

系统构架设计文档

北大 Mozart

系统架构分析与设计

【文档介绍】 本文档内容包括解决方案中的系统架构的分析过程、架构设计和最终的产品选择策略，并详细记录决策过程中，我们做出每一步选择时的依据。并记录了用ATAM方法进行架构评估的结果。

1. 企业高层业务视图

在对系统 Use Cases 深入分析的基础上，我们可以得出如下高层视图：

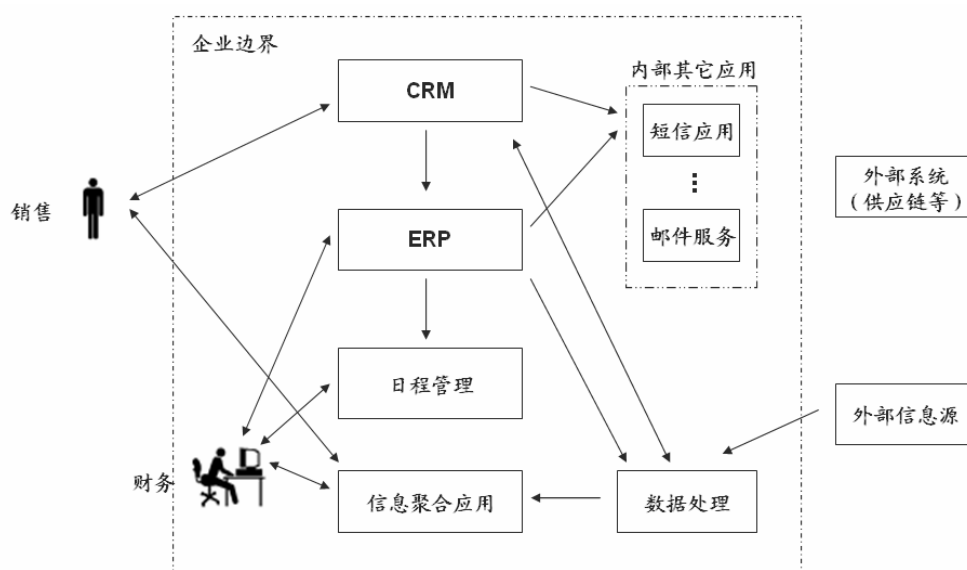


图 1 企业高层业务视图

如图 1所示，为位于企业边界之外的销售人员与企业内部网的财务人员提供所需服务的企业应用包括CRM系统、ERP系统、日程管理、信息聚合应用等。数据处理模块提供两方面的功能：一方面，其完成企业内部的数据同步，确保CRM中的产品、客户和订单等信息与ERP保持一致；另一方面，其聚合了CRM、ERP和所配置的企业外部信息源中的数据，提供给作为前台表示层的信息聚合应用。企业内还存有邮件系统、短信平台等内部应用，以提供所需的其它业务功能。此外，企业内部应用还可能需要与其他外部系统，如供应链的下游系统之间进行互联。

基于该视图，我们将进一步分析其中实体间交互的特点，并结合 IBM 相关的 e-business 模式、题目要求和我们的理解，给出最终的设计方案。

2. 企业业务模式决策

首先我们识别企业中存在的业务模式（Business Patterns）。在凤凰公司中存在三类业务模式：

- 财务人员与销售人员通过内部网络或 Internet 访问企业中系统提供的业务功能，如

使用 CRM、ERP、日程管理应用等等，这些属于 Self-Service 模式；

- 在我们所提出的扩展需求中，企业人员通过使用信息聚合应用提供的工具从企业内部的海量数据中获取所需的信息，这属于 Information Aggregation 模式；
- 数据处理模块与外部信息源间的交互，以及企业内部其它应用与外部系统的交互，均属于 Extended Enterprise 模式¹。

接着，我们识别企业中的集成模式（Integration Patterns），其同样包括三类：

- 根据竞赛题目给出的说明，凤凰企业内部的多个应用需要自动集成以提供新的业务功能。这属于 Application Integration 中的 Process Integration 模式；
- 数据处理模块需要处理企业内部应用间信息的聚合和同步等数据共享问题，这属于 Application Integration 中的 Data Integration 模式；
- 此外，在我们的扩展需求中，推荐为企业人员使用企业多个内部应用提供统一的访问入口，这是一种 Access Integration 模式；

综上所述，我们将相应的模式选择结果在图 2 中给出。

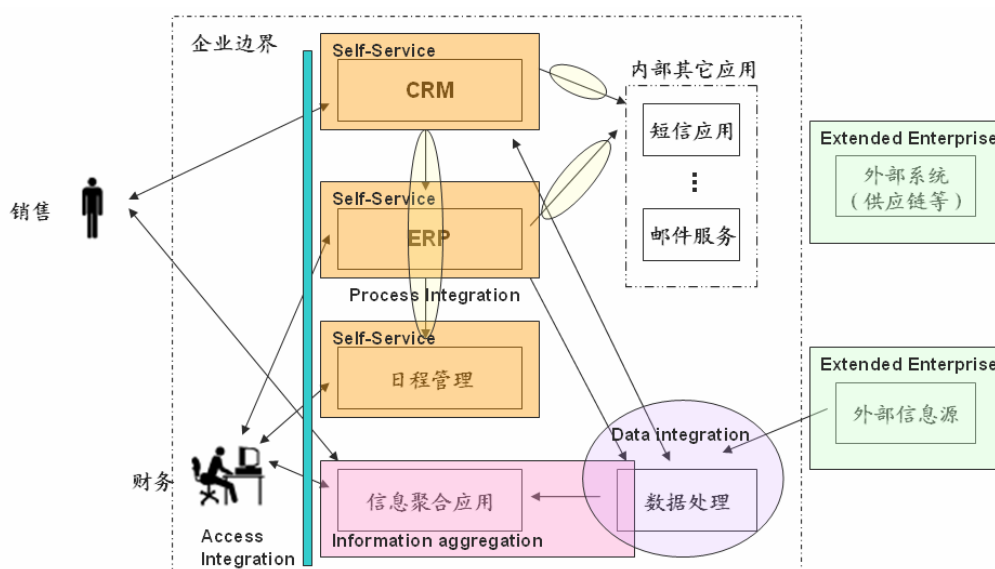


图 2 凤凰公司业务模式一览

3. 企业应用模式决策

接下来本文档进一步确定解决方案中所应采取的应用模式（Application Patterns）。我们在业务模式的基础上进行分析：

在我们的方案中，用户与企业后台应用间的交互都是通过 Web 通道来完成的，因此对于图 2 中的三处 Self-Service 模式，最合适的应用模式均是 Directly Integrated Single Channel 模式；

