

Panorama sur les Web Services

Didier DONSEZ

Université Joseph Fourier (Grenoble 1)

IMAG/LSR/ADELE

Didier.Donsez@imag.fr

Agenda Web Services

→ **Definition**

→ **Core Technologies**

➤ SOAP

➤ WSDL

➤ UDDI

→ **Related Technologies**

→ **Web Services and Security**

What's a Web Service ???

→ What are Web Services ?

- Business objects that can be deployed and combined over the internet, with loose dependence on technologies and protocols.
- (Components of) modular applications deployed over internet technologies, with a predictable format and behaviour.

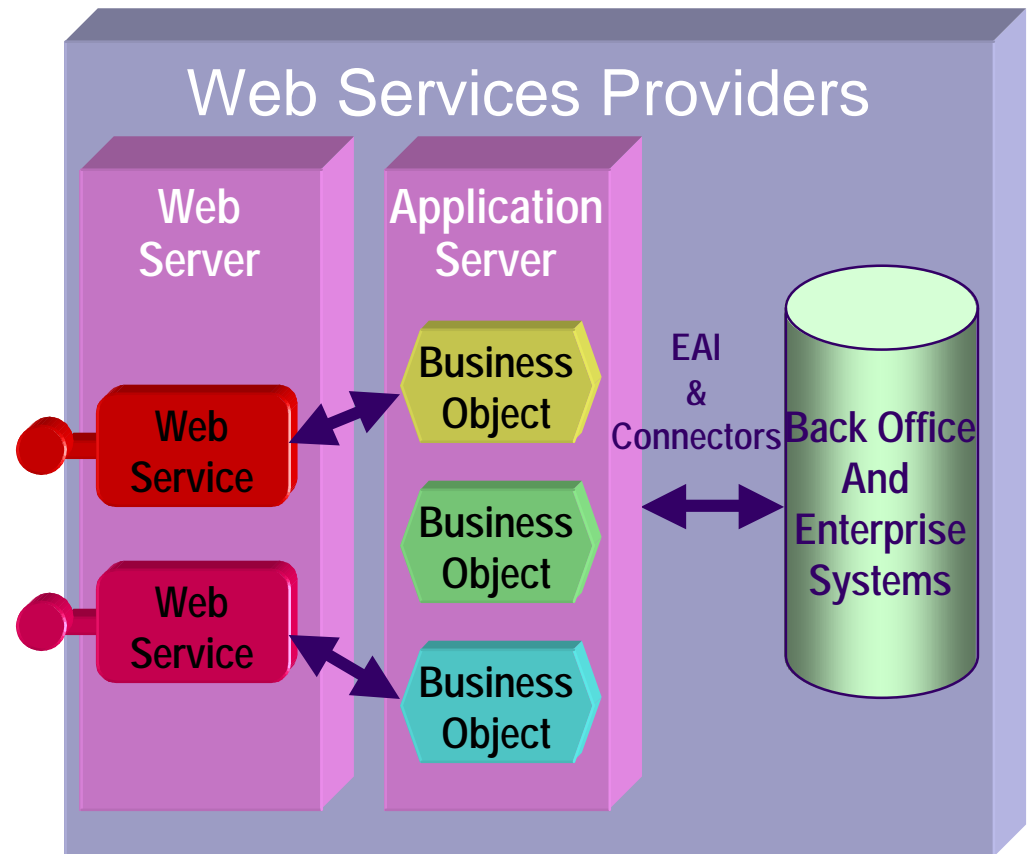
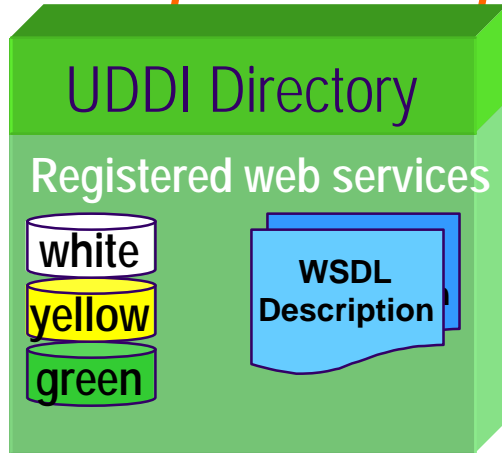
→ Yet another distributed objects technology ?

