

# Creating and Registering Mobile E-services Using Java, UDDI, and WSDL

Chris Peltz ([chris\\_peltz@hp.com](mailto:chris_peltz@hp.com))

Carollyn Carson ([carollyn\\_carson@hp.com](mailto:carollyn_carson@hp.com))

Hewlett-Packard Company

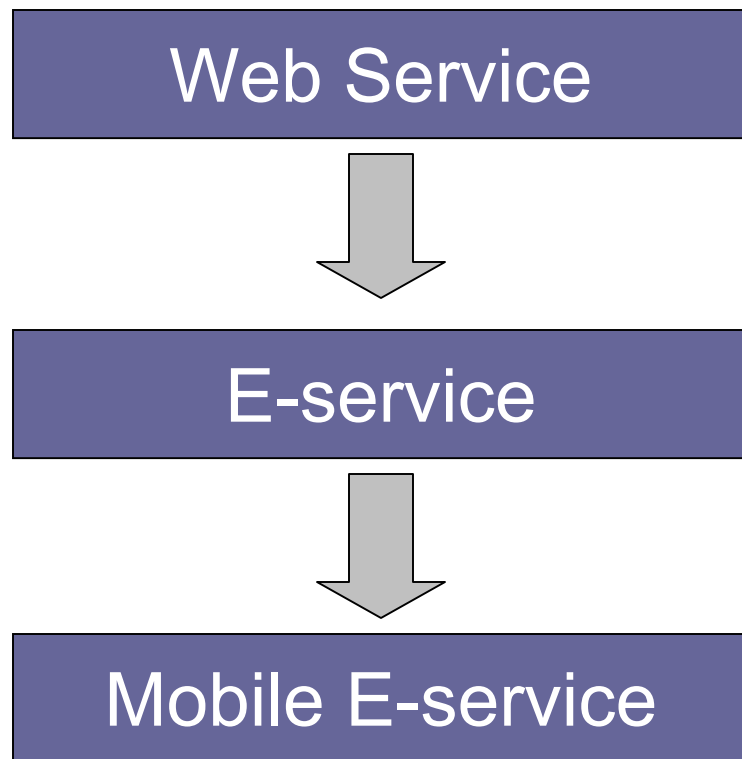
# Introduction

- Interest in web services has exploded over the past year
- There is a need to understand how these technologies can be leveraged today
- HP conducted a study to understand how web services technologies can be used within a mobile e-services ecosystem
- We will present a methodology developed from this study, including key best practices

# Agenda

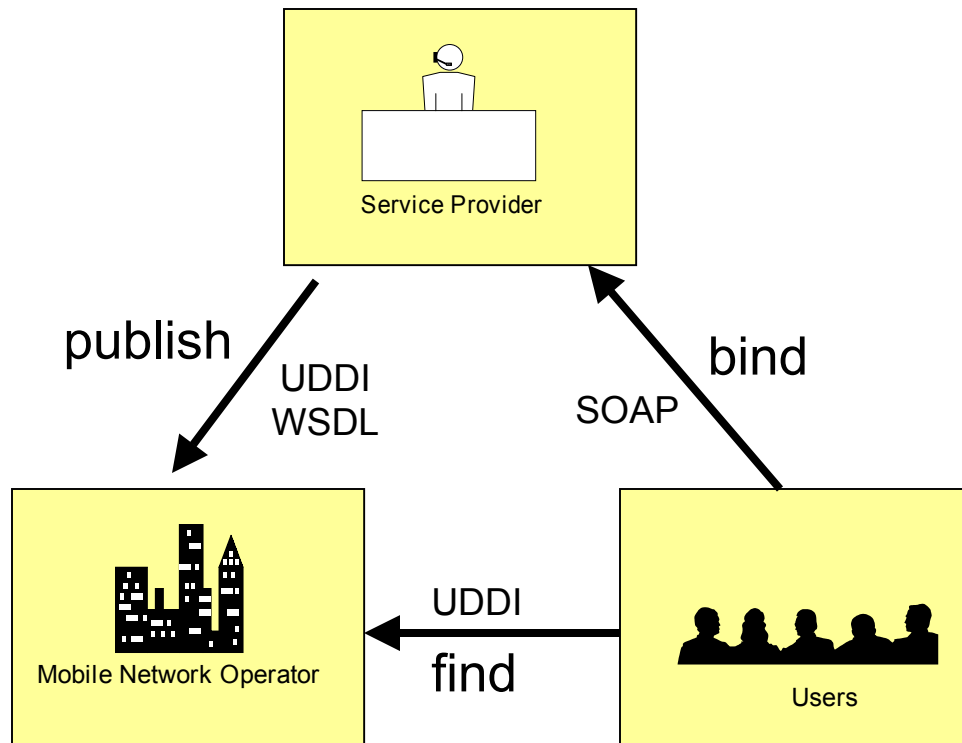
- Overview of web services standards
- Analysis and design issues
- Creating and deploying web services
- The registration and discovery process
- Key learnings and future considerations

