
Java
Web Services

Why? WebServices

Why Web Services?

"The Web can grow significantly in power and scope if it is extended to support communication between applications, from one program to another."

- From the W3C XML Protocol Working Group Charter

Presentation Roadmap

Past

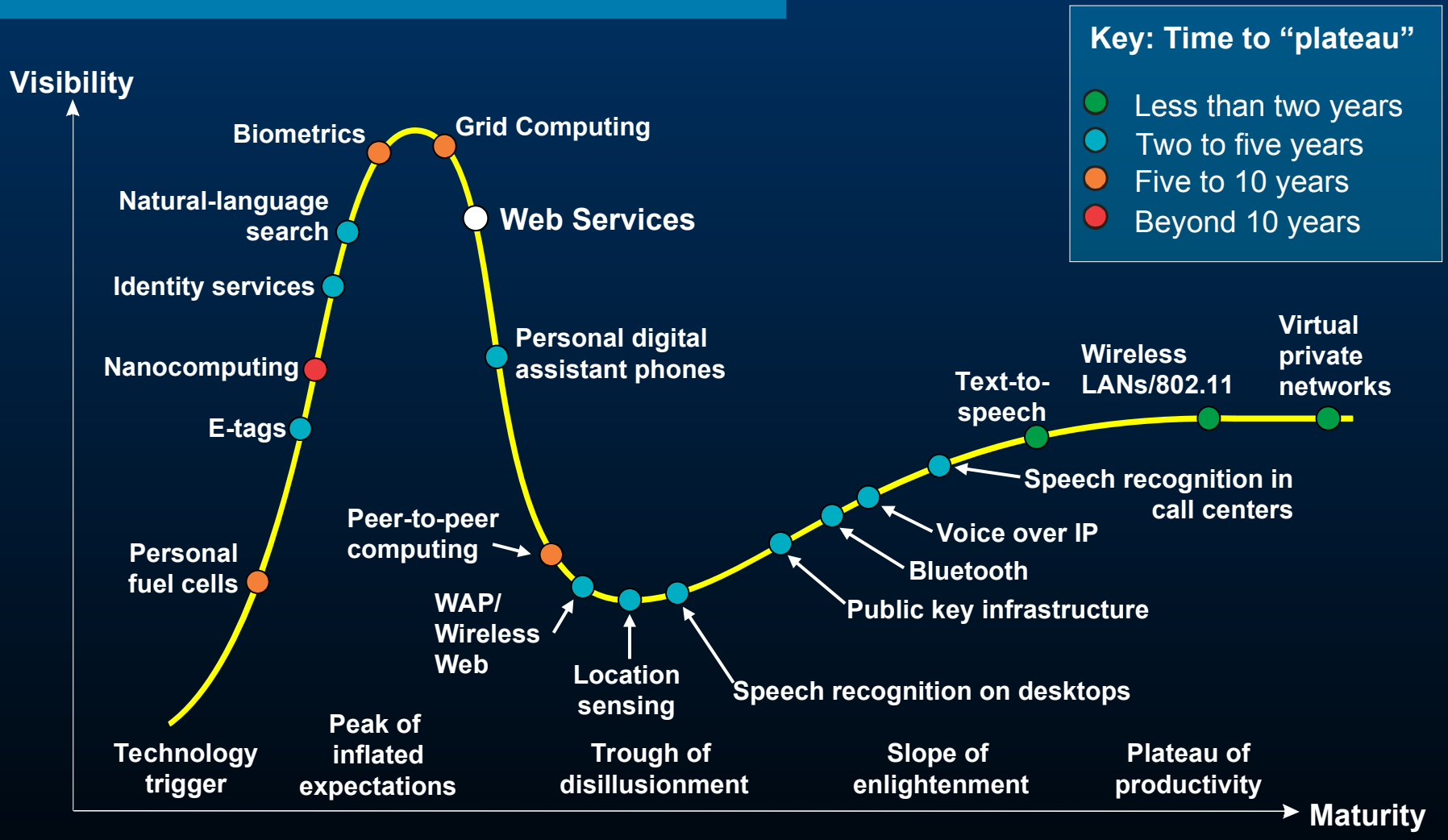
Present

Future

The Hype

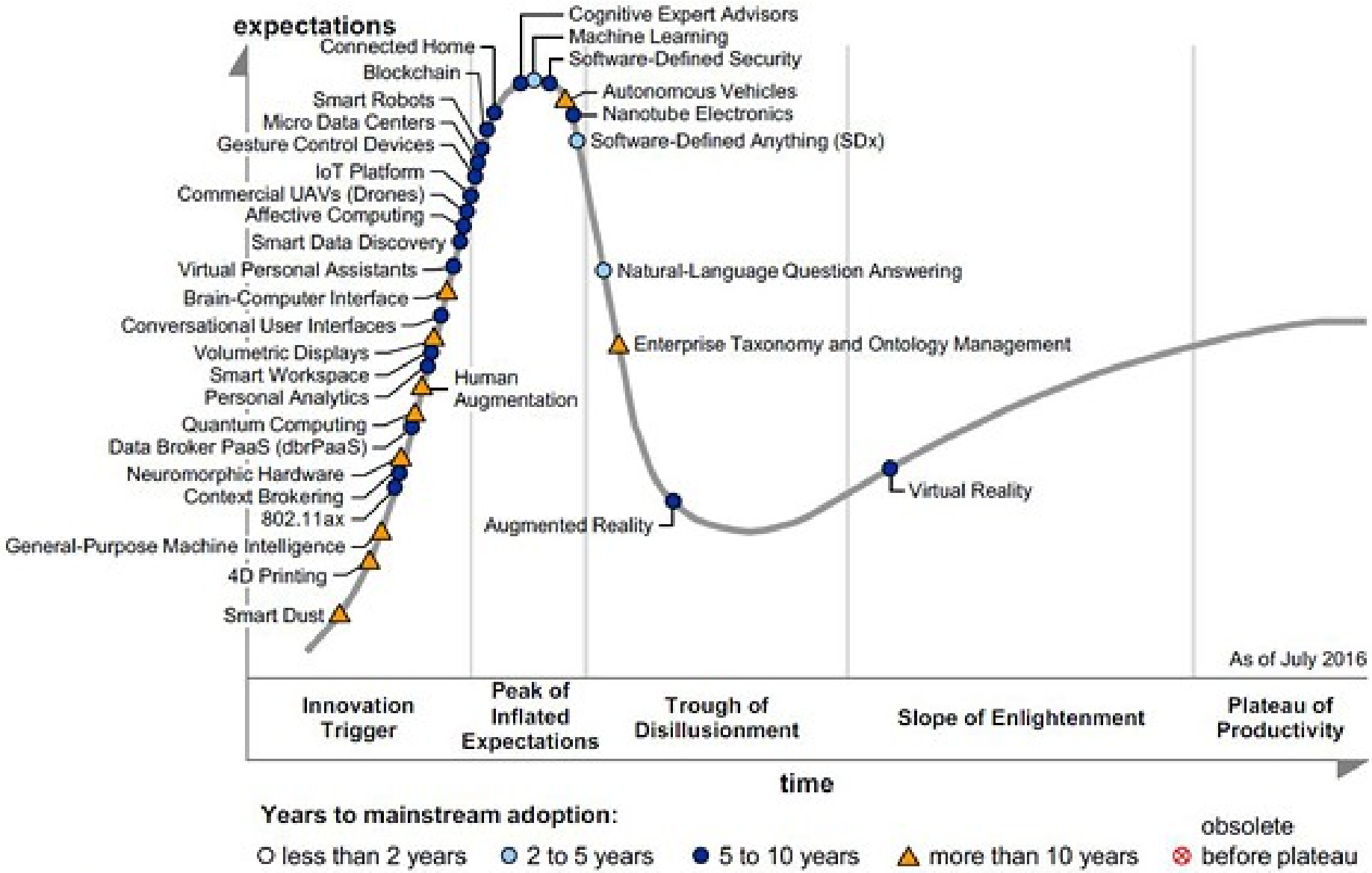
- New paradigm?
- Reason to move platforms?
- Replacement for EDI?