对于 Information Aggregation 模式而言，适合此处的是应用模式是 User Information Access 模式，基于扩展需求的需要，我们选择该模式的联邦变种（Federation Variation），因为解决

¹ 由于竞赛题目中没有涉及企业与外部应用，如供应链下游系统间的互联关系，因此本文档也没有标出具体连线；但是，基于我们对于扩展需求中外部信息源的分析讨论，这些外部系统均可采用类似的方式与企业内部应用集成。

方案中同时需要数据复制和同步两种功能，这种情况下联邦模式是最合适的选择²。

在企业与外部系统交互中，由于每个外部信息源与企业的聚合应用之间都是 1-1 互联的拓扑关系，结合对方案成本的考虑，我们这里选择 Exposed Direct Connection 模式。需要指出的是，未来可能会出现 1-N 的关联关系（如与多个供应链下游厂商的互联），此时可对应地采用 Exposed Router 应用模式。

接下来我们分析所采用的整合模式：

在 Access Integration 中，我们需要满足以下功能：一、用户可以通过统一的门户访问多个内部应用而无需重复登录；二、不同角色的用户将被引导至不同的应用组，如销售人员使用的是 CRM 与信息聚合应用，财务人员使用的是 ERP、日程管理与信息聚合应用等；三、已部署应用仍保持其原有的界面，以符合用户习惯，避免新的学习曲线；四、在这一过程中，企业安全方面的因素需要被纳入考虑范围。综合分析后，我们决定采取 Single Sign On 与 Page Aggregate 两个应用模式，前者提供对单点登录和安全认证的支持，后者可以聚合多个 Web 应用，并且，两者的运行模式（runtime pattern）中的大部分模块均可复用。

对于方案中采用的 Process Integration 模式而言，由于此流程中应用间的交互不存在并发流程，因此 Serial Process 模式是合适的方案。

对于 Data Integration 而言，由于在信息聚合应用中，企业内部数据以统一的逻辑视图展现给用户，因此适合选择 Federation 应用模式，这与之前 Information Aggregation 模式的相应决策也是保持一致的。

由此我们得到的应用模式可以总结如下：

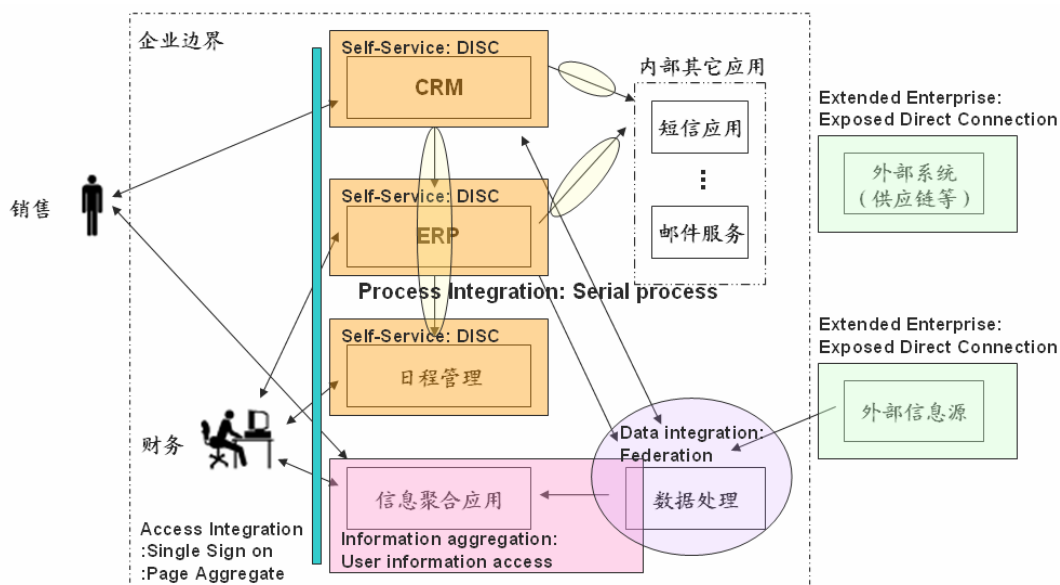


图3 方案应用模式一览

² 基于对中小企业可能出现的预算限制、相关的风险分析等考虑，本文最终的解决方案中包含多处变化点，包括模式的选择、产品的选择、实现的优先级建议等等。在预算不足时，此处可以安全地选择低成本的方案或“退化”为更简易的模式。详细内容参见“方案风险分析及产品映射决策”一节。

