SOA & Web Services

Contents:

- Motivation
- Service Model
- Web Services
  - SOAP
  - UDDI
  - WSDL
  - Composition
- Security (awareness)
- SOA Governance

Mariano Cilia
mcilia@gmail.com

Why SOA?

- Organizations need to quickly and effectively react to frequent changes in business requirements!
  - Business processes can be easily mapped to services
  - From monolithic to heterogeneous and "loosely-coupled"
- The SW architectural model provides many benefits:
  - flexibility, reusability, interoperability, …
  - Comprises loosely coupled, highly interoperable application services
  - Identification and categorization (discoverability)
- The model of SOA is here to stay!
  - It's an evolution

SOA Introduction

- SOA derives its technical strategy and vision from the basic concept of services
  - A service is an abstraction that encapsulates a software function
  - Services are most often designed to ignore the context in which they are used
    - It does not mean that services are stateless they are rather context independent
    - This is precisely the definition of "loosely coupled"
      - Services can be reused in contexts not known at design time
      - Be careful when interpreting data!
  - The SOA style promotes **reuse at the macro** (service) **level** rather than micro levels (e.g. Objects)

3

SOA Introduction (cont.)

- It is about a loosely coupled exchange of messages between services
- Loosely coupled across
  - Time
  - Location
  - Identity
  - Languages & environments
  - Platforms & middleware stacks

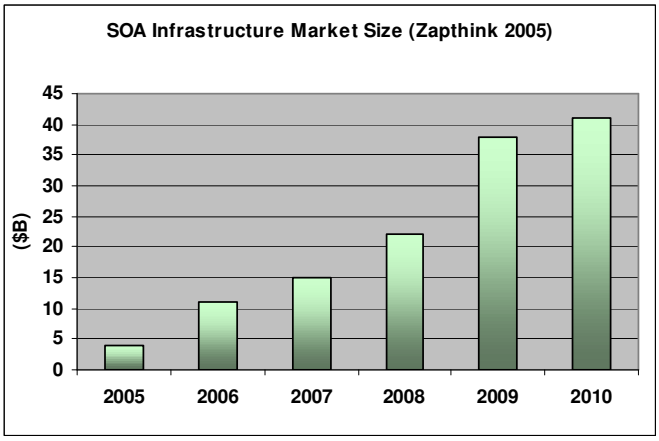
4

SOA Goals

- Reduce cost through reuse
  - Build applications faster
  - Use existing business logic rather than rewriting each time
  - Minimize cost of maintenance and upgrade by allowing incremental updates
- Increase agility to better align IT and the Business
  - Allow rapid change through business process management and composition tools
  - Allow incremental updates to enterprise applications
  - Minimize change cycles with business granular interfaces
- Reduce the risk, fragility and complexity of integration by improving interoperability through standards
  - Reduce investment in and risk of brittle proprietary integration techniques and technologies
  - Reduce frequency of data error caused by duplication

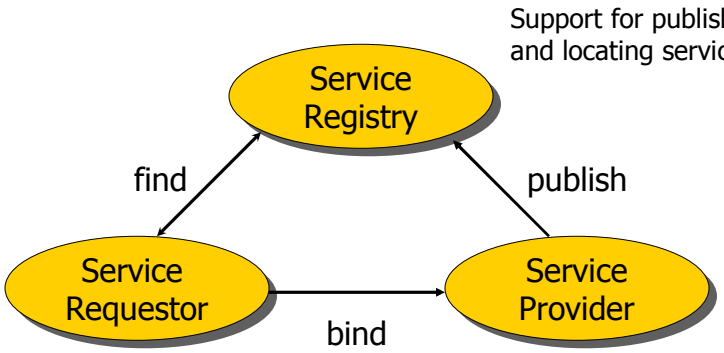
5

SOA Market



6

Service-oriented Architecture



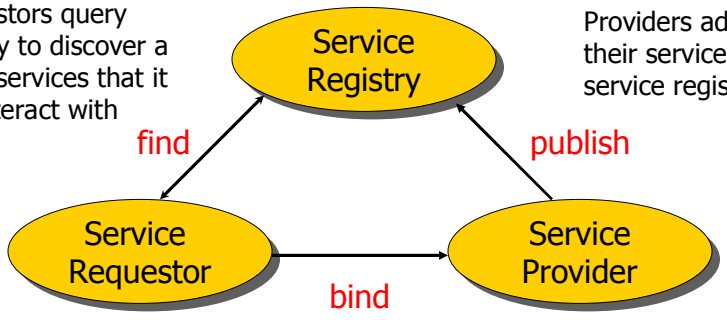
Locates required services via Registry. Binds to services via Provider

Publishes availability of these services through a registry

7

Web Service Operations

Requestors query registry to discover a set of services that it can interact with



Providers advertise their services to service registry

Requestors negotiate with providers to access and invoke services

8

SOA Benefits

- Supports vendor diversity
- Fosters intrinsic interoperability
- Promotes discovery
- Promotes federation
- Fosters inherent reusability
- Emphasizes extensibility
- Promotes organizational agility
- Supports incremental implementation
- (Based on open standards)

9

Until now

- the Web has provided for
  - Browsing of linked documents
  - Manually-initiated purchases and transactions
  - Downloading files
  - All of this is manual, by using a browser

10

A New Web Model

- Web services is a model for exploiting the Web (standards)
  - Transactions initiated automatically by a program, not necessarily using a browser
  - Can be described, published, discovered, and invoked dynamically in a distributed computing environment
  - New ways of using the Web:
    - Marketplaces, auctions, …
    - All built on XML

11

What is a Web Service?

