

Key Terms of SOA

Prof. Vipul Dabhi

Department of Information Technology,

D. D. University.

Service, Service Orientation and SOA

... a service?

A **repeatable business task** – e.g., check customer credit; open new account

... service orientation?

A way to integrate your **business as linked services** with their unique outcomes

... service oriented architecture (SOA)?

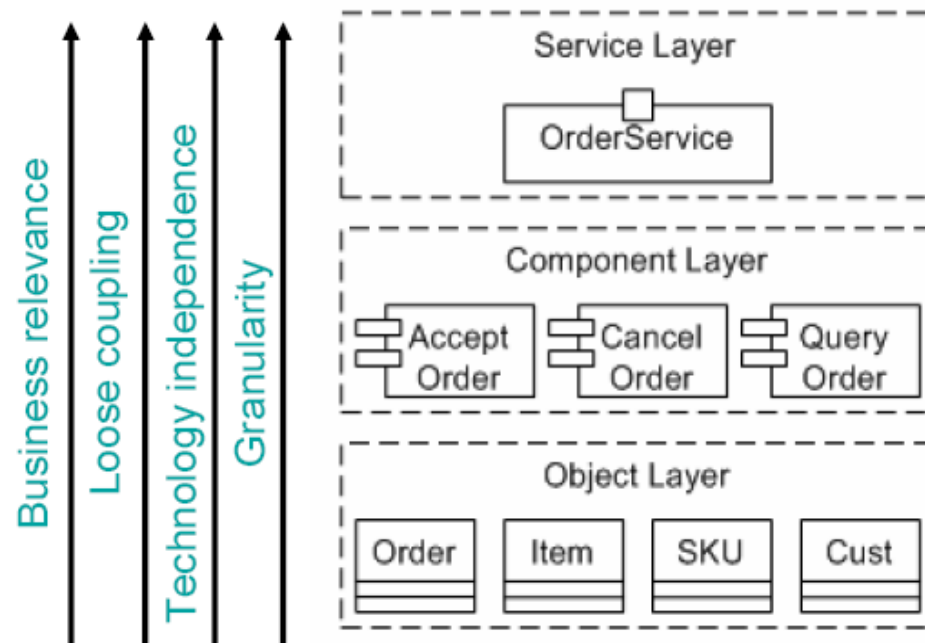
An IT **architectural style** that supports service orientation

... a composite application?

A set of **related & integrated** services that support a business process built with SOA

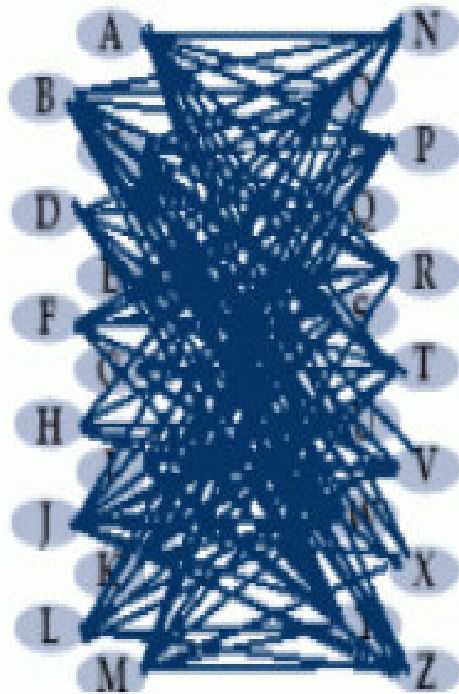
From Objects to Components to Services

- There has been a progression from Objects through to Services
- There is much similarity between these concepts and many object-oriented and component-based design techniques can be applied to SOA
- The key difference is the design principles behind individual services and their interaction patterns



