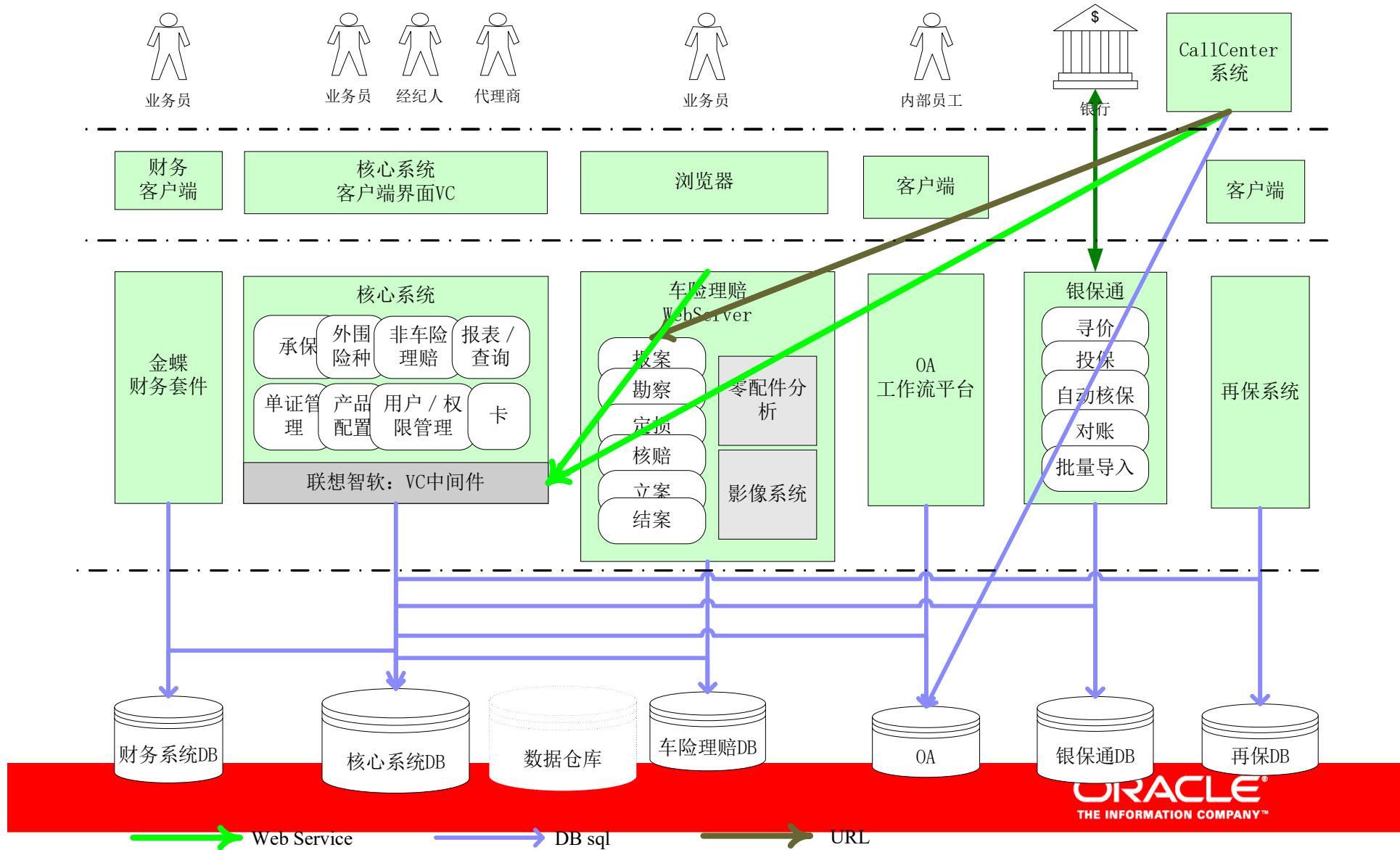


案例分析

- 保险业务系统整合：某保险公司**SOA**实施

系统现状



现状评析

- 期望与需求
 - 延伸业务渠道，提升客户服务，加强风险管理，产品智能化，优化流程管理，应对市场竞争。

 - 没有形成企业级的IT项目标准建设流程
 - 需求管理等配套的IT系统建设标准缺少
 - 项目建设处理比较传统的建设模式

 - 核心系统承担大部分业务支撑能力
 - 基本按业务功能条线组织和实现应用系统
 - 应用之间的功能耦合于数据结构或接口表
-
- ```
graph TD; Business((业务)) --- XX[XX 保险]; IT((IT)) --- XX; ProjectConstruction((项目建设)) --- XX; Architecture((架构)) --- XX; ApplicationSystems((应用系统)) --- XX; Security((安全)) --- XX;
```
- 业务支撑能力
    - 通过构建统一的企业服务平台实现全面的业务支持
    - 在产品智能化，客户关系与服务，销售线业务支撑方面能力不够。
  
  - 系统架构不统一
    - 架构灵活性不够，通过改造核心系统实现企业整体架构的灵活性和快速反应能力
  
  - 用户管理
    - 远程通过VPN使用动态口令卡接入企业业务系统。

# 挑战分析

| 问题   | 原因       | 影响         | 建议        |
|------|----------|------------|-----------|
| 组合业务 | IT架构     | 不能推出相应业务   | 改造架构      |
| 数据   | 数据架构     | 丢失企业IT资产   | 建立业务模型    |
| 维护   | IT架构     | 工作量大，成本高   | 改造架构      |
| 流程   | 代码实现业务流  | 不能快速回应市场变化 | 建立企业流程平台  |
| 产品配置 | 产品定义不完整  | 新产品推出速度慢   | 实现产品定义模型  |
| 数据交换 | 交换方式     | 业务扩展困难     | 交换以功能方式提供 |
| 用户管理 | 应用系统独立建设 |            | 统一用户管理    |

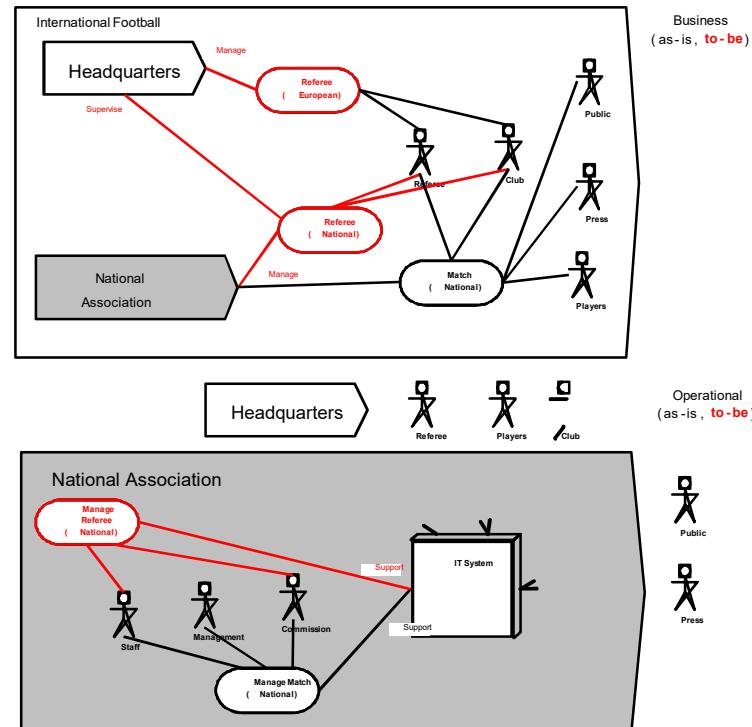
# 分析总结——业务策略与流程

## Problem Areas

- 业务策略中急需的业务能力扩展只能通过新系统的开发去实现，无法实现对已有业务功能的重用。因此，随着系统的增加，系统复杂性急剧上升；
- 业务策略中对业务流程的灵活性提出了较高的要求，现系统应对这些业务流程变化和政策法规调整不灵活，现有实现方式造成系统刚性很大，需要通过大量代码修改实现；
- 业务策略中对售前、售中、售后、客户服务等跨系统业务流程整合有需求，现系统缺乏一致的实现方式；
- 数据模型对业务实体描述简单、维度少，尤其没有客户实体的描述，造成业务分析困难，不适应当前业务发展。系统间共享数据困难，开发工作量巨大。