- Self-describing, self-contained modular app
  - Standard description languages based on XML
  - Create open standards that are both programming language and platform independent
  - Make these standards robust enough to handle the job and yet simple enough to facilitate widespread adoption
  - Based on open standards for description, discovery & invocation
- Web Services
  - Introduces non-binary messaging
  - XML messaging
  - XML over HTTP solves the problem of distributed applications within and across companies
    - the firewall!
- Available to a variety of clients
  - Facilitates building distributed applications (at least less difficult)
  - Platform & implementation neutral

12

What is a Web Service? (cont)

- A natural extension to the C/S model
  - Invoke methods on remote objects
  - Directory of available services and service providers, using UDDI
- Evolution of e-Business
  - Publishing of business functions on the web
  - Universal access to these functions
  - Typically transactional, requiring integration with existing systems
- A programmatically available application logic exposed over the Internet
  - An interface that describes a collection of operations that are network accessible through standardized XML messaging
  - Enabling systematic application-to-application interaction on the Web

13

Examples of Web Services

- Business information
  - Weather, calendar, news, credit check, auctions, stock quotes, airline schedules, …
- Transactional (for B2B or B2C)
  - Airline reservations, rental car agreements, purchase order, supply chain mgmt
- Business Process Externalization
  - Business linkages at a workflow level
  - Allows complete integration at a process level

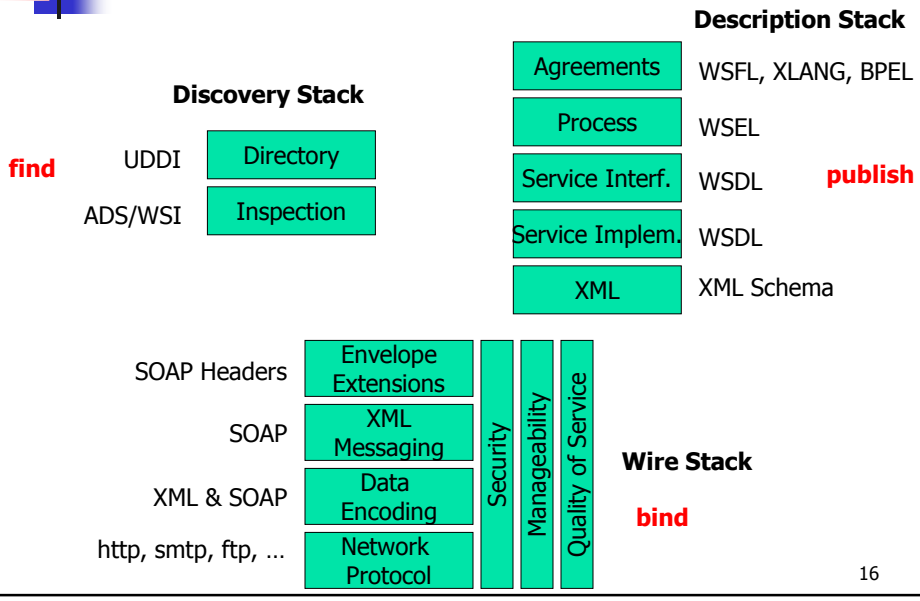
14

Infrastructure

- Need to communicate: XML
  - SOAP (attachments, security, reliable, …)
- Need to describe services: service specification language
  - WSDL
- Need to be able to find services: registry and advertisement/discovery
  - UDDI

15

WS – Interoperability Stack



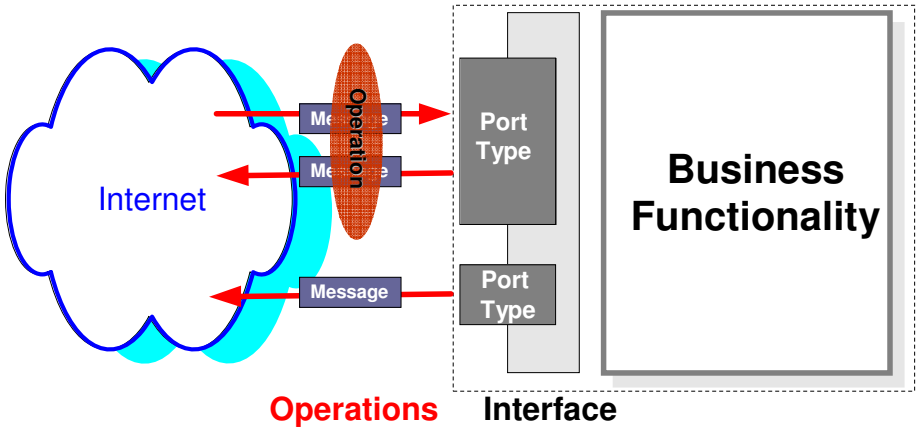
16

Web Services - Key Aspects

- SOAP (Simple Object Access Protocol)
  - XML-based messaging format
  - SOAP binds to application protocols
  - Both synchronous & asynchronous
- WSDL (Web Services Description Language)
  - XML-based language for describing Web Services, where to find them, how to access them, and what methods they accept
  - The API of the Web Service
- UDDI (Universal Description, Discovery and Integration)
  - A registry for Web Services (i.e. "yellow pages")
  - Publish business information and technical details of Web Services
  - Search for other Web Services and connect to them
  - Intended to help businesses find each other and make integration between partners easier

17

What is a Web Service?



18

Web Services Definition Language (WSDL)

19

WSDL

- An XML-based grammar for describing the capabilities of Web Services
- Extensible
- Similar in concept to IDL
  - IDL is platform dependent
  - WSDL is platform independent
- Contains operational information about the service

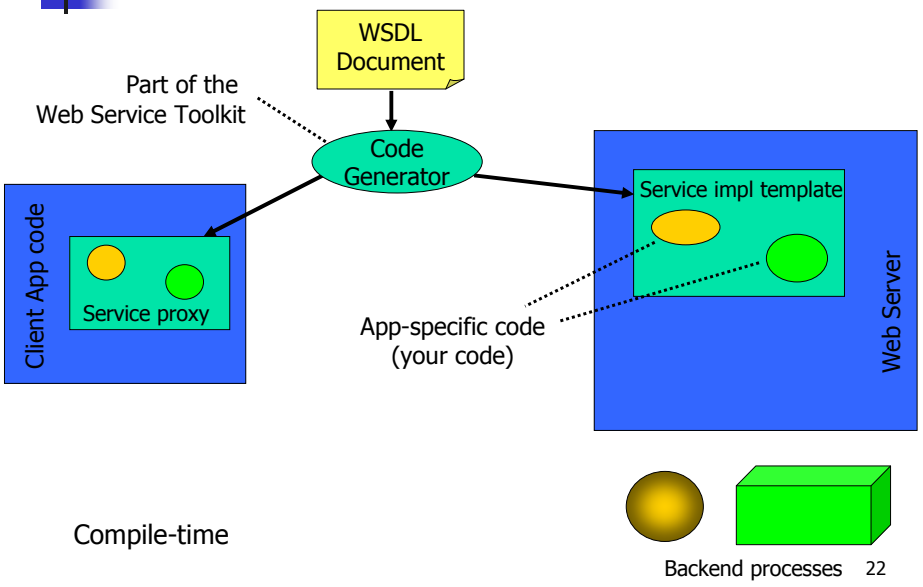
20

WSDL Usage Scenarios

- As IDL, it allows tools to generate client access code for a service
  - Web Service Toolkits
- Standardized service interface descriptions
  - Allows advertisement and dynamic discovery of service
  - Enables dynamic binding to service
  - Complements UDDI registry

