

Lecture 14

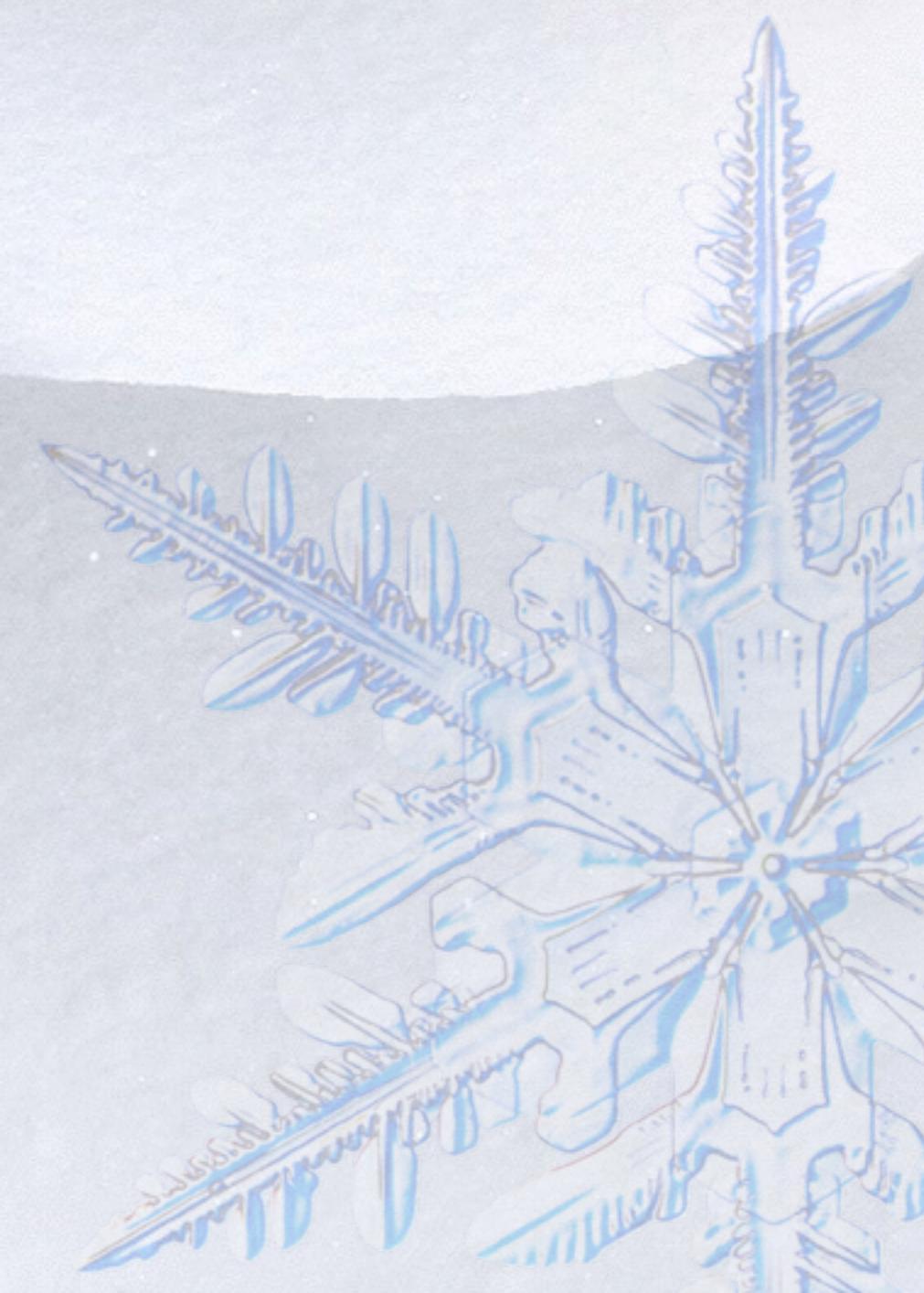
Web Services and

Mashups





Outline

- ❖ Web Services
 - ❖ Mashups
- 

What is a Web Service

- ❖ A web service is a piece of business logic, located somewhere on the Internet, that is accessible through standard-based Internet protocols such as HTTP or SMTP.
- ❖ Using a web service could be as simple as logging into a site or as complex as facilitating a multi-organization business negotiation. Ex: Creating a Sales Order and initiating a workflow.

Web Services: Alternate definition

- ❖ A web service is just a web page meant for a computer to request and process.
- ❖ More precisely, a Web service is a Web page that's meant to be consumed by an autonomous program as opposed to a Web browser or similar UI tool.

