
动态电子商务技术

赵卫东 博士 副教授
wdzhao@fudan.edu.cn
复旦大学

动态电子商务技术

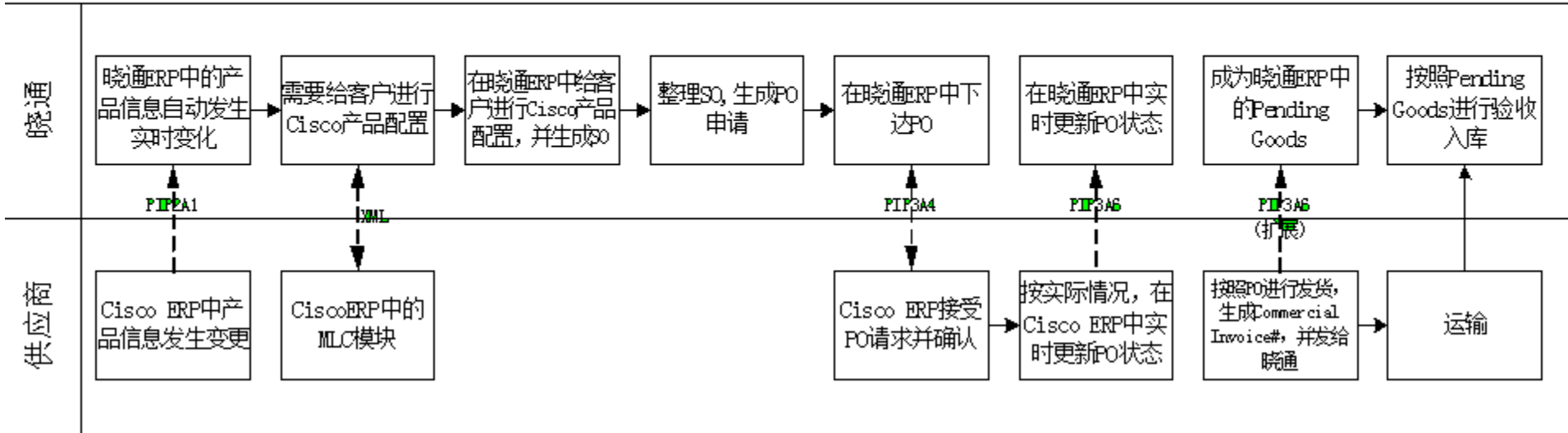
- 动态电子商务的内涵
- Web Services简介

什么是动态电子商务？

- IBM曾为动态电子商务下了一个简单定义是：“着重于 B2B 的综合性和基础设施组成上的下一代电子商务，通过调节因特网标准和通用基础设施为内部和外部企业计算创造最佳效益。”
- 动态电子商务预想了这样一个因特网，商业实体能在其自身的域内以及在贸易合伙人之间有步骤地管理交互作用。从发现新的合伙人到与另一商务实体的集成，动态电子商务着重程序对程序的交互作用。
- 动态电子商务实现的关键是Web Services。Web Services就是原来的组件技术思想在Internet时代中的进一步发展，它集成了CORBA、COM/DCOM等各种组件模型技术，是原有各种组件技术的继承和发展。它不仅已作为动态电子商务的基础，也是“按需应变的电子商务”技术发展的基础。

传统的B2B中间件系统集成（1）

B2B涉及流程



➤ 与B2B有关的活动用虚线表示，ERP内部的流程用实线表示，箭头表示信息传递的方向。

传统的B2B中间件系统集成（2）

- 通过对接，可以实现订单状态更新（及时地查询到订单的状态）、产品信息更新（根据Cisco的实际情况自动在ERP中更新其产品编号和价格的信息）、下达订单、MLC集成（在晓通的ERP中对Cisco产品进行配置）和发票预录入（商业发票的实时传达）。

按需应变电子商务的背景

- ▶ IBM提出了对电子商务下一步发展方向的展望：
按需应变的电子商务。
- ▶ 商务模式将转型为跨越供应商、分销商、客户与雇员，在人力、流程、信息上进行完全的集成。
- ▶ 向这个目标转型的商务公司需要将人力、流程、信息、包括供应链和分销商，客户和员工有机地集成起来，形成更强的竞争力。

按需应变企业的条件

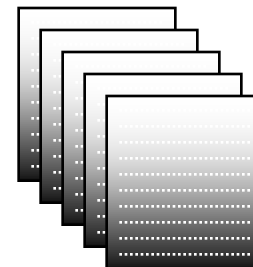
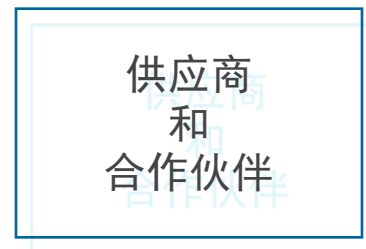
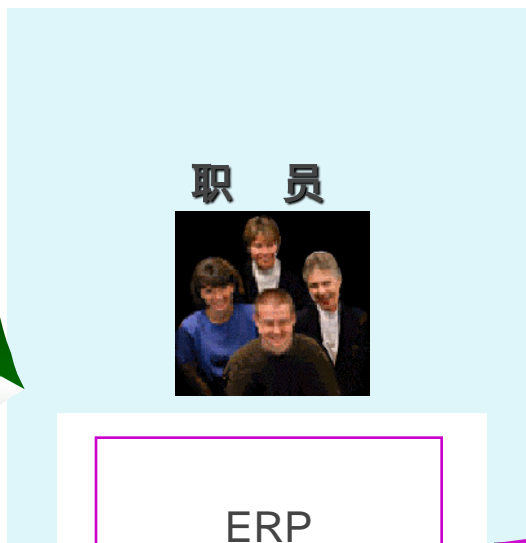
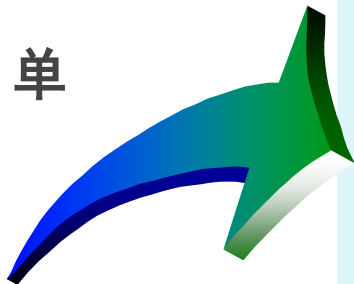
- ▶ **实时响应**：能够随时对动态的，不可预知的改变做出感知和响应。
- ▶ **随机应变**：能够在维持高生产力和财务可预测性时对业务流程和成本结构做出适应性修改，以降低风险。
- ▶ **聚焦核心**：能够集中资源于提高企业核心业务能力和那些体现企业竞争差异性的能力。
- ▶ **坚固可靠**：能够在始终贯彻其所有委托人(股东)的需求时，有效管理内部变化和外部威胁。

