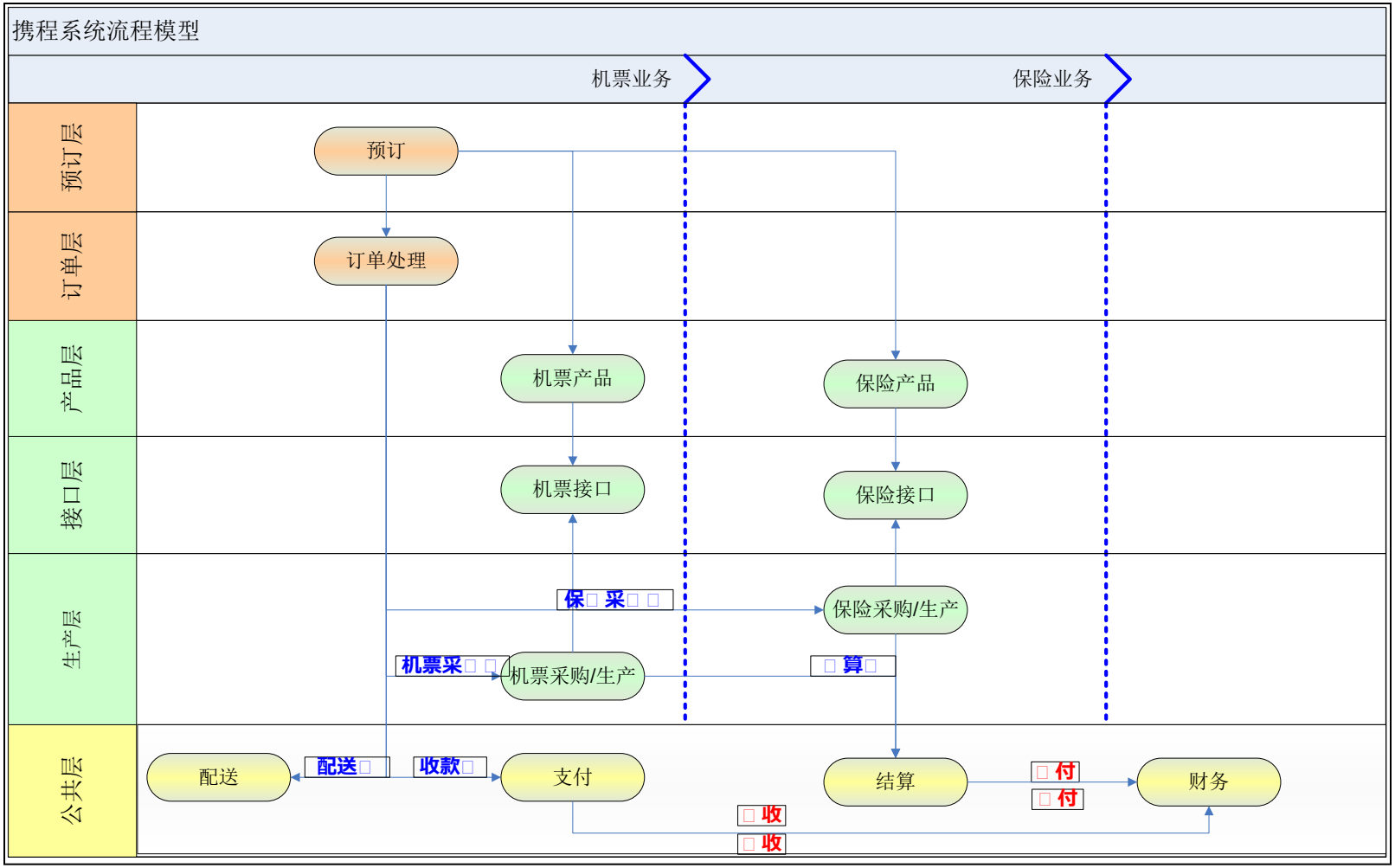
架构规划•业务架构•原则

- 业务系统流程中要体现清晰明确的物流、资金流、信息流
- 业务系统流程由订单驱动，订单由产品组成
- 订单根据产品组合的不同有灵活可变的生产流程
- 产品由供应商提供，具有固定的采购或生产流程
- 产品具有多种共同的属性(支付属性、配送属性等)，以此才能组合成订单
- 当产品具有库存特性(金额或数量)时，应该采用正向+/反向-的方式生成不同的单据，比如支付、结算
- 当产品与订单本身的一些特定属性关联密切时(比如机票与乘机人关系)，应该采用在相同单据上做回退处理，而不是产生新的反向冲销单据

架构规划•业务架构•业务模型



架构规划•业务架构•流程模型



架构规划•应用架构•系统划分原则

方面	目标
职责分离原则	子系统遵循业务类型按层级划分；子系统的业务职责范围应该明确，功能独立、完整
高内聚低耦合原则	子系统内部业务和数据联系紧密，趋向高内聚；子系统之间边界清晰并关联弱化，趋向低耦合
通用专用分离原则	具有通用性或专用性的模块应该考虑拆分独立子系统
技能分离原则	对开发人员有特殊要求的模块可以考虑成独立子系统

子系统拆分后应具备特征：

- 代码独立、应用独立、数据库独立
- 通过ESB对外提供服务或请求其他子系统的服务
- 可以并只能由单独的开发小组独立负责所有的改动

架构规划•应用架构•子系统划分

接入层	中文(31-Chinese)			繁体(32-Big5)		英文(33-English)			移动(34-WAP)			实验室(35-Labs)				基础平台	联络中心
	3101-Chinese.Site		...	3201-Big5.Site	...	3301-English.Site		...	3401-WAP.Server	3402-WAP.Client	...	3501 Labs.App	3502-Labs.Tech	3503-Labs.Open	3504-Labs.Platform	Arch-90	CTI-91
业务层			酒店	国内机票	国际机票	度假	商旅	保险	会展	游票	高铁	团购			...	9010-Arch.Search	9101-CTI.Base
		名称	Hotel	Flight	FlightIntl	Package	Corp	Isurance	MICE	TravelMoney	Train	Group					
	名称	编号	11	12	13	14	15	16	17	19	20	21				9009-Arch.Cache	
预订层	Booking	01	1101	1201	1301	1401	1501		1701		2001	2101					9102-CTI.App
订单层	Order	02	1102	1202	1302	1402	1502		1702		2002	2102					
生产层	Process	03	1103	1203		1403	1503	1603	1703			2103				9006-Arch.MDM	
产品层	Product	04	1104	1204	1304	1404	1504	1604	1704	1904	2004	2104					9103-CTI.MMP
供应商	Vendor	05	1105	1205	1305	1405	1505					2105					
代理商	Agent	06	1106	1206	1306	1406										9005-Arch.RDSys	9104-CTI.MiddleWare
接口层	Switch	10	1110	1210	1310			1610			2010						
...																9004-Arch.App	9105-CTI.IVR
公共层	业务	支付							配送	结算					财务	...	
	名称	Payment							Delivery	Account					Finance		
	编号	41							42	43					44		9106 CTI.Recorder
	类别	现金支付	电子支付	游票支付	支付基础	风控	支付直连		基础	供应商结算		客户结算	银企互联		基础	9003-Arch.Gov	
	名称	Casher	Epay	TMPay	Base	CardRisk	Switch		Base	Vendor		Corp	Switch		Base		
	编号	01	02	03	04	05	51		01	01		02	03		01	9002-Arch.SOA	9107-CTI.Fax
	子系统	4101	4102	4103	4104	4105	4151		4201	4301		4302	4303		4401		
	业务	客户关系管理平台				内部信息系统											9108-CTI.SMS
	名称	CRM				CMIS											
	编号	48				49											9109-CTI.Email
	类别	用户	支持	市场	销售	人事			办公							9001-Arch.FrameWork	
	名称	User	Support	Market	Sale	HR			OA								
	编号	01	02	03	04	01			2								
	子系统	4801	4802	4802	4804	4901			4902								