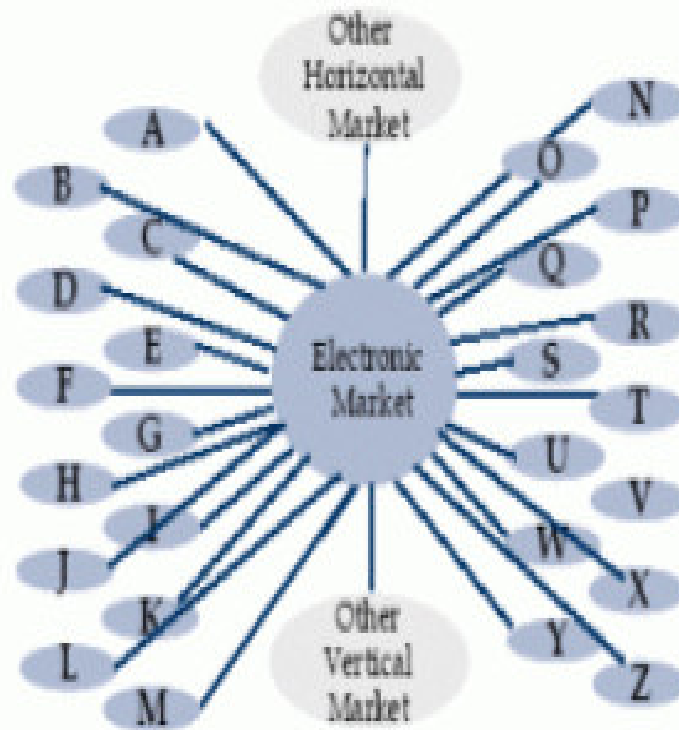
Business interaction models approaches compared

Traditional EDI or B2B



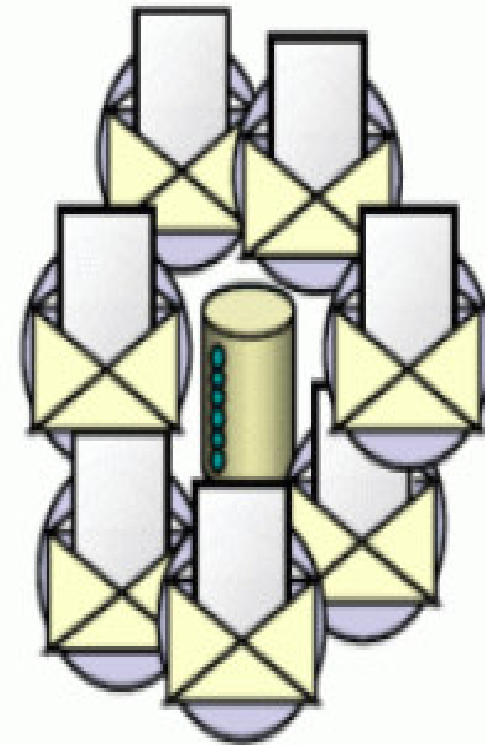
Integrate point to point

Marketplace architecture



Integrate once to the hub
Subscribe to limited services

SOA approach



Integrate once to the network
Publish service to others
Subscribe to others

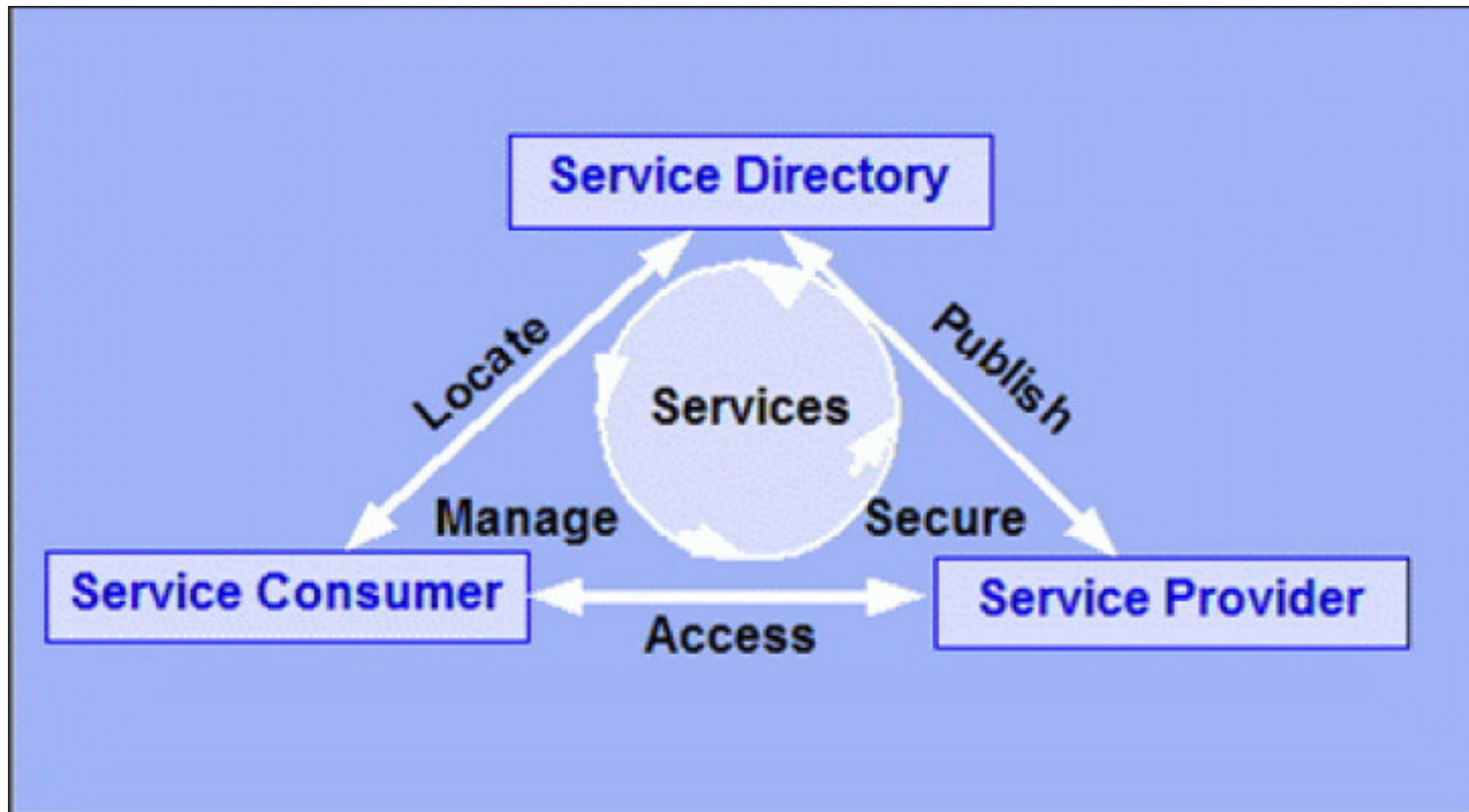
What is a Service Oriented Architecture?