## Best Practices

- 在愿景环境中，IT系统被视为企业业务流程的具体体现，新业务的实现是依靠扩展业务流程，不是建立孤立的系统和应用。同时随着系统的成熟，业务随之改善。



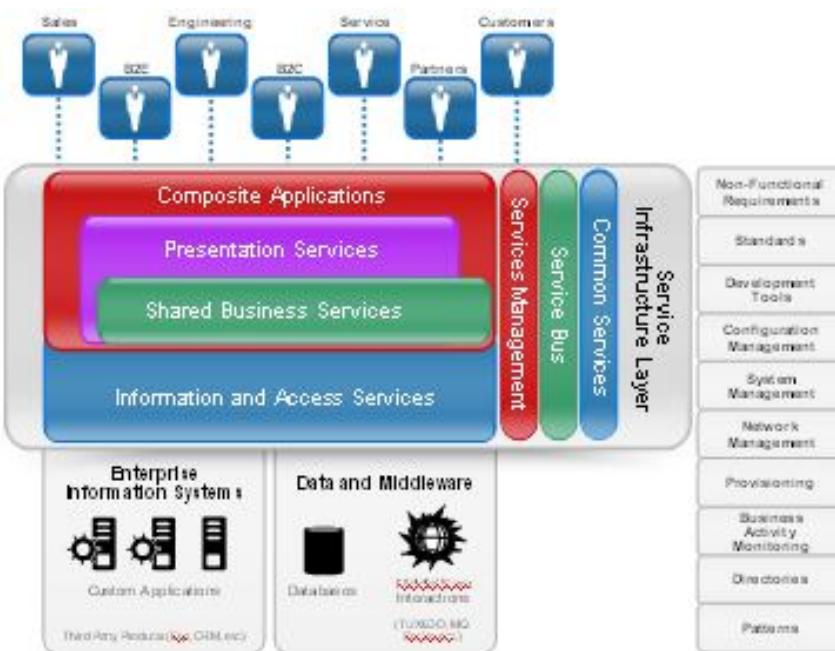
# 分析总结——IT架构

## Problem Areas

- 缺乏企业视角：目前系统的建设方式仍然是功能集合（或部门）方式的，如核心系统、车险理赔、银保通等。没有考虑如何按照标准方式去定义、配置、监控和管理对业务功能的访问。
- 缺乏业务视角：企业员工需要多种应用系统去完成日常工作，造成数据的重复记录、缺乏的全部流程的可视性和控制。
- 没有基于服务，缺乏标准：当前IT系统的实现技术种类较多，包括（J2EE、C / S、定制套件等），接口技术种类较多，包括（中间表、Webservice、URL联接、专用接口等）。对于各系统的互联，没有基于服务进行规划。
- 缺乏支持功能共享和标准化的基础设施：当前IT系统还没有可以提供功能路由、协议转换、功能管理能力的IT基础设施。

## Best Practices

- 支持和标准化现有企业活动需要一个面向服务的系统架构。该系统架构确定了如何在符合业务和用户利益的情况下交付和部署业务功能。



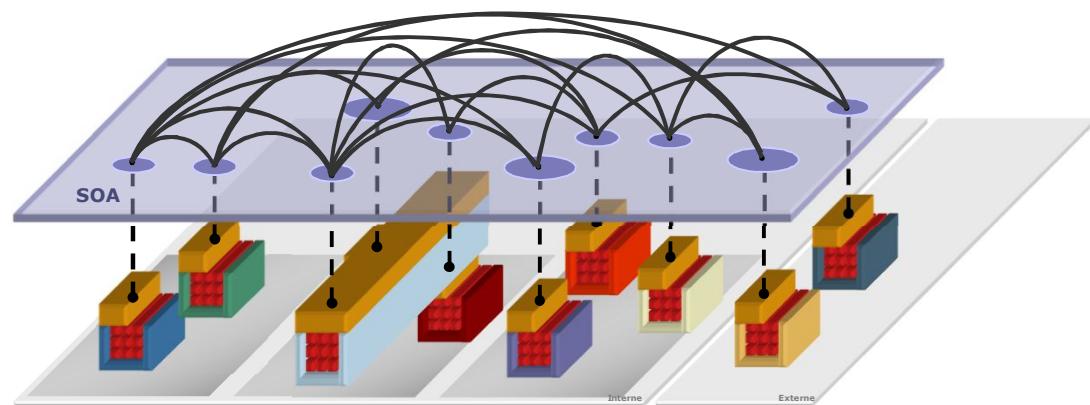
# 分析总结——业务服务

## Problem Areas

- 接口或可以共享的业务功能没有明确定义合约、接口和实现，使得系统互联与功能的实现紧耦合在一起，无法实现共享。
- 大量使用中间表作为IT系统交换信息的手段，使得数据和业务逻辑紧密耦合，系统扩展困难。
- 在业务逻辑的实现上，大量使用数据库存储过程，破坏了数据封装和增加了数据耦合度。

## Best Practices

- 服务的定义应基于标准、基于重用。
- 服务类型包括：
  - 基础设施服务
  - 公用服务
  - 共享业务服务
  - 数据服务
  - 显示服务
- 服务的提供和消费应该通过统一的SOA服务集成平台完成。



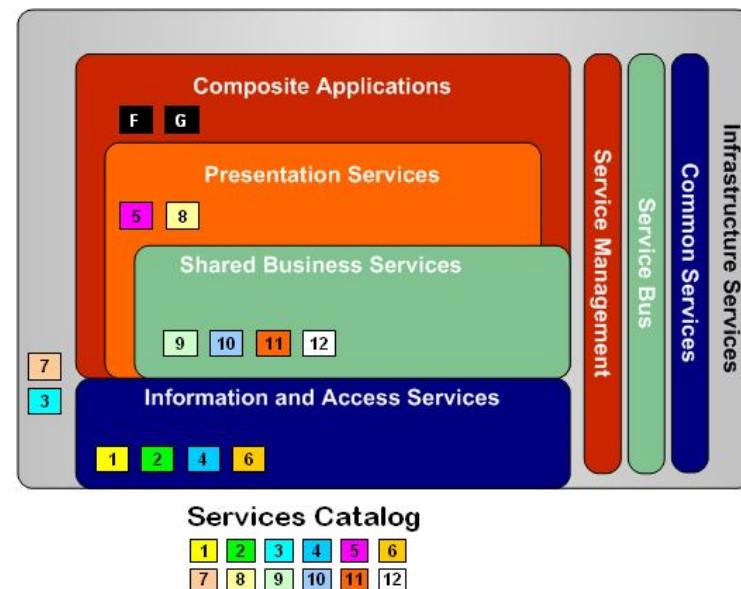
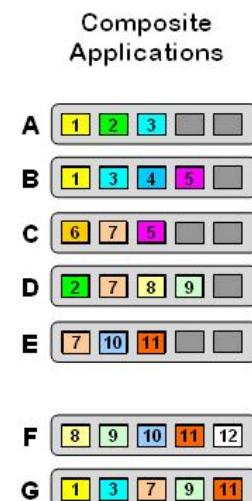
# 分析总结——项目和应用

## Problem Areas

- 目前已经规划了电子商务项目、数据仓库项目，以满足紧迫的业务要求；
- 在完全更新核心系统之前，系统存在过渡期。在此过渡期间，并行存在若干实施的项目，需要仔细规划这些项目实施的优先级和路线图。

## Best Practices

- 实施完整的SOA参考架构规划。其中的一项重要内容就是分析业务与IT的差距，制定一个完整的、注重实效的SOA项目线路图。



# 差距分析（一）：架构不统一

| 问题综述                                                          | 现象                                                                                                                                                                                                                                                  | 导致的原因                                                                                                                                                                                     | 对业务能力的影响 | 对开发能力的影响 | 对维护能力的影响 |
|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|----------|----------|
| 架构问题：<br>1、设计模型和数据模型不统一<br>2、架构的扩展困难，架构的可靠性差<br>3、系统维护和更改成本昂贵 | 1、系统有多个“烟囱式”的应用，无统一的应用架构；数据分散，无统一的数据模型，系统存在大量的冗余数据；渠道分立，没有整合<br>2、用户数大增或峰值访问时间，系统性能严重下降甚至不能提供服务，系统的服务质量不稳定<br>3、系统的资源很难或无法重用，系统不容易修改及扩展，增加了开发、维护及集成工作；系统中的模块没有明确的职责划分，并且互相之间紧密耦合，甚至会导致循环依赖；可扩展性差，代码的可修改性低；因为没有元数据的定义，所以多个应用没有统一规范的数据Schema，给SOA留下隐患 | 1、现有架构设计依照功能导向进行设计，耦合紧密，缺少统一的设计模式和开发规范未形成与SOA相适应的开发方法和项目管理方法。<br>2、现有架构没有考虑用户数量增长的情况下怎样满足当前系统的性能要求；当前架构缺少系统崩溃后的错误恢复机制。<br>3、没有规范系统间的交互方式，完全没有考虑系统间事务的交互；系统较少考虑分层设计，层与层之间职责不清，耦合紧密；系统没有元数据 | 长远影响，大   | 目前影响较大   | 小        |

# 系统交互

## 主要内部接口

| 子系统---调用方 | 子系统----提供方 | 接口内容     | 接口方式            |
|-----------|------------|----------|-----------------|
| 核心系统      | 财务系统       | 保单信息     | 中间表, 定时         |
| 财务系统      | 核心系统       | 付款信息     | 中间表, 定时         |
| 车险理赔      | 核心系统       | 查询保单信息   | Web service, 实时 |
| 核心系统      | 车险理赔       | 回写理赔案件信息 | 中间表, 定时         |
| 核心系统      | 再保系统       | 保单信息     | 中间表, 定时         |
| 核心系统      | 银保通        | 保单信息     | 中间表, 每日定时       |

