XML als Kommunikationssprache auch im Umweltbereich – erste Erfahrungen mit SOAP-Anwendungen

Wassili Kazakos (FZI Karlsruhe), Ulrike Freitag (Condat AG Berlin)

Workshop Umweltdatenbanken 2001, Jena

7./8. Juni 2001

Stand des XML-Einsatzes in der verteilten Programmierung

- Gartner Group schätzt, daß in 2001 70 % der B2B-Transaktionen XML nutzen werden
- Back ends sprechenXML:
  - Oracle XML SQL Utility (XSU).
  - Microsoft XML - ADO können XML-basierte Recordsets von jeder ODBC/OLEDB-Quelle erzeugen und verändern
  - XML Adapter for ERP-Systeme,
  - EDI. XML/EDI movement. -> ebXML
- Microsoft.Net unterstützt ein XML-Datenaustausch-Framework
- Orb-Hersteller IONA unterstützt SOAP-Plug in für den Orb “Orbix 2000”, Servlet Support in iPAS, XML-enabled Integrator, etc.

XML als Kommunikationssprache

- XML als Serialisierungssprache für Funktions- bzw. Methodenaufrufe (Protokoll)
- XML als allgemeines Datenaustauschformat zwischen verschiedenen Systemen
    - Business (EDI,…)
    - Datenbankschnittstellen

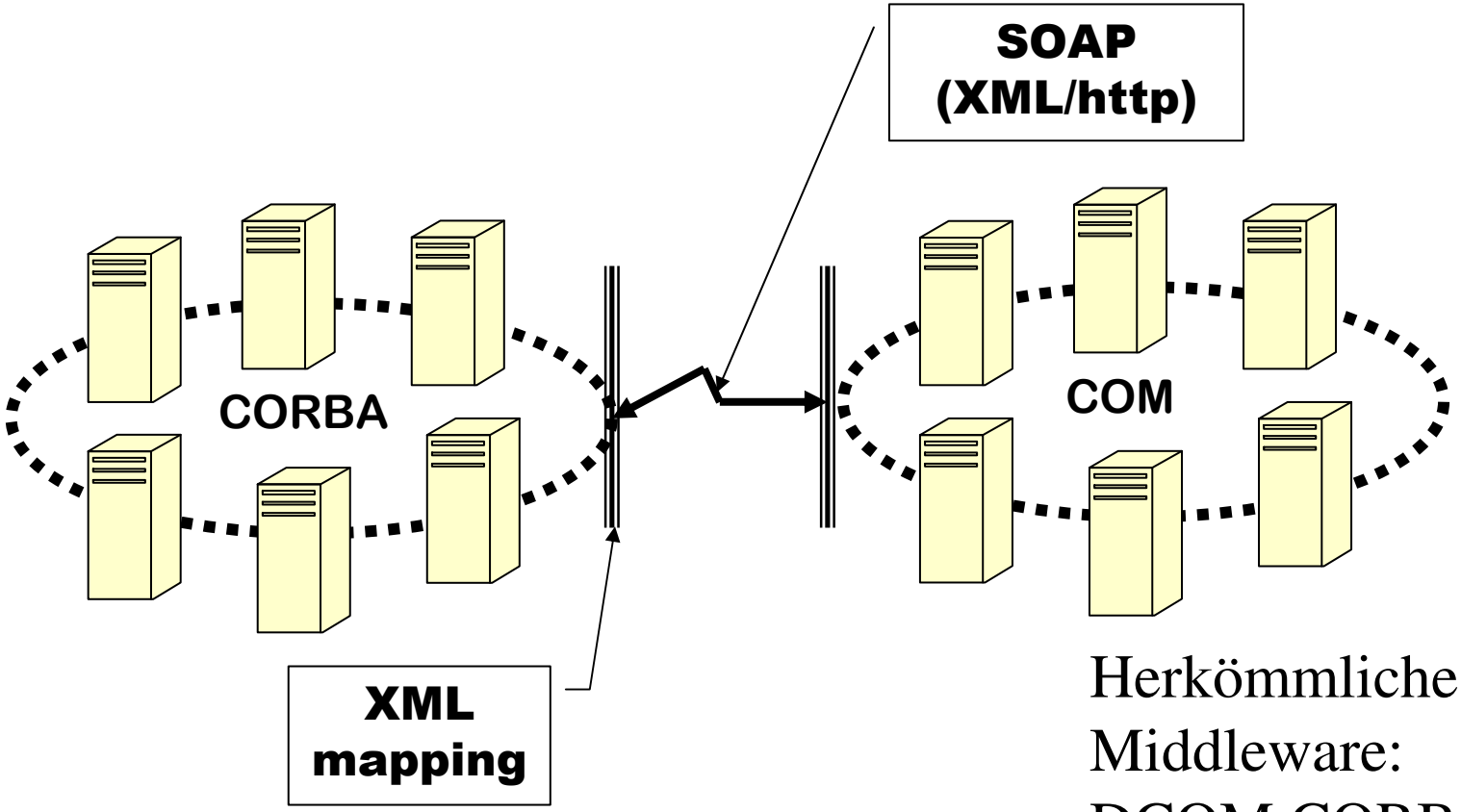
Vorteile von XML in der Kommunikation

- **einfach, in jeder Sprache implementierbar**
- **leicht erweiterbar**
- **portabel**
- **plattformneutral**
- **herstellerunabhängig**
- **“Web-freundlich”**

Konzept der Web Services

- Browser greifen auf HTML-Dateien zu
- Anwendungen können noch nicht via Web kommunizieren
  - Herkömmliche Middleware-Applikationen scheitern häufig am Firewall
- Vorteil von XML über http:
  - Firewalls erlauben http-traffic
  - XML-Daten können Applikationen mit Hilfe von WebServices abbilden
  - ***Web Services*** können via the Web veröffentlicht und zugreifbar gemacht werden
  - XML Nachrichten über http kommunizieren mit Web Services
  - Web Services dienen als Brücke zu beliebige Dokumenten oder Programmen

Moderne Middleware Standards als A2A-Verbindung im E-Commerce



Herkömmliche Middleware: DCOM,CORBA,RMI

Was ist ein Protokoll?