- An **approach** for **building** distributed systems that deliver application functionality as **services** to either end-user applications or other services

It defines :

- ✓ An architecture that leverages **open standards** to represent software **assets as services**.
- ✓ Provides a **standard way of representing and interacting** with software assets
- ✓ Individual software assets become **building blocks** that **can be reused** in developing other applications
- ✓ **Shifts focus to application assembly** rather than implementation details
- ✓ Used internally to **create new applications out of existing components**
- ✓ Used externally to **integrate with applications outside of the enterprise**

Elements of SOA



Service Oriented Architecture

- SOA is not a new idea. It have previously been implemented in proprietary ways. That means it was very difficult if not impossible for disparate computers to intelligently communicate with each other. So what's different now?
- New implementation technique has arisen to make SOAs much more cost efficient and productive. This new technique is called Web Services.
- It is important because it finally breaks the proprietary barrier between vendors and software programs.
- Major vendors like IBM and Microsoft have agreed to standards which allow their respective hardware and software to share information and data.

What is a Web service?

A Web service is a software system

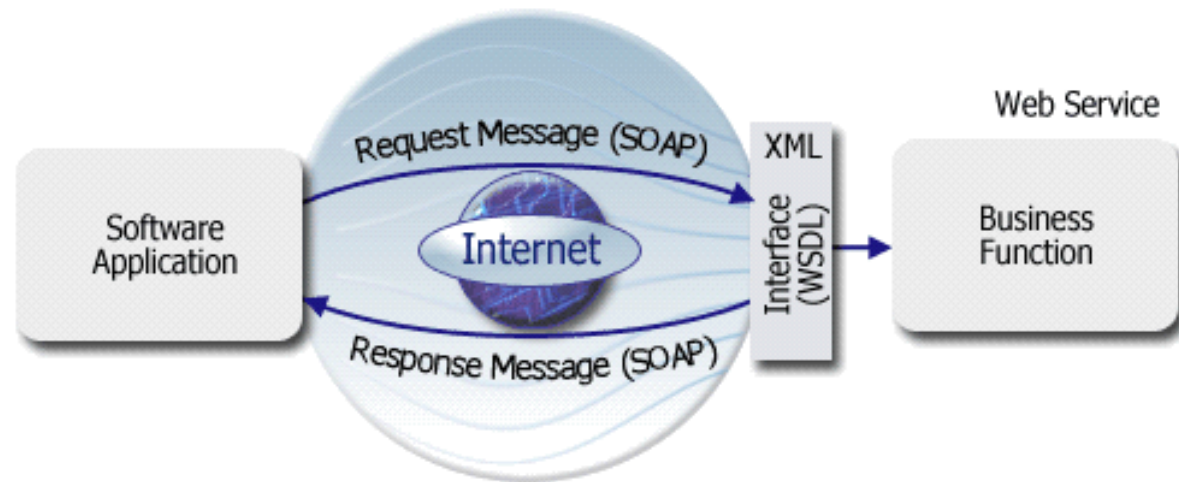
- *identified by a URL,*
- *whose public interfaces and bindings are defined*
- *and described using XML.*
- *Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML-based message conveyed by Internet protocols.*