## 主要外部接口

| 子系统---调用方   | 子系统----提供方 | 接口内容 | 接口方式            |
|-------------|------------|------|-----------------|
| 核心系统        | 制卡系统       | 制卡文件 | 文件              |
| Call center | 核心系统       | 保单信息 | Web service, 实时 |
| Call center | 车险管理系统     |      | URL界面引用         |
| Call center | OA系统       |      | 中间表, 定时         |
| 银保通         | 银行         | 保单信息 | 专用接口, 实时        |

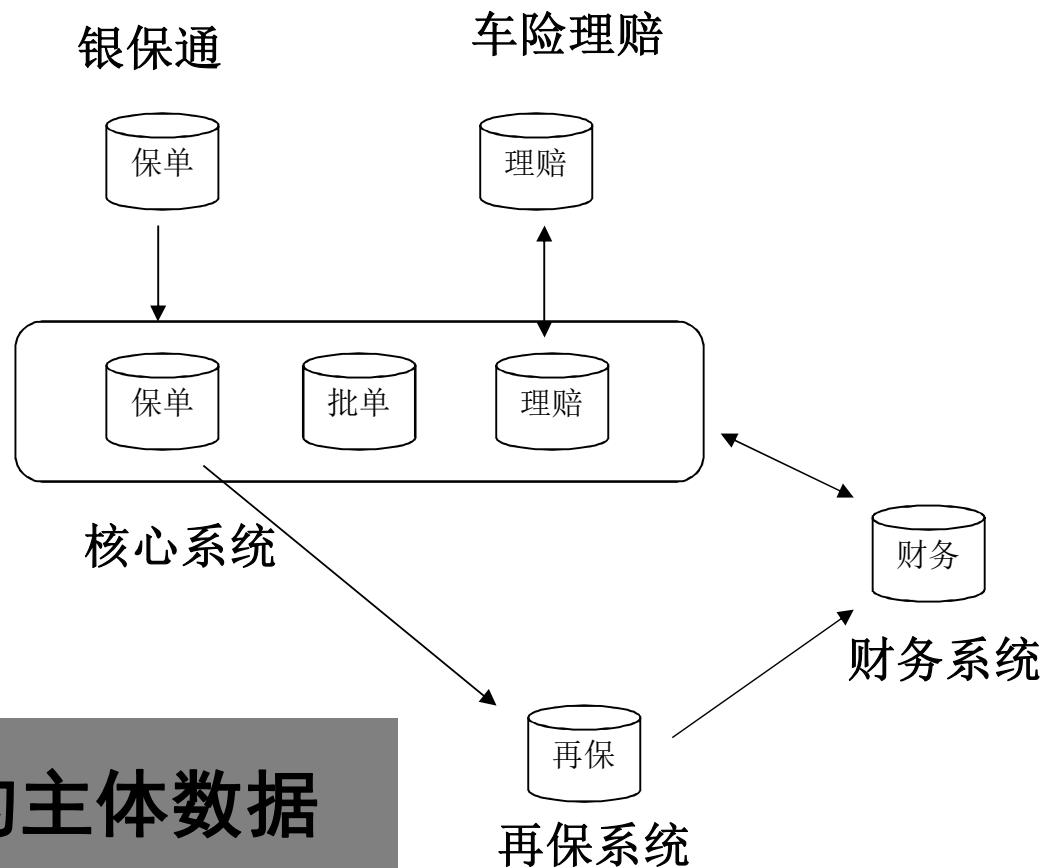
业务流程与  
业务逻辑与  
数据绑定的  
太紧，耦合  
性太高，维  
护成本大。

# 数据

- 数据分布

- 保单数据以为核心为中心，银保通与卡系统及脱机的出单系统，其他渠道系统的保单批量倒入到核心的数据库
- 理赔数据也以核心为中心，车险理赔系统批量倒入赔案及理赔数据。
- 财务数据以财务系统为主
- 目前没有客户与产品的相关主体数据。

缺少客户与产品的主体数据



## 差距分析（二）：流程管理问题

| 问题综述                                                           | 现象                                                                                                                | 导致的原因                                                                                                         | 对业务能力的影响                          | 对开发能力的影响 | 对维护能力的影响 |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|-----------------------------------|----------|----------|
| <b>业务工作门户问题：</b><br>1、界面不友好，用户使用不方便<br>2、页面很难重用<br>3、页面维护工作量巨大 | <b>1、用户必须在不同功能界面交换操作，体验极差</b><br><b>2、单个应用，无法适应今后专业化服务需求，部署和升级的依赖性很大</b><br><b>3、页面不容易修改及维护，出现问题不容易定位原因，工作量极大</b> | <b>1、仅提供单个应用的界面，没有企业级的门户规划，没有统一的界面样式</b><br><b>2、无统一的页面设计框架，缺少通用的页面控件，页面元素重用性不好</b><br><b>3、界面程序中过多涉及业务逻辑</b> | 直接影响，小                            | 长远影响大    | 大        |
| <b>流程维护问题：</b><br>1、业务流程更改困难<br>2、任务处理主要依赖用户的自觉性               | <b>1、现有业务流程很难适应新业务的拓展</b><br><b>2、用户的自觉性决定了业务能力</b>                                                               | <b>1、用状态的机制实现简单的业务流程</b><br><b>2、现有任务是主动查询而非任务控制</b>                                                          | 长远影响，中<br>(影响体现为客户服务质量和服务客户满意度降低) | 直接影响大    | 大        |
| <b>流程分析问题</b><br>1、缺少业务人员流程建模<br>2、缺少流程仿真过程                    | <b>1、业务人员描述流程，技术人员完成流程建模</b><br><b>2、没有进行业务流程评估。</b>                                                              | <b>1、流程基于套装软改造得到，受制于流程的状态机实现模式</b><br><b>2、没有支持业务流程仿真的技术</b>                                                  | 大                                 | 直接影响大    | 大        |

# 差距分析（三）：安全与集成问题

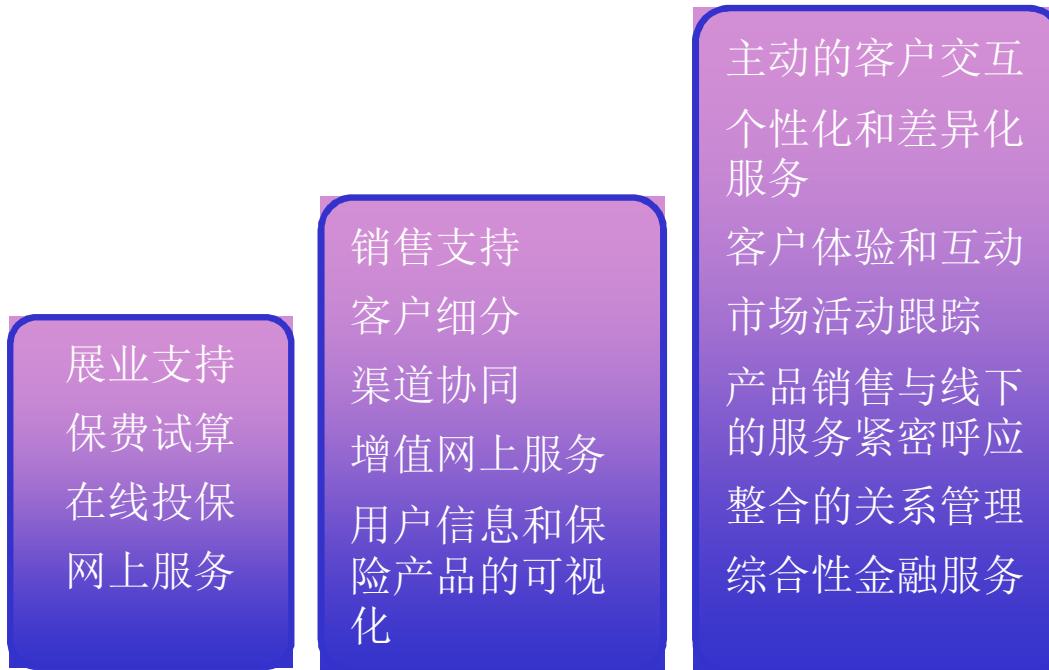
| 问题综述                                                                                                              | 现象                                                                                                                                                                                                                  | 导致的原因                                                                                                   | 对业务能力的影响 | 对开发能力的影响 | 对维护能力的影响 |
|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|----------|----------|----------|
| <b>安全管理：</b><br>1、多个应用使用不同的用户管理<br>3、操作的安全性太低                                                                     | 1、用户进入不同系统都要重新登录<br>2、企业内有多个用户管理应用，分别使用独立的用户认证；新应用的加入将导致维护成本的额外开销                                                                                                                                                   | 1、没有统一的用户认证，各个系统单独管理自己的用户认证<br>2、没有统一的用户管理                                                              | 长远影响，大   | 长远影响大    | 中        |
| <b>系统集成问题：</b><br>1、集成的依赖性较高，应用部署及测试困难<br>2、Call Center与其他系统的连接不方便且安全性低<br>3、不同应用集成时其SESSION的数据不能传递<br>4、系统缺少集成手段 | 1、通过jar包进行集成依赖性较高，应用部署及测试困难，且容易引起数据的并发问题<br>2、Call Center的用户点击URL连接系统，很不方便，很不安全<br>3、不同应用间的Session无法共享<br>4、当人力资源，财务系统等其他内部系统和外部机构系统需要与新系统有更紧密的数据和业务联系时，集成的成本高。而且现在已有要求与其他外部系统有更加紧密的业务集成的需求；不同应用中的简单数据需要通过数据库共享，效率低 | 1、未考虑内部系统的统一接口和集成方式<br>2、与Call Center的采用较简单的URL链接进行集成<br>3、系统设计时没有考虑多个系统间互操作的问题<br>4、尚未考虑其他内部系统的接口和集成方式 | 长远影响，中   | 中        | 中        |