敏捷的电子供应链

订单

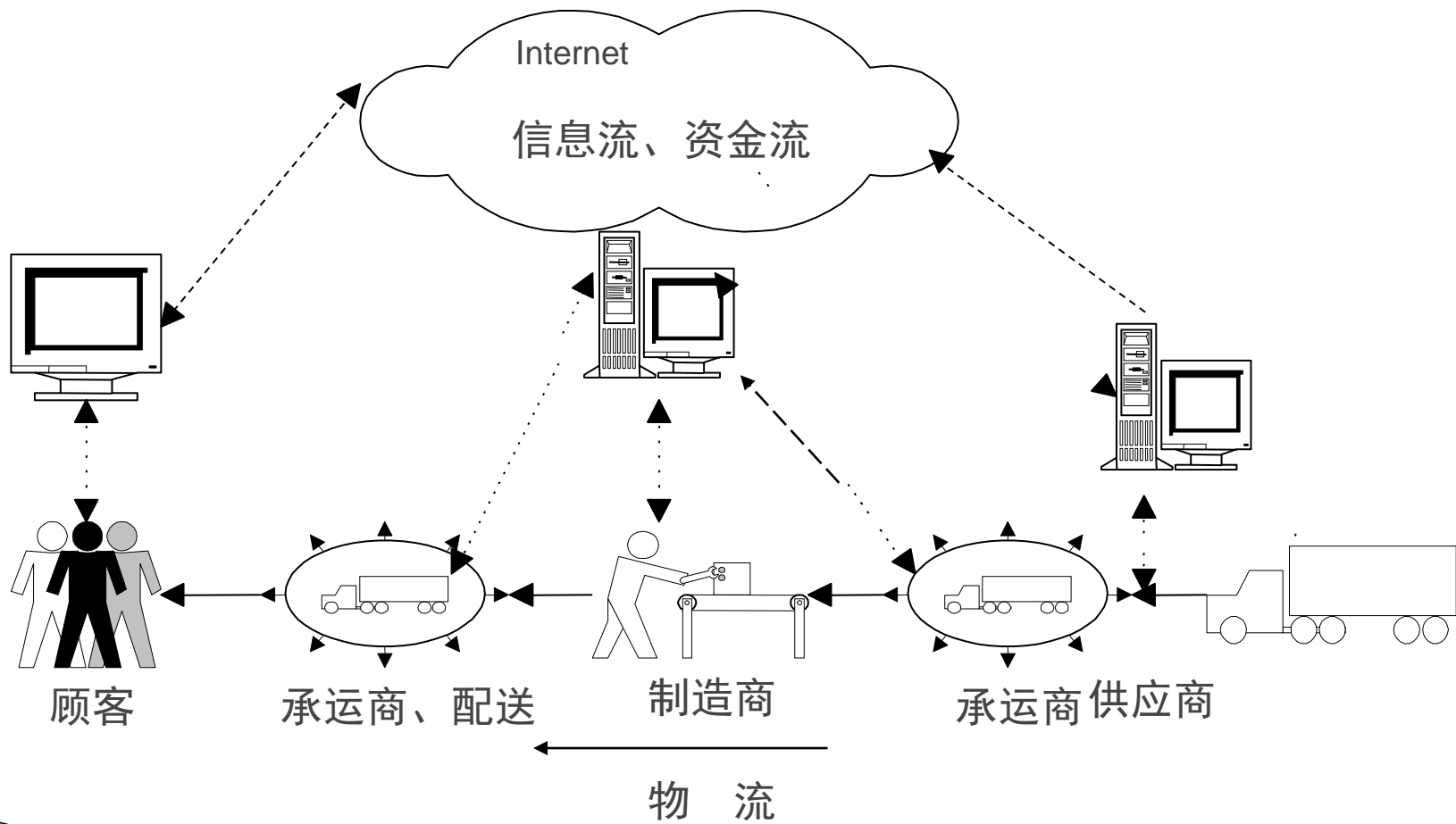


订单



传真

高效协同商务



按需应变运行环境的特点

- ▶ 整合性（Integrated）：企业将内部外部数据、应用、流程等有机的整合。
- ▶ 开放性（Open）：IT系统和应用遵循业界公认的开放技术标准和规范，如XML和Web Services等。
- ▶ 虚拟化（Virtualized）：在分布式计算环境下，利用grid computing等技术，使分布的IT资源真正实现随时随地的共享。
- ▶ 自主性（Autonomic Computing）：通过人工智能、神经网络等技术，使IT系统做到自保护、自优化、自配置。

新的模式

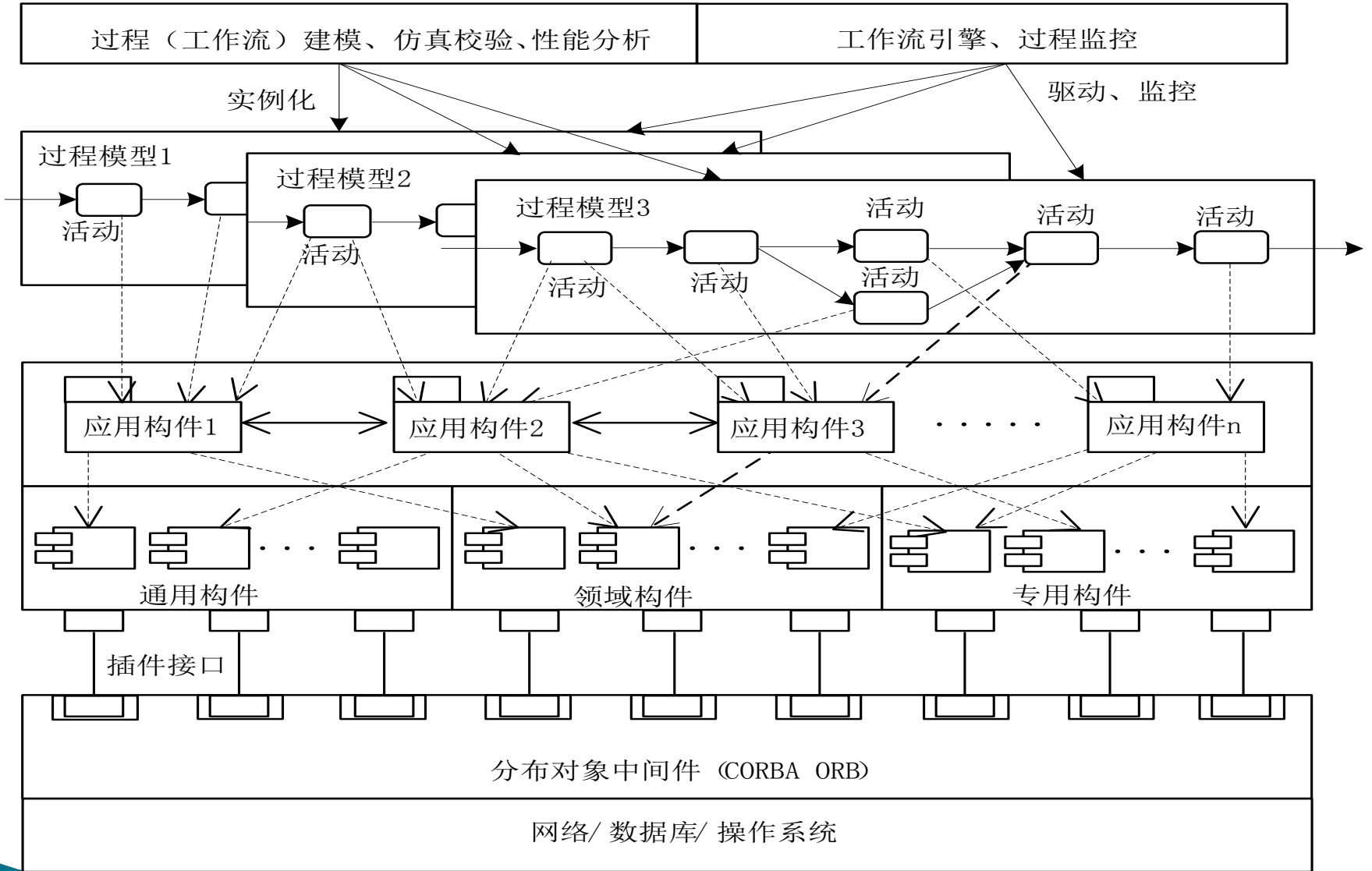
- ▶ Web Services是使用Web的崭新模式
 - 能够在分布式的计算环境中动态地描述、发布、发现和调用。
 - 许多基于Web Services的新型应用将出现