WS is an Open Standard for accessing component based applications.

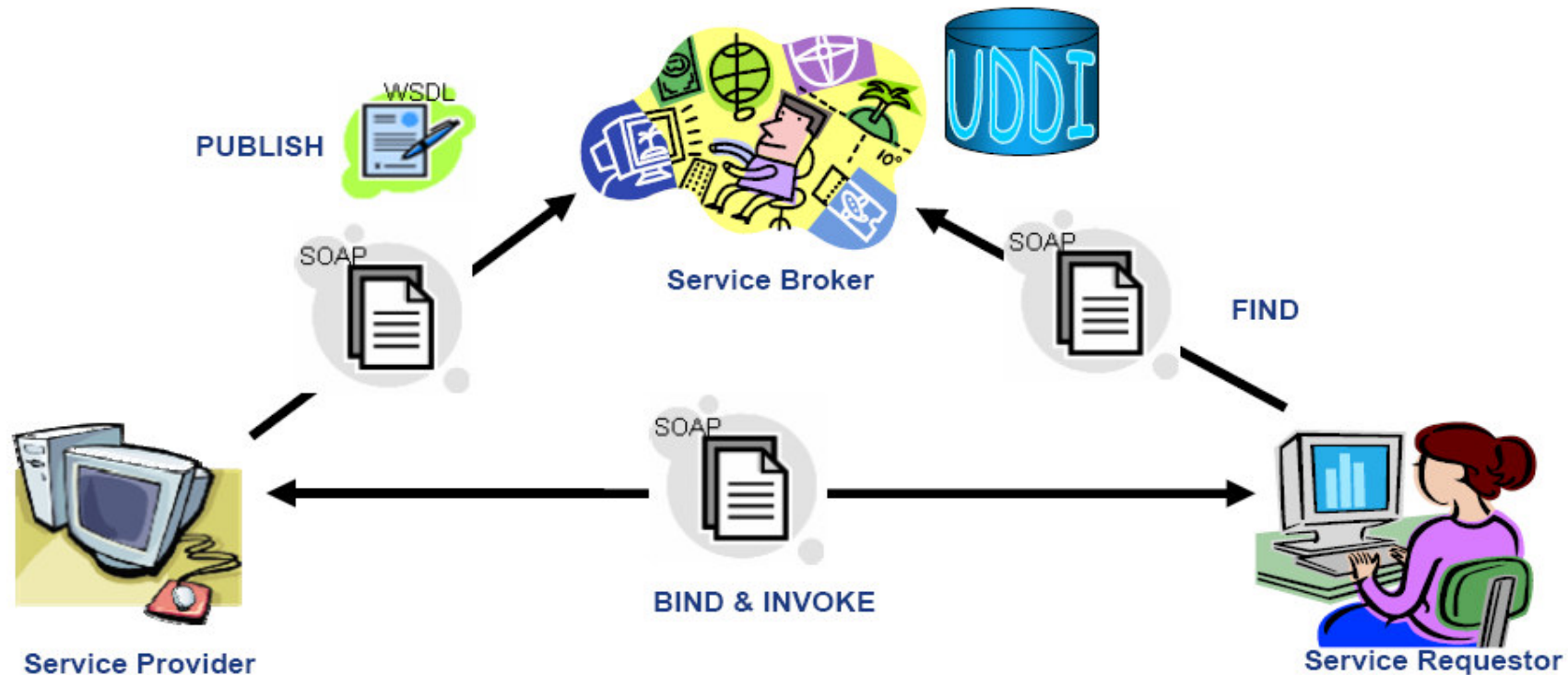
It is Standards-based, Cross-platform and cross-language, Widely supported and Message-oriented.

Web Services ...

