Web Services

Web Services

- A web service is any service that is available over the Internet, uses a standardized messaging system, and is not tied to any one operating system or programming language.

Figure 1-1. A basic web service

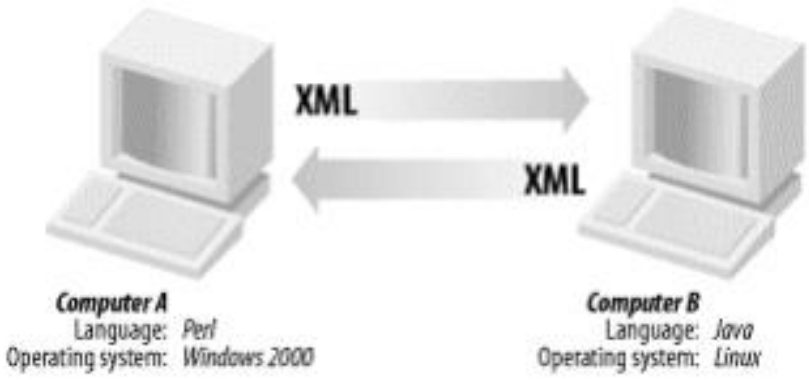
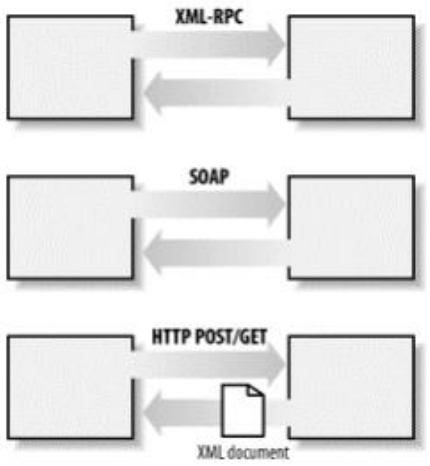


Figure 1-2. XML messaging for web services



- There are several alternatives for XML messaging. For example, you could use XML Remote Procedure Calls (XML-RPC) or SOAP, both of which are described later in this chapter. Alternatively, you could just use HTTP GET/POST and pass arbitrary XML documents.

Additional properties of WS

- **A web service should be self-describing**.
  - Publish a public interface to the service.
  - At a minimum, service should include human-readable documentation so that other developers can more easily integrate the service.
  - If it is a SOAP service, then it should also ideally include a public interface written in a common XML grammar. The XML grammar can be used to identify all public methods, method arguments, and return values.
- **A web service should be discoverable**.
  - Should be a relatively simple mechanism for the service to publish this fact.
  - Should be some simple mechanism whereby interested parties can find the service and locate its public interface.

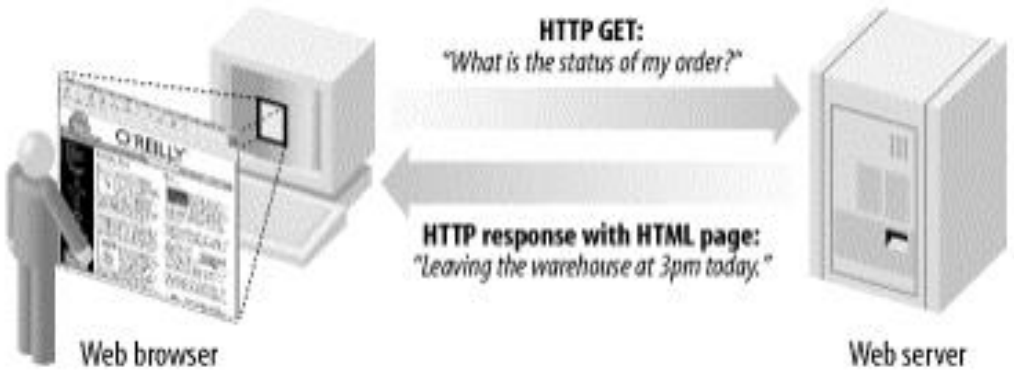
Summary of WebService

- Is available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system (SOAP over HTTP, SOAP over TCP, XML-RPC over HTTP, etc)
- Is not tied to any one operating system or programming language
- Following properties are desirable but may or may not be there
  - Is self-describing via a common XML grammar (WSDL)
  - Is discoverable via a simple find mechanism (UDDI)

