

Web Services

Contents:



- Motivation
- Service Model
- SOAP
- UDDI
- WSDL

Dimka Karastoyanova & Mariano Cilia
Cilia@informatik.tu-darmstadt.de

1

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

2

A New Web Model

- Web services is a new model for using the Web
 - 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

3

What is a Web Service?

- An interface that describes a collection of operations that are network accessible through standardized XML messaging
- A programmatically available application logic exposed over the Internet

4

What is a Web Service? (cont)

- Available to a variety of clients (platform independent)
 - Facilitates building distributed applications (at least less difficult)
- Self-describing, self-contained modular app
- Platform & implementation neutral
- Based on open standards for description, discovery & invocation
- Typically transactional, requiring integration with existing systems

5

What is a Web Service? (cont)

- Evolution of e-Business
 - Publishing of business functions on the web
 - Universal access to these functions
- A natural extension to the C/S model
 - Invoke methods on remote objects
 - Directory of available services and service providers, using UDDI
 - Standard description languages based on XML

6

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

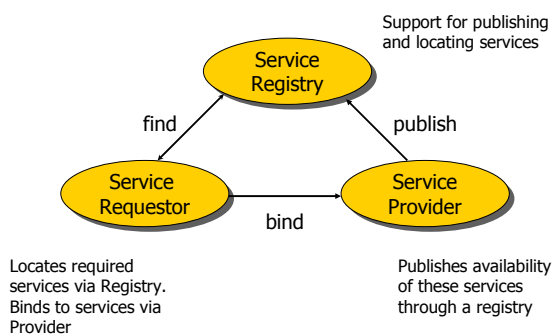
7

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

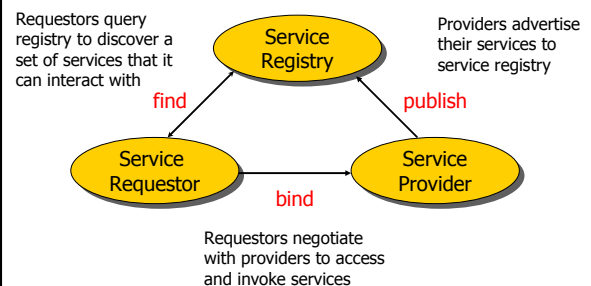
8

Service-oriented Architecture



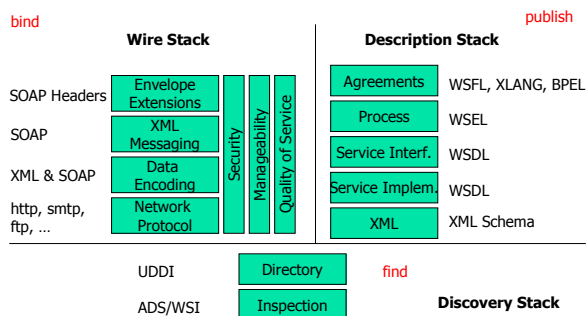
9

Web Service Operations



10

WS – Interoperability Stack



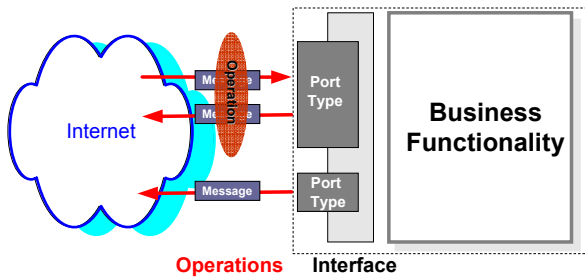
11

Web Services: Base Technology

- SOAP
 - An XML protocol to invoke a "function" on a server to perform a given operation
 - Request msg is sent by requestor
 - Response msg may be sent by provider
 - May be an asynchronous msg (notification)
- UDDI
 - Servers act as a directory of available services
- WSDL
 - An XML vocabulary to describe service interfaces

12

What is a Web Service?



