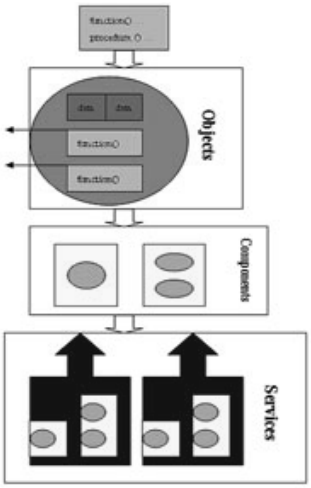
基于SOA架构设计

- 服务的设计与原则
- 面向服务架构的消息模式
- 评估基于服务的集成技术的过程和准则
- 服务模式与反模式
- Web服务的体系结构
- 改善web服务的协同工作能力

应用软件开发方法的演变过程

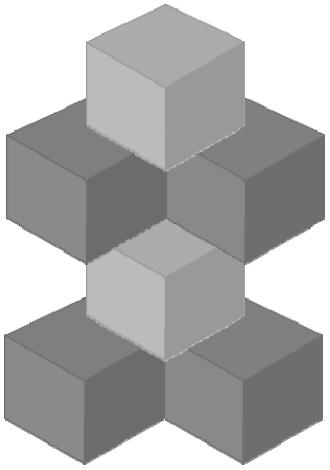
- 应用软件开发方法的演变过程：
  - 面向过程;
  - 面向对象;
  - 面向组件;
  - 面向服务。



软件开发方法的演变过程

什么是面向服务的体系架构?

- 一个组件模型
- 将应用程序的不同功能单元(*服务*)通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。
- 构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。



服务的设计与原则

service-oriented Architecture

- SOA（service-oriented Architecture，也叫面向服务的体系结构或面向服务架构）是指为了解决在Internet环境下业务集成的需要，通过连接能完成特定任务的独立功能实体实现的一种软件系统架构。
- SOA是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。

- 传统的Web（HTML/HTTP）技术有效的解决了人与信息系统的交互和沟通问题，极大的促进了B2C模式的发展。
- WEB服务(XML/SOAP/WSDL）技术则是要有效的解决信息系统之间的交互和沟通问题，促进B2B/EAI/CB2C的发展。
- SOA（面向服务的体系)则是采用面向服务的商业建模技术和WEB服务技术，实现系统之间的松耦合，实现系统之间的整合与协同。WEB服务和SOA的本质思路在于使得信息系统个体在能够沟通的基础上形成协同工作。

- 对于面向同步和异步应用的，基于请求/响应模式的分布式计算来说，SOA是一场革命。一个应用程序的业务逻辑（Business Logic）或某些单独的功能被模块化并作为服务呈现给消费者或客户端。这些服务的关键是他们的松耦合特性。
- 例如，服务的接口和实现相独立。应用开发人员或者系统集成者可以通过组合一个或多个服务来构建应用，而无须理解服务的底层实现。举例来说，一个服务可以用。NET或J2EE来实现，而使用该服务的应用程序可以在不同的平台之上，使用的语言也可以不同。

SOA具有的特性(2-1)

- SOA服务具有平台独立的自我描述XML文档。Web服务描述语言（WSDL， Web Services Description Language）是用于描述服务的标准语言。
- SOA 服务用消息进行通信，该消息通常使用XML Schema来定义（也叫做XSD， XML Schema Definition）。消费者和提供者或消费者和服务之间的通信多见于不知道提供者的环境中。服务间的通讯也可以看作企业内部处理的关键商业文档。

SOA具有的特性(2-2)

- 在一个企业内部，SOA服务通过一个扮演目录列表（Directory listing）角色的登记处（Registry）来进行维护。应用程序在登记处（Registry）寻找并调用某项服务。统一描述，定义和集成（UDDI， Universal Description， Definition，and Integration）是服务登记的标准。
- 每项SOA服务都有一个与之相关的服务品质（QOS， quality of service）。QoS的一些关键元素有安全需求（例如认证和授权），可靠通信（译注：可靠消息是指，确保消息“仅且仅仅”发送一次，从而过滤重复信息。），以及谁能调用服务的策略。

SOA三大基本特征(1)

- 独立的功能实体
  - 在Internet这样松散的使用环境中，任何访问请求都有可能出错，因此任何企图通过Internet进行控制的结构都会面临严重的稳定性问题。SOA非常强调架构中提供服务的功能实体的完全独立自主的能力。传统的组件技术，如.NET Remoting，EJB，COM或者CORBA，都需要有一个宿主(Host或者Server)来存放和管理这些功能实体；当这些宿主运行结束时这些组件的寿命也随之结束。这样当宿主本身或者其它功能部分出现问题的时候，在该宿主上运行的其它应用服务就会受到影响。
  - SOA架构中非常强调实体自我管理和恢复能力。常见的用来进行自我恢复的技术，比如事务处理(Transaction)，消息队列(Message Queue)，冗余部署(Redundant Deployment)和集群系统(Cluster)在SOA中都起到至关重要的作用。

- 大数据量低频率访问
  - 对于.NET Remoting，EJB或者XML-RPC这些传统的分布式计算模型而言，他们的服务提供都是通过函数调用的方式进行的，一个功能的完成往往需要通过客户端和服务器来回很多次函数调用才能完成。在Intranet的环境下，这些调用给系统的响应速度和稳定性带来的影响都可以忽略不计，但是在Internet环境下这些因素往往是决定整个系统是否能正常工作的一个关键决定因素。因此SOA系统推荐采用大数据量的方式一次性进行信息交换。

- 基于文本的消息传递
  - 由于Internet中大量异构系统的存在决定了SOA系统必须采用基于文本而非二进制的消息传递方式。在COM、CORBA这些传统的组件模型中，从服务器端传往客户端的是一个二进制编码的对象，在客户端通过调用这个对象的方法来完成某些功能；但是在Internet环境下，不同语言，不同平台对数据、甚至是一些基本数据类型定义不同，给不同的服务之间传递对象带来的很大困难。由于基于文本的消息本身是不包含任何处理逻辑和数据类型的，因此服务间只传递文本，对数据的处理依赖于接收端的方式可以帮忙绕过兼容性这个的大泥坑。

- 此外，对于一个服务来说，Internet与局域网最大的一个区别就是在Internet上的版本管理极其困难，传统软件采用的升级方式在这种松散的分布式环境中几乎无法进行。采用基于文本的消息传递方式，数据处理端可以只选择性的处理自己理解的那部分数据，而忽略其它的数据，从而得到的非常理想的兼容性。

