



Enabling Grids for E-sciencE

PPARC Summer School, May 2005

# Web Services & WSRF Introduction

Richard Hopkins, National e-Science Centre, Edinburgh

www.eu-egee.org





## Goals –

- An Appreciation of the role and context of
  - Web Services in general
  - The Web Services Resource Framework
  - Java-based tools
  - Structure of Standards

## Structure

- Web Services and the Resource Framework
- Outline of JAVA tools and their use
- Standards Structure



# **Evolution of Electronic inter- enterprise interaction**

Web Services is the next step in the automation of inter-enterprise interaction

#### **Web Browsing**

Human travel agent provides "organise holiday" service by surfing the web to look for and invoking services – book a hotel; book a plane; book a car hire;
 ....; confirm bookings of best options to meet client needs.

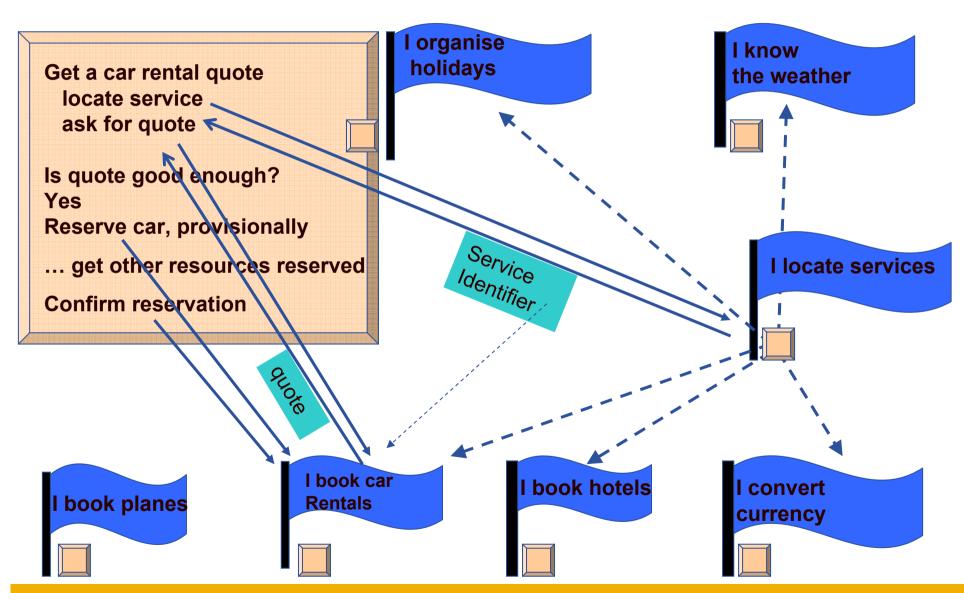
#### **Web Services**

- The aspiration of Web services is to provide a framework that allows that same model to be used in writing an application –
- which is itself becomes an "organise a holiday" service, finding and using useful services

Mode	human intervention at -	
	service provider	service consumer
E-mail	Yes	Yes
Web browsing	No	Yes
Web Services	No	<u>No</u>



## **Service Interaction**





## **Essential Requirements**

- Need to achieve effective cooperation even though
  - the different services are
    - produced by different organisations,
    - without any design collaboration,
    - on different platforms
      - Requires "interoperability"
  - the services are autonomously evolving
- Requires Loose coupling



## **Loose Coupling**

- Loose coupling minimum prior shared information between the designer of the two components of an interaction
  - Dynamically accessible Machine processable Meta data
    - Self-describing data in standard format XML documents
    - Description of structure of communications SCHEMAS (types)
    - Service description WSDL, using SCHEMAS for messsage stucture
    - Means for obtaining it from a repository, using standard such as UDDI
    - Communication protocol that supports this SOAP
  - Everything is a SCHEMA-described XML document soap message,
     WSDL definition, schemas themselves (meta-schema)
  - Tolerance of partial understanding
    - Schemas allows extension points one participant may have an older WSDL definition which accommodates extensions with additional information



