

Introduction to Web Services

- Motivation
- The Automated Web
- XML RPC
- SOAP Messaging
- WSDL Description
- Service Implementation & Deployment
- Further Issues

Web Services

*“a software application identified by a URI, whose interface and bindings are capable of being defined, described, and discovered as XML artifacts.
A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols”*

(W3C Consortium)

For my skiing vacation ...

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

... I search for a place with more than 1m of snow...

😊 All major areas offer up to date information

☹ Don't want to search pages

☹ Don't want to search on pages

☹ Don't understand language (Swiss ;)

😊 Will like an automatic snow information service

My automatic snow service

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

- Should inquire all sites in my preferred areas
- Should interpret/compare all results
- Should display the snowy skiing places



... a snow service ?

Problem 1: How to find sites/information services?

Problem 2: How to locate/extract information from sites?

Problem 3: How to process information (encoding)?

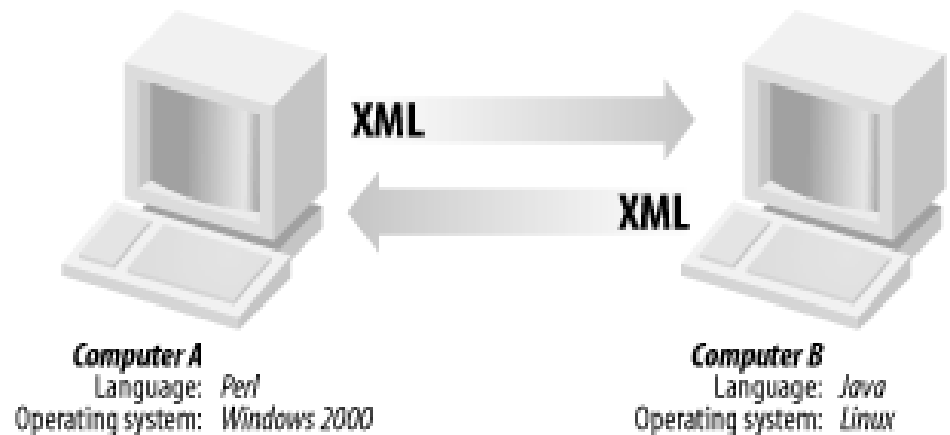
Problem 4: How to understand information (semantics)?

Problem 5: How to interoperate with services
(applications, OSs, languages)?

Requirements

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

- Technological
 - Standards on application layer
 - Standards in encoding + types
 - Network interoperability
- Functional
 - Request messaging
 - Response messaging
 - Discoverable description
- Semantically
 - Understanding of service
 - Understanding of elements
 - Understanding of values



Vision: The Automated Web

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

- Application centric Web based on interoperable standards
- Just-in-time integration of applications and information
- Services discover, contact and employ other services on request or on schedule
- Machine services 'understand' other machine services
- Humans encounter a very high-level interface for distributed knowledge and services

What is a Web Service?

- ① available over the Internet (Intranet)
- ② using a standardized XML messaging system
- ③ not tied to any operating system or programming language
- ④ self-describing via a common XML grammar
- ⑤ discoverable via a simple find mechanism

XML RPC

RPC Request

```
<?xml version="1.0" encoding="UTF-8"?>
<methodCall>
  <methodName>weather.getSnowheight</methodName>
  <params>
    <param><value>St. Moritz</value></param>
  </params >
</methodCall>
```

RPC Response

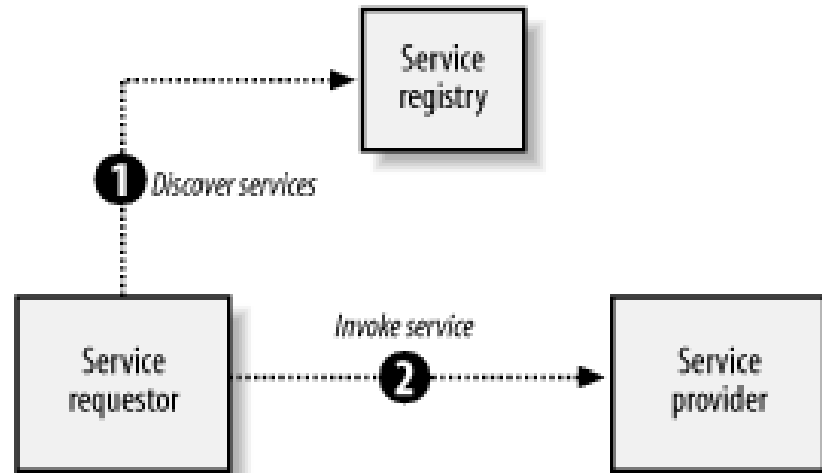
```
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <params>
    <param><value><int>128</int></value></param>
  </params >
</methodResponse>
```