4. 企业运行模式决策

下文我们依次给出所选择的应用模式所推荐的运行时模式：

首先给出的是 Access Integration: Single Sign On 的运行时模式：

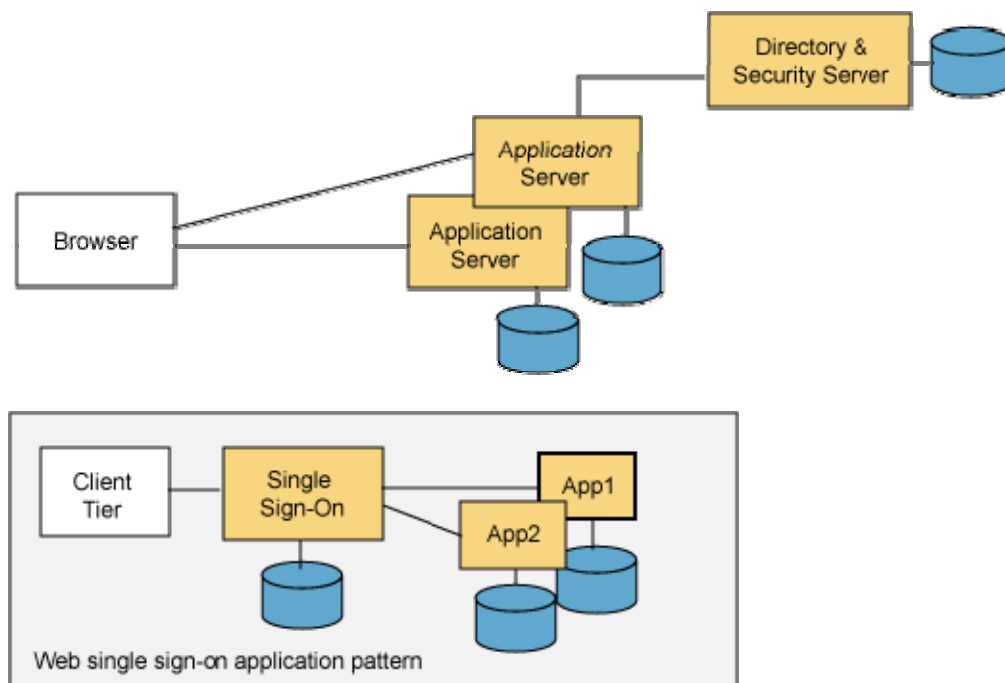


图 4 Single Sign On 的运行模式

图 5 给出 Access Integration: Page Aggregate 的运行时模式，其中的目录/安全服务器设施可与图 4 复用，Portal Server 则提供了多个 Web 应用的聚合视图。

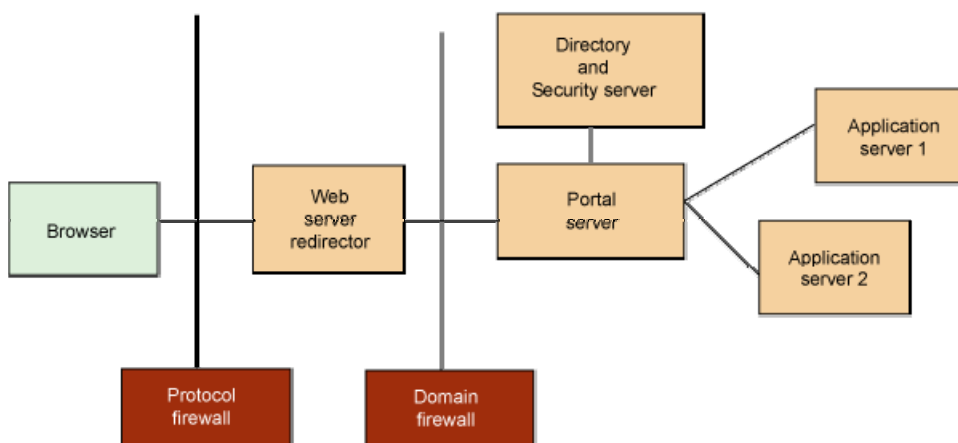


图 5 Page Aggregate 的运行模式

下图给出了 Self-Service: Directly Integrated Single Channel 的运行模式:

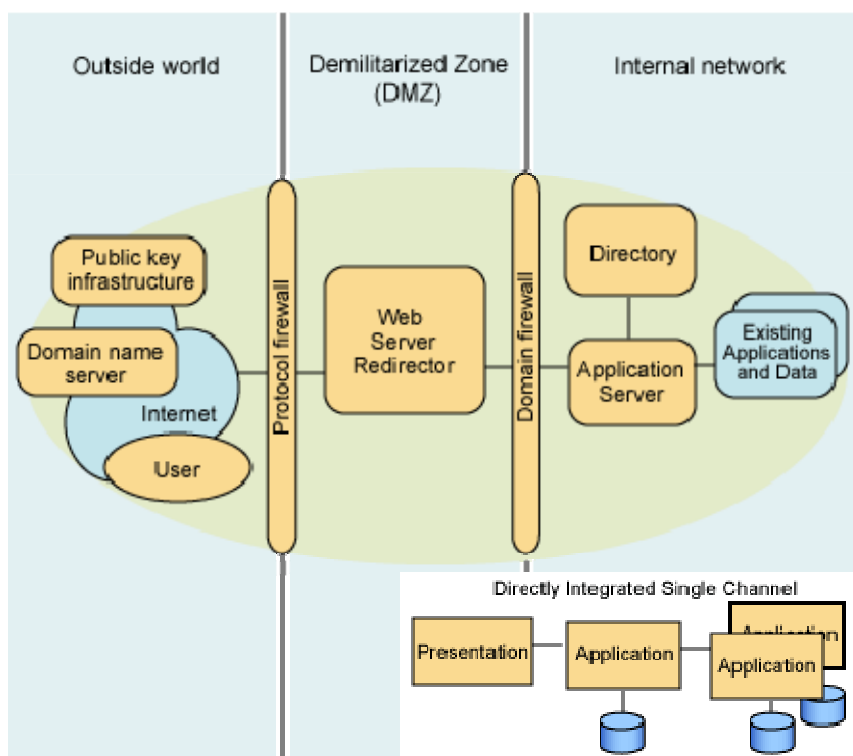


图 6 Directly Integrated Single Channel 的运行模式

图 7和图 8分别给出Information Aggregation: User Information Access = Federation Variation和Data Integration: Federation的运行模式, 其中大部分设施均是保持一致的, Federation模块在这里完成不同应用间数据的同步和聚合, 并为信息聚合应用等提供所需的数据内容。