Gartner's 'Hype' Curve 2002



Source: Gartner Group June 2002

Gartner's Hype curve 2016



Source: Gartner (July 2016)

W3C (working group) definition

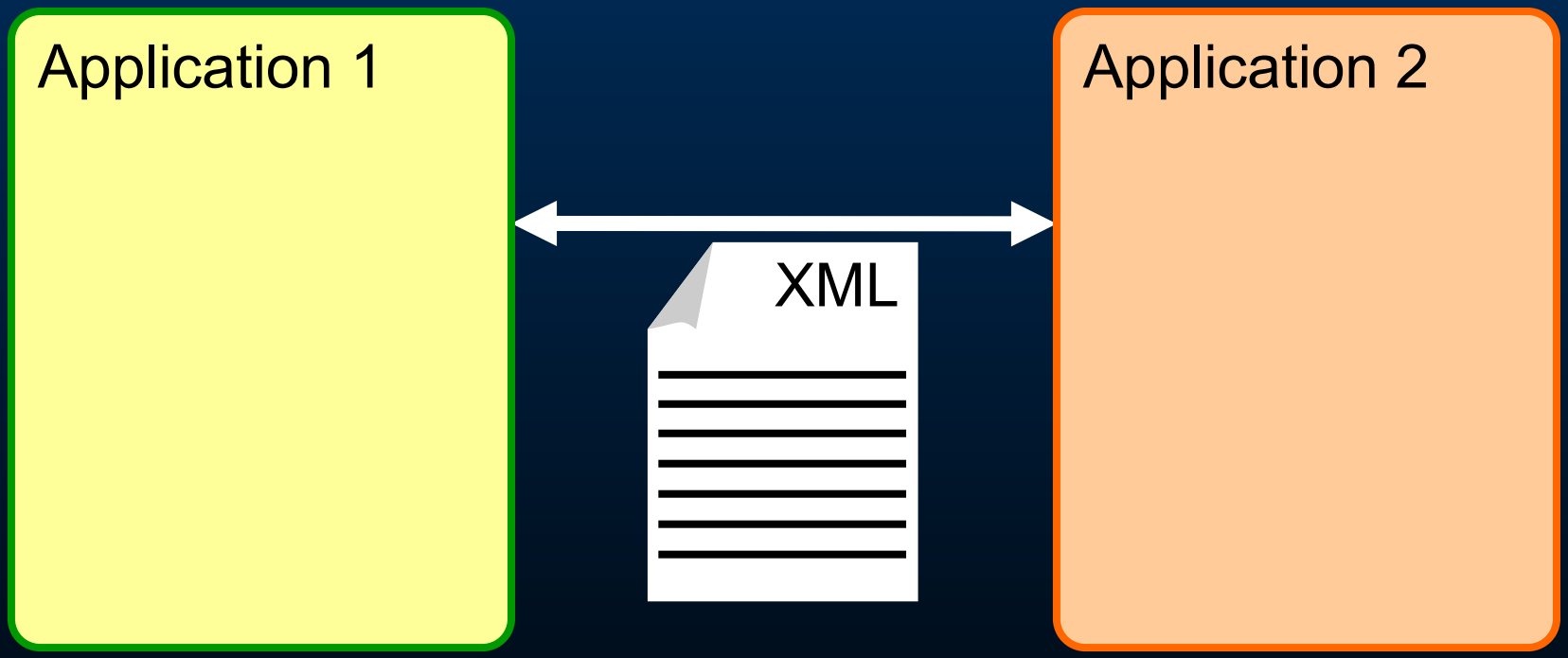
- ***"A Web service is a software application identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artefacts. A Web service supports direct interactions with other software agents using XML based messages exchanged via internet-based protocols."***

What are Web Services?

- Identified by a **URI**
- Interfaces defined using **XML**
- Can be **discovered** by other systems
- Interact using **XML** based messages conveyed by **Internet protocols**

Source: W3C Web Services Glossary

What are Web Services?



Transport

- HTTP POST is most common
- But other protocols such as
  - FTP
  - SMTP
  - HTTP GET
- And other exotic ones:
  - Jabber
  - BEEP

Packaging – Soap

- Used to mean
  - Simple
  - Object
  - Access
  - Protocol
- From SOAP 1.2 > SOAP is no longer an acronym
- Two Types of SOAP

Packaging – Soap

- SOAP RPC:
  - encode and bind data structures into xml.
  - encode an RPC call

Serialization

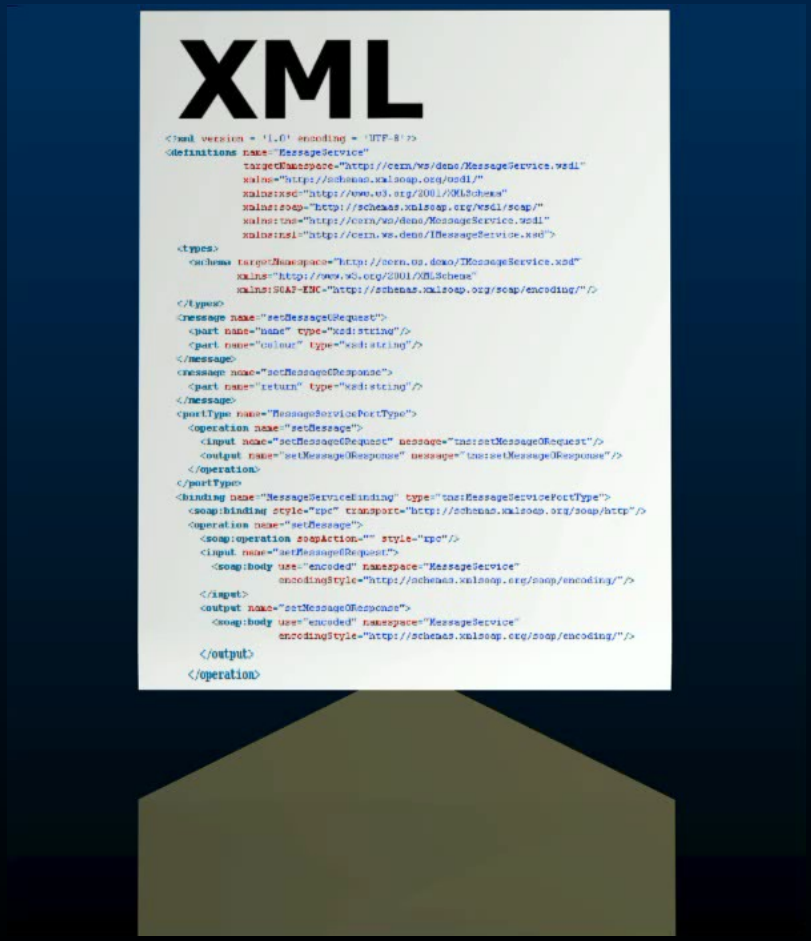
```
class PurchaseOrder {
String item = "socks";
int amount = 1;
}
```

Serializer

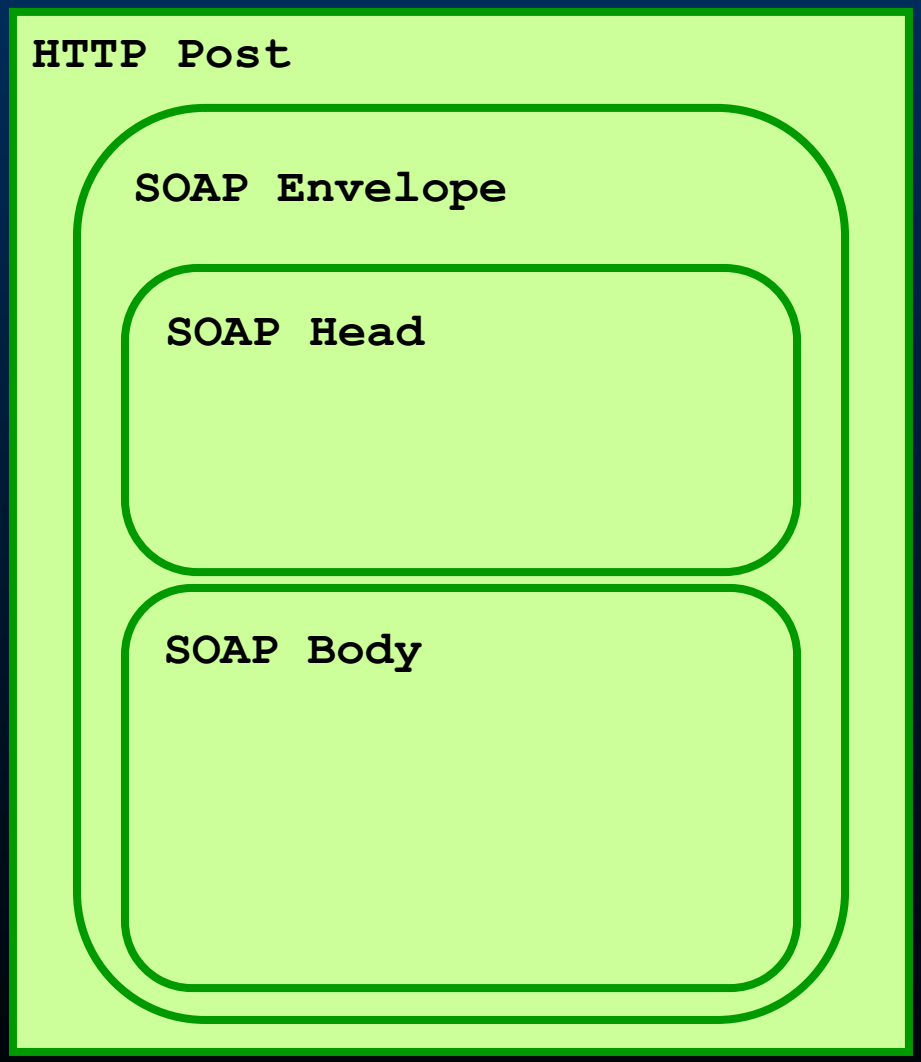
```
<PurchaseOrder>
<item type="xsd:string">
    socks
</item>
<amount type="xsd:int">
    1
</amount>
</PurchaseOrder>
```