Architecture of Web Services

- ❖ **RPC** – Remote Procedure Call
- ❖ **SOA** – Service-oriented architecture
- ❖ **REST** – Representational State Transfer

Remote Procedure Calls

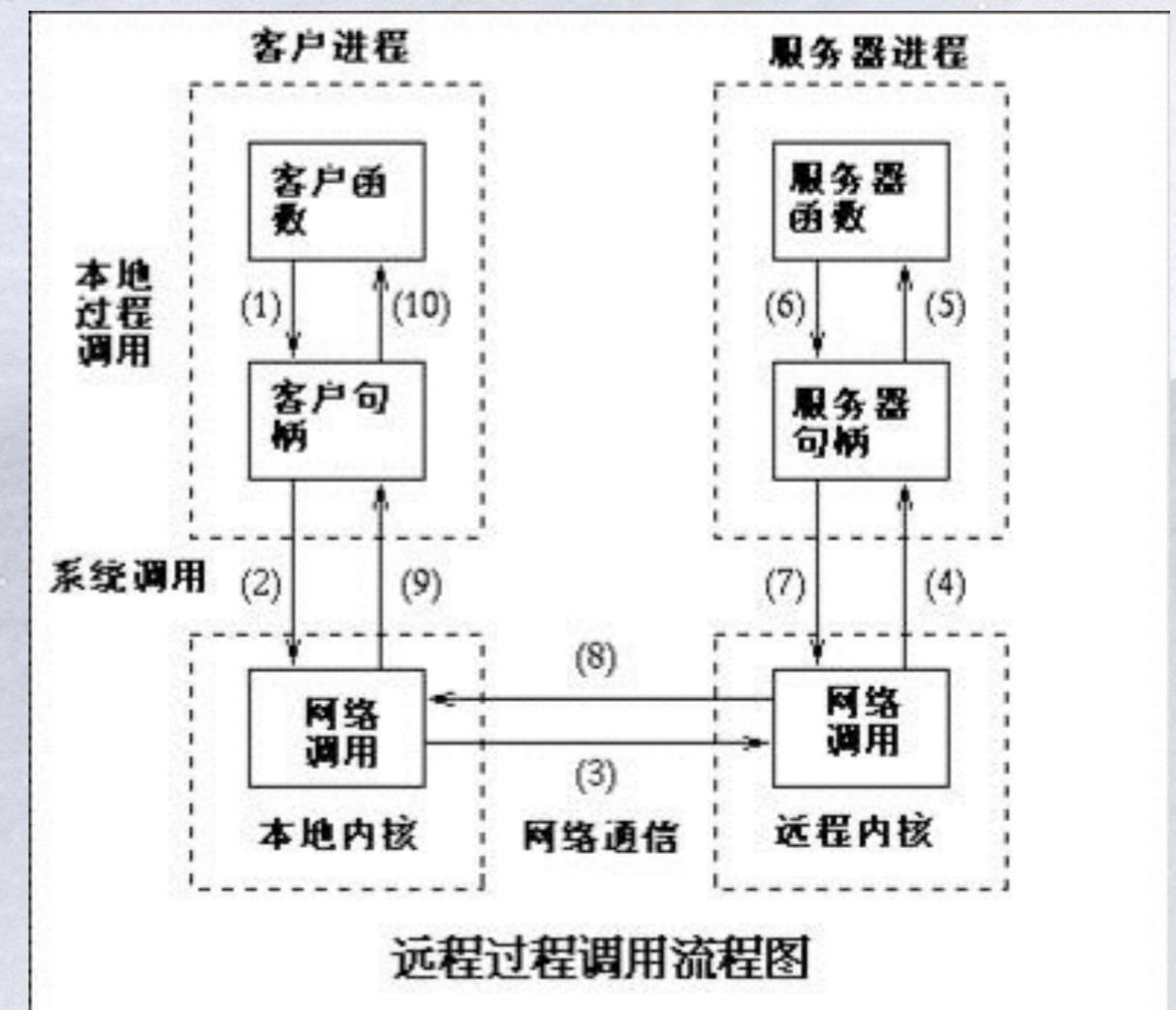
- ❖ Remote Procedure Call (RPC) is an inter-process communication that allows a computer program to cause a subroutine or procedure to execute in another address space.
- ❖ RPC (Remote Procedure Call) goes back at least as far as 1976, when it was described in RFC 707.

RPC – How does it work ?