- Remote invocation of Web services **SOAP**
- Describe Web services **WSDL**
- Register & Discover Web services **UDDI**

→ XML-based standards

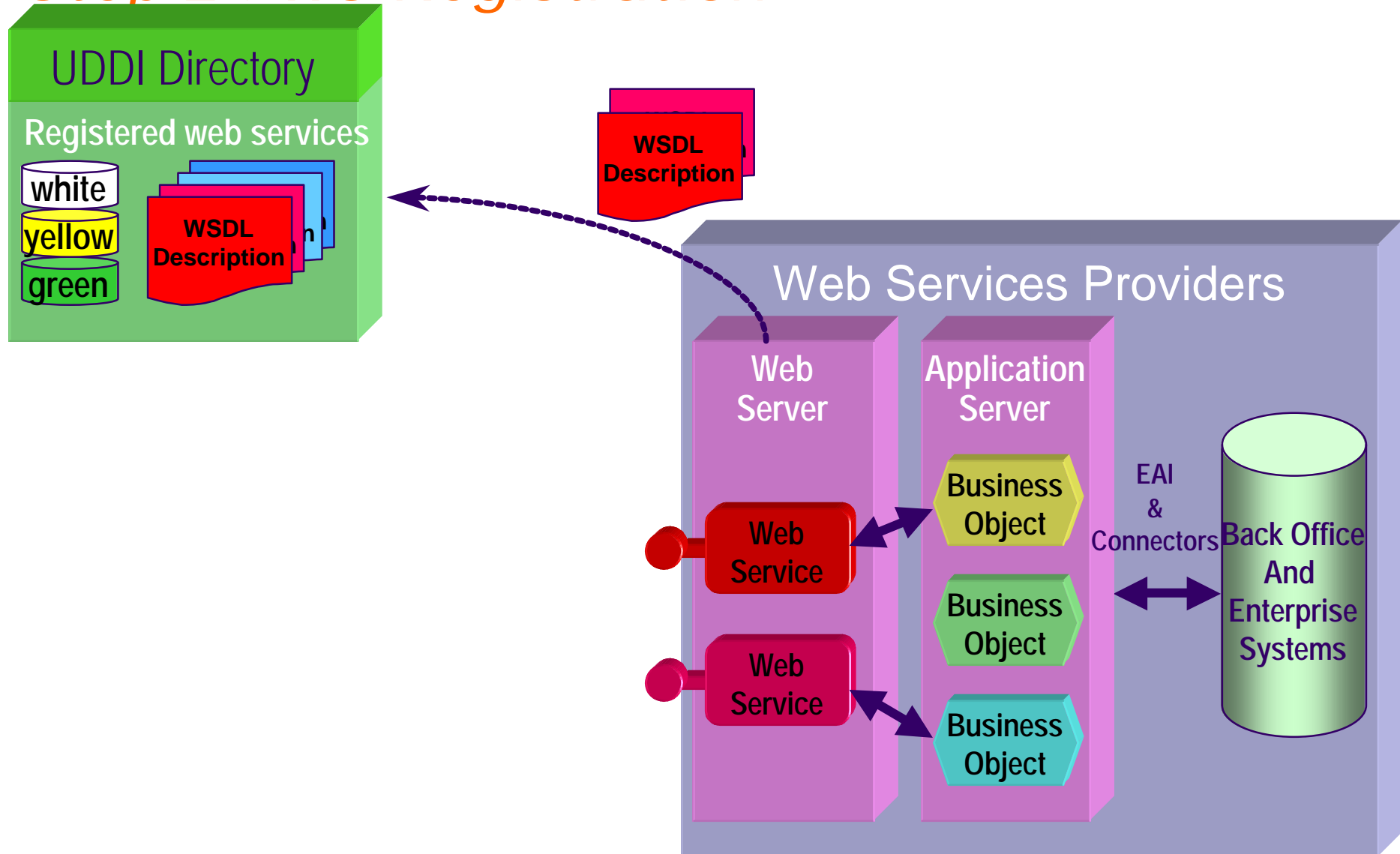
Architecture of an Web Service

Step 1 : Deployment



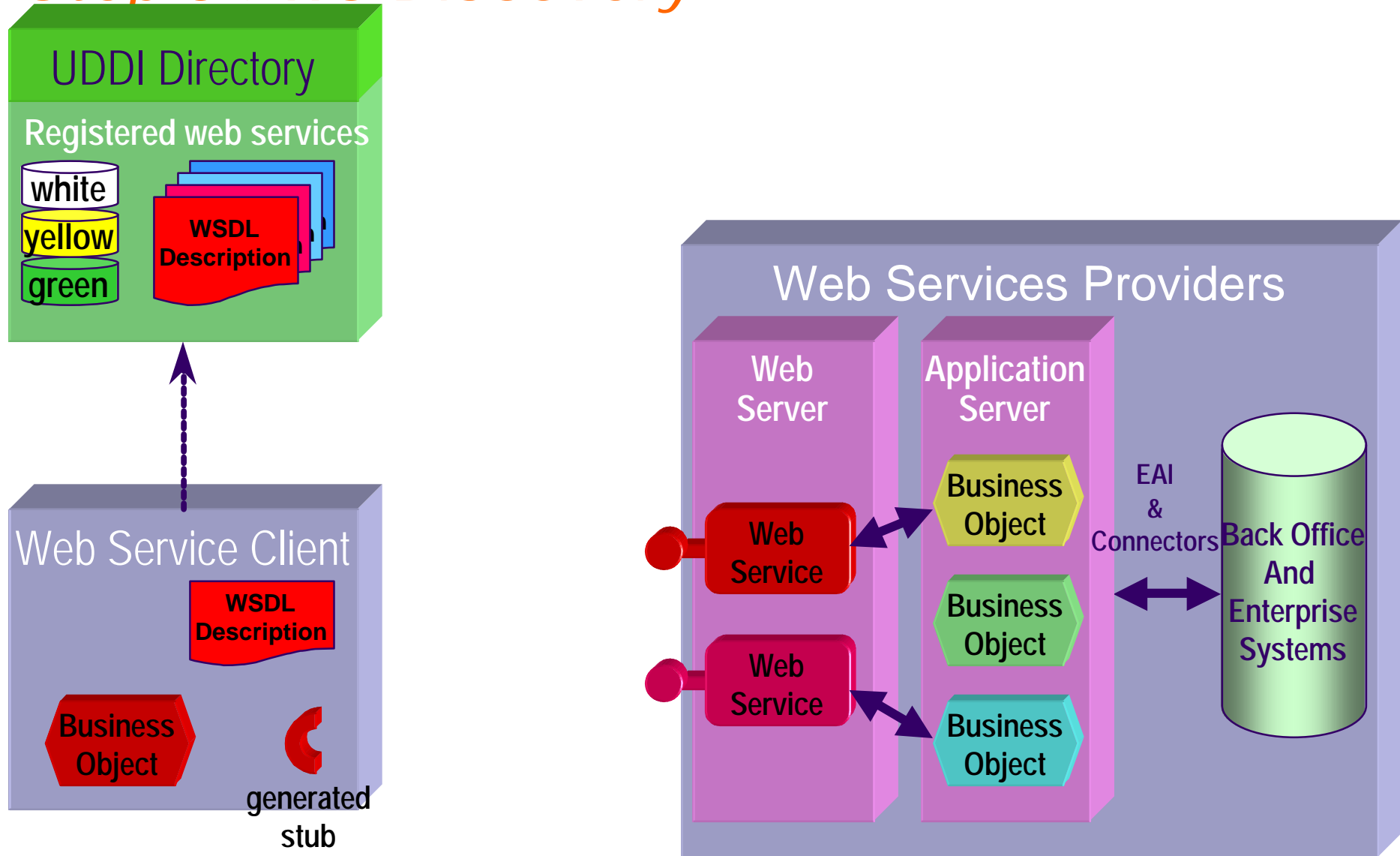
Architecture of an Web Service