- Easy and simple Web Service:
 - Machine/language independent remote procedure call
 - Fixed vocabulary
 - Fixed collection of (simple) data types
 - Well suited for 'rapid' system integration tasks
- But
 - No service description grammar
 - No schema/namespace support
 - No state management.

Web Service Architecture

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

- Service Provider
- Service Requestor
- Service Registry



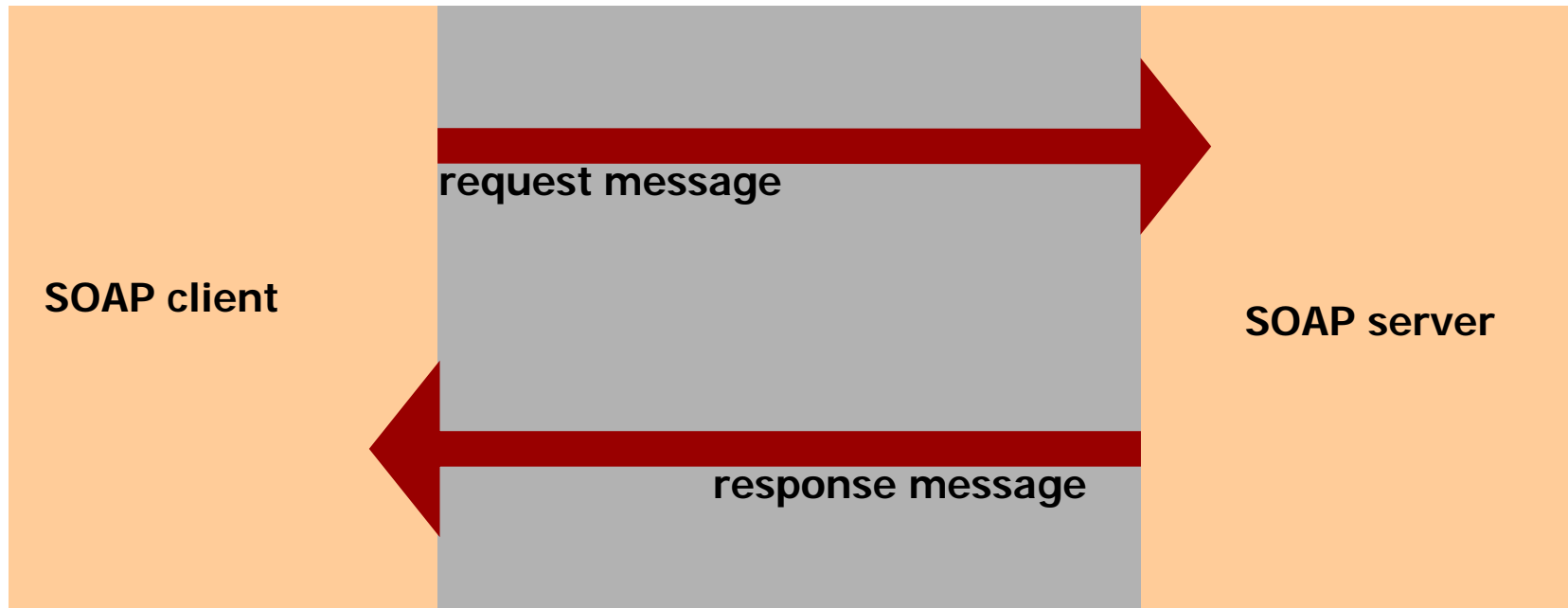
Protocol Stack

| | |
|---------------|-----------------|
| Discovery | UDDI |
| Description | WSDL |
| XML Messaging | XML RPC, SOAP |
| Transport | HTTP, SMTP, FTP |

SOAP

- Simple Object Access Protocol (V 1.2).
- Defined in XML, W3C standard.
- RPC-type via a simple messaging protocol.
- Transports XML coded messages.
- Conventionally used over HTTP.
- Addresses URIs.
- Employs XML namespace und schema.

