

# SOA专业人员指南

## 第2部分

### SOA参考架构



作者: *Surekha Durvasula*

### 参与本书的SOA专业人员

Surekha Durvasula, Kohls, 企业架构师

Martin Guttman, Intel公司Customer Solutions Group, 首席架构师

Ashok Kumar, Avis/Budget, SOA Architecture经理

Jeffery Lamb, Wells Fargo, 企业架构师

Tom Mitchell, Wells Fargo Private Client Services, 首席技术架构师

Burc Oral, 个人贡献者

Yogish Pai, BEA Systems, Inc., AquaLogic Composer首席架构师

Tom Sedlack, SunTrust Banks, Inc., 企业架构工程师

Dr Harsh Sharma, MetLife, 高级信息架构师

Sankar Ram Sundaresan, HP-IT, 首席电子商务架构师

### 审校人员

Steve Jones, Capgemini Group的Application Development Transformation部门, 首席技术官

Prasanna Deshmukh, WebEx Communications, 架构主管

Noam Fraenkel, Mercury Interactive, 首席IT技术官

Ashok Nair, 关于EAI和信息技术服务的系统分析师, 卡尔加里城

Jeff Pendelton, SOA Alliance, 执行主管

Annie Shum, BEA, SOA战略副总裁

Brenda Michelson, Elemental Links, Inc., 首席顾问及分析师

George Paolini, Georgepaolini.com, 顾问

作者十分感谢为本文档做出贡献的组织和个人, 他们撰写了部分文档, 进行了大量编辑工作, 还帮助审校了文档并提供反馈。此外, 作者还要特别感谢BEA Systems, Inc., 是这个公司提供了本文档中的开发和演示所需的基础架构和平台。

# 目 录

1 关于本文档 .....	4
1.1 摘要 .....	4
1.2 目标受众 .....	4
1.3 《SOA 专业人员指南》的优点 .....	4
1.4 SOA 专业人员指南：章节 .....	4
2 SOA——参考架构 .....	5
2.1 定义 .....	5
2.2 SOA 参考架构方法 .....	5
2.2.1 SOA 基础 .....	5
2.2.1.1 业务架构 .....	5
2.2.1.2 基础架构 .....	6
2.2.1.3 数据架构 .....	8
2.2.1.4 信息架构 .....	8
2.2.1.5 SOA 补充架构 .....	9
2.2.2 企业 SOA 成熟度模型 .....	11
2.2.2.1 Web 应用程序开发阶段 .....	11
2.2.2.2 开发复合应用程序 .....	12
2.2.2.3 自动化业务流程 .....	14
2.3 SOA 参考架构 .....	14
2.3.1 Web 应用层 .....	15
2.3.1.1 打包的应用程序 .....	15
2.3.1.2 自定义应用程序 .....	16
2.3.1.3 门户服务 .....	20
2.3.1.4 企业基础架构服务 .....	21
2.3.1.5 企业（基于角色的）门户 .....	22
2.3.2 服务层 .....	23
2.3.2.1 服务总线 .....	23
2.3.2.2 服务注册表 .....	24
2.3.2.3 SOA 储存库 .....	25
2.3.2.4 服务管理器 .....	26
2.3.2.5 共享数据服务 .....	26
2.3.2.6 业务流程管理 .....	31
2.3.3 SOA 框架 .....	32
2.3.4 应用程序层 .....	33
2.3.4.1 旧应用程序 .....	33
2.3.4.2 大型机应用程序 .....	33
2.3.4.3 企业应用程序集成 .....	33
2.3.5 企业安全 .....	34
2.3.6 业务服务管理 .....	34
2.3.6.1 监控的典型当前状态 .....	35
2.3.6.2 业务服务管理的未来状态 .....	36
2.3.7 面向服务的基础架构（SOI） .....	36
2.3.7.1 面向服务的基础架构（SOI）的业务价值 .....	38
2.3.8 将 SOA 参考架构映射到企业 SOA 成熟度模型 .....	39

# 1 关于本文档

## 1.1 摘要

SOA是一门相对较新的技术，所以很多致力于实现它的公司无法获得良好的实践专业指导。如果没有基于共同经历的通用语言和行业词汇，SOA技术很可能最终只会为IT基础架构带来更多自定义逻辑并提高其复杂度，而无法实现其提供企业内和企业间服务重用和流程互操作性的承诺目标。为了帮助开发共享语言和有关SOA的知识集，一群SOA专业人员创建了《SOA专业人员指南》这个文档系列。在这个文档系列中，这些SOA专家描述并评注了一些与SOA相关的最佳实践和关键知识，以帮助其他公司迎接SOA的挑战。他们对《SOA专业人员指南》的期望是，这个分为多个部分的知识集在出版之后，能够作为所有SOA相关人员的标准参考百科全书。

## 1.2 目标受众

本文档的目标受众为：

- 需要跨企业/LOB启动和管理SOA战略的业务和IT领导
- 需要促进实现SOA计划的理想和路线图以及其中包含的每个实现的架构的企业架构师
- 需要管理SOA整体业务战略中的子项目组合的程序管理员
- 需要映射依赖关系并开发满足业务期望的时间线的项目团队成员
- 为业务和IT提供新业务能力的解决方案和工具的供应商
- 需要更好地了解业务和IT计划如何利用技术实现其目标的使用案例的标准机构。

## 1.3 《SOA 专业人员指南》的优点

本文档可帮助读者：

- 向他人学习：很早就开始使用SOA的用户可共享其有关跨行业采用SOA的最佳实践、看法以及观点
- 比较替代产品：标识和定义SOA的关键技术组件，建立对选项比较的基线参考
- 改进合作：使用一种通用语言澄清本文档中定义的SOA组件的性质
- 加速实现：本指南定义了其服务生命周期和需求、建议的工具以及每个阶段的最佳实践。
- 了解标准的价值：本文档建议了用于SOA的各个方面的标准
- 避免潜在风险：本指南指出了供应商社区尚未指出的一些问题领域。

## 1.4 SOA 专业人员指南：章节

《SOA专业人员指南》包含三个单独的部分。

第1部分（《为何使用面向服务的架构？》）提供了一个高层次的SOA摘要。

本文档（《SOA参考架构》）是第二部分。它提供企业级SOA实现的工作设计，包括详细的架构图、组件描述、详细要求、设计模式、有关标准的观点、法规遵从性模式、标准模板以及成员的潜在代码资产。

第3部分《服务生命周期简介》提供了在整个服务生命周期（从项目初期到完成或重新定位服务）中提供服务管理的详细过程。该文档还包含一个附录，其中包含组织和治理最佳实践、模板、对关键SOA标准的评论以及建议的到详细信息的链接。

## 2 SOA——参考架构

### 2.1 定义

SOA参考架构为企业或LOB定义了一个理想的目标架构。有些人还将其称为企业的“未来状态”或“未来理想”。SOA参考架构是构造从当前状态到目标状态的路线图的关键。

### 2.2 SOA 参考架构方法

要想开发参考架构，必须了解SOA的两个方面：

- 三个SOA基础组件
- 企业SOA成熟度模型

#### 2.2.1 SOA 基础

SOA基础组件如下图所示。



图1：SOA基础

##### 2.2.1.1 业务架构

业务架构描述SOA要支持的业务战略、目标、优先级和流程。只有在业务架构上提供的SOA才会成功。与基础架构或数据组件的潜在重用相比，业务流程的重用能够提供较高的ROI。

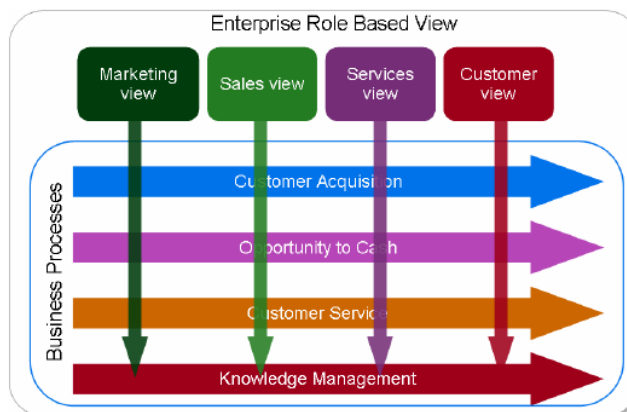


图2：业务架构的关注领域

开发业务架构的最佳实践包括：

- 审查当前系统规范和基础技术
- 将这些映射到业务战略以便找出差异
- 审查水平（业务流程）和垂直（基于角色的视角）要求
- 确定应用程序（服务）组合的优先顺序以便提供这些功能
- 跨应用程序标准化用户体验
- 定义有关关键方面的业务策略，例如应用程序和数据访问和法规遵从性

附加参考：开发业务架构视图（TOGAF）

[http://www.opengroup.org/architecture/togaf8-doc/arch/p4/views/vus\\_bus.htm](http://www.opengroup.org/architecture/togaf8-doc/arch/p4/views/vus_bus.htm)

### 2.2.1.2 基础架构

这是启用SOA的引擎。它应该能够解决可伸缩基础架构的所有方面，从网络、企业服务器、数据中心和防火墙到应用程序基础架构、安全性、监控和中间件。

基础架构团队要负责标识提供业务能力所需的基础架构组件（架构构建块）。

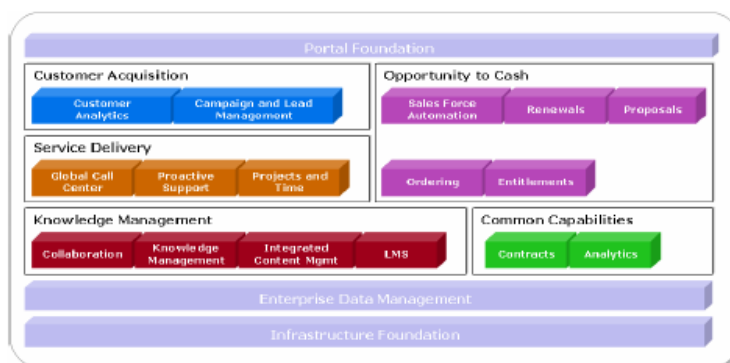


图3：架构构建块示例

上图显示了要提供主要关注业务流程的业务能力所需的架构构建块的一个示例。同时，基础架构还

需要包含基于角色的门户要求。

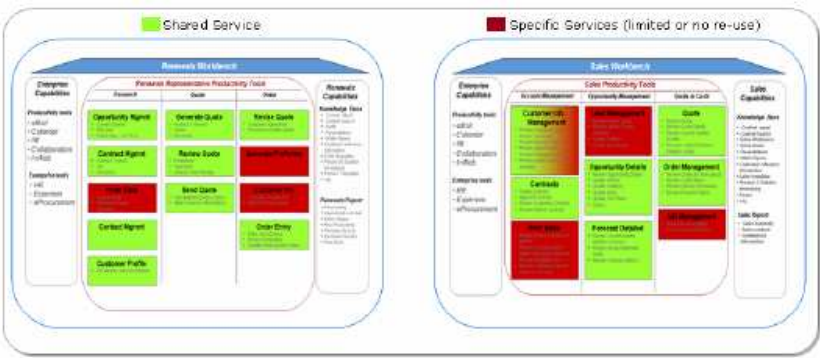


图4： 基于角色的门户的示例

基础架构需要将架构构建块和基于角色的门户组合起来以便实现以下目标：

- 高度重用公共服务
- 重用基础架构和基础组件
- 减少开发新功能所需的时间。

基础架构组件	描述
自定义应用程序框架 数据服务 记录服务 异常处理 审计服务 搜索框架 通知框架	开发自定义应用程序所需的公共组件
安全 身份验证 授权 单点登录（SSO） 委派管理	可扩展到企业级的安全框架
共享数据服务 主要数据管理 数据分析 数据质量服务 数据匹配	支持SOA的数据服务

数据验证 数据建模 分析服务	
门户服务 常见外观 个性化 报告 本地化 Web流量监控	用于一致用户交互的门户服务与利用WRSP的能力
企业基础架构服务 LDAP 电子邮件 协作（Chat/IM/Whiteboard） 内容管理 集成的结构化和非结构化搜索	需要的企业级公共服务
主要数据管理 客户数据集成 主要产品	跨应用程序和业务孤岛执行业务流程所需的功能

有关基础架构组件的附加详细信息可在“SOA参考架构”一节中找到。

附加参考：基础架构（TOGAF）

[http://www.opengroup.org/architecture/togaf8-doc/arch/p4/infra/infra\\_arch.htm](http://www.opengroup.org/architecture/togaf8-doc/arch/p4/infra/infra_arch.htm)

### 2.2.1.3 数据架构

数据架构处理数据的逻辑和物理建模以及数据操作和数据质量。SOA参考架构在任何可能的情况下通过提供方法、要求和设计模式详细地介绍所有这些领域。

### 2.2.1.4 信息架构

信息架构对给定业务流程的关键概念和事件进行建模。该业务概念表示需要由支持企业的流程或应用程序交换的任何业务实体。

信息建模可创建用XML模式表示的正则模型。正则模型是一种很脆弱的业务概念属性定义，包括属性关系、价值枚举、价值模式、XML文档相关属性的先后顺序以及该属性的必需性。SOA使用正则模型表示服务交易的请求和响应文档以及返回到消费者的内容负载。