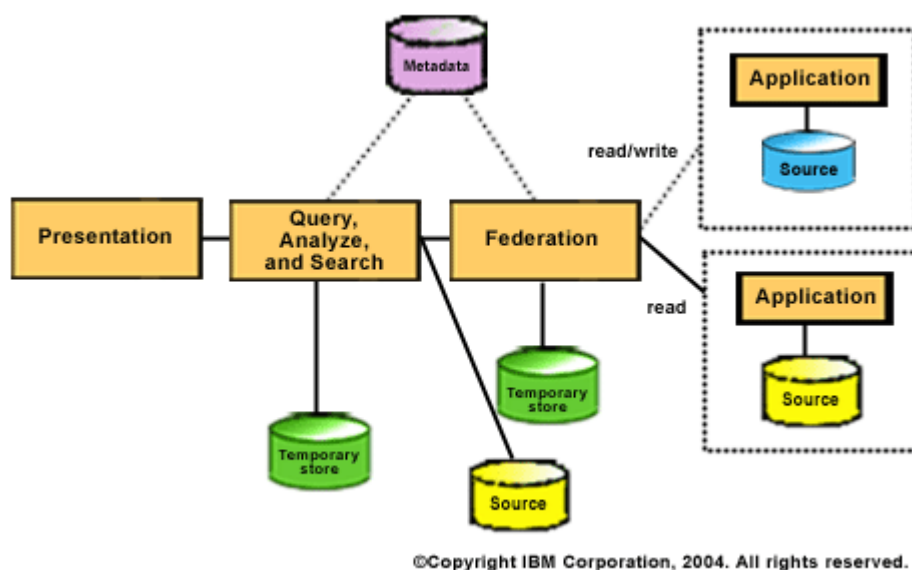
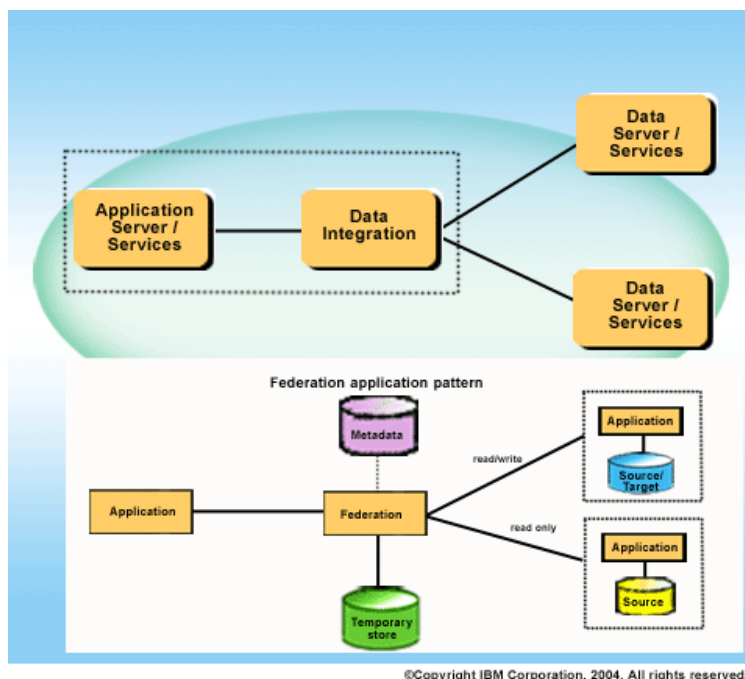


图 7 User Information Access = Federation Variation 运行时模式



©Copyright IBM Corporation, 2004. All rights reserved.

图 8 Data Integration: Federation 运行时模式

接下来讨论Extend Enterprise: Exposed Direct Connection的运行模式。我们选择的是IBM在去年发布的Exposed Direct Connection: SOA Profile，采用企业服务总线（Enterprise Service Bus, ESB）提供完全面向服务的运行设施。在此基础上，企业内部各种应用均可以在服务总线上以一致的方式进行互操作。整个结构如图 9所示。