# SOA项目切入点分析

- 核心系统功能重用
  - 展业的渠道系统
    - 电话销售
    - 电子商务
    - 代理与中介
- 系统整合
  - 企业级业务流程的梳理
    - 理赔业务流程
    - 承保业务流程
- 客户视图
  - 呼叫中心对数据的需求
  - 企业客户数据中心

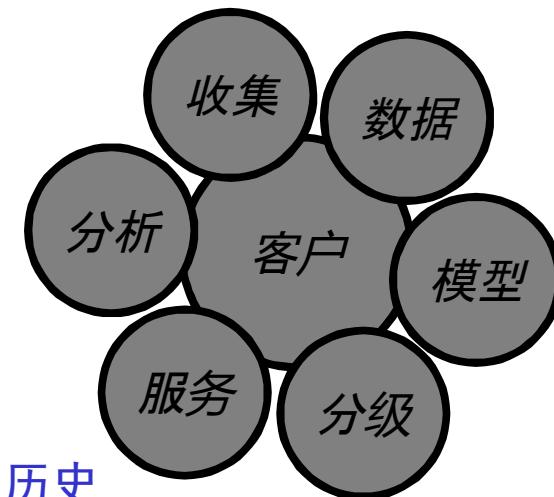
# 业务驱动力1：拓展新的渠道营销模式

- 新渠道和模式
  - 电子渠道：电子商务（网站/手机） + 电话销售

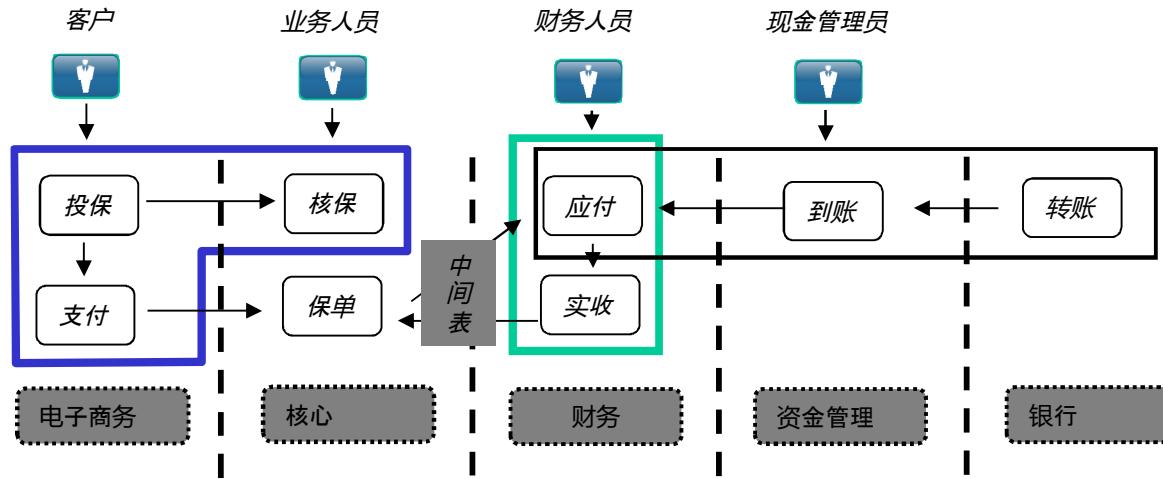


# 业务驱动力2: 客户为中心的战略

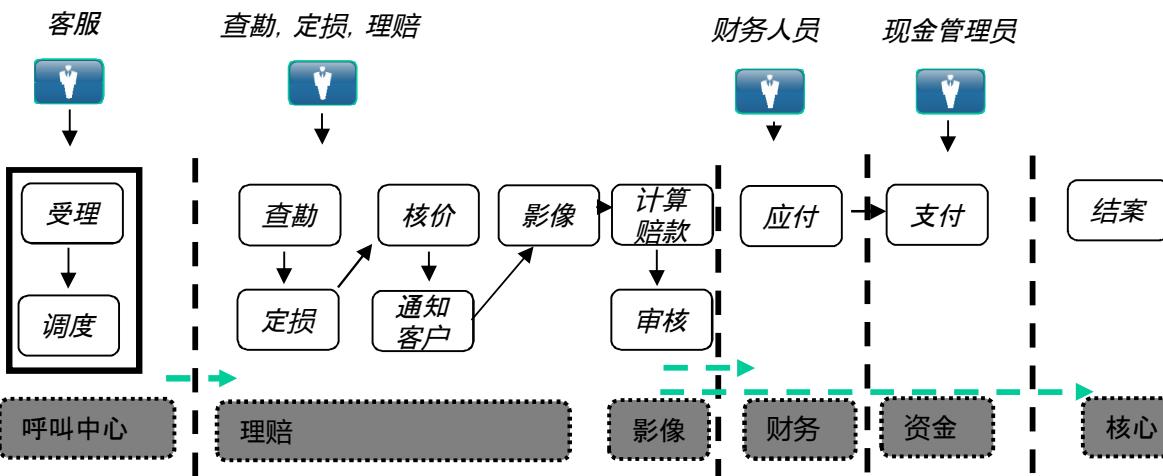
- 客户数据的共享需求
  - 通过一个客户号检索所有相关信息：接触历史、保单纪录、理赔纪录、财务信息等
  - 目前各个主要应用缺乏足够客户信息，需要采集补充客户资料（核心只有保单流水号），而呼叫中心（客服、电销）应用需要比较完整的客户资料
  - 产品定义需要考虑客户属性的分析结果，可以为客户分类、差异化营销提供依据
- 客户信息档案的管理
  - 历史保单向客户为中心的转化过程
  - 关于客户的建立元数据模型（各部门应用交互）
- 以客户为中心的流程
  - 承保流程
  - 理赔流程
- 以客户为中心的渠道整合
  - 整合营销模式需要各个渠道统一客户信息和接触历史