业务应用程序交换的正则表达式通常是业务概念。例如，搜索产品目录可能会返回“候选产品列表”。业务流程销售或者发布的正则模型通常是业务事件。例如，“销售订单批准”业务事件可能由“供应链管理”业务流程发布，并且需要提供商订阅。

信息架构的另一个方面是捕获业务级别信息的关键性能指标（KPI）的定义。KPI可帮助组织来定义



和度量到组织目标的进展。KPI是指报告了从业务流程提取的值的的信息提取。

附加参考：关于信息架构的Wikipedia定义

[http://en.wikipedia.org/wiki/Information\\_architecture](http://en.wikipedia.org/wiki/Information_architecture)

### 2.2.1.5 SOA 补充架构

SOA本身是一种架构样式，必须认识到其他架构是与SOA策略成功与否息息相关的。下面几节概述了SOA策略中可能存在的一些架构。

#### 2.2.1.5.1 模型驱动架构 (MDA)

对象管理组 (OMG) 的模型驱动架构 (MDA, <http://www.omg.org/mda/>) 是一种使用平台无关 (platform-agnostic) 模式的方法，了解和管理运营中的企业的所有方面：包括“什么，怎么样，在哪里，什么时候，哪些人和为什么这样”

([http://www.omg.org/mda/mda\\_files/09-03-WP\\_Mapping\\_MDA\\_to\\_Zachman\\_Framework1.pdf](http://www.omg.org/mda/mda_files/09-03-WP_Mapping_MDA_to_Zachman_Framework1.pdf))。MDA的主要目标包括可移植性、互操作性和重用性，以及使用模型在业务和IT之间架起一座桥梁

MDA可通过开发计算无关模型 (CIM) 帮助减少业务和IT之间的差异，该模型会去除特定于计算的详细信息，因此主题内容专家可更容易地了解该模型。

组织最好开发平台无关 (platform-agnostic) 的模型，因为这些模型可以帮助组织保留其知识产权，并在技术平台实现的灾难性急剧下滑中存活下来。具有足够的平台独立性级别的平台无关

(platform-agnostic) 的模型 (PIM) 可对此提供支持，并可帮助决策人员了解规定的过程流。一旦整体架构得到批准，这些就可以使用标准转换规则转换为平台特定的模型 (PSM)，并使用标准交换格式进行交换。

MDA基于良好地建立的OMG标准 (也是ISO标准)，包括：

- 软件行业的主要公司都使用和支持的普遍建模表示法统一建模语言™ (UML <http://www.uml.org/>)，它允许业务、架构、结构和行为建模。此外，UML Profiles可作为“专业化”开发以便用于特定的计算领域，例如EJB、Web服务、本体论 (ontologies)、调度、测试和信息建模。
- 元对象设施 (<http://www.omg.org/mda/specs.htm#MOF>) OMG的关于建模语言和转换模型的基础规范。MOF定义建模语言 (元模型) 以及模型的集成、互换与管理。
- XML元数据互换 (XMI, [http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm#XMI](http://www.omg.org/technology/documents/modeling_spec_catalog.htm#XMI))，这是一种使用XML跨工具存储和交换模型的标准。
- 公共仓库元模型 (CWM, <http://www.omg.org/technology/cwm/>) 是一种描述跨仓库生命周期、业务智能、知识管理和门户技术的元数据互换的规范。CWMI将转换成为更广泛的信息管理元模型 (IMM)，该模型允许使用UML进行数据和XML建模 ([http://adtf.omg.org/adptf\\_info.htm#RFIs,RFPs](http://adtf.omg.org/adptf_info.htm#RFIs,RFPs))。

#### 2.2.1.5.1.1 作为SOA的基础的MDA

由于具有将企业和软件的所有方面进行建模的成熟度和丰富的功能，MDA很适合用作SOA生命周期的基础。许多其他的OMG标准则可通过与MDA协作对SOA提供一种由业务和架构驱动的方法。相关概述请参见：<http://www.omg.org/attachments/pdf/OMG-and-the-SOA.pdf>和<http://soa.omg.org/>。此外，MDA允许开发集成的SOA注册表/储存库，后者支持发现服务及其组件，也可映射到业务流程和目标。

其他人也对用于SOA的MDA的价值进行了评价。下面的链接提供了有关MDA价值的第三方观点：

- 模型驱动的SOA: <http://www.opengroup.org/events/q405/mdasoa.pdf>
- The Open Group的SOA工作组主管Chris Harding博士曾在文中指出,“SOA以及OMG的MDA的组合正是要得到真实的灵活性所需要的;模型驱动架构(MDA)是一个很好的概念,很适合SOA……。那么它是否可以变得更加简单,更加容易使用,并更广泛地使用以推广SOA技术呢?答案是‘是’”。<http://www.ebizq.net/topics/soa/features/6639.html?&pp=1>
- IBM的SOA和Web服务部门的首席架构师Ali Arsanjani博士在文章中指出,“面向服务的建模方法提供了建模、分析、设计技术以及定义SOA基础的活动。该方法通过在SOA的每一层定义元素以及进行各种级别的重要架构决策提供帮助。”  
<http://www-128.ibm.com/developerworks/webservices/library/ws-soa-design1/>和  
[http://www.webservices.org/categories/enterprise/strategy\\_architecture/how\\_to\\_identify\\_specify\\_and\\_realize\\_services\\_for\\_your\\_soa/\(go\)/Articles](http://www.webservices.org/categories/enterprise/strategy_architecture/how_to_identify_specify_and_realize_services_for_your_soa/(go)/Articles)

OMG现在处理软件服务配置文件标准RFP,该标准可增强用于建模服务的UML,例如服务规范和合同。有关更多详细信息,请打开下面的链接:  
<http://soa.omg.org/SOA%20ABSI%20RFPs,%20RFIs.htm>

### 2.2.1.5.2 SOA的补充事件驱动架构(EDA)和复杂事件处理(CEP)

典型的SOA环境中的连接服务的出现顺序是线性可预测的,而EDA允许多个可预测性较低的异步事件并行出现并触发单个操作。在EDA中,业务内外会出现一个著名的“事件”。“事件系统”会感知并收集这些事件,并通过服务选择性地分散的模式与所有感兴趣的方面(手动或自动)关联起来。感兴趣的涉众会评估这些事件,还可以通过调用服务,触发业务流程或者发布或组织附加信息进行响应。EDA可根据过去趋势和未来预测使事件的多元联系相关联。EDA通常被认为是SOA的子集,但把它看作SOA的补充可能更加正确。

有关EDA及其与SOA的关系的附加信息,请参见下面的链接所指向的文章:

- 广泛采用的事件驱动架构  
<http://www.computerworld.com/softwaretopics/software/appdev/story/0,10801,81133,00.html>
- 入门书:事件驱动架构 <http://www.baselinemag.com/article2/0,1540,1606126,00.asp>
- 事件驱动架构与发布-订阅系统 <http://www.developer.com/design/article.php/3499031>
- SOA中的事件驱动服务 <http://www.javaworld.com/javaworld/jw-01-2005/jw-0131-soa.html>
- 使用企业服务总线把面向服务的架构与事件驱动架构组合起来  
<http://www-128.ibm.com/developerworks/webservices/library/ws-soa-eda-esb/index.html>
- 事件驱动SOA仅仅是整个EDA世界的一部分  
[http://www.psgroup.com/research\\_michelson.aspx](http://www.psgroup.com/research_michelson.aspx)

#### 2.2.1.5.2.1 复杂事件处理(CEP)和SOA

复杂事件处理(CEP)是一种新兴的技术,可帮助公司开发和管理业务活动监控(BAM)、企业应用程序集成、网络和系统安全以及业务流程。

斯坦福大学教授及CEP权威David Luckham认为,CEP的目标是提供有关IT系统中发生情况的可理解信息。反过来,这些信息又可用于各种目的,例如检测异常活动,提高安全性以及识别CRM和供应链系统中的优势方案。

“IT系统中的事件包含未使用的信息,CEP使您能够通过希望的方式提取并使用这些信息”,Luckham说到。他认为2005年CEP开始并入Web服务、中间件和应用程序服务器。他还估计,到2008年,CEP标准、语言和复杂事件模式的搜索引擎将出现,到2012年CEP将被广泛采用。

有关CEP的附加信息,请参见: <http://www.complexevents.com/>和  
[http://complexevents.com/?page\\_id=59](http://complexevents.com/?page_id=59)

#### 2.2.1.5.2.2 MDA、EDA-SOA和CEP解决的问题

通过利用MDA、EDA、SOA和CEP，组织可以开发模型驱动的、灵活的“感知和响应”系统，该系统可（近乎实时地）检测业务价值事件并触发服务来管理这些事件。了解哪个应用程序对每个架构进行了优化以及这些应用程序之间接口的方式将有助于改进架构。此外，这样的架构方法使得业务流程及其支持功能的端对端管理变为可能。

EDA的标准及其与SOA和CEP的关系仍在发展中。由MDA作为基础，OMG SOA SIG已经启动了一个子组来关注SOA的这一方面。（<http://soa.omg.org/SOA%20ABSIG%20RFPs,%20RFIs.htm>）。

### 2.2.2 企业 SOA 成熟度模型

SOA成熟度模型可帮助企业开发路线图来达到企业的目标状态。

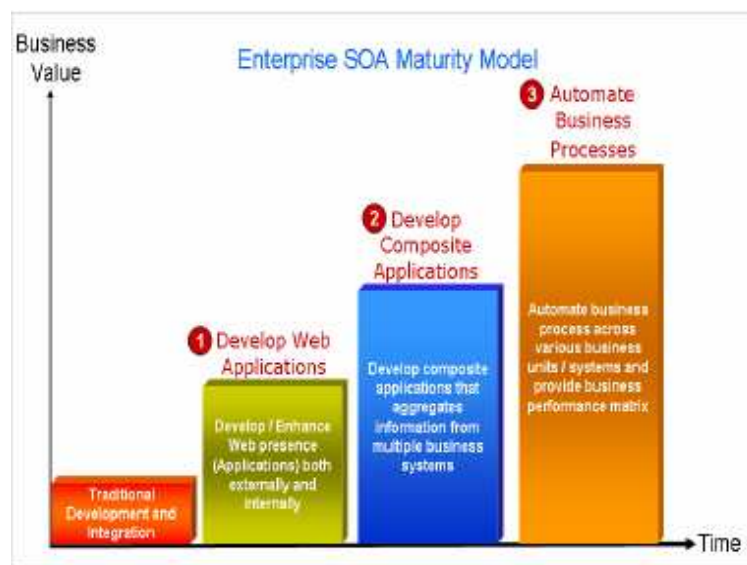


图5：企业SOA成熟度模型

上图显示了企业SOA成熟度模型各个阶段。

#### 2.2.2.1 Web 应用程序开发阶段

在这个阶段，团队关注为内外部用户提供丰富的基于客户和浏览器的业务解决方案。它们可能会选择推广基于Web的CRM、ERP或者支持连接的但偶尔会断开连接的操作的自定义应用程序。此外，IT组织通常会部署基于企业的解决方案和服务，例如内容管理、搜索、即时消息传递、博客、Wiki、讨论论坛和白板。

##### 2.2.2.1.1 业务要求

通常情况下，大多数企业都已经部署了外部网站以及许多内部网站和应用程序来支持每个业务单位的各种需要。第一个步骤是通过提供公共外观的基础架构标准化、共享和集成这些分散的解决方案。这样客户、合作伙伴和员工就可以更方便地找到所需要的信息。

在这个阶段，团队应该关注：

- 统一外部站点的用户体验，使得潜在用户、合作伙伴、客户和分析师能够容易地找到他们所需的信息

- 使所有站点（内部和外部）的外观以及发布内容的所有流程和过程标准化
- 创建一个my<company name>（例如http://my.company.com）站点，供所有员工、承包商、合作伙伴、客户个性化服务和内容
- 提供对所有内部和外部站点的机密信息的安全访问
- 提供具有高可靠性、可用性和可伸缩性的环境
- 允许使用AJAX进行站点操作来改进性能和用户体验。

#### **2.2.2.1.2 关键挑战**

这个阶段的关键挑战包括如下内容的开发：

- 应用程序支持升级路径
- 对无数并行活动的支持
- 团队的领导地位和技术质量
- 用于生产期间的开发的物理环境，以及版本管理流程和有经验的人力资源
- 专用的生产支持流程和人员配备
- 承载。

#### **2.2.2.1.3 退出条件**

出现下面的情况时团队可认为此阶段已经完成：

- 外部网站成立并且正在运行
- 为一个或多个打包的应用程序开发了门户前端
- 可通过门户站点访问一个或多个自定义应用程序
- 部署了大多数企业服务
- 业务用户可从多个应用程序请求信息
- 完成了建立程序管理办公室（PMO）以及用于部署应用程序门户的LOB治理模型的工作
- 业务部门对交付时间线很有信心，并且能够一致地构建可用于所有主要计划的程序办公室。