# Web Services Defined



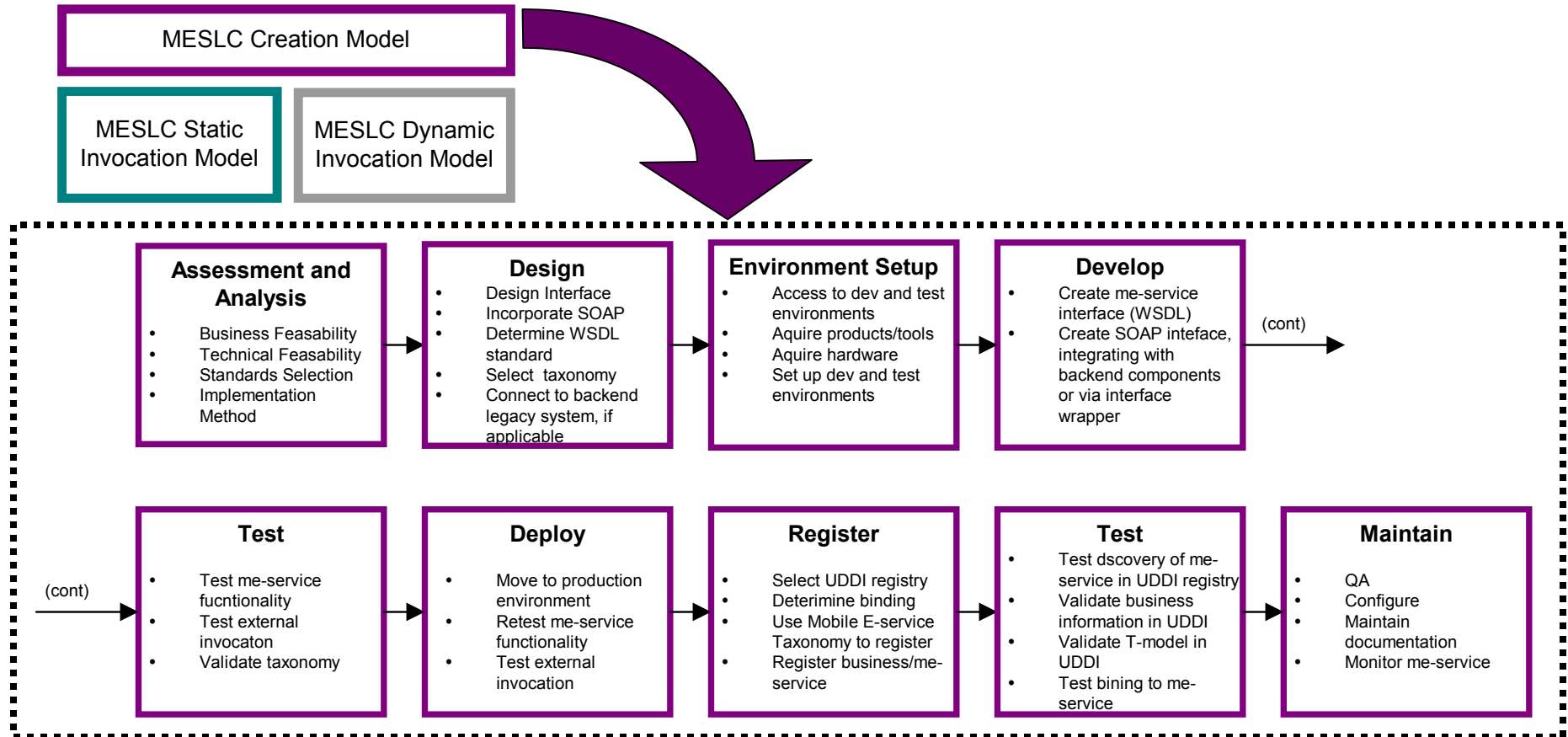
- **Web services** are Internet-oriented software components
- **E-services** are complete solutions derived from combining web services components
- A **Mobile E-service** is a service delivered to an end user over a mobile device

# Service-Oriented Architecture



- **WSDL** describes the capabilities of a web service
- **UDDI** is used to register and discover services
- **SOAP** is the web services communication protocol

# Mobile E-services Lifecycle



# Lifecycle Entity and Roles

Business Analyst	Determines the business feasibility, hosting options, and payment models
Architect	Develops the standards, web service design, and platform requirements
Engineer	Responsible for the development of the project
Administrator	Responsible for hardware availability, monitoring resources, managing UDDI registry
Operations	Responsible for the service deployment, product support, and monitoring of the service
Business Developer	Determines service advertising decisions (i.e., what & how to register in UDDI)