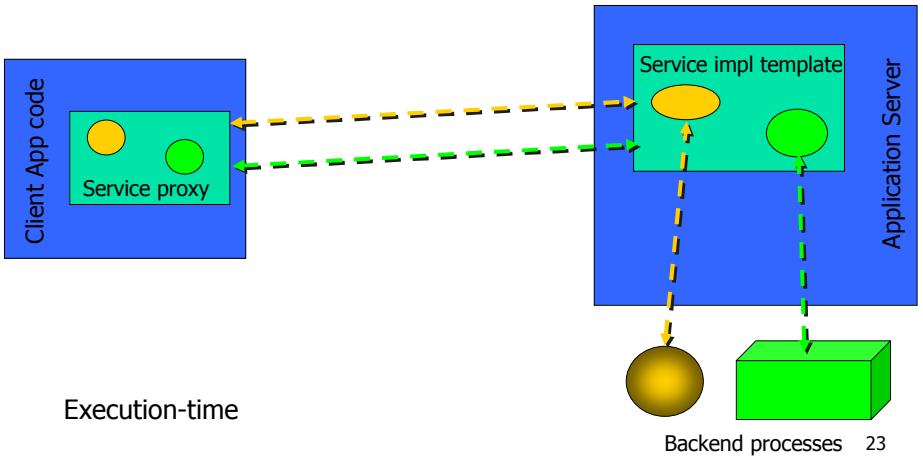
21

WSDL Coding



22

WSDL Coding



23

Service Definition Elements

- Types: provide data type definitions used to describe the msgs exchanged
- Message: represents an abstract definition of the data being transmitted
- PortType: set of abstract operations. Each op. refers to an input msg and output msg
- Binding: concrete protocol and data format specs for the operations and msgs defined by a particular portType
- Port: specifies an address for a binding, defining a single comm. endpoint
- Service: used to aggregate a set of related ports

24

WSDL Document: Two Parts

- Abstract part – independent of the specifics of any protocol
  - Service Interface
    - Abstract, reusable service definition
    - Published as tModel in UDDI registry
- Concrete part – defines how to "map" the abstract definitions onto the concrete protocols (defines bindings to many messaging protocols)
  - Service Implementation
    - Implementation of one or more service interfaces
    - Published as businessService in UDDI registry
- Several concrete definitions may reference one and the same abstract definition
- An abstract part may be in a separate file
  - Imported into the concrete definitions document

25

WSDL Service Interface

```
<?xml version="1.0"?>
<definitions name="StockQuoteService-interface"
...
 <message name="SymbolRequest">
   <part name="symbol" type="xsd:string"/>
 </message>
 <message name="QuoteResponse">
   <part name="quote" type="xsd:string"/>
 </message>
<portType name="StockQuoteService">
 <operation name="getQuote">
   <input message="tns:SymbolRequest"/>
   <output message="tns:QuoteResponse"/>
 </operation>
</portType>
<binding name="StockQuoteServiceBinding" type="tns:StockQuoteService">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getQuote">
   <soap:operation soapAction="http://www.getquote.com/GetQuote"/>
    <input>
      <soap:body use="encoded" namespace="urn:live-stock-quotes"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </input>
    <output>
       <soap:body use="encoded" namespace="urn:live-stock-quotes"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
    </output>
  </operation>
</binding>
</definitions>
```

26

WSDL Service Implementation

```
<?xml version="1.0"?>
<definitions name="StockQuoteService"
  targetNamespace="http://www.getquote.com/StockQuoteService"
  xmlns:interface="http://www.getquote.com/StockQuoteService-interface"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
<import namespace="http://www.getquote.com/StockQuoteService-interface"
location="http://localhost:80/services/sqs-interface.wsdl"/>
  <service name="StockQuoteService">
  <documentation>Stock Quote Service</documentation>
  <port name="localhost" binding="interface:StockQuoteServiceBinding">
    <soap:address location="http://localhost:8080/soap/servlet/rpcrouter"/>
  </port>
. . .
</service>
</definitions>
```

27

Simple Object Access Protocol (SOAP)
Service-Oriented Architecture Protocol

28

What is SOAP?

- Simple, lightweight XML protocol for exchanging structured and semi-structured information on the Web
- Exploit internet protocols and standards
- Sends XML-formatted messages over HTTP (or other protocols)
- Works with any programming language, object model, operating system, or platform

29

SOAP - Design Principles

- KISS
  - Can be implemented in a bunch of hours
- Modular and extensible
  - No application semantics and no transport semantics
- Vendor neutral
- Firewall friendly
- Flexible layering....substitutable:
  - Transport bindings
  - Language bindings
  - Data encodings

30

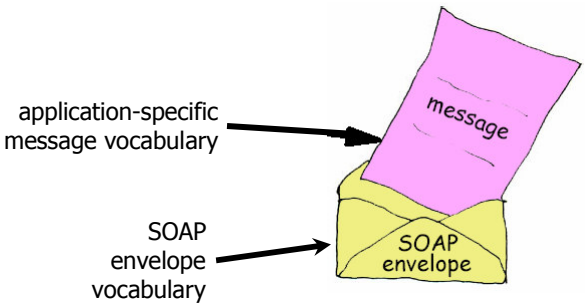
Usage Models

- SOAP RPC-like
  - Simple request/response protocol
    - Request invokes a method on a remote object
    - Response returns result of running the method
- SOAP Messaging
  - Send or process a SOAP message
  - May be a response to it (now or later)
  - Can be used for asynchronous processing

31

SOAP Message Structure

- Defines an envelope
  - The envelope wraps an XML message
  - The message has its own vocabulary
  - Each piece uses its own namespace



32

Parts

- Extensible envelope that express
  - What features and services are represented in a message
  - Who should deal with them
  - Whether they are optional or mandatory
- A set of encoding rules for data
  - Uniform model for serializing non-syntactic data models
- A convention for representing RPC
  - How to make calls and responses
- A protocol binding to HTTP