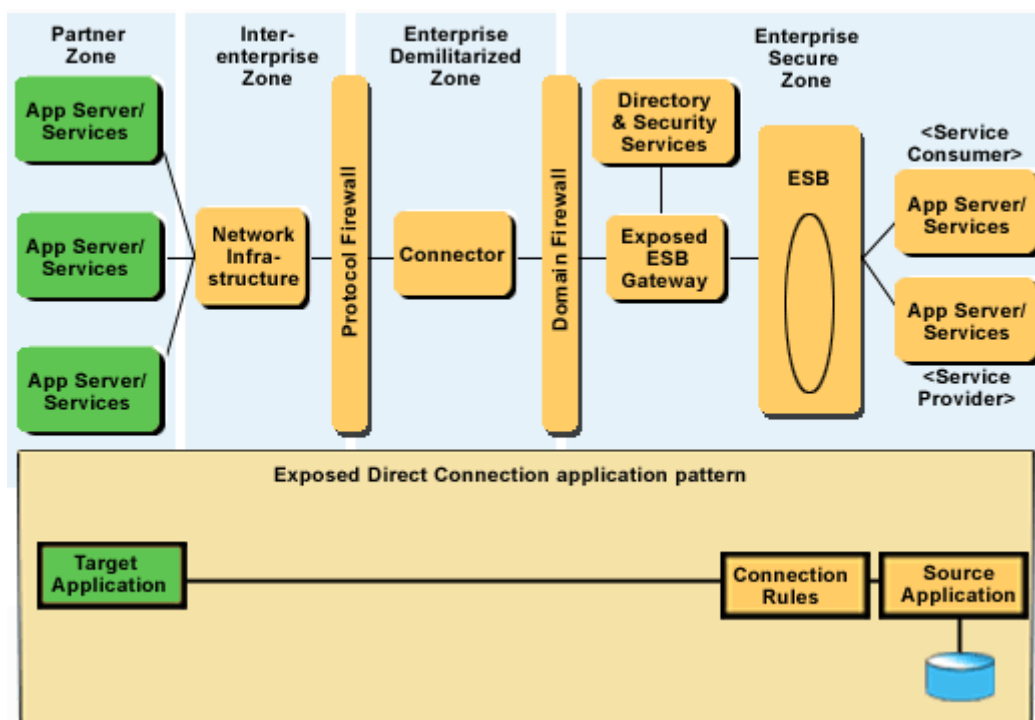


图 9 Exposed Direct Connection 的运行模式

Process Integration: Serial Process的运行模式参见图 10。一个服务流程引擎负责对多个

服务之间的交互进行协调（Choreography）。需要说明的是，这个协调服务同样作为一个服务发布在图 9给出的ESB总线上。

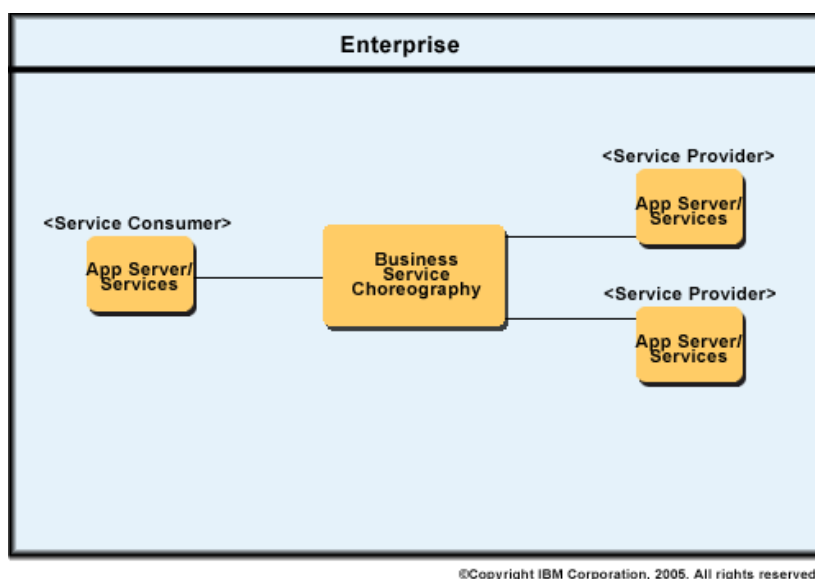


图 10 Serial Process 的运行模式

综合上面给出的七个运行时模式，我们在图 11中给出了一个组合视图，作为给凤凰公司的推荐方案（BSC即Business Service Choreography服务）：

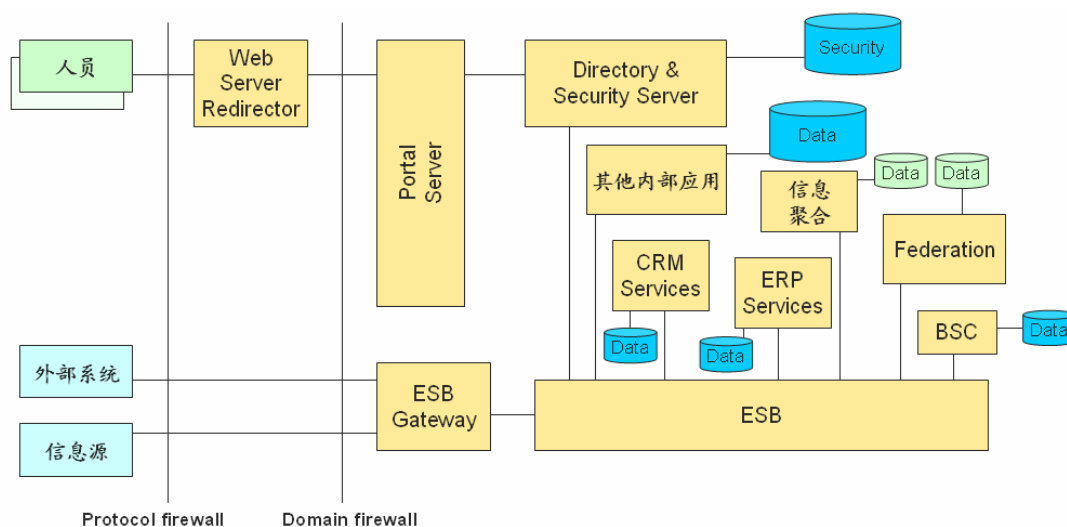


图 11 运行时方案的组合视图

5. 方案风险分析及产品映射决策

对于企业的IT基础设施而言，选择可靠性高、互操作性强、服务良好的产品是提供高质量服务的重要保证，因此我们推荐使用IBM的系列产品作为首选方案，在使用大量时间进行相关的技术和产品调研后，我们给出图 12所示的产品映射图：