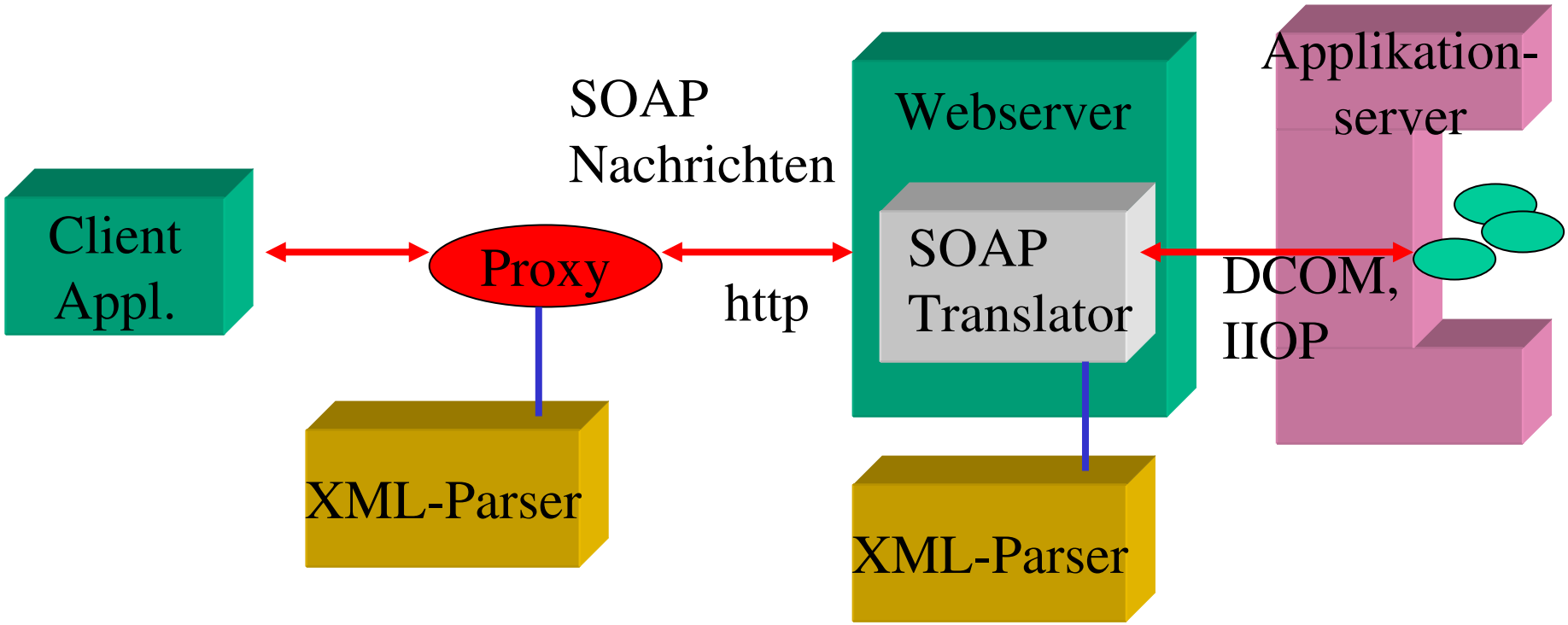
- **Die Spezifikation eines Protokolls besteht aus den folgenden Teilen:**
  - **Dienst, der über das Protokoll zur Verfügung gestellt wird (=Semantik)**
  - **Annahmen über die Umgebung, in der das Protokoll abläuft**
  - **Das Vokabular der Nachrichten (=Wortschatz)**
  - **Das Format der Nachrichten (=Syntax)**
    - **"Marshaling, Serialisierung"**

    XML
  - **Regelwerk, das den Nachrichtenaustausch beschreibt**

Simple Object Access Protocol (SOAP)

- ist ein **leichtgewichtiges, flexibles XML-Protokoll** für den Austausch und den Aufruf verteilter Objekte wie CORBA, DCOM usw.
  - RPC-ähnlich
- Die Spezifikation ist ein Internet Draft (IETF)
- SOAP nutzt das **HTTP-Protokoll zum Transport** und **XML als Serialisierung**
- SOAP ist **komplett erweiterbar** (transaktionen, security, asynch messaging, etc.)
- SOAP kann auch auf andere Transportprotokolle aufsetzen (IIOP, BXXP)

SOAP Szenario



SOAP Beispiel

- **Eine Schnittstelle in CORBA-IDL (nur als Beispiel, SOAP ist unabhängig vom Objektmodell):**

```
interface Konto {
....
float kontoStand(in string name, in string kontonummer);
....
}
```

- **Der Aufruf:**

```
float wert = Konto.kontoStand("Hugo Mustermann","12459-123");
```

SOAP-Request

```
POST /Konto HTTP/1.1
Host: www.muster.deSOAP
Content-Type: text/xml
Content-Length: 252
SOAPMethodName: urn:Konto#kontoStand
```

HTTP-Header

```
<SOAP:Envelope xmlns:SOAP="urn:schemas-xmlsoap-
    org:soap.v1">
    <SOAP:Body>
        <m:kontoStand xmlns:m="urn:Konto">
            <name>Hugo Mustermann</name>
            <kontonummer>12459-123</kontonummer>
        </m:kontoStand>
    </SOAP:Body>
</SOAP:Envelope>
float wert = Konto.kontoStand("Hugo Mustermann","12459-
    123");
```

SOAP Responce

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/xml
Content-Length: 219

<SOAP:Envelope xmlns:SOAP="urn:schemas-xmlsoap-
    org:soap.v1">
    <SOAP:Body>
        <m:kontoStandResponse xmlns:m="urn:Konto">
            <return>3890,25</return>
        </m:kontoStandResponse>
    </SOAP:Body>
</SOAP:Envelope>