13

Web Services Definition Language (WSDL)

14

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

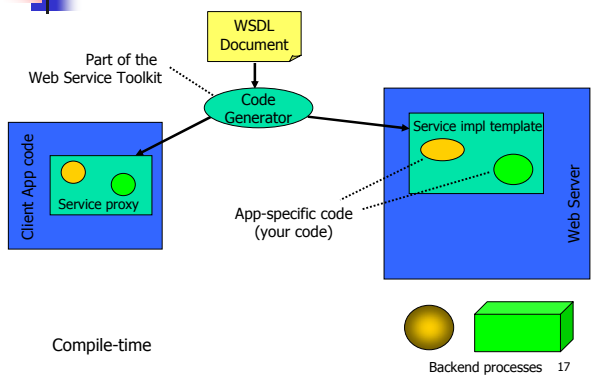
15

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

16

WSDL Coding



17

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

19

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

20

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>
```

21

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>
```

22

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

23

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

24

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

25

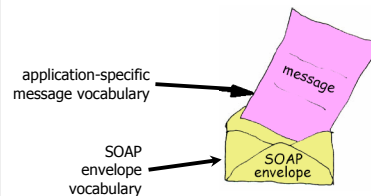
Usage Models

- 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

26

SOAP Message Structure

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



27

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

28

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>
```

SOAP envelope

29

A SOAP Response Message

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://{soaporg}/envelope/"
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI"
      SOAP-ENV:encodingStyle=
        "http://{soaporg}/encoding/">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP envelope

Result returned in Body

message

30

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>
```

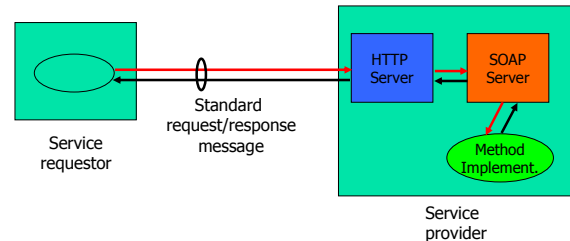
31

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)

32

SOAP in Action



33

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

34

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

35

Universal Description, Discovery and Integration (UDDI)

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**

36

37

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

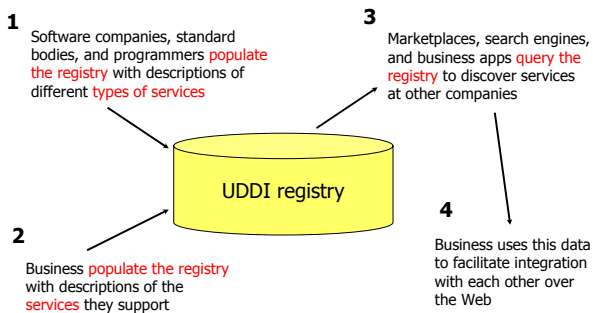
38

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

39

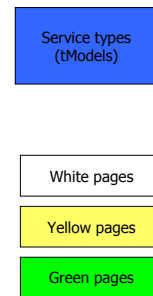
Overview



40

What's in the registry?

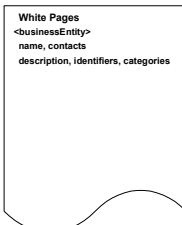
- Programmers, SW companies, etc. register service types: specifications, taxonomies
- Businesses register public information about themselves and their services



41

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)



45

UDDI Structure - Yellow Pages

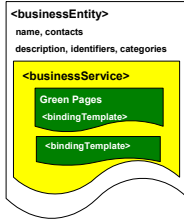
- Every organization can publish multiple services
 - Service = businessService element
- businessService object
 - Called Yellow pages
 - Contains a list of services of the same type
- The Yellow Pages contain categorization of Web Services
 - 5 standard taxonomies in Version 2.0 in 3 groups
 - Industry: 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



46

UDDI Structure – Green Pages

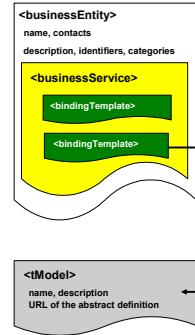
- The bindingTemplates
 - Called also Green Pages
 - Contain technical details about invocation of a Web service
 - Protocol - identify the transport and communication protocols to be used – HTTP, SMTP, SOAP ...
 - Access point - the URL to access the service
- Each Yellow Page (businessService) contains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