#### **2.2.2.1.4 成功因素**

如果具有如下条件，则此阶段是成功的：

- LOB级别的业务参与度很高
- 为所有复合应用程序建立了赞助/执行监管
- 可快速开发和交付基于Web的应用程序
- 项目管理已经到位，团队能够进行领导并了解重要程度和方向
- 开发、部署和状态报告的流程已经在LOB之间进行了标准化
- 团队已经开发了标识的和创建了有经验的资源。

### **2.2.2.2 开发复合应用程序**

复合应用程序聚集和提供来自各种源和通道的信息和数据，并在合适的时候使其对内部和外部用户

可用。Enterprise 2.0服务可提供合适的SLA级别。

#### **2.2.2.2.1 业务要求**

业务要求是指IT适应不断变化的业务要求。个别业务单位会帮助IT开发自定义应用程序，增强他们自己的品牌，提高生产力，或者为客户、合作伙伴或员工提供附加信息。

业务要求可能包括：

- 通过门户推出和公开多个应用程序
- 来自多个应用程序的信息的可用性
- 为用户提供基于Web的桌面
- 基于用户角色和责任的个性化服务
- 可降低用户培训要求的单个标准化外观
- 降低在一个平台上进行标准化的维护成本
- 降低操作和支持成本，允许IT将稀缺的资源部署在新功能上。

#### **2.2.2.2.2 关键挑战**

这个阶段的关键挑战包括如下内容的开发：

- 共享服务的应用程序支持升级路径
- 跨多个LOB多个并行活动的支持
- 共享服务的治理
- 团队的领导能力和技术质量
- 用于生产期间的开发的物理环境，以及版本管理流程和有经验的人力资源
- 专用的生产支持流程和人员配备
- 承载。

#### **2.2.2.2.3 退出条件**

如果满足以下条件，则此阶段已经完成：

- 已经创建了跨多个LOB的程序管理办公室（PMO），并且建立了用于部署共享服务的企业级治理模型
- 业务部门对交付时间线很有信心，并且使用能够完成所有主要计划的程序办公室
- 部署了多个利用SOA基础的应用程序门户
- 业务单位讨论应用程序或数据的集成时间框架。

#### **2.2.2.2.4 成功因素**

如果达到以下条件，则此阶段可认为是成功的：

- 所有复合应用程序的业务参与和赞助（包括执行监管）都已经到位
- 团队开发了快速开发和交付的方法
- 项目管理逐渐有了领导力、对紧急事务的判断力以及项目方向
- 开发、部署和状态报告的流程已经在企业之间进行了标准化以便提供共享服务

- 公司已经通过灵活的方法（并行开发）开发了有经验的资源。

### 2.2.2.3 自动化业务流程

这个阶段是指应用程序、数据和基础架构通过在合适的时间提供合适的信息来帮助用户有效地执行其角色。在这个阶段，企业开始通过将多个业务系统合并到单个系统获得较高的ROI。业务组织现在应该准备好丢弃他们的单点解决方案并转移到端对端业务流程管理的目标状态。

#### 2.2.2.3.1 业务要求

此阶段的基本要求如下：

- 业务的兴趣在于企业之间业务流程的标准化
- 通过基于标准的技术整合基础架构，同时降低成本
- 标准化业务流程可全局使用，但允许进行一定的本地化

#### 2.2.2.3.2 关键挑战

此阶段的关键挑战包括：

- 完成业务和IT转换
- 建立合适的治理和组织模型
- 实现打包的应用程序以便获得一定的短期收益。

#### 2.2.2.3.3 成功因素

如果满足如下条件，则此阶段是成功的：

- 业务参与和赞助以及执行监管使得业务和IT转换成为可能
- 有一个专门的团队关注业务流程
- 业务流程是企业的主要关注对象
- 将松耦合的业务服务装配到自动业务流程，并且这些业务服务可以重新组合以提供新的业务功能。

## 2.3 SOA 参考架构

下图显示了SOA参考架构，其中包括Web应用层、服务层、应用层和基础架构层。

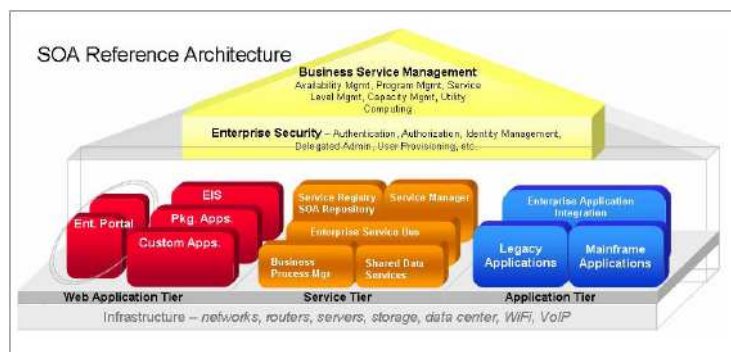


图6：SOA参考架构

并非所有IT组织都需要部署此SOA参考架构中标识的整个基础架构。一个SOA最佳试验是仅当需要提供业务解决方案时才投资于基础架构。

## 2.3.1 Web 应用层

此层的主要要求是所有业务系统和解决方案都可从任何支持的浏览器中访问。这一层是用户界面或者表示层，包含企业基础架构服务和应用程序等组件的业务逻辑。

### 2.3.1.1 打包的应用程序

通常情况下，企业会批准满足其大多数业务要求的最佳的打包应用程序，然后让IT组织和系统集成人员对打包的应用程序进行处理以便满足其需要。此类打包的应用程序的示例包括客户关系管理、企业资源计划或特定行业的成套应用系统。

现在大多数打包的应用程序都是基于Internet协议的，这意味着用户可以使用任何支持的浏览器访问许多功能。有些最新应用程序可以将有限的功能组作为离散的协作服务或外部控制的业务流程公开。

利用打包的应用程序的最佳实践包括：

- 限制自定义开发的数量，使得维护和升级简单廉价
- 尝试获得世界级的标准实现，从而减少集成和维护成本
- 在可能的情况下使用打包的应用程序提供的UI和业务流程
- 利用发布的应用程序编程接口（API）而不是直接访问数据库。

下面是在SOA成熟度模型中采用打包的应用程序的规定方法：

#### 2.3.1.1.1 开发Web应用程序

- 部署可由任何浏览器访问的应用程序的最新版本；最好是支持适当的门户标准（如WSRP）的版本。
- 公开供自定义应用程序使用的应用程序服务，最好是作为Web服务。这可能要求适配器允许访问应用程序。应用程序的一些最近版本能够通过集成网关或Web服务直接访问应用程序服务。
- 通过合并企业外观（模板、皮肤、骨架和CSS）以及集成企业单点登录解决方案以提供完美的用户体验。
- 通过集成到企业标识和存取管理程序（通常是LDAP）将身份验证具体化。

#### 2.3.1.1.2 开发复合应用程序

- 标识可以作为复合应用程序在企业之间共享的业务对象
- 将事件通知（触发器）发送到复合应用程序以启动特定操作
- 修改业务流程和用户界面以启用复合应用程序
- 公开附加业务服务以便复合应用程序能够与打包的应用程序同步。

#### 2.3.1.1.3 自动化业务流程

- 了解并建模业务流程以便标识重构机会
- 标识业务流程中可由业务流程引擎自动化的可重用部分

- 扩展公开的服务和业务流程的数目
- 减少和合并部署的应用程序的数目。

### 2.3.1.2 自定义应用程序

组织可以选择开发自定义应用程序以创建独一无二的品牌并为客户和合作伙伴提供独特的体验。这要求为内部和外部用户提供一致的完美接口。公司通常倾向于开发自定义应用程序，而不是对打包的应用程序进行自定义，因为：

- 修改核心事务的用户导航和用户界面并非易事
- 大多数主要的打包应用程序并非基于公开的或标准技术，无法对其性能进行调整以适应业务需要
- 专有开发模型使查找资源或快速部署新业务功能变得困难
- 与其他技术的集成不太容易实现，导致点对点的集成或者数据质量低下。

自定义应用程序的开发具有如下三个选项：

- 1、在应用程序服务器上开发和部署自定义应用程序
- 2、利用门户开发和部署自定义应用程序
- 3、使用基于开放标准的工具或者专用开发工具开发强大的客户端。

对于选项1和2，IT组织的第一个步骤是确定开发自定义应用程序的方法、基础架构和工具。此外，IT组织需要定义治理和组织模型以便开发自定义解决方案。

对于选项3，强大的客户端自定义应用程序通常是使用SWING、Visual Studio或类似工具开发的。这些强大客户端中的大多数都需要与一些外部系统相连接，建议的方法是利用开发标准（如SOAP、Web服务、XMPP或WebDAV），而不是直接访问外部资源。此方法使得IT组织可以方便地支持和升级集成。

#### 2.3.1.2.1 自定义应用程序业务需求

大多数企业都已经部署了外部站点以及多个内部站点和应用程序，以支持各个业务单位的不同需要。第一个步骤是在企业之间标准化这些站点的外观和基础架构，使得客户、合作伙伴或员工可以比较方便地获得他们要查找的信息。

此阶段的业务要求包括：

- 统一外部站点上的用户体验，使得潜在用户、合作伙伴、客户和分析人员能够方便地找到他们想要查找的信息
- 跨所有内部和外部站点标准化外观，并标准化发布内容的过程和流程
- 为所有员工、承包商、合作伙伴和客户创建一个my<company name>站点以便提供个性化的服务和内容
- 提供的所有内部和外部站点的保密信息的安全访问
- 提供具有高可靠性、可用性和可伸缩性的环境
- 通过公共门户帮助铭记（branding）和访问多个应用程序
- 允许用户在登录一次后访问所有服务
- 基于用户的角色和责任提供个性化服务



- 通过平台或环境的标准化降低维护多个系统和应用程序的维护成本
- 标准化外观以消除多种用户培训要求
- 降低操作和支持成本，允许IT将稀缺的资源部署在开发新功能上。

### 2.3.1.2.2 自定义应用程序架构方法

由于门户提供了一组经验证的功能来支持表示层，大多数IT组织已经开始标准化门户以便开发自定义应用程序。

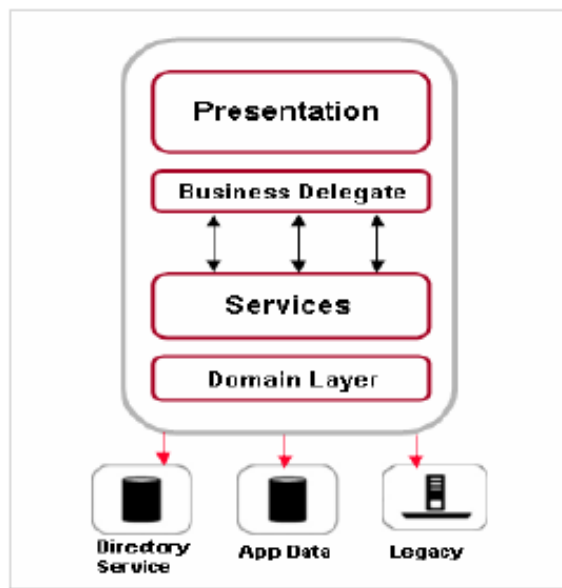


图7：建议的自定义应用程序架构

这个建议的架构具有以下优点：

- 遵循SOA原则以提高各个级别的重用性
- 提供在数周内而不是数月内提供产品或服务的能力（当具有了稳定的框架时）
- 将产品用在它最擅长的领域，例如，使用门户进行基于权限的演示
- 允许业务将服务组合起来以便提供新功能
- 在域层提取数据源和相互关系，从而将更改源系统的影响降到最小
- 使用松耦合的表示和业务逻辑使其更可靠，更可伸缩。

拟用结构中的每一个层都扮演着特定的角色：

**表示层：**“门户”负责处理所有表示服务。**Portlet**可促进用户体验的提升，其中**portlet**充当应用程序的一个视图。

**业务委派层：**业务委派层是负责表示层和业务层之间的通信的组件。它们可提取访问业务层时所包含的通信细则和复杂性。此层包括一个模型视图控制器框架，可帮助用户浏览网站。

**服务层：**服务层包含应用程序服务器的功能。该层由公开高级业务功能性的无状态功能组成。它具有一个会话表面，这是到业务层的入口点，可提取处理来自表示层的细粒度业务实体的详细信息。大多数业务逻辑都可以直接在会话表面或应用程序对象的子层上实现。

**域层：**域层也使用核心应用程序服务器。它是定义一致业务概念的业务实体集合。由于这些组件表示不变状态，此层中需要使用处理数据库存储的技术。**Entity Bean**可用于实现对象模型的一些组件。或者，简单传统Java对象（**POJO**）在数据访问对象（**DAO**）的帮助下提供一致性。**Entity Bean**是实现此层的最佳机制，但是，如果对象模型比较复杂，则需要将此技术进行组合。