float wert = Konto.kontoStand("Hugo Mustermann","12459-123");
```

HTTP-Header

SOAP Responce Error

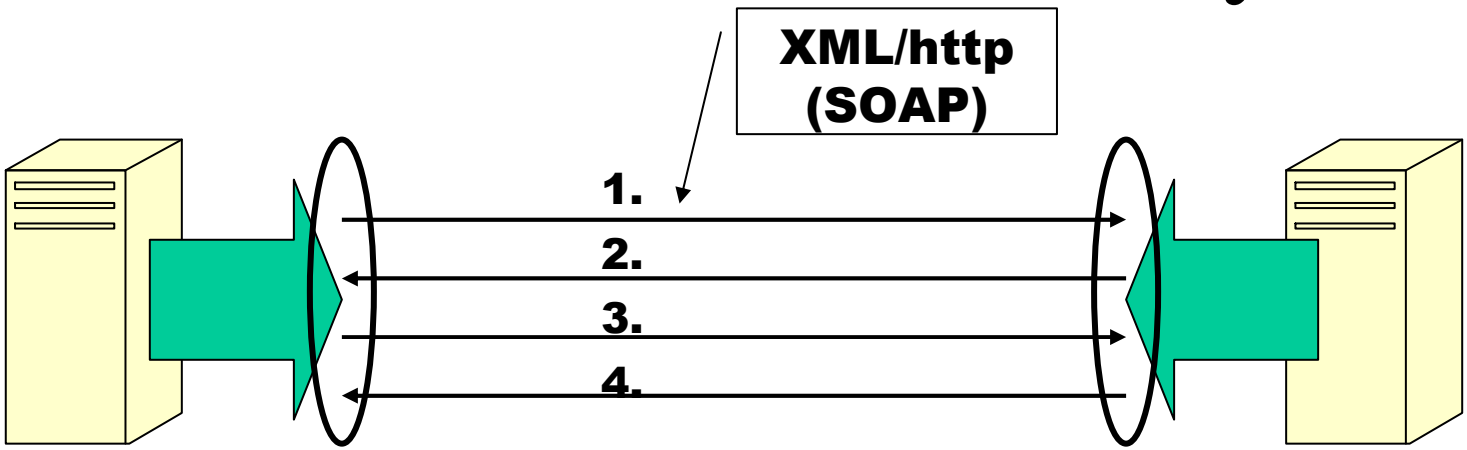
HTTP-Header

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/xml
Content-Length: 368
<SOAP:Envelope xmlns:SOAP="urn:schemas-xmlsoap-
    org:soap.v1">
    <SOAP:Body>
        <SOAP:Fault>
        <faultcode>400</faultcode><faultstring>Error</faultstring>
        <runcode>1</runcode>                    maybe | no | yes
        <detail xmlns:e="...." ...>
                <message>Apperror</message>
    Applikationsspezifische
                <errorcode>2</errorcode>        Fehlerbeschreibung
        </detail>
        </SOAP:Fault>
    </SOAP:Body>
```

Web Service Description Language (WSDL)

- Es gibt ein spezielles XML-Schema für Web Services
- Übernimmt die Rolle der Interface Definition Language (IDL) von CORBA aber in XML / http
- Definiert:
  - Message types
  - Operations on messages (i.e. services)
  - Mapping to protocols (http, MIME)
- Verweise basieren auf URL's

Web Service discovery



1. Charley uses the URL provided by Harry to get the Web Service description
2. Charley downloads the Web Service description, getting the format of the XML messages Harry expects
3. Charley generates an XML message to match and sends it to Harry via SOAP
4. Harry returns the defined XML response message, also via SOAP

WSDL example

```
<?xml version='1.0' encoding='UTF-16' ?>

 <!-- Generated 01/24/01 by Microsoft SOAP SDK WSDL File
Generator, Version 1.0 -->

<definitions
   name ='TypeTest00'
   targetNamespace='http://www.myserver.com/soap/TypeTest00.wsdl`

   xmlns:tns='http://www.myserver.com/soap/TypeTest00.wsdl'

   xmlns:xsd1='http://www.myserver.com/soap/TypeTest00.xsd'

   xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'

   xmlns='http://schemas.xmlsoap.org/wsdl/'>
```

WSDL example (cont)

```
  <types>
     <schema
targetNamespace='http://www.myserver.com/soap/TypeTest00.xsd'
          xmlns='http://www.w3.org/1999/XMLSchema'>
     </schema>
  </types>
  <message name='boolean_in'>
    <part name='val' type='boolean'/>
  </message>
  <message name='boolean_inResponse'>
    <part name='Result' type='boolean'/>
  </message>
```

WSDL example (cont)

```
<portType name='TypeTest00PortType'>
   <operation name='boolean_in' parameterOrder='boolean_inInput1'>
     <input message='tns:boolean_in' />
     <output message='tns:boolean_inResponse' />
   </operation>
   <operation name='boolean_inout' parameterOrder='boolean_inoutInOut1'>
     <input message='tns:boolean_inout' />
     <output message='tns:boolean_inoutResponse' />
   </operation>
```

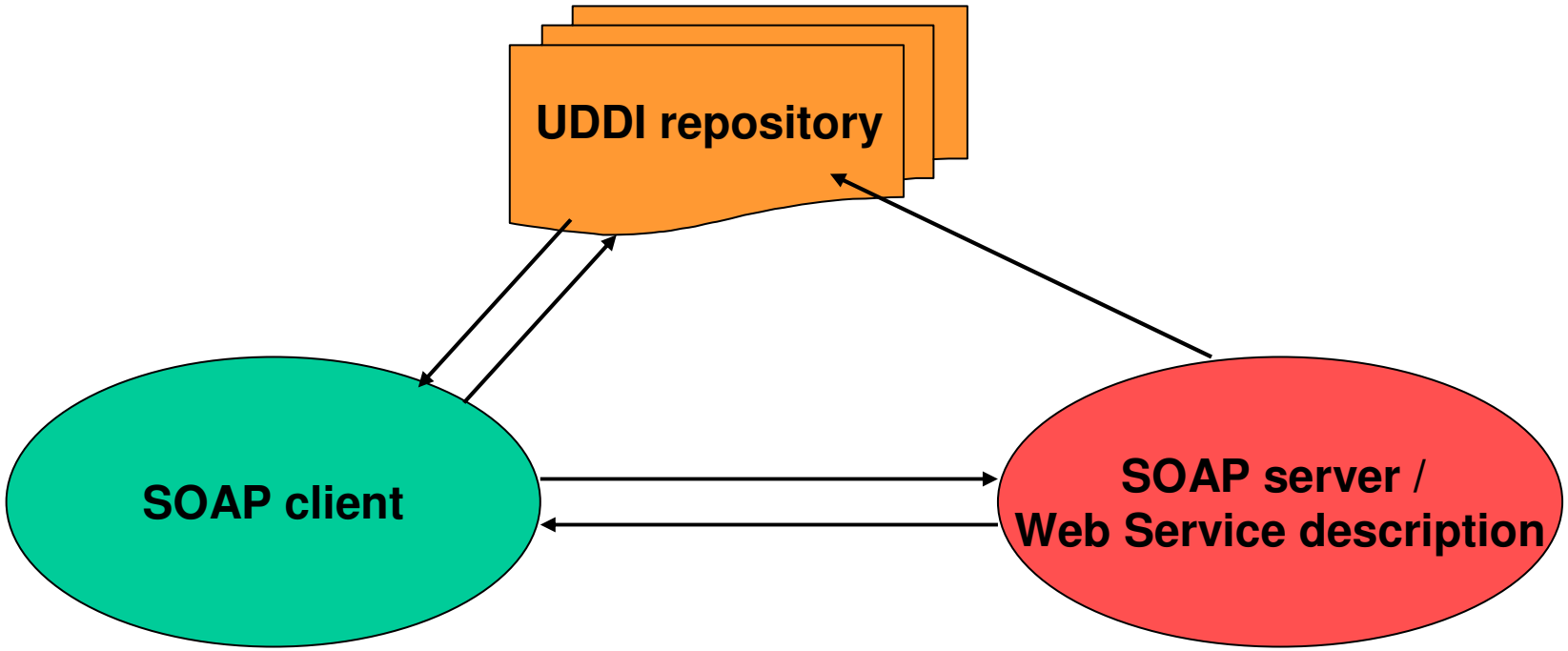
WSDL example (cont)

```
<binding name='TypeTest00Binding' type='tns:TypeTest00PortType' >
    <soap:binding style='document'
transport='http://schemas.xmlsoap.org/soap/http' />
    <operation name='boolean_in' >
      <soap:operation
soapAction='http://www.myserver.com/soap/myservice.asp' />
      <input>
        <soap:body use='encoded'
namespace='http://www.myserver.com/soap/TypeTest00.xsd'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
      </input>
      <output>
        <soap:body use='encoded'
namespace='http://www.myserver.com/soap/TypeTest00.xsd'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/' />
      </output>
    </operation>
```