# 业务驱动力3：通过流程管理提高运营效率



电子支付投保流程

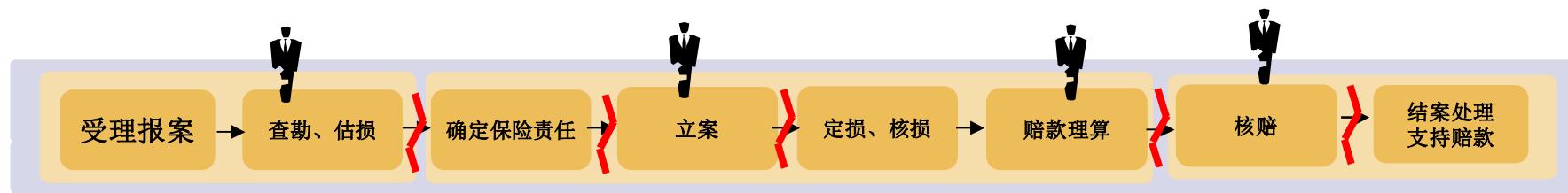


业务驱动力：  
通过流程管理和优化来提高运营效率

理赔流程

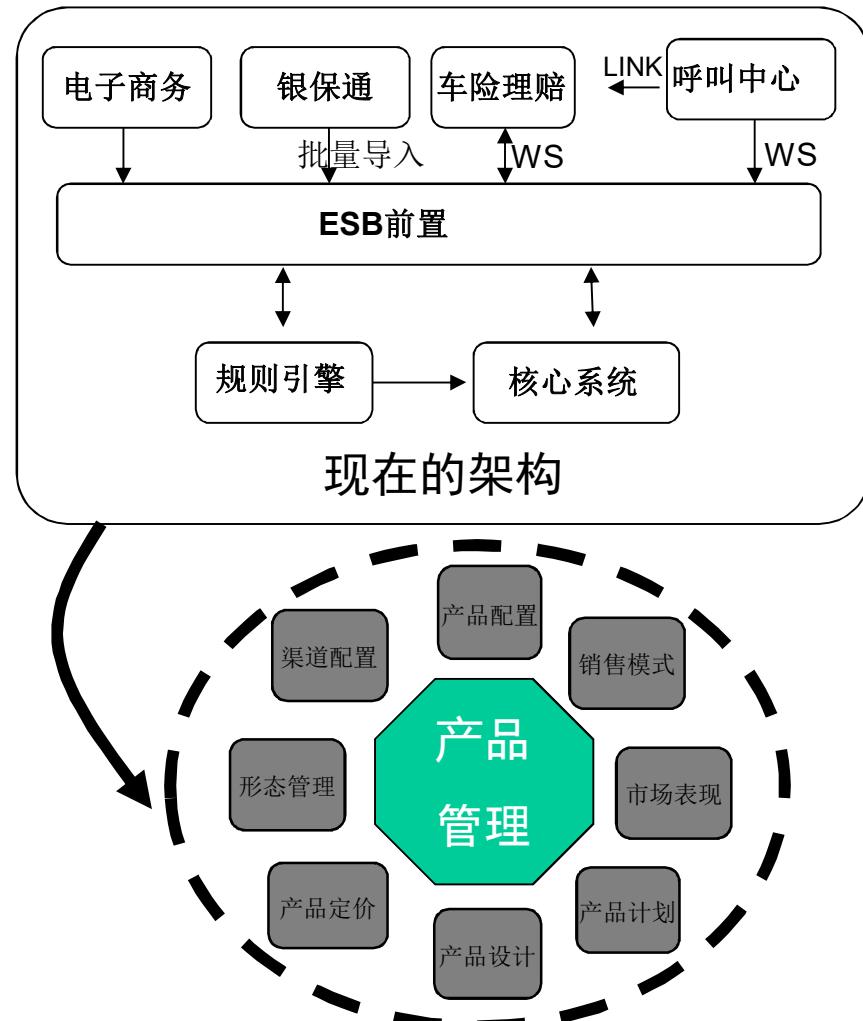
## 业务驱动力3：流程管理

- 处理流程固化、单一,对案件无法区分对待
- 理赔程序和环节过于繁杂；环节之间的任务划分清晰度不够
- 信息零散，不一致信息透明度差，决策人不能及时拿到准确的信息，难以防范重复索赔、骗赔案件的发生。
- 查勘定损理赔核批时限长，对理赔环节和理赔岗位的考核缺乏量化指标
- 许多业务流程都是人工的、多余的、不连续。理赔质量和服务水平不高



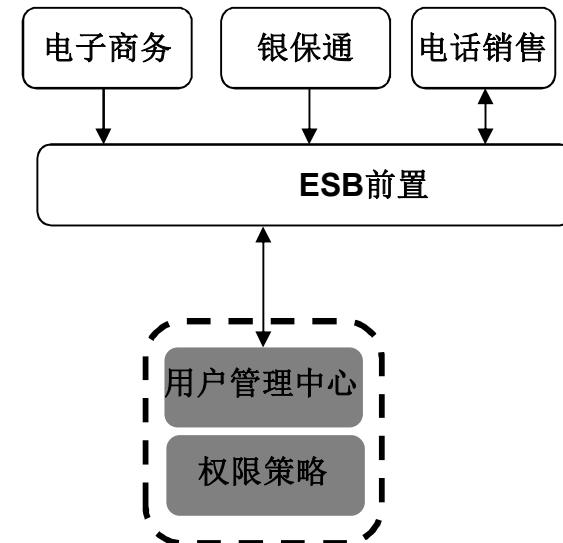
# 业务驱动力4：更灵活的产品管理策略

- 现有架构下产品涉及的过程
  - 承保规则应用（规则引擎）
  - 核保规则应用（规则引擎）
  - 保费计算（规则引擎）
  - 核心和规则的控制流程/逻辑（ALSB）
  - 渠道中需要定制产品页面要素
- 产品管理的方式
  - 现状：核心业务的产品只能增加险种，业务流程需要写代码实现，无法定制。只要核心新建一个产品，电子商务必须迅速产生相应销售应用逻辑。目前的产品定制过程：
    - 核心修改：数据、逻辑代码、中间件接口
    - 规则引擎中的承保、核保、保费规则
    - 电子商务应用中的界面
    - ESB中的服务组合（基本不变）
  - 问题：
    - 开发：没有产品开发流程控制、核心代码修改和版本更新、需要制定产品类服务的规约
    - 运营：缺少产品的运营指标数据分析



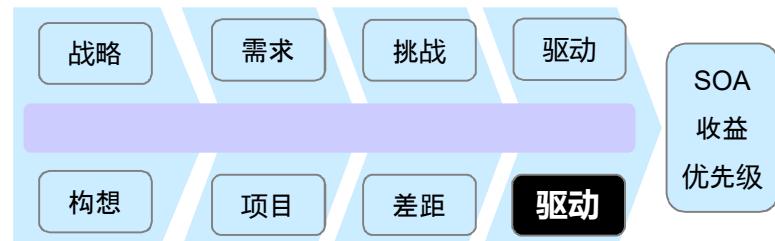
# IT驱动一：共享的用户和权限管理

- 现状：
  - 已有的系统中核心，理赔，财务，再保，保卡，银保通，呼叫中心已有独立的用户管理模块
    - 同步的用户数据：车险理赔、报表系统、核心系统
    - 跨系统的用户：核心与报表，呼叫中心与电子商务，电子商务与理赔
      - 目前采用系统间信任原则，存在安全漏洞
  - 待建的电销系统，非车险理赔系统等系统也考虑建立自己的用户管理模块
  - 权限管理模式
    - 部分系统授权到人（核心等）
    - 部分系统授权到组/机构/级别/人（车险理赔）
    - 人员变更需要权限的维护工作比较多
- 问题与驱动讨论：
  - 考虑在新建项目中共享用户管理
  - 采用一致的权限分派模式（基于角色）
  - 逐步改造现有系统过渡到企业的用户管理中心模式



# IT驱动之二：统一的数据视图

- **现状：**
  - 与渠道有关的客户接触历史，交易历史，带有渠道特性的信息（话务信息，浏览信息）各自存放在单独的渠道系统中
  - 与客户有关的保单信息，存放于核心系统
  - 与客户有关的理赔信息散落于核心系统，车险理赔系统
  - 其他与客户相关的信息也存放在不同的独立的应用系统中
- **问题：**
  - 电销系统的坐席需要很迅整并方便的获取或看到与客户相关的所有信息用于展业，包括二次销售和交叉销售等
  - 承保与核保的中台操作人员需要方便的拿到与投保标的有关的理赔历史与交易历史，以确定客户的风险情况。
- **讨论：**
  - 统一企业数据视图的实现的不同侧重：客户，产品，交易等



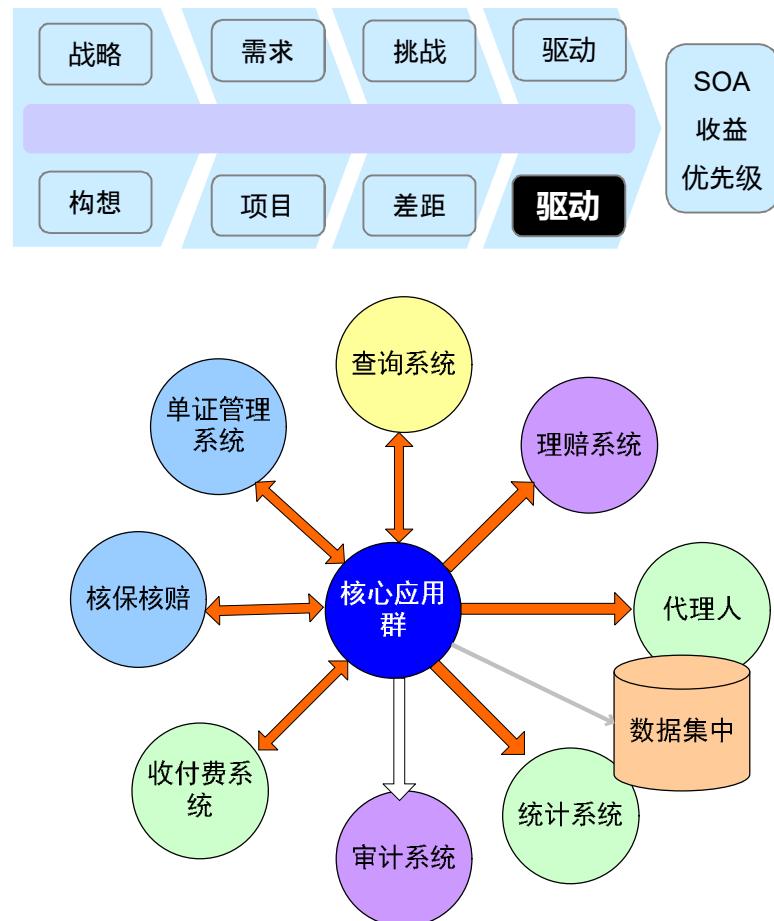
# IT驱动之三：SOA的架构管理

- 现状：
  - 项目之间的架构交流比较贫乏
  - 例如：电销系统，电子商务系统在基础架构上的需求
    - 电子商务和电销同时考虑CIF和产品管理的可能实现
    - 渠道系统及应用系统都希望实现一些公共的基础设施
      - 用户管理中心
      - 产品管理中心
      - 一致的权限管理模式
        - » 角色，资源，组
    - 各系统通过项目的集成商和开发商已经或将会再次形成的IT系统在基础架构层有重叠
  - 讨论：架构管理模式
    - 各项目架构人员的沟通机制
    - 定期的审核项目架构与企业的架构的关系、冲突，处理好企业IT架构的更迭