Step 2 : WS Registration



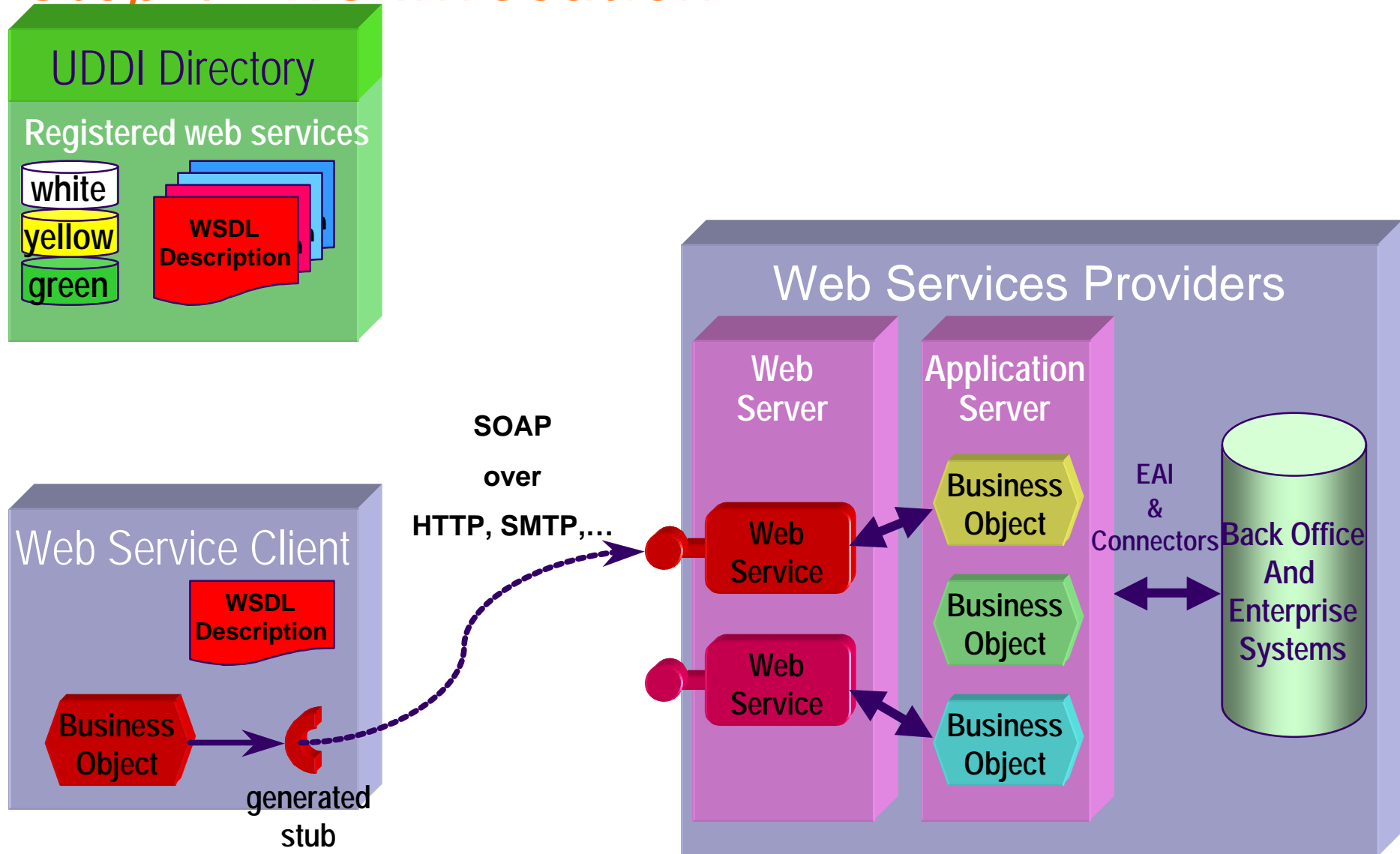
Architecture of an Web Service

Step 3 : WS Discovery



Architecture of an Web Service

Step 4 : WS Invocation



Who is involved ?

→ **Targets**

- B2B (Business To Business)
- EAI (Enterprise Application Integration)
- P2P (Peer to Peer)