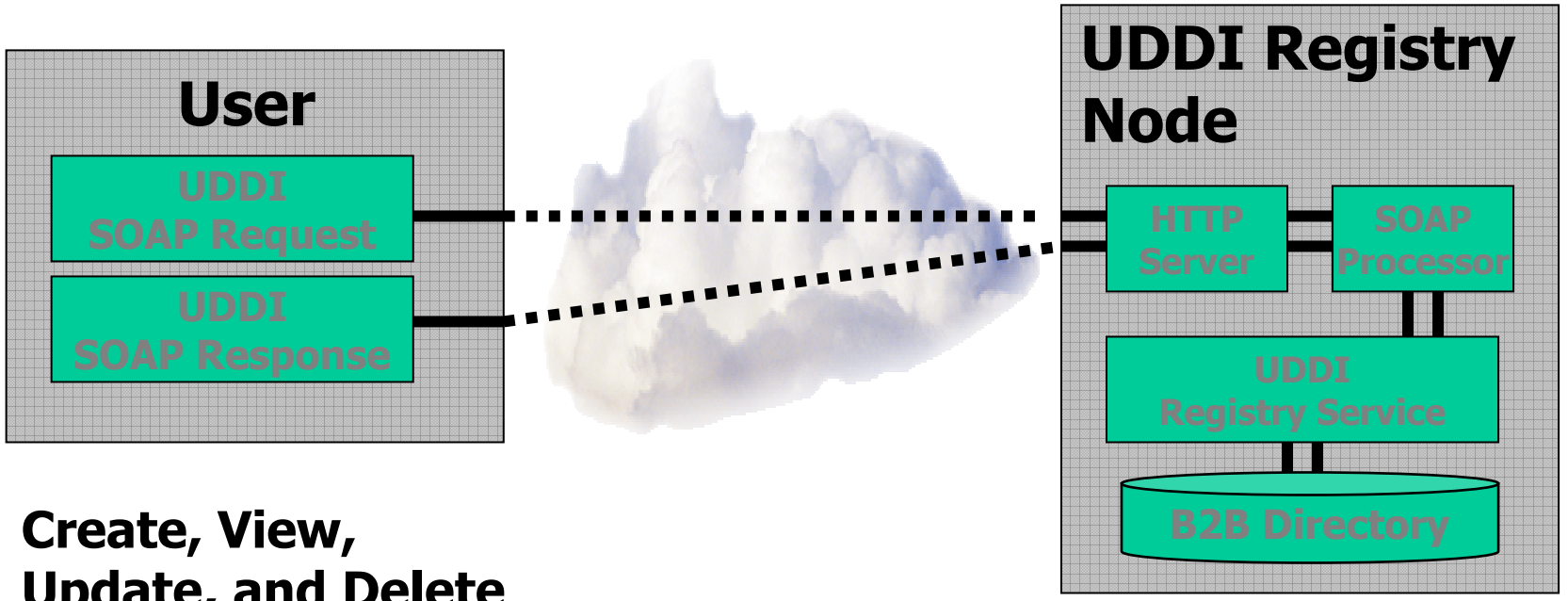
Universal Description, Discovery, and Integration (UDDI)

- Repository of business data for the Web
- XML schemas define data
- SOAP APIs for registration/search
- Contains search and contact information for businesses
- Includes pointers to WSDLs and other web service descriptions

UDDI model


UDDI repository
SOAP client
SOAP server /
Web Service description

UDDI and SOAP



Create, View, Update, and Delete registrations

Implementation-neutral

Registry APIs (SOAP Messages)

- Inquiry API
  - Find things
    - find_business
    - find_service
    - find_binding
    - find_tModel
  - Get Details about things
    - get_businessDetail
    - get_serviceDetail
    - get_bindingDetail
    - get_tModelDetail
- Publishers API
  - Save things
    - save_business
    - save_service
    - save_binding
    - save_tModel
  - Delete things
    - delete_business
    - delete_service
    - delete_binding
    - delete_tModel
  - security…
    - get_authToken
    - discard_authToken

Simple CORBA Object Access Protocol (SCOAP)

- Proof of Concept extending CORBA's reach to new protocol domains
  - COM/DCOM
  - EJB
  - now SOAP/HTTP
- GIOP messages now carried over another transport protocol
- Interoperate with non-CORBA clients

Jboss/ZOAP

- http://www.jbossgroup.com/business/jboss-zoap.html
- Zero-Effort Object Access Package (ZOAP) is an Open Source SOAP implementation for the Java2 platform running under the GNU General Public License

Electronic Business XML (ebXML)

- Initiative established by UN/CEFACT and OASIS – **replacement for EDI**
- Top-down approach (start with business process)
- Massive specification set for transport, repository, trading partner agreements, vocabularies, etc.
- Sort of competition for SOAP, WSDL, UDDI (all same pieces)

Simple CORBA Object Access Protocol (SCOAP)

- Proof of Concept extending CORBA's reach to new protocol domains
  - COM/DCOM
  - EJB
  - now SOAP/HTTP
- GIOP messages now carried over another transport protocol
- Interoperate with non-CORBA clients