向服务架构（SOA）的原则

- SOA的强大和灵活性将给企业带来巨大的好处。如果某组织将其IT架构抽象出来，将其功能以粗粒度的服务形式表示出来，每种服务都清晰地表示其业务价值，那么，这些服务的顾客（可能在公司内部，也可能是公司的某个业务伙伴）就可以得到这些服务，而不必考虑其后台实现的具体技术。更进一步，如果顾客能够发现并绑定可用的服务，那么在这些服务背后的IT系统能够提供更大的灵活性。
- 但是，要得到种强大和灵活性，需要有一种实现架构的新方法，这是一项艰巨的任务。企业架构设计师必须要变成“面向服务的架构设计师”，不仅要理解SOA，还要理解SOA的实践。在架构实践和最后得到的架构结果之间的区别非常微妙，也非常关键。本文将讨论SOA的实践，即：面向架构的设计师在构建SOA时必须要做的事情。

SOA的原则

- SOA是一种企业架构，因此，它是从企业的需求开始的。但是，SOA和其它企业架构方法的不同之处在于SOA提供的业务敏捷性。业务敏捷性是指企业对变更快速和有效地进行响应、并且利用变更来得到竞争优势的能力。
- 对架构设计师来说，创建一个业务敏捷的架构意味着创建这样一个IT架构，它可以满足当前还未知的业务需求。

- 要满足这种业务敏捷性，SOA的实践必须遵循以下原则(1)：
  - 业务驱动服务，服务驱动技术
    - 从本质上说，在抽象层次上，服务位于业务和技术中间。面向服务的架构设计师一方面必须理解在业务需求和可以提供的服务之间的动态关系，另一方面，同样要理解服务与提供这些服务的底层技术之间的关系。

- 要满足这种业务敏捷性，SOA的实践必须遵循以下原则(2)：
  - 业务敏捷是基本的业务需求
    - SOA考虑的是下一个抽象层次：提供响应变化需求的能力是新的“元需求”，而不是处理一些业务上的固定不变的需求。从硬件系统而上的整个架构都必须满足业务敏捷的需求，因为，在SOA中任何的瓶颈都会影响到整个IT环境的灵活性。

- 要满足这种业务敏捷性，SOA的实践必须遵循以下原则(3)：
  - 一个成功的SOA总在变化之中
    - SOA工作的场景，更象是一个活的生物体，而不是象传统所说的“盖一栋房子”。IT环境唯一不变的就是变化，因此面向服务架构设计师的工作永远不会结束。对于习惯于盖房子的设计师来说，要转向设计一个活的生物体要求崭新的思维方式。如下文所写的，SOA的基础还是一些类似的架构准则。

SOA基础

- 在IT行业有两个越来越普遍的发展方向，一个是架构方面的，一个是方法学方面的，面向服务的架构设计师可以从中有所收获。第一个就是MDA（模型驱动架构），由提出CORBA的OMG模型提出。MDA认为架构设计师首先要对待创建的系统有一个形式化的UML（也是由OMG提出）的模型。MDA首先给出一个平台无关的模型来表示系统的功能需求和Use Cases，根据系统搭建的平台，架构设计师可以由这个平台无关的模型得到平台相关的模型，这些平台相关模型足够详细，以至于可以用来直接生成需要的代码。

SOA基础

- MDA的核心就在于在设计阶段系统就已经完全描述，这样，在创建系统的时候，几乎就没有错误解释的可能，模型也就可以直接生成代码。但MDA有一些局限性：首先，MDA假设在创建模型之前，业务需求已经全部描述，而这一点，在当前典型的动态业务环境中几乎是不可能的。第二，MDA没有一个反馈机制。如果开发人员对模型有需要改动的地方，并没有提供给他们这么一个途径。

SOA基础

- SOA的另一个基础是敏捷方法（AM），其中非常有名的方法是极限编程（XP）。象XP这样的AM提供了在需求未知或者多变的环境中创建软件系统的过程。XP要求在开发团队中要有一个用户代表，他帮助书写测试来指导开发人员的日常工作。开发团队中的所有成员都参与到设计之中，并且设计要尽量小并且非形式化。AM的目标是仅仅创建用户想要的，而不是在一些形式化模型上耗费工作量。AM的核心思想就在于其敏捷性－处理需求变更的敏捷性。AM的主要弱点是其规模上的限制，例如，XP在一个小团队和中型项目中效果不错，但是当项目规模增大时，如果没有一个一致的清晰的计划，项目成员很难把握项目中的方方面面。

SOA基础

- 从表面看来，MDA和AM似乎是相对立的－MDA假定需求是固定的，而AM恰恰相反。MDA的中心是形式化的模型，而AM恰恰要避开它们。但是，我们还是决定冒险把这些不同方法中的一些元素提取出来，放入到一个一致的架构实践中。
- 在SOA中有三个抽象层次，按照SOA的第一条准则：业务驱动服务、服务驱动技术。AM将业务模型直接和实践连接起来，表现在平台相关的模型之中。MDA并没有把业务模型和平台无关模型分开来，而是把平台无关模型做为起点。SOA必须连接这些模型，或者说抽象层次，得到单一的架构方法。我们将从五个视图的架构实现方法来实现这个连接。

SOA的五视图实现方法

- 企业架构设计师发现他们的职业非常有竞争力并且值得骄傲，因为他们要从很多方面来通盘考虑IT系统。Kruchten（RUP的开发负责人）将这些方面提取出来，在应用到SOA时，我们称为五视图实现方法（five-view approach）。

SOA的五视图实现方法

- 四个方框表示对一个架构的不同审视方法，分别代表不同的涉众（stakeholder）。第五个视图，use-cASe视图涵盖了其它视图，在架构中扮演的是一个特殊的角色。部署视图将软件映射到底层平台和相关硬件上，是系统部署人员对架构的视图；实现视图描述了软件代码的组织，是从开发人员角度出发的视图；业务分析人员则利用过程视图进行工作，它描述的是软件系统的运行时特性。最后，逻辑视图表示的是用户的功能需求。在SOA中，面向服务的架构必须能够以use-case视图中的用例将用户连接到服务，将服务连接到底层的技术。

SOA的五视图实现方法

- 为了表示面向对象的架构是如何工作在这些视图之上，让我们将他们置于SOA元模型的上下文之中。SOA中两个领域存在重叠：由业务模型和服务模型表示的业务领域和由服务模型及平台相关模型表示的技术领域（两个领域共享服务模型）。业务用户通过逻辑视图和过程视图处理粗粒度的业务服务，根据变化的业务需求，按照需要将它们安排在过程之中。另一方面，技术专家的工作是创建并维护服务和地层技术之间的抽象层。表示这些服务的中间模型，起到的是轴心的作用，业务以它为中心进行。