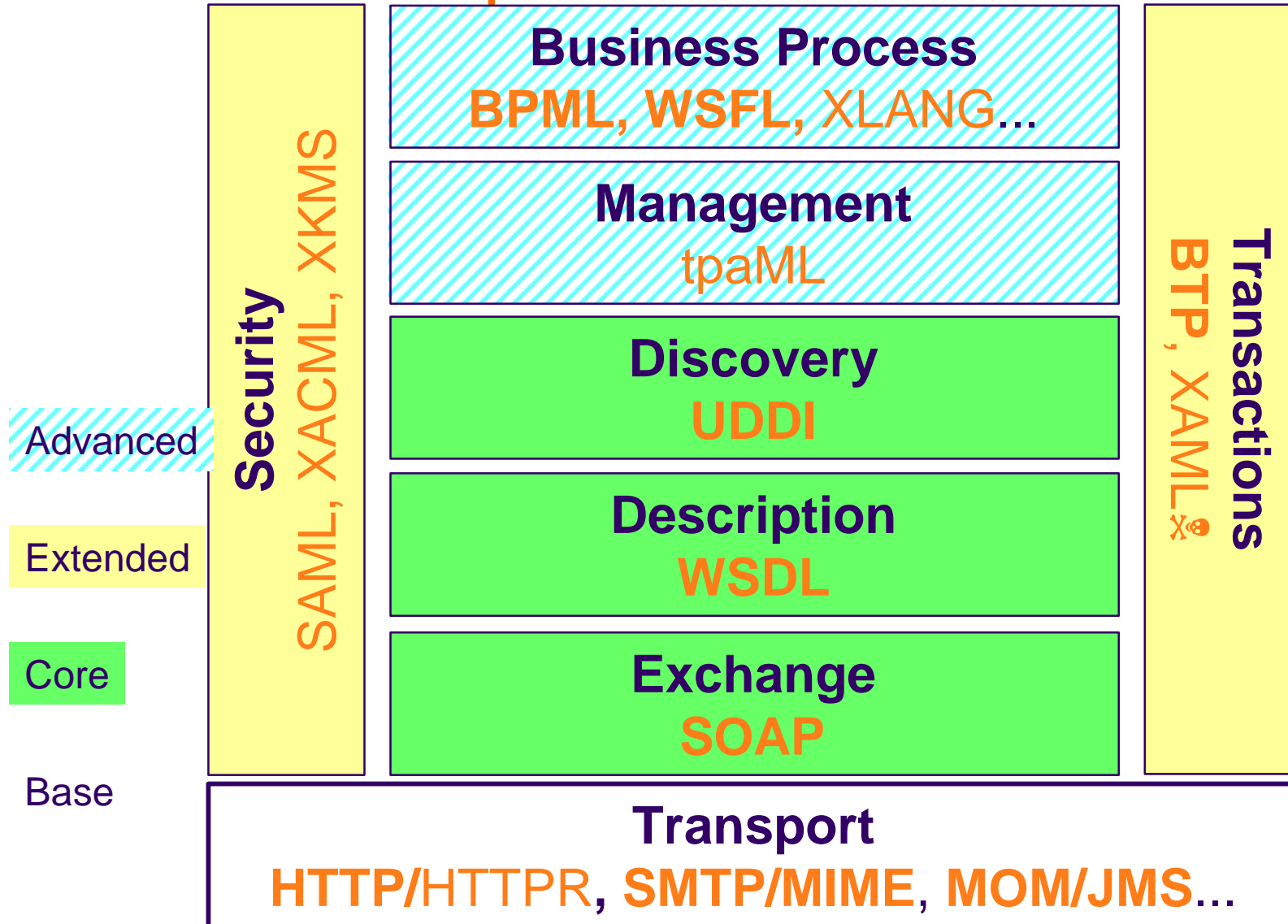
→ **Defining the standards**

- W3C (SOAP, WSDL)
- Ariba/IBM/Microsoft (UDDI)

→ **Frameworks & Developing tools**

- MicroSoft .NET
- Sun JavaONE : J2EE + WebServices (JAXy $y \in \{M, P, R, \dots\}$)
- Apache, IBM, Oracle, BEA, Iona, Enhydra