Jboss/ZOAP

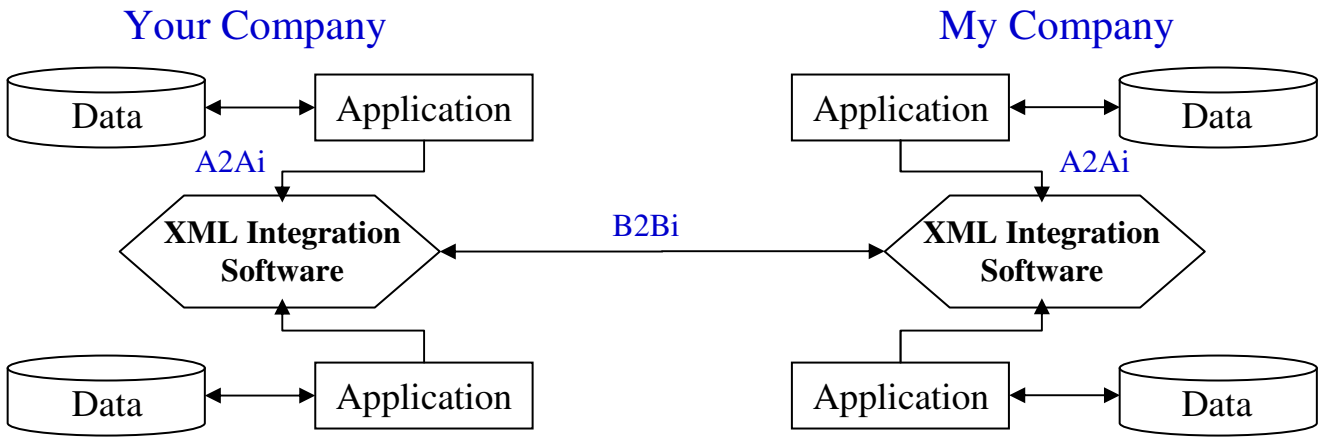
- http://www.jbossgroup.com/business/jboss-zoap.html
- Zero-Effort Object Access Package (ZOAP) is an Open Source SOAP implementation for the Java2 platform running under the GNU General Public License

XML compared with EDI

- EDI requires **proprietary software** over **private networks** (Value Added Networks).
- XML uses the Internet (FTP, HTTP, HTTPS)
- EDI **requires dedicated servers**, special C++ programming, and **high priced** consultants
- XML has no specific hardware requirement, and can be programmed in any language
- EDI data formats are **difficult to learn** and read and requires specialized expertise
- XML is well known, requires no special software purchase, and leverages open standards

Future View

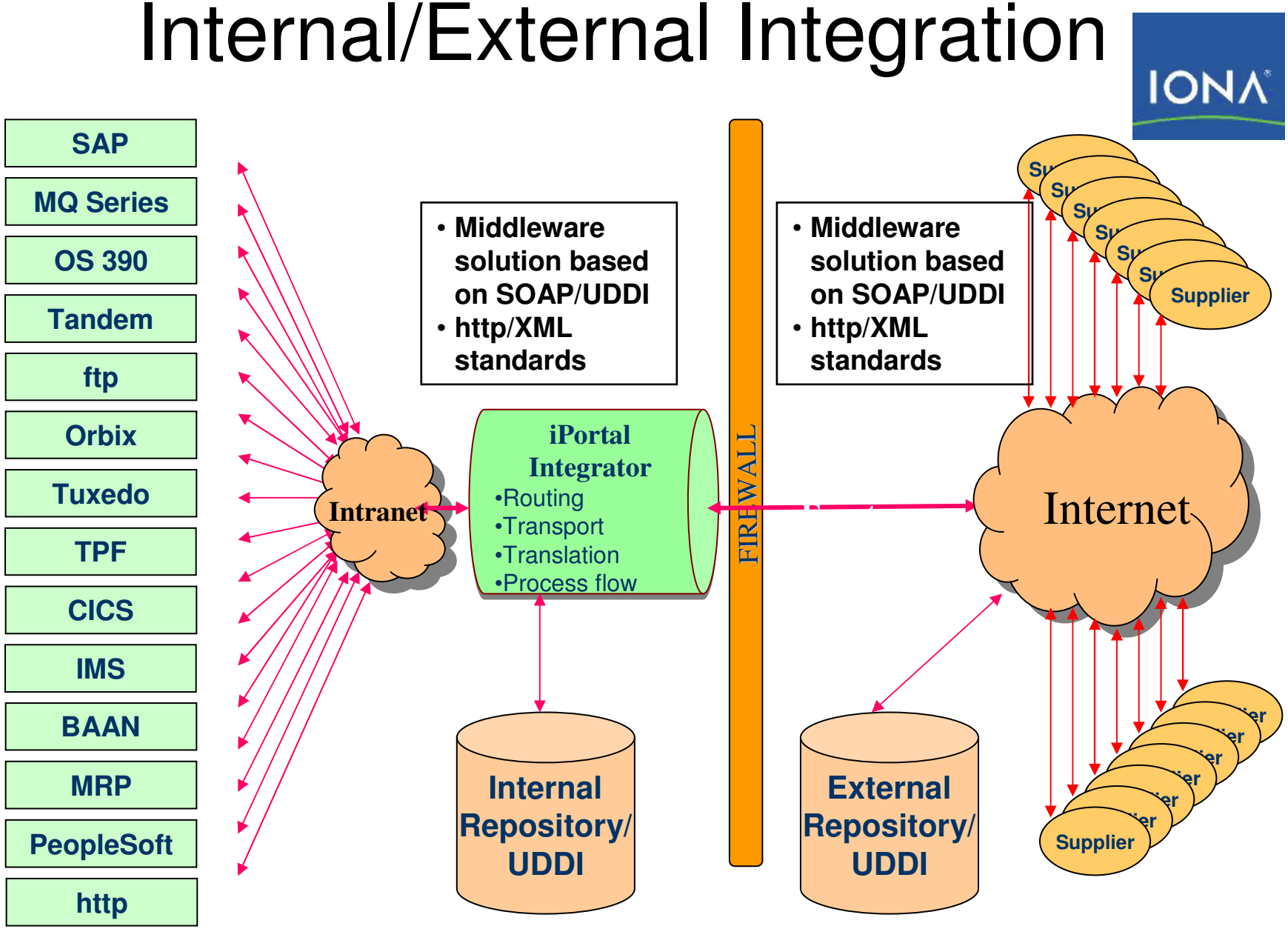
- **Applications will all speak XML** and talk to each other using **standard integration software**
- Open standards such as SOAP, WSDL, UDDI, ebXML, and RosettaNet will ensure compatibility
- XML is flexible and extensible enough to make it happen