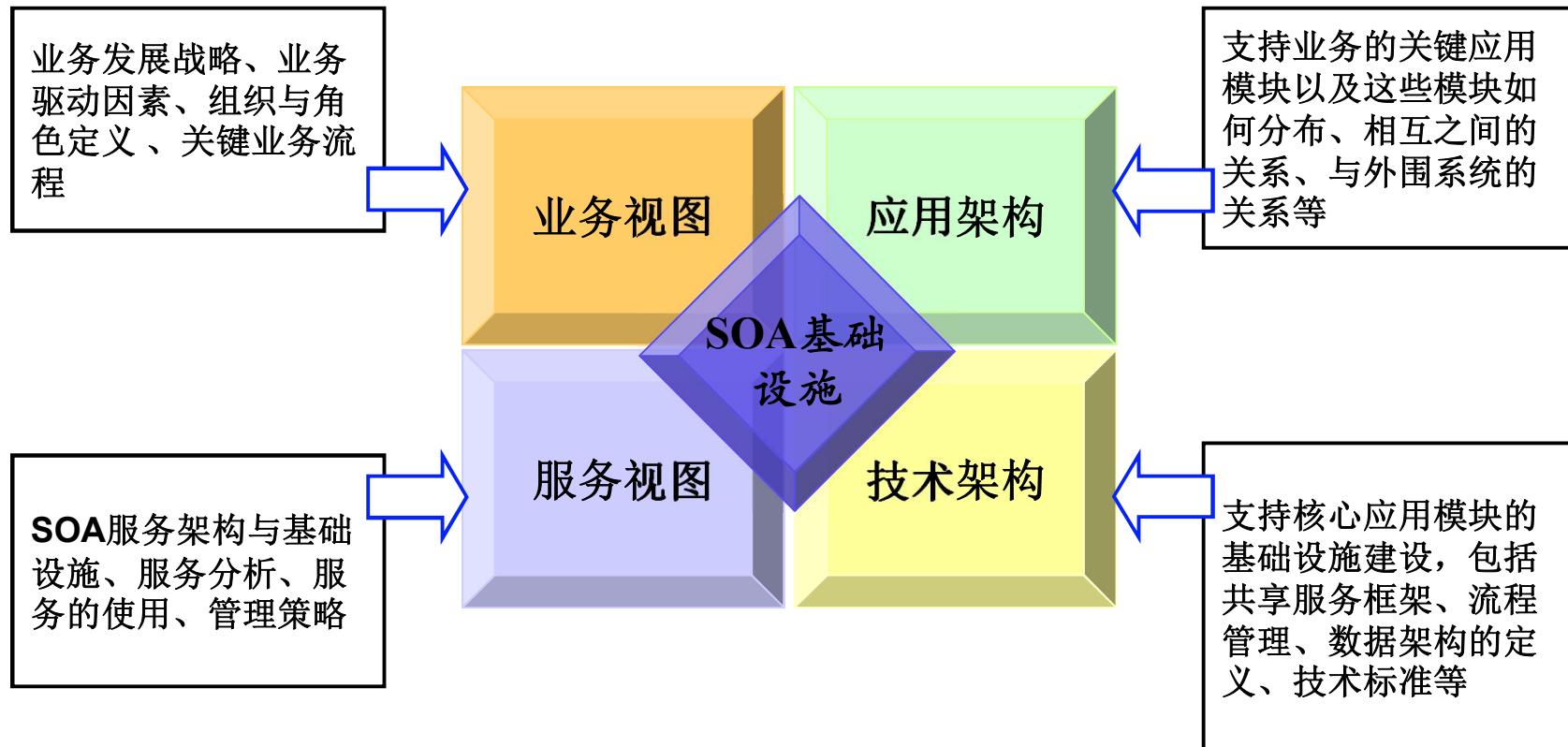


# IT驱动四：战略驱动的核心应用群

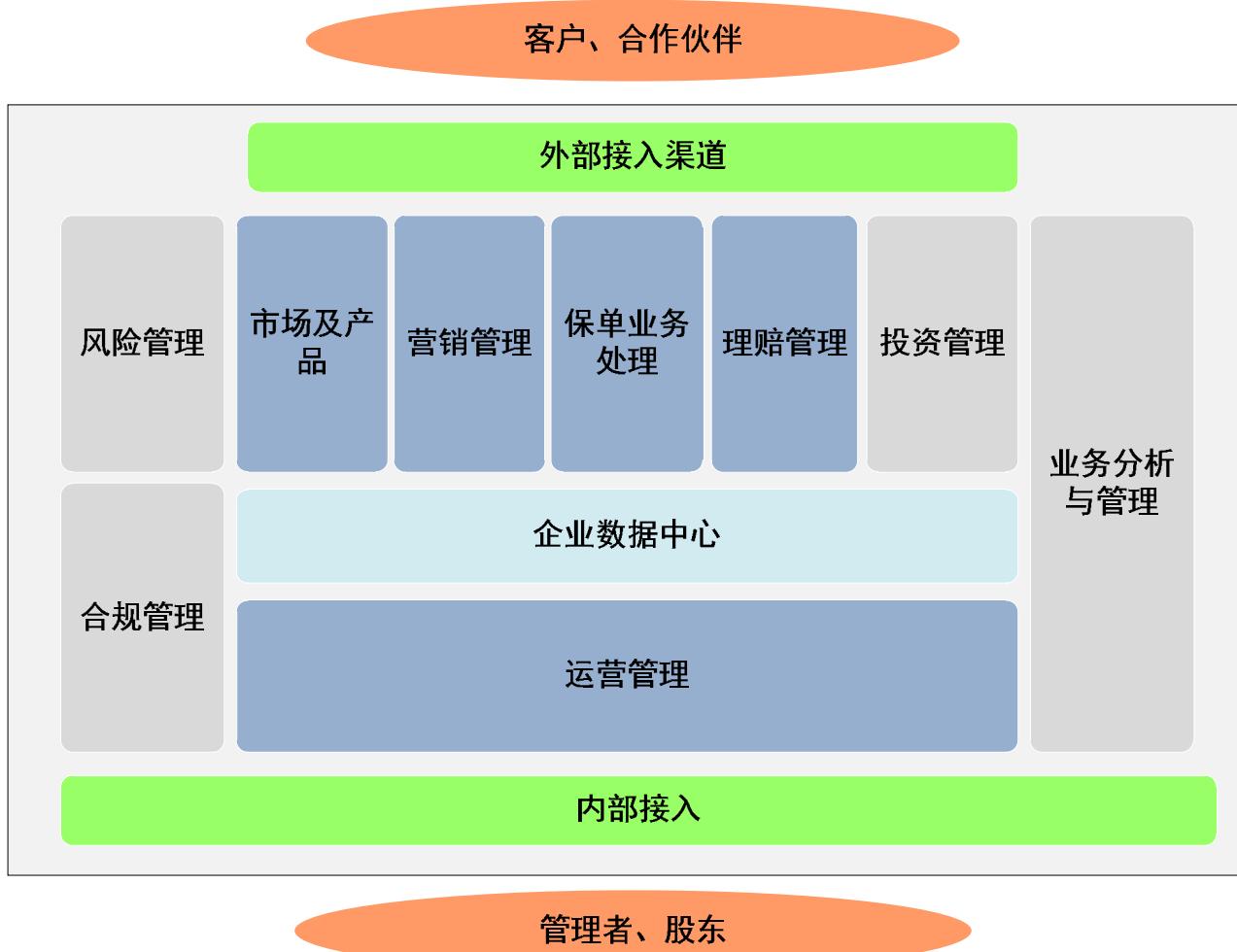
- 现状：
  - XXIT系统中缺失的部分业务支撑职能，属于核心的功能范畴
    - 客户实体：客户的定义，
    - 产品管理中心：
  - 核心维护工作量主要集中于新产品的增加
  - 与渠道有关的系统在不同的程度上制约于核心应用的功能缺失，损失一些灵活性来弥补核心应用的功能短缺
- 讨论：
  - 在核心改造的同时，拆分目前核心系统的不完备功能，重新审视目前核心的功能定位（保单中心）
  - 建立目前核心系统缺失的支撑职能的应用模块（产品中心，客户中心，用户中心）
    - 业务逻辑与控制逻辑的拆离
    - 保险产品的抽象
    - 用户管理与权限管理模式的抽象
    - 模型的抽象