Packaging - SOAP

- SOAP ‘document style’
  - packages xml in an envelope



Packaging – Soap



Packaging – Soap

```
<s:Envelope xmlns:s="URN">
  <s:header>
    <s:transaction xmlns:m="soap-
  transaction">
  <m:transactionID>
       1234
  </m:transactionID >
    </s:transaction>
  </s:header>
```

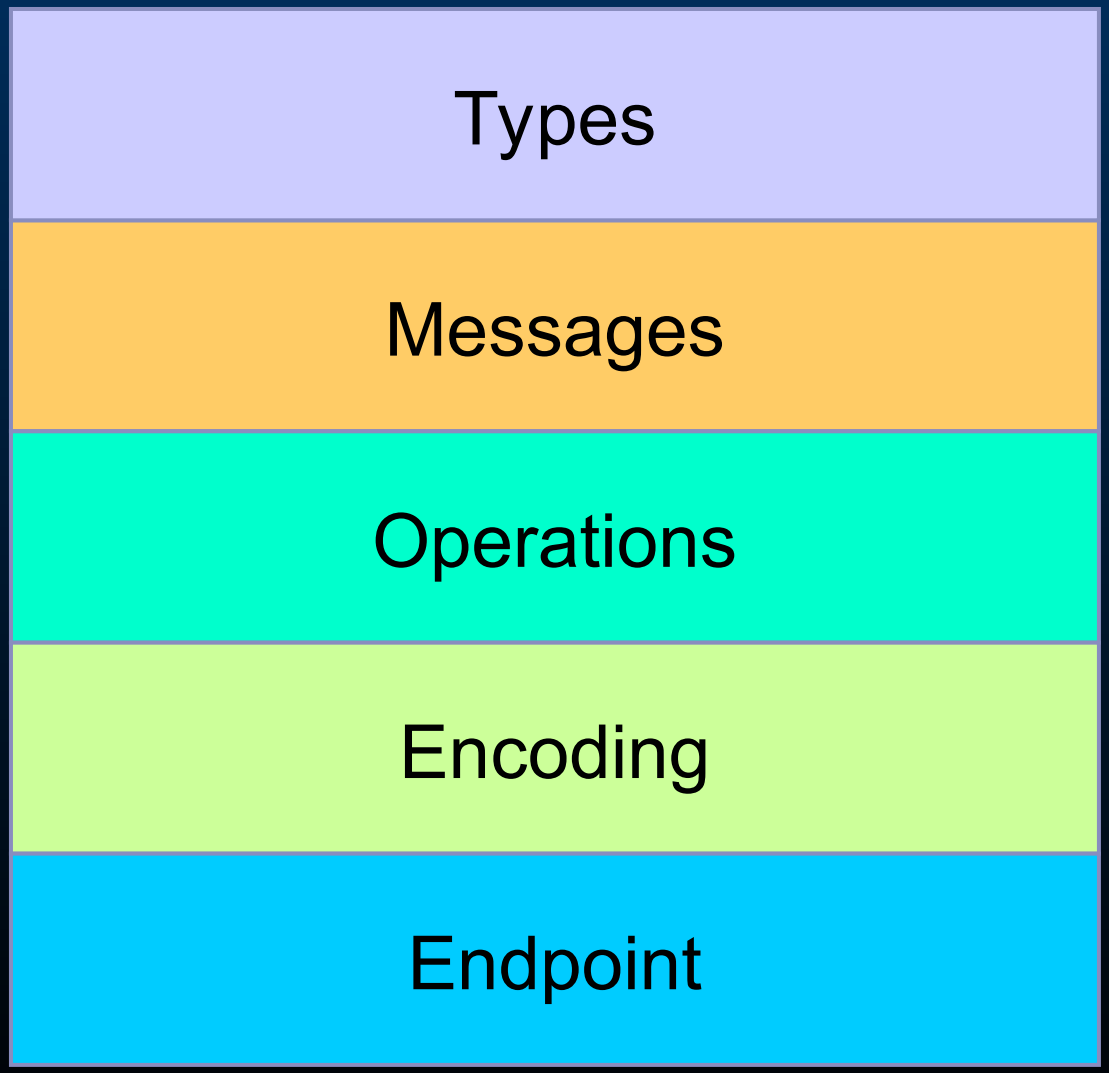
Packaging – Soap

```
  <s:Body>
    <n:purchaseOrder xmlns:n="URN">
     <n:item>socks</n:item>
     <n:amount>1</n:amount>
    </n:purchaseOrder>
  </s:Body>
</s:Envelope>
```

Description – WSDL

- Web Services Description Language
- “Web Services Description Language (WSDL) provides a model and an XML format for describing Web services.” w3c.org

Description – WSDL



Types

```
<types>
<schema targetNamespace=" IMessageService.xsd"
xmlns="…/XMLSchema"
    xmlns:SOAPENC="…/soap/encoding/"/>
</types>
```

Messages

```
<message name=“purchase">
    <part name=“item" type="xsd:string"/>
    <part name=“quantity" type="xsd:integer"/>
</message>
```

Operations

```
<operation name="setMessage">
    <input name="setMessageRequest“
message="tns:setMessageRequest"/>
    <output name="setMessageResponse“
message="tns:setMessageResponse"/>
</operation>
```

Encoding

```
<soap:operation soapAction="" style="rpc"/>
    <input name="setMessage0Request">
        <soap:body use="encoded"
namespace="MessageService"
encodingStyle="…/soap/encoding/"/>
    </input>
```