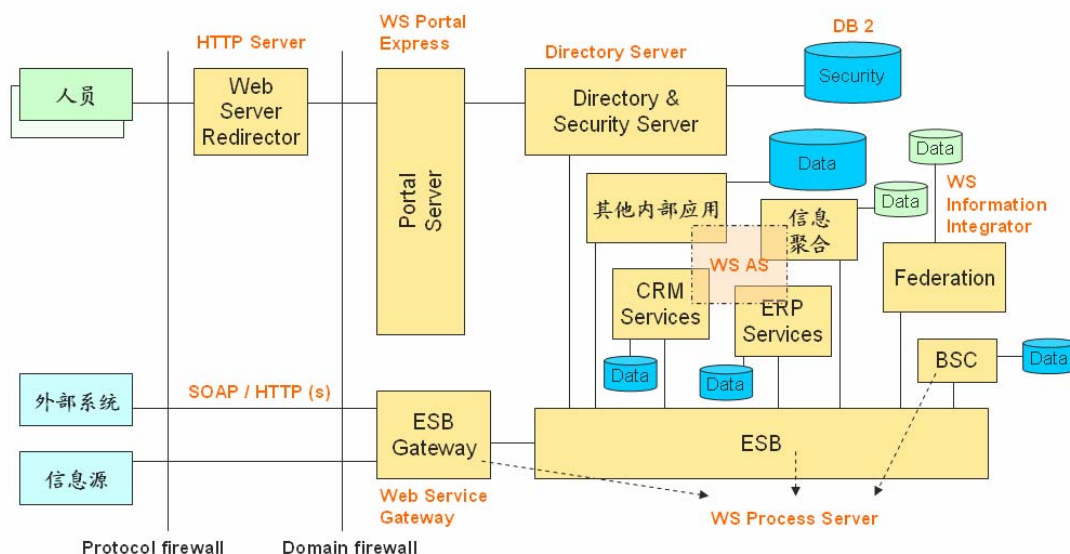


图 12 方案的产品映射图

其中，应用部署的应用服务器为 WebSphere Application Server，采用的数据库为 DB2，在两类数据整合产品 WebSphere Data Integration Suite 和 WebSphere Information Integrator 中我们选择了后者，因为它同时提供了聚合和同步两方面的功能，其 Omni Edition 还能方便地提供企业内部信息搜索，这一需求可能随着凤凰公司规模扩大而变得重要；Portal Server 选择的是 WebSphere Portal Express，服务总线方面选择的是 WebSphere Process Server，它为面向服务的架构提供了强大的支持，同时其包含的受限的 ESB Gateway 也足以满足本方案中对网关产品的需求。在 Http Server、Directory/Security Server 和防火墙方面，我们假定凤凰公司应该已具有相关的产品，如无特殊情况可继续复用原有设施。

在考虑最终方案时，有两方面的因素需要考虑，其一是企业的风险分析，其具体讨论超出了本文档的范围，请参考文档“项目实施计划”中的相关内容；另一方面是需要从企业的实际出发，就企业的现有条件，根据性价比最高的原则做出权衡。我们认为其中如下因素需要重点考虑：

- 对企业原有 IT 产品的复用：企业对 IT 的已有投资如果运转良好，应得到保护，未充分调研就换用新产品可能引入原本没有的问题。
- 企业的预算：题目中已经强调凤凰公司一直致力于企业的信息化建设，因此可以认为企业愿意给出较多的预算；但是，作为一家中小型企业，其不可能投入过量的资金。
- 企业人员的学习成本：引入新的技术会带来对应的学习曲线，可能需要企业付出一段时间适应并进行相关的人员培训，这也将带来费用的支出。
- 企业未来的发展：考虑到企业规模的扩大和新的各式各样的企业应用的引入，现有的基础设施应具备较好的扩展能力。

我们对可能采用产品的价格进行了调研，由于 IBM 在中国大陆的产品价格并未公开发布，我们使用了 IBM 在香港地区发布的产品价格作为参考：

产品名称	价格（单位:美元） ³
WebSphere Process Server (D558JLL)	85,000.00
WebSphere Portal – Express (D51RKLL)	33,300.00
WebSphere Application Server (D5CB9LL)	10,000.00
Information Integrator Standard Edition (D51LILL)	25,000.00
DB2 Universal Database Express Edition (D535PLL)	4,874.00

表 1 方案一：“IBM 豪华套装”产品列表

不难看出如果为每个应用（包括原有的 ERP 系统等）均配备独立的 WebSphere 应用服务器和 DB2 数据库，整个解决方案的软件产品成本接近两百万元，并且不包括开发费用。我们将这一方案列为“豪华套装”：它具备极佳的性能和扩展性，但只有在预算足够高时选用。

事实上，对于上述产品，考虑到凤凰公司作为一家中小规模的企业，完全可以使用 IBM 提供的功能略受限制的产品版本，如 WebSphere Application Server Express (D520ELL) 报价为 2000 美元，用于内部网、用户限制为 20 人的版本(D522LLL)仅需 500 美元；DB2、Portal 也有相应的产品。此外，对于原有的系统，仍然采用其最初的产品，不管其是否为 WebSphere 系列产品。

在这一折衷方案中我们保留了 WebSphere Information Integrator 与 WebSphere Process Server 两件产品。其中，前者主要是为了支持题目中数据的同步、以及为扩展需求提供数据仓库等对性能十分敏感的需求而保留的，事实上，如果将扩展需求简化，基于本方案在需求分析阶段对数据同步所需服务的分析及对其变化性的规约，完全可以通过自行设计相应的数据处理服务来实现。

与其不同的是，Process Server 在我们的方案中强烈建议保留，这是因为：一方面，在本方案中，Process Server 作为支撑各种应用的核心设施，其对于性能的影响极大，需要使用高质量、有良好服务支持的产品；另一方面，基于企业服务总线所提供的标准接入方式，可以较好地屏蔽其他应用的异构性，从而降低整合系统的开发费用，以及减少未来引入新的应用和系统的开销。

此外，当前开源社区中同样有不少开源产品能够提供我们需要的功能，并且与使用 IBM 产品受限版本相比，性能差别也不是很大；但与此同时，开源方案也有其自身的弊端，包括：大部分开源产品的文档不够健全，学习曲线较长；很少能够提供针对性的技术支持；服务较好的开源产品同样需要付出相应的服务费用，如 JBoss 等。当然，如果凤凰、万合具有较强的技术团队，采用开源产品是种不错的选择。因此，这里我们给出两种经济适用方案，分别为 IBM 产品简易版和开源产品版。

产品名称	价格
WebSphere Process Server (D558JLL)	85,000.00
WebSphere Portal – Express (D529GLL)	1,700.00
WebSphere Application Server Express (D520ELL)	2,000.00
(optional) Information Integrator Standard Edition (D51LILL)	25,000.00
DB2 Universal Database Express Edition (D52BILL)	624.00

表 2 方案二：“经济套装-IBM 简装版”产品一览

³此处及后续方案价格单位缺省均为美元，且均为税前报价；IBM 产品均包含有 12 个月的维护服务。

产品名称	价格
WebSphere Process Server (D558JLL)	85,000.00
Portal / JBoss4	Open source
JBoss	Open source
(optional) Information Integrator Standard Edition (D51LILL)	25,000.00
MySQL	Open source