### 2.3.1.2.3 自定义应用程序框架组件

自定义应用程序框架组件可扩展应用程序服务平台中固有的服务。框架组件包括：

**数据服务：**为应用程序提供的持久层。容器管理足够健壮，可以利用**CMP**完成大多数简单的事务，但作为处理复杂事务的选则应该提供**DAO**。

**记录服务：**应用程序用来记录和跟踪错误和活动的服务。要记录的消息类型包括要跟踪为进行诊断而记录的所有问题、错误或故障以及为审计跟踪和用法分析而记录的活动的调试消息。每个企业都应该标准化应用程序使用的记录服务，最理想的是利用**JDK 1.4**提供的功能。如果记录服务在整个企业都一致，则允许工作人员更有效地确定性能或事务瓶颈。记录服务应标准化机制，使其与企业内的整个开发社区进行通信，并确保与标准的兼容。不需要为此服务开发特定代码。

**异常处理：**管理和传达异常的机制。它与记录服务的类似之处在于团队应该利用标准应用程序服务器功能。团队需要确定使用哪个机制并传达给企业内的整个开发社区。无需为此服务开发特定代码，但如果团队提供了处理异常的示例的话也会十分有用。

**部署/应用程序配置：**记录配置详细信息的文档。这涉及标准化在各种环境下构建和部署应用程序的机制，包括开发、QA、UAT、阶段划分和生产。

**监控：**监控过程的标准化和文档记录。由于操作人员必须监控平台和应用程序并预测性地解决问题，IT内的大多数部门都已经标识和部署了监控工具。问题在于要标准化并集成监控过程和技术，以确保所有系统中的数据的一致性。企业可能需要购买或开发并部署附加的专业监控工具。

**搜索框架：**搜索数据的共享功能。大多数门户应用程序都需要以表格形式向用户显示数据。团队不应让每个开发人员都试图解决此问题，而应开发一个可应用于全部应用程序和门户的“搜索框架”。下图演示了一个用作搜索框架的基础架构。

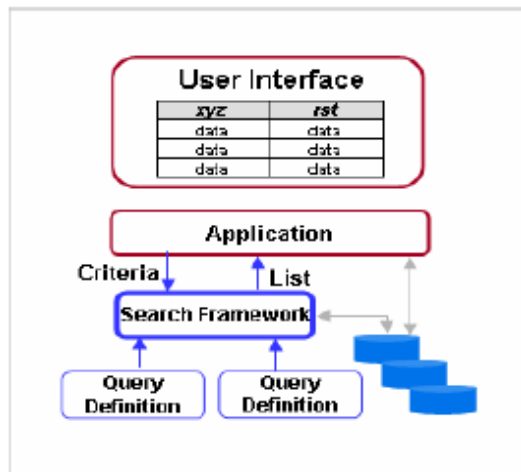


图8：搜索框架

搜索框架提供：

- 基于用户输入的动态查询生成
  - ✧ 排序次序、合并，等等
  - ✧ 用于显示的全部搜索结果

- 一致的搜索处理机制
  - ✧ 字符转义和通配符解释
  - ✧ 标记页数
- 从应用程序提取所有数据库访问代码
  - ✧ 用作输入的标准
  - ✧ `java.util.List` 等标准中需要的搜索结果
- 外部文件上的查询
- 处理公共UI任务的实用程序
  - ✧ 标记页数
  - ✧ 标准一致性。

**通知框架：**到所有应用程序的单个通知客户端。它应该支持到通知引擎的同步和异步接口，还应提供通过多个通道发送通知的能力。

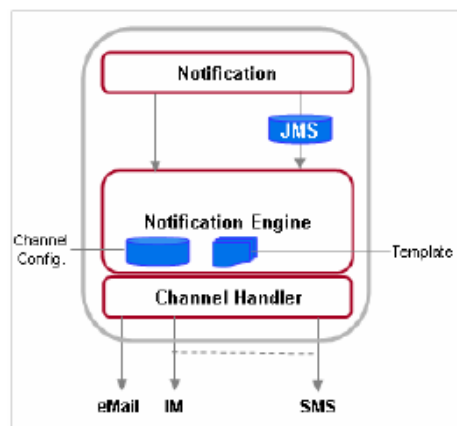


图9：通知框架

到不同通道的接口应该根据业务要求进行开发。

**服务代理框架：**提取服务实现详细信息。团队可在本地或远程部署服务，而不一定要使用服务的实现详细信息或位置对调用应用程序进行编程。相反，服务定位器会确定服务的位置并通过合适的方式进行调用。这支持多种代理，例如EJB、Web服务和Service总线代理；也可根据要求开发附加代理类型。这也可由业务委派层用于将表示层从服务层分离开来。

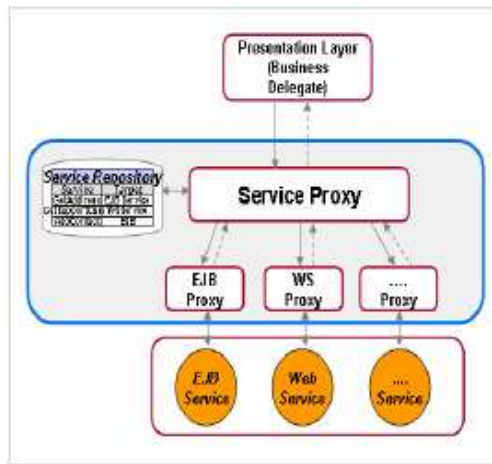


图10: 服务代理

**安全框架：**提供安全功能的企业级层。由于当前企业安全解决方案无法满足他们全部的业务需要，大多数应用程序项目团队都开发了自己的安全层。支持客户端的安全框架可减少（如果不能消除）为每个应用程序开发自定义安全代码的需要。下面是安全框架应该提供的一些功能：

- 单点登录（SSO）：登录一次后无需再次登录就可以在应用程序之间来回切换的功能
- 访问控制：用于三个主要领域的一组安全功能：
  - ✧ 身份验证：确定与应用程序交互的用户身份
  - ✧ 授权：确定用户是否允许执行特定操作
  - ✧ 审计：跟踪用户执行的操作。

此外还需要几个辅助业务，例如注册、权限授予以及权限查询。这些功能应作为可由不同的应用程序使用和重用的一般框架提供，每个功能都具有稍有不同的需要，但其基本要求都相同，包括：

- 身份管理：具有多个用于管理一组应用程序或服务的访问控制信息的存储区可能会导致严重的管理问题。身份管理可通过集中访问控制管理功能以及跨企业供应用户提供帮助。
- 整合用户配置文件：门户提供这个功能的目的是允许应用程序扩展基本配置文件。此功能从多个数据源提取用户配置文件，例如从应用程序特定的储存库提取基本配置文件和应用程序特定的配置文件。
- 注册、委派管理、供应和储存库：这些都是在访问控制顶部构建的，用于满足应用程序特定的业务需要的安全扩展功能。或者，也可能是与访问控制集成的打包的解决方案。

门户产品可提供这些非自有功能的一大部分。如果不具有自己的开发自定义应用程序的门户，IT组织将必须开发和支持此功能。

### 2.3.1.3 门户服务

门户服务管理应用程序的表示层。表示层通常都是基于权限的，因此需要支持此功能。

**表示：**门户表示功能为每个应用程序组提供外观、模板、构架和样式表。它还包括一些示例应用程序，来帮助开始开发和利用门户导航功能，包括用于垂直导航条和水平选项卡。

**个性化：**门户提供个性化服务，例如portlet布局和背景模板选择。本文将在“企业安全”的配置文件管理部分介绍应用程序环境下的附加个性化。

**身份验证：**所有门户产品都提供此功能。最佳实践就是使这项服务具体化。大多数企业都在其内部实现了全局目录服务（如LDAP）。自定义应用程序框架应提供一个身份验证界面并使服务具体化。

**单点登录（SSO）：**企业应该不需要多次登录就能提供完美的用户体验。此框架组件不仅要支持自定义应用程序，还应该支持打包的应用程序和企业服务。

#### 2.3.1.4 企业基础架构服务

这些是基于应用程序的服务，该服务可供整个外部和内部用户群使用。大多数企业服务都是基础架构组件，用户可将其中的一些组件提供的功能作为应用程序使用。下面是企业服务的一些示例。

**目录服务：**这是在全企业范围内提供的标准目录服务，通常与电子邮件服务结合使用。大多数企业都实现了元目录，以便跨企业管理身份。

**个人信息管理：**这是标准的电子邮件、日历和地址簿功能，其中包括从任何通道访问这些信息的功能。

**协作：**这提供了白板、电话会议、即时消息传递、论坛、新闻组和工作区等功能。

**企业内容管理系统：**这是驱动自定义应用程序如知识管理、资产或合同管理以及协作的基础架构服务。规定方法是利用门户或内容管理提供商提供的API。所有领先的内容管理系统都提供了开发用于上传和编写内容的模板以及用于管理批准流程的工作流的功能。

下面是实现企业内容管理系统的最佳实践：

- 首先定义分类信息，理想情况是创建企业级分类信息。
- 创建单个文件基础或储存库或企业级基础。这可能不太实用，但想法很好。
- 将所有内容发布到生产环境中的单个位置，并将所有应用程序配置为从该位置检索内容，以便降低TCO。
- 培训作者和内容审批人以使用该系统。
- 通过让提供商的架构师参与每个项目 与内容管理系统提供商紧密合作，尤其是在项目的设计阶段。
- 利用预先构建的portlet编写、审查和管理内容。
- 在SI或内容管理系统（CMS）提供商的专业人员的帮助下将业务流程映射到CMS工作流。

**搜索服务：**这就使得所有外部或内部用户都可以找到他们有权访问的信息。存在多个搜索解决方案：

1. 关键字搜索，大多数用户都习惯的标准搜索功能
2. 自然语言搜索，通常用于刚刚使用该技术，想要通过使用自己当地的语言询问问题来查找信息的非技术用户或Internet用户
3. 联合搜索，允许搜索结构化和非结构化的数据类型。

搜索引擎的集成非常简单直观。搜索引擎按请求的顺序处理XML或HTTP请求并返回结果。

搜索引擎与内容管理系统紧密协作。下面是最大程度地利用搜索引擎的一些最佳实践。

- 为内容管理系统创建企业级分类信息。
- 为内容定义元标记，并将其用于门户以便根据用户权限向用户显示内容
- 使用搜索引擎根据需要crawl和创建多个集合和子集合
- 根据需要在不同业务单位之间进行联合搜索

- 利用门户标记和权限保护安全内容
- 在应用程序服务器级别存储安全内容。

对于大型站点，**crawl**整个内容储存库的时间很成问题。根据业务需要，公司可以通过如下方式解决这个问题：创建多个集合并在搜索条件中包含所有集合，周期性执行部分**crawl**，设置多个搜索引擎，以及利用联合搜索。这有助于与搜索解决方案提供商协作开发架构和流程。

### 2.3.1.5 企业（基于角色的）门户

通过实现本文档中定义的网络层组件，企业可获得下图中描述的“当前状态”。

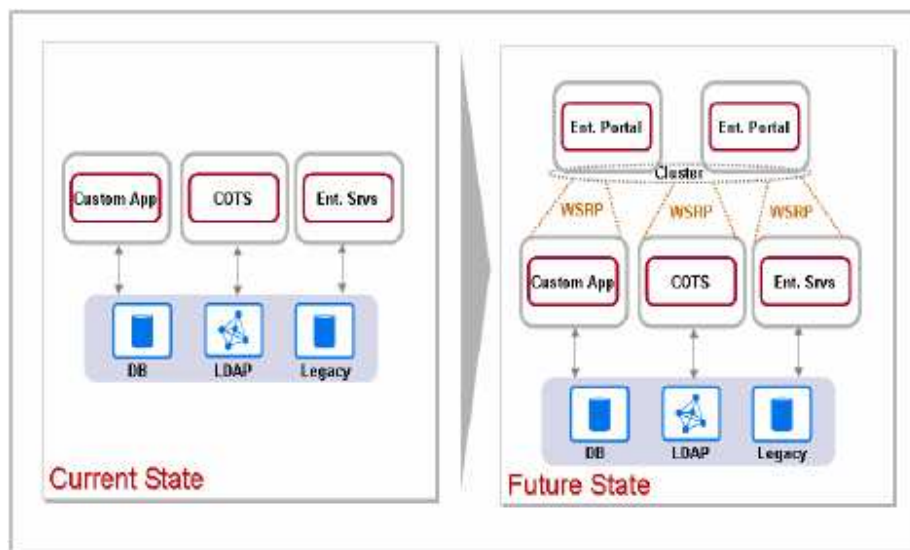


图11：企业（基于角色的）门户

在这种状态下，IT组织可以以客户应用程序、打包的应用程序、企业服务或这些组件的组的形式快速部署业务解决方案。自定义应用程序框架允许业务提供格外好的用户体验。不过，它也具有如下缺点：