33

A SOAP Request Message

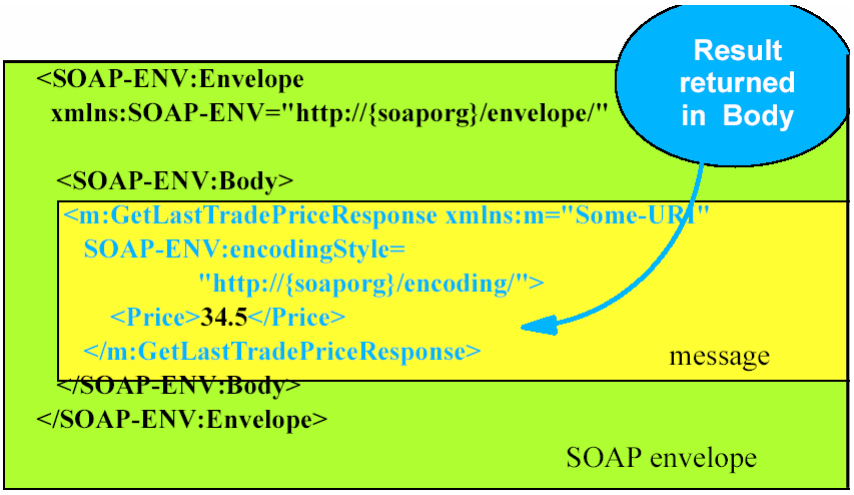
```
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV=
      "http://www.w3.org/2001/06/soap-envelope"
    SOAP-ENV:encodingStyle=
      "http://www.w3.org/2001/06/soap-encoding">

   <SOAP-ENV:Body>
        <m:GetLastTradePrice xmlns:m="Some-URI">
            <symbol>DIS</symbol>
        </m:GetLastTradePrice>
   </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

message

SOAP envelope

34

A SOAP Response Message



35

SOAP over HTTP - Example

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
       <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

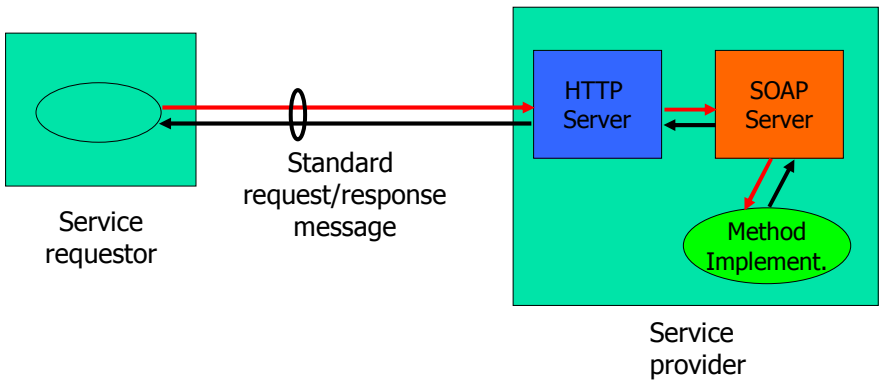
36

SOAP Encoding

- Spec defines a type system and a serialization of that type system to an XML form based on XML Schema
- Use of this encoding style is not required
- Serialization rules are reasonable
- use XML Schema as much as possible, augment where necessary (arrays)

37

SOAP in Action



38

SOAP is a Protocol!

- It's not a distributed object system
- It's not an RPC system
- It's not even a Web application
- Your app decides what your app is!
  - Tightly coupled vs. loosely-coupled
- You have to think about how to design your application

39

Why SOAP Will Succeed?

- Other distributed technologies failed on the Internet because they strongly coupled the endpoints
  - ✘ RMI - requires Java at each endpoint
  - ✘ CORBA - requires compatible ORBs at each endpoint
  - ✘ DCOM - requires Windows at each endpoint
  - SOAP is the platform-neutral choice
    - simply an XML wire format
    - places no restrictions on the endpoint implementation technology choices
    - implementations are free, some are open-source

40

Universal Description, Discovery and Integration (UDDI)

41

UDDI

- A registry for web services
  - Provides a way to publish and find information about Web Services
  - Semantic description is missing
- Helps to find a web service and its description
  - Search by business
  - Search by service type
- Advanced discovery features are under development
  - **Business discovery** is not yet supported
  - For example:
    - Geographical proximity, price, time zone etc. are not considered as search criteria
    - Give me all services implementing **Currency Converters** used in **Financial Applications** for **Accounting**

42

Public Registry Operators

- Peer nodes
  - M$, IBM, HP, SAP, Intel
  - Companies register with any operator
  - Registrations replicated on a daily basis
  - Complete set of registered records are available at all operator nodes
- Common set of SOAP APIs supported by all operators

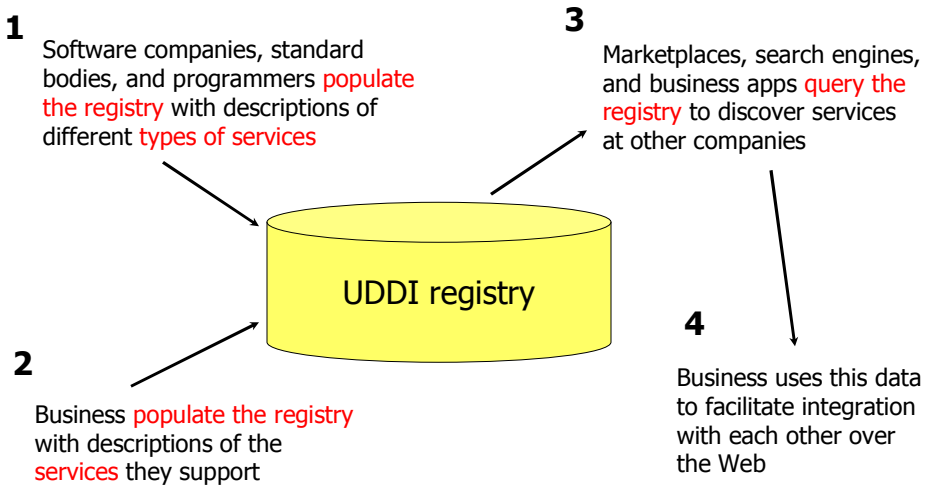
43

UDDI Private Registries

- Community-based registry
  - Within a business domain
  - Within a trusted community
- Extended to support
  - Privacy
  - Security
  - Data integrity
  - Reliability
  - Manageability
  - Richer query capabilities

44

Overview



45

What's in the registry?

- Programmers, SW companies, etc. register service types: specifications, taxonomies

Service types (tModels)

- Businesses register public information about themselves and their services

White pages

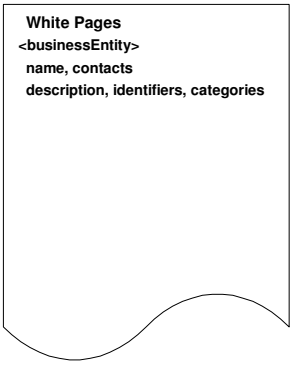
Yellow pages

Green pages

46

UDDI Structure - White Pages

- Hierarchical organisation of the registry
- Problem:
  - What information is logically the root - services or providers?
- The top level contains information about Web Service Publisher Organization