完全基于XML以及其他相关的Internet标准

电子商务应用实施的特点

- ▶ 经常会对电子商务的流程进行更改，这常常每周或每几天发生一次；
- ▶ 经常应用户的需求而进行更改，这甚至每个小时都会发生，尤其是当需要为每个客户、每个合作伙伴或每个企业员工都定制其首选的电子商务应用的时候。



不恰当的解决方案: 应用的复杂连接(1)

- ▶ 对于每个应用，程序员都为每个需要的企业资源或外部资源编写连接代码，以使得应用得以运行。
- ▶ 程序员还需要编写更多的代码以使得大量的用户能够访问每个应用。
- ▶ 应用与应用之间的集成同样需要编写大量的代码。

不恰当的解决方案: 应用的复杂连接(2)

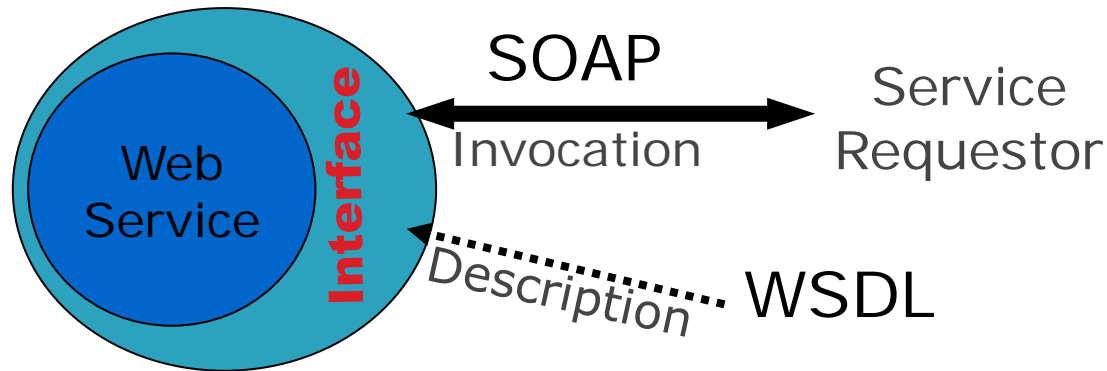
- ▶ 第一个应用，企业的为之付出的总的费用应该是该应用的开发和部署费用，以及运营时态的维护和更新费用。
- ▶ 第二个应用，应用的开发和部署费用是一样的，但是企业需要为之花费额外的集成费用，同时由于整个企业应用环境变得更加复杂，其运营时态的维护和更新费用可能呈指数形式增加。
- ▶ 当第三个、第四个应用被部署后，企业所支出的费用可能是高得惊人。

正确的解决方案Web Services

- ▶ 由程序员主导的由里向外的开发模式应当被由用户主导的由外向里的开发模式取代。
- ▶ 冗长的串行的开发循环应当被即时的，快速的应用装配所取代。
- ▶ 应用应当天生就具备高可定制性。
- ▶ 商业技术概念：“即时制造”以及“规模可伸缩”。

什么是Web Services(1)?

- ▶ 一个能够使用XML消息通过网络来访问的Interface, 这个Interface描述了一组可访问的操作。
 - 由SOAP+WSDL包装的Object
 - 适应松散耦合的网络环境, 可通过Web访问, 手段是 SOAP Message
 - 服务的行为、输入/输出都可使用WSDL描述



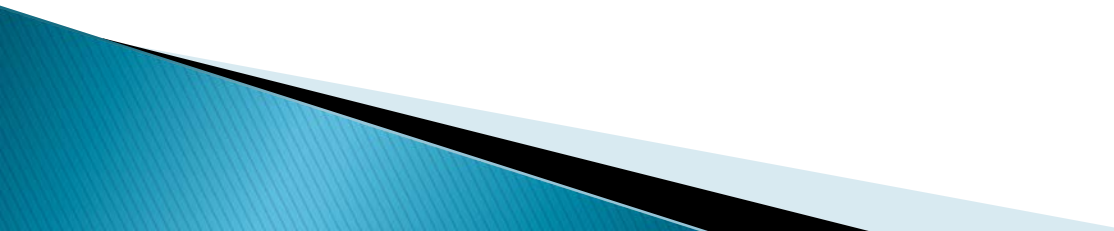
什么是Web Services(2)?