架构规划•应用架构•系统流程

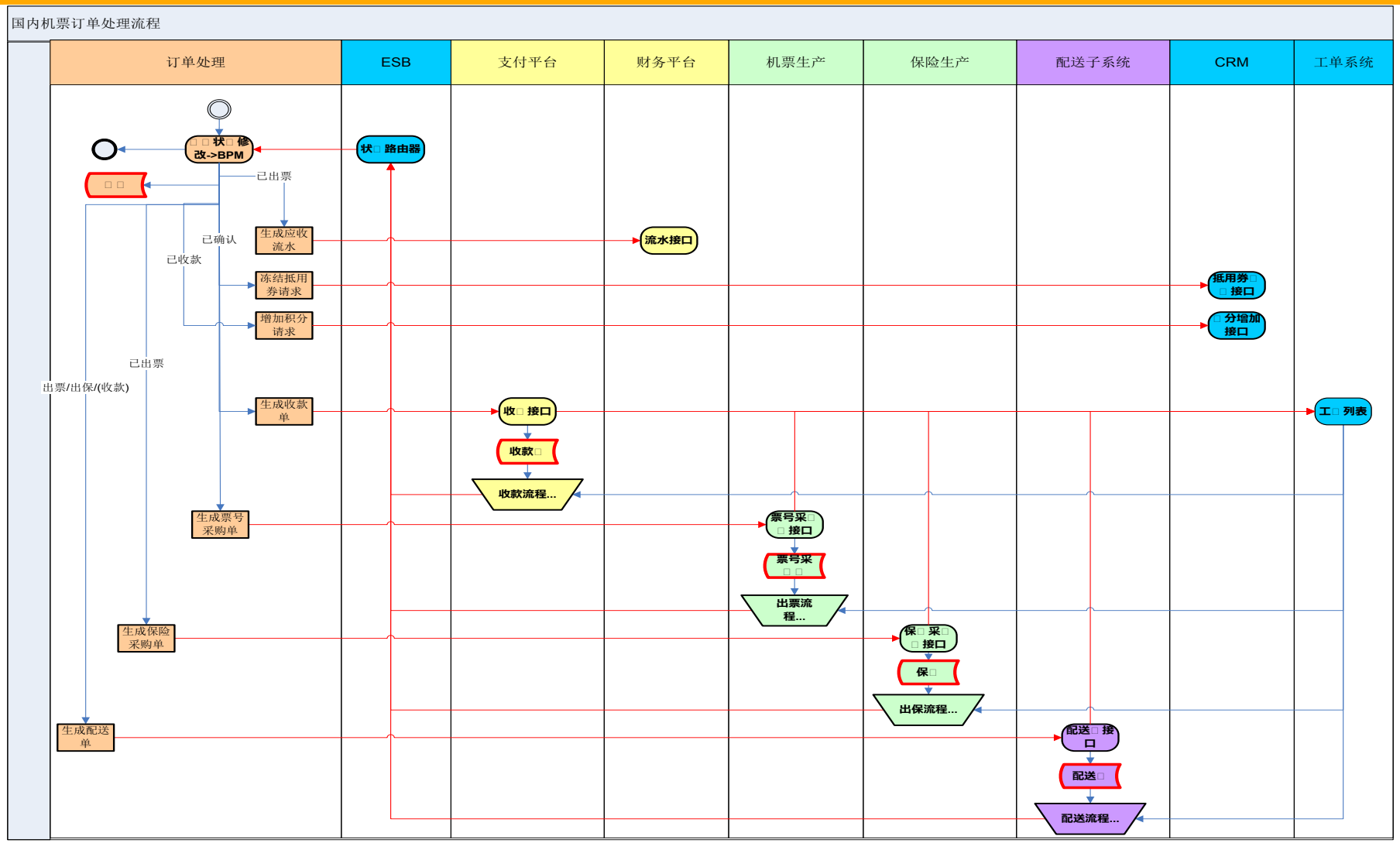
总体流程：

- 流程模板：定义统一的流程模板，各业务流程在标准流程模板进行补充或裁减；
- 流程职责：定义流程中各系统或应用的具体职能；

系统流程：

- 核心流程：定义以状态驱动为核心的订单处理流程；
- 支撑流程：定义各个产品生产、采购的支撑流程；
- 异常流程：对系统或业务级的异常提供完善的处理机制。

架构规划•应用架构•系统流程



架构规划•应用架构•服务抽取

名称	自包含原则
解释	对客户端而言，服务应是自包含的。自包含是指客户端使用本服务不需要依赖其他服务，服务内部是紧耦合而服务之间是松耦合的。
参考	考虑客户端是否需要调用其它服务组件模块才能满足业务请求?
案例	比如一个保险调整服务，包含取消和出保两个部分，并且是先取消再出保的。那么我们说“保险调整”是一个服务，而“取消”和“出保”在这个场景下不是独立的服务。

名称	无状态原则
解释	原子服务应该都是独立的请求，服务的实现不应该依赖于其他服务的上下文和状态，即服务应是无状态的。
参考	考虑该服务实现是否需要其它服务的状态或结果?
案例	如果有两个候选服务,机票流程服务,出票完成状态通知服务。机票流程服务是由出票完成状态通知服务和出保服务构成,因为在出保之前,需要先出票完成,然后根据出票的结果出保。机票流程服务是有状态的,需要BPM实现，而出票完成状态通知服务和出保服务是无状态的。

名称	跨子系统边界原则
解释	服务应跨子系统边界。SOA的目标之一是实现分布式系统间的整合，因此服务至少应跨子系统边界。那些在子系统内部调用的模块不适合服务化，如果服务化，反而降低效率。
参考	考虑该服务是否会被其它外部子系统访问
案例	同一个子系统内部不需要提供服务。

名称	业务复用原则
解释	业务层面上的复用是SOA从具体实施角度上的最大价值，一个服务被复用的频度越高，其对整个SOA架构的贡献就越大。服务暴露时应尽可能考虑复用，复用是降低开发成本，提高IT敏捷性的关键之一。
参考	评估该候选服务被其它外部系统访问频度和数量。

名称	服务粒度原则
解释	几个服务在业务上关联性大，应考虑合并这些服务为一个更粗粒度的新服务。在交易数量众多的情况下尤其如此。服务之间关联性可以从业务逻辑和业务数据两方面考虑。
参考	评估多个候选服务的业务关联性。

架构规划•应用架构•架构库

方面	目标
应用库	对所有产品、子系统、应用的相互管理、配置、部署、数据使用等情况进行统一管理，并提供相应的变更治理流程。
服务库	对所有SOA相关服务统一进行管理，并由此建立完整的服务库。服务库将是一个十分重要的系统资源库。其中完整记录了各个服务的接口契约、设计文档、版本记录等等。帮助开发人员熟悉业务系统的相关服务；为客户端所有开发小组提供完整详细的服务使用指南。
流程库	记录完整的系统流程演变历史，以供开发人员参照、熟悉相关的业务系统；在服务注册审核、状态注册审核、访问权限审核等环节起到辅助作用。
状态库	状态库，包含所有子系统的状态列表，状态迁移图，以及各子系统之间状态的互相影响的规则。

架构规划•应用架构•关系矩阵

应用/应用矩阵

		产品	Flight					
		子系统	Flight.Process			Flight.Booking		
		应用	Flight.Segment.Management	Flight.Ticket.Invalidate	Flight.Ticket.Refund	...	...	...
产品	子系统	应用						
Package	Package.Booking	Package.Booking.Offline						
		Package.Booking.Online		▲	▲			
		...	▼					
	Package.Order	...						
		...				▼		▼
		...					▼	