Competitive Landscape

- Competing standards:
  - MSFT, IBM driving SOAP, WSDL to W3C standardization (bottom up)
  - Sun, IBM driving ebXML (top down)
  - MSFT driving .NET
  - Sun incorporating XML into J2EE
- *Everyone* focusing on Web Services
- Lots of proprietary stuff still out there
- Lots of new ISVs implementing XML

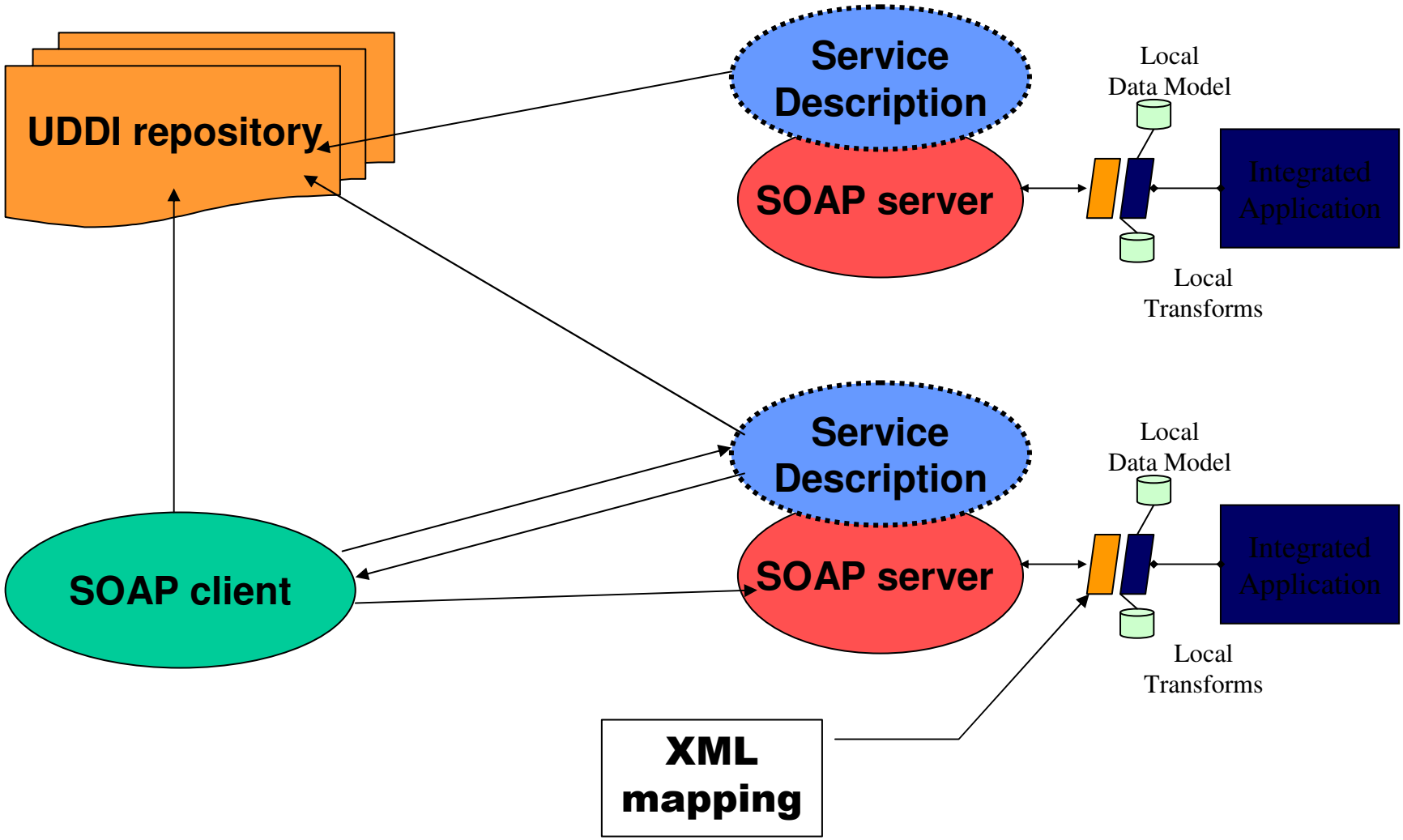
Internal/External Integration



SOAP for Intranets

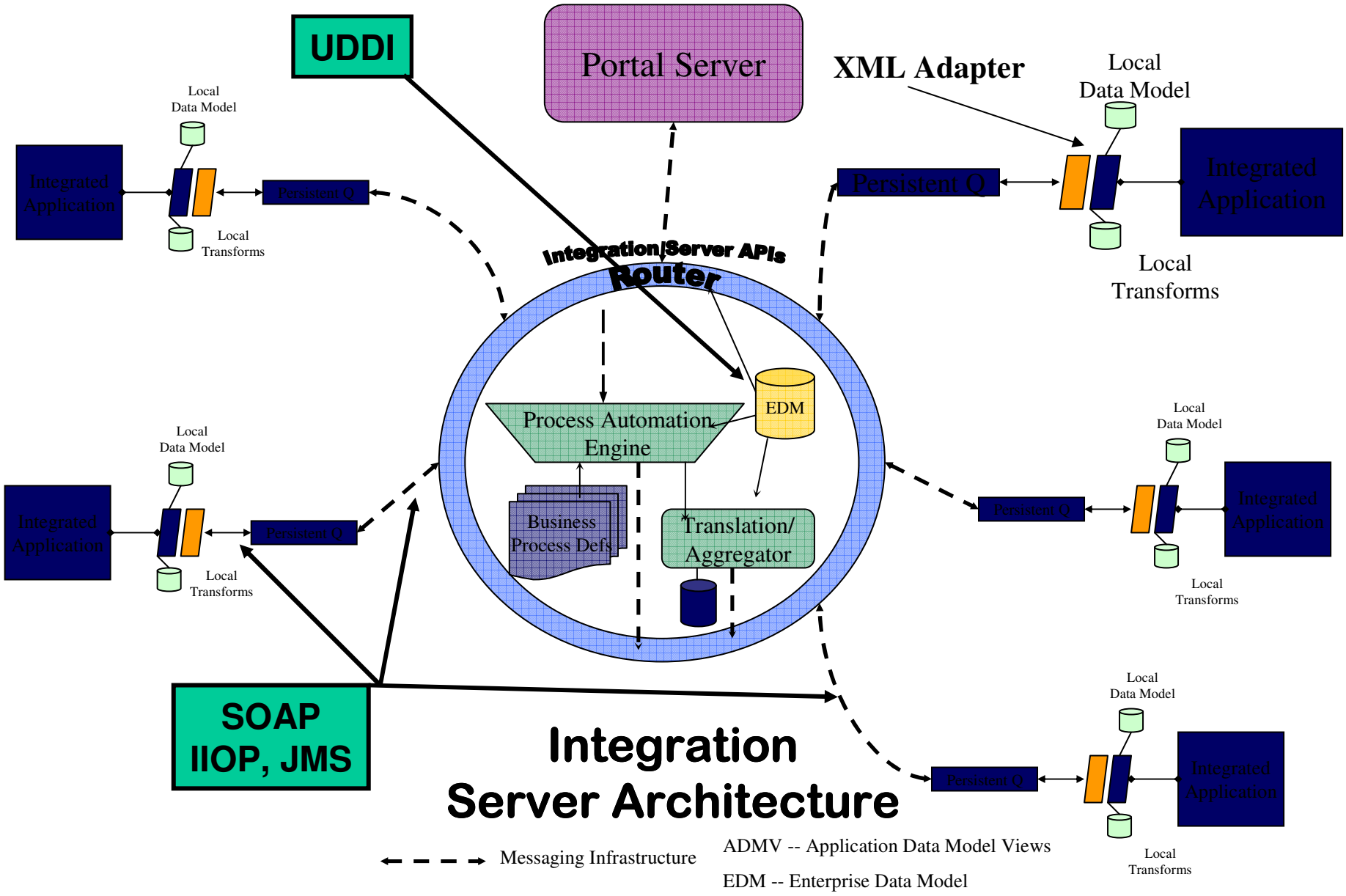
- Define internal use Web services
- Register with internal UDDI
- Extend UDDI for internal use (additional search/description fields, etc.) & standardize
- Include basic process flow in service definition
- Clients discover service definitions through UDDI
- Service definitions point to routers, metadata repositories, etc. for incremental introduction

XML for integration (A2A, B2B)

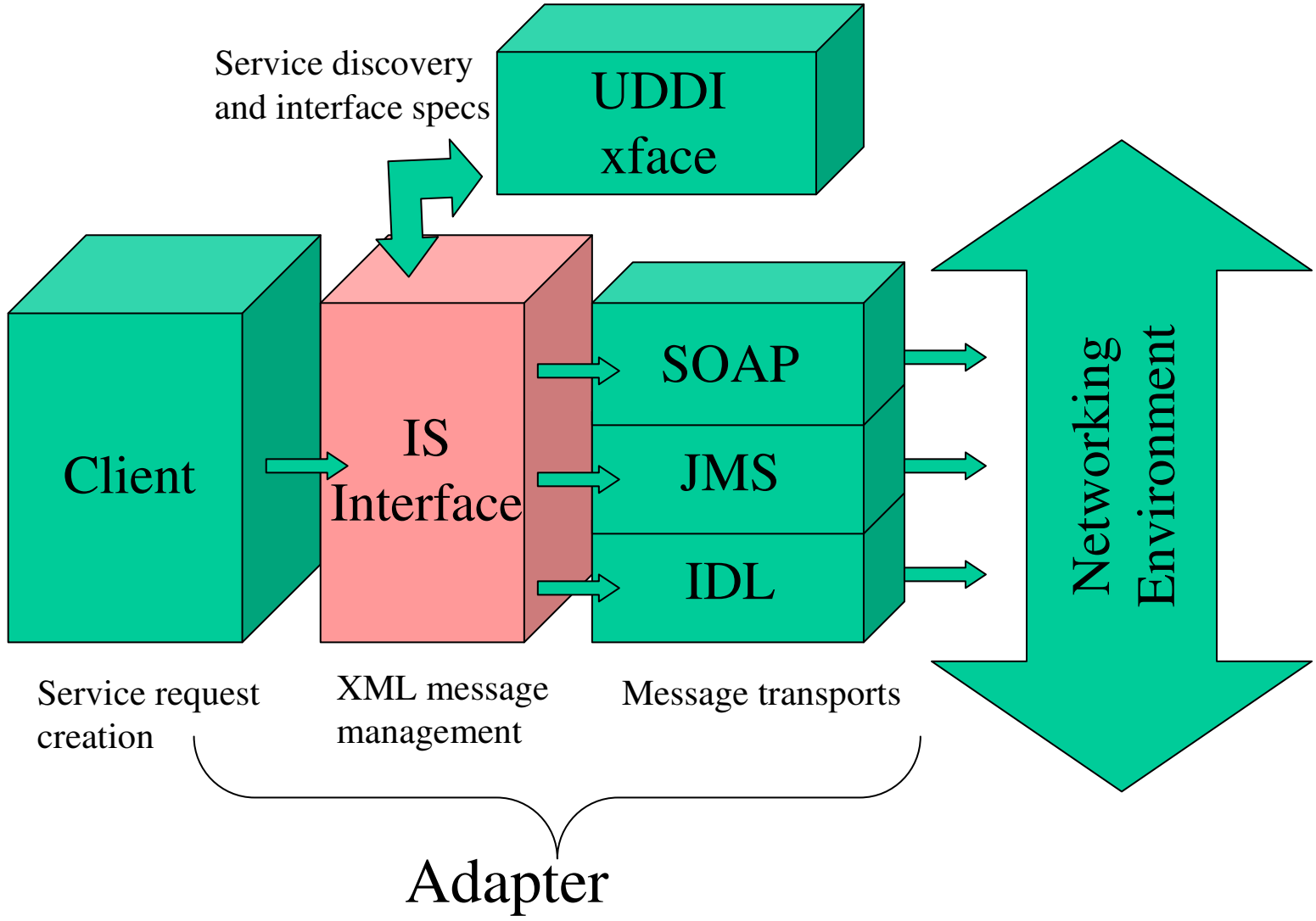


iPortal Integrator and UDDI/SOAP

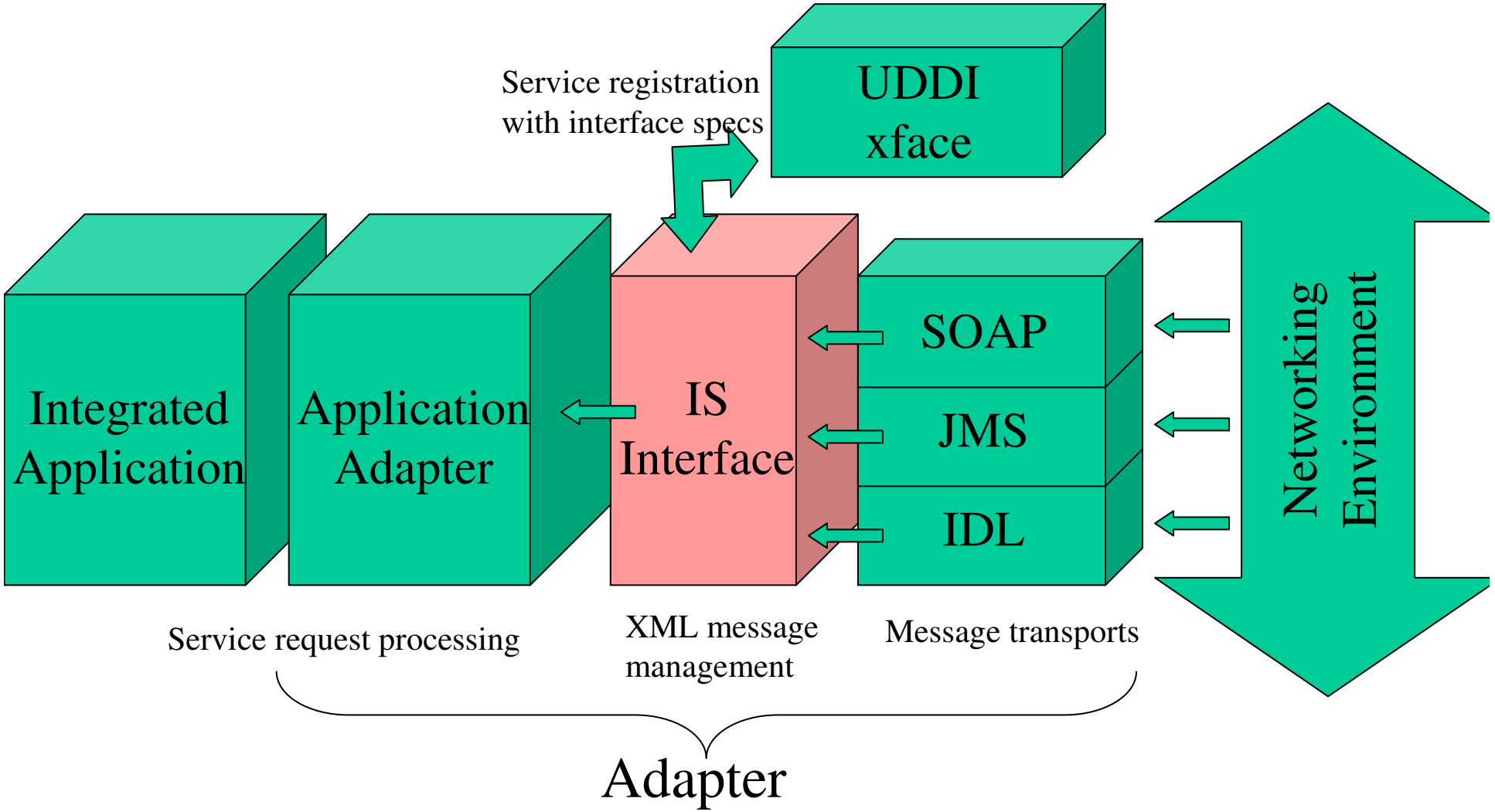
- iPIntegrator service definition, registration and discovery - thru WSDL/UDDI/SOAP
  - Interface classes hide UDDI
  - Add transformation, routing, aggregation
- iPIntegrator service delivery thru multiple transports: JMS, IIOP, and SOAP
- Common approach, multi-protocol
- Works for B2B and A2A or both


UDDI
Portal Server
XML Adapter
Local
Data Model
Integrated
Application
Persistent Q
Local
Transforms
Integration Server APIs
Router
EDM
Process Automation
Engine
Business
Process Defs
Translation/
Aggregator
SOAP
IIOP, JMS
Integration
Server Architecture
Messaging Infrastructure
ADMV -- Application Data Model Views
EDM -- Enterprise Data Model

Service Requestor



Service Provider



Major Players John Hawley - ZEFER

- CrossWorlds - eBusiness Modules
- IBM- WebSphere B2Bi
- IONA-IONA suite (iPortal AppSvr, Orbix2000, iPortal Integrator, …)
- iPlanet -Commerce Platform - app server
- Mercator - Commerce Broker
- Microsoft - BizTalk, MS.Net
- TIBCO - ActivePortal, ActiveExchange
- Vitria - BusinessWare
- Vignette - eSeries - cms/app server

References

- www.w3.org/xml (W3C consortium site)
- www.xmledi-group.org (XML/EDI group site)
- www.xml.com (XML info site)
- www.uddi.org
- www.ebxml.org
- www.oasis-open.org
- www.iona.com, www.iona.com/soap
- msdn.microsoft.com/soap (Microsoft SOAP site)