- ▶ e-Business驱动了Web Services的发展，而Web Services的基石是Web技术、IT技术和对象技术的融合。
 - 高度可集成的、基于Web的对象。
 - 通过SOAP Message实施的面向对象编程。
 - 能够将你现有的企业应用使用SOAP包装、WSDL描述，从而发布企业的商务功能或商务数据。

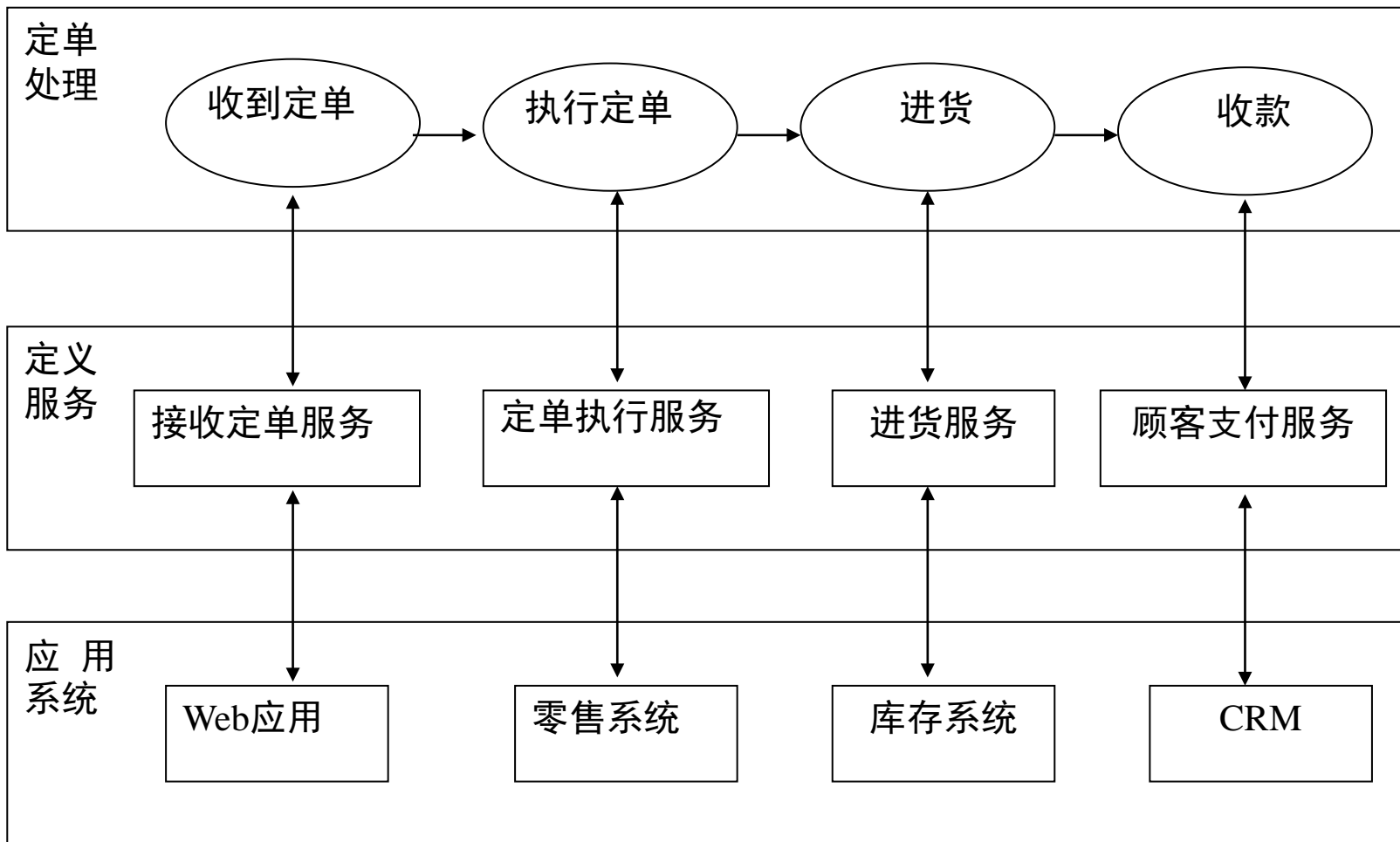
什么是Web Services(3)?

- ▶ Web services是基于XML的标准集，目的是方便应用的互操作（集成）。
- ▶ *“A Web service is a software system identified by a URI whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols”*(W3C , 2004)

Web Services的特点

- ▶ 与业务对应
 - ▶ 可组装
 - ▶ 可重用
- 

面向服务的订单处理流程



什么是 XML Web Service?

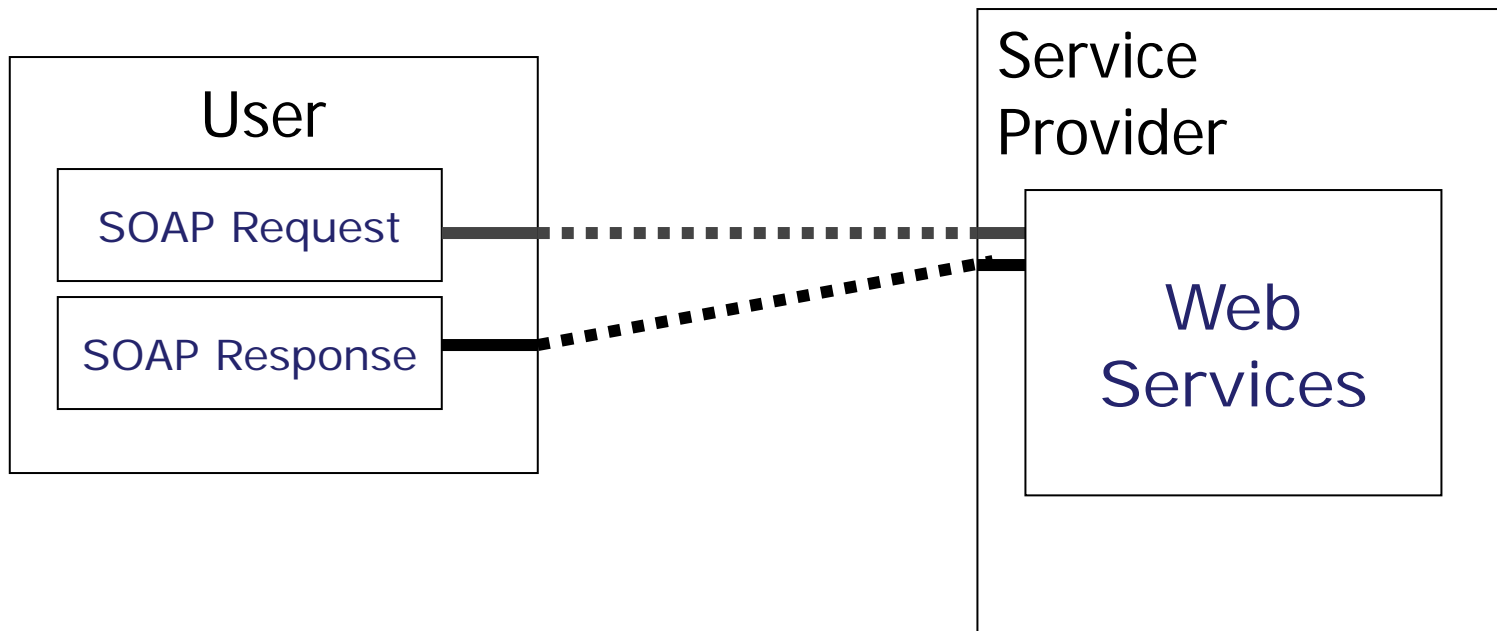


- UDDI - Web services 的黄页
- WSDL – 描述 Web service 的语言规范，相当于访问 Web services 的接口
- SOAP（简单对象访问协议）– 用 XML 实现 Web services 的标准协议
 - 基于开放的 Internet 协议