# Analysis and Design

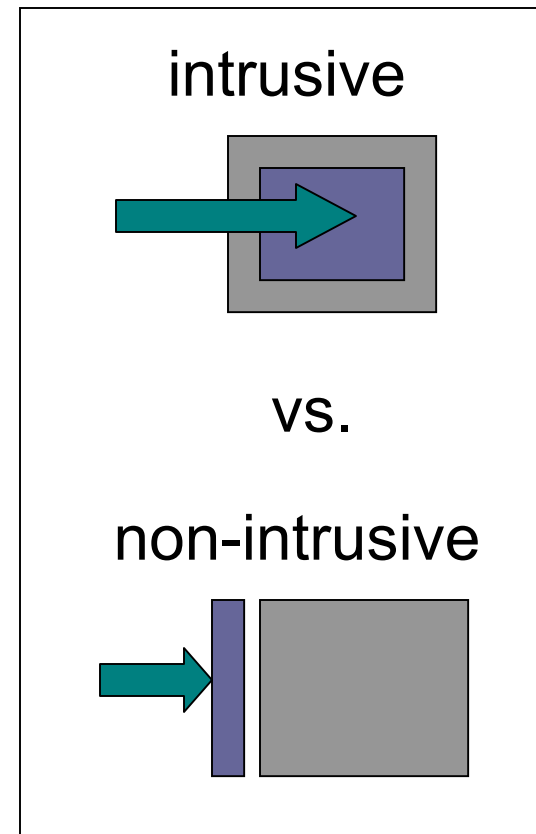
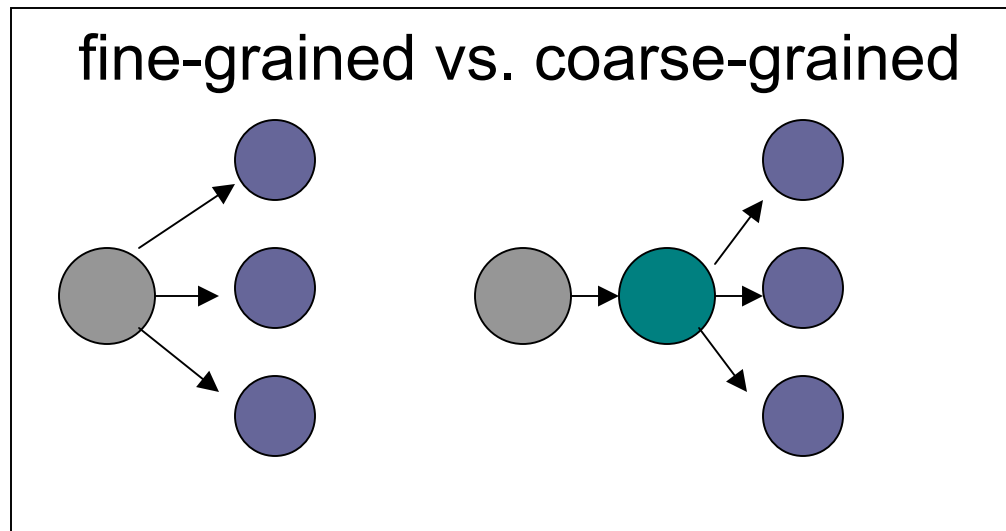
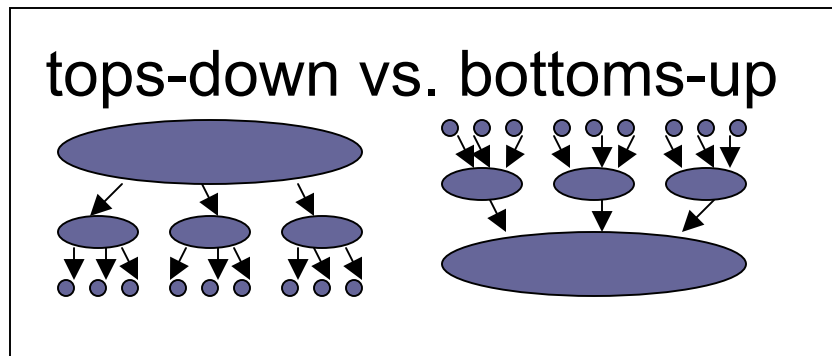
- Assessment and Analysis
- Common Design Approaches
- Registration Considerations



# Assessment and Analysis

1. What is the business value in providing this functionality as a web service?
2. What technologies, platforms, and languages are involved?
3. How would a consumer discover the service?
4. What web services interfaces (WSDL) have to be supported?
5. Will the service be hosted from within the firewall?
6. Will consumers directly invoke the service?
7. Where will the service be advertised?