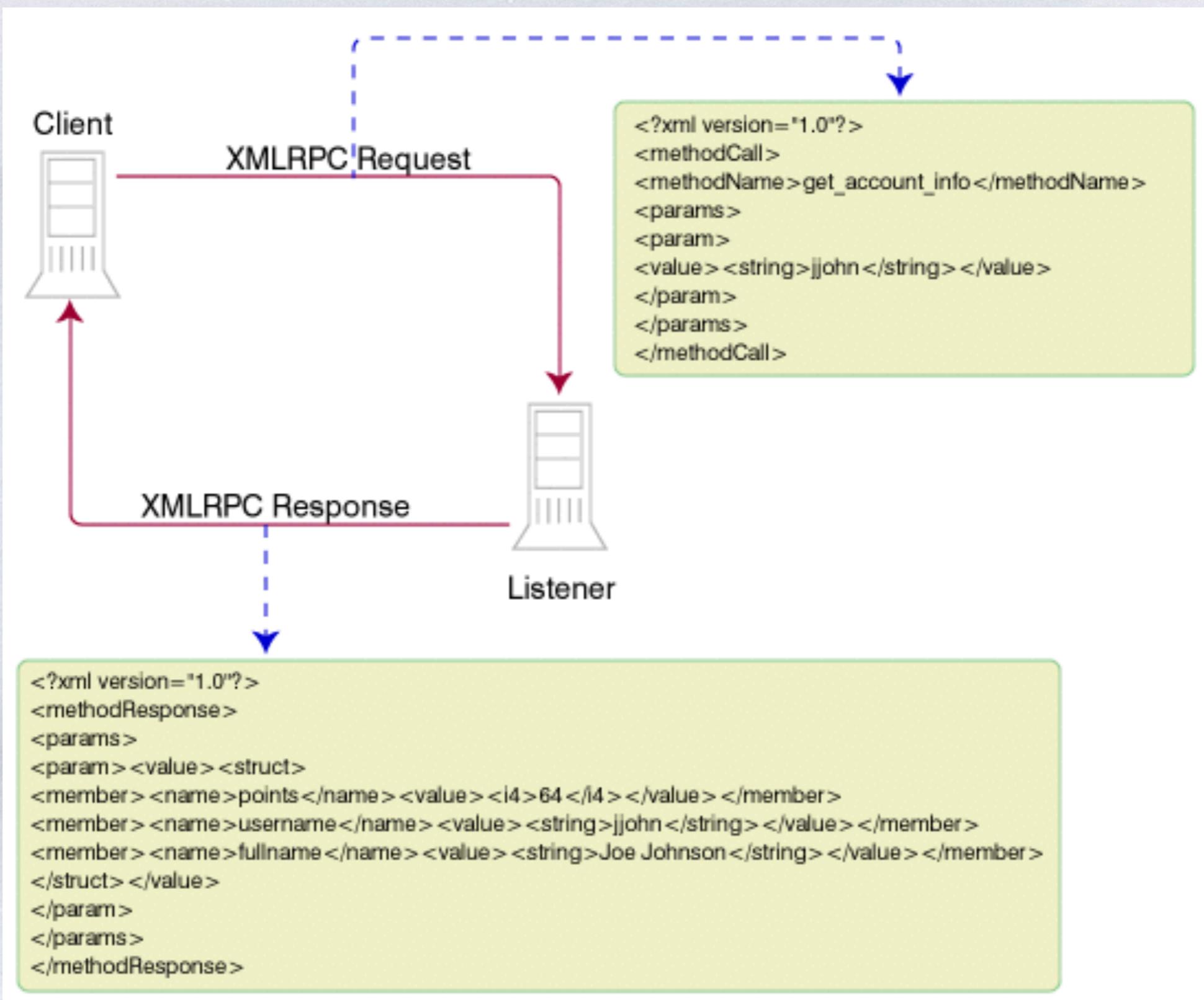
- ❖ A client makes a call to the procedure in a remote machine (server) which has the business logic with appropriate parameters.
- ❖ The server receives the parameters and executes the procedure locally and then transmits the results back to the client.
- ❖ The client receives the results.

RPC – More formally

- * The client calls the Client stub. The call is a local procedure call, with parameters pushed on to the stack in the normal way.
- * The client stub packs the parameters into a message and makes a system call to send the message. Packing the parameters is called marshalling.
- * The kernel sends the message from the client machine to the server machine.
- * The kernel passes the incoming packets to the server stub.
- * Finally, the server stub calls the server procedure. The reply traces the same in other direction.