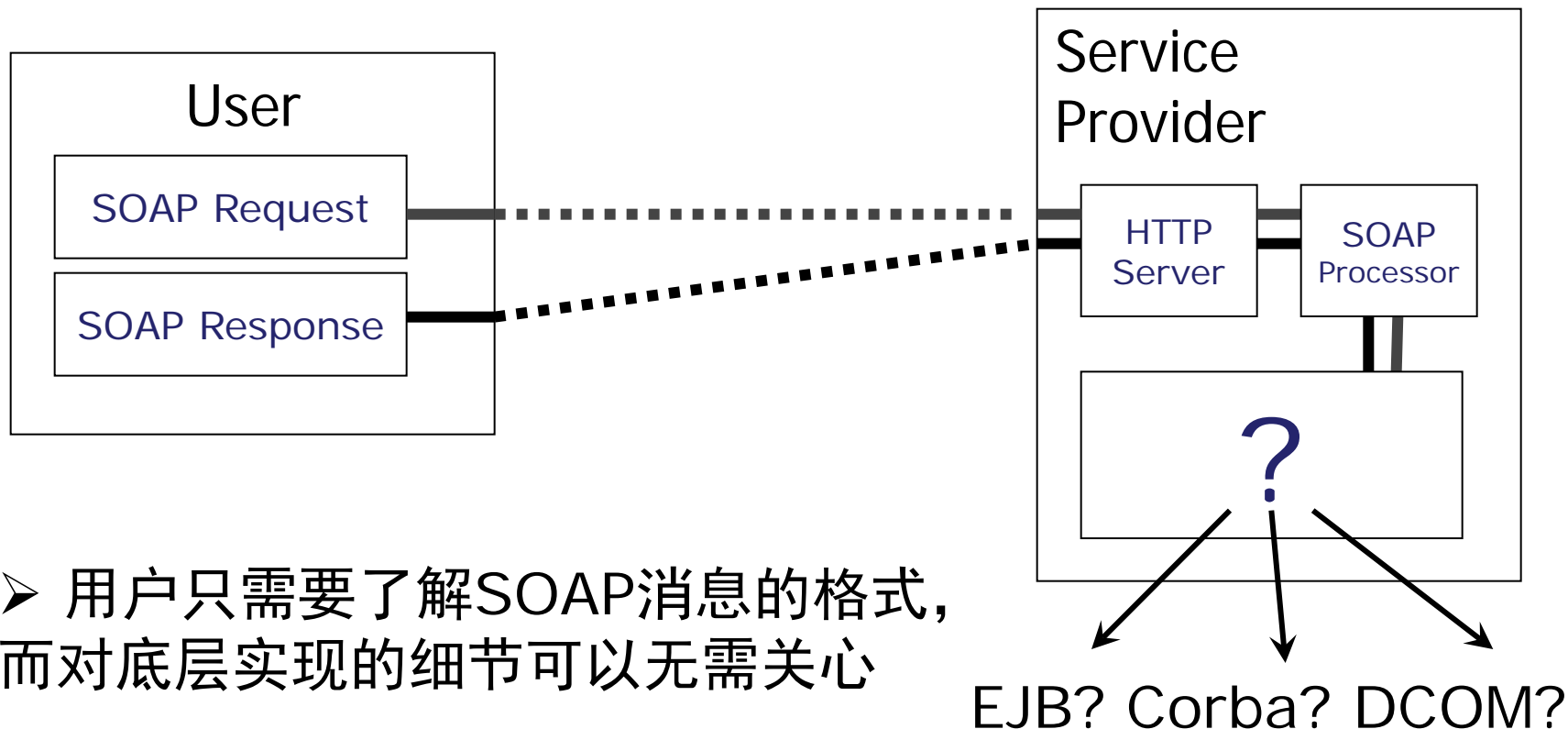


SOAP 消息结构

- ▶ Request/Response Message
 - Request 调用远端对象的某个方法
 - Response 返回该方法运行后的输出结果



SOAP 隐藏了实现细节



- 用户只需要了解SOAP消息的格式，而对底层实现的细节可以无需关心

什么是SOAP协议?

- ▶ SOAP是在网络上传递XML消息的协议。
- ▶ 一个 SOAP信息本身是一个XML文档。
- ▶ 网络、编程语言无关。

Envelope

Header

Holds processing information and directives

Body

Holds application specific information – for example, parameters for a Web service call

SOAP消息实例

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header/>
  <soapenv:Body>
    <p424:getQuote xmlns:p424="http://stockquote">
      <symbol>IBM</symbol>
    </p424:getQuote>
  </soapenv:Body>
</soapenv:Envelope>
```

Web Services 描述语言(1)

- ▶ 使用XML进行描述
 - 类似IDL, 不过是使用XML格式
- ▶ 描述了服务的操纵信息
 - Service Interface
 - Implementation Details
 - Access Protocol
 - Contact Endpoint
- ▶ WSDL是早先技术的综合
 - IBM's NASSL
 - Microsoft's SDL

Web Services 描述语言(2)

- ▶ **Web Services** 定义语言 (WSDL) 提供了描述服务接口和绑定（调用）细节的重要标准。
- ▶ 使用XML语言描述Web services:
 - 操作
 - 调用
- ▶ **WSDL** 向潜在的用户描述了服务可以提供的功能、这些功能使用的参数以及这些功能返回的结果等。

WSDL的要素

- ▶ **类型 (Type)**
 - 定义操作参数的XML数据类型。
- ▶ **信息 (Message)**
 - 定义特定的服务操作需要的输入、输出等信息的结构。
- ▶ **操作 (Operation)**
 - 定义服务支持的功能。
- ▶ **端口类型 (Port type)**
 - 定义服务的功能，由一些操作描述。对每种操作，描述操作所需要的输入以及产生的输出数据。
- ▶ **绑定 (Binding)**
 - 定义服务如何调用。
- ▶ **端口 (Port)**
 - 描述服务所在的地址。

WSDL实例