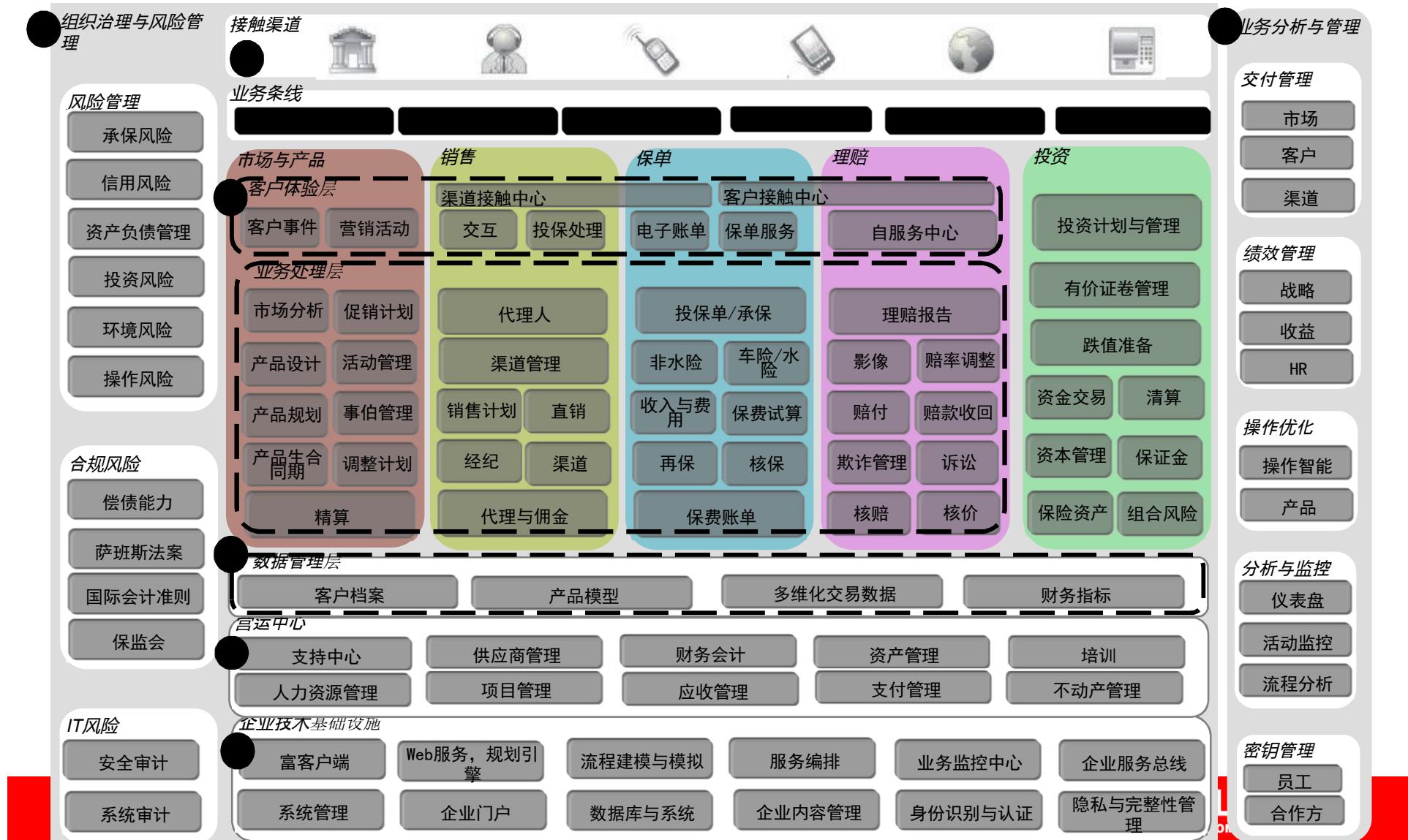
# 企业IT架构的多视图表达



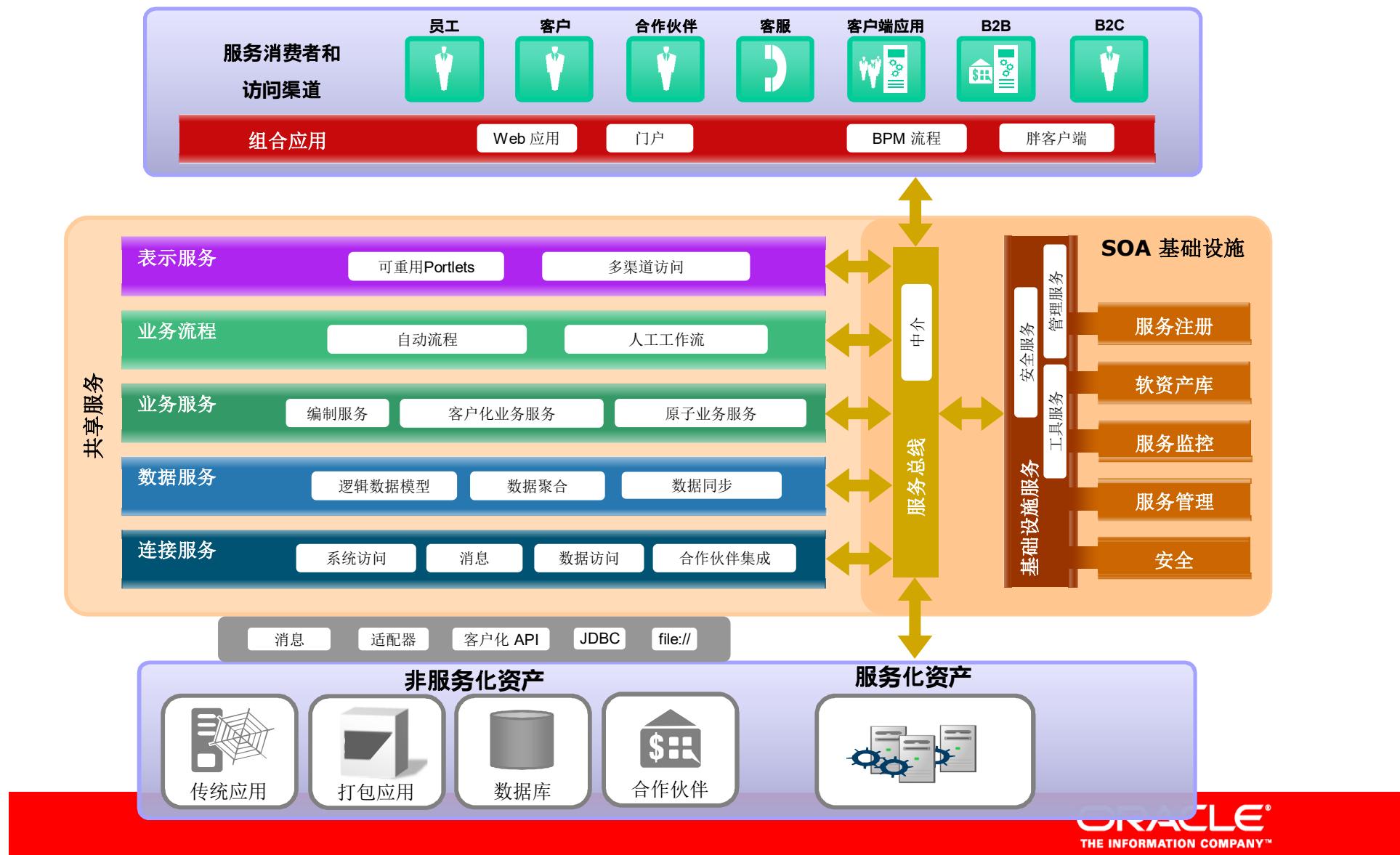
# 业务视图：0级视图



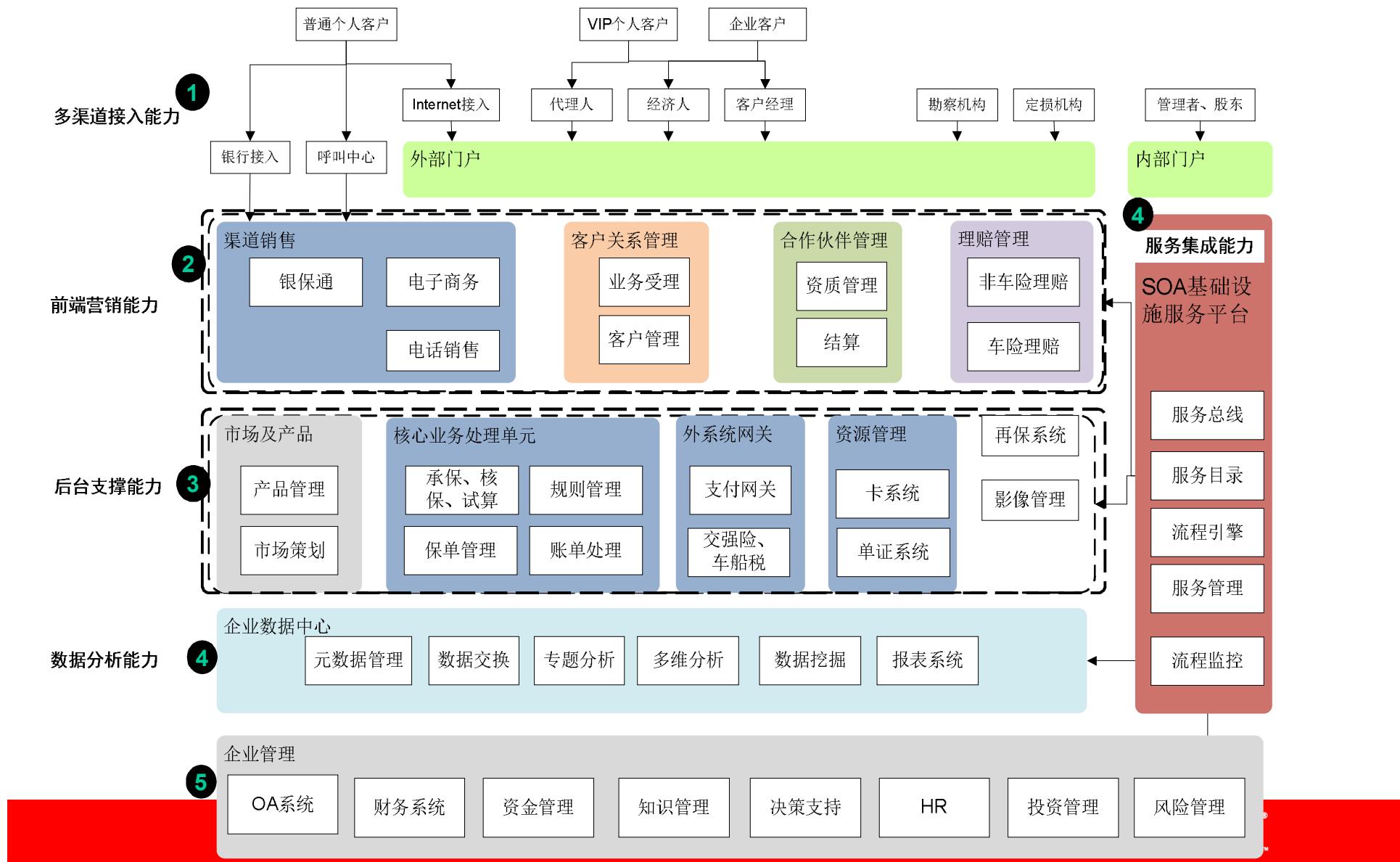
# 业务视图：1级视图



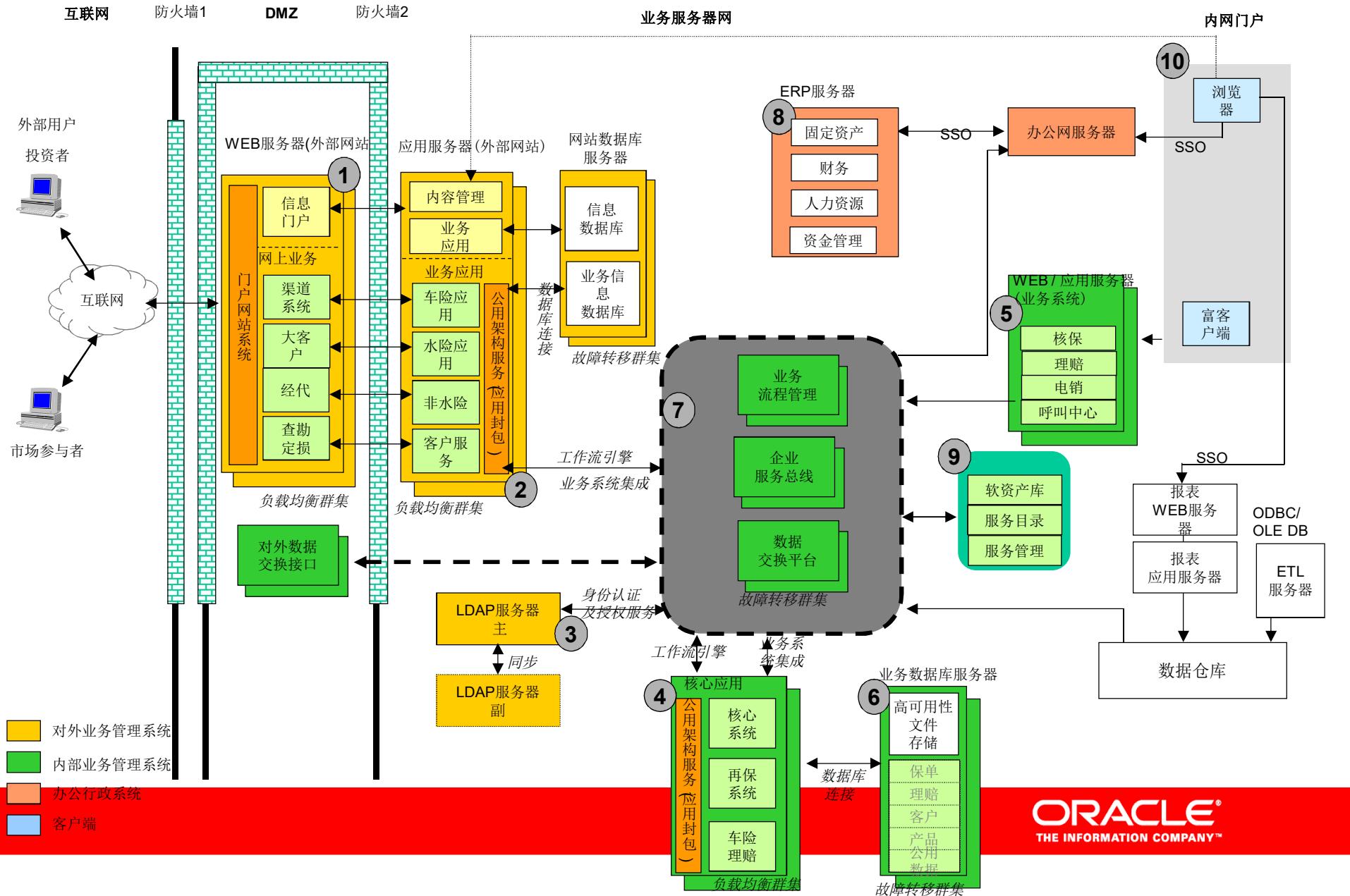
# 层次化的架构参考模型



# 应用架构



# 技术架构部署视图



# 架构演进战略

根据XX的现状和远景架构设想，我们建议三个主要阶段实施。



- IT策略的三个要点
- 1 以客户为中心
  - 2 市场定位对于IT系统的要求
  - 3 提高企业IT开发、运维和管理效率
  - 4 和两个成功关键因素
  - 5 IT建设对企业的推动作用
  - 6 各级人员对IT建设的重视

- |   | 短期                                                                              | 中期                                                                                  | 长期                                                                                  |
|---|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 1 | <ul style="list-style-type: none"> <li>• 建设标准技术基础架构，为各实施阶段打下基础</li> </ul>       | <ul style="list-style-type: none"> <li>• 扩展第一阶段建设的多门户能力，为客户提供联合门户</li> </ul>        | <ul style="list-style-type: none"> <li>• 构建产品中心，支持新增或改进的产品</li> </ul>               |
| 2 | <ul style="list-style-type: none"> <li>• 实现表现层门户化，建立多门户统一用户管理，提高客户体验</li> </ul> | <ul style="list-style-type: none"> <li>• 通过BPM和规则控件进一步增强服务组合能力。</li> </ul>          | <ul style="list-style-type: none"> <li>• 建立企业数据交换和管理平台，包括ODS, EII和CIF等应用</li> </ul> |
| 3 | <ul style="list-style-type: none"> <li>• 建立服务基础设施(ESB)，将现有应用功能逐步服务化</li> </ul>  | <ul style="list-style-type: none"> <li>• 选取典型流程，建立端到端流程的支持能力</li> </ul>             | <ul style="list-style-type: none"> <li>• 进一步完善服务基础设施，建立企业软资产管理库</li> </ul>          |
| 4 | <ul style="list-style-type: none"> <li>• 规范化已有系统间集成接口</li> </ul>                | <ul style="list-style-type: none"> <li>• 集中管理“客户、产品”等主数据，建立主数据交换和应用的闭环流程</li> </ul> |                                                                                     |
| 5 | <ul style="list-style-type: none"> <li>• 提供快速渠道扩展业务能力（电子商务、电销）</li> </ul>       |                                                                                     |                                                                                     |