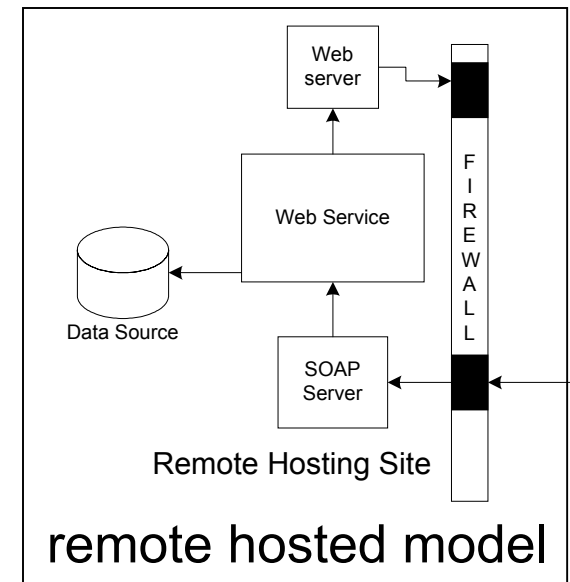
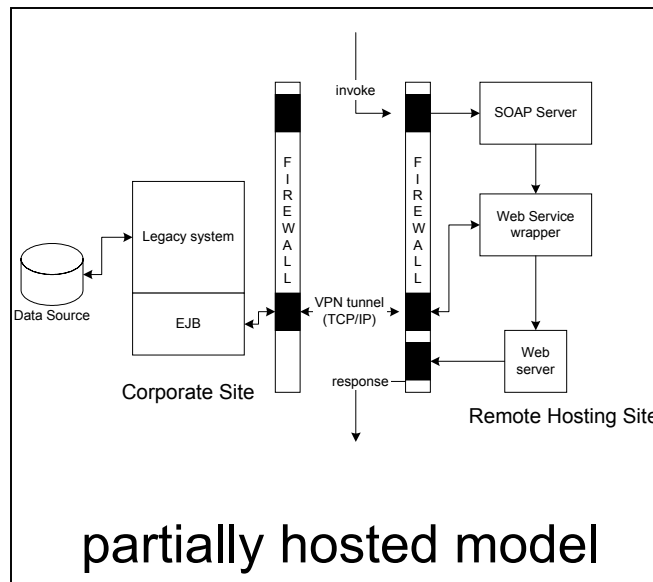
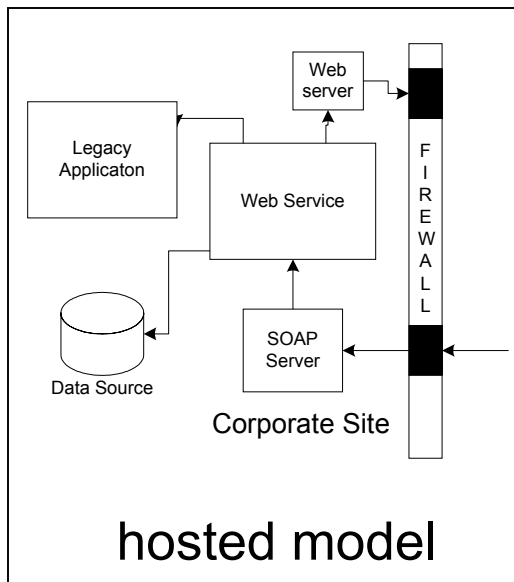
# Common Design Approaches



# Designing The Interface

- WSDL enables service interoperability
  - Compliance with a standard WSDL interface can ensure interoperability with the ecosystem
  - Many web services vendors offer tools to automatically create clients from a WSDL interface
- A service can support a variety of WSDL-based interfaces
- Conforming to an existing WSDL may require a wrapper around the existing implementation

# Hosting Models



# Registration Considerations

- How is the service registered?
  - A taxonomy is used to classify a service
    - Categories: NAICS, UNSPSC
    - Identifiers: D-U-N-S, Thomas Register
- Where is the service registered?
  - Public Registry: available to public
  - Private Registry: target industry segments
- Registration method will impact the visibility of the service to potential customers

# Creating and Deploying The Service

- Web Services Platform Requirements
- Steps to Web Services Creation
- Testing and Deploying The Service
- An Example

# Platform Requirements

SOAP Server	Processes SOAP messages between applications
J2EE Platform	Hosts the services and SOAP server
Developer Tools	Simplify creation of WSDL and client proxies
UDDI Registry	Testing or hosting the service registration process
Registry Tools	Registration and discovery of web services
Build/Deploy Tools	Tools to build and deploy the web service
Testing Tools	Unit testing, interoperability, and load testing

