Web Services & WSRF Introduction

Richard Hopkins
National e-Science Centre, Edinburgh
February 23 / 24 2005

Information Society

Goals

- **Goals –**
  - An Appreciation of the role and context of Web Services
  - An understanding of the structure of this event

Evolution of Electronic inter-enterprise interaction

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

Web Browsing

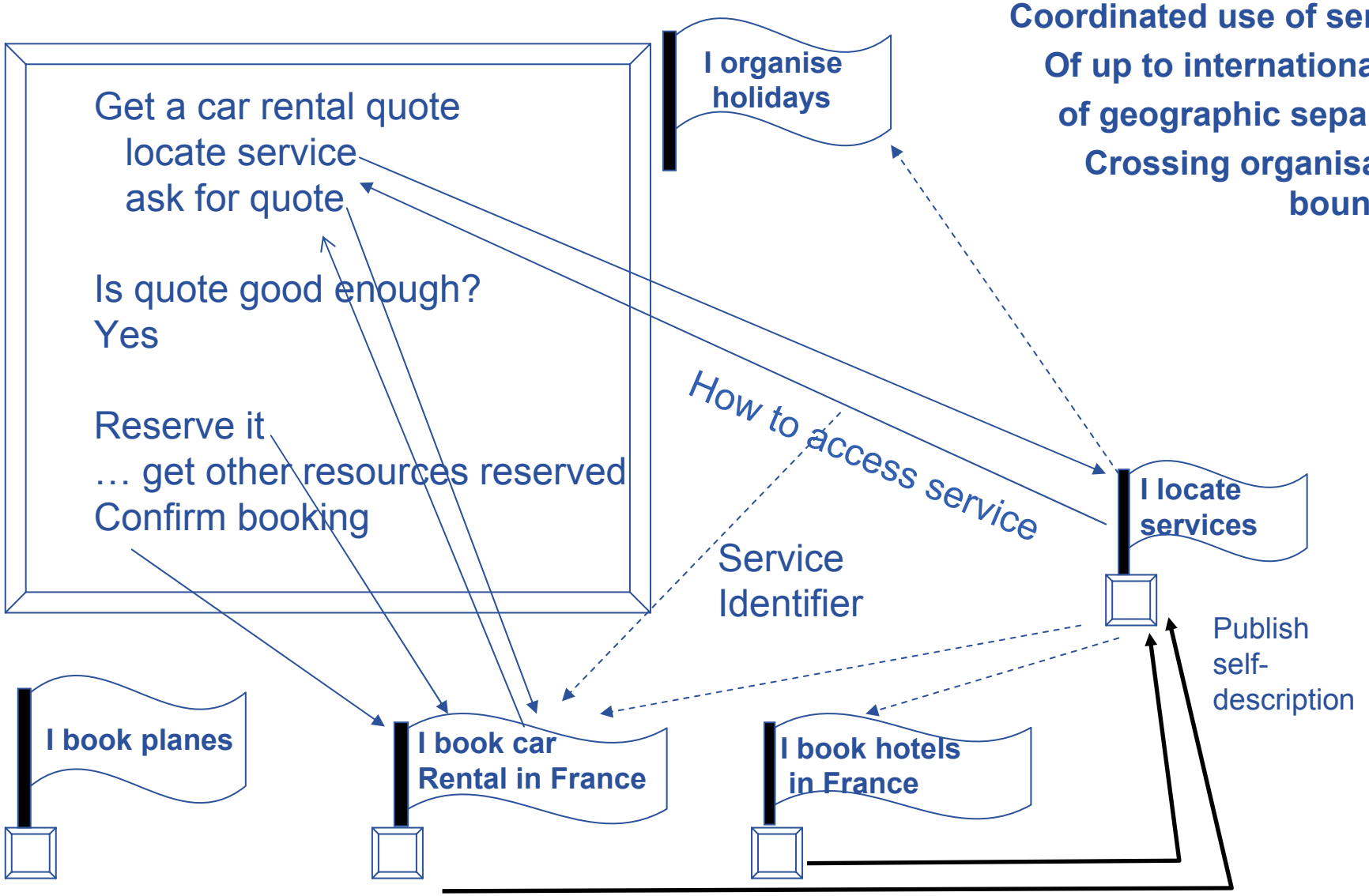
- **Human travel agent provides "organise holiday" service by surfing the web to look for and invoking services – book a hotel; book a plane; book a car hire; ….; confirm bookings of best options to meet client needs.**

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	No

Service Interaction



Essential Characteristics of Web Services Approach

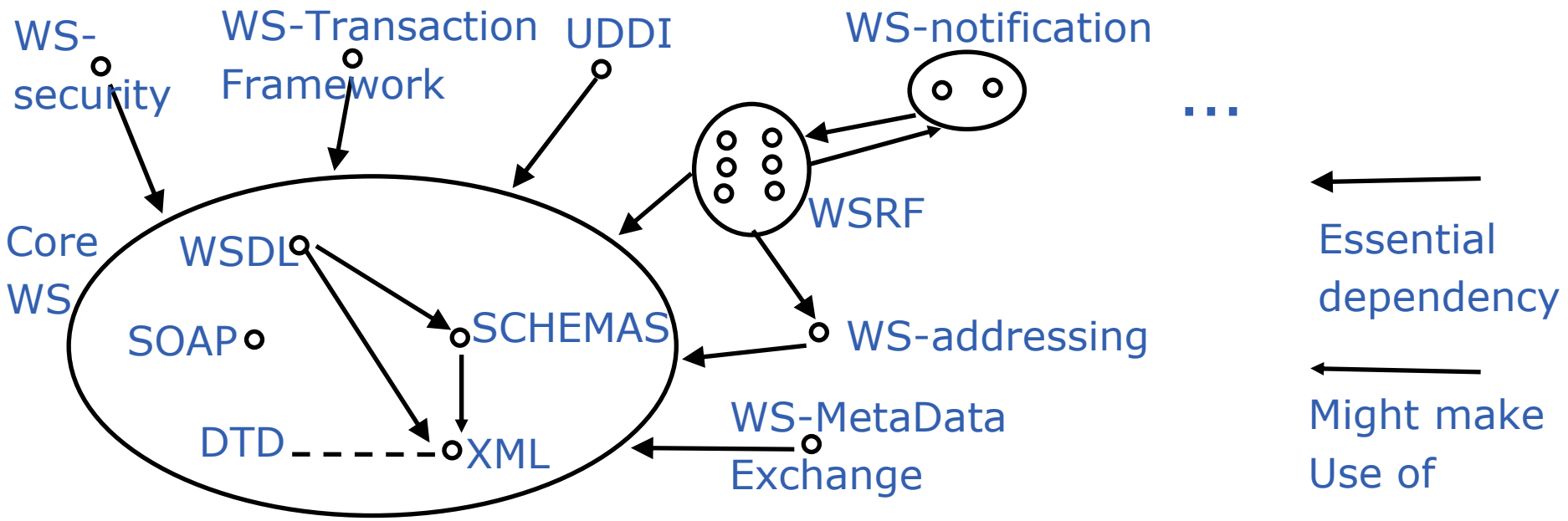
Need to achieve effective cooperation even though

- the different services are produced by different organisations, without any design collaboration, on different platforms (**interoperability**)
- the services are autonomously evolving

- **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**
    - 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
    - XML allows extension points – one participant may have an older WSDL definition which accommodates extensions with additional information

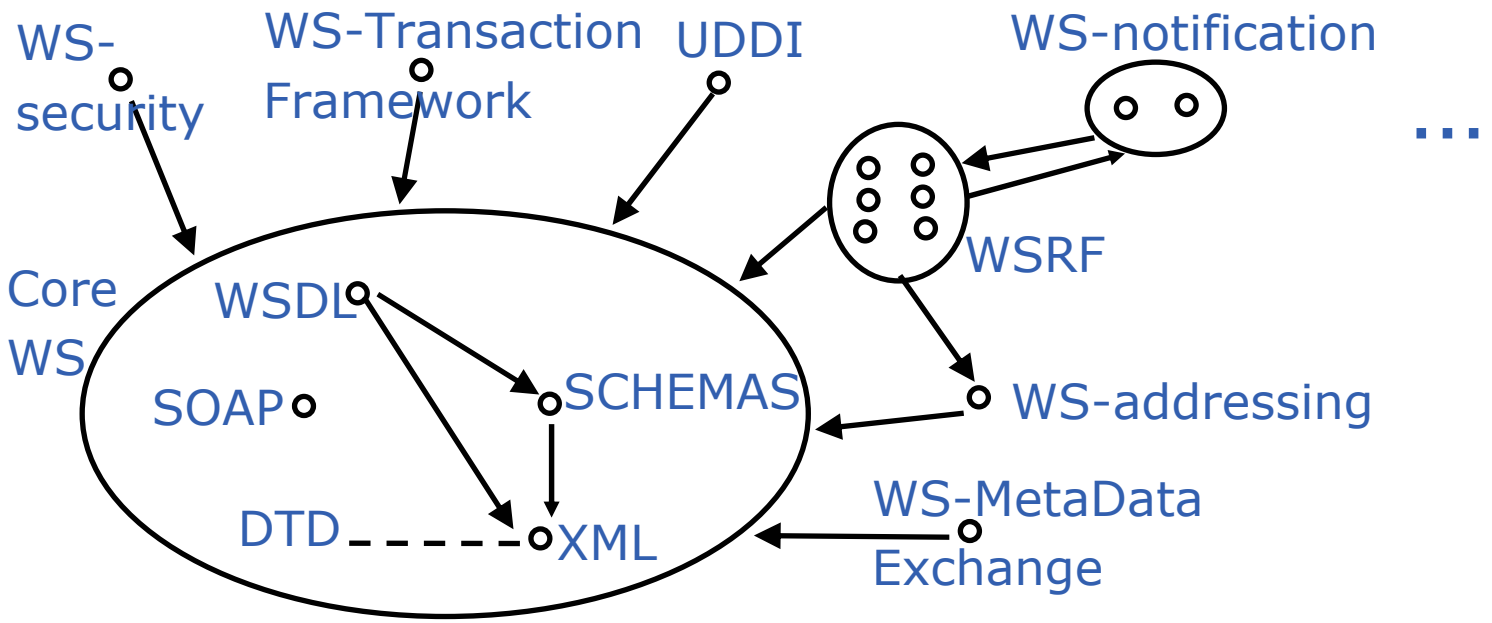
Standards

- **Collaboration is on defining generic standards, rather than on specific design**
- **Two main standards bodies –**
  - W3C – actually produces “recommendations” – web community
  - OASIS – industry – IBM, Microsoft, Sun, ….
- **These standards are factored to allow partial adoption and combination**
  - Higher level standards build on the basic ones



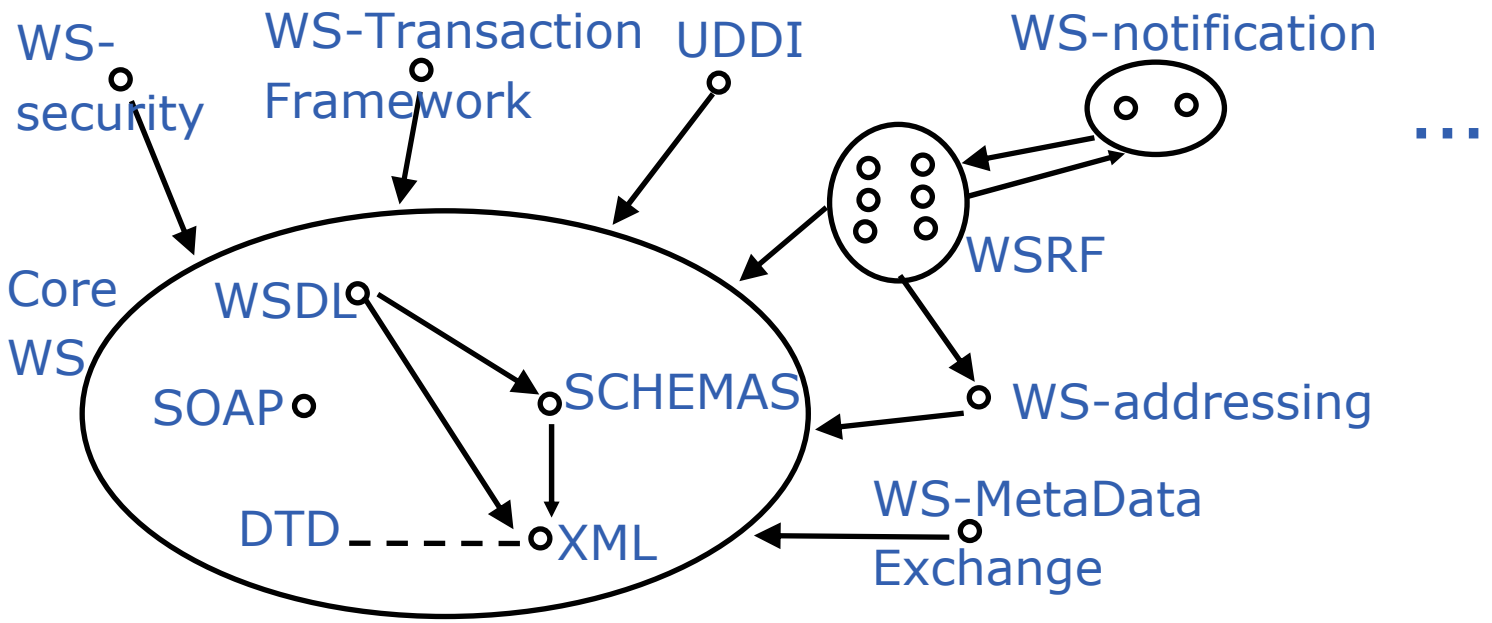
Some Arising Standards

- **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**
- **WS-MetaDataExchange – how to communicate meta-data**
- **….**



WSRF-Related

- **WS-Addressing - For communication of identities between services**
- **WS-Notification - Framework of notification interaction – subscribe, publish**
  WSRF – Web Services Resource Framework
  - E.g. a quote is a persistent entity which will need to be identified in subsequent interactions to finalise a provisional booking – a **resource**
  - Consistent standard framework for creating, identifying, destroying resources
  - Close to core – ubiquitous pattern; other standards use resources



WS vs O-O

- **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 (or Interface)–**
  - a collection of operations
- **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 less constrained than O-O model**

A Perspective on Web Services Model

- **COUPLING – about intensity of communication**
  - Degree of statically shared knowledge between two end of an interaction (knowledge which the programmer/designer has to know and build-in) – how much has to be communicated
  - Frequency and extendt of communication relative to processing
- **A scale of looser coupling (in both senses)**
- **Shared variable**
  - interation 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 consummer) requesting a service
  - Other end (service provider) servicing the reques
  - Quite similar to O-O (but might not be a reply!)

A Perspective on Web Services Model

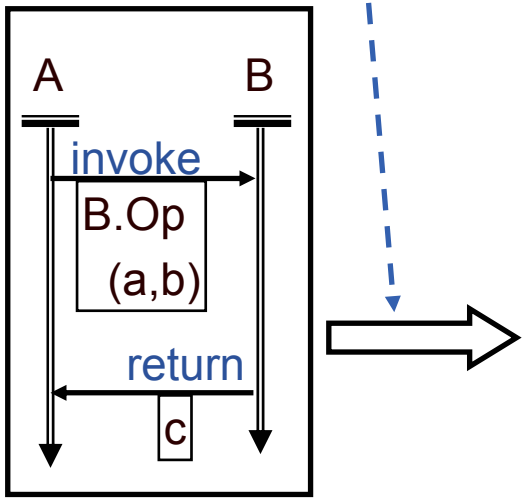
- **Shared Variable Model - Close coupling**
  - The programmers of user side of an interaction know all about representation
  - Shared implementation
  - Suitable for single-programmer level
  - Interaction of order of nanosecond
  - Fine granuality
    - almost no work in a variable assignment
    - 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

A Perspective on Web Services Model

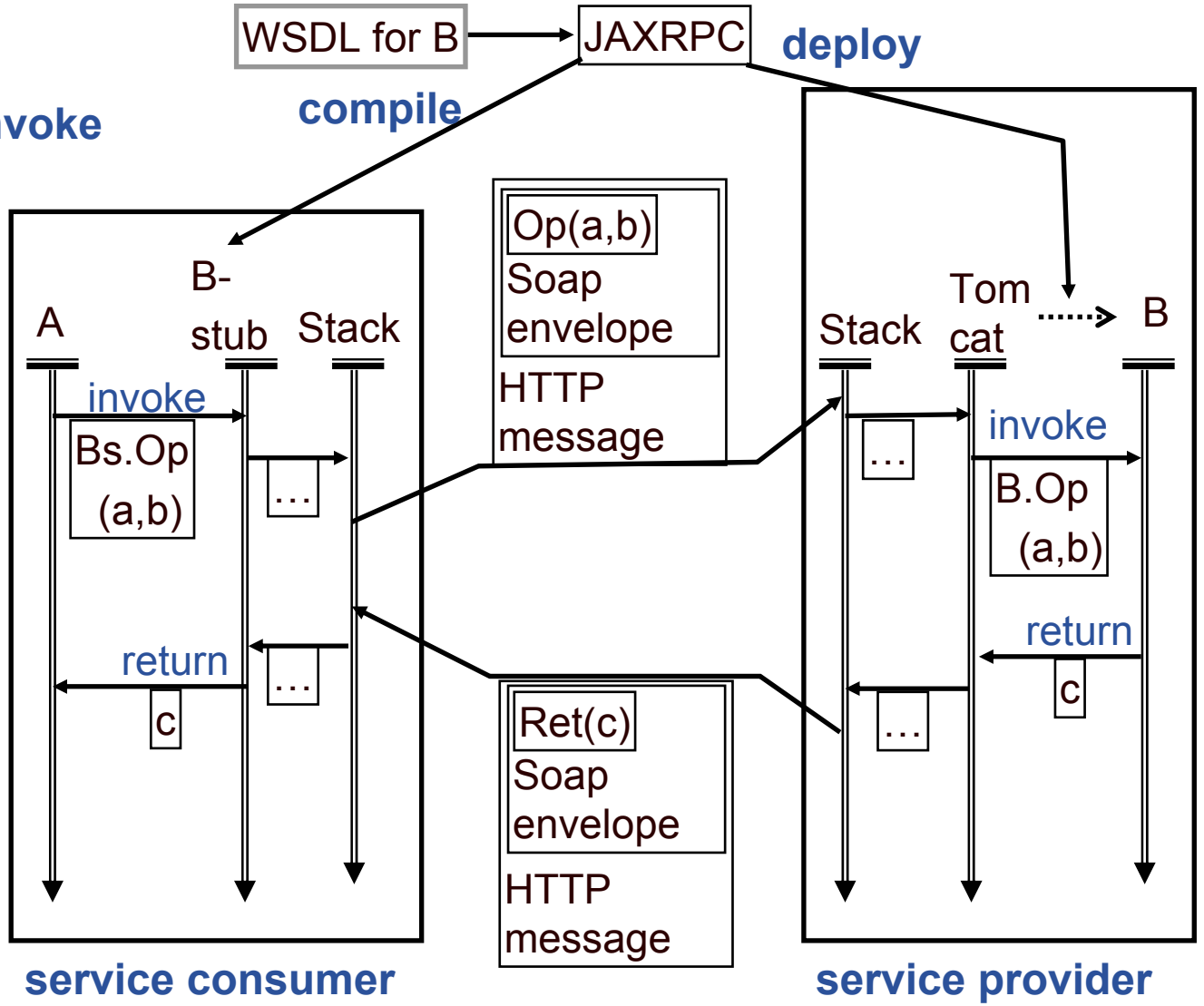
- **Shared Variable Model - Close coupling**
  - Shared implementation ; single-programmer ; nanosec interaction
  - nanosecond interaction; fine granuality;
- **Object-Oriented Model – Medium Coupling**
  - Shard Class Design ; single organsiation ;
  - Micro/milli-sec interaction; medium granuality
- **Web Services - Loose coupling**
  - Programmers on user side knows how to programme the discovery of a service
  - Shared standards and knowledge of standard repository
  - Interaction of order of 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 Objert-oriented model – service request-response is mapped to method invocation-return
- **Progressively –**
  - looser coupling; more time-expensive interaction, coarser granuality
  - Each model builds on the previous one – uses it internally

O-O (JAVA) Web Services

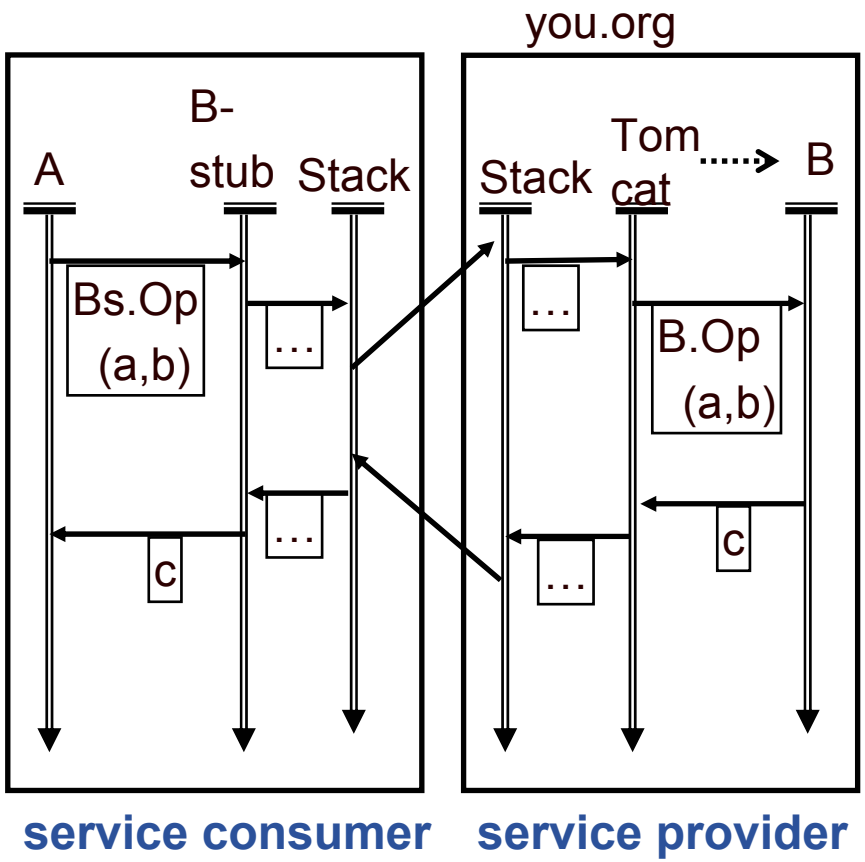
Insert the web into the invoke and into the return

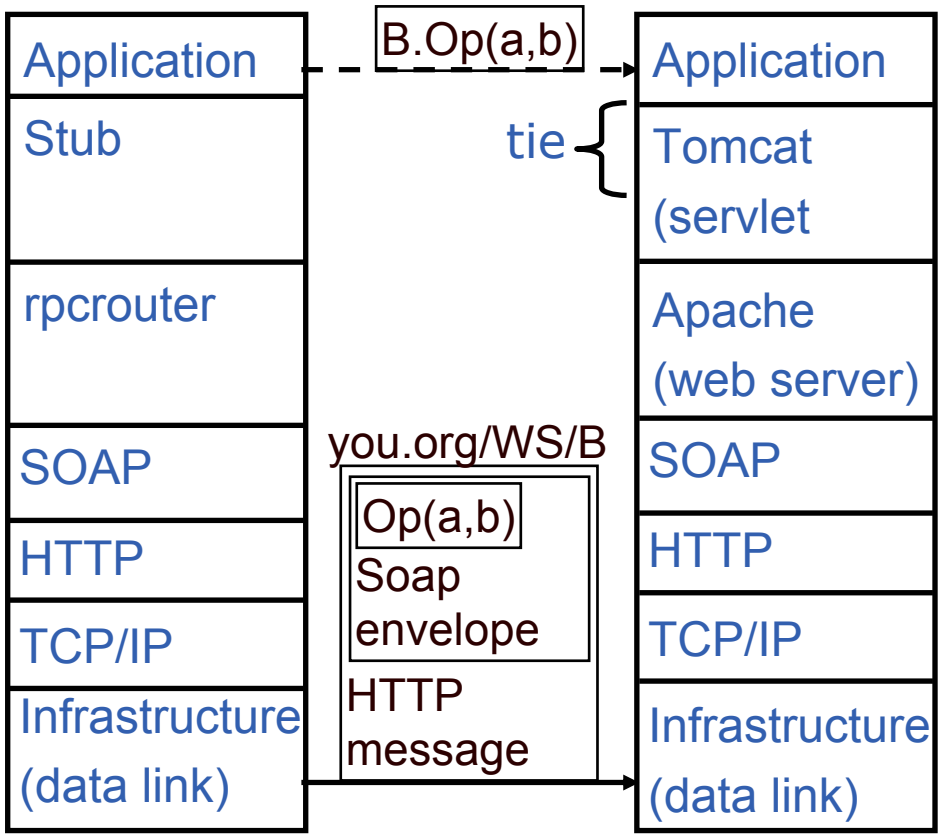


Tomcat – an example container



Communication Stacks





- **Stub –represents service to invoker**
- **Tie - represents invoker to service**
- **Glue = tie + stub**

Schedule

DAY 1

- 09.30 - 10.00 Introduction
- 10.00 - 11.00 XML
- 11.00 - 11.15 Coffee
- 11.15 - 12.15 Schemas
- 12.15 - 13.00 XMLSPY practical on Schemas
- 13.00 - 14.00 Lunch
- 14.00 - 15.00 SOAP
- 15.00 - 16.45 Quote of The Day Tutorial (with 15 min coffee break)

DAY 2

- 09.00 - 10.00 WSDL
- 10.00 - 12.00 File Repository Tutorial Part I (with 15 min coffee break)
- 12.00 - 13.00 WSRF
- 13.00 - 14.00 Lunch
- 14.00 - 16.00 File Repository Tutorial Part II (with 15 min coffee break)

Approach

- **The talks contain a lot of detailed information**
  - Not expected to learn all this during the course
  - For a lot of scenarios you will not need to know full details
- **Intended to**
  - Give an appreciation of capabilities
  - To be reference material when needed
    - A first port of call rather than the formal standards definitons
      - *Which are not always easy to understand*

The End

THE END