XML RPC



JSON

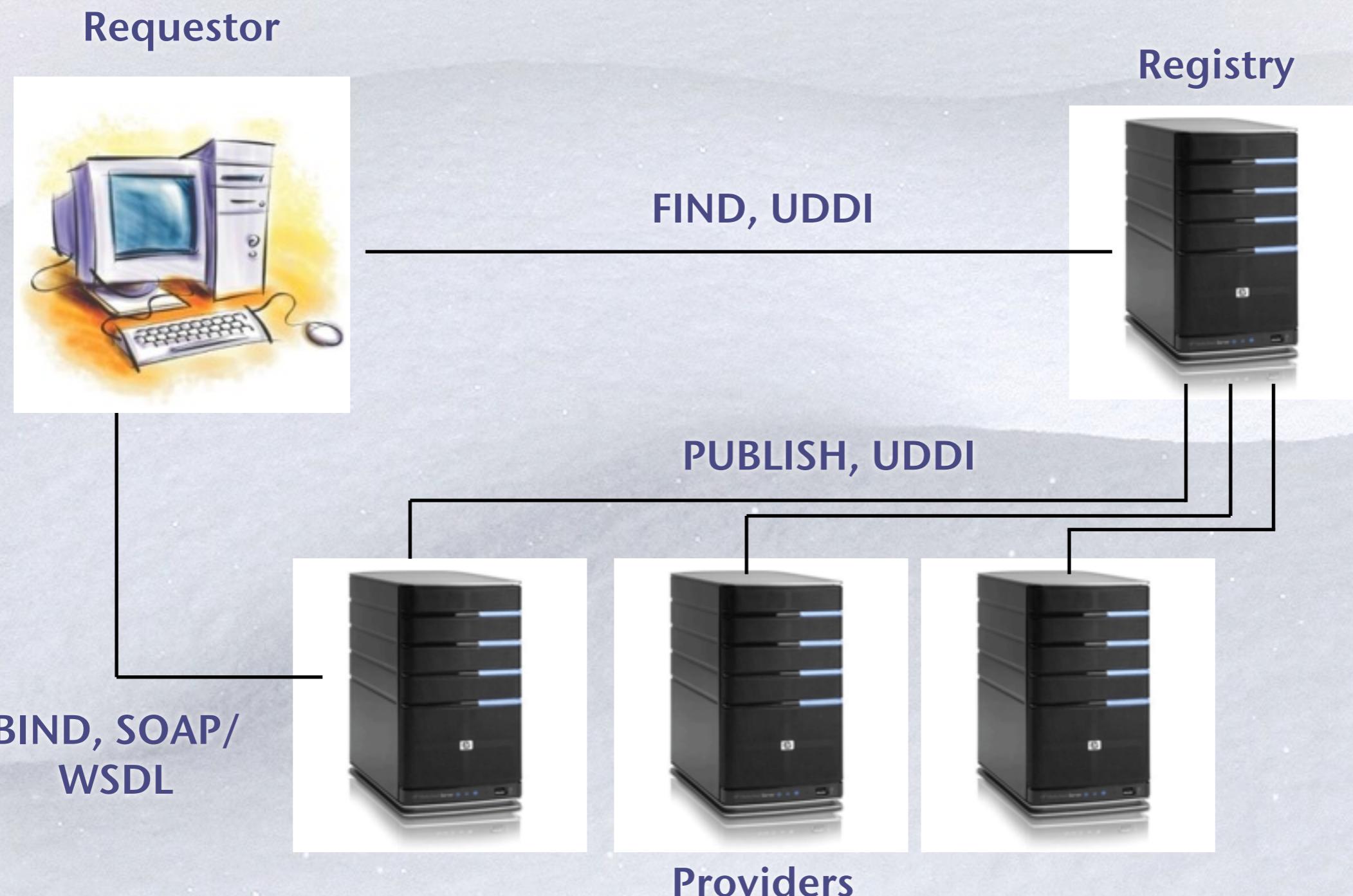
- * Object: collection of comma separated Attributes
 - * var object = { }
- * Attributes: name-value pairs
 - * "name": "Duke"
 - * "age": 10
- * Array: List of comma separated objects
 - * [{"name": "Duke", "age": 10}, {"name": "Tux", "age": 20}]
- * Data types
 - * (String, Number, boolean, array, object, null)