```
<wsdl:definitions targetNamespace="http://stockquote" xmlns:impl=http://stockquote
  xmlns:intf=http://stockquote
  xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/
  xmlns:wsdlsoap=http://schemas.xmlsoap.org/wsdl/soap/
  xmlns:wsi=http://ws-1.org/profiles/basic/1.1/xsd
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <wsdl:types>
    <schema targetNamespace="http://stockquote"
      xmlns=http://www.w3.org/2001/XMLSchema
        xmlns:impl="http://stockquote" xmlns:intf=http://stockquote
        xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/
        xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <element name="getQuoteResponse">
        <complexType>
          <sequence>
            <element name="getQuoteReturn" type="xsd:float"/>
          </sequence>
        </complexType>
      </element>
      <element name="getQuote">
        <complexType>
          <sequence>
            <element name="symbol" nillable="true" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
    </schema>
  </wsdl:types>
</wsdl:definitions>
```

WSDL信息与端口类型

```
<wsdl:message name="getQuoteResponse">  
  <wsdl:part element="impl:getQuoteResponse" name="parameters"/>  
</wsdl:message>  
<wsdl:message name="getQuoteRequest">  
  <wsdl:part element="impl:getQuote" name="parameters"/>  
</wsdl:message>
```

Message

```
<wsdl:portType name="StockQuotePT">  
  <wsdl:operation name="getQuote">  
    <wsdl:input message="impl:getQuoteRequest"  
      name="getQuoteRequest"/>  
    <wsdl:output message="impl:getQuoteResponse"  
      name="getQuoteResponse"/>  
  </wsdl:operation>  
</wsdl:portType>
```

Port type

WSDL绑定

```
<wsdl:binding name="StockQuoteServiceSoapBinding"
              type="impl:StockQuotePT">
  <wsdlsoap:binding style="document"
                    transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getQuote">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getQuoteRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="getQuoteResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

Binding

WSDL服务

```
<wsdl:service name="StockQuoteService">  
  <wsdl:port binding="impl:StockQuoteServiceSoapBinding"  
    name="StockQuotePort">  
    <wsdlsoap:address location="http://localhost:9080/..." />  
  </wsdl:port>  
</wsdl:service>
```

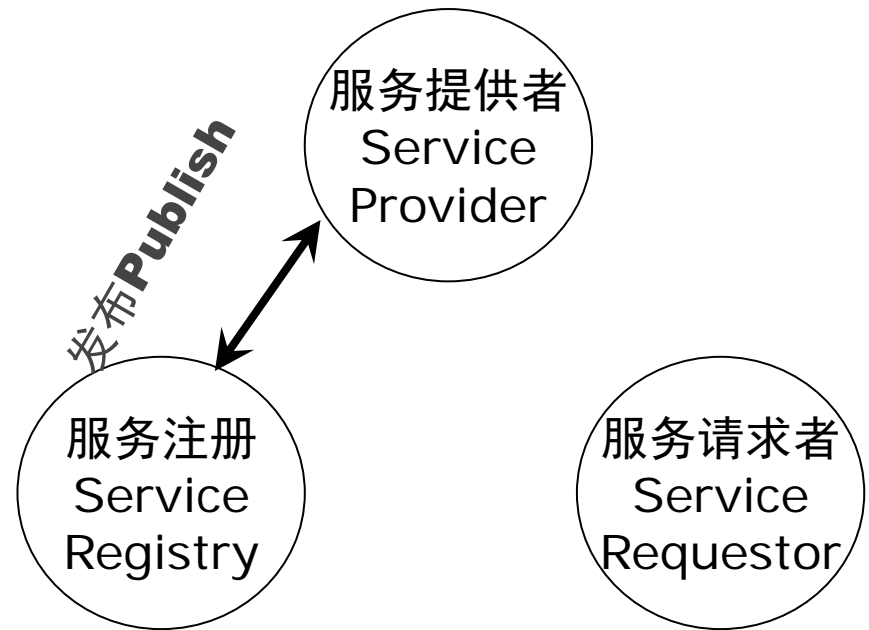
Service

什么是 UDDI?

- ▶ Universal Description, Discovery and Integration
- ▶ 为加速Web Services的推广、加强Web Services的互操作能力而推出的一个计划
 - 基于标准的服务描述和发现的规范(specification)
 - 以资源共享的方式由多个运作者一起以Web Services的形式运作UDDI商业注册中心

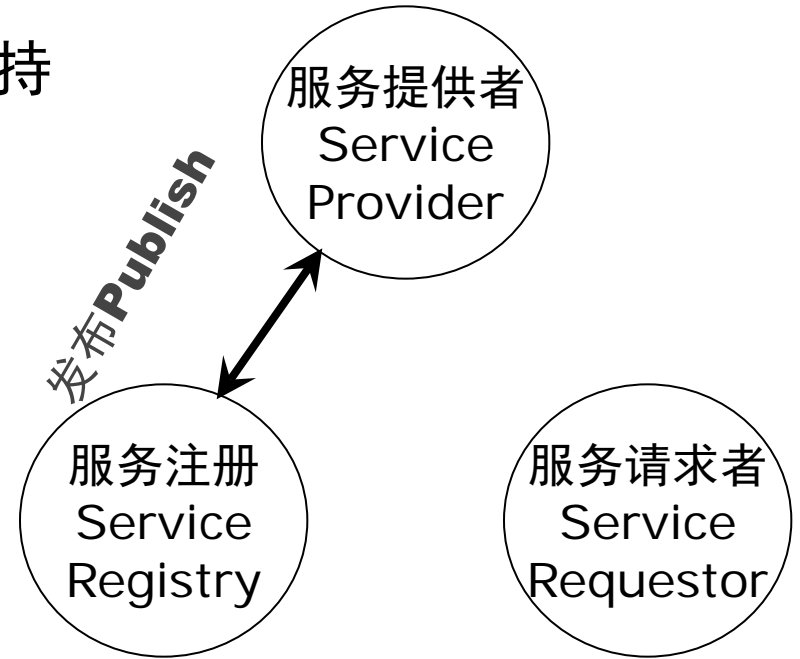
UDDI 角色和操作(1)

- ▶ 服务提供者 (Service Provider)
 - 提供e-Business Service
 - 通过服务注册Service Registry发布(Publish)其提供的可用的Service