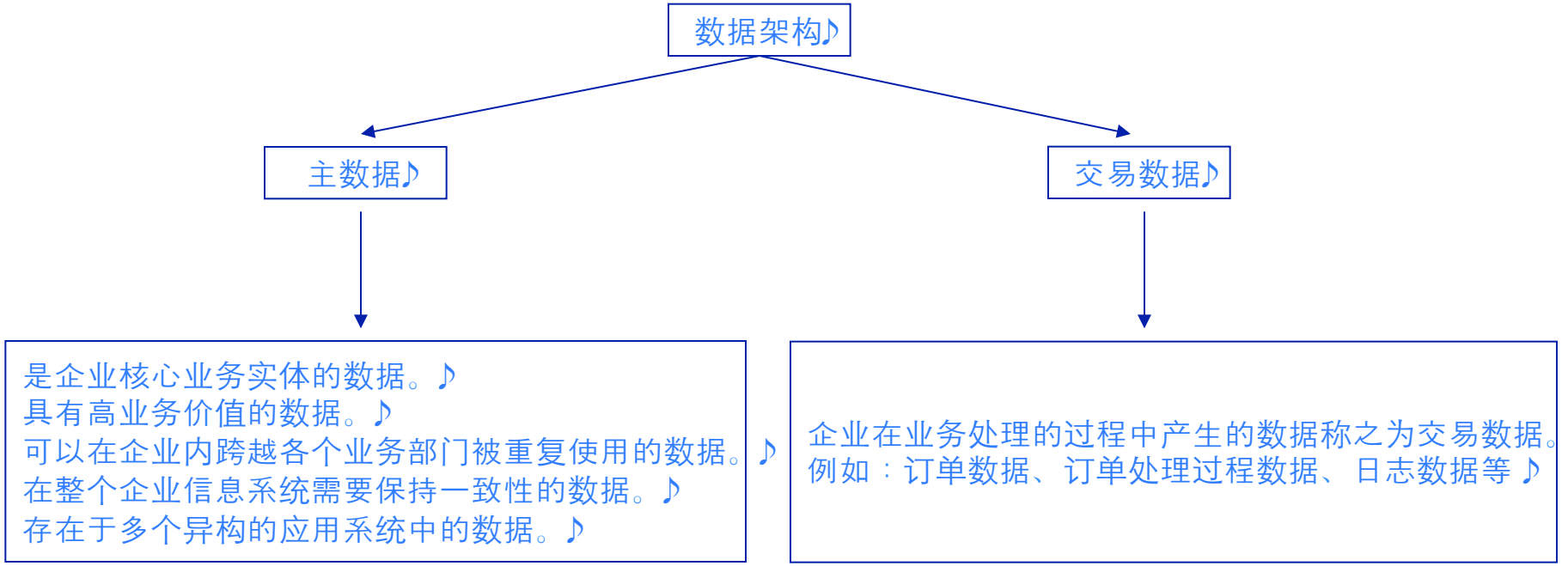
应用/部署矩阵

		产品	Flight					
		子系统	Flight.Process			Flight.Booking		
		应用	Flight.Segment.Management	Flight.Ticket.Invalidate	Flight.Ticket.Refund	...	...	...
服务器	域名	虚拟目录						
SVR215	accint.sh.ctriptravel.com	AccHotel						
SVR216		AccFltTicket		●	●			
SVR219		AccFlgSegment	●					
...	flight.sh.ctriptravel.com	...						
		...				●		●
		...					●	

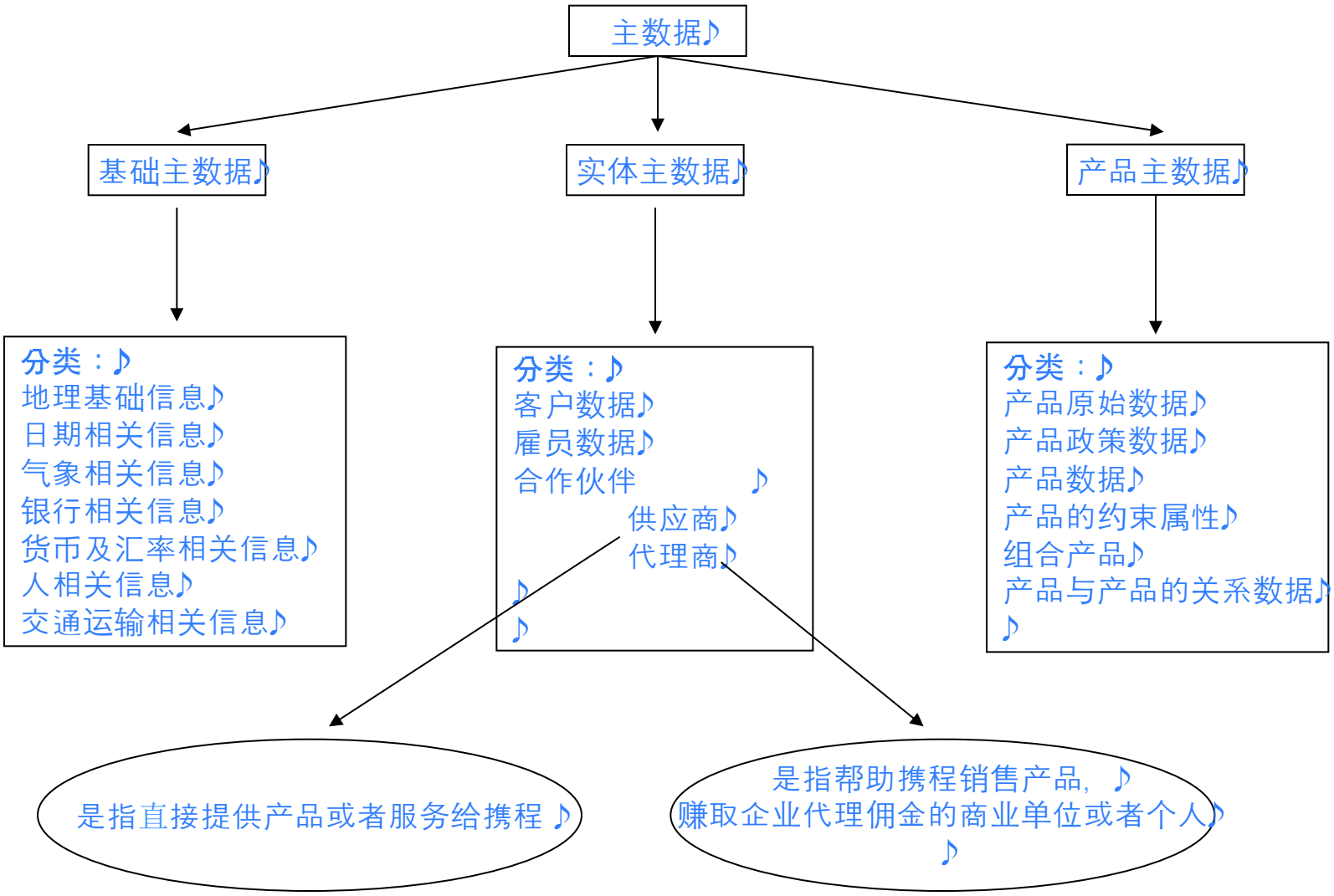
应用/数据矩阵

		产品	Flight					
		子系统	Flight.Process			Flight.Booking		
		应用	Flight.Segment.Management	Flight.Ticket.Invalidate	Flight.Ticket.Refund	...	...	...
服务器	域名	数据库						
SVR325	AccFlt.db.sh.ctriptravel.com	AccFltDB	●	●	●			
...	FltOrder.db.sh.ctriptravel.com	FltOrderDB			●			

架构规划•数据架构


数据架构♪
主数据♪
交易数据♪
是企业核心业务实体的数据。♪
具有高业务价值的数据。♪
可以在企业内跨越各个业务部门被重复使用的数据。♪
在整个企业信息系统需要保持一致性的数据。♪
存在于多个异构的应用系统中的数据。♪
企业在业务处理的过程中产生的数据称之为交易数据。♪
例如：订单数据、订单处理过程数据、日志数据等♪

架构规划•数据架构


主数据♪
基础主数据♪
实体主数据♪
产品主数据♪
分类：♪
地理基础信息♪
日期相关信息♪
气象相关信息♪
银行相关信息♪
货币及汇率相关信息♪
人相关信息♪
交通运输相关信息♪
分类：♪
客户数据♪
雇员数据♪
合作伙伴 ♪
供应商♪
代理商♪
♪
♪
分类：♪
产品原始数据♪
产品政策数据♪
产品数据♪
产品的约束属性♪
组合产品♪
产品与产品的关系数据♪
♪
是指直接提供产品或者服务给携程 ♪
是指帮助携程销售产品，♪
赚取企业代理佣金的商业单位或者个人♪
♪

架构规划•数据架构•交易数据

交易数据主要由属主子系统所产生和使用，存储在子系统所对应的数据库。如果其他子系统在业务处理的过程中，需要这部分数据，则需要通过以下四种方式进行数据的交互、转移和使用：