- Specification to allow an application (component) to be invoked remotely. Caller need to know **nothing** about how application is implemented
- Described as a series of well adopted technology standards:
 - WSDL, SOAP, UDDI, more....
- New applications will expose themselves as Web Services
- Existing applications can be exposed as Web Services through adapters without changing code



Web Services – an implementation of SOA



Loosely coupled standards based implementation of SOA

Web services relationship to SOA

- Web services is a leading technology choice to use for implementing SOA today:
 - Standards-based
 - Cross-platform and cross-language
 - Widely supported
 - Message-oriented
- Faster tooling support speeds implementation of SOA
- There are many Web services implementations that are not SOA, such as connecting two heterogeneous systems directly together.

Characteristics of Web services

- **Protocol based** – They do not dictate the technology used by the requester or provider's applications, just the protocols by which they exchange messages.
- **Open standards** – They are based upon by the community to ensure interoperability.
- **Platform independent** – They are platform independent.
- **Transport independent** – They are transport independent.
- **Machine readable** – They are machine readable and understandable.
- **Self describing** – They are self describing and required to use standard protocols.
- **Composable** – They are composable and required to fully describe the services they provide.
- **Extensible** – Can be easily extended with additional protocols.

Independence

- Platform
- Location
- Implementation
- Information Format

Web Services Core Standards Stack

Standard discovery and registry- UDDI

Standard Descriptor - WSDL

Standard Messaging - SOAP

Standard Data Types – XML Schema

Standard Language - XML

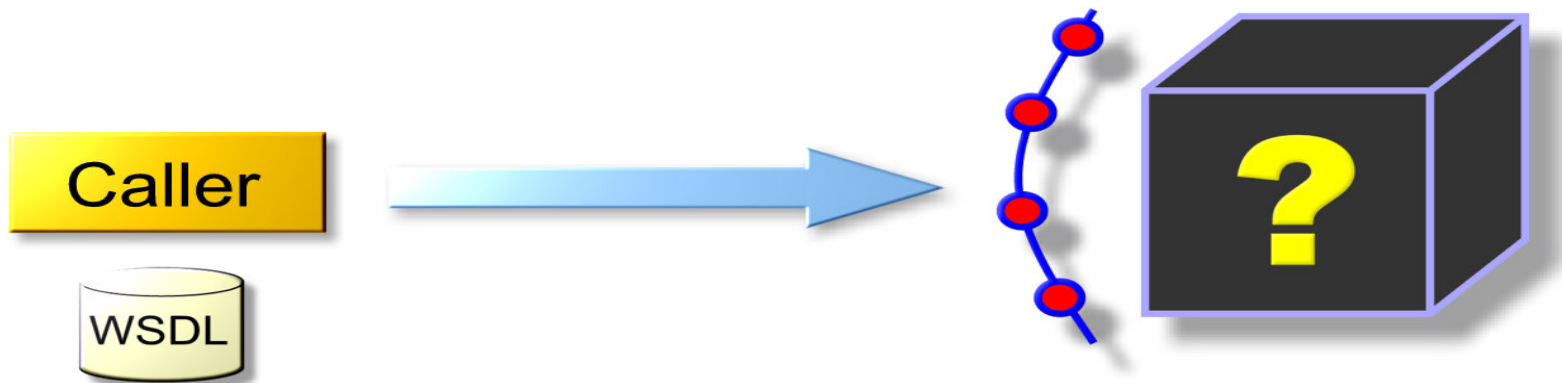
What is XML?

- XML was designed to structure, store and to send data/information.
- At its core XML is text formatted in the form of tags and text that follow a well-defined set of rules.
- This text may be stored/represented in:
 - A normal file stored on disk
 - A message being sent over HTTP
 - A character string in a programming language
 - A CLOB (character large object) in a database
 - Any other way textual data can be used

```
<purchaseOrder id="12345" secretKey="4x%$^">  
  <customer id="A6789">  
    <name>John Smith Co</name>  
    <address>  
      <street>1234 W. Main St</street>  
      <city>Toledo</city>  
      <state>OH</state>  
      <zip>95141</zip>  
    </address>  
  </customer>  
  <itemList>  
    <item>  
      <partNo>A54</partNo>  
      <quantity>12</quantity>  
    </item>  
    <item>  
      <partNo>985</partNo>  
      <quantity>1</quantity>  
    </item>  
  </itemList>  
</purchaseOrder>
```

WSDL ...