SOA的五视图实现方法

- SOA元模型从MDA中继承平台无关模型和平台相关模型，但是添加了AM和用户交互以及敏捷的反馈这两部分，后者通过椭圆之间的双向箭头来表现。类似地，元模型通过引入由中心的服务模型提供的中间层抽象解决了AM在伸缩性方面的问题。这样，服务模型中的任何需求的变化，都会反映到用户每天的业务处理中。同样，由于底层技术是模型驱动的，技术专家也可以根据这些变化的需求迅速而有效地作出应变。

SOA的五视图实现方法

- SOA实践和过去解决企业架构传统方式的不同之处就在于其对敏捷性的支持。如前所说，SOA的第三条原则就在于它总在变化之中。这种恒在的变化性环境是SOA实践的基石。如图所示，涉众（stakeholders，译者注：RUP中也有这个词，表示软件开发中涉及到的各种角色如：用户、设计人员、开发人员乃至测试人员等等。）在一个必需的基础上影响到整个架构的变化。在当技术专家在每天的日常工作中不断对变化的业务需求作出响应的这种情况下，设计阶段和运行阶段之间的界限变得模糊起来，很难清晰地分离这两个阶段。

剩下的部分

- 我们已经为面向服务的架构提供了一个高层次的框架，其中MDA和AM的元素帮助工具的使用者来创建和维护SOA。但是，SOA中还缺少一些内容－那就是软件开发商和专业的服务组织必需提供的。理想情况下，开发商必需提供面向服务的业务流程、工作流以及服务的协调工具和服务；另外，能够以一种敏捷的、平台无关的方式充分反映业务服务的建模工具也是必须的；技术专家必须配备可以从模型中自动生成代码，并在代码变化时更新模型的工具，最后，开发商必须提供支持SOA的软件，帮助面向服务的架构设计师以一种可信并且可伸缩的方式创建位于服务和底层技术之间的抽象层次。幸运的是，这方面的产品即将上市。

- 另外，最重要的就是贯穿本文的自顶而下的SOA实现方法了。今天关于Web services的大部分思考都是自底而上的："这是如何创建Web services的方法，现在，我们来使用它们集成吧"，对Web services技术的这种方法是伟大的第一步，因为它可以惊人地降低集成的开销，这是现在的技术管理人员最乐意见到的了。但当经济进一步发展，IT走出低谷，企业会寻求IT的帮助来提高组织战略意义上的核心价值。使用面向服务的架构，IT可以提供给企业实现业务敏捷性的这样一个框架。

为什么选择面向服务架构（SOA）？

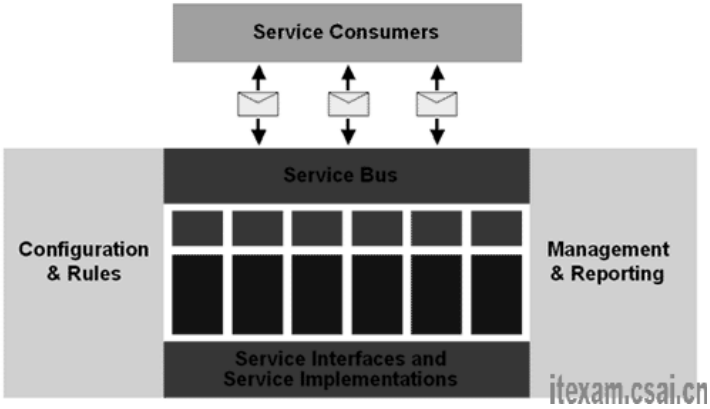
- 不同种类的操作系统，应用软件，系统软件和应用基础结构（application infrastructure）相互交织，这便是IT企业的现状。一些现存的应用程序被用来处理当前的业务流程（business processes），因此从头建立一个新的基础环境是不可能的。
- 企业应该能对业务的变化做出快速的反应，利用对现有的应用程序和应用基础结构（application infrastructure）的投资来解决新的业务需求，为客户，商业伙伴以及供应商提供新的互动渠道，并呈现一个可以支持有机业务（orGAnic business）的构架。
- SOA凭借其松耦合的特性，使得企业可以按照模块化的方式来添加新服务或更新现有服务，以解决新的业务需要，提供选择从而可以通过不同的渠道提供服务，并可以把企业现有的或已有的应用作为服务，从而保护了现有的IT基础建设投资。

- 如图1的例子所示，一个使用SOA的企业，可以使用一组现有的应用来创建一个供应链复合应用（supply chain composite application），这些现有的应用通过标准接口来提供功能。

itexam.csai.cn

服务架构

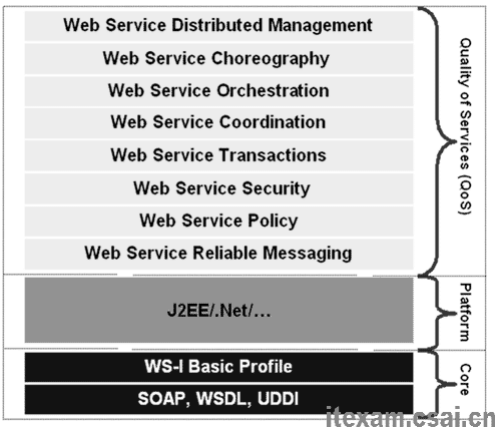
- 为了实现SOA，企业需要一个服务架构，图2显示了一个例子：



- 在图2中， 服务消费者（service consumer）可以通过发送消息来调用服务。这些消息由一个服务总线（service bus）转换后发送给适当的服务实现。这种服务架构可以提供一个业务规则引擎（business rules engine），该引擎容许业务规则被合并在一个服务里或多个服务里。这种架构也提供了一个服务管理基础（service management infrastructure），用来管理服务，类似审核，列表（Billing），日志等功能。此外，该架构给企业提供了灵活的业务流程，更好地处理控制请求（regulatory requirement），例如Sarbanes Oxley（SOX），并且可以在不影响其他服务的情况下更改某项服务。

面向服务架构(SOA)基础结构

- 要运行管理SOA应用程序，企业需要SOA基础，这是SOA平台的一个部分。SOA基础必须支持所有的相关标准，和需要的运行时容器。图3所示的是一个典型的SOA基础结构。接下来的章节将逐一讨论该结构的每个部分。



itexam.csai.cn

SOAP， WSDL， UDDI

- WSDL，UDDI和SOAP是SOA基础的基础部件。WSDL用来描述服务；UDDI用来注册和查找服务；而SOAP，作为传输层，用来在消费者和服务提供者之间传送消息。SOAP是Web服务的默认机制，其他的技术为可以服务实现其他类型的绑定。一个消费者可以在UDDI注册表（registry）查找服务，取得服务的WSDL描述，然后通过SOAP来调用服务。