  - Organization = **businessEntity**
  - Top Level called **White Pages**
- Business Entities contain general information about a company
  - Business Name
  - Text Description
  - Contact information - names, phone, web sites…
  - Known Identifiers
    - List of identifiers for the publisher (organization)
    - D-U-N-S (UDDI registry generated unique number for each business - see category bags)



50

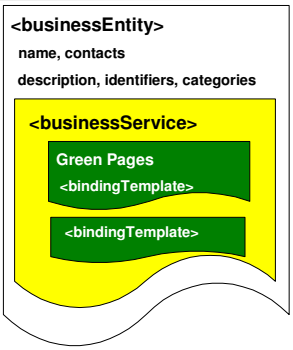
UDDI Structure – Yellow Pages

- Every organization can publish multiple services
  - Service = businessService element
- businessService object
  - Called Yellow pages
  - Contains a list of services of the same type
- The Yellow Pages contain categorization of Web Services
  - 5 standard taxonomies in Version 2.0 in 3 groups
    - Industry: NAICS (Industry codes - US Govt.)
    - Product/Services: Standard Industrial Classification, USPSC
    - Location: Geographical taxonomy (GGC, ISOGT)
  - Implemented as name-value pairs to allow any valid taxonomy identifier to be attached to the business white page

51

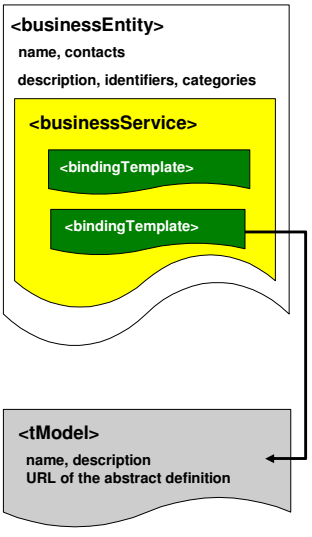
UDDI Structure – Green Pages

- The bindingTemplates
  - Called also Green Pages
  - Contain technical details about invocation of a Web service
  - Protocol - identify the transport and communication protocols to be used – HTTP, SMTP, SOAP …
  - Access point - the URL to access the service
- Each Yellow Page (businessService) contains one or more Green Pages / bindingTemplate elements
  - Every Web Service Port Type can be bound to many protocols



tModel

- A tModel contains
  - The location of the service interface (abstract definitions)
    - Pointer (URL) to the WSDL file created by the publisher and hosted by the publisher
  - A reference to the Yellow page categories
- tModel allows a developer to find out
  - Where the interface (WSDL) of a Web Service is located
  - Which Web Services implement this Interface (abstract definitions)
- Two different companies
  - can implement the same tModel,
  - but never the same bindingTemplate
- The UDDI registry automatically registers a taxonomy of a tModel if it does not exist

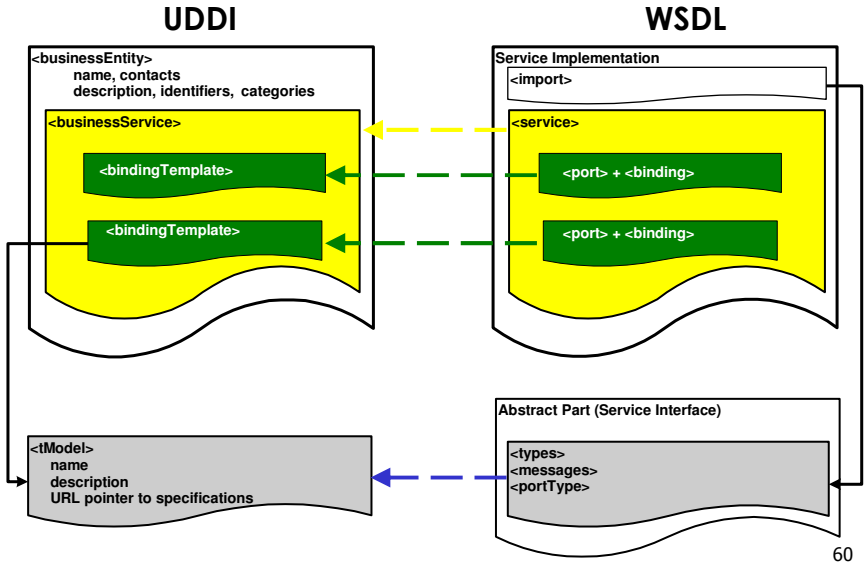


tModel

- Represents a technical model
  - Service type or specification type
    - Green pages
  - Categorization
    - Used for yellow pages
  - Identification
    - Used for white pages
- Generated by UDDI
  - tModelKey (unique identifier)

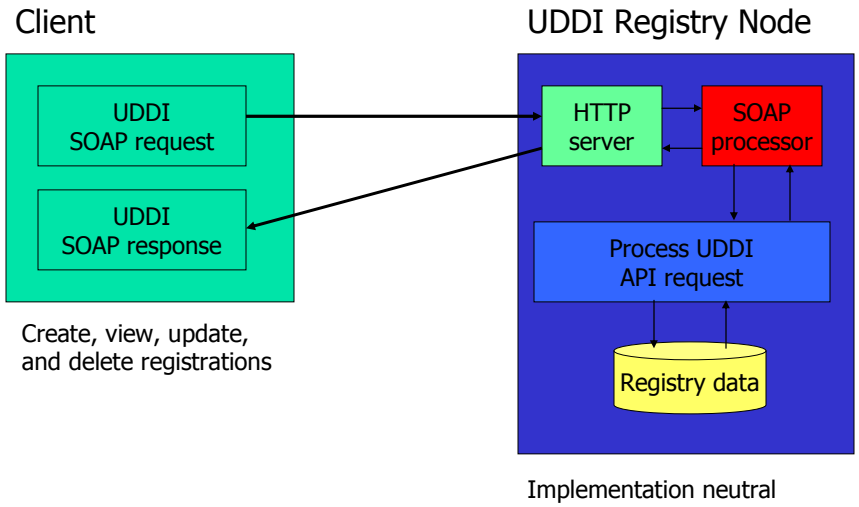
54

WSDL – UDDI Relation



60

UDDI & SOAP



Create, view, update, and delete registrations

Implementation neutral

61

Registry API

- Inquiry API
  - Find entries, get details about them
- Publishers API
  - Save, delete entry
- UDDI4J: Makes client coding easy
  - Implements UDDI operates as a Java class library
  - Open source
  - Available at oss.software.ibm.com

62

Web Services - Other Issues

- Reliability (HTTPR)
- Security (WS-Security)
  - Authentication
  - Privacy
  - Non-repudiation
  - Auditing
- Transactions (WS-Transactions)
- Business Process (BPEL4WS)
  - Completely dynamic e-business legal templates and automatic terms negotiation integration (in minutes or seconds)

63

Web Services and WS-Flows

Business Processes

Workflows

BPEL4WS

64

Business Processes

- Business process:
  - Collection of activities
  - Performed by human users or software applications
  - To achieve a particular business objective
- Process model - describes:
  - Structure of a real world business process
  - Actions to be performed
  - Rules, alternative paths
- Process Instance
  - The instance of the process model is a process
  - Is carried out according to the rules and paths of the model
  - Put into context
  - Humans and/or programs perform tasks
  - Executed not only on a computer

65

Business Processes (2)

- A business process specifies:
  - The potential execution order of operations (from a collection of applications)
  - The data to be shared
  - The partners' involvement in the business process
  - Joint exception handling
  - How multiple parties and services participate
  - Might reflect organizational structures
  - Might specify the exact application programs involved in the workflow
  - Enforce constraints, business rules …
  - …

66

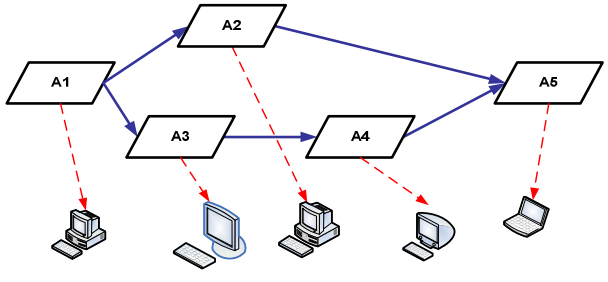
Workflows

- Parts of processes run on computers:
  - Called **workflows** (process instances)
  - Workflow model (process model)
- Workflow
  - executable description of a business process
- Executed by a Workflow Management System (WfMS)

67

What is a Workflow?

- A description of:
  - Tasks to be performed
  - By participants (human, programs, resources)
  - In a certain order (sequential, parallel, etc.)
  - Taking into account alternative paths → control flow
  - Data to be exchanged → data flow
  - Work items assigned to participants



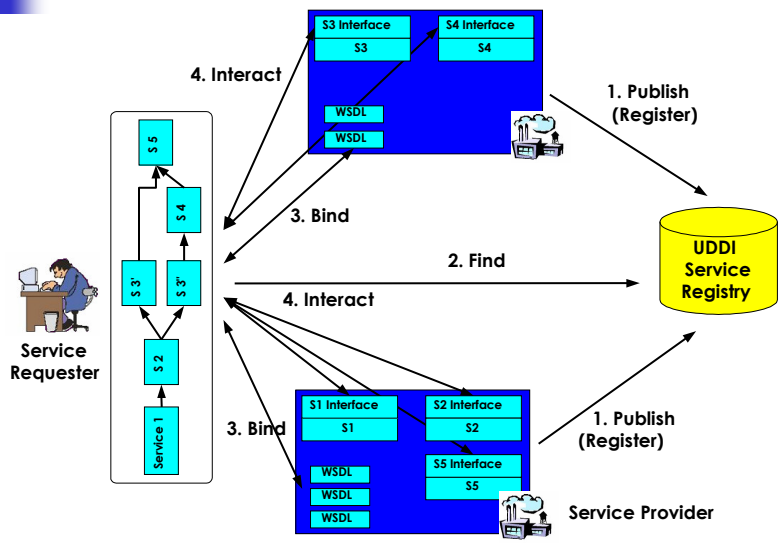
68

Application of Workflow

- Image/document processing
- Groupware – Coordination of team work
- Project Management
- Manufacturing
- Supply chain management (SCM)
- Customer relationship management (CRM)
- …

69

Workflows and Web Services



70

BPEL4WS

- BPEL4WS = Business Process Execution Language for Web Services
  - BPEL4WS is a flow language
  - Microsoft and IBM
  - A combination of WSFL and XLang, replacing them
    - WSFL – support for graph oriented processes
    - XLang – structural constructs of processes
- BPEL4WS describes business processes
  - XML based Web Service Flow definition language
  - A process is a combination of activities, involving multiple WS
  - The process itself is a Web Service, too
- Goal
  - Standard language for Web Service compositions

71

BPEL4WS Concepts

- BPEL4WS Business Process Elements
  - Process
  - Participants, relationships
  - Activities
    - Simple
    - Complex
  - Messages, Containers/Variables, Correlation data
  - Fault handling and compensation
  - Business protocols and views
- All elements are put together in two groups
  - Two XML documents are needed to deploy a BPEL4WS Process
    - WSDL document – interface description
    - BPEL document – process description

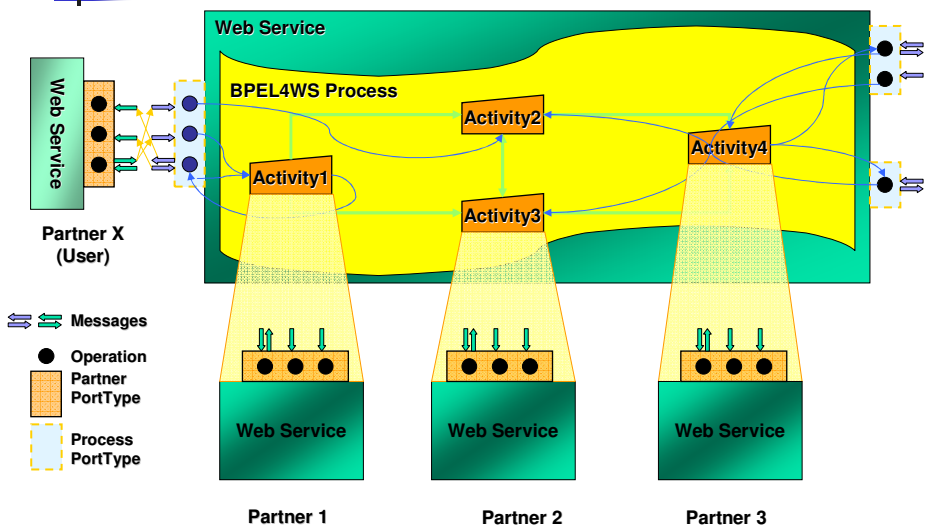
72

BPEL4WS Processes

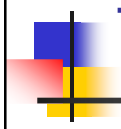
- BPEL4WS Process – a composition of Web services
  - Execution order of activities
  - Process's and partners' roles within an interaction
  - Data flow
  - Control flow
  - Error Handling
  - Compensation activities
- Process interface description – Web service interface
  - A WSDL document
  - portTypes and operations the process exposes
  - References to partners' Web services taking part in the interaction
  - Types of relationships to partners (ServiceLinkTypes)
- Executable and abstract BPEL4WS processes

73

BPEL4WS Process



74



Towards SOA

79

7 steps to SOA

- Create/Expose Services
  - Re-build or expose
  - Establish granularity (be useful to multiple different apps)