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "age": 25,  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": "10021"  
  },  
  "phoneNumber": [  
    { "type": "home", "number": "212 555-1234" },  
    { "type": "fax", "number": "646 555-4567" }  
  ]  
}
```

Service Oriented Architecture

- ❖ As per Wikipedia, SOA is a flexible set of design principles used during the phases of systems development and integration in computing.
- ❖ A system based on a SOA will package functionality as a suite of interoperable services that can be used within multiple separate systems from several business domains.

Service Oriented Architecture



Components of SOA

* Participants:

- * Provider
- * Registry (broker)
- * Requestor

* Interactions:

- * Publishing
 - * Direct
 - * HTTP GET request
 - * Dynamic discovery
- * Service location (finding)
- * Binding

SOA: Publishing (dynamic)

Provider



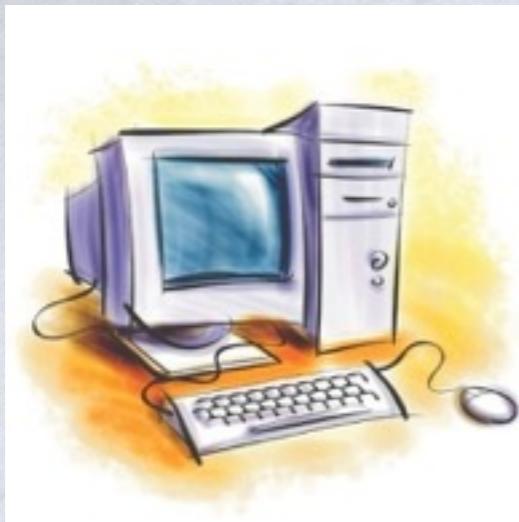
Registry



Here is my
service !

SOA: Locating Service

Requestor



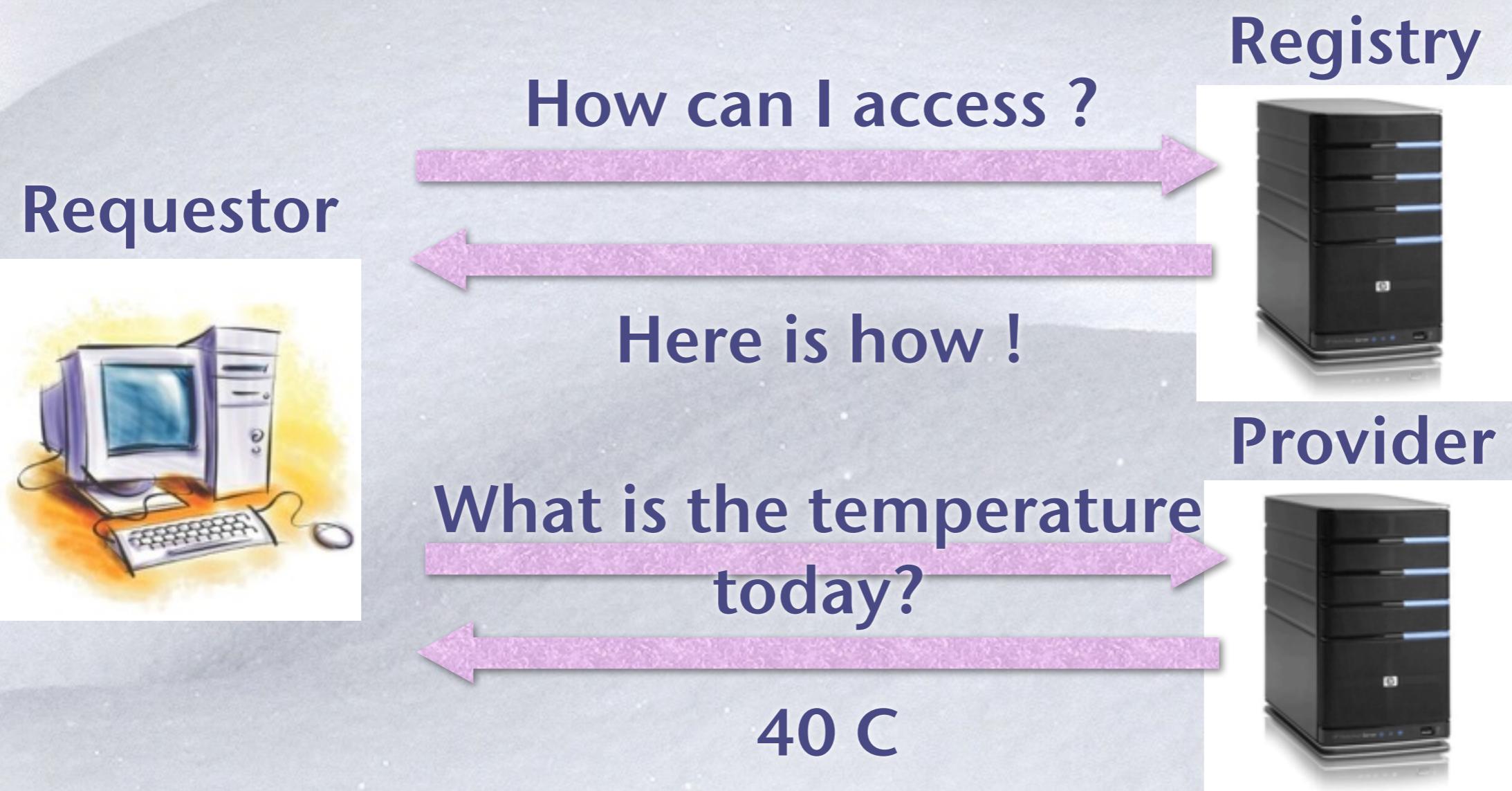
Registry



Where is the
service?

Find it here !

SOA: Binding



What is REST?

- ❖ REST stands for Representational State Transfer.
- ❖ Its basically a stateless communication protocol relying on lightweight protocols like HTTP.
- ❖ It is basically an architectural design methodology.

REST: The other verbs

- ❄ GET to retrieve information
- ❄ POST to add new information, showing its relation to old information
- ❄ PUT to update information
- ❄ DELETE to discard information

RESTful Application Cycle

Resources are identified by URIs



Clients communicate with resources via requests using a standard set of methods



Requests and responses contain resource representations in formats identified by media types



Responses contain URIs that link to further resources

REST: Give Everything an ID

ID is a URI

- * <http://example.com/widgets/foo>
- * <http://example.com/customers/bar>
- * <http://example.com/customers/bar/orders/2>
- * <http://example.com/orders/101230/>
customer

Characteristics of a Web Service

- ❖ XML-based
- ❖ Loosely coupled
- ❖ Coarse-grained
- ❖ Ability to be synchronous or asynchronous
- ❖ Supports Remote Procedure Calls
- ❖ Supports document exchange

Major Technologies for WS

❖ SOAP – Simple Object Access Protocol

❖ WSDL – Web Services Definition/
Description Language

❖ UDDI –Universal Description, Discovery,
and Integration

SOAP

- ❖ SOAP stands for Simple Object Access Protocol. It is based on XML.
- ❖ It is a wired protocol and was defined to be highly interoperable.
- ❖ As of today there are extensive libraries support in virtually all languages to support SOAP based communication.

SOAP: Layers

SOAP-ENV:Envelope

SOAP-ENV:Header

SOAP-ENV:Body

SOAP: Example

POST /InStock HTTP/1.1

Host: www.example.org

Content-Type: application/soap+xml; charset=utf-8

Content-Length: 299

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:GetStockPrice xmlns:m="http://www.example.org/stock">
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Creating a SOAP message

```
<student>
  <roll>120</roll>
  <firstName>Kiran</firstName>
  <lastName>Reddy</lastName>
  <grades>
    <english>A</english>
    <maths>B</maths>
  </grades>
</student>
```



```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://
www.w3.org/2003/05/soap-envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <student>
      <roll>120</roll>
      <firstName>Kiran</firstName>
      <lastName>Reddy</lastName>
      <grades>
        <english>A</english>
        <maths>B</maths>
      </grades>
    </student>
  </soap:Body>
</soap:Envelope>
```

SOAP: Envelop

- ❖ The SOAP envelope declaration is simply the outermost XML tag that delineates the boundaries of the SOAP document.
- ❖ The following envelope tag shows three required attributes, which specify the namespace and the schema to be used for this envelope:

```
<SOAP-ENV:Envelope  
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"  
    xmlns:xsd="http://www.w3.org/1999/XMLSchema">  
...  
</SOAP-ENV:Envelope>
```

SOAP: Header & Body

- ❖ The SOAP header and body are syntactically similar.
- ❖ It is simply a place to put directives to the SOAP processor that receives the message.
- ❖ The sending and receiving parties need to agree on which elements go there and what they mean.

SOAP: Fault

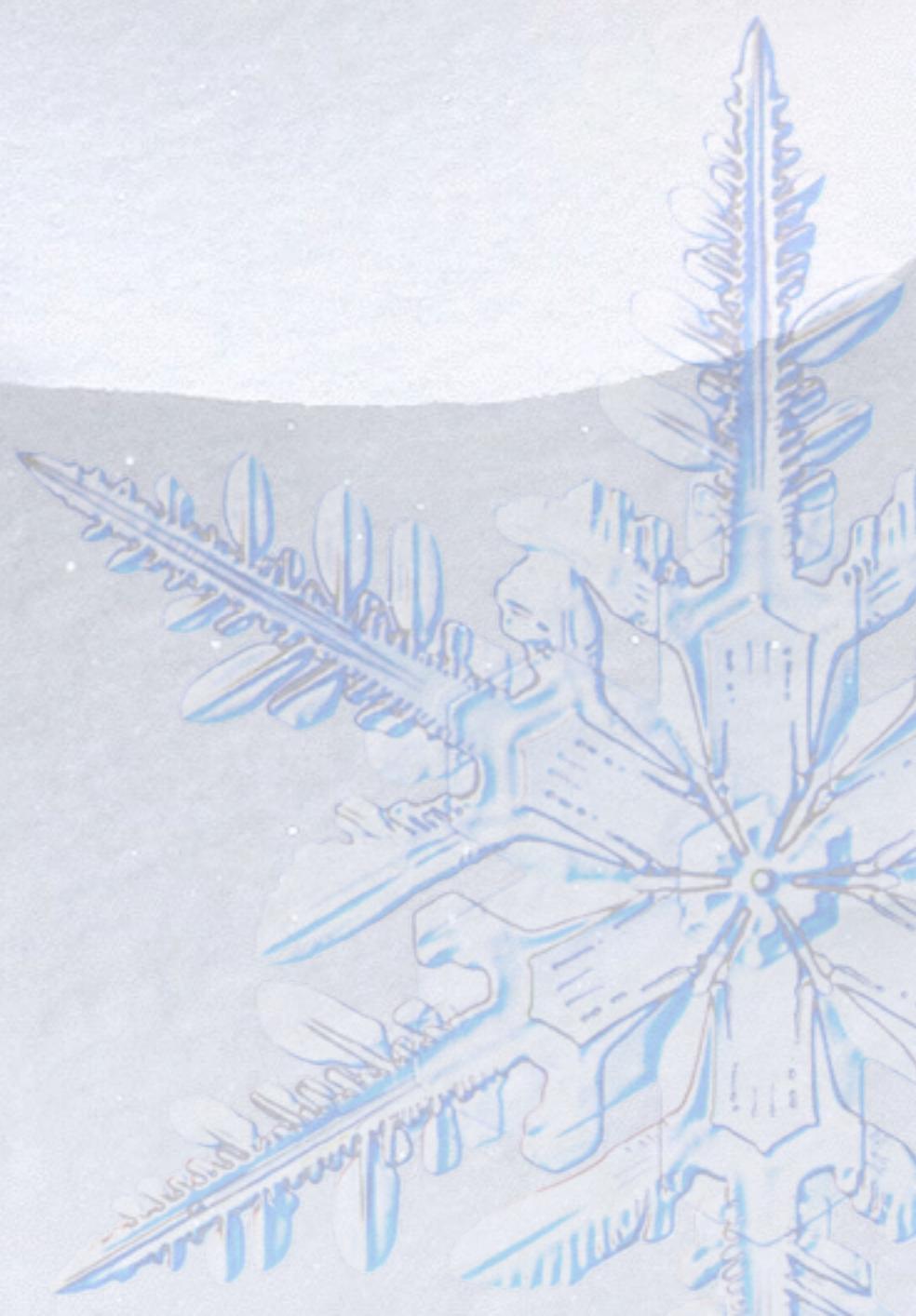
- * SOAP errors are handled using a specialized envelope known as a Fault Envelope.
- * If an error occurs while the server processes a SOAP message, it constructs a SOAP Fault and sends it back to the client.

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">

  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Test Fault</faultstring>
      <faultactor>/soap/servlet/rpcrouter</faultactor>
      <detail>
        <stackTrace>[SOAPException: faultCode=SOAP-ENV:Server;
          msg=Test Fault]
          at StockQuantity.getQty(StockQuantity.java:21)
          at java.lang.reflect.Method.invoke(Native Method)
          at org.apache.soap.server.RPCRouter.invoke(RPCRouter.java:146)
          ...
          at org.apache.tomcat.util.ThreadPool$ControlRunnable.run(
            ThreadPool.java:501)
```



Outline

- ❖ Web Services
 - ❖ Mashups
- 

What is a Mashup?

In Food

To crush a particular type of fruits or vegetables until it changes its shape and produce a new dish.

Cambridge

In Music

An audio / video recording that is a composite of samples from other recordings, usually from different musical styles.

Dictionary.com

In Web

Website or web application that combines content from more than one source into an integrated experience.

TheFreeDictionary

What is a Mashup?

❖ Definition : Wikipedia

- * Mashup is a Web application that combines data from two or more sources into a single integrated application.

❖ What's missing?

- * The integration/combination of different sources is not limited to data but also to functionality and layout styles.

❖ A Complete, Accurate definition is required.

Mashup Definition

- ❖ Mashup is a Web-based Application that is created by combining and processing on-line third party resources, that contribute with data, presentation or functionality.
- ❖ Note: Online third party resources refer to any type of resource available in the Internet.
 - API
 - Web feeds
 - Screen Scraping
 - Excel/PDF files etc

What is a Mashup?

In Food

To crush a



Cambridge

In Music

An audio / video

**U2 - Window
In The Skies
(latest album)**

composite of
samples from
usually from
different musical
styles.

Dictionary.com

In Web

Website or web

Digg.com
combines
content from
more than one
source into an
integrated
experience.

TheFreeDictionary

Housingmaps.com

* History

- Creation of Paul Rademacher.
- One of the first Mashups created.

* Whats being Combined?

- List out houses, apartments, and rooms that are for sale or rent from Craigslist and displays them on a Google map.

* Housingmaps = Craigslist + Google Maps

www.housingmaps.com

[For Rent](#) [For Sale](#) [Rooms](#) [Sublets](#)

Powered by [craigslist](#) and [Google Maps](#)
(this site is in no way affiliated with craigslist or Google)

City: **Boston**

Price: \$1500 - \$2000

[Show Filters](#) New [Refresh](#) [Link](#)

[About / Feedback](#)

Egremont Rd & Mt Hood Rd
Boston

617-990-7715

[Map](#) [Satellite](#) [Hybrid](#)

| pics | price | bd | description | city | date |
|------|--------|-----|--|------------|------|
| | \$1500 | 2bd | Charming 2-Bedroom | Arlington | 5/31 |
| | \$1600 | 1bd | Spacious One Bedroom with balcony in well-maintained Prewar building | Brookline | 5/31 |
| | \$1500 | 3bd | Small Upscale Home Steps To The Beach Year Round Or Seasonal Weekly~ | 18th St | 5/30 |
| | \$1700 | 2bd | Heart of Cleveland Circle | Brighton | 5/30 |
| | \$1500 | 3bd | 1.5 bath Duplex | Framingham | 5/30 |
| | \$1945 | 2bd | Beautiful and Spacious Two Bedroom Handicap Unit! | Quincy | 5/30 |
| | \$1850 | 2bd | Showing Today 5:30-6:15* 10 mins from Harvard, pets Ok | Somerville | 5/30 |
| | \$1600 | 2bd | BU Medical district sunny apt | Boston | 5/30 |
| | \$1500 | 2bd | Beautiful 5 room, 2 bed w/hardwoods, laund,storage & pkg | Melrose | 5/30 |
| | \$1600 | 1bd | Luxury without the Price! *** | Quincy | 5/30 |
| | \$2000 | 4bd | 4 Bedroom Apartment near T | Newton | 5/30 |
| | \$1500 | 2bd | Luxury Loft, Near Lake, Avl Aug 1, No Fee | Wakefield | 5/30 |

Why Mashups are Popular?

- ❖ Even non technical users can create Mashups without any programming knowledge.
- ❖ Explosive growth in “user-generated content”
- ❖ Wide deployment of XML web services.
- ❖ Increased broadband access.
- ❖ Wider conceptualization of the Internet as a platform (“Web 2.0”)
- ❖ Lot of user friendly tools are available.
 - * Microsoft Popfly
 - * Dapper

Classification

- ❖ Explored various Mashups available on ProgrammableWeb.
- ❖ Studied existing Mashup categorization models.
- ❖ They are:
 - * Market Overview of Enterprise Mashup Tools: By Hoyer, V., and Fischer, M
 - * A New Way of Providing Web Mapping and GIS Services.: By Li, S., and Gong, J
 - * Is IBM making enterprise mashups respectable? ZDNet Blog 2006.
 - Available: <http://blogs.zdnet.com/Hinchcliffe/?p=49&tag=nl.e622>

Classification contd..

- * Classified Mashups based on following four questions:
 1. What to Mash up?
 2. Where to Mash up?
 3. How to Mash up?
 4. For Whom to Mash up?.

1. What to Mash up?

- ❖ Depending on the sort of assets being **combined or integrated**, Mashups are assigned to one of the following three categories:

I. Presentation Mashups:

- * Focuses on retrieving information and layout of different Web sources without regarding the underlying data and application functionality.
- * The creation of a presentation mashup requires little or no knowledge of programming languages.
- * Example:
 - * Pre-built widgets that can be drag and drop to common user interface

1. What to Mash up?

2. Data Mashups:

- * Merges data provided by different sources into one content page.
- * The user mixes data from multiply sources and customizes the data flow of for example the Web page containing data from different sources.
- * Different sources are: Web Services, Feeds, and HTML etc.