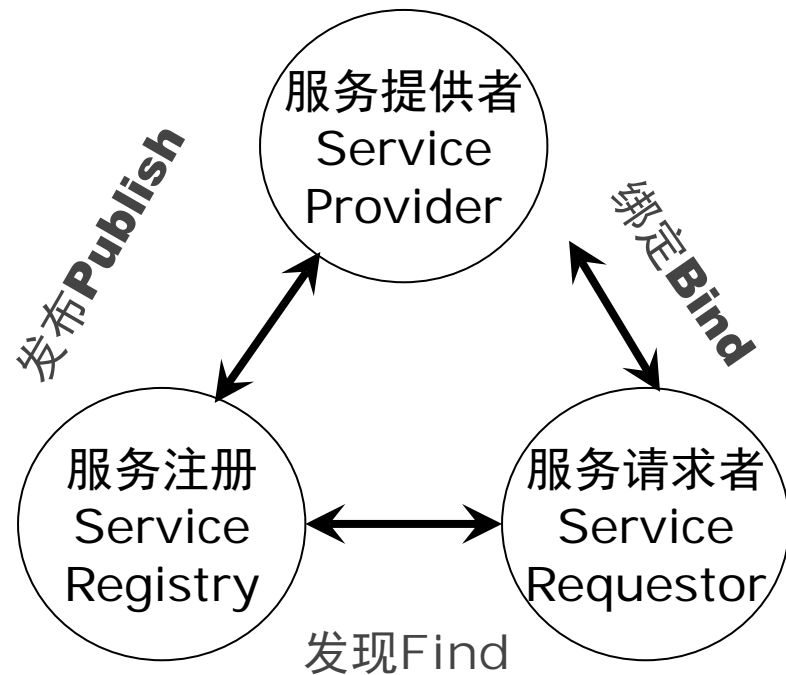
UDDI 角色和操作(2)

- ▶ 服务注册Service Registry
 - 为Service的发布和定位提供支持
 - 类似电话黄页



UDDI 角色和操作(3)

- ▶ 服务请求者Service Requestor
 - 通过 服务注册Service Registry发现(Find)需要的Service
 - 绑定(Bind) Service Provider提供的Service, 并实施调用



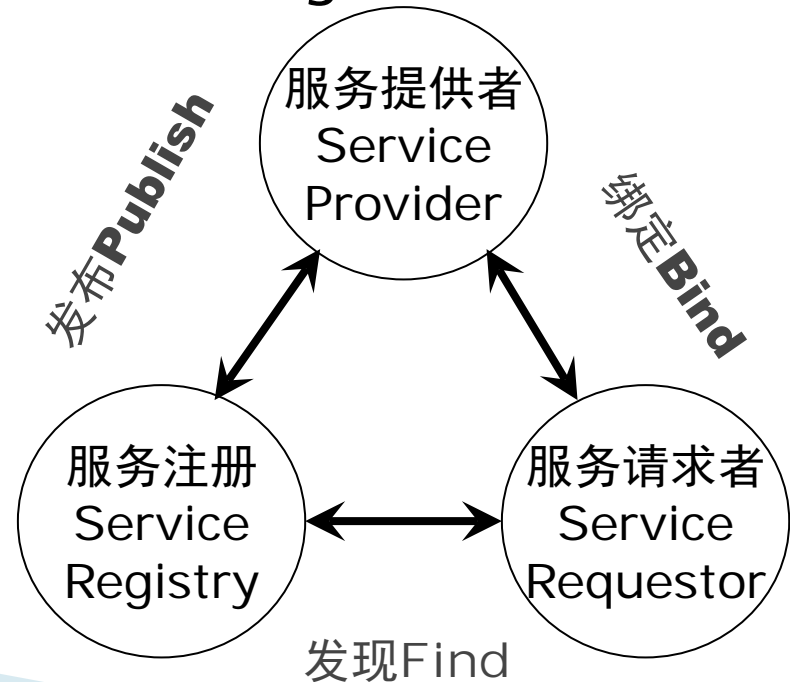
SOAP 与WSDL?