➢服务接口

优先推荐通过服务接口的形式进行子系统之间进行数据交换。具体为后端子系统提供相关的单据服务，前端子系统将这部分数据通过单据消息的方式发送到需要数据的后端子系统。在必要情况下，后端系统对数据进行业务处理之后，通过回调接口的方式将部分数据返回给前端系统。

➢SSIS

优点：可以跨服务器，功能很强，可实现复杂的业务逻辑，稳定性较好，开发周期较短；
缺点：维护复杂；
适用：在逻辑复杂、流程十分清晰的跨子系统的大数据量处理情况下，可以采用SSIS进行系统之间的数据交互。

➢Job

优点：Job可处理大批量数据，开发周期短，维护简单
缺点：不能跨服务器，不适用于大量复杂计算。
适用：子系统内部的大批量数据的简单处理。

➢JobApp

优点：可以跨服务器，可实现复杂的大量计算，可以调用外部程序。；
缺点：性能一般，一次只能处理少量数据，开发周期较长，后续的维护升级复杂。
适用：计算复杂、少量数据的跨子系统的数据处理，一般需要调用其他子系统接口。

架构规划•数据架构•基础主数据

基础主数据

分类	说明
范围	地理信息、日期信息、气象信息、银行信息、货币信息、支付信息、人信息、交通信息、通讯信息
存储	基础数据的存储在主数据管理系统中的BaseMDDB，然后通过复制分发，发布到各个子系统Server的BaseMDDB
使用	◆通过主数据管理系统，从MDM系统的CommMDDB来统一操纵（CUD）和维护公共基础数据。 ◆各个子系统直接访问复制分发到其对应Server上BaseMDDB中的公共基础数据。 ◆各个子系统也可通过接口来使用公共基础数据

架构规划•数据架构•产品主数据

产品主数据

分类	说明
范围	◆原始产品数据、产品政策数据、销售产品数据、组合产品 ◆产品与产品的关系数据 ◆产品的约束属性（产品与地点，时间和人员的关系）
存储	产品主数据，按照它们各自所属的产品领域，数据存储在属主产品子系统对应的物理数据库中。

使用方式	相同软件产品领域	不同软件产品领域
简单数据	直接访问数据库	●WebService 访问（推荐） ●通过数据分发，直接访问数据库
复杂数据	●WebService 访问（推荐） ●直接访问数据库	WebService 访问

架构规划•数据架构•实体主数据

实体主数据

分类	定义
客户主数据	客户的自然基本属性，业务中的角色，行为习惯，银行账户信息，详细联系信息。
雇员主数据	雇员的基本属性，业务中的角色，权限信息。
供应商主数据	供应商基本信息，供应商与采购，结算，财务的相关信息。
代理商主数据	代理商基本信息，代理商与结算，支付，财务的相关信息。

分类	使用规则
一般实体主数据	数据的操纵（CUD）和维护由MDM系统完成。通过复制分发，把数据分发到要使用它的系统中。
敏感实体主数据	通过SOA提供接口访问。

架构规划•技术架构

分类	重点
技术标准	编码规范：ASP/C#/SP… 应用开发指南：Web/App/WebService/WinService 公共资源说明：UI/服务API/公共类库/基础框架…
基础框架	基础类库：日志、缓存、安全… 模式框架：MVC、AOP、ORM
基础平台	ESB平台：ESB/MSMQ/AUTOSEND/StatusService 治理平台：产品/子系统/应用/配置/部署… 监控平台：监控/预警/日志/报表… 管理平台：门户/项目/需求/设计/开发/测试/发布… 开发工具：开发时生成代码/运行时生成程序…

目录

架构实施

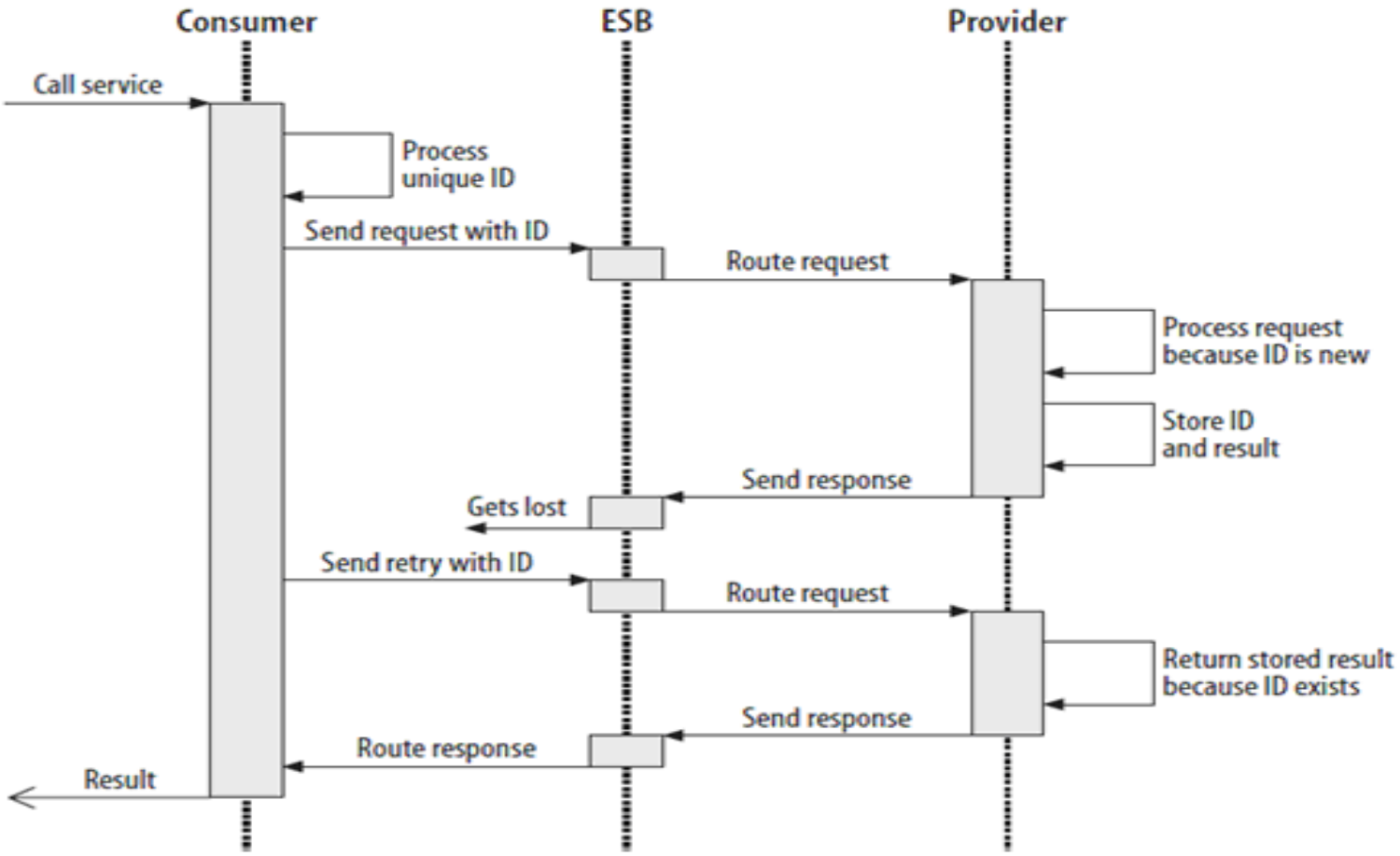
实施方面	实施重点
应用架构	SOA项目 一期：公共业务平台 二期：生产+产品 三期：订单+预订
数据架构	主数据管理项目 一期：基础主数据 二期：实体主数据 三期：产品主数据
技术架构	代码库项目 一期：技术标准/资源库 二期：开发框架 三期：二次开发平台 其他：ESB平台改造/治理平台改造/管理平台改造/监控平台