- 重新树立品牌（re-branding）用户体验可能会要求对所有站点进行更改。
- 用户仍然需要知道每个站点的URL。通过采用一些最佳实践可以将这个缺点的程度减轻，但无法完全消除。
- 此模型可能会导致每个点解决方案的硬件和软件冗余。这是因为每个业务单位都想要调度自己的维护窗口，而实现这个目标的唯一方式就是为每个点解决方案提供专用的基础架构。

目标状态是利用联合门户的概念创建基于角色的企业级门户。此方法的优点如下所示：

- 为所有员工、客户、合作伙伴和其他用户提供一个入口点。
- 基于用户角色的应用程序（Portlet）访问
- 硬件和软件基础架构的整合
- 总是打开的功能
- 更简单的站点重新冠名
- 联合门户通过利用服务提供的多通道交付。

## 2.3.2 服务层

服务层是SOA的主要启用程序，包括本节中描述的组件。

它允许在企业之间进行集成与业务流程自动化。此层基于粗粒度、松耦合的和基于标准的服务的SOA原则。它通过为减少的应用程序和基础架构复杂性，业务服务增加的重用性，以及服务编排能力，来帮助IT响应不断变化的业务需要。

### 2.3.2.1 服务总线

服务总线是为IT灵活性和与业务需求的调整提供面向服务的基础架构的关键组件。它应该与服务注册表和服务管理组件进行完美集成，以便加速配置和部署管理并简化企业之间共享服务的管理。

该服务总线应该能够通过任何协议接受任何同步或异步消息，并根据配置规则将其路由到目的地。此外，它应该能够将消息转换为目标要求的格式。由于这控制消费者和生产者之间的消息流，服务总线在管理、监控和实施服务级别方面具有独特的地位。

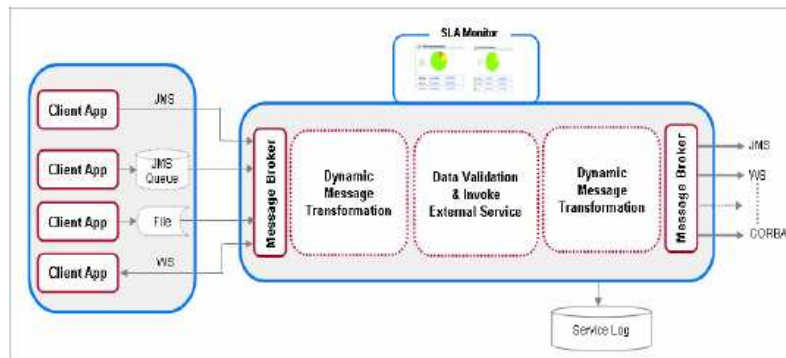


图12：企业服务总线架构

上图表示企业服务总线（ESB）。ESB充当动态可配置的消息和服务代理。它提供了以下功能：

- 异构环境之间的消息代理
  - ✧ 支持异步、同步、发布和订阅消息传递
  - ✧ 支持同步和异步桥接
  - ✧ 支持多种消息格式，包括SOAP、带附件的SOAP、XML、结构化非XML数据、原始数据、文本以及带附件的电子邮件
- 服务端点之间的异构传输
  - ✧ 支持多种协议，例如文件、FTP、HTTP(s)、多JMS提供商、RMI、Web服务、CORBA、DCOM以及电子邮件（POP、SMTP和IMAP）和SIP。
- 允许消费者与生产者交谈的消息转换，但不提供完全成熟的转换引擎
- 配置驱动的路由
  - ✧ 基于策略进行消息路由，或者调用外部服务以支持复杂路由
  - ✧ 支持点对点和一对多路由方案，允许请求-响应和发布-订阅模型
- 监控



- ✧ 带搜索功能的服务监控、记录和审计功能
- ✧ 捕获消息和传输属性的关键统计数据，包括消息调用、错误、性能、容量和SLA违反情况
- 高可用性
  - ✧ 支持集群并跨集群收集统计信息以审查违反SLA的情况
  - ✧ 简化服务供应
  - ✧ 通过配置动态部署新的服务版本
  - ✧ 在设计、阶段划分和生产之间迁移配置的服务和资源
  - ✧ 支持消息资源的多个版本，通过灵活路由进行选择性的服务访问来配置这些版本。
- 可配置的策略驱动安全
  - ✧ 支持用于进行身份验证、加密-解密和数字签名的最新的安全标准
  - ✧ 支持用于HTTP和JMS传输的SSL
  - ✧ 支持多个身份验证模型
- 策略驱动的SLA执行
  - ✧ 根据各种属性建立SLA，包括吞吐时间、处理容量、消息处理的成功/失败比率、错误数、安全违反情况和模式验证问题
  - ✧ 使用灵活的机制启动自动警告或启用操作人员启动的对规则违反情况的响应，这些机制包括电子邮件通知、触发的JMS消息、触发的带JMS消息的集成过程，带JMS消息的Web服务调用或者管理控制台警告。

下面是服务总线的一些最佳实践：

- 在任何时候服务数目超过50时采用服务总线。如果服务数目超过150则必须使用服务总线。
- 从小型规模开始，例如将目标定位单个复合应用程序或为跨越多个系统的分部业务流程。
- 考虑让多个LOB根据自己的策略管理自己的服务总线，并使用一个跨不同的业务单元共享服务的可充当代理的企业级服务总线。
- 确定是部署供应商提供的服务总线还是内部开发的抽象层。

### 2.3.2.2 服务注册表

SOA要求服务是粗粒度、松耦合的和基于标准的服务。开发和部署服务后，必须为架构师、开发人员、操作人员和业务经理提供可用的服务目录。

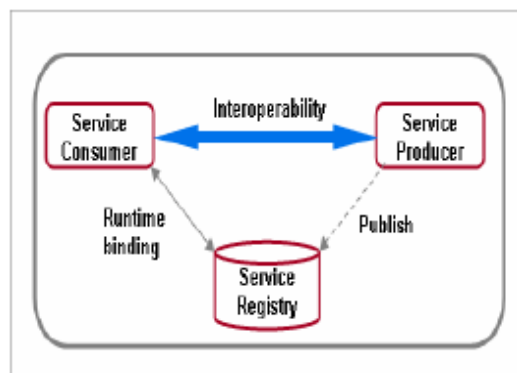




图13: 服务注册表

上图演示了服务注册表的架构。服务生产商为服务消费者用于运行时绑定的服务注册表发布了服务。注册表还可充当业务策略在运行时执行的记录系统。

服务注册表应该提供下面的功能:

- 核心服务, 包括复制、UDDI数据存储和安全
- 信息服务, 包括数据验证、SOA映射、高级分类以及业务数据访问服务
- 生命周期服务, 包括批准和更改管理、更改通知、业务服务发现和QoS管理
- 配置的基于Web的业务服务控制台
- 与领先的启用、管理和安全产品连接的独立于平台的开放架构。

服务注册表的最佳实践包括:

- 从小规模开始并逐步增长
- 将服务注册表的每次实现复制到企业服务注册表
- 为架构师、开发人员和操作人员提供服务浏览功能以帮助实现重用和标识服务依赖关系
- 维护服务合同信息以及服务定义
- 对所有服务进行版本控制。

### 2.3.2.3 SOA 储存库

SOA储存库是在服务生命周期(从项目初期到完成)中管理元数据的关键组件。其主要目的是存储详细元数据以便在部署前管理和治理资产。SOA储存库存储服务定义, 以便团队可以检查企业内定义了哪些服务。SOA储存库还存储了其他类型的元数据, 包括过程映射、业务规则、实体和关系、编排、转换、参考数据模型、业务活动和事件、审计要求、角色和授权映射, 以及治理规则和策略。

储存库还可通过标识和管理依赖关系以最大化重用率来帮助减少“服务sprawl”。如果储存库中包含来自组织的所有产品的元数据, 则可为架构师和IT决策人员提供有关服务、系统和数据依赖关系的有价值的总体观点。为了帮助业务、IT和操作人员正确的决策, SOA储存库应该将有价值的资产存储为标准化治理流程、供销业务和技术最佳实践以及开发方针。

SOA储存库可在设计阶段帮助治理SOA资产。他们允许在SOA生命周期的不同阶段共享元数据, 并在资产填充到储存库中时提供一个理想的位置以触发批准 workflow。储存库可帮助确保在资产在其生命周期中得到合适的批准, 同时降低服务的“独立开发”并支持进行较高级别的重用。

SOA储存库还提供了一个中心位置, 用于管理与运行时执行(如路由、安全和SLA)的服务关联的策略。储存库的依赖关系跟踪和影响分析功能可帮助团队管理对策略或其他资产的更改, 并预测性地分析更改对其他资产的影响。

SOA储存库应提供如下内容:

- 发布和发现元数据(服务、业务流程、用户交互等)的能力
- 按类别、复合应用程序名称、服务描述、范围(如分部或部门)或者储存库内跟踪的任何其他元数据类型进行的复杂的元数据搜索
- 服务依赖关系映射, 用于跟踪服务依赖的资产以及依赖于此服务的资产
- 元数据更改的通知
- 可扩展元数据分类信息, 以便业务可以根据自己的要求自定义分类信息

- 授权过程，用于控制哪些人可以创建和操作储存库中包含的元数据
- 所有资产的版本信息的维护
- 影响分析，用于在进行更改前预测性地测量更改的影响
- 与开发环境同步的开放的可扩展界面
- 与其他外部存储的同步，例如注册表或其他储存库
- 基于设计时策略的元数据的设计时语义验证
- 批准工作流，用于提升或拒绝元数据
- 多个同步储存库的联合和分区。

根据服务及其依赖关系的数目，为三个以上的项目采用SOA的IT组织可能会需要SOA储存库，尤其当他们具有多个分布式开发团队以及许多服务时。如果组织成熟到可以重用服务来开发复合应用程序、流程或服务，就需要使用储存库以便管理和治理重用性。

#### 2.3.2.4 服务管理器

企业内的SOA实现变得越成熟，对总体服务管理器的需要就越发强烈。此服务管理器的主要功能在于管理、监控和报告整个企业内的所有服务。下面是服务管理器需要提供的一些功能：

- 管理和维护整个企业的服务级别
- 跨企业映射和管理服务层次结构，并为操作提供依赖关系指标
- 检测和管理异常情况
- 审查和监控业务事务，并提供审查动态事务的功能
- 管理服务生命周期并在部署前进行验证
- 跨不同的系统提供非插入式服务发现功能
- 管理多个服务总线和服务注册基础架构并与之集成
- 利用现有监控基础架构。

#### 2.3.2.5 共享数据服务

共享数据服务提供了多项关键功能：

- **电子数据互换（EDI）**，使用网络在不同公司之间传输数据
- **数据操作**
  - ✧ 提取、转换和加载（ETL）
  - ✧ 企业信息集成（EII）
- **数据质量**
  - ✧ 数据匹配引擎
  - ✧ 数据工作
  - ✧ 工作流。

EDI和ETL是传统的方法，对于以批量模式处理大量数据尤其有用。不过，SOA有一个要求，就是能

够将一些EDI/ETL功能作为服务调用。例如，电子基金转帐必须从事件触发的源系统填充操作数据存储。

### 2.3.2.5.1 企业信息集成 (EII)

EII引用软件系统，此软件系统可从各种内部和外部源获取不同格式的数据并将其看作单个数据源。

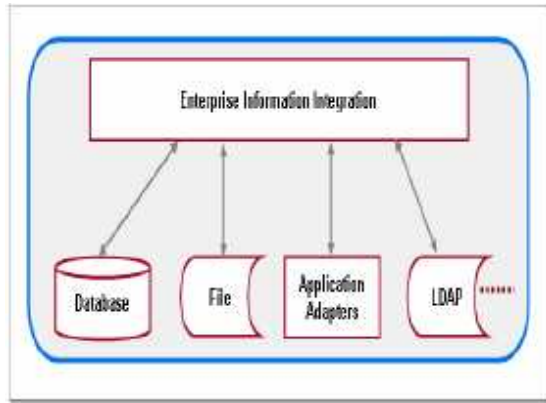


图14：企业信息集成 (EII)

EII应该提供以下功能：

- 跨多个源的数据建模
- 查询（读写）开发以便从多个数据源提取信息。
- 支持多种数据源，例如数据库、文件、应用程序适配器、LDAP和Web服务
- 数据转换
- 数据验证
- 使用RMI或Web服务将数据服务公开到客户应用程序。
- 坚持SQL、XQuery、XML、Web服务、JDBC以及J2EE等标准。

虽然定义了服务数据对象（SDO）标准以简化和统一应用程序处理数据的方式，但业界尚未清楚地定义EII的标准。所有供应商都具有自己扩展，来处理对于每个数据存储的数据读取、数据更新以及数据插入操作。