WS-I Basic Profile

- WS-I Basic Profile，由Web服务互用性组织（Web Services Interoperability Organization）提供，是SOA服务测试与互用性所需要的核心构件。服务提供者可以使用Basic Profile测试程序来测试服务在不同平台和技术上的互用性。

J2EE 和 .Net

- 尽管J2EE和。NET平台是开发SOA应用程序常用的平台，但SOA不仅限于此。像J2EE这类平台，不仅为开发者自然而然地参与到SOA中来提供了一个平台，还通过他们内在的特性，将可扩展性，可靠性，可用性以及性能引入了SOA世界。新的规范，例如 JAXB（Java API for XML Binding），用于将XML文档定位到Java类；JAXR（Java API for XML Registry）用来规范对UDDI注册表（registry）的操作；XML-RPC（Java API for XML-based Remote Procedure Call）在J2EE1.4中用来调用远程服务，这使得开发和部署可移植于标准J2EE容器的Web服务变得容易，与此同时，实现了跨平台（如。NET）的服务互用。

服务品质

- 在企业中，关键任务系统（Mission-critical system，译注：关键任务系统是指如果一个系统的可靠性对于一个组织是至关重要的，那么该系统就是该企业的关键任务系统。比如，电话系统对于一个电话促销企业来说就是关键任务系统，而文字处理系统就不那么关键了。）用来解决高级需求，例如安全性，可靠性，事物。当一个企业开始采用服务架构作为工具来进行开发和部署应用的时候，基本的Web服务规范，像WSDL，SOAP，以及UDDI就不能满足这些高级需求。正如前面所提到的，这些需求也称作服务品质（QoS，quality of services）。

- 与QoS相关的众多规范已经由一些标准化组织（standards Bodies）提出，像W3C（World WIDE Web Consortium）和OASIS（the Organization for the Advancement of Structured Information Standards）。下面的部分将会讨论一些QoS服务和相关标准。

- 安全
  - Web服务安全规范用来保证消息的安全性。该规范主要包括认证交换， 消息完整性和消息保密。该规范吸引人的地方在于它借助现有的安全标准，例如，SAML（as Security Assertion Markup Language）来实现web服务消息的安全。OASIS正致力于Web服务安全规范的制定。

- 可靠
  - 在典型的SOA 环境中，服务消费者和服务提供者之间会有几种不同的文档在进行交换。
  - 具有诸如“仅且仅仅传送一次”（ once-and-only-once delivery），“最多传送一次”（ at-most-once delivery），“重复消息过滤”（duplicate message elimination），“保证消息传送”（guaranteed message delivery）等特性消息的发送和确认，在关键任务系统（mission-critical systems）中变得十分重要。
  - WS-Reliability 和 WS-ReliableMessaging是两个用来解决此类问题的标准。这些标准现在都由OASIS负责。

- **策略**
  - 服务提供者有时候会要求服务消费者与某种策略通信。比如，服务提供商可能会要求消费者提供Kerberos安全标示，才能取得某项服务。这些要求被定义为策略断言（policy assertions）。一项策略可能会包含多个断言。WS-Policy用来标准化服务消费者和服务提供者之间的策略通信。

- **控制**
  - 当企业着手于服务架构时，服务可以用来整合数据仓库（silos of data），应用程序，以及组件。整合应用意味着例如异步通信，并行处理，数据转换，以及校正等进程请求必须被标准化。在SOA中，进程是使用一组离散的服务创建的。
  - BPEL4WS 或者 WSBPEL（Web Service Business Process Execution Language）是用来控制这些服务的语言。WSBPEL目前也由OASIS负责。

- 管理
  - 随着企业服务的增长，所使用的服务和业务进程的数量也随之增加，一个用来让系统管理员管理所有运行在多相环境下的服务的管理系统就显得尤为重要。
  - WSDM（Web Services for Distributed Management）规定了任何根据WSDM实现的服务都可以由一个WSDM适应（WSDM-compliant）的管理方案来管理。

- 其它的qos特性，比如合作方之間的溝通和通訊，多個服務之間的事務處理，都在WS-Coordination 和 WS-Transaction 標準中描述， 這些都是OASIS 的工作。

SOA 不是Web服务

- 在理解SOA和Web服务的关系上，经常发生混淆。根据2003年4月的Gartner报道，Yefim V. Natis就这个问题是这样解释的：“Web服务是技术规范，而SOA是设计原则。特别是Web服务中的WSDL，是一个SOA配套的接口定义标准：这是Web服务和SOA的根本联系。
- ”从本质上来说，SOA是一种架构模式，而Web服务是利用一组标准实现的服务。Web服务是实现SOA的方式之一。用Web服务来实现SOA的好处是你可以实现一个中立平台，来获得服务，而且随着越来越多的软件商支持越来越多的Web服务规范，你会取得更好的通用性。

面向服务架构（SOA）的优势

- SOA的概念并非什么新东西，SOA不同于现有的分布式技术之处在于大多数软件商接受它并有可以实现SOA的平台或应用程序。
- SOA伴随着无处不在的标准，为企业的现有资产或投资带来了更好的重用性。
- SOA能够在最新的和现有的应用之上创建应用；SOA能够使客户或服务消费者免予服务实现的改变所带来的影响；SOA能够升级单个服务或服务消费者而无需重写整个应用，也无需保留已经不再适用于新需求的现有系统。
- 总而言之，SOA以借助现有的应用来组合产生新服务的敏捷方式，提供给企业更好的灵活性来构建应用程序和业务流程。

采用服务驱动型方法的企业体验着以下业务和 IT 好处(1)

- 面向服务架构的业务好处
  - 效率： 将业务流程从“烟囱”状的、重复的流程向维护成本较低的高度利用、共享服务应用转变。
  - 响应： 迅速适应和传送关键业务服务来满足市场需求，为客户、雇员和合作伙伴更高水准的服务。
  - 适应性： 更高效地转入转出让整个业务变得复杂性和难度更小，达到节约时间和资金的目的。

采用服务驱动型方法的企业体验着以下业务和 IT 好处(2)

- 面向服务架构的 IT 好处
  - 复杂性降低： 基于标准的兼容性，与点到点的集成相比降低了复杂性。
  - 重用增加： 通过重用以前开发和部署的共享服务，实现了更有效的应用程序 / 项目开发和交付。
  - 遗留集成： 用作可重用服务的遗留应用程序降低了维护和集成的成本。
- 如今的服务驱动型企业都在体验着开发的高效率，服务的高可靠性和服务的高质量，以最大限度获得业务机会所带来的这些好处。