表 3 方案三：“经济套装-开源混装版”产品一览

方案二的成本在 100 ~ 120 万人民币之间，方案三则基本能控制在百万元以下，已较符合中小企业的需要。

如果需要进一步降低预算，完全可以按照本组方案之前对需求的分析，结合开源产品自行开发数据同步和聚合服务，其代价是开发人员的费用、较长的开发周期和一定程度上的性能损失。此外，对于企业服务总线，当前也已出现了一些逐渐成熟的开源产品，如 IONA 与 ObjectWeb 联合推出的 Celtix 1.0 等，从而得出了完全使用开源产品的方案四。

产品名称	价格
Celtix 1.0	Open source
Portal / JBoss	Open source
JBoss	Open source
开源数据仓库 DeepGreen + 自主开发	Open source
MySQL	Open source

表 4 方案四：“平民套装-开源万岁版”产品一览

需要强调的是，方案四并不是免费的，由于性能带来的业务损失、缺少售后服务带来的维护费用，都会是可观的。

6. 实现相关技术

根据这一阶段的需求，实现相关的内容并非本文档的关注点，因此这里只是对可能涉及的产品、技术等给出简单的说明，以用于估计工作量和制定技术学习的计划。

在整合多个异构数据源的数据方面，WebSphere Information Integrator 提供了 SQL 和 Web Service 两种标准接口供 Portal 调用。本次大赛的光盘中提供了该产品的试用版，可考虑基于该产品完成应用后端的信息处理模块；否则，需要自行开发实现所需的服务，其详细设计请参见服务模型分析设计文档。

必须通过编程完成的模块是信息聚合应用的前端表示层，即其可视化部分。其主要技术点包括 Ajax 技术、数据向曲线图的转换，以及整个框架和管理界面（定制数据项等）的设计；此外，对于获取 RSS 信息源的信息，有相应的开源工具包，也需要阅读其 API 文档了解使用方法。

⁴ JBoss的服务与文档均需要收费。

7. 软件架构评估

在软件开发过程中,问题存在的时间越长,影响越大,解决问题的代价越大。在设计阶段改正需求阶段中出现的错误要比在测试阶段更正的代价小的多。而系统架构,是最初设计决策的体现,对系统的开发有着深远的影响。不适当的软件架构将给软件系统的开发带来灾难性的后果。

在完成了架构设计之后,对其进行评估,能够为构架设计的正确性提供保证,同时也对质量属性的需求是否得以满足,存在哪些潜在的风险等方面进行分析。在这里选用了构架权权衡分析方法 ATAM (The Architecture Trade-Off Analysis Method) 来对其进行评估。

7. 1. ATAM 介绍

ATAM(The Architecture Trade-Off Analysis Method) [2]方法是 CMU 在 1998 年提出来的,这个方法给出了一个设计和分析的螺旋模型。它提供了一个在多个质量属性(易修改性、安全、性能等)之间互相影响的情况下分析理解软件架构的原则化的方法,并且认为识别出多个质量属性之间的折中权衡在软件架构的设计中是非常重要的。

在 ATAM 出现之前已经出现了很多软件架构的分析方法,但是这些方法一般都是考虑对某一个属性单独分析。而实际上,一个系统的质量属性之间是相互影响的。ATAM 方法分析架构层上的设计方案,考虑多个质量属性互相影响的情况下架构的设计怎样才能符合要求。ATAM 方法识别出属性之间的“折中点”(即属性之间的依赖关系),从每一个属性的视角增进系统相关人员之间的交流,明确、精化需求,并且提供了一个并行的系统设计、分析的框架。ATAM 所具有的这些特点,也是我们选择其作为构架评估方法的原因。

7. 2. 系统构架评估

ATAM 包括了表述部分,调查和分析部分,测试部分和形成报告的部分,主要分为九个步骤:(1) ATAM 方法的表述,(2) 商业动机的表述,(3) 架构的表述,(4) 确定架构的方法,(5) 生成质量场景效用树,(6) 分析架构方法,(7) 集体讨论并确定场景的优先级,(8) 分析架构和(9) 结果的表述。其中第一个阶段由评估小组和项目决策者参与的包括前 1-6 步;而第二阶段,先由第一阶段的参与者分别进行第一阶段的总结,然后由评估小组,项目决策者和所有风险承担者共同进行第 7-8 个步。目前由万合整合公司所作的是第一阶段的评估。

7. 2. 1. ATAM 方法的表述

介绍了 ATAM 评估方法,并留出一些时间,讨论关于该方法及其结果和目标等问题。

7. 2. 2. 商业动机的表述

对凤凰公司的商业动机进行了说明。首先基于业务建模部分(超链接)对企业业务目标和组件化业务模型的分析,介绍系统所需要完成的任务。之后对这些任务的具体需求进行了介绍。主要总结项目的商业目标,以及商业,技术和经济上的约束。同时也提出一些质量属性方面的要求,例如要尽量减少成本和维护所需的 IT 技术,以及对各个系统信息需要同步等。

7. 2. 3. 架构的表述

随后,对系统的软件架构进行了介绍。关于架构的详细信息,可参看文档架构设计(超链接)

7. 2. 4. 确定架构的方法

在这里，对软件架构的设计进行了进一步的介绍，基于前面对软件架构的整体介绍，针对一些系统约束和质量目标等，对一些重要的设计决策进行了解释。其中包括：

- Pattern 的选择
- 产品的选择

7. 2. 5. 生成质量属性效用树

评估小组和项目决策者（包括软件架构设计小组、经理和客户代表）针对该系统的最重要质量属性目标进行讨论。风险承担者一开始强调可维护性、可操作性和可靠性，后来通过探讨以后系统可能面对的新需求，风险承担者发现可扩展性和性能随着公司业务的发展可能会更重要。经过这个讨论，得到了如下的系统质量属性效应树。