架构实施•SOA项目•规划

SOA一期 第一阶段 项目	规模	开始时间	结束时间	试运行结束	关联系统改造
总体项目	大	2010.12	2011.06	2011.06	对以下项目提供需求和设计的来源和依据，并进行总体把握和控制
基础平台	中	2010.03	2011.06	2011.06	**机票产品**：使用新的缓存平台
保险平台	大	2010.12	2011.05	2011.06	**机票预定**：查询可出保情况； **结算系统**：接收保险结算单； **机票生产**：出、退票完成后，将状态返回机票订单子系统； **机票订单**：发起出、退保请求，接收保险返回的状态。
支付平台	大	2010.12	2011.05	2011.06	**酒店预定**：接入统一的支付页面 **酒店订单**：提供状态服务，接收收退款完成或失败的状态。
配送平台	大	2010.12	2011.05	2011.06	**机票生产**：打印行程单结束后，在部分点将状态返回机票订单子系统； **机票订单**：部分发起、修改、取消配送单； **支付平台**：接入配送单查询接口，并将完成状态回置到配送系统。
结算平台	大	2010.12	2011.04	2011.05	**财务平台**：提供接收应付、实付流水的接口； **保险生产**：发起保险结算单； **机票生产**：发起机票结算单。
财务平台	中	2011.01	2011.04	2011.05	**结算平台**：发起应付、实付流水，本次只包含酒店类别流水的两个点

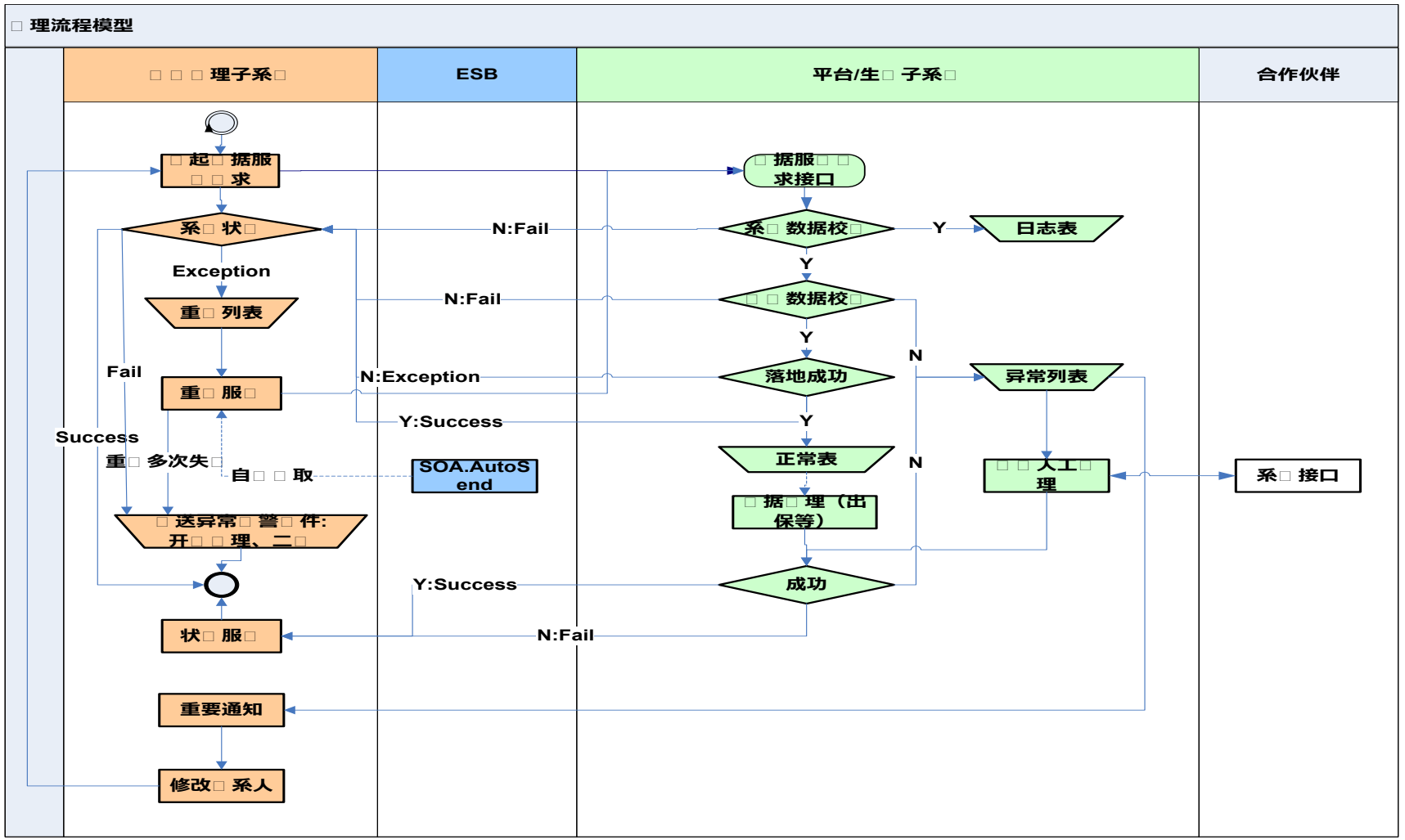
架构实施•SOA项目•幂等服务

所有单据服务和状态服务都必须实现幂等性，这是实现数据一致性的基础。在服务的上下文环境中，它意味着对同一服务调用的多次递交/处理不会引发问题。下图是对幂等服务的图解。