SOA在国际市场上反响强烈

- 自2004年初业界推出SOA后，Bea、IBM、ORACLE、微软等业界巨头纷纷发布自己的SOA战略，建议用户在进行企业IT建设时考虑SOA。
- ZapThink调研公司在最近发表的一份报告中预测，到2006年，基于SOA架构(面向服务的架构)的中间件产品将成为网络化商业系统的主要设计思路，其中70％的商业企业公司将使用SOA架构。
- 按照Gartner的预测，到2008年，SOA将成为占有绝对优势的软件工程实践方法，它将结束传统的整体软件体系架构长达40年的统治地位。届时，将有60％的商业公司在进行商业IT建设时会转向SOA。
- IDC预测到 2007年，包括软件、服务和硬件在内的SOA市场将达到210亿美元，其中商业企业方面的市场将达到120亿美元。
- 综上所述SOA已经成为大势所趋，有着广阔的市场空间和巨大的发展潜力；而在商业企业中的应用，将成为SOA未来发展的一大亮点。

SOA已经引起国内商业企业的重视

- 国内基于SOA架构Web服务目前还是集中在一些企业内部，而国内一些有影响的行业用户正在搭建其核心业务系统，比如商业领域的流通行业和销售行业的大集中正在起步。因此当商业企业需要更好地服务客户，需要更好地与上、下游合作伙伴协同工作，并且自己内部的核心业务之间也需要协同工作时，基于SOA架构中间件产品就会为这类新的业务应用提供理想的底座，这种新的应用被称作面向服务的业务应用。
- 现在，很多商业企业都准备在2006年内开始规划使用这些基于SOA架构的应用，可想而知，这些SOA架构的中间件产品将在两年内迅速发展，并在五年内在整个IT行业内获得广泛应用。

商业企业信息化存在的问题

- 商业企业信息系统多数处于封闭运行的状态，企业之间、企业与上游供应商、下游消费者之间信息不对称。商业企业之间无法形成协同效应。信息系统即无法满足消费者的综合需求也无法达到企业间的商务协同自动化和智能化的需求。信息化的经济效益难以有效发挥。同时信息化标准不健全，如电子交换接口标准、业务流程协同标准；流通中的票证、单据格式标准；电子数据交换所必须的结构化数据标准等。

商业企业信息化存在的问题

- 采用传统的系统架构技术和传统的**EAI**和**B2Bi**技术则存在系统封闭、厂商依赖性强、耦合度高、重用性差，扩展性差、无法和上下游企业的系统建立统一的接口等问题。而采用**SOA** 技术则可以有效解决上述问题，由于**SOA**基于**HTTP/SOAP/WSDL**等开放式技术，对于特定厂商产品依赖性小；系统开放、互操作性强，可以建立统一的**WEB**服务用于和不同的上下游企业信息系统实现供应链协同。由于**SOA**的松耦合特性、比较符合集团和各下属机构的商业关系，业务流程整合和项目协调的阻力会有效降低。

商业企业信息化存在的问题

- SOA以服务为基本单元，更加贴近于企业的商业活动，业务梳理和建模的复杂度会有效降低，重用性也会有效提高。另外采用SOA，企业IT系统所提供的服务会更容易扩展、组合和变更，符合该集团目前业务发展变化较快的特点，可以有效的降低该集团IT系统的长期拥有总体成本。我们将该集团公司作为一个试点，推进SOA技术的运用，来有效解决上述问题。

“协同商务”的新经济时代即将到来

- 采用SOA技术最终将使得各个商业企业之间、各个关联的经济实体之间实现高效实时的联接，使得整个产业链实现自动化的协同商务，将会有力的提高商业企业的应变能力，转变现有的商业运作模式，转变经济增长的方式。SOA技术将促进信息系统在商业企业贸易活动中的全面渗入和发展，对于简单的贸易活动，将会由信息系统自动化实现；对于复杂的贸易活动，信息系统将会为企业管理人员提供足够的决策信息并可以高效的执行决策。SOA技术的应用将会全面提高商务的自动化、智能化和实时化水平。

“协同商务”的新经济时代即将到来

- 采用**SOA**技术实现协同商务可以提高城市范围内商流、物流、资金流和信息流的运行效率，扩大北京市商业企业整体规模效益，加强商业企业的整体对外竞争力，拉动经济增长，降低企业运营成本，推动城市流通信息技术创新体系的建立，提高北京市流通现代化水平，促进城市管理现代化和城市社会经济信息化的进程。

“协同商务”的新经济时代即将到来

- 采用**SOA**技术可以将将物流企业、物业企业、商业企业、消费者整体整合在一起，对供应链关联企业、物流企业以及网上支付体系、安全认证体系等环境建设具有明显的带动作用，有利于促进支撑环境协同发展。

促进商业企业信息化标准的制定，完善政府职能

- 采用SOA技术为信息系统的沟通提供了技术基础，而随着SOA在商业企业的应用，必将促进统一的商业领域电子商务行业标准的发展和制定，对促进国家商业企业信息标准体系的建立和完善具有重要支撑作用。
- SOA技术为政府对商业经济的运行状况提供了实时监测和指导的技术可能性，将从根本上改变政府对社会经济的管理方式。

促进商业企业信息化标准的制定，完善政府职能

- 基于SOA的协同商务带来的最直接的好处就是由于贸易范围的空前扩大而产生的全球贸易活动的大幅度增加，因而提高了贸易环节中大多数角色的交易量，因此，全球范围的经济形势将向一个良好的增长趋势发展。它还可以扩大地方商业企业整体规模效益，加强商业企业的业务整合和商业协同效应，提高商业企业的整体对外竞争力，通过协同商务有效降低企业运营成本，推动城市流通信息技术创新体系的建立，提高地方的流通现代化水平，促进城市管理现代化和城市社会经济信息化的进程。

促进商业企业信息化标准的制定，完善政府职能

- SOA在商业企业的应用可以将物流企业、物业企业、商业企业、消费者整体整合在一起，对供应链关联企业、物流企业以及网上支付体系、安全认证体系等环境建设具有明显的带动作用，可推动信息化各环节的全面应用与发展，有利于促进产业链和支撑环境协同发展，从而也创造了更多的就业机会和社会财富。

促进商业企业信息化标准的制定，完善政府职能

- 信息产业是知识经济的核心和主要的推动力，而企业信息化又是目前信息产业中最具前途的发展趋势，因此说企业信息化的发展，必将直接或间接地推动知识经济的浪潮。这种知识经济有着大量的无形成本和高附加值，在东南亚金融危机的同时，高科技给美国带来的是"高增长速度、高就业率、低通货膨胀率"。这也是我国在宣传知识经济的热潮中应注意的一个真正有价值的切入点。
- SOA技术由于其前所未有的信息系统整合与自动协同能力，成为继互联网以来又一个革命性的技术，将会把目前基于WEB/互联网的知识经济推进到一个前所未有的新阶段。