SOAP Messaging



SOAP messages consist of an Envelope containing

- An optional header
- A message body

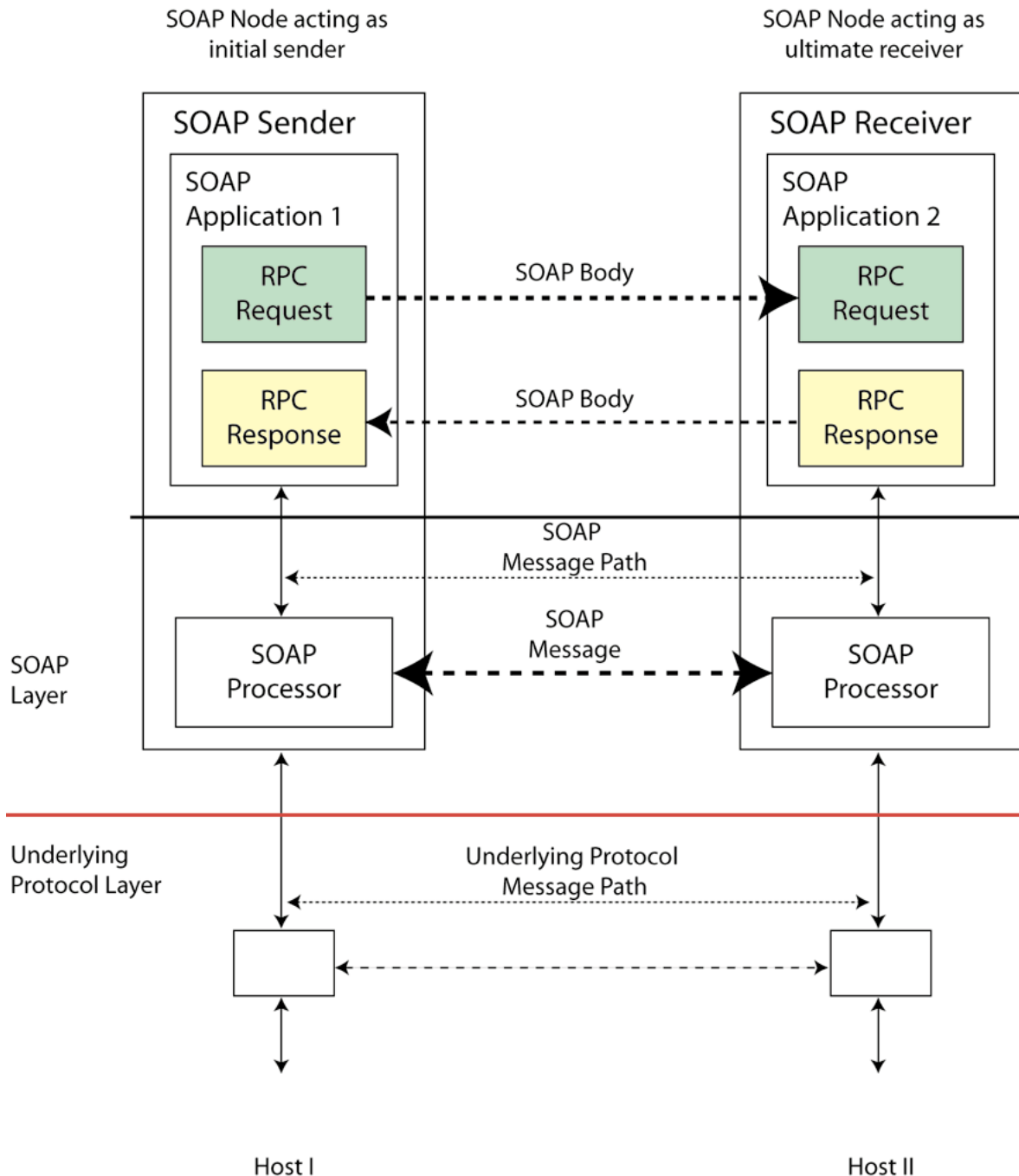
SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2002/06/soap-
  envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema/">