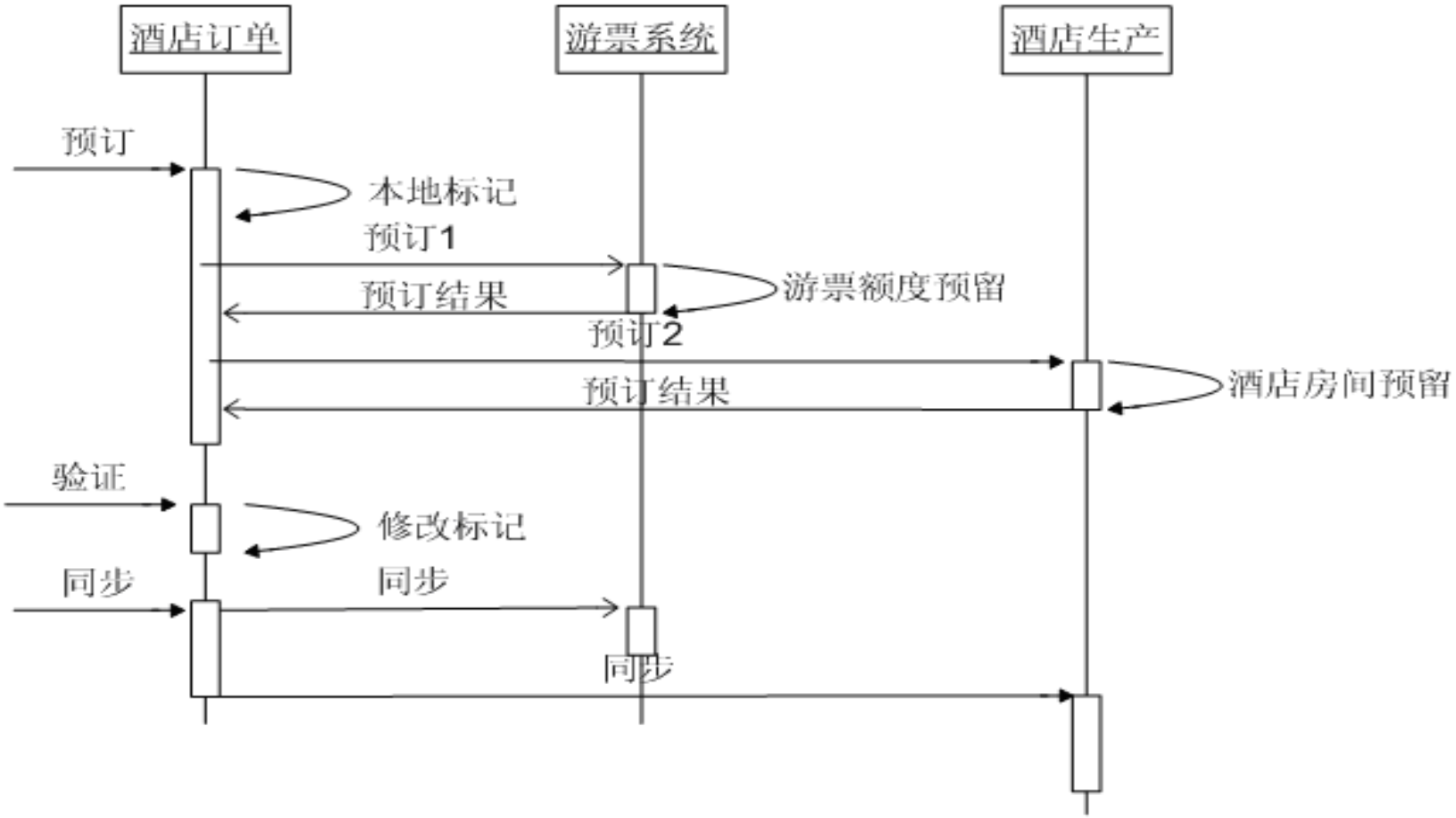


架构实施•SOA项目•交互模型

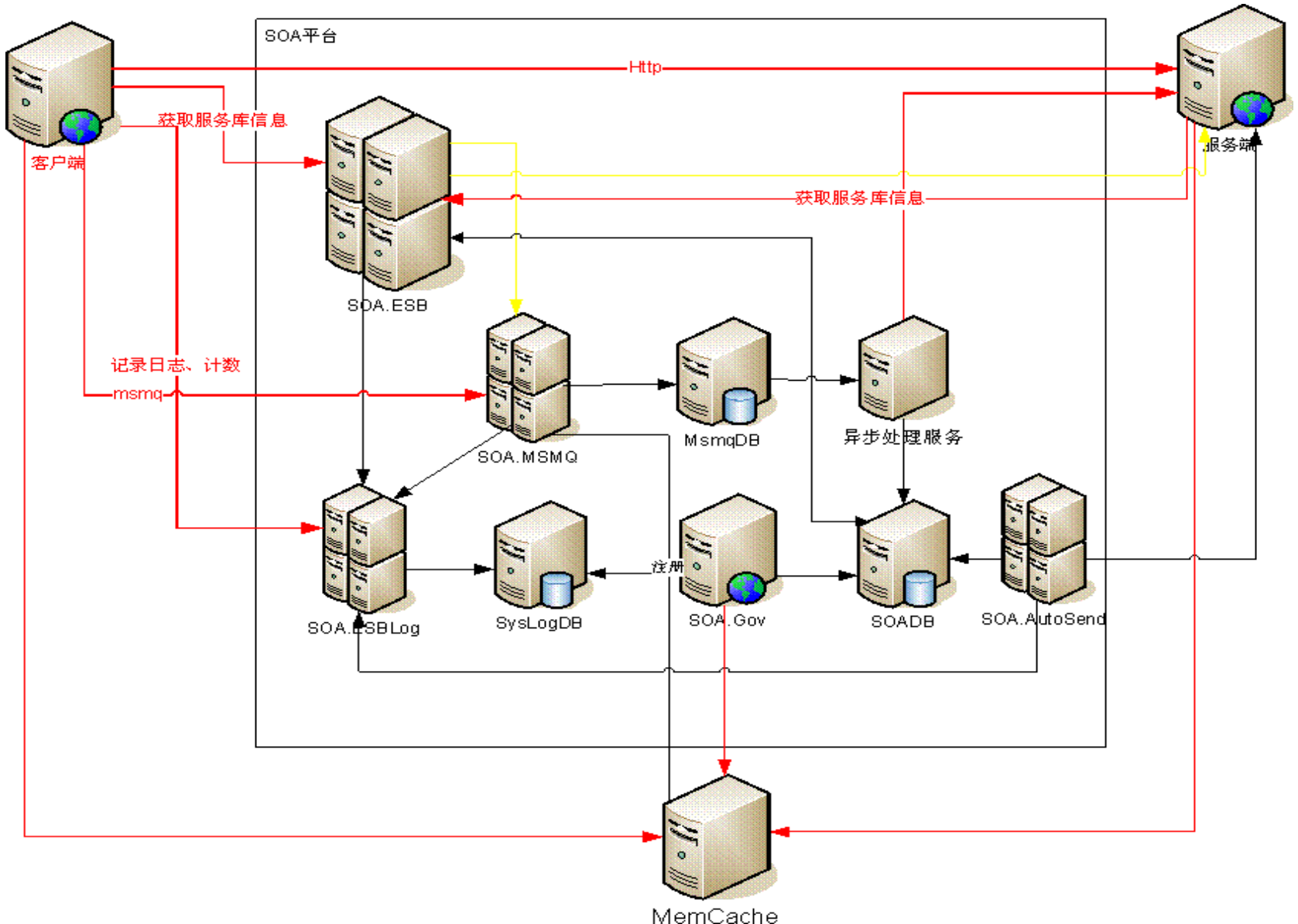
系统解耦带来的一个直接问题，就是系统之间的交互增加。在此过程中，出现异常情况应该如何处理，成为一个需要解决的重点问题。



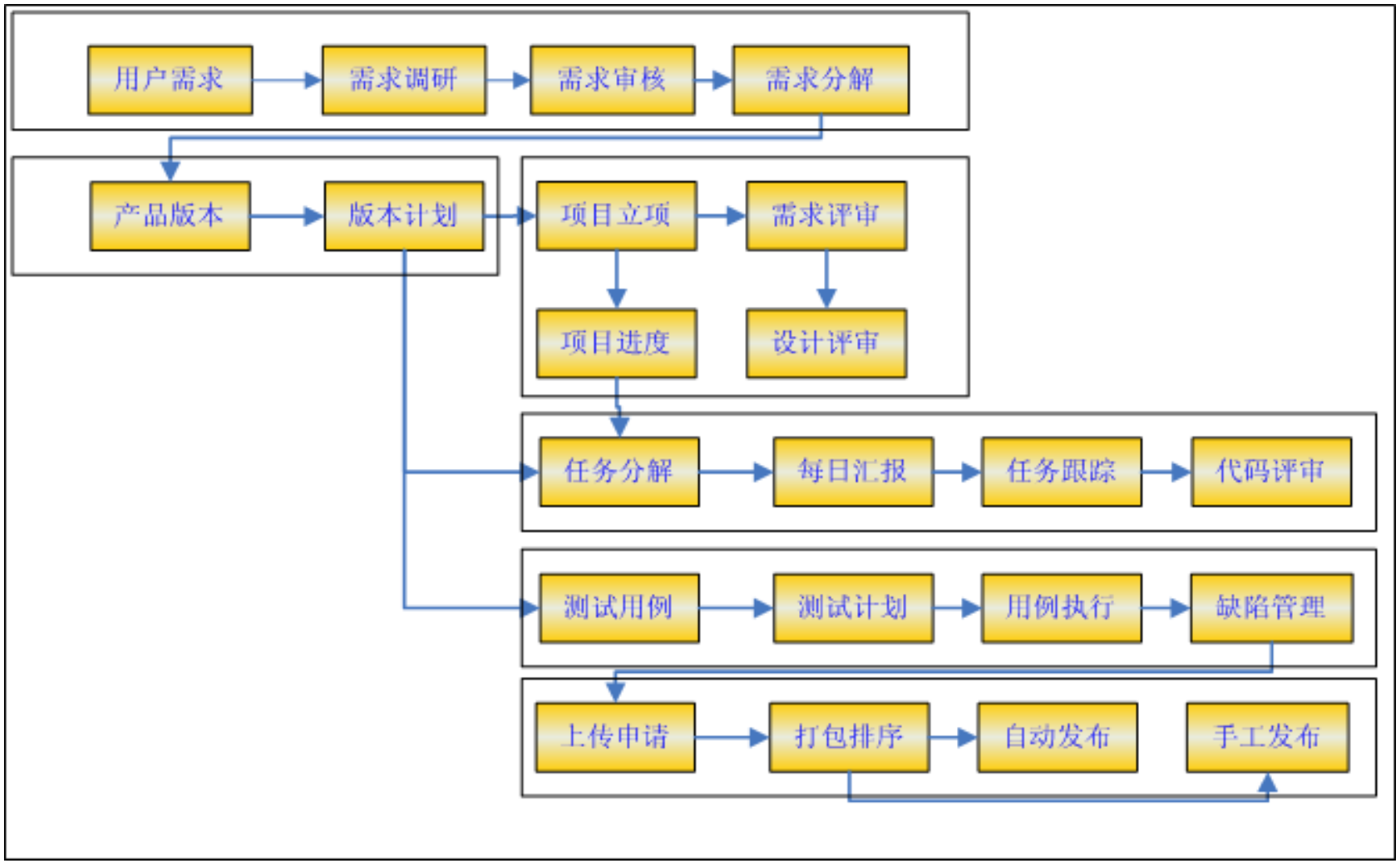
架构实施•SOA项目•分布式事务



架构实施•其他项目•ESB平台



架构实施•其他项目•管理平台



架构实施•其他项目•治理平台

系统	模块	功能
治理平台	应用架构	产品
		子系统
		应用
		模块
		功能点
		服务
		服务依赖关系
	数据架构	数据库
		数据字典
		主数据列表
		数据分发路径
	部署架构	服务器
		域名
		站点
		虚拟目录
	配置管理	配置类别
		配置项
	治理流程	应用注册
		数据注册
		部署注册
		配置注册
	数据服务	获取以上各类数据