- Register Services
  - Browse and search capabilities
  - Organization
- Secure Services
  - auth*, privacy, non-repudiation, auditing
- Manage/Monitor Services
  - Monitor services, compositions, SLAs, disasters,…
- Mediate and Virtualize Services
  - Versioning, increase capacity,
  - Virtualize: proxy for routing, load-balancing, transforming, mediation, bridging, …
- Govern the SOA
- Integrate Services (ESB)

80

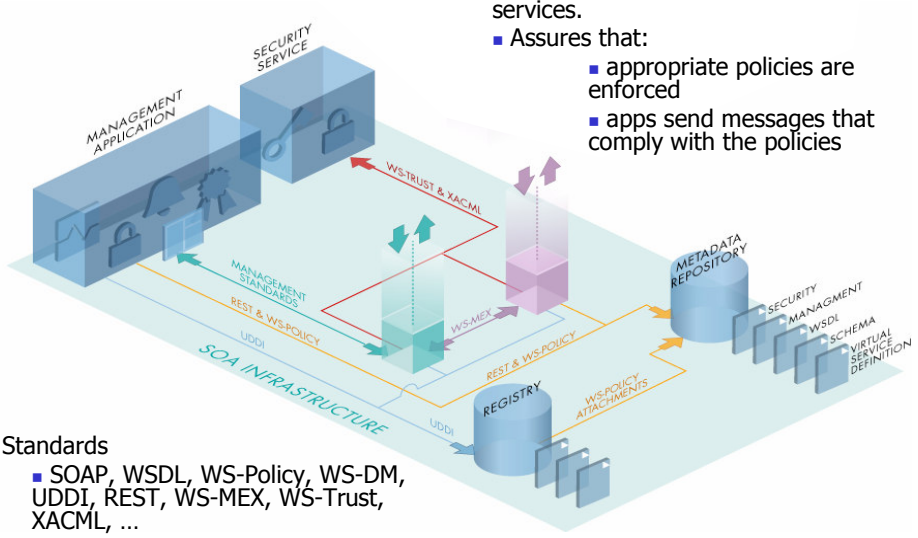
SOA Infrastructure (SOI)

- SOA Infrastructure provides core infrastructure services to the SOA and XML applications and messaging layer
- Infrastructure services include:
  - Management Application
    - Implements management standards like WS-DM to provide central performance and health monitoring and reporting capabilities
  - Security Service
    - Implements standards like WS-Trust and XACML as well as common PKI features
  - Registry
    - UDDI services for core service discovery
  - Metadata Repository
    - Serves policies, WSDLs, Schema, virtual service definitions and many other key meta-data items
- Service providers, consumers, enterprise service bus platforms along with other service proxies, leverage these infrastructure services

81

SOI

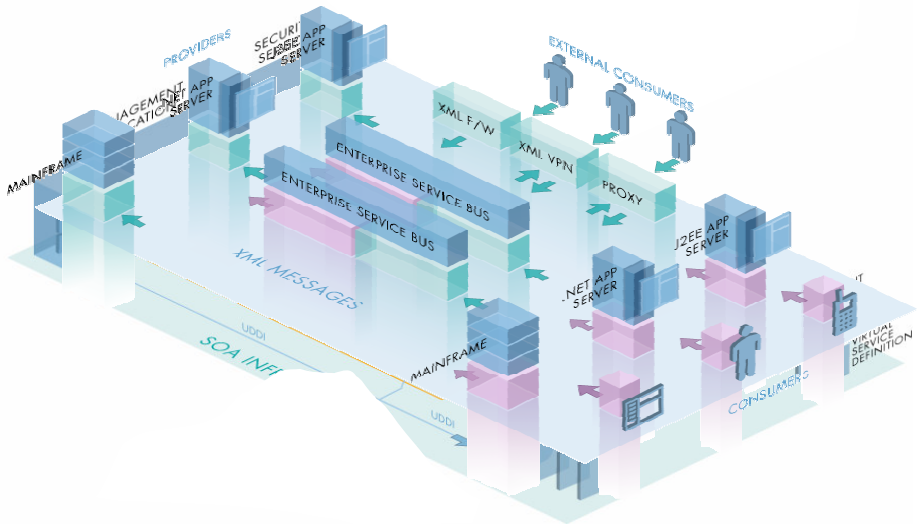
- Provides security, management, mediation, governance and monitoring services.
- Assures that:
  - appropriate policies are enforced
  - apps send messages that comply with the policies



- Standards
  - SOAP, WSDL, WS-Policy, WS-DM, UDDI, REST, WS-MEX, WS-Trust, XACML, …

SOI in context

- App Servers, ESBs, BPMs reside on SOI
- Apps & services expose I/F



Security issues in service oriented architectures

84

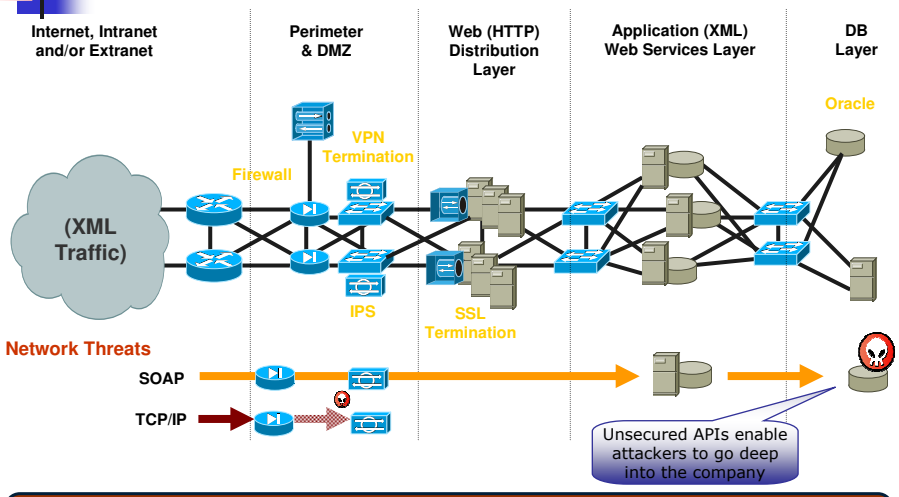
Why looking at SOA, Why Security?

- **SOA is a huge success…**
  - **71% of the companies** have already invested in SOA, 85% is predicted for 2008 (Source: IDC 2006)
  - **Microsoft** just launched their SOA stack called Windows Communication Foundation (March 2007)
  - **Sun** released Java 6, the first version to include a SOA stack (Dec 2006)
- **B2B integration is becoming a reality**
  - $7000B will be spent on B2B transactions in 2007 (i.e. 45% of the total) (Source IDC)
- **SOA simplifies B2B but also exposes a bigger attack front**
  - Shared APIs allow partners and attackers to access the core business apps.
  - Automated attacks are easier than ever
    - Web Services Description Language (WSDL)
    - Universal description, discovery, and integration (UDDI)
    - Off the shelve software stacks are everywhere
  - 75% of hacks occur at the Application/Service level (Source: Gartner)
- **WS-Security is to SOA what SSL is to HTTP (albeit at different rates ☺)…**
  - WS-Security is the **only standard way to secure SOA**
    - backed by the big guys (IBM, MS, Verisign) at OASIS (2006)
    - WS-Security adoption doubled in 2005

WS-Security is needed to support SOA growth…

Anatomy of the SOA Security challenge

The need for content based security



Perimeter defense is not enough, WS-Security can help with data integrity and authentication

SOA Governance

[From SOA Software]

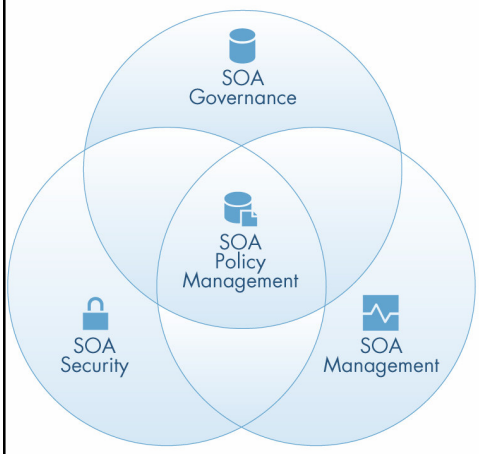
87

SOA Governance

- Lifecycle-related Concerns
  - Measure and prove the business value of SOA to offset cost concerns
  - Control the proliferation of duplicate or otherwise unnecessary services
  - Facilitate the identification of appropriate services by potential users to reduce initial development cost