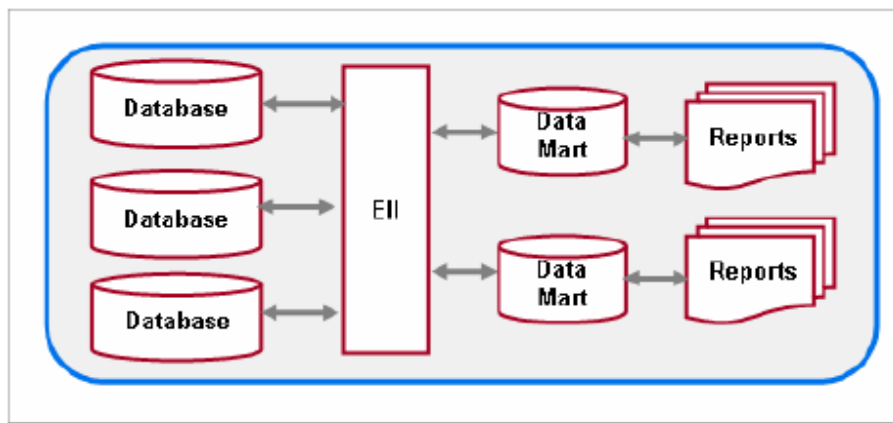


图15：将EII用作数据集市

最佳实践是使用业务智能（BI）工具从数据仓库、ODS或数据集市提供分析报告。ETL是面向批处理的，因此业务通常得到的是延迟的报告，不总是能够表示企业的当前状态。为了提供实时信息，IT组织开始利用EII工具直接从源系统将数据实时提取到BI工具中。所有主要的BI供应商都支持此方法，这还可以促进ETL和EII工具之间的会合。

这种会合不会影响组织的文化，尤其当大多数数据架构师、数据分析师和DBA更熟悉ETL工具而不是EII工具时。在进行此类项目前，IT组织应该计划培训人员和引进EII专家。

### 2.3.2.5.2 数据质量

企业需要关注数据质量，因为：

- 数据质量直接影响信息质量
- 数据质量低下会导致依赖该数据的业务流程效率低下
- 根据低质量数据进行重要决策会导致非常严重的后果
- 低数据质量又会反过来影响组织，降低客户信任度
- 如果数据错误，公司会浪费时间、金钱甚至是损害其名声。

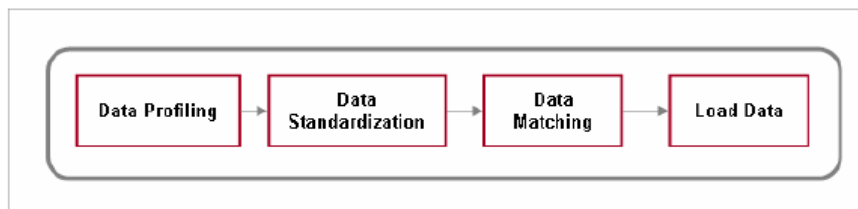


图16：改进数据质量的过程

要改进数据质量，组织需要关注：

- **数据分析：**创建高质量数据环境的第一个步骤就是了解当前环境下的数据质量水平。度量数据质量改进计划是否成功就在于在项目开始时是否正确评估数据质量状态。
- **数据标准化：**组织必须定义和应用数据标准化的业务规则。
- **数据匹配和加载数据：**IT必须使标准化数据与现有的经过清理的数据匹配，并创建或更新现有数据。此外，IT必须公开共享服务以便客户应用程序利用数据匹配。

除了数据质量基础架构，企业还需要一个主数据管理（MDM）功能。由于使用了这么多不同的应用程序，组织会发现企业内外存在许多种对实体的表示方法。MDM合并和合理化关键实体（例如客户和产品）的表示方法，以确保得到精确一致的信息。

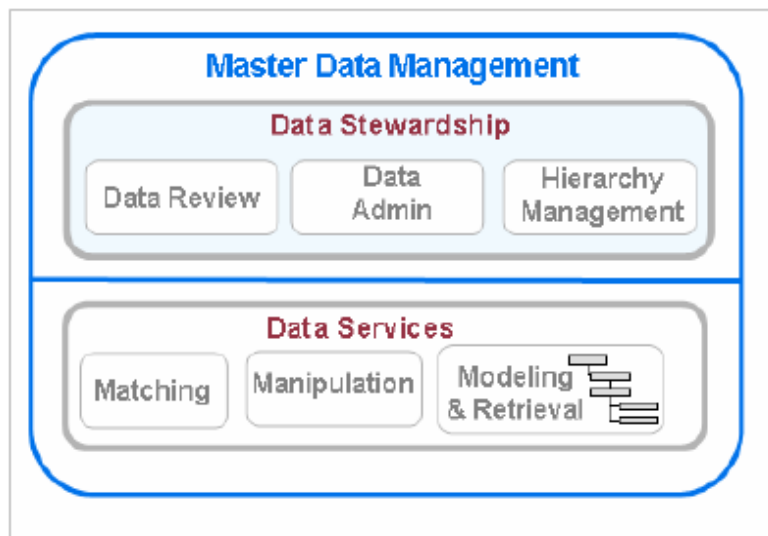


图17：主数据管理

MDM由数据服务和数据工作组成。

- **数据工作**是允许用户审查异常数据，管理数据并创建多个层次结构以进行报告的应用程序。
  - ✧ **数据审查**提供审查链接的和未链接的数据并进行校正操作的功能
  - ✧ **数据管理**允许工作人员修改或手动覆盖匹配项，修改匹配项和数据生存规则，并管理用户。此外，它还允许团队利用第三方源的数据优化规则，例如Dun & Bradstreet。
  - ✧ **层次结构管理**允许工作人员根据行业、地理位置、收入和组织创建主数据层次结构。
- **数据服务**是为MDM提供基础架构的一组可配置规则。其功能包括：
  - ✧ **匹配**，使用模糊逻辑根据业务的预先定义的匹配规则匹配和标准化主要数据。
  - ✧ **操作**，用于数据移动和微调。此外，它还提供用于匹配、清理和标准化数据以及实施数据生存规则的基本 workflow
  - ✧ **建模和检索**，使业务组织可在IT有限参与或不参与的情况下修改MDM。传统情况下，ETL或自定义工具提供了检索功能，但现在各公司通常使用EII工具实现此目标。

### 2.3.2.5.3 最佳实践

下面是共享数据服务的一些最佳实践。

#### 2.3.2.5.3.1 实现主数据管理

公司需要关注总体业务流程才能获得SOA的全部优点。由于企业要使用数据获得事实真相，因此将企业数据流映射到与其相关的业务流程十分重要。下图是将关键数据元素映射到业务流程的重要性的示意图。

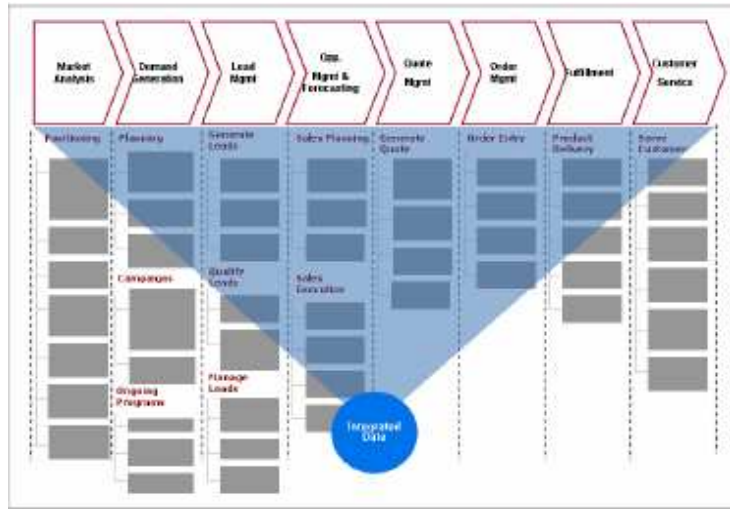


图18: 集成数据

在采用SOA之前，开发集成数据模型并非易事。建议的方法是在实现SOA时分阶段实现数据模型，每次完成一个业务流程。业务需要区分业务流程的优先次序，而IT则并行工作以开发一个将数据流映射到业务流程的企业数据模型。

各公司很快意识到，在存在许多流程和子流程的同时，还存在一组有限的企业数据对象，例如客户、合同和产品，这些对象是大多数公司都需要的。数据模型的普及对于组织来说是一个威胁，这个威胁需要通过实现MDM解决方案解除。

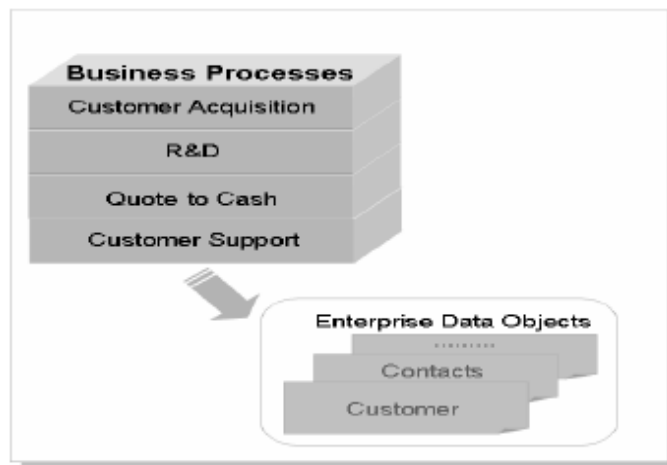


图19: 标识企业数据对象

正在实现MDM的组织发现：

- 从来就不存在什么捷径。进行高级别业务流程映射可帮助标识企业公共对象或实体。
  - ✧ 业务定义流程
- 解决MDM（公共对象）是优先级最高的任务。
  - ✧ 业务定义企业的数据需要
- ODS和数据仓库可通过使用公共对象/模型派生。
  - ✧ 团队根据需要开发合适的基础架构

- 业务项目团队必须开发填充ODS/数据仓库的功能或MDM解决方案。

#### 2.3.2.5.4 结构化和非结构化数据的会合

“结构化数据”指数据库中存储的企业数据，而“非结构化数据”是指存储在不同文档中的企业数据。目前大多数企业都具有搜索非结构化数据的解决方案，也可以利用应用程序访问结构化数据。不过，用户可能希望能够在相同的页面上同时审阅结构化和非结构化数据。

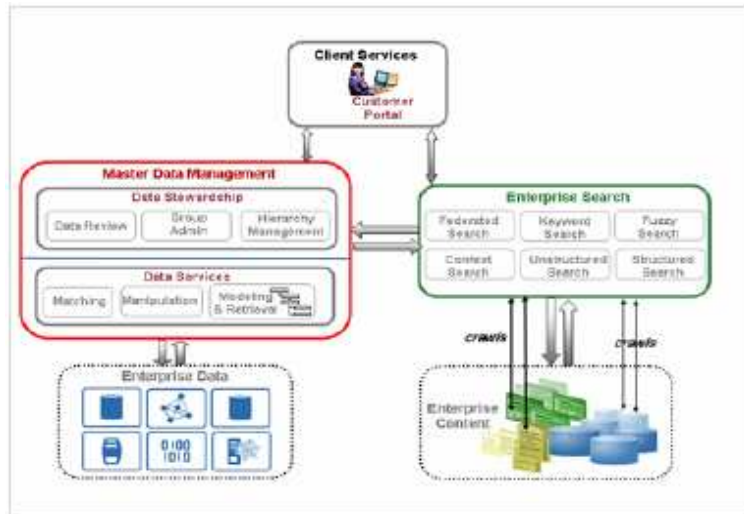


图20：聚合结构化和非结构化数据

会合要求有三个组件：

- 向用户显示信息的门户
  - ✧ 联合门户利用WSRP来显示结构化和非结构化信息
- 搜索引擎
  - ✧ 将非结构化内容提供给门户
- 共享数据服务
  - ✧ 访问结构化数据，最好是使用EII
  - ✧ 利用MDM搜索企业数据对象。

每个企业搜索组件都扮演了一个不同的角色：

- **联合搜索：**将搜索请求提交给外部服务提供商
- **关键字搜索：**根据关键字进行搜索
- **模糊搜索：**根据自然语言或模式搜索
- **上下文搜索：**维护搜索的上下文，而不一定包含操作符
- **非结构化搜索：**搜索不同类型的内容，包括200个以上的文档类型
- **结构化搜索：**搜索数据库

#### 2.3.2.6 业务流程管理

BPM用于管理长期运行的同步和异步业务流程。服务总线执行轻量级服务编排，而BPM执行下面的

功能：

- 可视流程建模，以修改视图和业务流程模型
- 开放标准兼容，最好是BPEL
- 私有流程、公共流程、人员任务和错误处理之间的业务流程编排和自动化
- 支持高级建模的嵌套的并发流程以及启用快速自定义需要的自定义逻辑
- 优化的流程性能以提供配置全状态（长期运行）和无状态（短期运行）流程设计模式以及同步和异步流程执行的灵活性
- 状态监控功能，以便以图形化方式显示用户端到端流程并根据服务级别协议测量性能
- 流程实例监控，以提供有关运行流程的统计数据，找出单个的详细信息，并终止、删除或延迟有问题的流程实例
- 允许任务创建人、工作人员和管理员与运行中的业务流程交互，以便处理流程异常、批准和状态跟踪
- 用户和组管理，以集中分配集成项目的角色、用户和组
- 通过使用领先的标准协议（如RosettaNet、ebXML和EDI），安全的消息传递、数字签名和加密，可恢复和可跟踪的消息以及动态配置升级，B2B协议支持与供应商和客户的快速安全的在线连接。