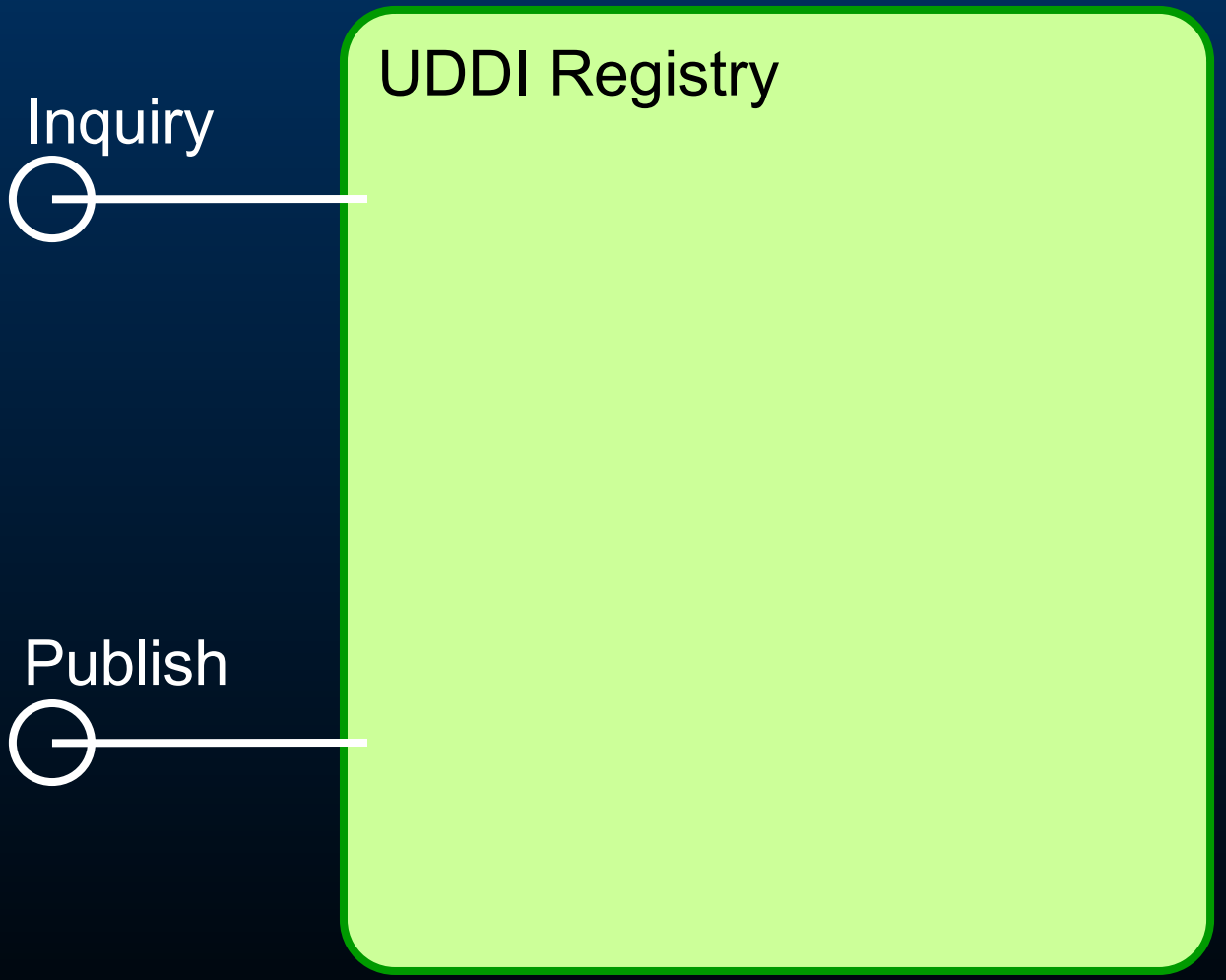
Endpoint

```
<service name="MessageService">
<port name="MessageServicePort"
binding="tns:MessageServiceBinding">
<soap:address location="http://localhost:8080/setMessage/"/>
</port>
</service>
```

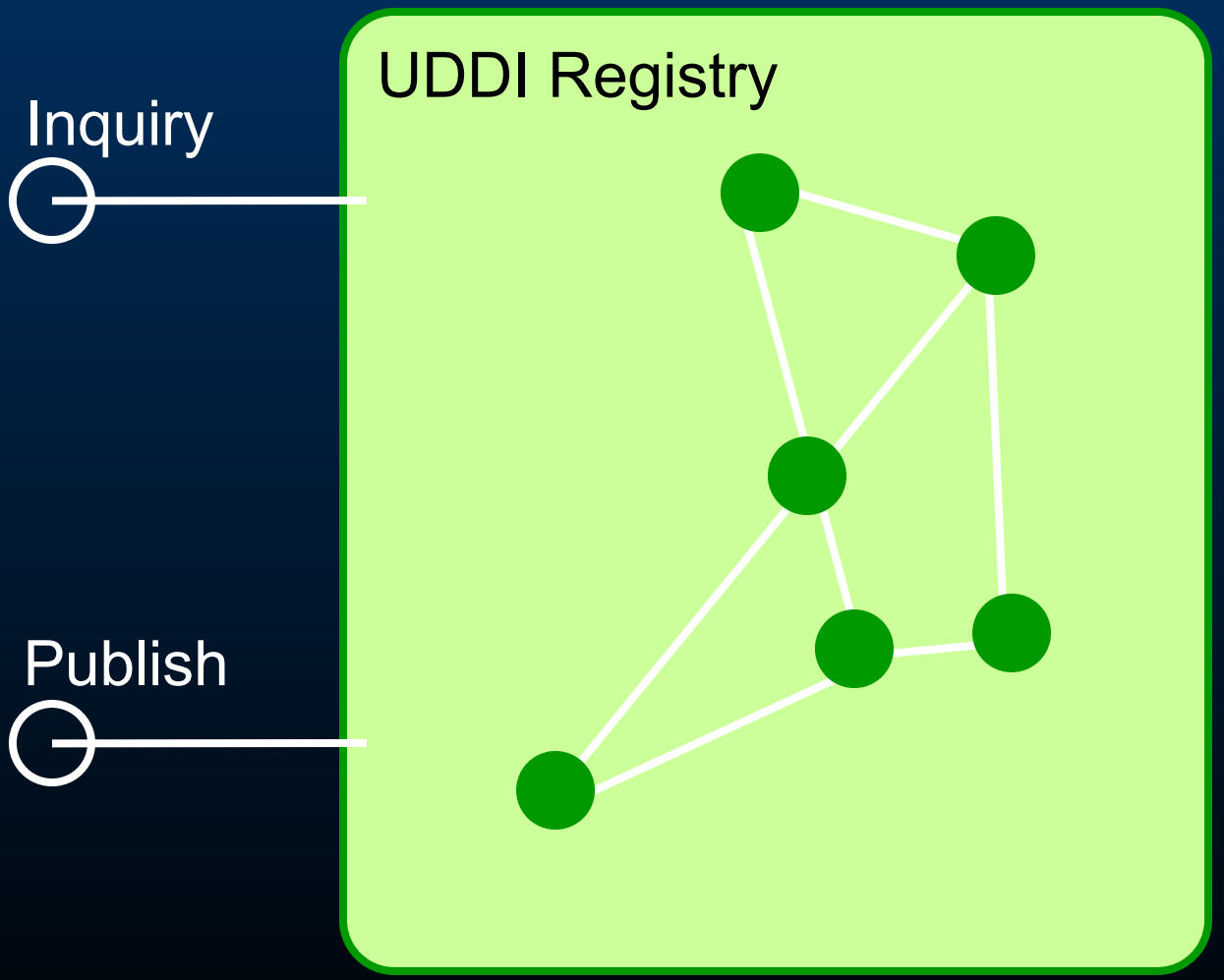
Discovery – UDDI

- Universal Description, Discovery and Integration
- A UDDI Server acts as a registry for Web Services and makes them searchable.