- Web Services Description Language
- Open Standard for describing Interfaces to Services
- Characteristics
 - Describes data expected to be sent and received
 - Describes what the Service can do
 - Describes how to reach the service
- WSDL description is an XML document that conforms to the WSDL standard



WSDL Document

■ Service Interface

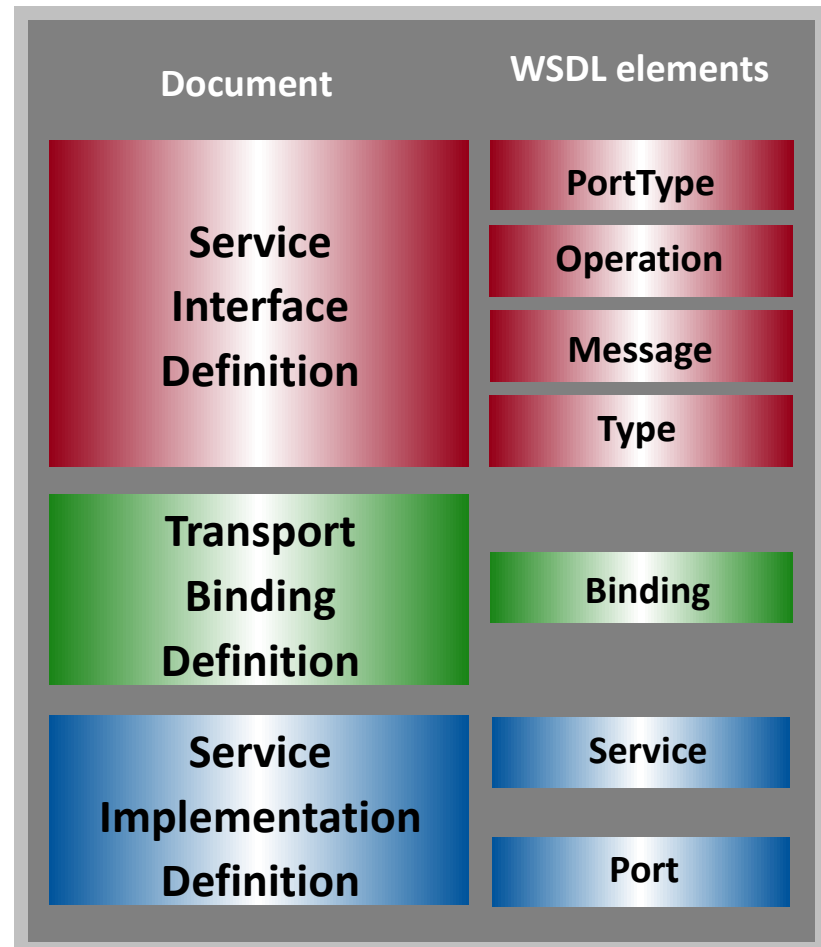
- f Abstract, reusable service definition
- f Represents a type of service that can be implemented
- f Elements: types, message, portType