SOA 企业考虑事项

- 服务驱动型企业在对客户、合作伙伴和雇员的高效化服务方面得到了优化 – 并加速了业务服务响应时间。然而，成为服务驱动型企业，需要的不仅仅是产品的部署。对实现服务驱动型架构感兴趣的企业将希望能与一个有经验的 **SOA** 提供商合作，它提供的服务可以保护企业在业务和 **IT** 方面的投入，他们考虑到了以下几个方面：

- **业务战略**： 组织需要明确驱动关键业务流程的业务战略，它将用于成形 **SOA** 的框架。一旦识别出业务问题，就可以用一种一致的、可复用的方法对其进行定义，并实现解决方案。在这个关键的基础阶段，业务通常需要与一个拥有开发 **SOA** 业务战略经验、并能共享横向和纵向市场最佳实践的提供商进行合作。

- **体系结构**： 为了解决方案快速和动态的交付，企业必须开发一种允许装配组件和服务的体系结构框架。通过与有经验的 SOA 提供商合作，企业可以获得相应的参考案例，以快速搭建一个关注复用、避免 " 烟囱 " （ stovepipe ）式应用程序和 IT 资源 " 孤岛 " 的体系结构。此外，有经验的 SOA 提供商还可以帮助企业对项目的易管理性进行设计。

- **构建模块**： 不管是对体系结构还是对编程模型来说， SOA 都是是思考构建软件模型的一种优秀方式。与 SOA 提供商进行合作能让组织能够识别可在 SOA 实现中使用或重用的构建模块代码、服务、应用程序和组件。与有经验的 SOA 提供商进行合作还有一个好处，企业可以获得对构造组件、企业域（ domains ）、服务和规范数据模型的参考经验。

- **项目和应用程序：** SOA 创造了一种在更强大、更灵活的编程模式中搭建应用程序的新方法。与 SOA 提供商合作的企业可以更好地识别将被合并到 SOA 结构体系中的现存的和正在使用的应用程序。有经验的 SOA 提供商还将引导项目基础架构的搭建，并对正在进行中的项目提供有效的管理。

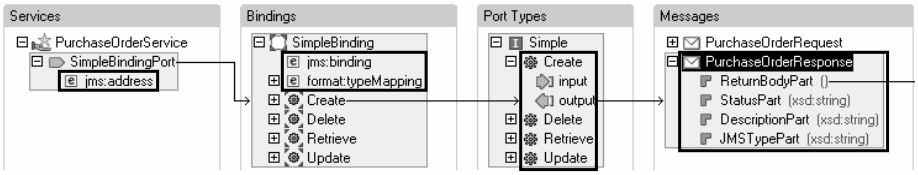
- 成本和收益： 在一个 SOA 项目中，开发和维护成本将大大削减，。有经验的 SOA 提供商可以帮助企业构造 SOA 基金模式，并构建 " 行动案例 "，包括评估基础构造成本和效益、实现项目的最佳投资回报（ ROI ）以及开发商务案例。

- **组织和统辖：** 组织需要为新的面向服务的 IT 组织识别角色和职责，并优化经验集便于以后使用。有经验的 SOA 提供商可以帮助企业实现这些目标，同时组织一个有效的设计 " 复用工厂 "（ Reuse Factory ），帮助定义统辖模式，并最终保证客户满意。

服务描述

- Services are described using a standardized interface (metadata)
  - Defines the service, its operations, and input and output parameters
  - Defines how the service is reached and its location
- Input and outputs may be simple parameters or XML documents
- An existing application can be given a service 'façade'
  - The façade (e.g. an EJB) is then described as the service
- Web Services Description Language (WSDL) is the standard used to describe the service

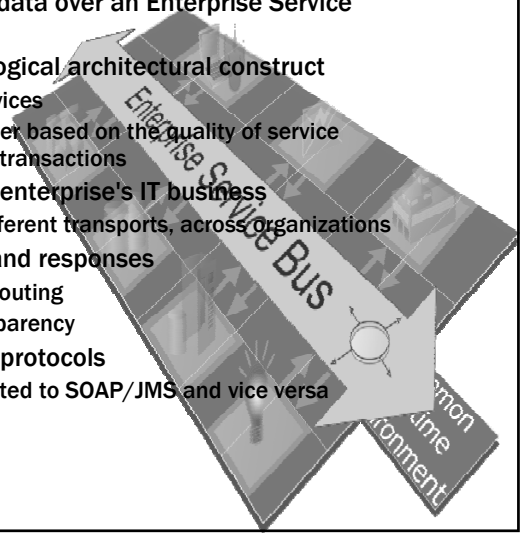
服务描述: WSDL



- WSDL (Web Services Description Language) is an XML document that describes a Service using a number of key elements:
  - A Port Type defines **what** the service does, and is described by a number of operations. For each operation the data that it receives and sends is described by a Message
  - The Binding defines **how** the service (as described by the Port Type) is invoked e.g. SOAP/HTTP, SOAP/JMS.
  - The Port specifies the address **where** the service is located

服务调用

- Services interact by exchanging data over an Enterprise Service Bus
- The Enterprise Service Bus is a logical architectural construct
  - Provides inter-connectivity services
  - Services interact with each other based on the quality of service requirements of the individual transactions
- ESB connects and integrates an enterprise's IT business
  - In different locations, using different transports, across organizations
- ESB mediates service requests and responses
  - Performs transformation and routing
  - Enables connection type transparency
- ESB enables the use of multiple protocols
  - e.g. SOAP/HTTP can be converted to SOAP/JMS and vice versa



企业服务总线(ESB)

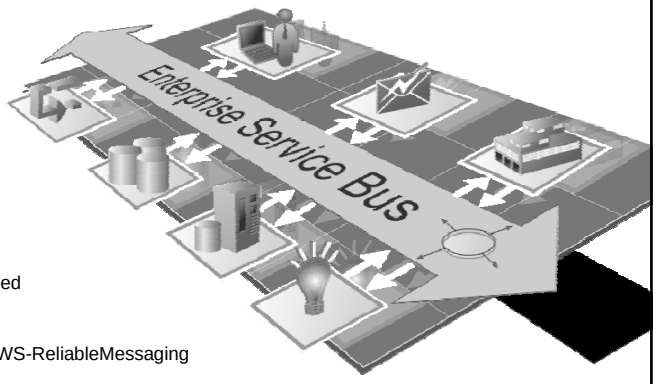
Central to the Workings of a Service Oriented Architecture

Mediation Services
Routing
Transformation

Event Services
Publish and Subscribe

Transport Services
Synchronous/Asynchronous
Persistent/Non-persistent
Loosely-coupled/Tightly-coupled