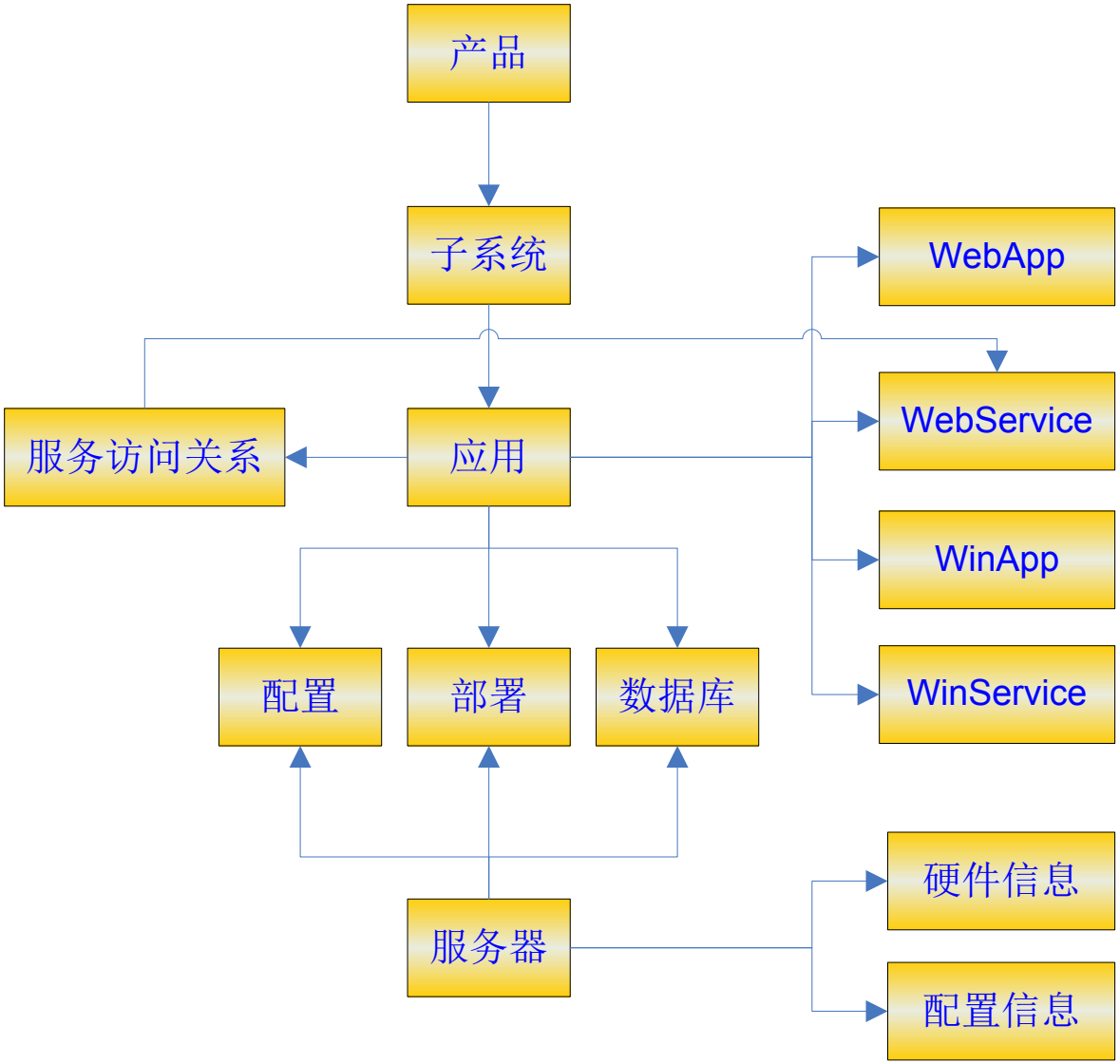
架构实施•其他项目•监控平台

系统	模块	功能
SOA.Monitor	监视	监控CPU、内存
		监控应用程序池、SP、心跳、Web页面
	预警	邮件预警通知
		短信预警通知
	处理	回收应用程序池
		重启Windows Service
		重启Windows Application
		重启IIS
ESB.Monitor	监视	监视ESB计数器
		监视ESB转发请求失败率
	预警	邮件预警
WebMonitor	监视	站点上传检测
		站点巡检监控
		数据表监控
		MSMQ 监控
		Windows Service监控
	预警	邮件预警
		短信预警
		BB 预警
	处理	WMI远程回收应用程序池
		WMI远程重启IIS

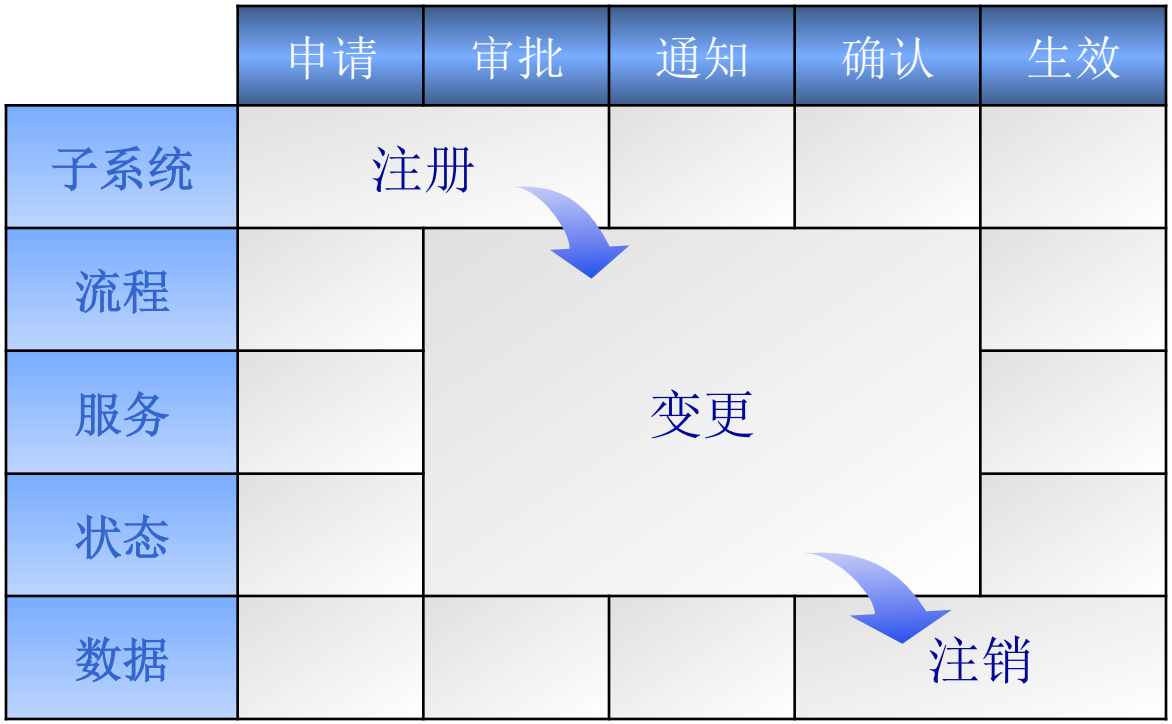
系统	模块	功能
服务器性能监控	监视	CPU 性能指标
		Memory 性能指标
		Disk 磁盘空间
		运行性能报表
		I/O 指标(数据库)
		锁监控(数据库)
	预警	预警
治理平台	日志	SOA 请求日志
		SOA 平台/失败日志
		SOA 单据状态日志
		TraceLog 日志
		AspErrorLog 日志