The Web Today: Human-centric Web

- For example, Widgets, Inc. sells parts through its web site, enabling customers to submit purchase orders and check on order status.
- To check on the order status, a customer logs into the company website via a web browser and receives the results as an HTML page.
- Humans are the primary actors initiating most web requests

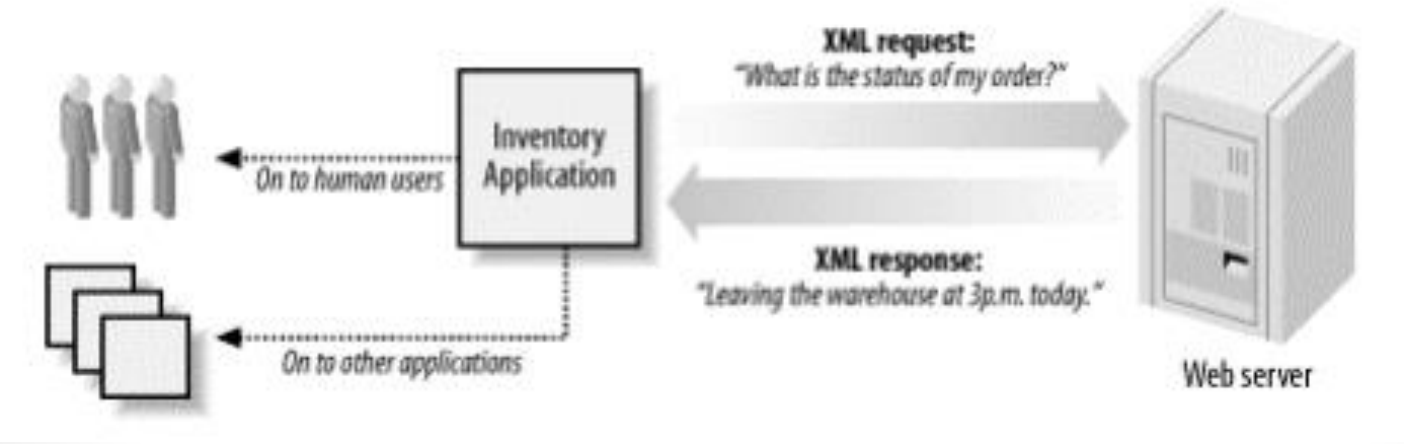
Figure 1-3. The human-centric Web



Web Services: The Application-Centric Web

- Conversations can take place directly between applications as easily as between web browsers and servers.
- Eg. An inventory application can query Widgets, Inc. on the status of all orders. The inventory system can then process the data, manipulate it, and integrate it into its overall supply chain management software.