Web Services Architecture: A wider picture



D'après M.Pontacq, Evidian

Erwan Donsez, Web Services

SOAP

Simple Access Object Protocol

→ **Transport**

- HTTP, RPC, MOM (JMS), SMTP/POP3/IMAP4 ...

→ **XML Encoding**

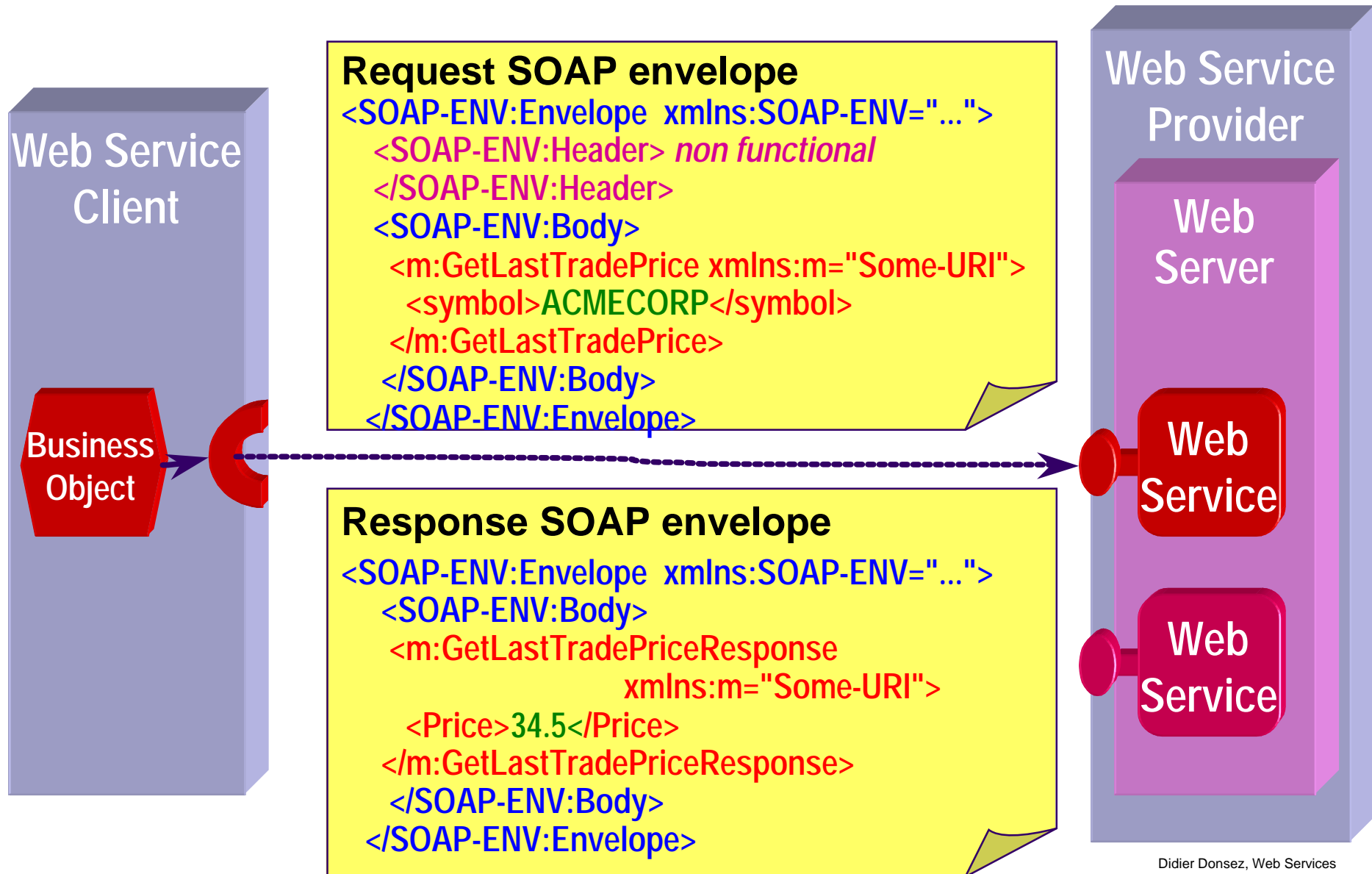
- XML serialization of simple/complex types