### 2.3.3 SOA 框架

SOA框架是作为SOA主干的可重用服务。这些可重用服务必须是企业级服务，具有良好的设计以至于可根据负载进行伸缩，并可满足各种消耗很大的应用程序和涉众的需要。

通过帮助开发团队快速设计、开发和部署设计良好的、模块化的、灵活的、可伸缩的和可支持的Web服务、Web应用程序和portlet，SOA框架支持到SOA的移动。在公司开始采用SOA原则转换IT架构时，必须创建这些基础服务以作为企业级资产。

框架可定义为可重用的主干应用程序，团队可对其进行扩展以构建特定服务或应用程序。框架可改进提交的软件的一致性。框架可控制本身以及以其为基础创建的应用程序或服务。

框架通常提供一组高级编程提取以及创建企业级服务的强有力的起点。他们通常为合并多个设计模式和软件工程最佳实践的服务指定一个分层架构。该架构还指定各个层中组件的责任以及它们之间的协作。

服务继承内置到框架中的有效架构和最佳实践。通过使用框架，一组中等水平的开发人员就可以开发架构良好的、能够利用设计模式和最佳实践的服务。服务创建框架可提供的典型的层包括：

- **转换层：**支持协议和数据类型转换以支持多个访问协议，同时又保持大多数（如果不是全部）服务实现协议和访问机制的广泛适应性。
- **业务逻辑层：**保存系统中的所有业务逻辑。这包括如请求、结果、UseCaseController和BusinessPolicy对象等的提取。
- **业务数据层：**域对象的层，所谓域对象是指在企业中的许多应用程序中都具有一致定义的对象。业务数据层应该提供位置透明性，这样域对象的用户就不需要知道基础一致数据的确切物理位置。此层应该能够管理企业中的各种持久性储存库的持久性，并可对这些储存库进行检索。
- **集成层：**从JDBC到JNI到Java连接器的连接技术实现的占位符。访问扩展企业系统（例如ERP系统和内容储存库）所需的所有基础架构代码都放入此层。



SOA框架可同时为开发人员和公司带来好处。当开发人员使用框架时，他们：

- 获得创建服务、Web应用程序和portlet的坚实基础
- 通过合并设计模式和最佳实践提高生产力
- 利用框架的现有功能并编写少量代码
- 不需要了解J2EE标准和规范的具体细节
- 不是面向对象的设计和模式模式的专家也可以从中受益。

对于IT组织和公司这个整体，SOA框架提供了：

- 快速并以低廉成本实现SOA的催化剂
- 项目之间设计和开发一致性
- 重复性以及保证架构和设计的严格程度最低的能力。
- 通过很容易更改（通常是通过配置更改）的模块化解决方案得到的改进的业务灵活性。
- 在不同技术水平的开发人员中利用软件工程最佳实践
- 更一致的、可预测的和得到更好测试的解决方案
- 项目中开发人员更好的可动性

IT组织正在使用SOA原则积极创建封装和公开关键业务流程的可重用服务。通过把分层架构、易用性、以及对良好架构和重用性的重视结合起来，SOA将各公司组织起来以便创建供应商中立的和可移植的任务重要的企业级服务。

## 2.3.4 应用程序层

IT组织的最大投资在于应用程序。企业在SOA方面的投资不断增长，但应用程序层不会很快发展。

### 2.3.4.1 旧应用程序

这些是强大的客户端应用程序。它们可能是打包的应用程序的尚未升级的旧版本。这些应用程序可能最适合用于客户端/服务器模式。

这些应用程序通常具有一些发布的API或逻辑模型以进行集成。

### 2.3.4.2 大型机应用程序

运行于专有的大型机系统之上的业务解决方案可以通过消息传递或数据库网关与其他系统集成。大型机软件供应商正在致力于将所有API作为Web服务公开。不过，在大多数情况下，最好的方法是通过利用中间件开发抽象层并公开合适级别的服务来支持业务要求。

### 2.3.4.3 企业应用程序集成

企业应用程序集成（EAI）通常使用中间件提供下面的功能：

- **消息传递：**面向消息的中间件（MOM），用于允许资源发布和订阅消息
- **业务流程管理器（BPM）：**自动化业务流程的专有功能
- **应用程序适配器：**预先配置的到各种打包应用程序的连接器，允许访问应用程序视图或者其他

技术的技术适配器（如数据库、消息传递（MQ和JMS）和Web服务）。

EAI的最佳实践是利用集成集线器。不过，如果没有合适的支持方法，这会在集线器上导致点对点的解决方案。虽然服务层提供了服务总线和BPM，允许企业采用SOA并从EAI迁移出来，这种迁移也需要花费很长的时间，尤其当IT组织在EAI方面进行了大量投资时。

### 2.3.5 企业安全

目前大多数打包的或自定义应用程序都实现了自己的安全解决方案。不过，每个企业都具有多个应用程序，每个应用程序都具有自己的身份验证实现，这样在不同的时间更新不同的身份验证实现时就有可能会引发问题。

这个问题可以通过将企业安全划分为三个主要组件得到简化：

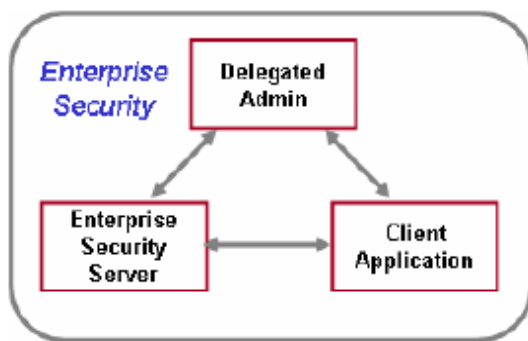


图21：企业安全组件

**委托管理：**用户和资源管理应用程序允许管理员根据用户角色相应地创建、修改和删除用户特权。此应用程序更新与企业安全服务器所使用的相同的存储库。

**客户端应用程序：**具体化和利用企业安全服务。

**企业安全服务器：**提供安全服务，例如用户身份验证、用户身份管理、授权、审计、用户配置文件管理和用户供应。

- 身份验证通过多种身份验证机制验证用户身份。可以根据密码或者通过数字证书或智能卡验证身份。为了提高生产力，用户需要能够通过一次登录访问多个资源，但企业需要实现正确的控制和策略，因此进行一次身份验证后用户只能访问得到使用授权的资源。
- 身份管理将多个身份映射到单个用户，或者跨应用程序链接身份以支持多个用户身份的共存。
- 授权涉及控制用户对整个企业内的多种资源的访问。
- 审计涉及跟踪用户活动。
- 配置文件管理及管理用户的配置文件。
- 供应自动完成管理账户及其生命周期的费时过程。它允许跨企业内的多个应用程序集中激活、修改或取消账户。用户供应应该包括显式授权或调用用户对资源的访问以及建立用户授权策略。

### 2.3.6 业务服务管理

业务服务管理主要关注监控和报告网络、基础架构、应用程序或业务流程的操作。

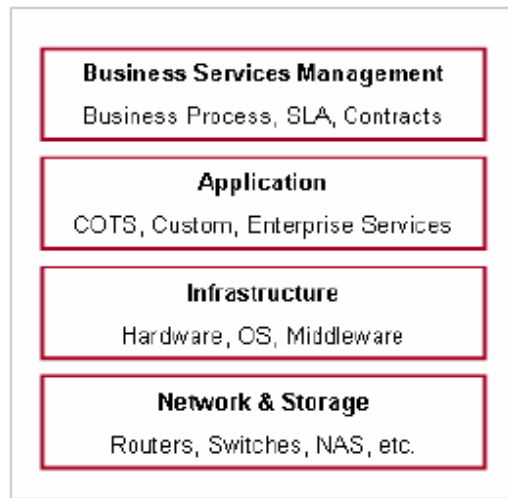


图22：业务服务管理

操作部门通常有专门的团队来架构、设计和监控此网络，其中大部分广域网是从电信网络提供商那里购买的。

**基础架构：**所有硬件、操作系统和中间件，如应用程序、服务器、数据库和LDAP。IT操作部门负责为基础架构层监控和提供第1层支持，基础架构层有时也称为数据中心监控。

**应用程序：**监控应用程序和企业服务，如电子邮件、搜索和内容管理系统。IT操作部门提供第1层支持，而有一个专门的团队提供应用程序支持。

**业务服务管理：**监控业务流程、策略、活动和IT服务级别的协议。

### 2.3.6.1 监控的典型当前状态

大多数企业不具有综合性监控功能，从而导致：

- 功能相似的多个监控系统
- 计划的人员配备无法达到需要的水平，导致软件仍然各自为政
- 仅有有限的或者不存在跨业务或产品silo关联事件的机制
- 缺乏统一的IT配置视图来帮助进行关联
- 仅具有有限的或者不存在将配置元素映射到SLA管理的业务流程的服务模型视图
- 仅具有有限的跟踪历史可用性和服务性能的机制
- 仅具有有限的返回关联关系以便防止重复通知的集成。

下面是支持BSM所需的功能的完整列表。IT操作部门需要审查当前状态并开发一个路线图以部署这些功能。不幸的是，IT组织的最大挑战是获得筹资批准，尤其当业务想要使业务功能的优先级高于监控和管理功能的优先级时。

- **票务：**跟踪已经标识的由事故导致的问题
- **事件：**通过监控系统或服务台接收的任何警报（不一定是问题）
- **通知：**将票证（ticket）转发到服务台或负责的操作工作组；这可能包括以前的为分页标识的要求
- **诊断和根源分析：**诊断问题以标识根本原因的工具；诊断过程通常包括已知问题解决方案或变

通方法的数据库

- **事件控制台：**提供存储在监控数据存储中的可用性和性能管理数据的视图
- **关联引擎：**用于解释和使用来自不同的源的数据的软件
- **决策引擎：**根据关联结果，此软件决定需要执行什么操作，例如记录票证，警告负责的部门，运行脚本，或者不执行任何操作
- **监控数据存储：**包含事件数据和配置信息的中央数据库，也称CMDB
- **服务级目标：**与应用程序或业务流程关联的可用性和性能目标的中央储存库
- **服务级报告：**针对服务级别目标测量实际性能和可用性的能力
- **自动化引擎：**允许针对已知问题自动执行操作，如记录票证，运行脚本或启动其他恢复流程
- **目标方法：**用于管理已知问题的流程和工作流。
- **自动脚本：**测试或响应已知问题的预定义操作
- **容量计划：**配置项目的当前的或预测的利用率
- **系统建模：**组成系统的配置项目的表示形式
- **工作负载模拟：**使用根据测试基础架构生成综合事务的自动化工具进行的性能和容量测试。
- **探测器：**收集有关设备的故障或性能信息，但未安装在设备上的软件
- **系统监控器：**安装在计算机和网络设备上，收集可用性信息的软件
- **最终用户监控器：**通过综合性事务或监控实际的用户活动测量最终用户体验的软件
- **业务可用性控制台：**从业务的角度提供可用性管理数据如何影响业务的信息

#### 2.3.6.2 业务服务管理的未来状态

SOA允许业务通过业务服务管理来监控和管理其业务流程。业务必须在要求阶段定义和捕获业务流程策略并将其实现。为了支持这个目标，IT组织需要投资于附加软件并保证附加资金的安全，无论是用于基础架构还是人数。

### 2.3.7 面向服务的基础架构（SOI）

由于基础架构需要支持SOA，面向服务的基础架构（SOI）要求资源计算、联网和存储数据：网络、路由器、网络设备、服务器、存储、数据中心、VoIP和无线功能。

目前，存储产品通过SNIA SMI-S标准整合了可管理型模型和操作过程。SOI将该标准作为SOI架构的一部分使用。相反，SOI网络集成主要依赖于专有协议。除了过时的简单网络管理协议（SNMP），目前尚不存在可用的行业级可操作管理标准。

对于计算资源，存在两个由DMTF开发的过程产品的标准。DMTF服务器管理工作组（SMWG）定义的服务器可管理性标准关注单个系统级别，以便在预启动前定义服务资源建模。实用计算工作组（UCWG）定义处理虚拟的服务器池管理建模的标准。架构师可将这些阶段产品标准用作基础，并建议进行附加扩展以便利用技术获得更好的性能和其他功能。

SOI计算资源存在两个主要组件。第一个是基础架构管理器，这是一个聚合计算资源并将其提取到虚拟计算池中的软件。第二个是计算资源本身，它可以组合启用计算资源虚拟化的自动功能的软件和硬件模块。

一旦基础架构管理器聚合和提取了计算资源，IT就可以选择分区功能，例如传统的一个服务器一个操作系统的分区，在一个计算系统中提供多个操作系统分区，以及跨多个系统提供的操作系统分区。基础架构管理器中的软件功能允许IT实现SOI服务：