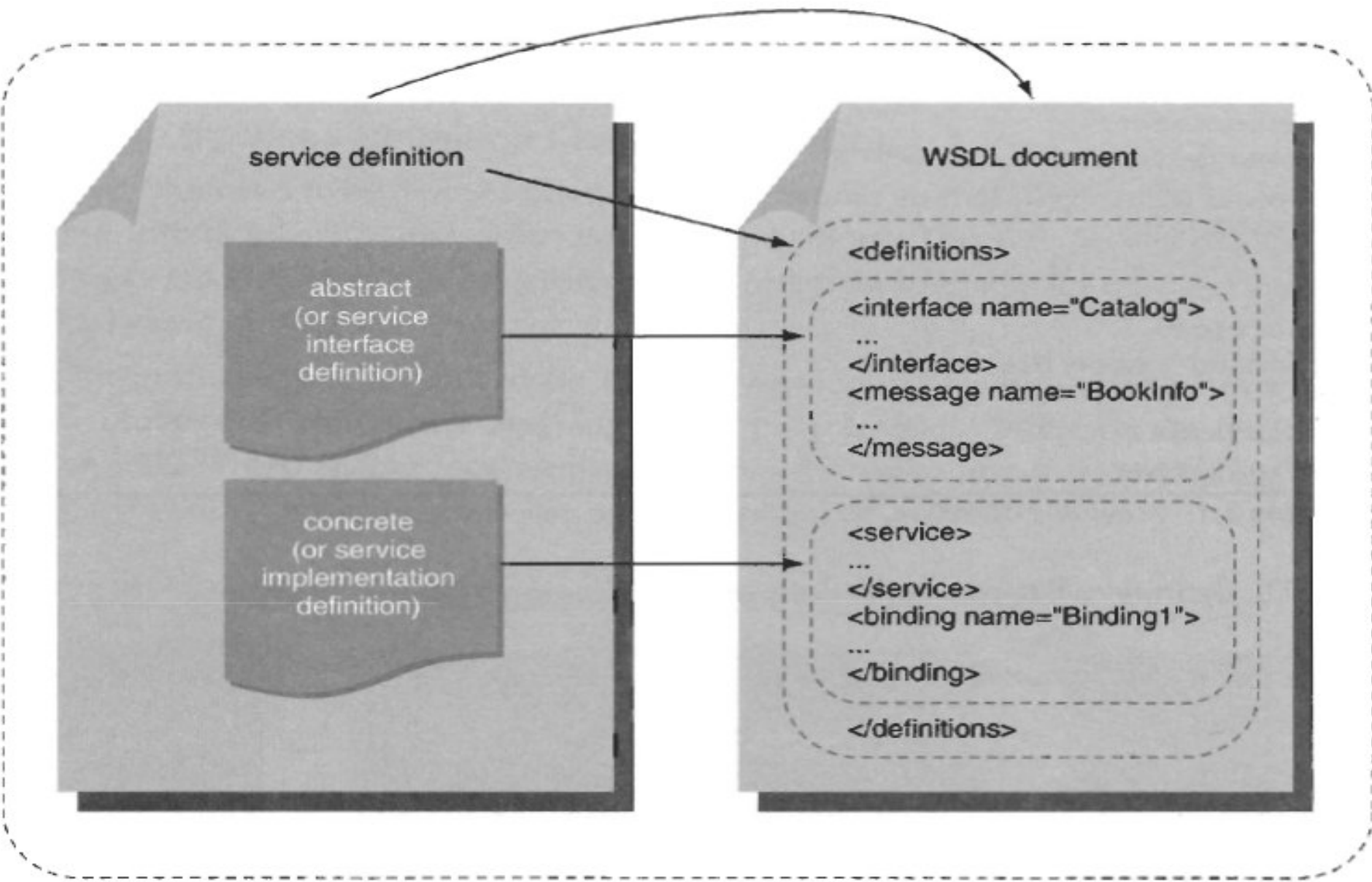
■ Service Binding

- f how messages are placed on a given transport
- f Elements: types, message, portType

■ Service Implementation

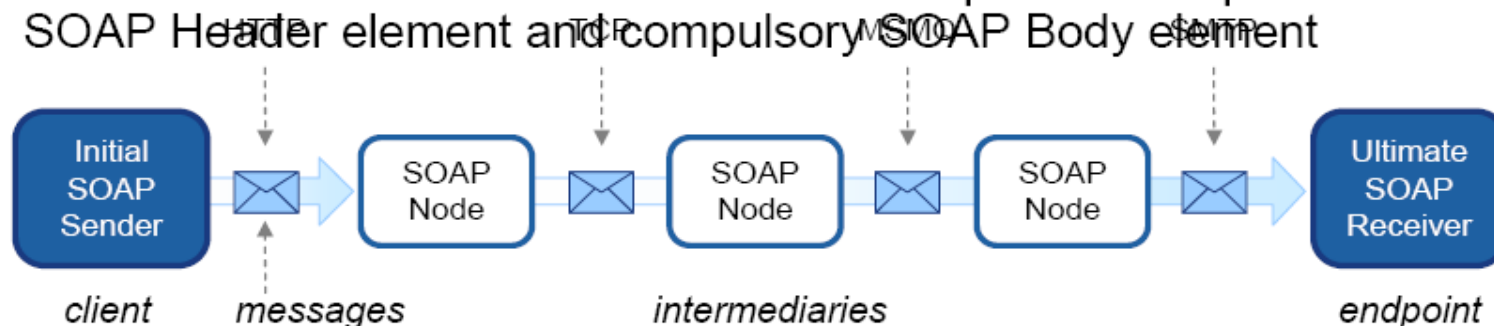
- f Implementation of one or more service interfaces
- f Contains the endpoint reference
- f Elements: port and service