Discovery – UDDI



Discovery – UDDI



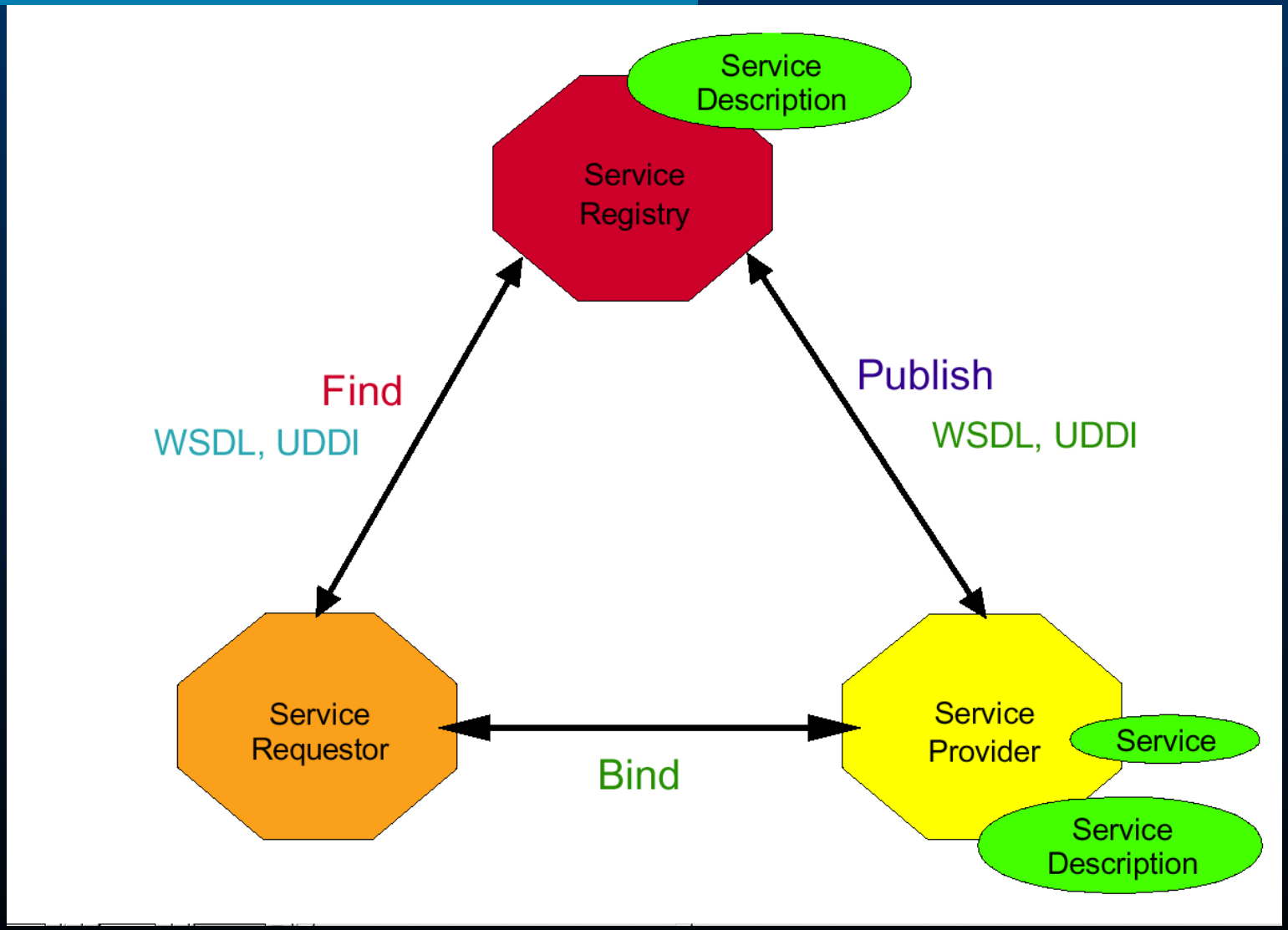
SOAP-based communication

- SOAP:
  - Data in a well-defined XML format
  - Transport over various protocols
    - HTTP, SMTP are the most used, perhaps because they are firewall-friendly
  - server side: either an RPC call or a message delivered

Web services

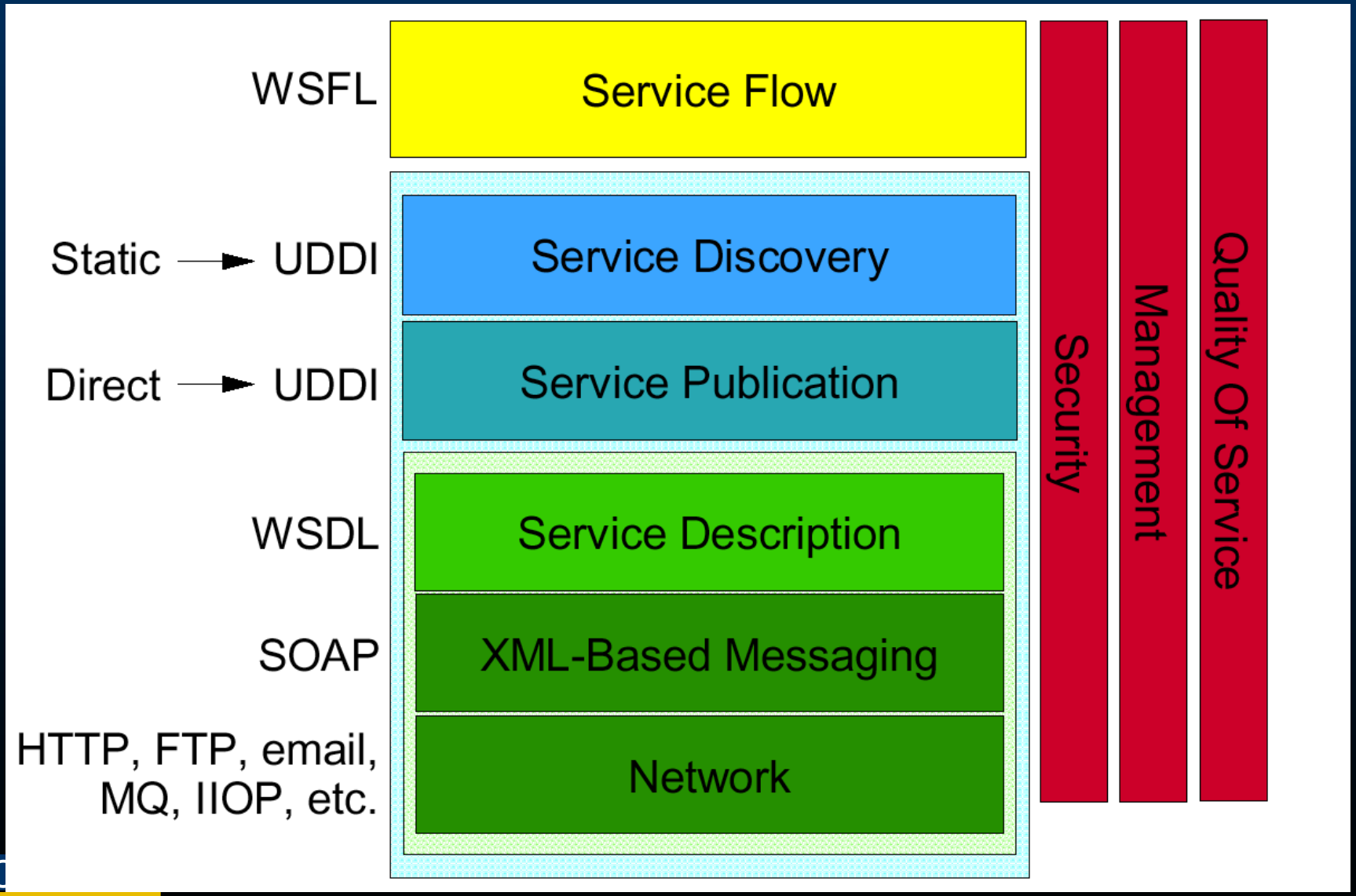
- A collection of XML-based technologies developed by the e-business community to address issues of:
  - service discovery
  - interoperable data exchange and/or application invocation
  - service compositions (workflow, business processes)
- Major developers include:
  - Apache, IBM, HP, SUN & Microsoft (.NET)
    - http://www.webservices.org/