  <SOAP-ENV:Body>
    <my:getSnowRequest
      xmlns:my="http://my.org/snow-services"
      SOAP-ENV:encodingStyle=
      "http://www.w3.org/2002/06/soap-encoding/">
      <town xsi:type="xsd:string">St. Moritz</town>
    </my:getSnowRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2002/06/soap-
  envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema/">
  <SOAP-ENV:Body>
    <my:getSnowResponse
      xmlns:my="http://my.org/snow-services"
      SOAP-ENV:encodingStyle=
      "http://www.w3.org/2002/06/soap-encoding/">
      <return xsi:type="xsd:int">128</return>
    </my:getSnowResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



SOAP Protocol Architecture

WSDL Service Description

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

The Web Service Description Language is an XML Grammar for specifying

- a public interface of a Web Service
- *service access* mechanisms
- *service locations*

From interpreting a WSDL file a client can locate a Web Service and invoke any publicly available functions.

WSDL Example

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="SnowService"
  targetNamespace="http://my.org/snow-services.wsdl"
  xmlns:my="http://my.org/snow/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:my="http:// my.org/snow-services.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema/">

  <message name="getSnowRequest">
    <part name="town" type="xsd:string"/>
  </message>
  <message name="getSnowResponse">
    <part name="hight" type="xsd:int"/>
  </message>
  ...
  ...
</definitions>
```

Creating a Service

1. Create core functionality
2. Create a service deployment descriptor
3. Create a WSDL service descriptor
4. Deploy Service
5. Register new service via UDDI

Hello World Service

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

```
package ws.example.services.hello;

/**
 * Simple webservice saying hello.
 */
public class HelloService {
    public String sayHello(String name) {
        return "Hello " + name + ", nice to meet you!";
    }
}
```

Service Deployment

- Depends on Web Server
- On JBoss:
 - Copy a J2EE conformal .war archive into the deploy directory
 - Register new service

Command:

```
java -cp $AXISCLASSPATH  
org.apache.axis.client.AdminClient -l  
<deploy-host> deploy.wsdd
```

Deployment Descriptor: deploy.wsdd

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

```
<?xml version="1.0" encoding="UTF-8" ?>
<deployment name="helloService"
  xmlns="http://xml.apache.org/axis/wsdd/"

  xmlns:java="http://xml.apache.org/axis/wsdd/providers
  /java">

  <handler name="URLMapper"
    type="java:org.apache.axis.handlers.http.URLMapper"
    />
  <service name="HelloService" provider="java:RPC">
    <parameter name="className"
      value="ws.example.services.hello.HelloService"/>
    <parameter name="allowMethods" value="*" />
  </service>

</deployment>
```

Creating a Service Request

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

1. Find service via UDDI
2. Retrieve service description file
3. Create SOAP client
4. Invoke remote service

Hello World Client

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

```
package ws.example.services.client;

import java.rmi.RemoteException;
import javax.xml.rpc.ParameterMode;
import javax.xml.rpc.ServiceException;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import org.apache.axis.encoding.XMLType;

/* Simple Simple web service client. Calls the method 'String
   sayHello(String name)'.
*/

public class HelloClient {
    /* URL of host that is providing the web services.
       */
    protected static String HOST = "http://myhost.org";
    /* Specifies path to service on web service host.
       */
    protected static String SERVICE_PATH = "/services/HelloService";

    ...
}
```


Hello World Client

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