## Web Service Statefull Resources

Reserving itcreates a **I** locate reservation entity: Res.22 services Get a car rental quote can be referenced in locate service future interactions ask for quote Service Identifier i.e. a reservation is a book car "WS statefull resource" quote Rentals Managed by a service Reserve a car. • Res.22 **Uniquely identified** reserved provisionally May be multiple instances Might notify change of state -**Confirm reservation** cancelled the car rental service cancels a reservation WSRF is a collection of emerging standards for interacting with WS (stateful) resources -Handle cancellation Naming: State – defining structure; accessing; modifying; notifying changes



- Service-Orientation A service is a
  - S/W system designed to support interoperable machine-to-machine interaction over a network. (W3C Glossary)
- Has some of the characteristics of O-O architecture
- The O-O class roughly corresponds to a PortType (i.e. Interface)
  - a collection of operations each with defined input and ouptput data
- Object roughly corresponds to either -
  - a Service an instantiation of a PortType
    - at a particular web location
    - using a particular communication protocol and message representation
  - a resource within a service.
    - A closer correspondence
      - Multiple instances with the same interface, but different data
      - Dynamically created and destroyed by service user
      - Has defined state
- But (a) Less constrained than O-O model
  - (b) Focussed on loose coupling



# Coupling

- COUPLING about intensity of communication
  - Execution Coupling -
    - Frequency and extent of communication relative to processing
    - telephone conversation is tightly coupled, e-mail conversation is loosely coupled
  - Design Coupling -
    - How much design knowledge has to be communicated between the designers of the software at the two ends of an interaction
    - Which they then build into their software
    - The extent of statically shared knowledge between two ends of an interaction



# A scale of coupling

- A scale of looser coupling (in both senses)
- Shared variable, interaction is
  - One end updating a variable; other end using it
- Object-Oriented
  - One end invoking method; other end being invoked
- Web Services
  - One end (service consumer) requesting a service
  - Other end (service provider) servicing the request
  - But may be other patterns, e.g. provider notifies consumer



# **Close / Medium Coupling**

Enabling Grids for E-sciencE

## **Shared Variable Model - Close coupling**

- The programmers of user side of an interaction know all about representation
- Shared implementation
- Suitable for single-programmer level
- Interaction is of order of a nanosecond
- Fine granuality
  - almost no work in a variable assignment
  - Simplest of tasks involves many interactions with variables
- Object Oriented Model Medium Coupling
  - User side of interaction knows what classes exist and their interface
     But not their representation
  - Shared class design
  - Suitable for single-organisation level
  - Interaction of order of micro/milli-sec (possibly distributed objects)
  - Medium granuality do some work in a method invocation 20 lines of code
  - Within an object, typically use the Shared Variable model



# **Loose Coupling**

**Enabling Grids for E-sciencE** 



## **Shared Variable Model - Close coupling**

- Design Coupling: shared implementation; single-programmer level
- Execution Coupling: nanosecond interaction; fine granuality;

#### **Object-Oriented Model – Medium Coupling**

- Design Coupling: shared Class design; single organisation level
- Execution Coupling: micro/milli-sec interaction; medium granuality

#### Web Services - Loose coupling

- Programmers on user side knows how to program the discovery of a service
- Shared standards and knowledge of standard repository
- Interaction of order of a second
- Coarse granuality do enough work in a service request to justify the time taken by the communication overhead
- Within a service, typically use the Object-oriented model service requestresponse is mapped to method invocation-return

#### Progressively –

- Looser design coupling; more time-expensive interaction, coarser granuality
- Each model builds on the previous one uses it internally