Web Services Architecture

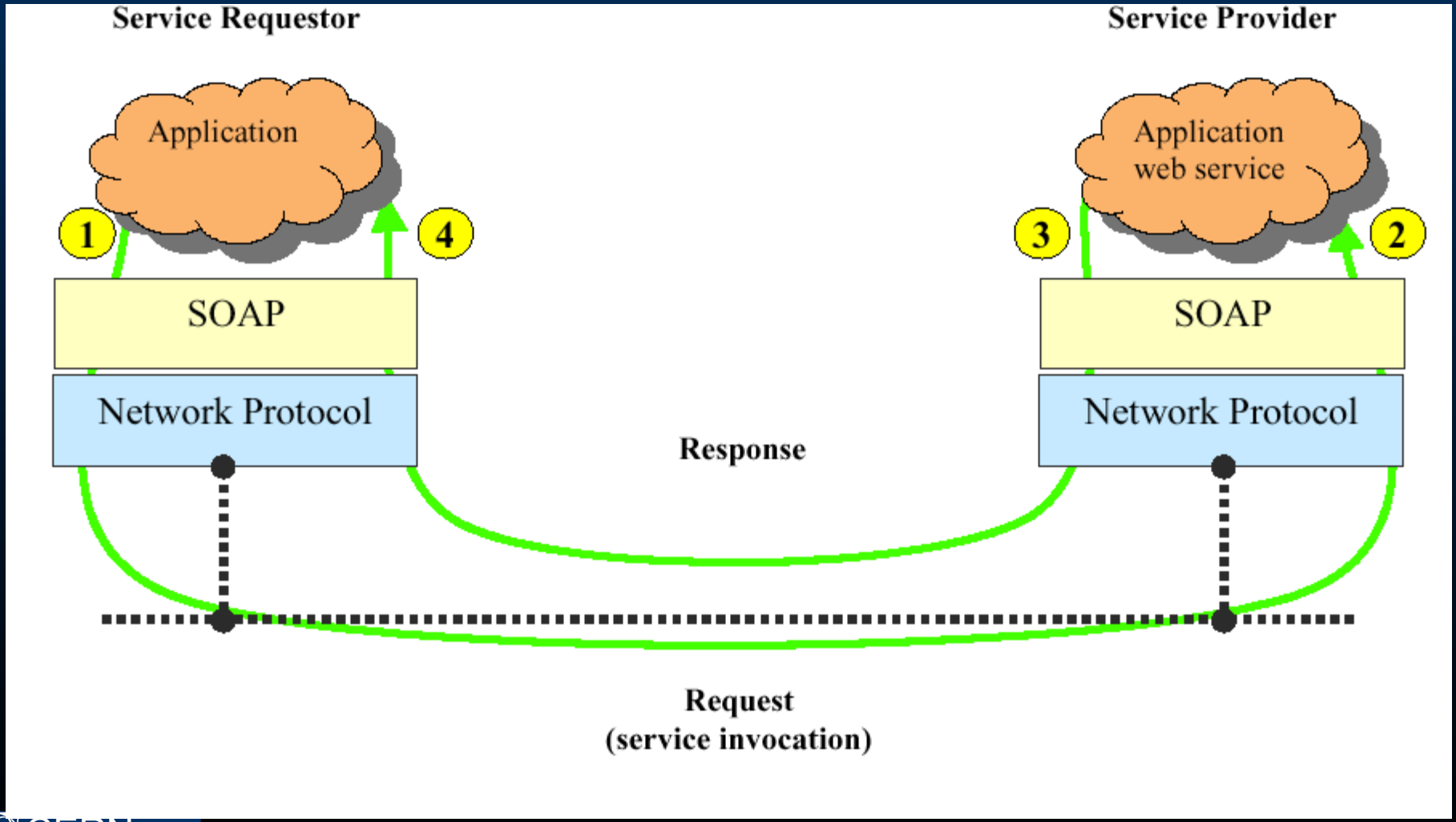


Let a program "click on a web page"

Web Services Stack



XML Messaging Using SOAP



WSDL

- Web Services Definition Language
  - http://www.w3.org/TR/wsdl/
- An XML-based language for describing Web Services
  - what the service does (description)
  - how to use it (method signatures)
  - where to find the service
- It *does not* depend on the underlying protocol
- But: It is not much human-readable