```
/* Main method that performs web service call.
 * @param args - Declared by default - currently not in use.
 */
public static void main(String[] args) {
    //Create Service
    Service service = new org.apache.axis.client.Service();
    try {
        //try 'Hello' web-service
        Call call = (Call) service.createCall(); //create Call
        call.setReturnType(XMLType.XSD_STRING); //specify return-type
        call.setOperationName("sayHello"); //specify called method
        call.setTargetEndpointAddress(HOST + SERVICE_PATH);
        call.addParameter("name", XMLType.XSD_STRING, ParameterMode.IN);
        //register (passed) parameter for name string .
        String ret = (String) call.invoke(new Object[] { "Charly Brown" });
        //invoke remote method - passing the 'String' value as parameter.
        System.out.println("Webservice says [" + ret + "]);
    } catch (ServiceException ex) {
        System.err.println("Can't create Call - reason [" +
            ex.getMessage() + "]); ex.printStackTrace();
    } catch (RemoteException ex) {
        System.err.println("Remote Invocation Error - reason [" +
            ex.getMessage() + "]); ex.printStackTrace();
    }
}
```

Service Discovery

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

Consists of three categories

White Pages

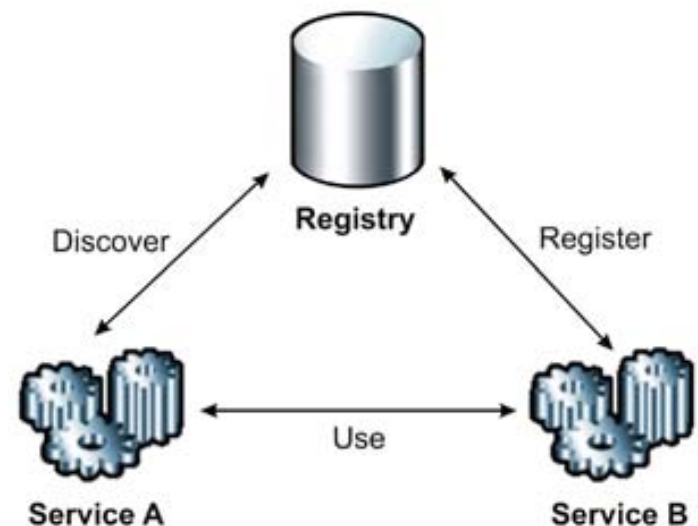
General information about the service provider

Yellow Pages

General classification of the offered service

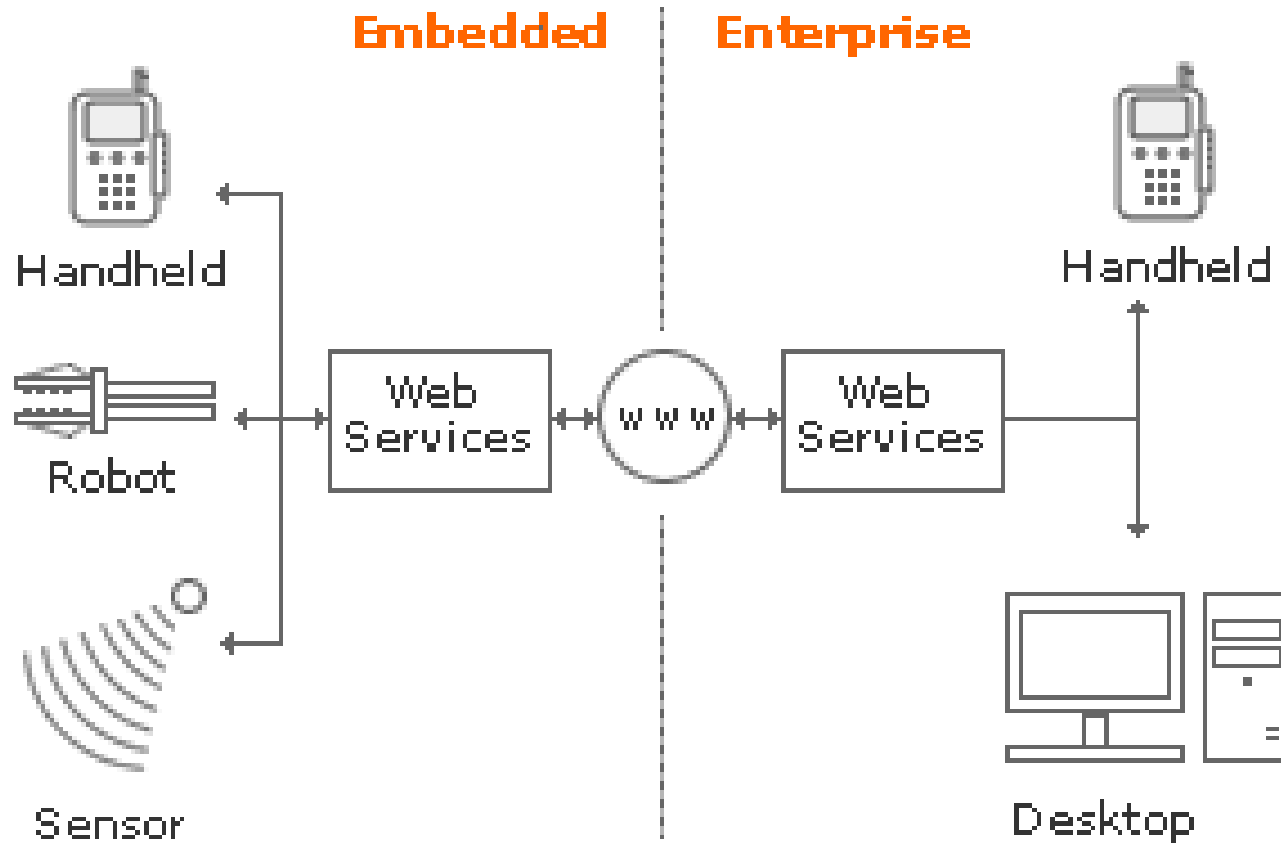