- www.develop.com (DevelopMentor SOAP site)
- http://www.startkabel.nl/k/xml/ (good XML site)
- http://www.soap-wrc.com/webservices/default.asp

Diverse Protokoll-Konzepte

- **XML-RPC - Userland**
  - **http://www.xmlrpc.com**
- **SOAP (Simple Object Access Protocol) - Microsoft**
  - **http://msdn.microsoft.com/workshop/**
  - **xml/general/soaptemplate.asp**
  - **http://www.develop.com/soap/**
- **XMLTP.org**
  - **"an online community effort to standardize the transport mechanism for XML data"**
  - **http://www.xmltp.org/**

Diverse Protokoll-Konzepte (2)

- **Nepal XP - Innovision**
  - **Framework um XML-basierte Kommunikationsprotokolle zu definieren**
  - **http://www.innovision.com/**
- **ODBC Socket Server**
  - **ODBC Socket Server is an open source database access toolkit**
  - **that exposes Windows NT ODBC data sources with an XML-based**
  - **TCP/IP interface**
  - **http://odbc.linuxbox.com/index.html**

Diverse Protokoll-Konzepte (3)

- **Information and Content Exchange (ICE) Protocol**
  - **http://www.w3.org/TR/NOTE-ice**
  - **Datenbankschnittstellen**
  - **XML SQL – Microsoft http://www.microsoft.com/xml/**
  - **XSQL Servlet - Oracle http://www.oracle.com/xml**

Jboss/ZOAP

- http://www.jbossgroup.com/business/jboss-zoap.html
- Zero-Effort Object Access Package (ZOAP) is an Open Source SOAP implementation for the Java2 platform running under the GNU General Public License