目录

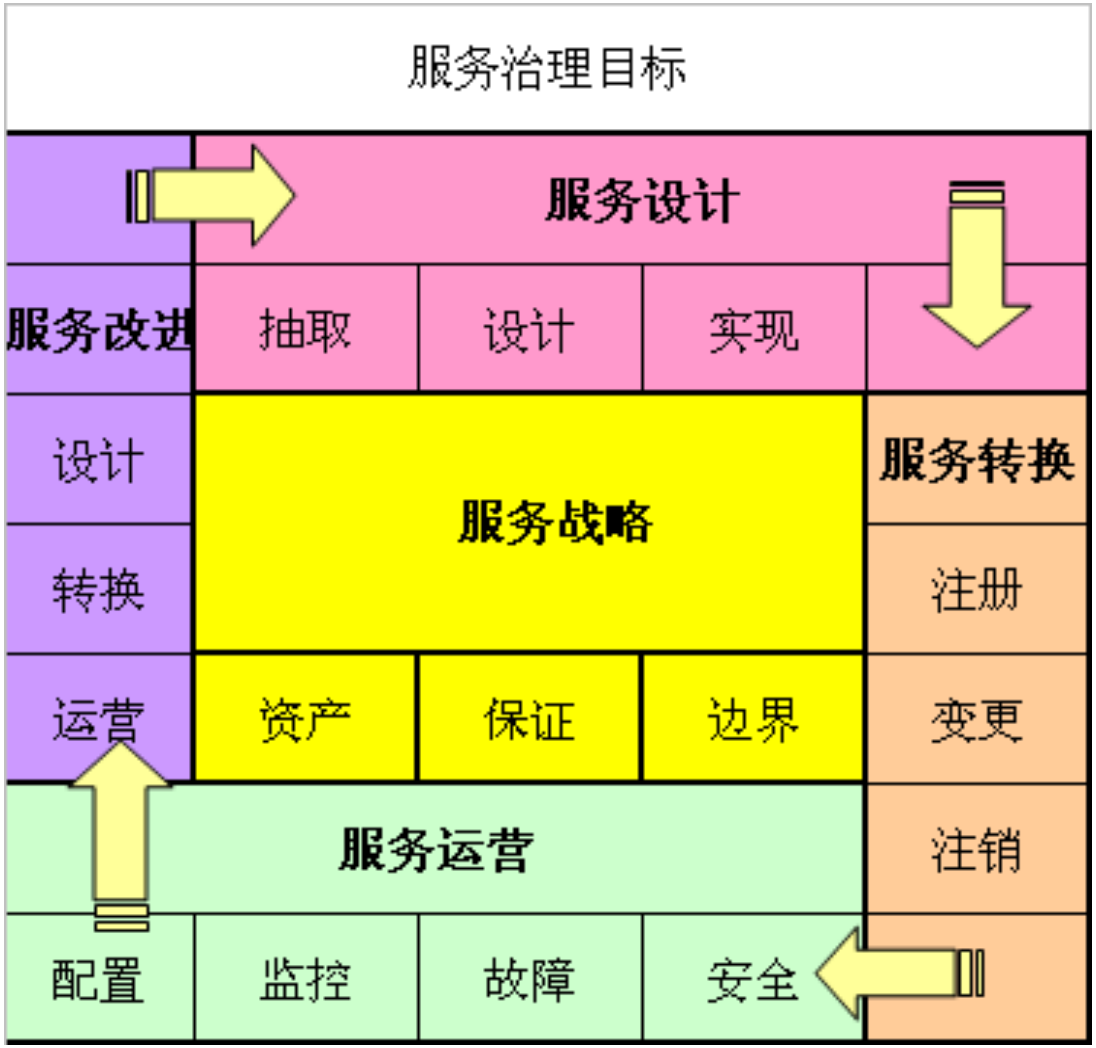
架构治理•范围


产品
子系统
WebApp
WebService
WinApp
WinService
服务访问关系
应用
配置
部署
数据库
服务器
硬件信息
配置信息

架构治理•流程



架构治理•服务治理


服务治理目标
服务设计
服务改进
抽取
设计
实现
设计
服务战略
服务转换
转换
注册
运营
资产
保证
边界
变更
服务运营
注销
配置
监控
故障
安全

广告时间

岗位	系统架构师
要求	1.计算机相关专业本科以上学历; 2.具有5年以上大型电子商务网站和互联网公司核心业务系统架构设计经验，精通业务抽象和模型设计; 3.掌握业界企业架构方法，擅长面向服务的大型企业或互联网系统的规划与设计，有很强的风险意识; 4.深厚的架构与规范文档撰写能力,有较强的主动性与责任性，良好的表达与宣讲能力; 5.有成熟的项目组织协调经验，具有较强沟通、协调组织和推进能力; 6. 知识面广，思路开阔，学习能力强；团队合作精神好，自我驱动，沟通交流能力强; 7.较强的表达和沟通能力；工作认真、严谨、敬业，能够协调其他应用架构师推进架构改进项目。
职责	1.参与企业级应用架构规划，以及建设路线图的制定，参与各领域架构建设的推进; 2.重点参与携程SOA改造项目的架构方案设计与规划以及实施的推进; 3.研究业界的技术发展、评估外部技术与解决方案。
岗位	网站架构师
要求	1.本科及以上学历，三年以上大型网站系统架构、设计、实施与运维经验，并具有较强的技术前瞻性; 3.熟悉各种架构模式、设计模式，以及.Net领域的新技术; 4.熟悉网络安全、性能、瓶颈分析；熟悉大型系统的安全性、负载均衡、数据容错、备份恢复的设计; 5.较强的学习欲望和能力，沟通表达和实践能力，能够协助业务开发组进行新技术的推广和应用; 6.具良好的敬业精神、沟通能力和团队合作精神、高度责任感、敬业精神和工作压力承受能力。
职责	1.参与网站架构的总体规划及不断优化；参与制定网站技术架构的具体方案; 3.根据业务开发组的要求，提出技术方案；分析解决系统性能瓶颈等疑难问题; 4.对系统的异常问题进行总结，并进行知识分享。

广告时间

岗位	分布式系统架构师
要求	1.大学统招本科以上学历，计算机相关专业； 2.五年以上相关工作经验，精通Java或.Net，以及Web的开发和应用；两年以上大规模高并发访问的分布式应用架构设计和核心开发经验； 3.精通分布式系统设计及应用及多线程及高性能程序设计与编码；精通Linux 操作系统和大型数据库； 4.精通常用的分布式开源框架及相关技术，如Lucene、Memcached、hadoop、HBase等，有实际工作经验者优先； 5.精通各种常用数据结构和算法，掌握多种架构设计模式,精通高性能服务器架构，应用集成、大规模分布式系统设计, 海量数据处理； 6.有良好的团队意识和协作精神，有较强的内外沟通能力；做事主动、积极、踏实、细致； 7.有推荐引擎开发、设计经验者优先。
职责	1.负责分布应用系统的技术研究、架构设计和优化改进； 2.负责内部搜索引擎平台的架构和开发； 3.指导软件工程师开发工作。
岗位	**推荐引擎高级软件工程师**
要求	1.有2年以上搜索引擎或推荐引擎领域设计/开发实现工作经验； 2.对全文检索和推荐引擎有深入了解；了解至少一类推荐引擎开源框架，如Taste/Mahout等； 3.有较强的学习与自学能力； 4.对算法和数据结构有深入的理解，对中文自然语言处理技术有深入了解优先； 5.有机器学习和数据挖掘系统经验优先；有互联网SNS系统设计/开发经验者优先； 6.有良好的合作性和沟通能力和较强的自我驱动能力。
职责	1.设计、开发、维护公司内部的推荐引擎系统； 2.参与搜索引擎以及其他分布式计算平台的开发和设计； 3.研究互联网领域推荐相关技术的发展趋势。

谢 谢

Email:lyvvvv@sina.com

微博:@帮主刘


QCon
北京站·2012年4月18~20日
www.qconbeijing.com（11月启动）
QCon杭州站官网和资料
www.qconhangzhou.com
全球企业开发大会
INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE