Web Services

Technology Review

Background

- Build a distributed computing platform for the Web
- Applications are encapsulated, loosely coupled Web "components" that can bind dynamically to each other
- Easily find these components and integrate them to meet specific requirements
- Plug-and-Play
  - Buy and integrate code into your system
  - Subscribe to necessary services

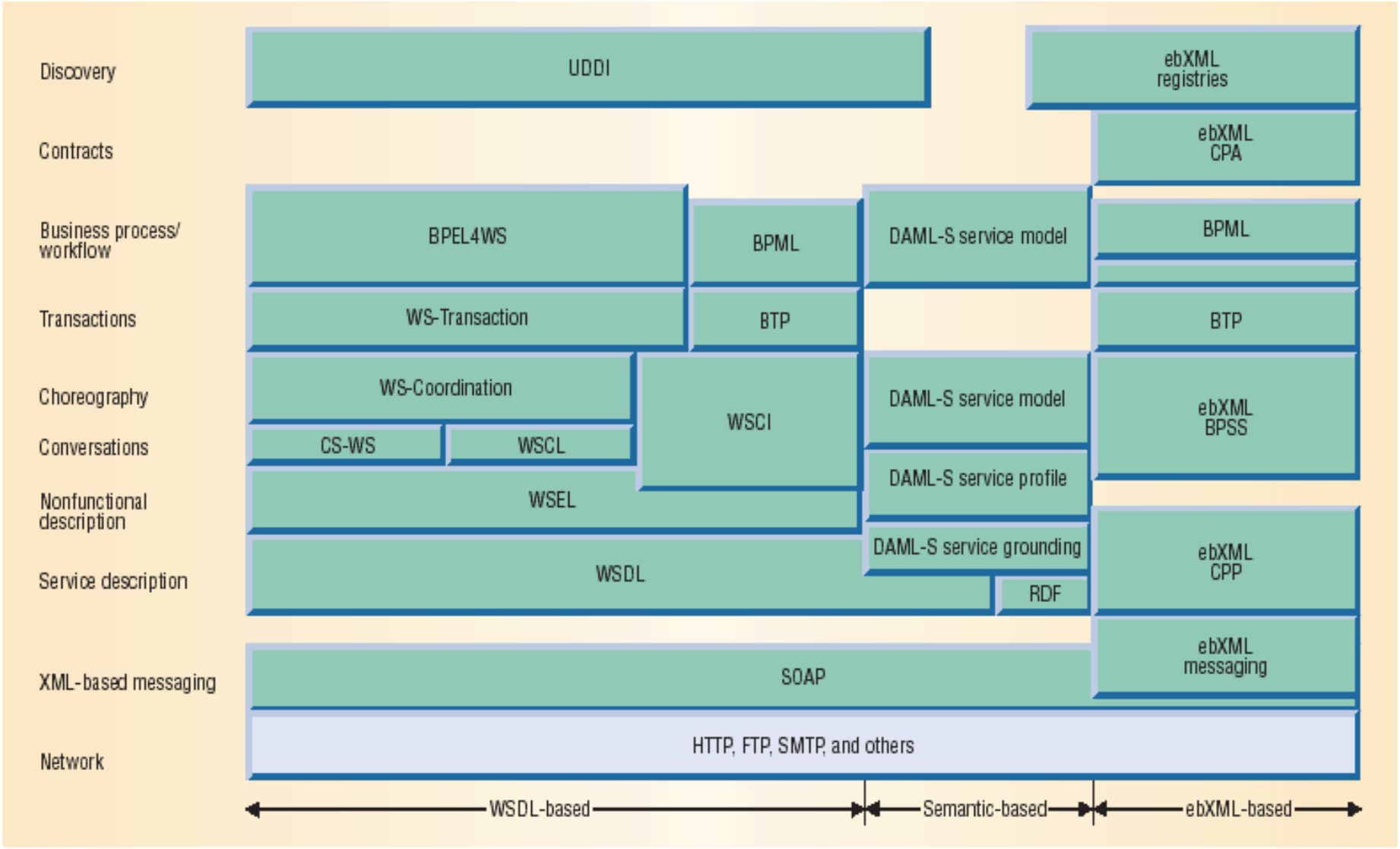
What are Web Services?

- New breed of Web application
- Self-contained, self-describing, modular applications that can be published, located, and invoked across the Web
- Perform functions, which can be anything from simple requests to complicated business processes
- Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service

Web Services Standards

- **UDDI** provides a mechanism for clients to find web services. A UDDI registry is similar to a CORBA trader, or it can be thought of as a DNS service for business applications.
- **WSDL** defines services as collections of network endpoints or *ports*. A port is defined by associating a network address with a binding; a collection of ports define a service.
- **SOAP** is a message layout specification that defines a uniform way of passing XML-encoded data. In also defines a way to bind to HTTP as the underlying communication protocol. SOAP is basically a technology to allow for “RPC *over the web*”.

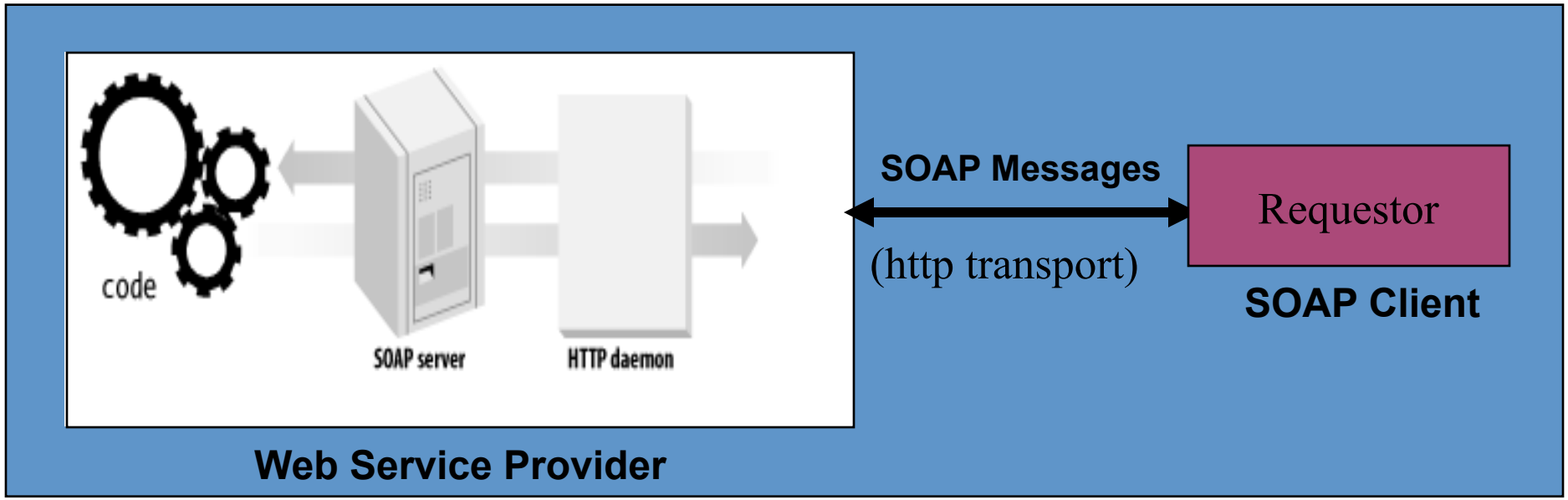
Web Services Stack (Turner et al., 2003)



Web services technology stack

- Collection of Web Services standards has been put together and categorized into different levels depending upon their function
- Divided into four levels: protocol, description, interaction and security
- Helps us to understand the purpose of the standards and their underlying assumptions that might be required
- Example: interaction level (CS-WS) presupposes process description level (BPEL4WS), and focuses on exchange of messages based on the order of sequence described in the process description level

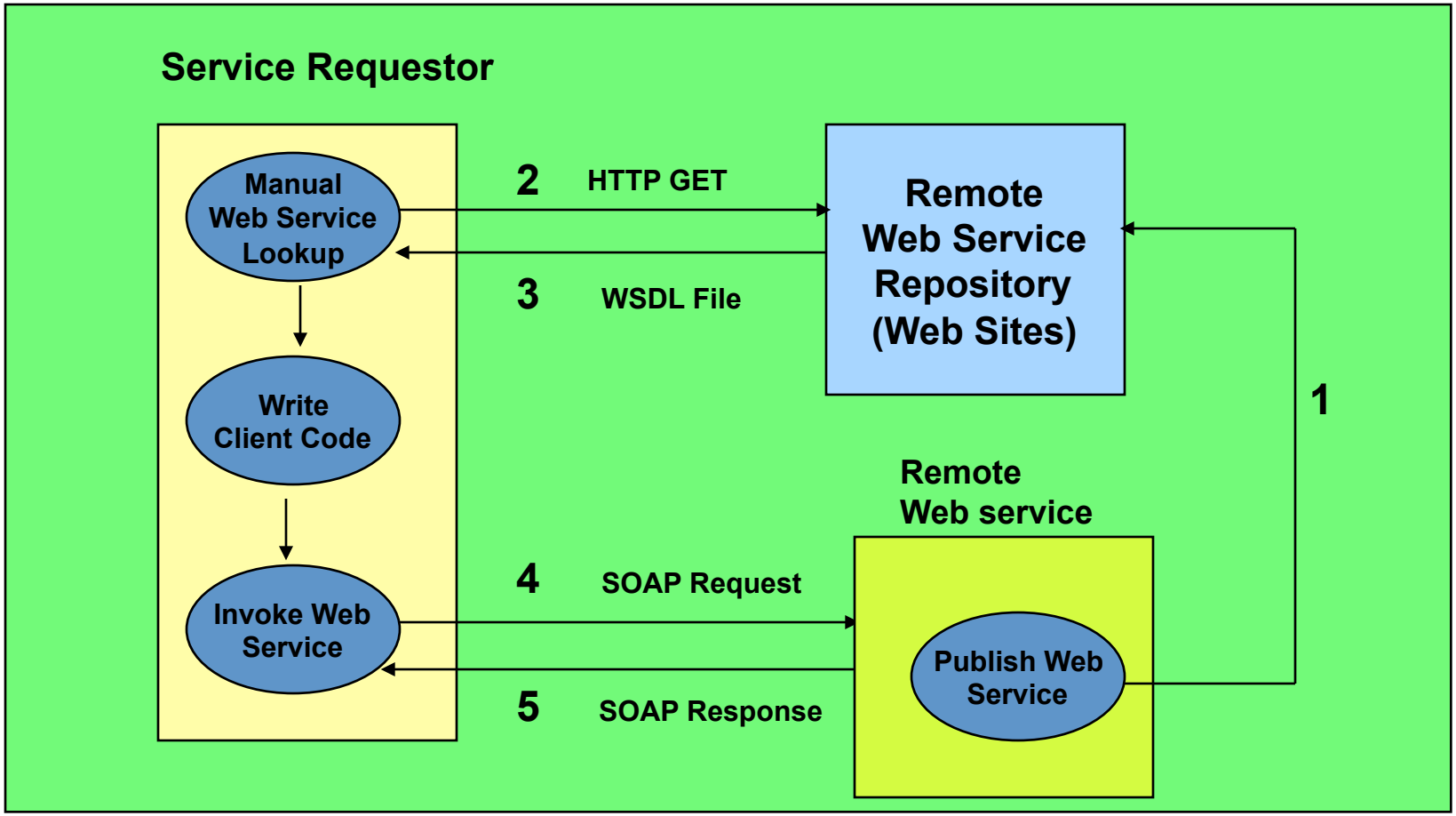
Web Services: How They Work?



- Components required
  - Software which needs to be exposed as a Web service
  - A SOAP Server (Apache Axis, SOAP::Lite, etc.)
  - HTTP Server (if HTTP is used as the transport level protocol)
  - SOAP Client (Apache Axis, SOAP::Lite etc.)

From S. Chandrasekaran's Talk

Simple Web Service Invocation



WSDL - Web Service Description
SOAP - Web Service Message Protocol

From S. Chandrasekaran's Talk

Web Service Message Protocol - SOAP

- SOAP is an XML Messaging Protocol
  - that allows software running on disparate operating systems, running in different environments to

    make Remote Procedure Calls (RPC).

Web Service Description

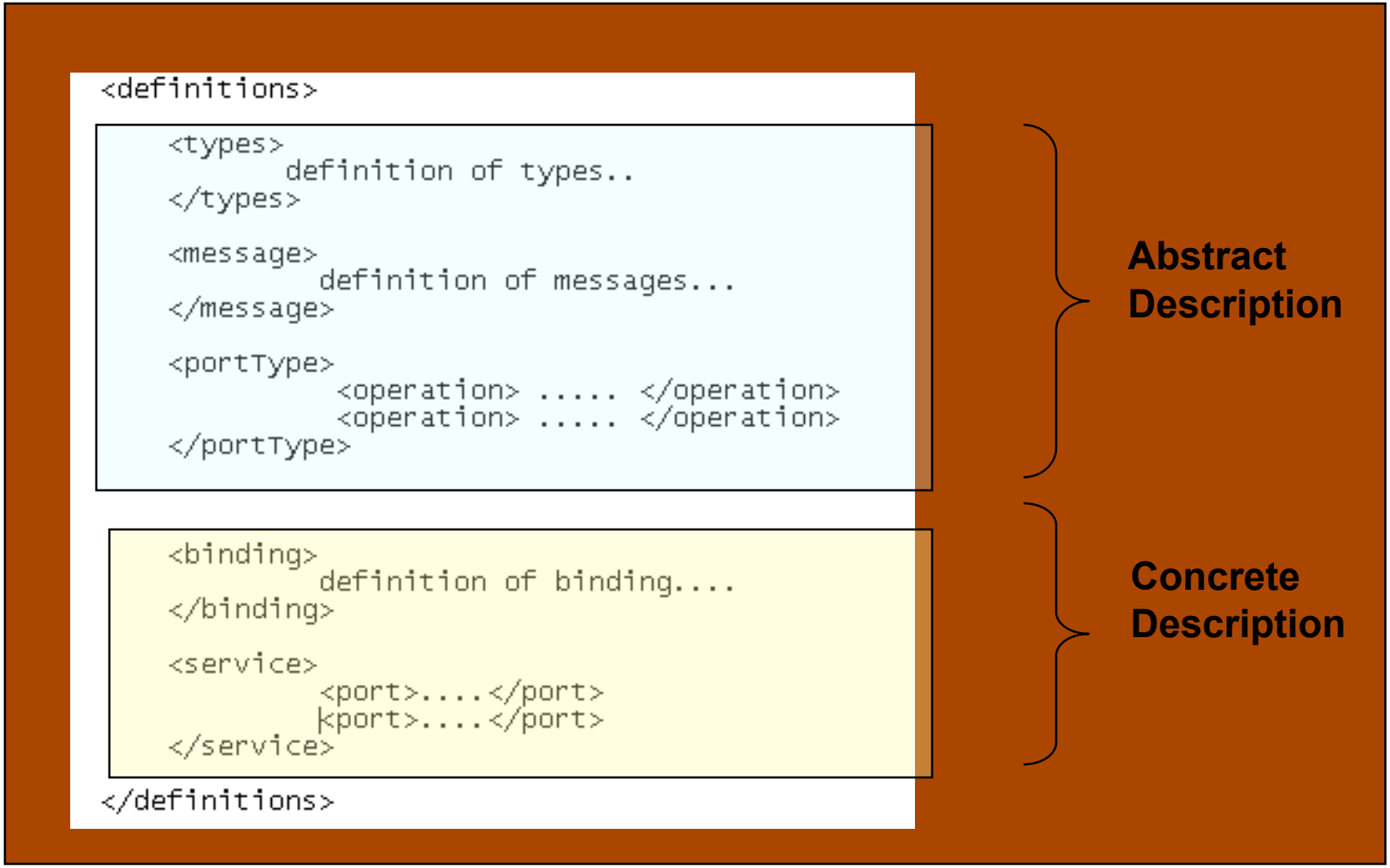
- Why describe Web services?
  - A service requestor needs to analyze a service for his requirements
  - A Web service needs to provide the following information
    - the operations it supports
    - the transport and messaging protocols on which it supports those operations
    - the network endpoint of the Web service
- Languages such as WSDL, DAML-S, RDF can be used for describing Web services
  - WSDL – describes the syntactic information of a service
  - DAML-S and RDF – describe the syntactic as well as the semantic information

From S. Chandrasekaran's Talk

WSDL – Web Services Description Language

- Describes the programmatic interface of a web service – similar to IDL
- Two parts of WSDL description:
  - Application-level (abstract interface)
    - Vocabulary, Message, and Interaction
      - data type definitions, operations supported by the service, input/output message formats
  - Specific protocol-dependent details
    - network address, protocol bindings, etc.
      - *what* communication protocol to use
      - *how* to accomplish individual service interactions
      - *where* to terminate communication
    - A port element describes a single end point as a combination of a binding and a network address
    - A service element groups a set of related ports

Web Service Description (WSDL)



From S. Chandrasekaran's Talk

UDDI – Universal Description, Discovery, and Integration

- Mechanism to register and locate web services
- Interaction with UDDI accomplished via a set of pre-defined SOAP interfaces
- Web services register two types of information within UDDI:
  - tModel (technical model)
    - Abstract service protocols that describe a particular web service's behavior
  - businessEntity
    - Describes the service implementation
    - Refers to multiple tModels and provides descriptions about the behavior of the collection of tModels

How UDDI Works ?

1. 

SW companies, standards bodies, and programmers populate the registry with descriptions of different types of services

2. 

Businesses populate the registry with descriptions of the services they support

UDDI Business Registry

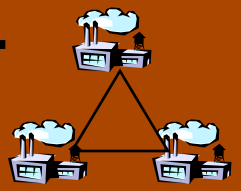
Business Registrations

Service Type Registrations

3. UBR assigns a programmatically unique identifier to each service and business registration

4. 

Marketplaces, search engines, and business apps query the registry to discover services at other companies

5. 

Business uses this data to facilitate easier integration with each other over the Web

Source : http://www.uddi.org/pubs/UDDI_Overview_Presentation.ppt

Credits

- **Vijayan Sugumaran.** School of Business Administration, Oakland University.