Green Pages

Technical information about the Web Service



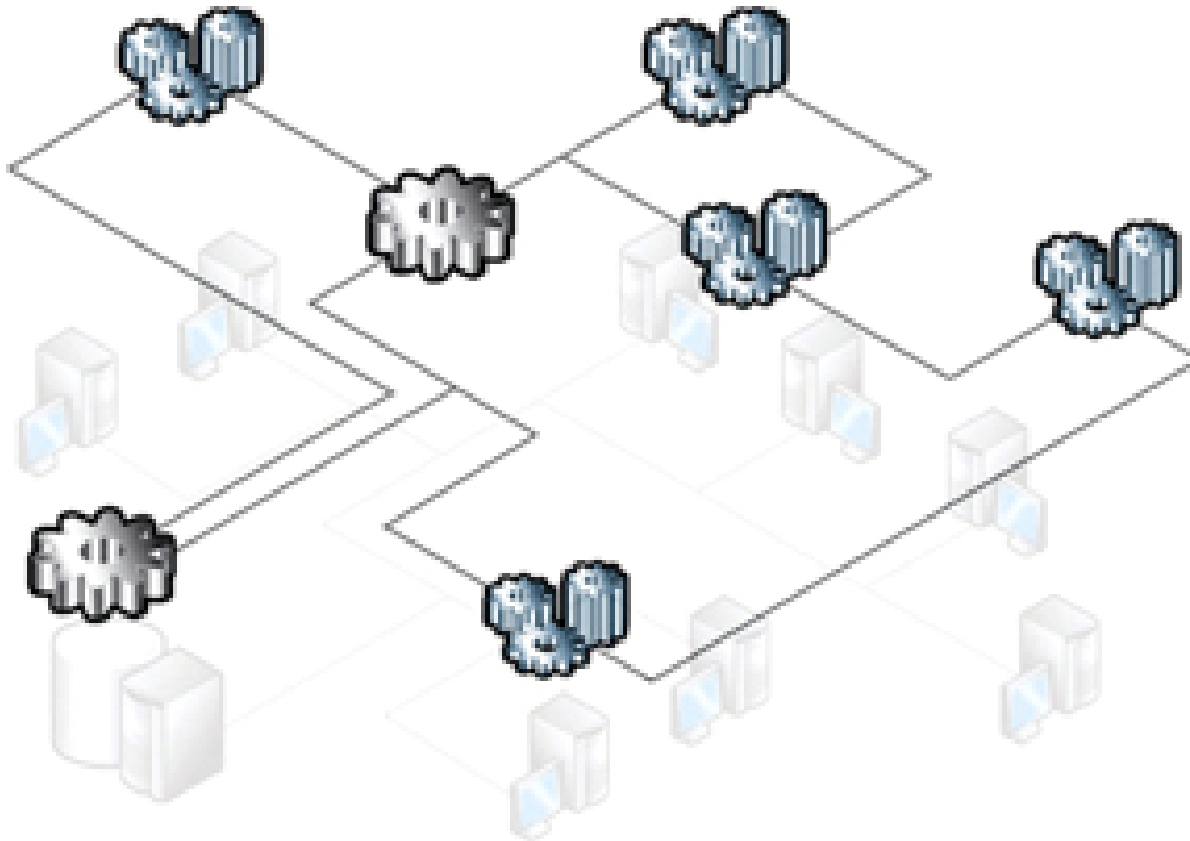
Scenarios

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de



Scenarios: Overlay Networks

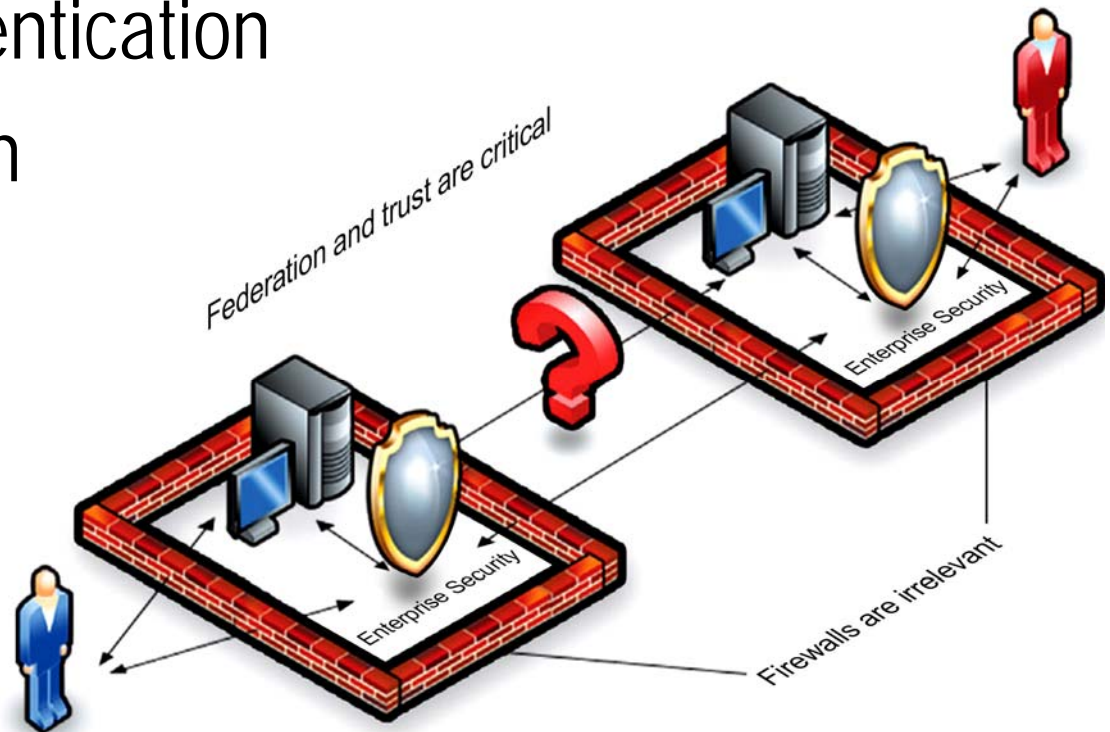
Thomas Schmidt
schmidt@informatik.
haw-hamburg.de



Security Issues

Thomas Schmidt
schmidt@informatik.
haw-hamburg.de

- Mutual authentication
- Message authentication
- Data encryption
- Networking:
Firewalls
become
irrelevant



References

- ↪ The W3C standards <http://www.w3.org>.
- ↪ E. Cerami: *Web Services Essentials*, O'Reilly, Sebastopol, 2002.
- ↪ G. Alonso et. al.: *Web Services*, Springer, Berlin 2004.
- ↪ J. Snell et al.: *Programming Web Services with SOAP*, O'Reilly, Sebastopol, 2002.