→ **Message Structure : Envelope**

- **HEADER** : non functional aspects
 - Session Context, Security context, Transaction context, ...
- **BODY** : functional aspects
 - Business methods, ...

→ **Example ...**

SOAP Request / Response



Deploy an JOnAS EB as an Apache/SOAP WS

→ Step 1 : Install (Apache &) TomCat & Apache/SOAP Servlet

→ Step 2 : Deploy an (existing) Enterprise Bean with this file

```
<isd:service xmlns:isd="http://xml.apache.org/xml-soap/deployment"  
  id="urn:lastprice_web_service">  
  <isd:provider      type      ="org.apache.soap.providers.StatelessEJBProvider"  
                    scope     ="Application"  
                    methods  ="create lastprice"  
  <isd:option key="JNDIName" value="tradeprice"/>  
  <isd:option key="FullHomeInterfaceName"  
    value="com.mycomp.trade.TradePriceHome" />  
  <isd:option key="ContextProviderURL"  
    value="jrm://trade.mycomp.com:12340/tradeprice" />  
  <isd:option key="FullContextFactoryName"  
    value="org.objectweb.jeremie.libs.services.registry.jndi.JRMIInitialContextFactory"/>  
  </isd:provider>  
  <isd:faultListener>org.apache.soap.server.DOMFaultListener</isd:faultListener>  
</isd:service>
```

WSDL

Web Service Description Language

→ Describe services as a set of operations and messages bound to protocols and servers

→ Define a message

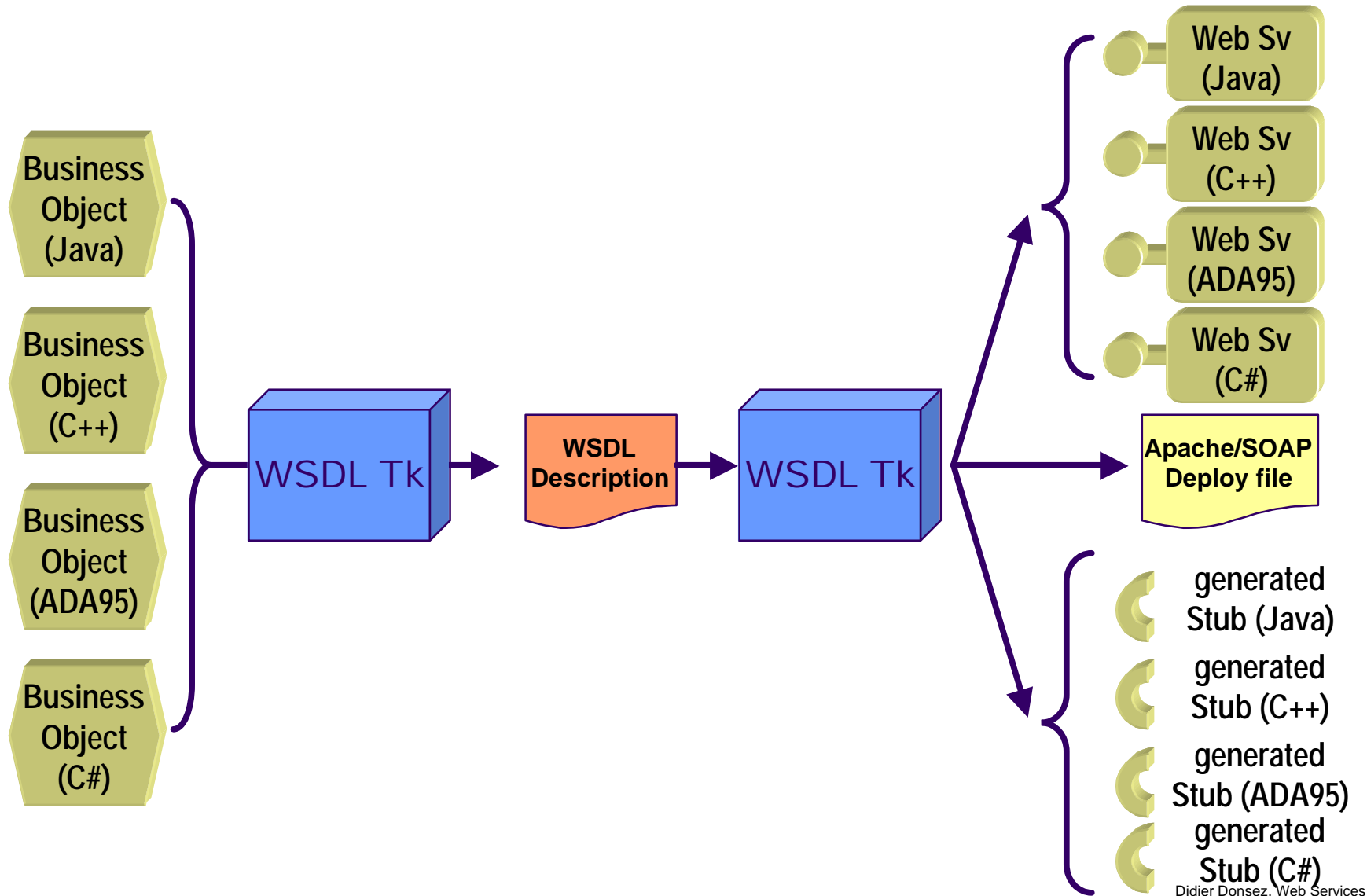
```
<message name="EntryMessage">  
  <part name="name" type="xsd:string"/>  
  <part name="address" type="typens:address"/>  
</message>
```