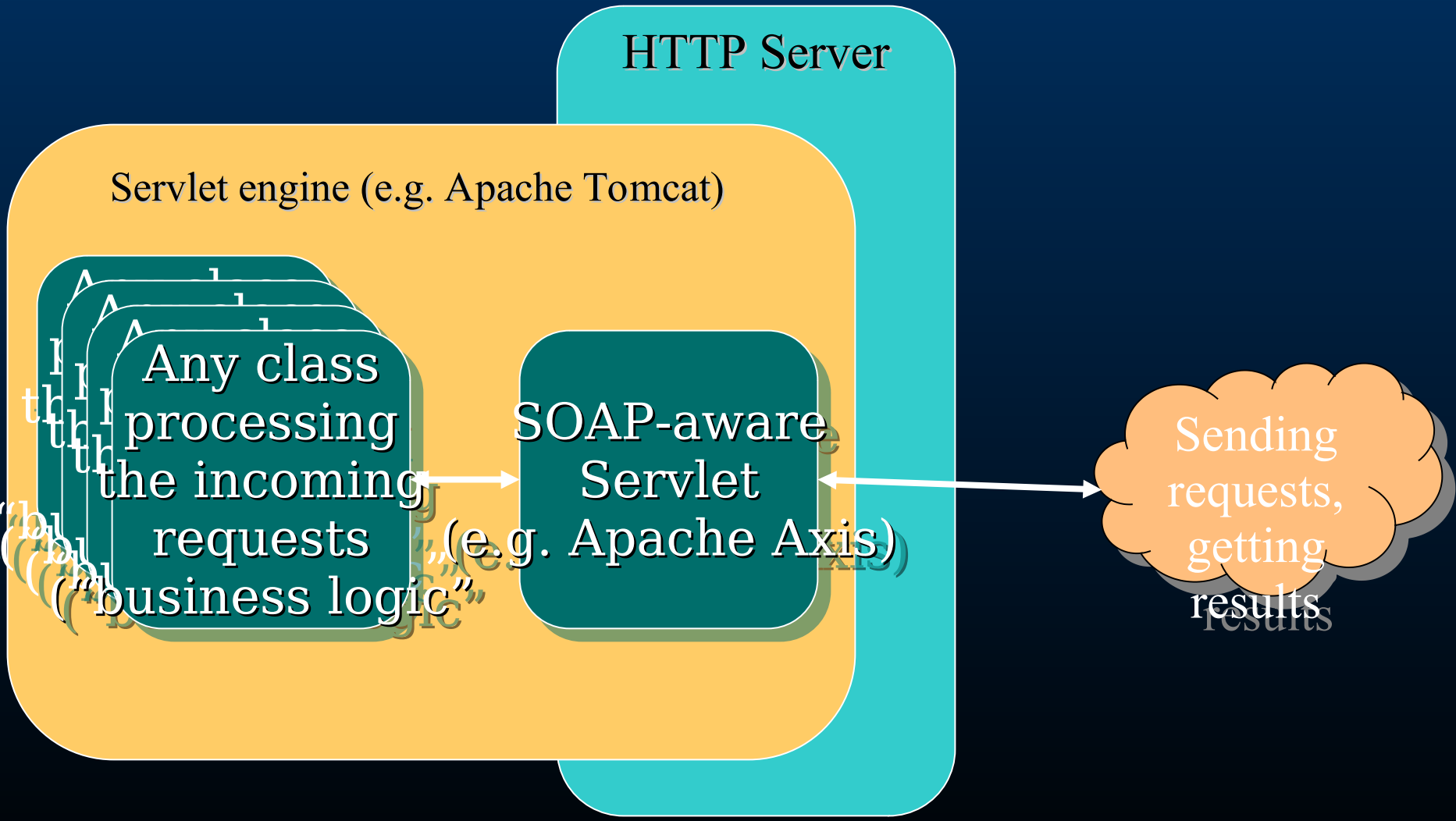
Hello.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
   targetNamespace="http://localhost:8080/axis/services/Hello"
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:impl="http://localhost:8080/axis/services/Hello-impl"
   xmlns:intf="http://localhost:8080/axis/services/Hello"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <wsdl:message
   name="setHelloMessageRequest">    <wsdl:part name="in0"
   type="xsd:string"/>  </wsdl:message>  <wsdl:message
   name="getHelloMessageResponse">    <wsdl:part name="return"
   type="xsd:string"/>  </wsdl:message>  <wsdl:message
   name="setHelloMessageResponse">  </wsdl:message>  <wsdl:message
   name="getHelloMessageRequest">  </wsdl:message>  <wsdl:portType
   name="HelloWorldService">    <wsdl:operation name="getHelloMessage">
    <wsdl:input message="intf:getHelloMessageRequest"/>      <wsdl:output
   message="intf:getHelloMessageResponse"/>    </wsdl:operation>
   <wsdl:operation name="setHelloMessage" parameterOrder="in0">
   <wsdl:input message="intf:setHelloMessageRequest"/>      <wsdl:output
   message="intf:setHelloMessageResponse"/>    </wsdl:operation>
   </wsdl:portType>  <wsdl:binding name="HelloSoapBinding"
   type="intf:HelloWorldService">    <wsdlsoap:binding style="rpc"
   transport="http://schemas.xmlsoap.org/soap/http"/>    <wsdl:operation
   name="getHelloMessage">      <wsdlsoap:operation soapAction=""/>
   <wsdl:input>        <wsdlsoap:body
   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   namespace="getHelloMessage" use="encoded"/>      </wsdl:input>
   <wsdl:output>        <wsdlsoap:body
   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   namespace="http://localhost:8080/axis/services/Hello" use="encoded"/>
     </wsdl:output>    </wsdl:operation>    <wsdl:operation
   name="setHelloMessage">      <wsdlsoap:operation soapAction=""/>
   <wsdl:input>        <wsdlsoap:body
   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   namespace="setHelloMessage" use="encoded"/>      </wsdl:input>
   <wsdl:output>        <wsdlsoap:body
   encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
   namespace="http://localhost:8080/axis/services/Hello" use="encoded"/>