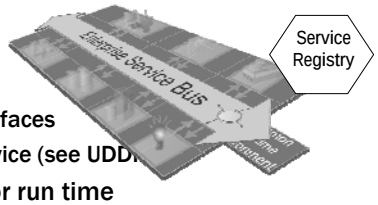
Standards Based
HTTP/HTTPS with option for WS-ReliableMessaging
JMS, JAX-RPC, SOAP
WS-Security, WS-Policy, WS-Addressing



服务的编排术

- Business processes are a set of activities carried out in a sequence
- Services can be choreographed to implement a business process
  - i.e. each step or activity in a business process is implemented by a service
- The business process itself becomes a service
- Service choreography is described by the Business Process Execution Language (BPEL)
  - A standard proposed to OASIS for process definition
- Service Choreography provides:
  - Processes that combine applications and people
  - Transactionality and compensation
  - Manipulation of process data

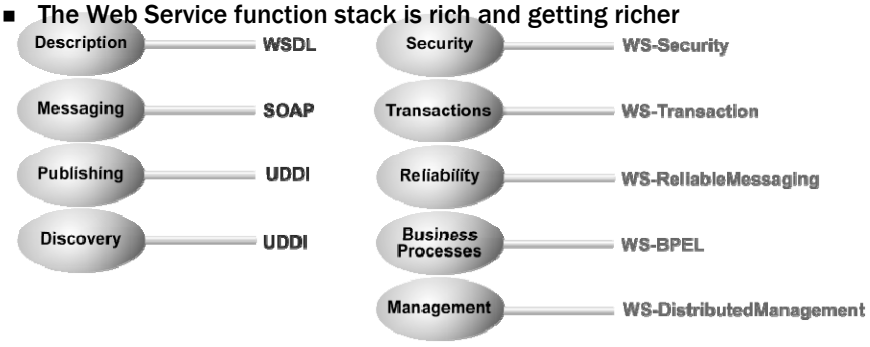
服务的发现



- A Service Registry or Directory
  - Publishes service descriptions
  - Finds business, services and service interfaces
  - UDDI is a standard for a registry web service (see UDDI
- Service discovery may be at design time or run time
  - WebSphere Studio provides Web Services Explorer to publish and search UDDI
  - UDDI Service entries contain URL of its WSDL
- A UDDI directory may be private to an enterprise
- WSDL for services may be stored in local files and directories
- ESB mediations may be used to determine service's location at run time

SOA 的服务栈

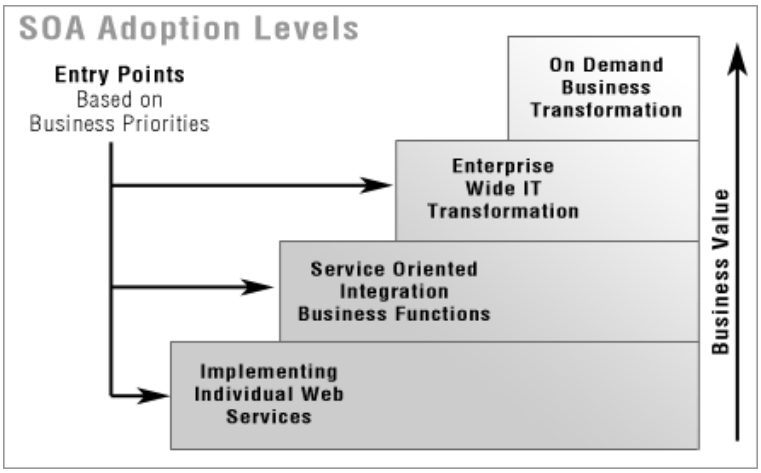
- SOA uses Web Services specifications to implement its features
- Many Web Services specifications are being standardized
- IBM and other vendors have implemented these specifications, anticipating the standards
- The Web Service function stack is rich and getting richer



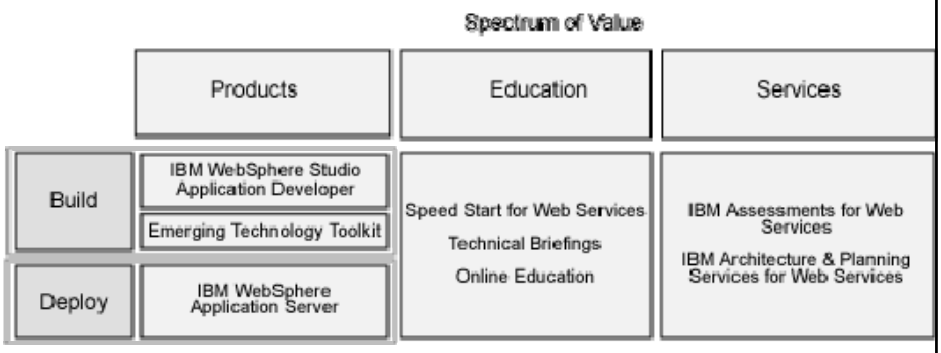
如何构建SOA的系统

- Develop a service oriented mentality
  - Everything is a service
- Understand and define services
  - Good service decomposition is the challenge
  - Remember coarse granularity
  - Some service implementations will exist, others will need to be created
- Determine service interfaces and interaction patterns
  - What formats will be adopted for data exchange?
  - What qualities of service will bound interactions?
  - Where will mediation be required/appropriate?
- Define service orchestrations in terms of supported business processes
  - Composite services will generally reflect business processes?
  - What are the business processes?
  - Business models in WebSphere BI Modeler can prove invaluable here