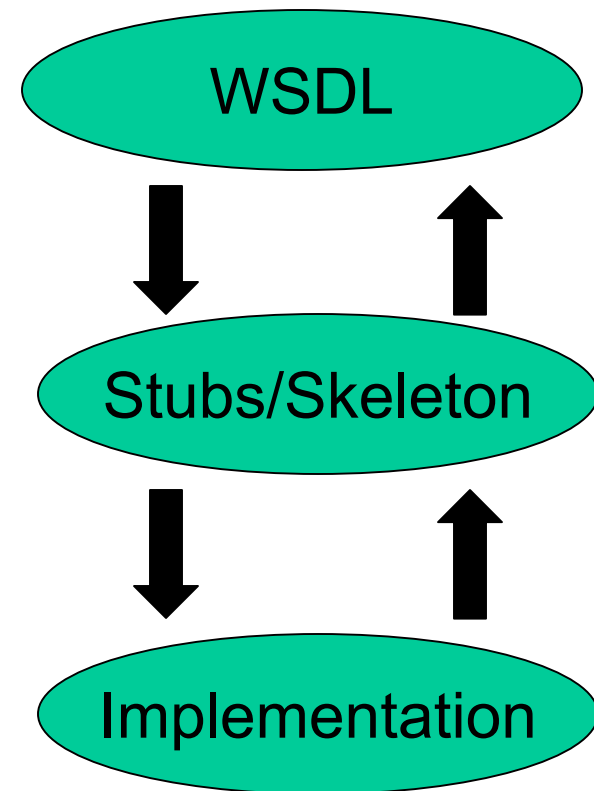
# Creating The Service

- Existing technology must be considered
  - Java, J2EE, HTTP, C/C++, Python, CORBA, Perl
- A Java wrapper may be required in the following situations:
  - Partial remote hosting of a service
  - To conform to an existing WSDL standard
  - C/C++ is being used, but platform is J2EE-based
- Developer tools can simplify the web service creation process



# Creating The Service

- Three methods for creating the web service interface:
  - A WSDL can be created from scratch
  - An existing WSDL can be leveraged
  - It can be created from an existing implementation
- Consider RPC or document-exchange



# Deploying The Service

- The SOAP server and the web services are deployed in a J2EE container
  - Developer tools can automate this process
- An automated build process can simplify the deployment process
  - **Apache ANT** is a XML/Java based build tool that can execute on any operating system
  - ANT can automate the build process, and can be used to generate required web services components

# Testing the Service

## Functional Testing

- WSDL Verification
- URL Bindings
- Boundary Testing

## Interoperability Testing

- SOAP Compatibility
- Firewall Testing
- Client Interaction

## Load Testing

- Performance Testing
- Stress Testing

## Enterprise-Level Testing

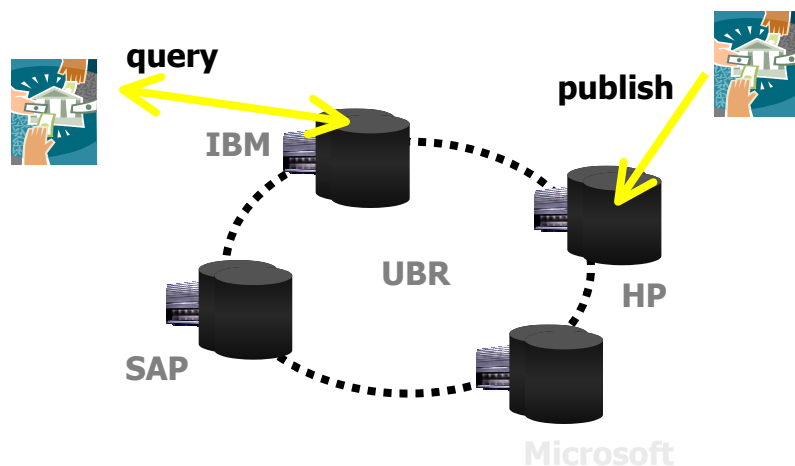
- Security
- Transactions
- Conversations

# Registering and Discovering the Service

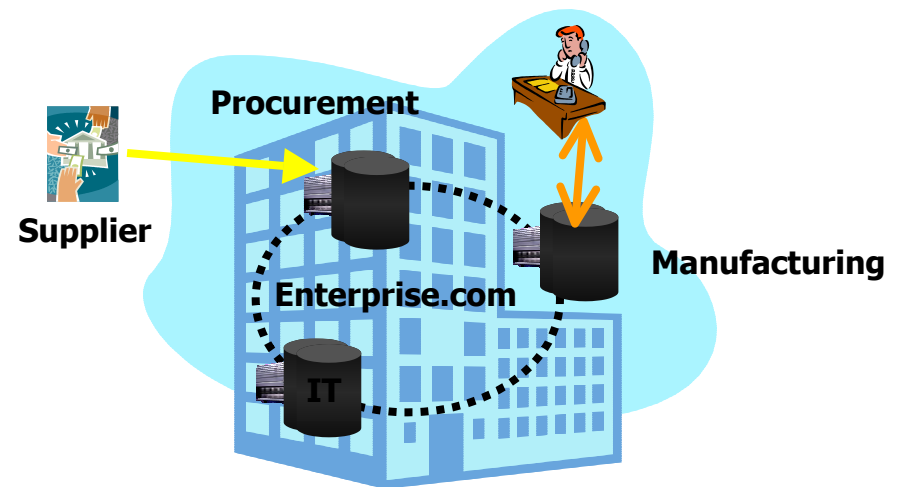
- Overview of UDDI
- Registering the Service
- Discovering and Invoking the Service

# Overview of UDDI

UDDI defines how a business can publish information about itself and the services that it offers

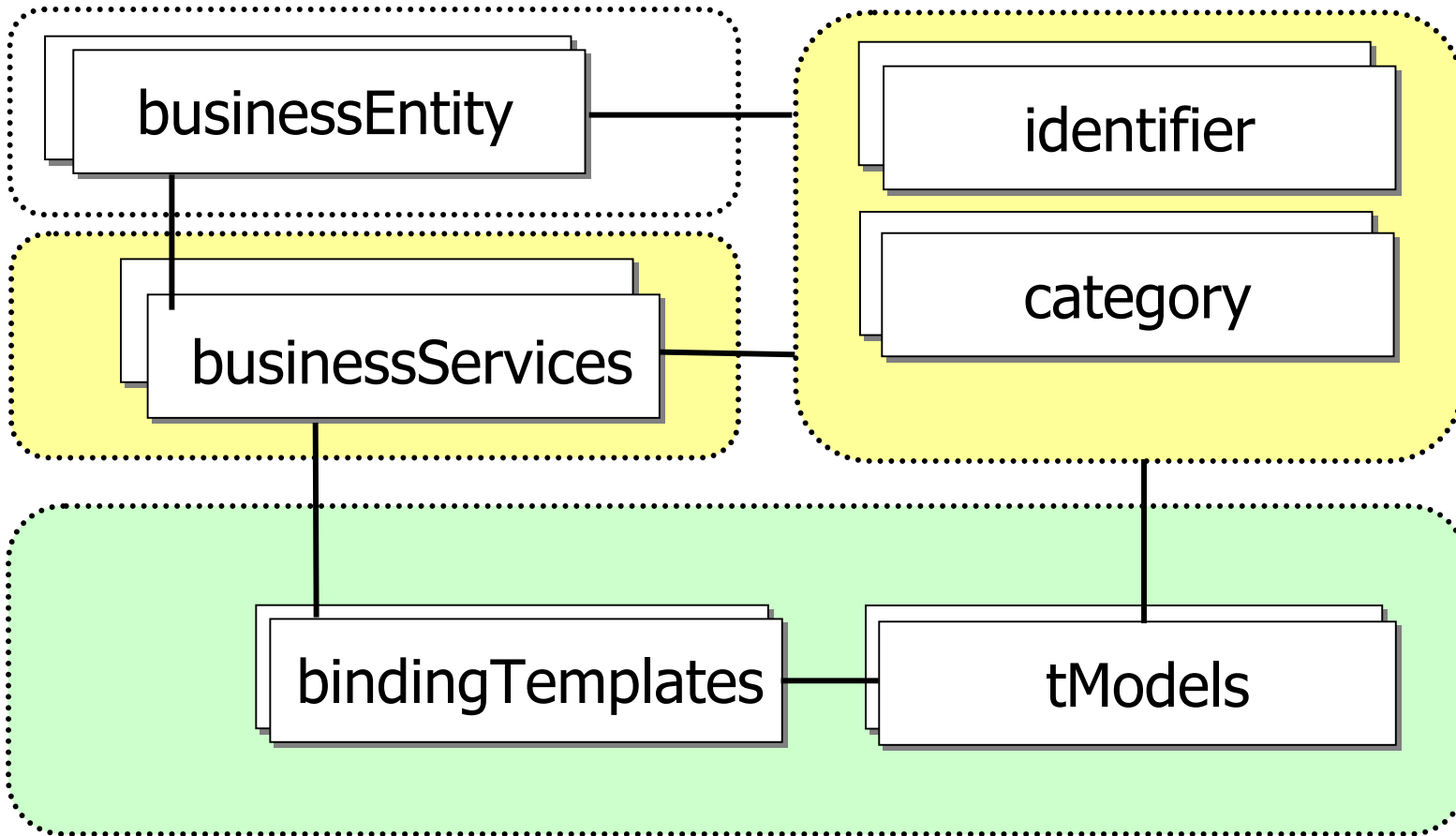


The **Universal Business Registry** is the public registry maintained by a group of node operators. A replication process ensures data integrity



**Private Registries** are used for build private ecosystems (e-marketplaces, portals, partner catalogs, EAI)

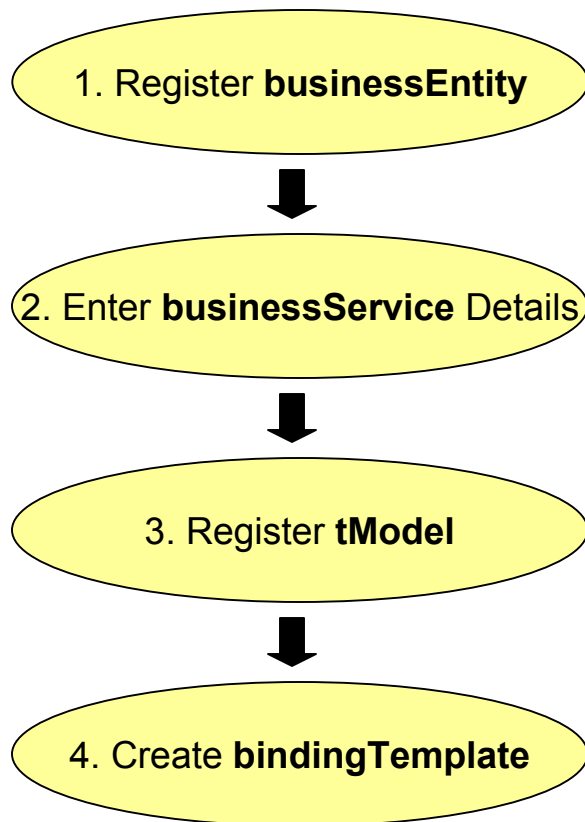
# Modeling a Service in UDDI



# Preparing for the Registration

- Programmatic or interactive?
  - UDDI4J is a popular Java API
  - GUI tools are available to simplify the registration process
- The taxonomy used must be registered as a **tModel** if it doesn't already exist in UDDI
- All UDDI “publish” calls require an authentication token
  - Logins must be obtained prior to registering the service

# Registering the Service



## Best Practices

- Add identifiers/categories to business/service to simplify lookup
- Re-use existing tModels (WSDL) if possible
- Consider separating the URL location from the WSDL and place into the bindingTemplate