→ Define a request/response service

```
<portType name="AddressBook">  
  <operation name="addEntry">  
    <input message="EntryMessage"/>  
    <output message="EntryMessage"/>  
  </operation>  
</portType>
```

→ Define types, and bindings to transport protocols and servers.

Generation WSDL



UDDI

Universal Directory & Discovery Interface

→ **Worldwide directory of companies, services, products...**

- White pages,
- Yellow pages,
- Green pages

→ **"Green pages"**

- Namespace to describe how to use the service, etc...
 - Identifier of who published the service
 - Unique identifier (tModelKey) of this service for registration

→ **Accessing web services**

- Bindings declared in directory entries:
 - for example, (tModelKey, URL) associations

→ **UDDI directories, search engines**

- xmethods.net, soapware.org, salcentral.com, soap-wrc.com, ...

Other related technologies

➔ **WSFL/XLang (Flow), WSCL (Conversation), ...**

➔ **“Full Packages”**

➤ **ebXML (OASIS/UN initiative)**

- define all the layers in the web services stack. That includes categories such as registries, business process modeling, service descriptions, and transport/packaging/messaging.

➤ **e-speak (open software platform)**

- designed for supporting the description, registration, and discovery of e-services, the ability to compose multiple e-services into higher-level e-services, the ability to negotiate among e-services, and the ability to manage e-service interactions.

➤ **BizTalk/.NET (Microsoft initiative)**

- define all the layers in the web services stack. That includes four categories, registries(UDDI), business modeling languages (X-Lang), service descriptions (WSDL), and transport/packaging/messaging(SOAP).