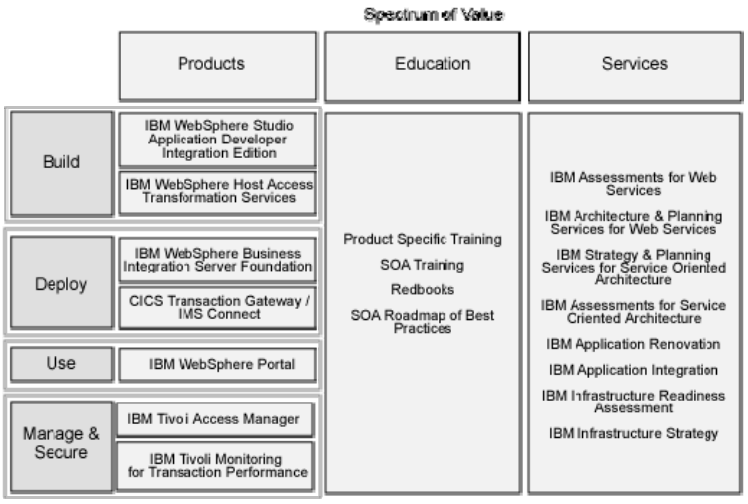
采用 SOA 的四个层次



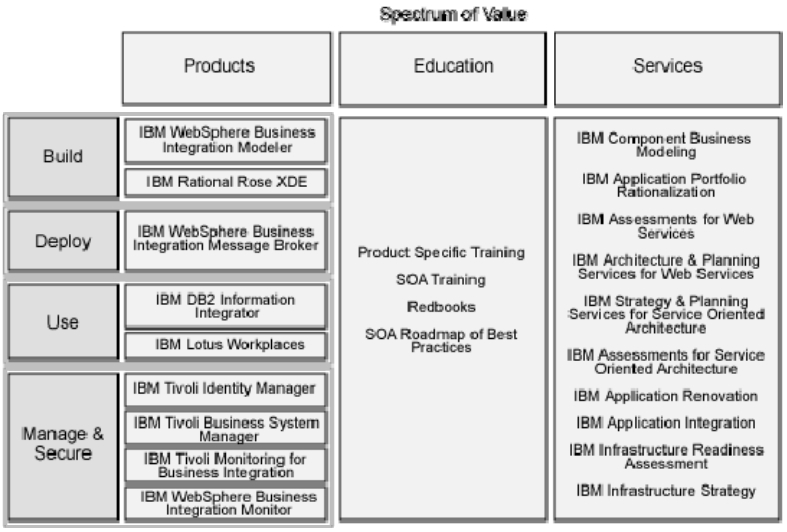
实现单独的 Web 服务



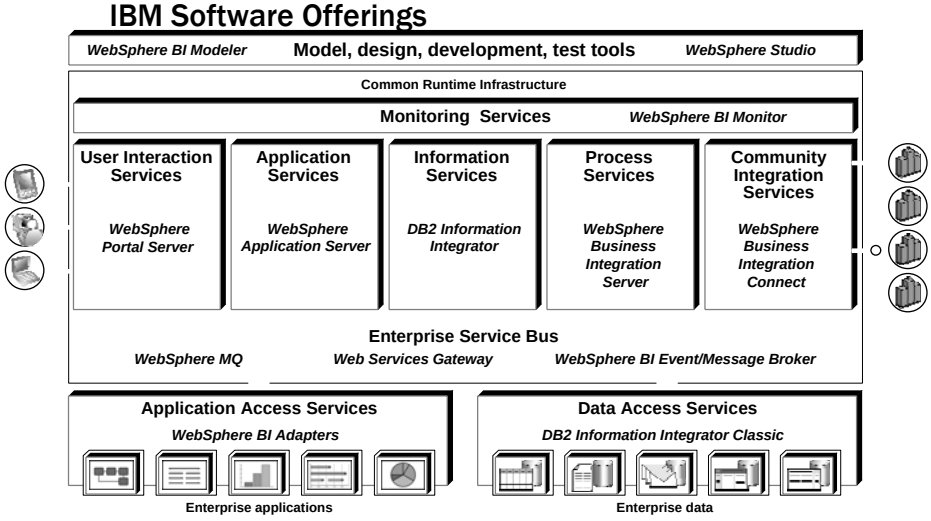
面向服务的集成



企业范围内的 IT 转换

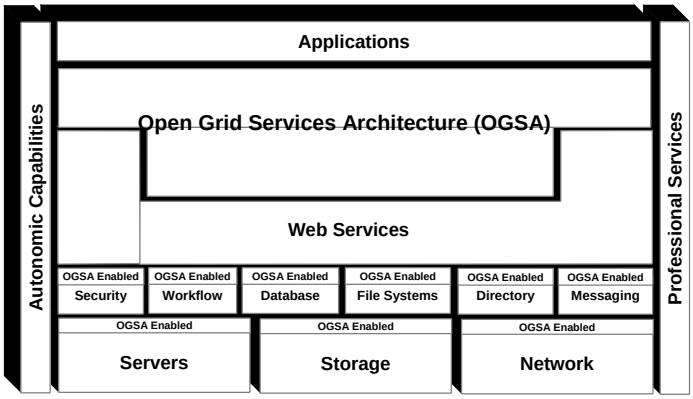


IBM商务整合的参考架构

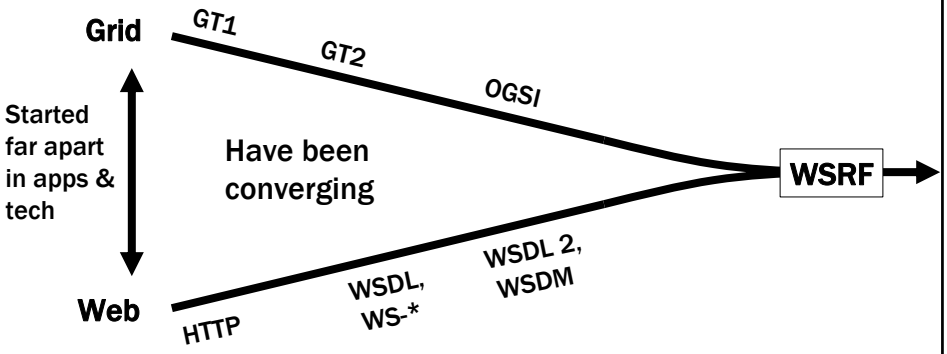


OGSA

OGSA Structure



Grid和Web服务

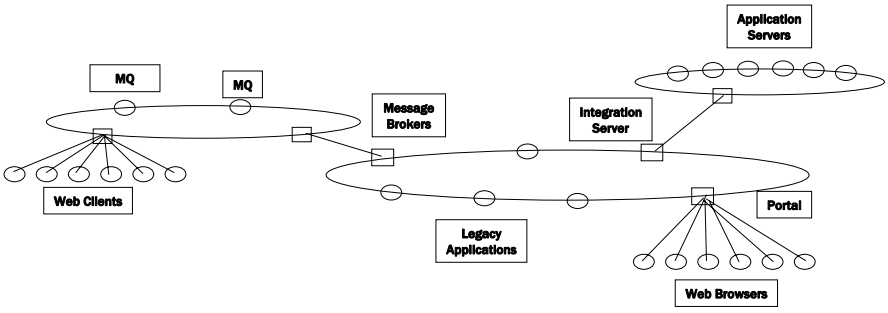


The definition of WSRF means that Grid and Web communities can move forward on a common base

网格计算与SOA

- 网格系统本身就是SOA
- 在网格服务上构建应用程序级 SOA

ESB实现服务网格: 例证



- The ESB within the enterprise will evolve based on QOS requirements required by service interactions
- This will be augmented by mediations running in Message Broker and/or the Web Services Gateway.

总结

- Service Oriented Architecture provides the flexibility required to operate in an On-Demand Environment