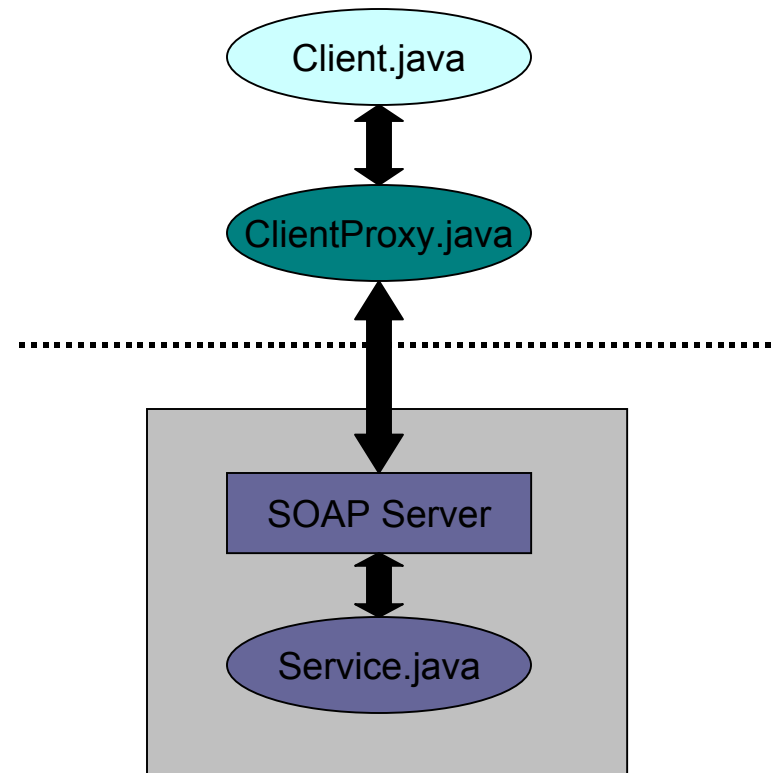


# Discovering the Service

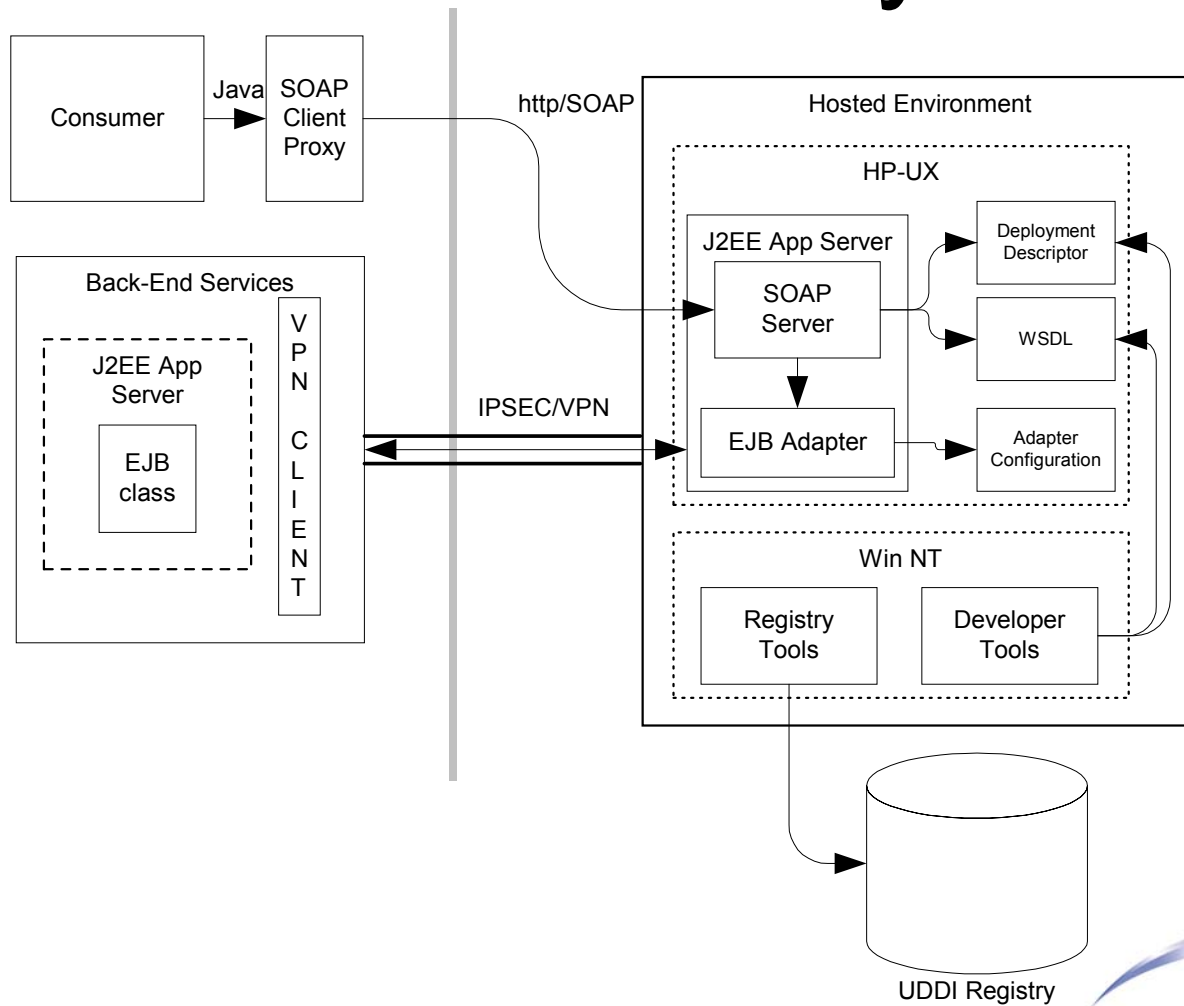
- Process involves qualifying a provider as a potential company to do business with
  - e.g., partner assessments, WSDL conformance, security requirements, scalability
- UDDI can be searched by:
  - Business: using DUNS, UNSPSC, or name
  - Service: using service taxonomies
  - tModel: using an existing WSDL implementation
- Programmatic (JAXR, UDDI4J) and interactive interfaces

# Invoking The Service

- JAXM/JAX-RPC can be used to construct and send SOAP messages
- Or use client proxy tools (e.g., **Apache Axis WSDL2Java**) to automatically generate the code from a WSDL



# Case Study



# Conclusion

- Key Learnings
- Futures

# Key Learnings

- WSDL is at the heart of service interoperability
- Web services must provide well-defined, useful interfaces to the end consumer
- A non-intrusive design may have to be considered for certain hosting models
- Taxonomies are important to enhance visibility
- Almost any technology can be exposed as a service
- Developer tools can greatly simplify the web service registration and creation process

# The Future of Web Services

- Asynchronous messaging support
- Web services orchestration
  - WSFL, BizTalk, X-LANG
- Atomic and “cohesive” transactions
  - Business Transaction Protocol (BTP)
- Security – authentication, authorization
  - SAML, WS-Security, XKMS
- J2EE/.NET interoperability
- Web services manageability

# References

- [www.searchwebservices.com](http://www.searchwebservices.com)
- [www.soapagent.com](http://www.soapagent.com)
- [www.soaprpc.com](http://www.soaprpc.com)
- [www.stencilgroup.com](http://www.stencilgroup.com)
- [www.webservices.org](http://www.webservices.org)
- [www.webservicesarchitect.com](http://www.webservicesarchitect.com)