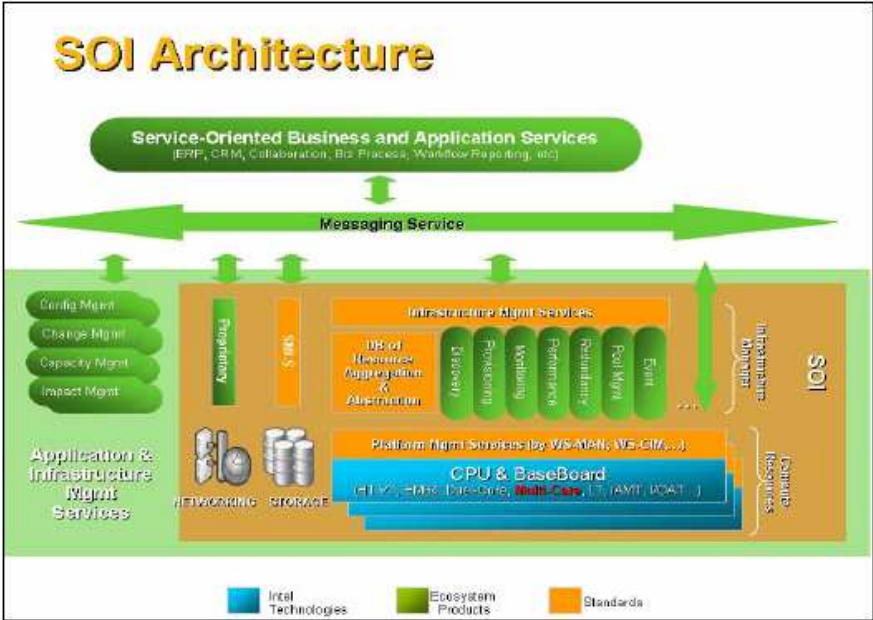


图23: 面向服务的基础架构

- **资源聚合和提取的数据库：**此储存库是SOI架构的信息基础。根据DMTF的公共信息模型（CIM）将计算、网络 and 存储资源提取到逻辑模型中后，它就会存储这些资源。根据实现，储存库可能仅提供简单的资源配置，也可能包含计算、存储和网络资源及其操作流程的依赖关系。

一旦IT定义了资源及相关性的公共提取，此储存库就可在内部和外部应用程序之间共享和同步这些信息，以确保资源管理的互操作性和一致性。该数据库还包括操作过程，即充当标准化IT流程的构建块的泛型IT微操作。例如，服务器供应可使用跨服务器操作系统安装、网络转接VLAN设置和SAN存储分区编排配置过程的泛型操作进行简化。目前，DMTF实用计算工作组正在关注这些用于服务器虚拟化的泛型微操作流程。随着范围的扩展，UCWG或其他标准团体将标准化更多的资源操作流程。

IT工作人员可利用这些构建块为自己的环境开发更复杂的自定义流程。产品供应商可以扩展操作流程以更好地利用其产品功能。供应商应该通过根据活动管理技术（AMT）扩展“裸机供应”的操作流程将自己与竞争对手区分开来。

- **基础架构管理模块：**这些软件模块提供了服务的执行。服务可以包括自动资源发现、操作系统虚拟分区的供应、资源运行状况监控及其事件通知以及资源池管理。SOI SDK可以通过提供软件示例和库帮助加速这些管理模块的开发，这些示例和库显示了开发人员是如何与提取和聚合资源数据库进行交互以及如何注册WSDL服务以及将这些服务应用于外部应用程序的。
- **基础架构管理协议：**这一组XML协议允许上层业务或系统管理应用程序（基于Web服务的应用程序或旧应用程序）以便与基础资源交互。SOI管理模块封装其复杂性，将其与业务或系统管理应用程序隔离，并在内部维护运行时详细信息。此管理协议规范定义了管理交互的资源对象处理程序的详细信息。

例如，J2EE应用程序服务器可能会达到性能阈值，需要更多的资源才能遵循其SLA。这样，它会根据需要的jAppServer SPEC基准向SOI发送一个XML资源请求以要求提供附加计算资源。该请求可能会指定应用程序配置和操作流程以加入J2EE应用程序服务器集群。不过，此XML请求中并不需要指

定详细的系统和操作系统配置，因为SOI将自动验证可用的J2EE应用程序服务器平台并以SOI中的可用选项进行回应。这可能基于使用指定JAppServer SPEC基准和可用的操作系统和可用性的专用Xeon服务器或共享的Itanium 24路服务器。J2EE应用程序可根据资源的可用性和成本选择一个选项。SOI将跨计算、网络 and 存储资源维护配置上下文，并封装来自业务应用程序的信息。这仅仅是资源性能指标的一个起点。当前的SPEC基准指标还需要进行大量改进和扩展才能完成SOI的最终目标。

除了上述基础架构管理器，计算资源本身的许多关键硬件和软件功能都允许实现SOI服务。

- **资产标识符：**提供一种可靠的方式远程唯一地标识附加到网络的物理资源并提供相关目录。这就允许企业管理控制台发现和计算连接到公司网络的所有IT资源，包括无线设备。
- **带外（OOB）可访问性：**允许网络访问平台管理功能，即使平台因为尚未加载操作系统或应用程序软件或者由于软件bug、系统故障、恶意攻击或其他问题而无法工作。这不会要求另外提供网络电缆，但在平台本身内提供了替代路径和智能。此功能允许远程“逻辑”供应，重新加载损坏的软件、运行远程诊断、远程重新配置以及其他的许多功能。
- **服务处理器：**和OOB功能一样，即便尚未为主要的平台处理器提供软件或者该处理器还无法正常工作，专用的管理服务处理器也能够支持功能。一个分离的服务处理器还允许在任何平台系统状态下提供更有效的监控、故障检测和诊断功能。它还允许平台执行自动功能（如周期性自由检测和诊断功能）、自动硬件和软件配置发现和报告以及自我修复操作（例如将其自身从网络中隔离开来以及在删除病毒时发出警报）。

为了在组成面向服务的企业的硬件和软件组件之间提供互操作性，SOI必须基于定义公共功能、数据结构和界面的开放业界标准。

### 2.3.7.1 面向服务的基础架构（SOI）的业务价值

SOI为企业提供了一些基本优点。它为IT自动化提供了更好的基础，从而能够提高IT生产力并降低操作成本。SOI还能够从静态的“一个应用程序一个框”的方法改为动态的资源收集，根据需要为应用程序分配虚拟处理、存储和网络资源。这样就可以通过提供更好的资源利用率降低资本成本。还可以得到更高的可靠性，并减少服务宕机和延迟，因为不需干涉应用程序即可将其故障转移到可用的资源。IT可提供更一致的服务水平，因为随着应用程序工作负载的增加软件可以实时自动分配附加资源。

就长期而言，更重要的是SOI对排除“复杂性障碍”的贡献，这种障碍可以限制设计和实现新信息支持的业务流程的组织能力。SOI就位后，业务可以重新分配IT开销方面的资金，用于可创建真实的业务差异性的新的创新性系统。例如，SOI可帮助企业在数据可用时实时地完全利用自动数据源（如RFID或eForms）。在公司防火墙内部和外部，SOI增强了实现和管理事件驱动的、基于消息的服务和应用程序的能力。更好地管理移动无线设备使得业务能够将服务分布在网络边缘，使得员工能够提高生产力，并可以为移动客户提供增值服务。SOI还能提供更好的业务可靠性，例如，在精确减值计划的IT资产储存库的领域改进Sarbanes-Oxley遵从性，以及提供及时的软件版本和许可证管理。这些都是高性能功能：通过使用自动IT资产储存库技术，一个高科技公司在五个季度内可以节省40,000小时的人力资源。

从纯IT管理的角度来看，SOI提供了许多优点：

- 降低操作工作负载，通过降低硬件和软件的手动配置可达到这个目的。
- 高生产力，通过从单个标准化管理控制台执行企业级平台、网络、数据和应用程序管理可达到这个目的。
- 更好的资源利用率，降低资本支出以实现IT目标。
- 更好的灵活性，通过动态资源分配以及关注核心竞争力和跨防火墙将商品IT服务（如公司电子

邮件) 移动到第三方提供商的选项可达到这个目标。SOI跨防火墙进行操作的能力意味着IT可以一致地管理服务, 无论这些服务是来自公司内部还是通过telco或ISP连接性外包给外部提供商。

- 简单和成本有效的升级, 来自模块化的松耦合SOI架构。这允许IT刷新其基础架构以利用新技术, 而排除由硬件实现管理依赖项导致的复杂性。这就消除了令人痛苦的“删除并替换”(“forklift”)升级, 允许IT对于快速而优美的改变进行相应, 减少导致IT由于预期中断而拒绝并入新技术的惯性。
- 支持按需应变的使用模型, 允许通过“使用付费”的退款方法使得成本与利益更准确地对应。
- 支持管理所有客户机平台, 从台式机到手持移动设备, 作为具有公共平台功能和访问界面的基础架构的主要部分。

开发人员还可以从面向服务的基础架构的演变中直接获益:

- SOI从应用程序代码中删除了资源管理和安全责任。通过自上而下、自内而外、标准的一致的方式, 基础架构管理协议确保将这些功能包含在启用了Web服务的硬件、SOI中间件和应用程序构建块的架构中。开发人员不需要作出总是一成不变的假设, 也不需要应用程序中嵌入逻辑, 而只需要指定虚拟应用程序要求。然后他们就可以依靠SOI来确保动态满足这些要求。
- 开发人员可以将预先构建的服务用于分布式的、实时的、事件驱动的多平台应用程序, 因为架构中已经包含了这些元素以便使其与标准化SOI一致并运用它。由于开发人员可以方便地重用为其他服务和应用程序构建的组件, 因此他们能够更快速地构建和修改应用程序, 从而可以提供更多的新方法和业务价值。
- 开发人员可以更有效地在异构环境中工作, 包括跨企业的Internet/Web应用程序, 因为他们使用基于Web的、面向服务的和虚拟化的构建块, 意味着他们符合产业标准功能和界面

## 2.3.8 将 SOA 参考架构映射到企业 SOA 成熟度模型

下图表示采用SOA的常见模式。每个IT组织都应该使用适合其特定需要的方式引入技术组件。



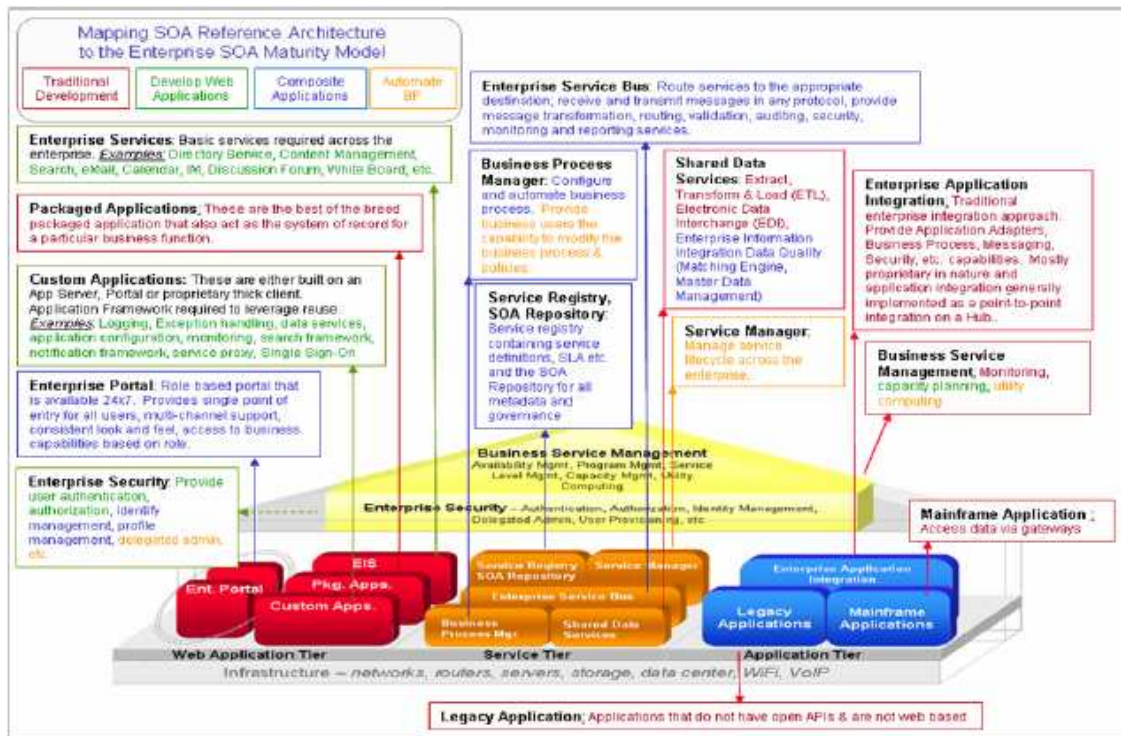


图24: 将SOA参考架构映射到企业SOA成熟度模型