47

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



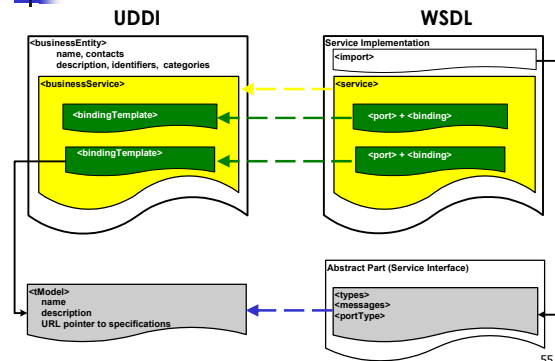
48

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)

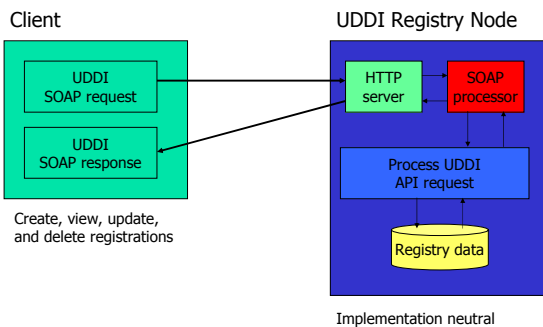
49

WSDL – UDDI Relation



55

UDDI & SOAP



56

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

57

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)

58

Web Services and WS-Flows

Business Processes
Workflows
BPEL4WS

59

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

60

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 ...
 - ...

61

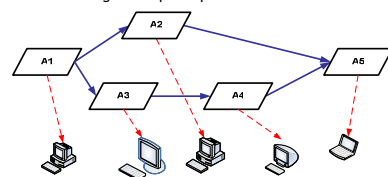
Workflows

- 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)

62

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



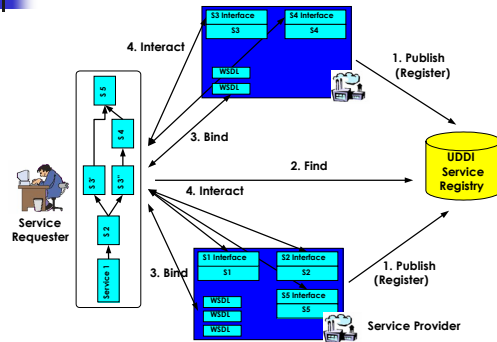
63

Application of Workflow

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

64

Workflows and Web Services



65

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

66

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

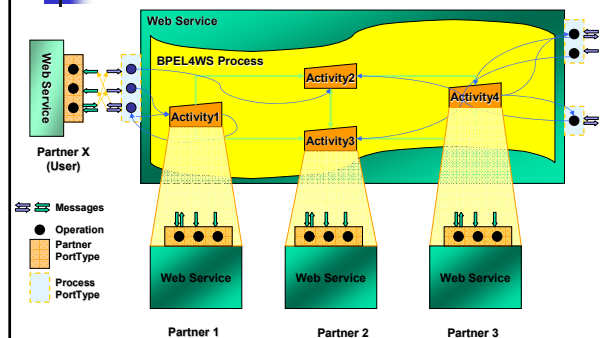
67

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

68

BPEL4WS Process

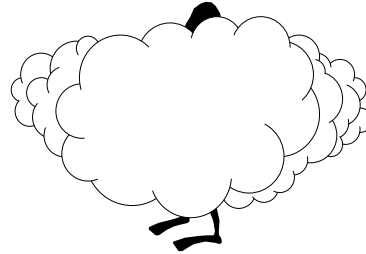


69

Wrap up

- Web Services exclusively focus on
 - Interface description
 - Search & discovery
 - Protocol independent method call over the Web
- Compositions of Web Services are possible
- Enhanced concepts – enterprise strength
 - Composition (Web Service Flows, Processes)
 - Web Service Transactions
 - Security (authentication, encryption, etc.)

74



75