质量属性目标	标识号	针对质量属性的需求
可维护性	M1 M2	对一个子系统的更改不要求改动其他子系统 较快适应企业主数据源的更改 较快适应订单请求审批的流程的改变
可操作性	O1 O2	子系统共享的信息必须是同步的 订单请求传送的实时性
可靠性	R1	订单请求流量达到高峰时不会导致系统崩溃或者请求丢失
可扩展性	S1 S2	可为信息聚合增加新的数据源 可增加新的业务模式，如客户可以直接从 portal 中提交订单请求
性能	P2	数据更改频率增加不会较大影响系统效率

之后，会议的参与者们通过讨论，又对质量属性需求进行进一步求精，得出了可分析的质量属性场景。然后通过用数值来衡量每个场景的重要性和实现难度，从而得到每个场景的讨论优先级。讨论结果经过整理如下表：

第 2 层: 质量属性	第 3 层: 质量属性求精	第 4 层:质量属性场景	重 要 性	难 度	累 加 和
可维护性	M1: 对一个子系统的更改不要求改动其他子系统 M2: 较快适应企业主数据源的更改 M3: 较快适应订单请求审批的流程的改变	M1.1: 加入生产管理系统不会要求改动现有的 ERP 和 CRM 系统 M2.1: 加入物流管理系统之后库存数据以物流管理系统中的数据为主数据源 M3.1: 订单请求在经过库存检测之后，必须经过基本的客户信息检测才能提交给财务人员	30 20 20	20 20 10	50 40 30

可操作性	O1: 子系统共享的信息必须是同步的	O1.1: CRM 中的数据必须和 ERP 中的数据保持同步	30	10	40
	O2: 订单请求传送的实时性	O1.2: 加入新的生产管理系统之后, 其产品数据和 ERP 数据源保持同步	30	10	40
		O2.1: 订单请求从 CRM 到 ERP 的实时传输	30	10	40
可靠性	R1 订单请求流量达到高峰时不会导致系统崩溃或者请求丢失	R1.1: 订单请求高峰期不会导致请求信息丢失	30	10	40
		R1.2: 产品库存信息和产品价格在规定时间内频繁变化不会导致系统错误或者崩溃	30	20	50
可扩展性	S1: 信息聚合中数据源的增加和减少	S1.1: 为信息聚合增加外部的数据源	20	10	30
		S1.2: 信息聚合中数据源的更换	20	20	40
	S2: 客户可以直接从 portal 中提交订单请求	S1.3: 减少信息聚合中已有的数据源	20	30	50
		S2.1: 客户在网上直接从 portal 提交订单请求	20	10	30
性能	P1: 数据更改频率增加不会较大影响系统效率	P1.1: 产品价格和库存信息的频繁更改不会较大影响 ERP 和 CRM 系统的效率	20	20	40

7. 2. 6. 分析架构方法

这个时候, 通过前面的两步, 已经得到了划分好优先级的具体质量需求 (第 5 步) 以及软件架构的设计和其中所用的方法 (第 3、4 步)。这一步则通过对两者之间的分析, 来确定软件架构中的有风险决策、无风险决策、敏感点和权衡点。

分析从优先级高的场景开始逐个进行讨论, 评估小组一般不对全部场景进行讨论, 而是选择一定比例来进行分析 (评估选择了 50% 的场景进行分析)。对所有进行分析的场景, 评估小组都按照和下面类似的分析模版对其进行记录, 对于其中出现的有风险决策, 则将其记录下来, 并且分析其涉及的敏感点和权衡点, 而如果没有敏感点和权衡点, 则认为其是无风险决策, 同样进行记录。

下面列出最高优先级的三个场景的分析:

场景号:	M1.1: 加入生产管理系统不会要求改动现有的 ERP 和 CRM 系统				
质量属性	可维护性				
环境	在常规运行阶段				
刺激	企业增加新的子系统				
响应					
架构决策	有风险的决策	敏感点	权衡点	无风险决策	
订单激发事件	R1	S1	T1		

场景号:	R1.2: 产品库存信息和价格信息的频繁更改不会导致系统错误或者崩溃				
质量属性	可靠性				

环境	正常操作			
刺激	更新库存信息和产品价格信息频繁			
响应	不影响系统的正常运行			
架构决策	有风险的决策	敏感点	权衡点	无风险决策
订单激发事件				N1

场景号:	S1.3: 减少信息聚合中已有的数据源			
质量属性	可扩展性			
环境	正常操作			
刺激	减少数据聚合的数据源			
响应	不涉及此数据源的其它数据源聚合视图能够正常显示			
架构决策	有风险的决策	敏感点	权衡点	无风险决策
订单激发事件	R2	S2	T2	

- R1: 没有将现有遗产系统彻底改造成 SOA 的系统
- R2: 提供的数据聚合视图依赖于现有企业数据
- S1: 加入新的子系统需要对相应系统
- S2: 企业数据的组织方式不能改变
- T1: 时间和资源的限制
- T2: 数据聚合是作为提供决策支持的基础设施
- N1: 使用 Federation 来为数据提供统一的视图

7. 2. 7. 结果的表述:

经过上面的过程, 通过对质量属性效应树和其场景的分析, 评估小组将评估所得的结果进行汇总, 做一个简要报告。系统构架中仍然有一些带有风险的决策: 没有将现有遗产系统 CRM 和 ERP 改造成 SOA 的系统, 如果在加入新的子系统的时候, 可能需要将现有的遗产系统包装出相应的服务来支持, 其中的折中是作为中小型企业, 有时间和资源的约束, 希望遗产系统的改造是分步进行的。而数据聚合的视图依赖于现有的企业数据, 所以数据的组织格式不能随意改变, 折中是企业中主要数据的组织方式还是比较固定, 而数据聚合作为决策支持的基础设施, 本身需要带有一定的数据推理能力。

参考文献:

- [1] L.Bass, P.Clements, and R.Kazman, Software Architecture in Practise Reading, mass:Addison-Wesley,1998
- [2] Paul.Clements, Rick Kazman, and Mark Klein, Evaluating Software Architectures. Reading, mass:Addison-Wesley,1998
- [3] Liliana Dobrica and Eila Niemela, A Survey on Software Architecture Analysis Methods