  - Manage the lifecycle of services to minimize the cost and risk of ongoing maintenance and change
  - Simplify the actual USE of appropriate services (decoupling location, transport, policy, standards, messaging styles)
- Operations-related Concerns
  - End-to-end security - trust and protect the privacy of message senders, receivers, and content
  - Identify, manage, and repair exceptions as they occur
  - Reliability and performance of a distributed set of services and consumers
  - Interoperability between different platforms and technologies
  - Decoupling of services and consumers

88

Integrating SOA governance…



- Integrated SOA Governance is the combination of Registry/Repository, Management and Security with a comprehensive SOA Policy Management solution
- Governance offers value when integrated with runtime enforcement and feed back and compliance data
- Runtime solutions (security and management) offer maximum value with central policy control and service governance capabilities

[SOA Software]

89

The difference between Management and Governance

- SOA Management
  - Focuses on whether the systems and processes (machines) in your SOA environment are behaving correctly relative to operations policies that have been specified.
  - Are the machines behaving correctly?
  - Use Cases:
    - End-to-end security - trust and protect the privacy of message senders, receivers, and content
    - Identify, manage, and repair exceptions as they occur
    - Reliability and performance of a distributed set of services and consumers
    - Interoperability between different platforms and technologies
    - Simplify the actual USE of appropriate services (decoupling location, transport, policy, standards, messaging styles)

90

The difference between Management and Governance

- SOA Governance
  - Focuses on whether the people, policies, processes and metrics supporting your SOA program (people AND machines) are producing the desired business outcomes.
  - Is the organization doing the right things?
  - Use Cases:
    - Facilitate the identification of appropriate services by potential users to reduce initial development cost
    - Manage the lifecycle of services to minimize the cost and risk of ongoing maintenance and change
    - Provide Governance Automation through workflow, approvals, notifications and RBAC.

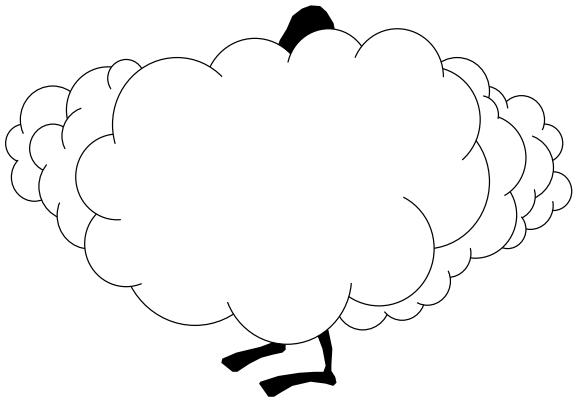
SOA – Wrap up

- SOA is the model…
  - Web services an incarnation
  - …and there are many others…
- Business facilitator
  - Agility
  - Reuse
- Web Services exclusively focus on
  - Interface description
  - Search & discovery
  - Protocol independent method call over the Web
- Compositions of Web Services are possible (see BPEL)
- Enhanced concepts – enterprise strength
  - Composition (Web Service Flows, Processes)
  - Web Service Transactions
  - Security (authentication, encryption, etc.)

Web Services – Wrap up

- Constructing application-to-application Web services is simple, in principle
- SOAP, WSDL, and UDDI are
  - Unavoidable frameworks for XML communication and service description
  - Complicated and their benefits are not always obvious, although the basic ideas behind them are great
  - Still under development…
- Performance penalties
- UDDI not widely adopted
- Many specifications around it
  - Business Process, Security, Reliable Messaging, Transaction, Pub/Sub, Management,…

93



94