What is SOAP? Simple Object Access Protocol

- Is XML based stateless, one-way message exchange paradigm for exchanging structured and typed information in a decentralized, distributed environment
- Provides the framework by which application specific information may be conveyed in an extensible manner
- Applications can create more complex interaction patterns (e.g., request/response, request/multiple responses, etc.) by combining such one-way exchanges with features provided by an underlying protocol and/or application-specific information
- Is silent on the semantics of any application-specific data it conveys, as it is on issues such as the routing of SOAP messages, reliable data transfer, firewall traversal, etc
- Skeleton of SOAP contains a SOAP Envelope with an optional SOAP Header element and compulsory SOAP Body element



SOAP Request-Response

SOAP request message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <m:GetLastTradePriceInput xmlns:m="Some-URI">
      <m:tickerSymbol>IBM</m:tickerSymbol>
    </m:GetLastTradePrice>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP response message

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <m:GetLastTradePriceOutput xmlns:m="Some-URI">
      <m:price>75.5</m:price>
    </m:GetLastTradePriceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

What is UDDI? Universal Description Discovery and Integration

- An UDDI implementation that provides the ability for a potential service consumer to determine the identity and connection details of available services
- The UDDI server provides the relevant details for specific services categorized on the basis of industry, service type etc
- The UDDI API provides the ability to search for specific services based on functionality
- Adoption of UDDI is likely to be at an intra enterprise level initially

The UDDI implementation enables a business to

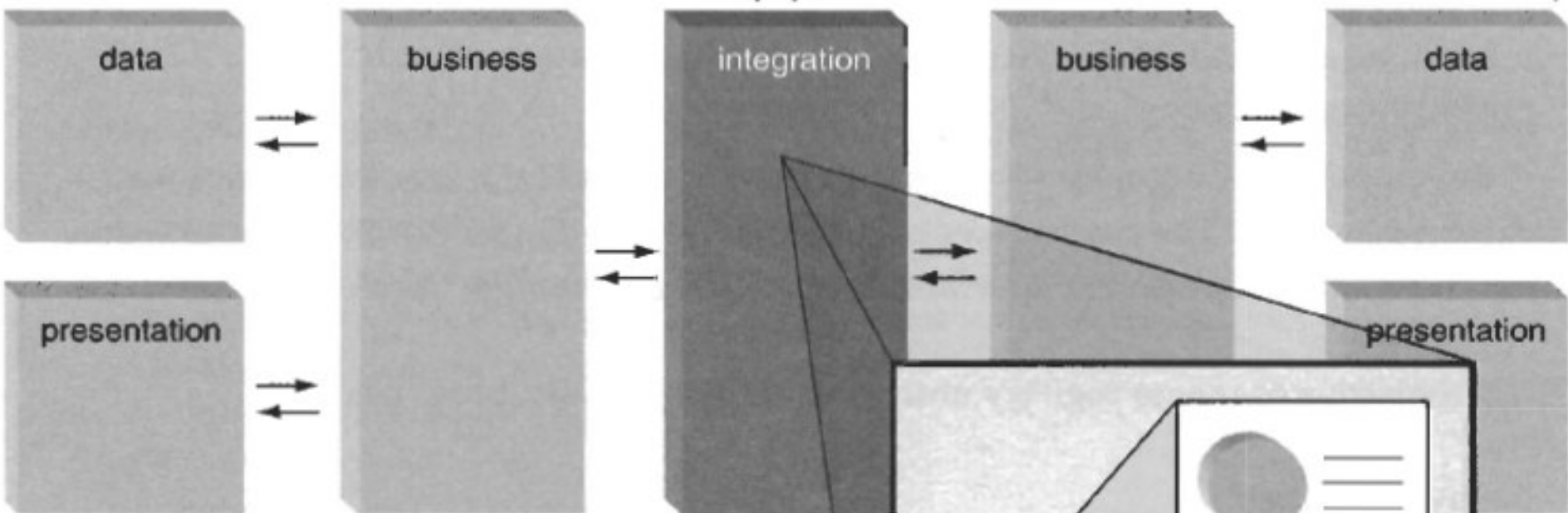
describe its
business and
its services

discover other
businesses that
offer desired
services

integrate with
services
provided by
other
businesses

application A

application B



```
<find_business xmlns="urn:uddi-org:api_v3">
  <findQualifiers>
    <findQualifier>
      uddi:uddi.org:findQualifier:exactMatch
    </findQualifier>
  </findQualifiers>
  <name>
    XMLTC Consulting Inc.
  </name>
</find_business>
```

UDDI
registry

service
descriptions

Web Services in Action with Core Standards