Figure 1-4. The application-centric Web



The Web Services Vision: The Automated Web

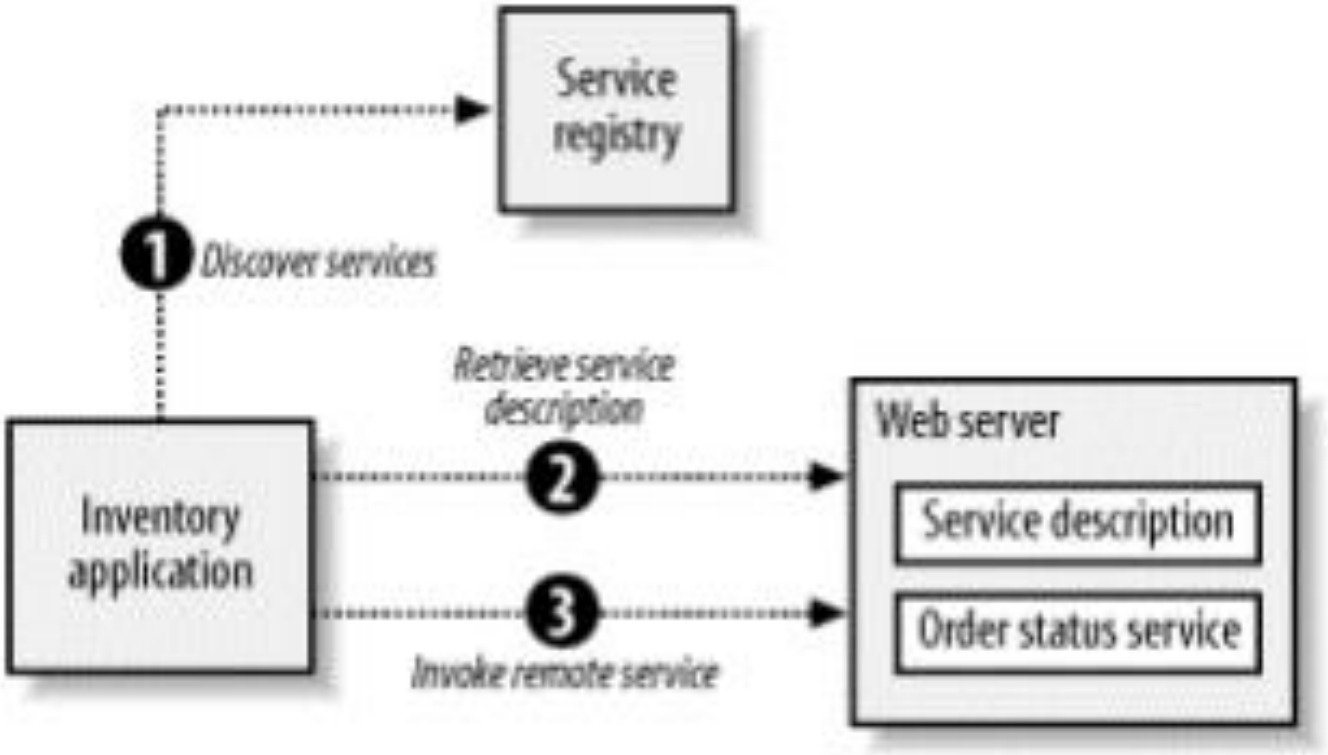
For example, consider the case of MegaElectric (ME). ME wants to buy parts from Widgets, Inc. and also wants to seamlessly integrate order status into a unified inventory system. At some point in the future, ME will be able to buy software that automates this entire process. Here's how it might work :

1. The inventory application wakes up and connects to a centralized directory of web services: "Does Widgets, Inc. provide an order status service?" The directory returns information on Widgets, Inc.'s service and includes a pointer to the service description.

2. The inventory application connects to Widgets, Inc. and retrieves the service description.

3. The service description file includes complete details about how to connect to the specified service. The inventory application can therefore automatically invoke the order status service.

The Automated Web

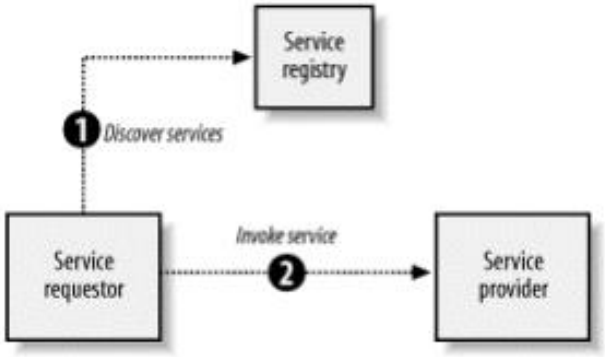


Web Service Architecture

Web Service Roles

- **Service provider**
  - Provider of the web service.
  - Implements the service and makes it available on the Internet.
- **Service requestor**
  - Consumer of the web service.
  - Utilizes an existing web service by opening a network connection and sending an XML request.
- **Service registry**
  - Logically centralized directory of services.
  - Provides a central place where developers can publish new services or find existing ones.

Figure 1-6. Web service roles



Web Service Architecture

Web Service Protocol Stack

Service transport

- Responsible for transporting messages between applications.
- Hypertext transfer protocol (HTTP), Simple Mail Transfer Protocol (SMTP), file transfer protocol (FTP), and newer protocols, such as Blocks Extensible Exchange Protocol (BEEP).