▶ WSDL

- 发布Publish的内容、发现Find的返回结果和绑定Bind的信息都是WSDL描述的服务信息。

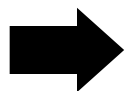
▶ SOAP

- 服务注册Service Registry的访问(Publish/Find)、Service的访问都是通过SOAP Message实现。



UDDI解决了什么问题？

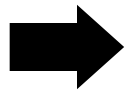
更广泛的
B2B



一个中等规模的制造型企业需要和大约400个合作伙伴架构在线的交易关系，而每一个交易关系的连接可能都有其自身的标准和协议

描述服务

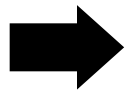
更智能地
搜索



中国的一家花店想要将他提供的服务加入到全球所有合适的e-Marketplace中去，但却不知道该如何寻找这些e-Marketplace

发现服务

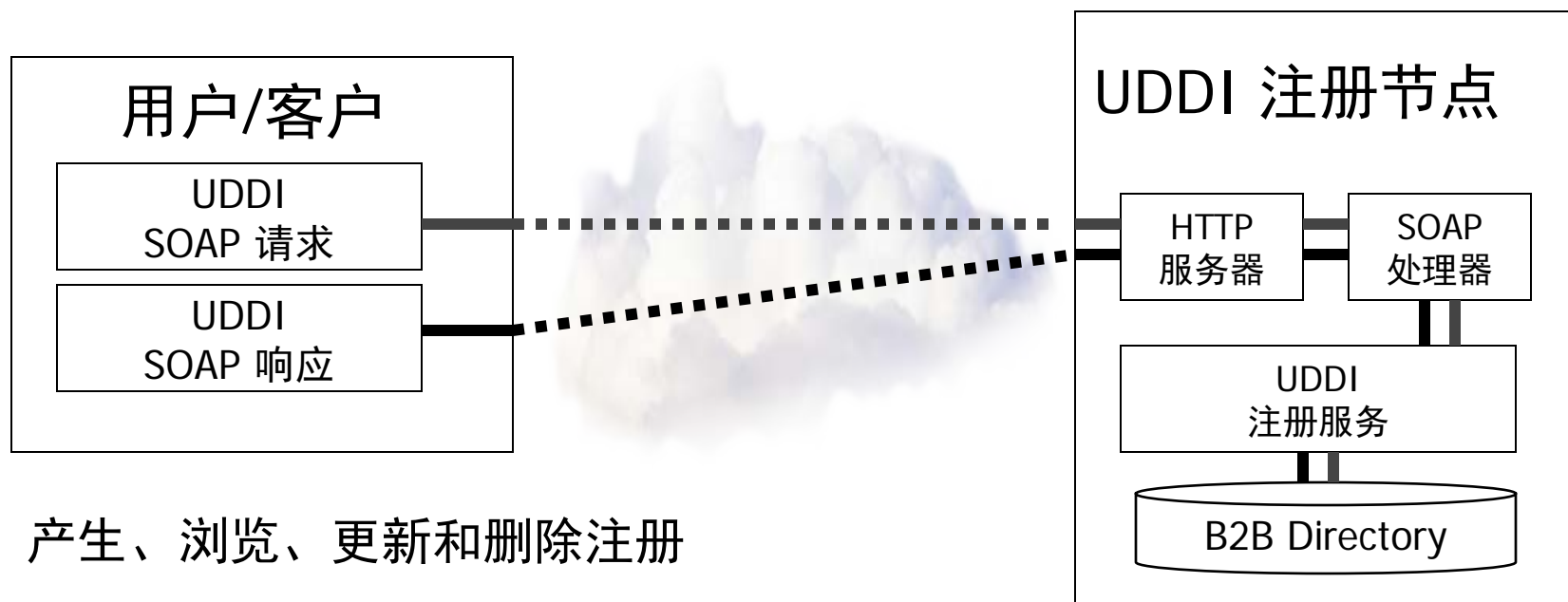
更容易的
资源汇聚



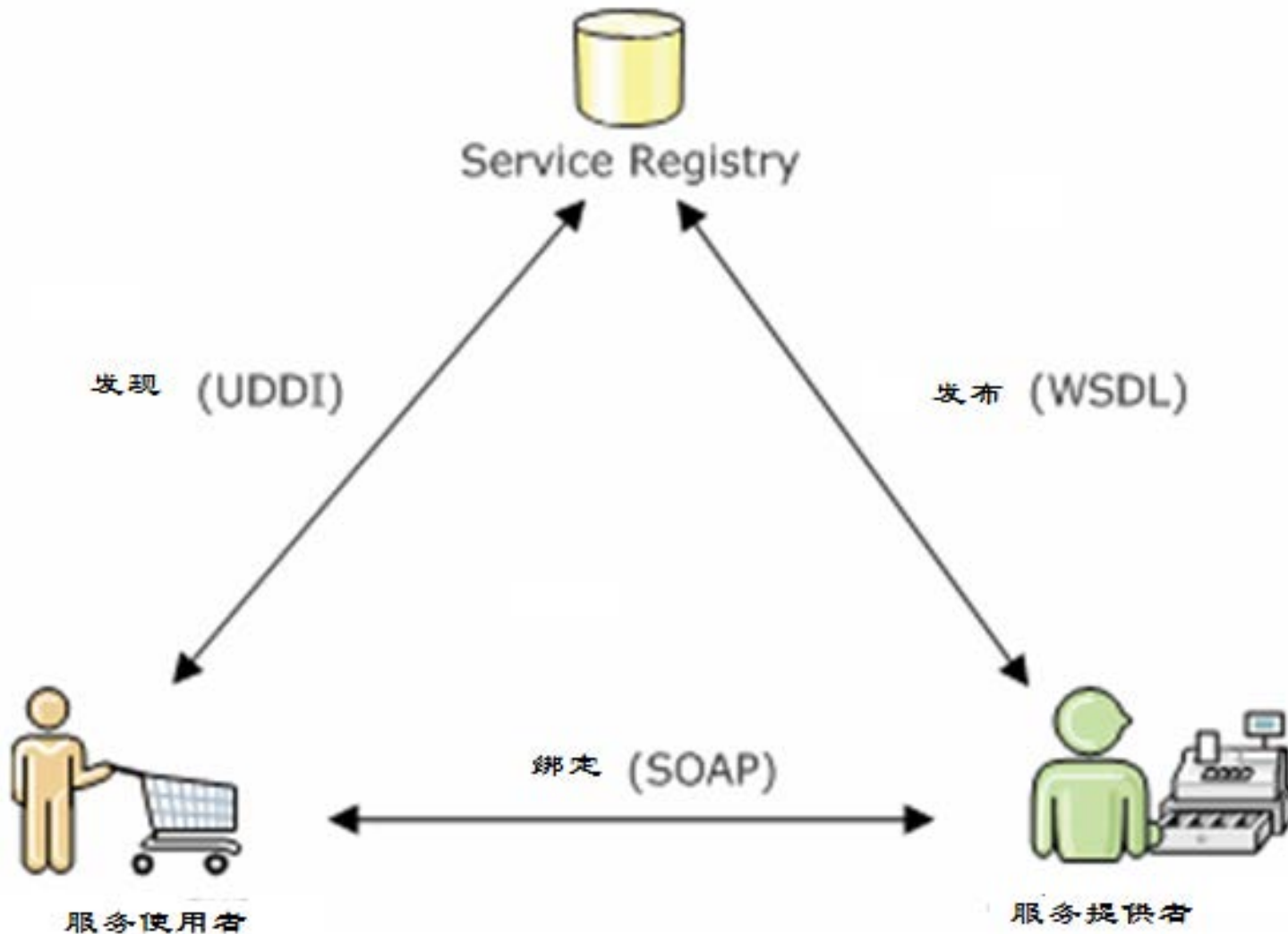
一家B2B的e-Marketplace无法顺利地获取行业内及行业外的相关供应商的供应目录数据，对于行业内的承运商等也同样如此。

互相集成

UDDI 和 SOAP



UDDI、WSDL和SOAP



Web Services: 业务革命

- ▶ “doing e-business”革命性的方式
 - 全球商务产品和服务的数码黄页
 - 使得新型e-Business应用以及动态服务集成能够迅速发展。
- ▶ 机遇: 第三方的增值服务的出现
 - 基于UDDI的业务搜索 (Business Search)
 - 全球性的行业市场给予如此丰富集中的高价值的商业数据, 事实上可能出现的服务的范围将异常地广泛。

Web Services: 软件革新

- ▶ 并非软件的革命
 - 基于现有标准
 - 是现有系统的一个延展
 - 并不需要一个新的编程语言
- ▶ SOAP 模型不是新事物
 - 与20年前的RPC具有同样的目的

Web Services: 软件革新

- ▶ What's new: 为调用远端的对象提供了一整套的Internet规范，而使远程调用轻便非凡
 - 使用了实现中立的消息格式
 - UDDI: 一个可用service的通用统一的目录
 - 为商业聚合提供了不同层次的服务

Web Service会成功吗?

- 美国InfoWorld杂志在2001年进行了一次有关Web Services应用的调查，结果显示：在被调查的企业中，有38%的企业希望在未来3个月内实施Web Services战略；11.4%的企业认为Web Service对企业的发展非常重要。目前，业界最重要的软、硬件厂商都已经宣布了对Web Services的支持。



Microsoft



intel.

ORACLE



COMMERCE ONE.



COMPAQ



webMethods.

Rational
the e-development company

Entrust
Technologies
We Bring Trust to e-Business™

KPMG Consulting

NORTEL
NETWORKS

FUJITSU

CLARUS

xmlsolutions



Dealeasy.



accenture

PROGRESS
SOFTWARE

documentum

VerticalNet

VeriSign
Internet Trust Services

电子商务按需应变电子商务

- “按需应变的电子商务” (e-business on demand)。简单地说，就是企业用户需要企业管理程序、商业数据库资料时，不必再独立投资建立内部的全套软件和程序，只需到IBM提供的网上企业电子商务应用软件库里去调用。

ASP(application service provider)模式

- 好处:不用再付软件和程序本身的高额费用，而只需付软件使用费，成本大大降低，使用更加简便和经济。其次，企业的电子商务软件在使用过程中，可以更直接地得到IBM专家的协助和技术支持，效率可以大大提高。