3. Functionality Mashups

- * Combines data and application functionality provided by different sources to a new service.
- * The functionalities are accessible via APIs.

2. Where to Mash up?

- ❖ Mashups can be distinguished depending on the **location** where they are mashed up.
 1. Server-side Mashups integrate resources on the server.
 2. Client-side Mashups integrate resources on the client, often a browser.
- ❖ Usually a mixture of Client-side and Server-side applications is used for the creation of Mashups.

3. How to Mash up?

- * Depending on the modality the resources are integrated or combined to one representation.
- * Extraction Mashup: data wrapper collecting and analyzing resources from different sources and merging the resources to one content page.
- * Flow Mashup: user customizes the resource flow of the Web page combining resources from different sources.

4. For Whom ?

- ❖ Target audience the Mashups are created for and addressed to:
 - * Consumer Mashups: For public use
 - * Combines resources (e.g., layout or data) from different public or private sources in the browser and organizes it through a simple browser based user interface.
 - * Enterprise Mashups: For Business use
 - * Merges multiple resources of systems in an enterprise environment. (e.g., data and application functionality).
 - * Requires considering security, governance or enterprise policies.

Classification of Mashups

What?

- Presentation Mashups
- Data Mashups
- Functionality Mashups

How?

- Extraction Mashups
- Flow Mashups

Where?

- Server – side Mashups
- Client – side Mashups

For Whom?

- Consumer Mashups
- Enterprise Mashups
-

Tools

- ❖ Several tools have been published that provide functionalities for building, storing and publishing Mashups.
- ❖ The range of these Mashup tools spans from open- source tools to highly-cost license tools.
- ❖ Some of the vendors offer a coding editors.
- ❖ While others focus on users with no programming skills that provide easy-to-use access and application to their tool suites.

Tools contd..

* Some examples :

- * Yahoo Pipes
- * IBM MashupCenter
- * Intel Mashmaker
- * Microsoft Popfly
- * Derri Pipes
- * Dapper

- Evaluated how these tools can be used to create Mashups and were classified.

| | Presentation | Data | Functionality | Extraction | Flow | Consumer | Enterprise |
|---------------------------------|--------------|------|---------------|------------|------|----------|------------|
| Apatar | | ✗ | | | ✗ | | ✗ |
| Data Mashups | ✗ | ✗ | | | ✗ | | ✗ |
| Dapper | ✗ | | | ✗ | | ✗ | |
| DERI pipes | | ✗ | | | ✗ | ✗ | |
| Grazer | ✗ | | | ✗ | | ✗ | |
| IBM InfoSphere MashupHub | | ✗ | | ✗ | | | ✗ |
| Intel MashMaker | ✗ | | | | ✗ | ✗ | |
| JackBe Presto | | ✗ | ✗ | | ✗ | | ✗ |
| Microsoft Popfly | ✗ | ✗ | | | ✗ | ✗ | |
| Openkapow | ✗ | | | ✗ | | ✗ | |
| Procession | | ✗ | ✗ | | ✗ | | ✗ |
| Rssbus | ✗ | | | ✗ | ✗ | | ✗ |
| Serena Mashup Suite | | ✗ | ✗ | | ✗ | | ✗ |
| Snap Logic | | ✗ | | | ✗ | | ✗ |
| TIBCO PageBus | | ✗ | | | ✗ | | ✗ |
| Yahoo! Pipes | | ✗ | | | ✗ | ✗ | |

Challenges

❖ Cataloguing.

- * Some Web pages are already available that list Mashups and provide an interface for searching of mashups such as programmableweb.com.
- * Mashup creators can insert their mashups in the list and share their Mashups with others.
- * But what is missing is a directory that stores and catalogues the mashups in a consistent way.

❖ Making Data Web enabled

- * Currently a lot of data and functionalities are not set up on the Web and they are not accessible via feeds, HTML or Web services.
- * To make more resources “Web-enabled” require formats and tools that facilitate an efficient access and connection of resources to the Web.

Challenges contd..

❖ Security and Identity

- * Requires mechanisms to control the user connection and the data security.

❖ Sharing and Re-using

- * Vendors of Mashup tools should provide mechanisms to allow end-users sharing their built Mashups with others to facilitate the reuse of pre-built Mashups.
- * Easy-to-use access to Mashups.
- * Efficient Mashup search functionalities
- * lightweight formats that enable even for non-programmers a smooth Mashup reuse.

Challenges contd..

❖ Version Control Mechanisms

- * Mashups consist of different resources collected from various sources.
- * Resource owners are responsible for their content and can change and update its content whenever they regard it as necessary.
- * To keep the content up-to-date a version control mechanism is required that automatically informs the Mashup owner about updates.

❖ Trust Certificates

- * No certification mechanisms exist that guarantee end-users the trustworthiness of the Mashup.

Conclusion

- ❖ Mashups are suitable to build novel Web applications and to create new forms of visualization without little knowledge of programming languages.
- ❖ Further research is especially needed in the fields of version control mechanisms, Mashup certification, Mashup quality and data integrity.

Thanks!!!