## Goals –

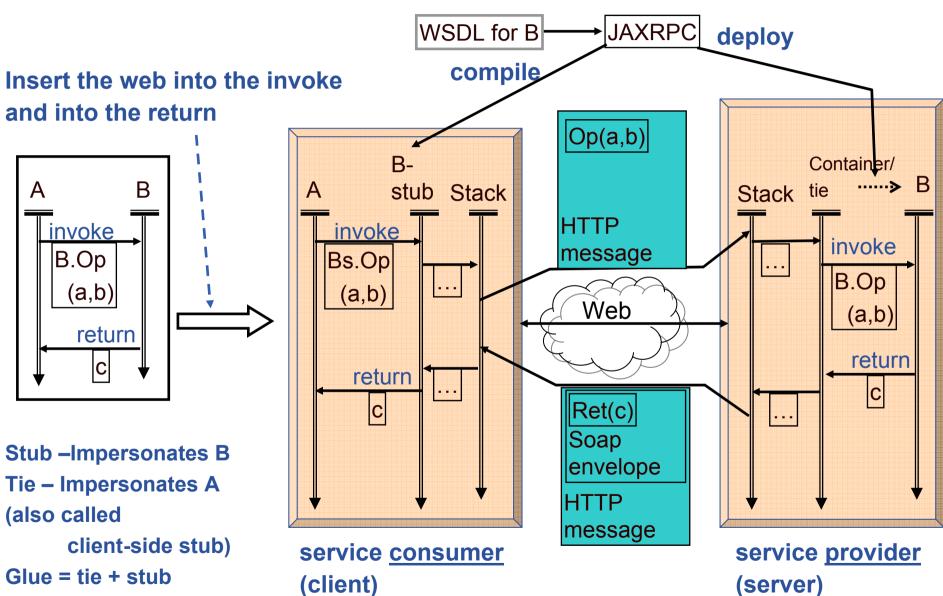
- An Appreciation of the role and context of
  - Web Services in general
  - The Web Services Resource Framework
  - Java-based tools
  - Structure of Standards

## Structure

- Web Services and the Resource Framework
- Outline of JAVA tools and their use
- Standards Structure



# O-O (JAVA) Web Services





## WSDL – how and when

- WSDL is keystone of web services
  - Defines a service's abstract interface operations and message structures
  - Defines the **physical** protocol and a data formants used to realise that abstract interface
  - The WSDL is published
  - Service consumer uses the WSDL to determine how to communicate with the service – stub generation
- There are different approaches to how WSDL is
  - Produced
    - Automatically
    - Manually
    - Semi-automatically
  - Consumed
    - Static Binding access the WSDL at compile-time
    - Dynamic Binding access the WSDL at run-time



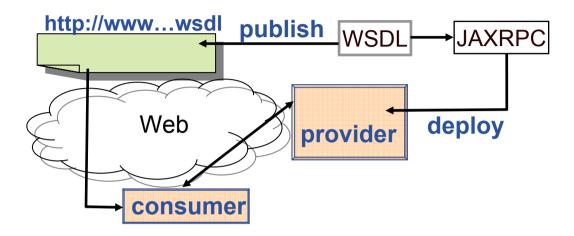
# **WSDL Producer Options**

Manual - Directly write WSDL

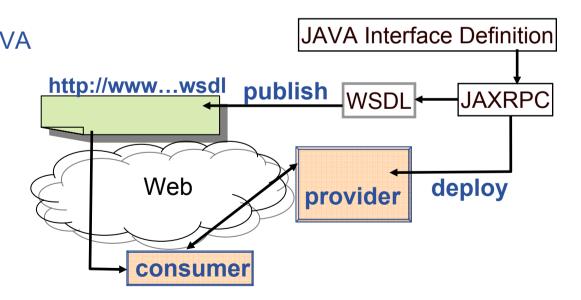
Generate JAVA from that

Unnatural for JAVA developer

WSDL is hard to write from scratch



Automatic - Derive WSDL from JAVA
Natural for JAVA developer
May compromise interoperability
too JAVA specific



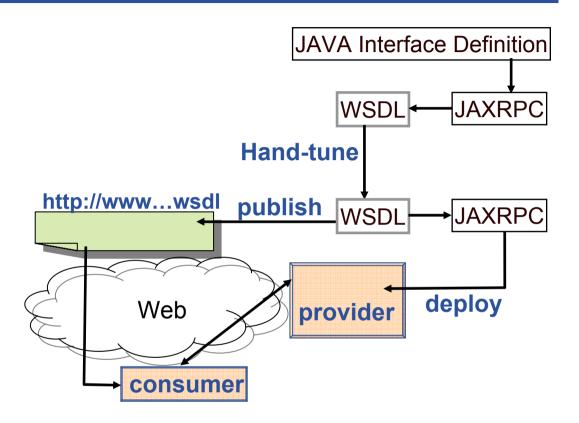


# **WSDL Producer Options**

**Enabling Grids for E-sciencE** 

#### Semi-Automatic -

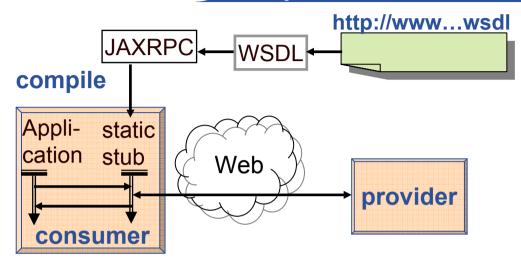
Start with JAVA definition
Derive initial WSDL
Modify it for
interoperability
extensibility
deploy (, test) and publish that
Natural for JAVA developer
Addresses interoperability
and extensibility

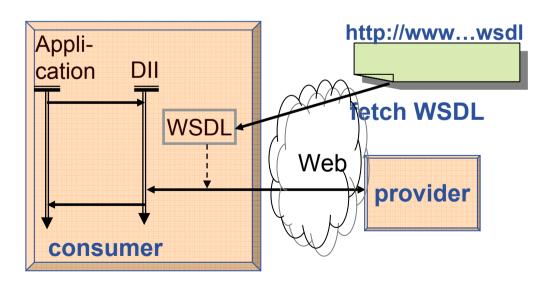




## **WSDL – Consumer Options**

Enabling Grids for E-sciencE





### **Static Binding -**

Manually obtain WSDL

Use it to compile a stub

Which interacts with the service

Need to recompile if WSDL changes

Can test before going live

## **Dynamic Binding -**

Via DII – Dynamic Invocation Interface

On each use of service,

get the WSDL

Use that to

construct message to the service

interpret the rsponse message

Always using latest WSDL

But un-tested

Possibly inefficient

## Goals –

- An Appreciation of the role and context of
  - Web Services in general
  - The Web Services Resource Framework
  - Java-based tools
  - Structure of Standards

## Structure

- Web Services and the Resource Framework
- Outline of JAVA tools and their use
- Standards Structure



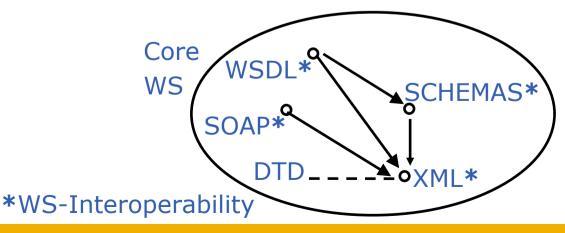
# **Evolving Standards**

- Collaboration is on defining generic standards, not specific design
- Two main standards bodies
  - W3C web community
    - actually produces "recommendations" not standards
  - OASIS industry IBM, Microsoft, Sun, ....
- These standards are factored to allow partial adoption and combination
  - The core standards
  - WS-I clarifications to aid interoperability
  - Higher level standards built on them
- Take-up is pragmatic
  - some standards ignored,
  - some pre-standards taken as stadndards
- Standards have built-in extensibility
  - A standard typically defines some document -
    - E.g. The WSDL document for defining the interface to a service
    - The definition of the WSDL structure incorporates specific points where that structure can be extended





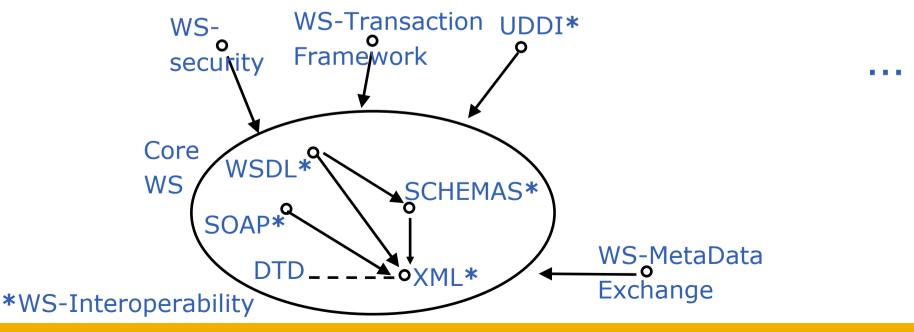
- XML the standard format for all information
- SCHEMA the standard language for defining the structure (syntax/type)
  of a unit of information
  - **DTD** is a deprecated predecessor of Schemas
- WSDL the language for defining a service
  - Operations; Logical Message Structure; Bindings; locations
- SOAP the standard message format





## **Some Further Standards**

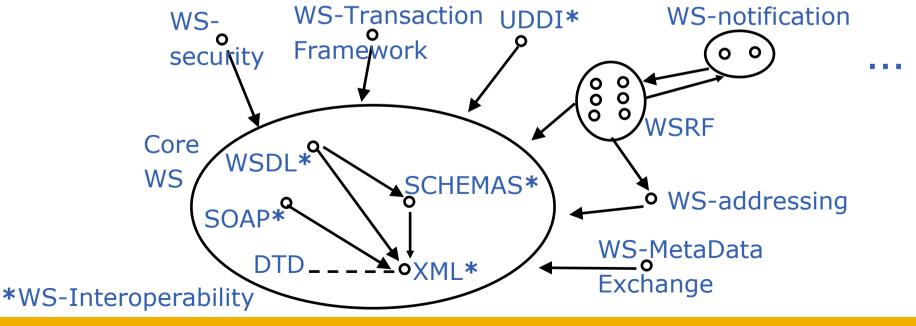
- WS-Security Framework for authentication and confidentiality
- WS-Transaction Framework for robustness of correlated interactions,
   e.g. two phase provisionally book everything, then confirm everything
- UDDI standard repository interface (included in WS-I)
- WS-MetaDataExchange how to communicate meta-data
- •





## **WSRF-Related Standards**

- WS-Addressing For communication of identities between services
  - Includes resource identifiers
- WS-Notification Framework of notification interaction
  - subscribe, publish
- WSRF Web Services Resource Framework
  - Consistent standard framework for resources
    - creating, identifying, changing, destroying





## The Talks / Practicals

- XML Review
- XSD Schemas
- SOAP brief
- WSDL the main purpose
- Tools
- Quote of the Day tutorial creating a simple service and client
- WSRF talk only



## **Reference Information**

#### **Standards**

- XML specification <a href="http://www.w3.org/XML/">http://www.w3.org/XML/</a>
- XML Schema specification <a href="http://www.w3.org/XML/Schema">http://www.w3.org/XML/Schema</a>
- SOAP specification <a href="http://www.w3.org/2000/xp/Group/">http://www.w3.org/2000/xp/Group/</a>
- WSDL specification <a href="http://www.w3.org/2002/ws/desc/">http://www.w3.org/2002/ws/desc/</a>

#### **WSRF**

- General information <a href="http://www.globus.org/wsrf/">http://www.globus.org/wsrf/</a>
- Proposed standard –
- http://www.oasis-open.org/committees/tc\_home.php?wg\_abbrev=wsrf
- Container implementation –
   http://www.globus.org/toolkit/downloads/4.0.0/



## **Reference Information**

#### **Web Service Containers and Toolkits**

- Sun JWSDP (Java) <a href="http://java.sun.com/webservices/jwsdp/index.jsp">http://java.sun.com/webservices/jwsdp/index.jsp</a>
- JWSDP compatible containers -<a href="http://java.sun.com/webservices/containers/">http://java.sun.com/webservices/containers/</a>
- Apache Axis (Java and C++) <a href="http://ws.apache.org/axis/">http://ws.apache.org/axis/</a>
- Microsoft .NET (ASP.NET, C#) <a href="http://www.microsoft.com/net/">http://www.microsoft.com/net/</a>
- IBM WebSphere (Java) <a href="http://www-306.ibm.com/software/websphere/">http://www-306.ibm.com/software/websphere/</a>
- White paper comparing J2EE web services to .NET web services -
- http://www.theserverside.com/articles/article.tss?I=J2EE-vs-DOTNET
- Series of articles describing how to build a web services container in Python –
- http://webservices.xml.com/pub/a/ws/2004/01/20/salz.html
- Perl WSRF container implementation (experimental) –
- http://www.sve.man.ac.uk/Research/AtoZ/ILCT





# THE END