    </wsdl:output>    </wsdl:operation>  </wsdl:binding>  <wsdl:service
```

UDDI (and alternatives)

- Universal Description, Discovery and Integration
  - http://www.uddi.org
- UDDI creates a platform-independent, open framework & registry for:
  - Describing services
  - Discovering businesses
  - Integrating business services
- The UDDI may be less used than predicted, especially on the Internet level

A Web Service example in Java



Design Recommendations

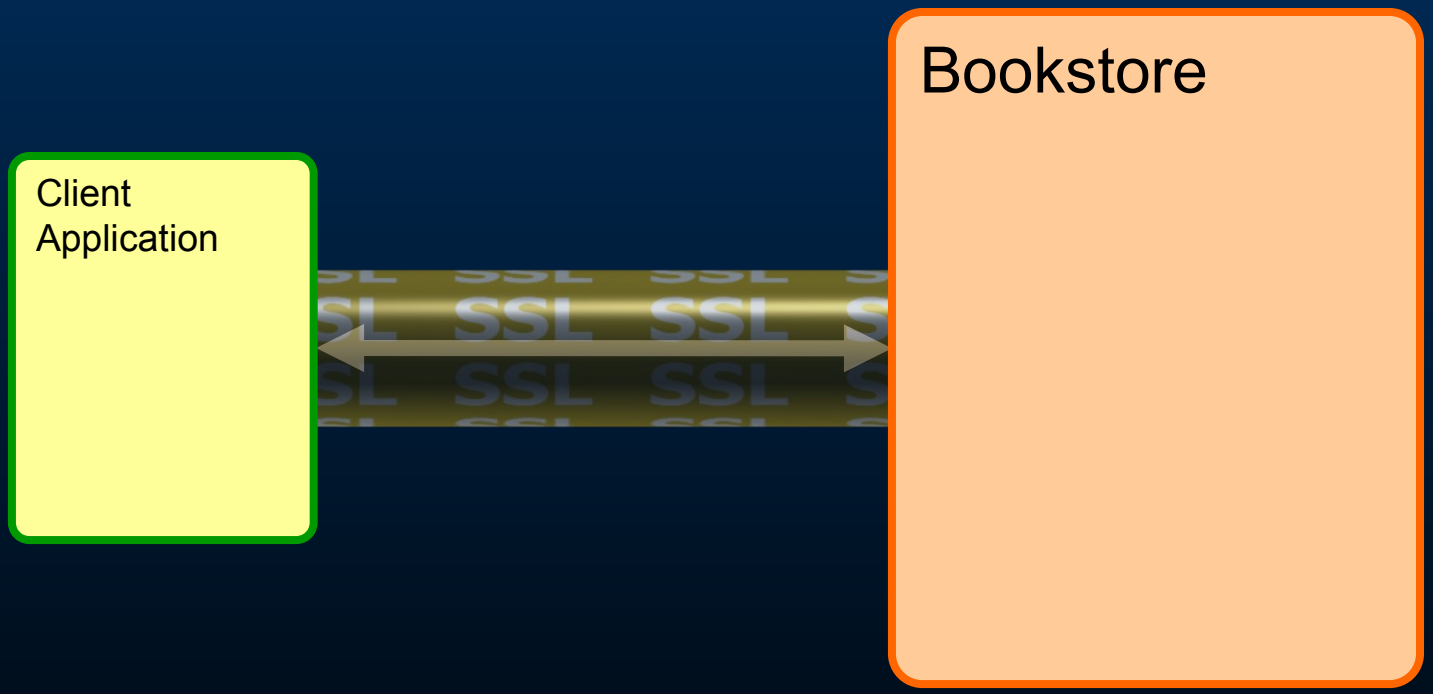
- Create a local class
- Create a method with the same name

Web Services Future

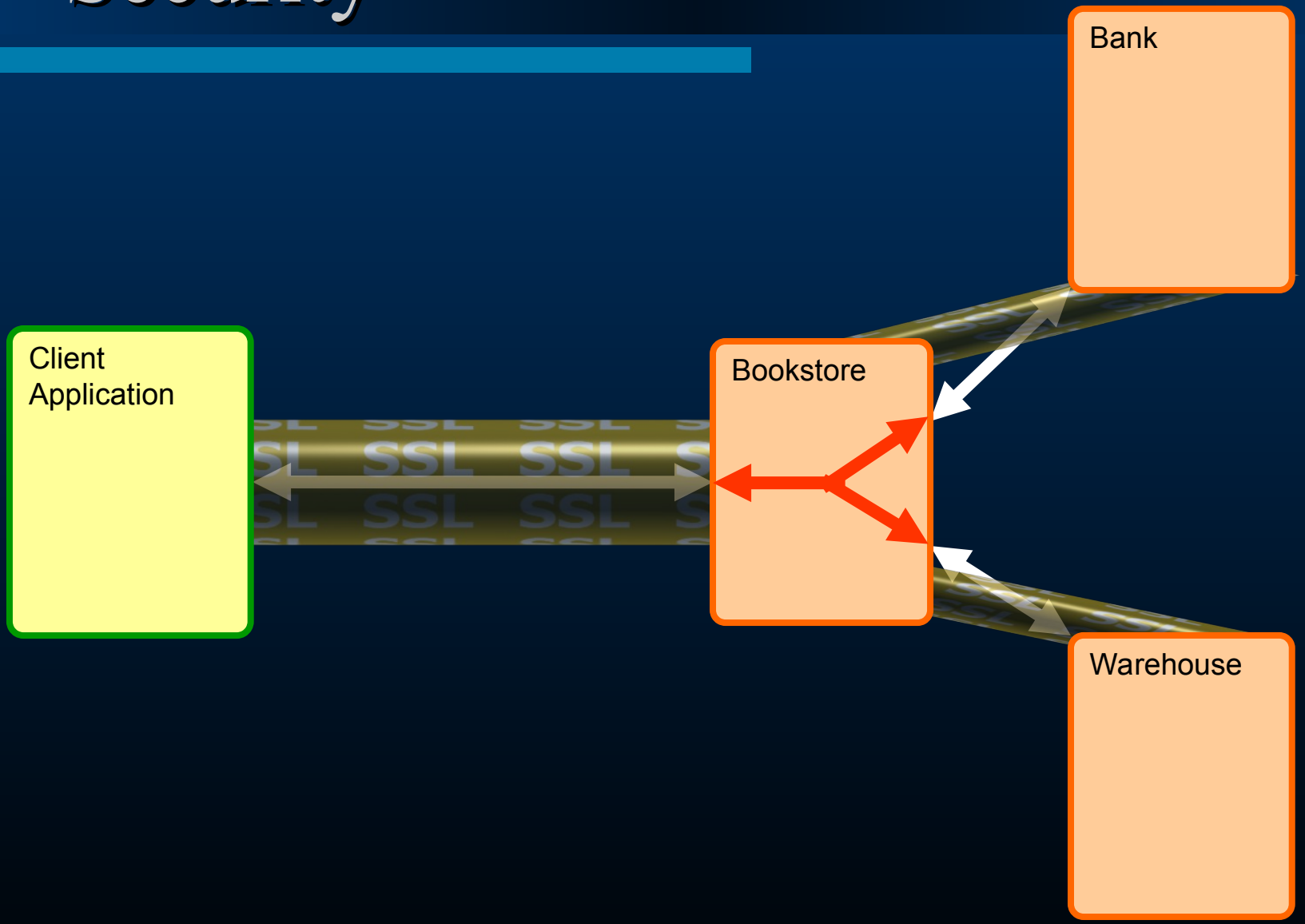
Security

- WS do not define *how* to do security, they rely on other mechanisms layered on top.
- Very common to use SSL
  - Good for simple cases
    - Weak when multi-tier
    - Forces encryption of all data – sometimes not needed

Security



Security



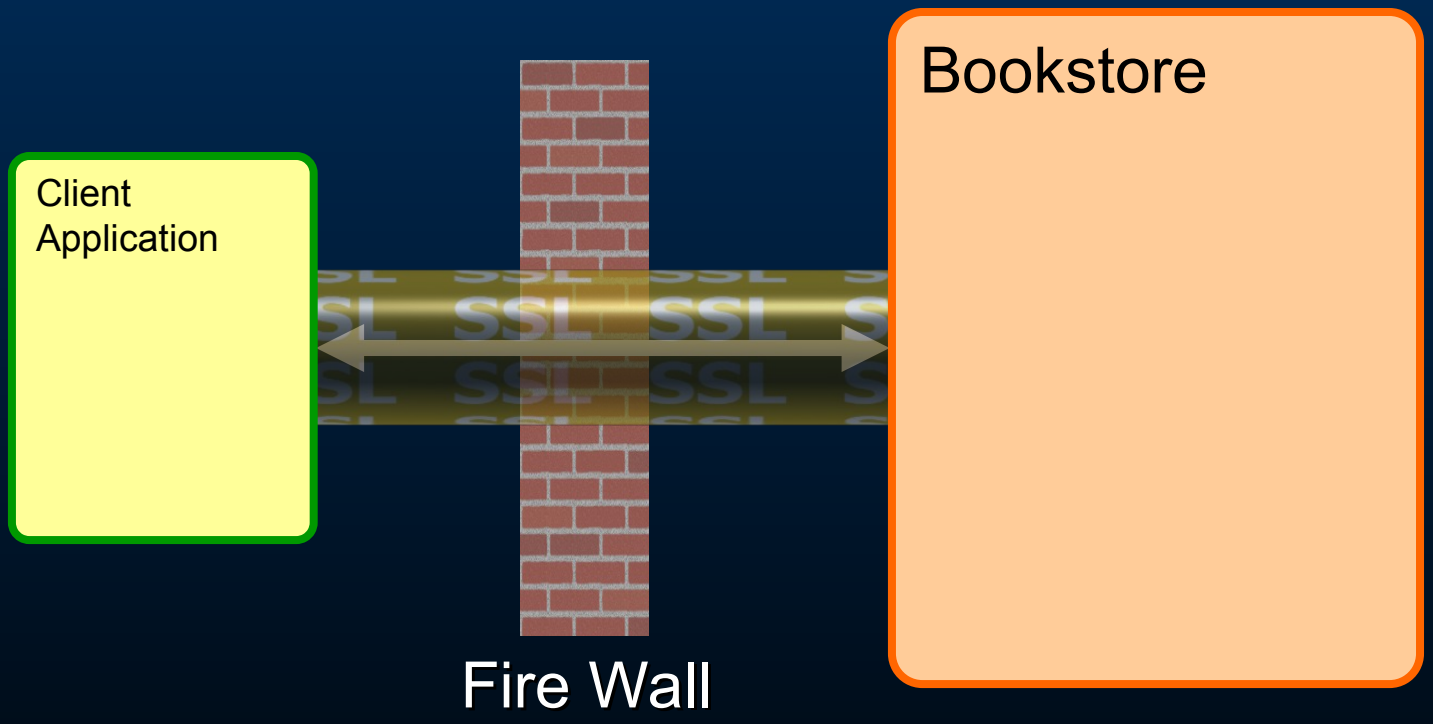
WS Security Standardisation

- **W3C - http://www.w3c.org**
  - XML Encryption
  - XML Digital Signatures

- **WS-I - http://www.ws-i.org**
  - WS Security Profile

- **OASIS - http://www.oasis-open.org**
  - WS-Security
  - SAML - Security Assertion Markup Language
  - XACML - Extensible Access Control Markup Language
  - XKMS - XML Key Management Specification

Security – Fire Walls



Security

- Firewall ‘transparency’
- Double-edged sword
  - Simplifies deployment
  - Opens up potential holes in enterprise security
    - Similar to CGI, etc.

WS-I

- **Web Services Interoperability Organization**

  `http://www.ws-i.org`

R1017 A RECEIVER MUST NOT mandate the use of the `xsi:type` attribute in messages except as required in order to indicate a derived type

WS-I Basic Profile Version 1.0

Other Statndards

- Security
  - Single Sign-on, credentials
- Transactions
- Quality of service
  - Timeliness guarantees
- Asynchronous operations
  - Co-ordination, workflow

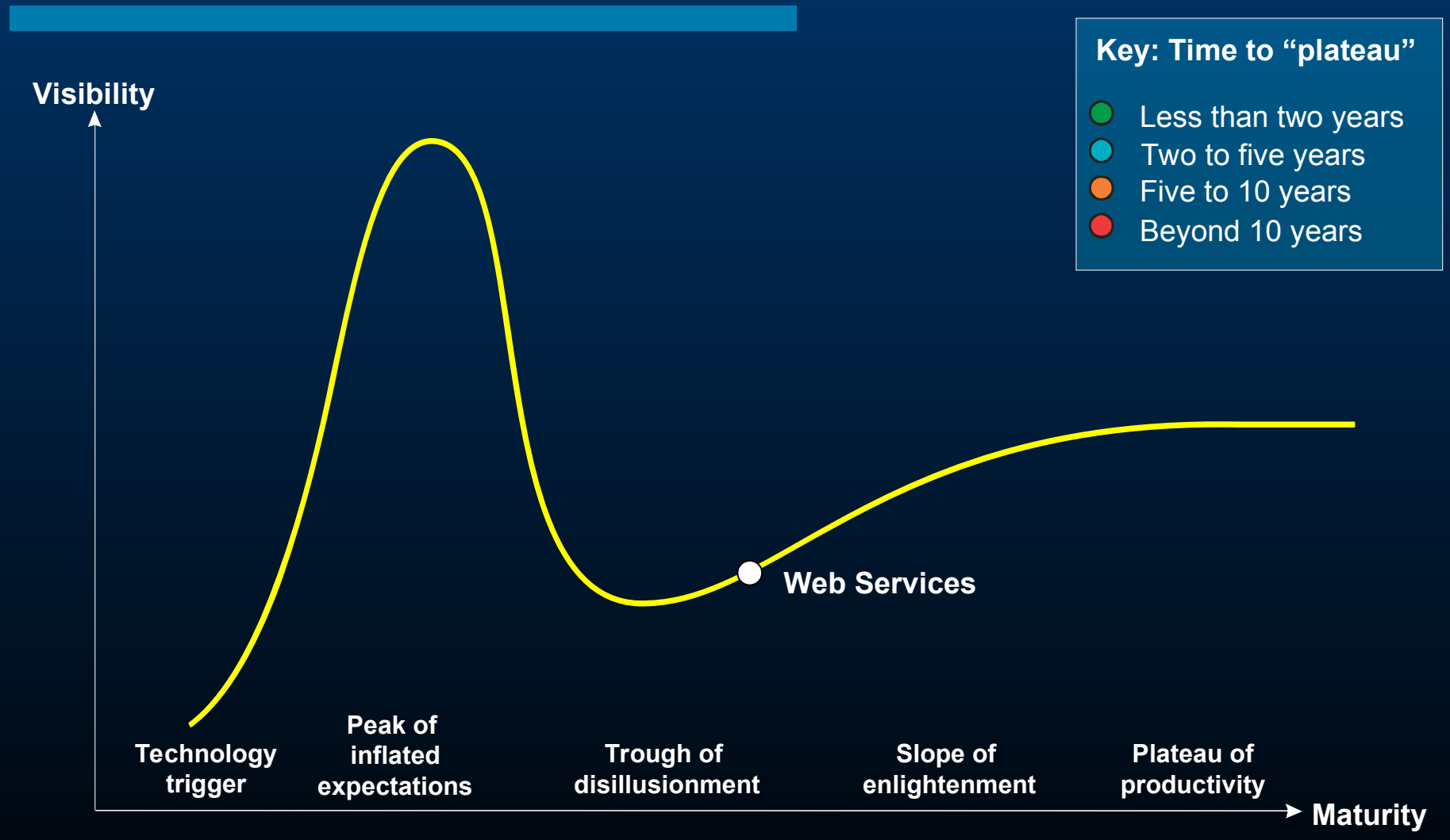
Real Examples

- Amazon Web Services API
- Google Web API
- HP & IBM online stores

Next Steps

- Overtime Interface
- Other documents (Materials Request, TID, Transport Request)
- E-Business – with ebXML

Gartner's 'Hype' Curve



CERN IDS

Source: Gartner Group June 2002

Thank you