XML messaging

- Responsible for encoding messages in a common XML format so that messages can be understood at either end.
- XML-RPC, SOAP

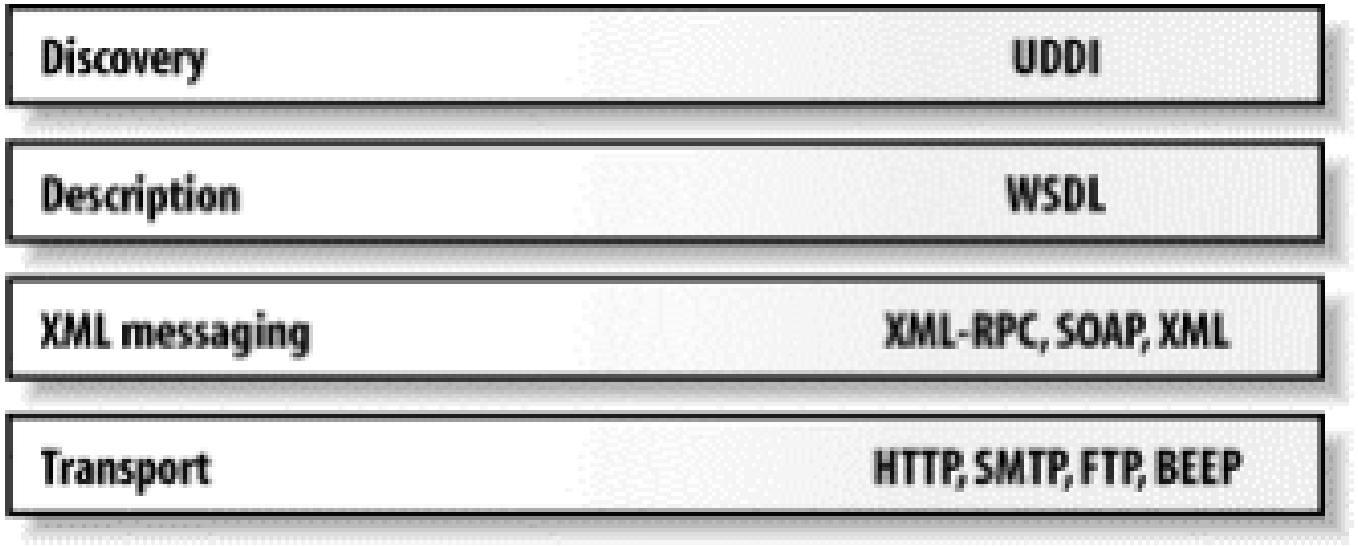
Service description

- Responsible for describing the public interface to a specific web service.
- Web Service Description Language (WSDL).

Service discovery

- Responsible for centralizing services into a common registry, and providing easy publish/find functionality.
- Universal Description, Discovery, and Integration (UDDI).

Web Services Protocol Stack



Architectural Snapshot: The IBM Web Services Browser

Figure 1-8. The IBM Web Services browser

Figure 1-9. Results of web service search

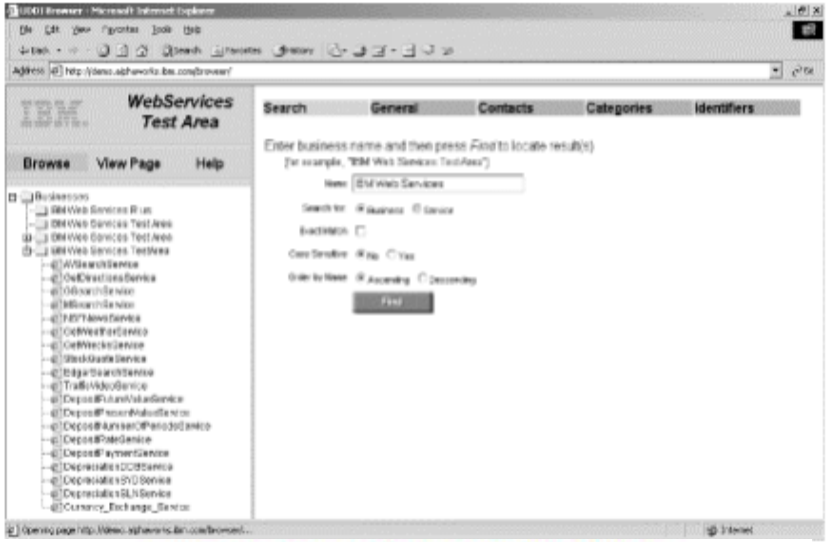


Figure 1-10. Details of the IBM weather service

Figure 1-11. Invoking the IBM weather service