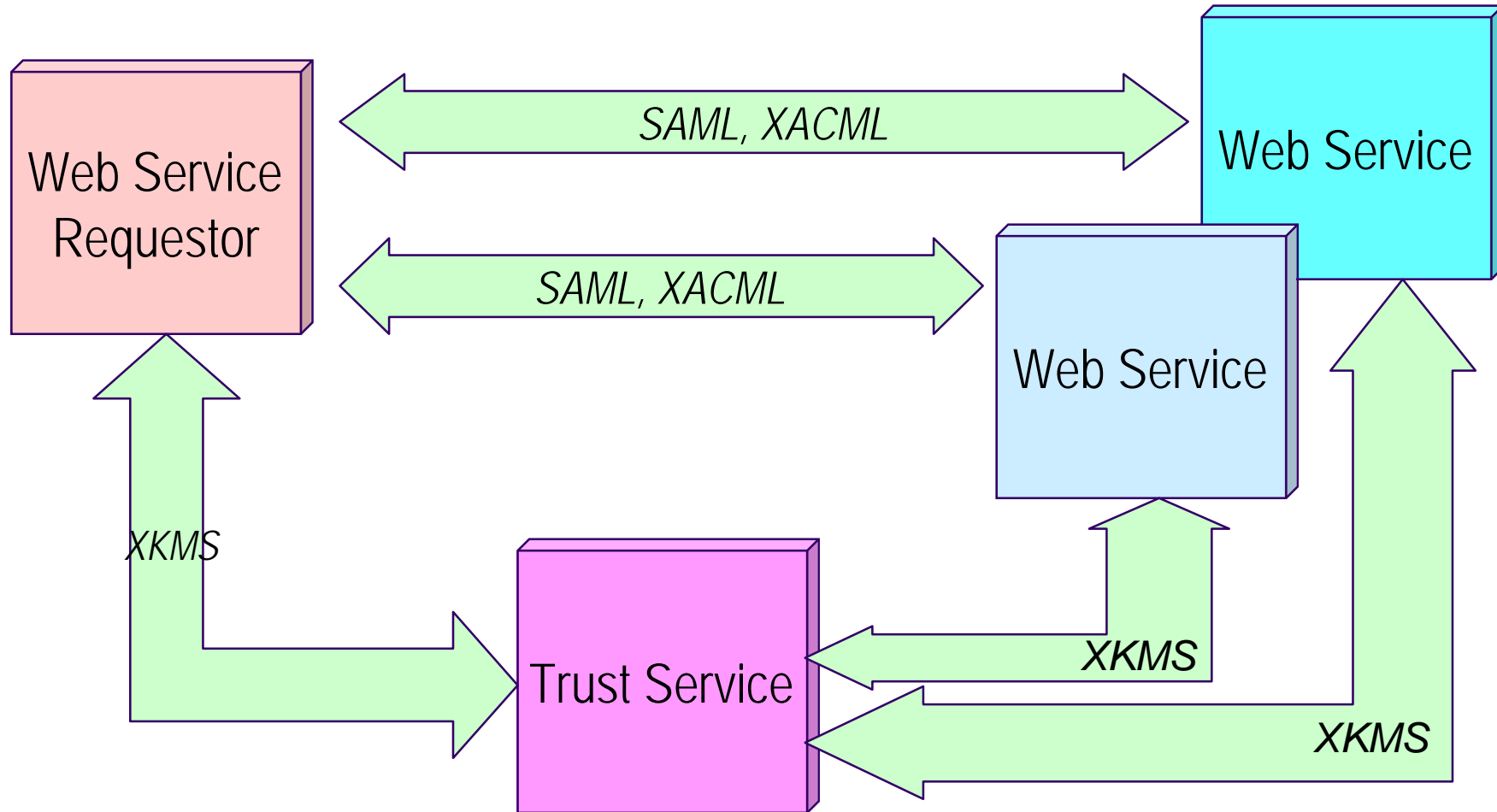
➤ **RosettaNET, ...**

Security and Web Services

SAML, XACML, XKMS

SSL/TLS, S/MIME, PGP
for transport layers

XML Sig, XML Enc
for presentation layers



Security and Web Services: SAML, XACML, XKMS

→ SAML (Security Assertion Markup Language)

- XML-based security standard for exchanging authentication and authorization information
- defines formats and protocols for exchanging and validating "security assertions", which identify a user or a user's access rights
- allows a single user session to span multiple enterprises and multiple security managers (**log in once at one site and access all partner sites**)

→ XACML (eXtensible Access Control Markup Language)

- complements SAML to define an XML schema for representing authorization policies for XML objects.

→ XKMS (XML Key Management Standard)

- XML way of representing PKI (Public Key Infrastructure) certificates and some of its operations (digital certification processing, revocation status checking, etc.)

→ **Remark: SAML, XACML, XKMS use XML Sig & Enc**

Trust (Web) Services

→ Motivation

- Gestion des PKI par des échanges basés sur des messages XML transportés sur SOAP.
 - En évitant d'utiliser la syntaxe ASN.1

→ Standards

- XKMS XML Key Management Specification
 - <http://www.xmltrustcenter.org/xkms>
 - Basé sur
 - XML Digital Signature <http://www.w3.org/TR/xmlsig-core/>
 - XML Digital Encryption <http://www.w3.org/TR/xmlenc-core/>

XKMS - XML Key Management System

→ Motivation

- Remplacer les formats et protocoles PKI (PKIX, Card Management Services, OCSP, etc.) par des documents XML transportés par SOAP.

→ Définit les messages de requête et de réponse pour

- Requérir (request) un certificat
- Renouveler (renew) un certificat
- Valider (validate) un certificat (expiration, CRL, OCSP, etc.)
- Révoquer (revoke) un certificat (CRL)

→ Basé sur XML Signature & XML Encryption

→ W3C

- Initié
- XKMS XML Key Management Specification
 - <http://www.xmltrustcenter.org/xkms>
- API Java : JSR 104 XML Trust Service APIs

XKMS

Exemple de message de révocation

A request to revoke the key specified by <KeyID>

```
<?xml version="1.0"?>
```

```
<Request>
```

```
  <Prototype>
```

```
    <AssertionStatus>Invalid</AssertionStatus>
```

```
    <KeyID>unique_key_identifier</KeyID>
```

```
    <ds:KeyInfo>.....</ds:KeyInfo>
```

```
  </Prototype>
```

```
  <AuthInfo>
```

```
    <AuthUserInfo>
```

```
      <ProofOfPossession>[RSA-Sign]</ProofOfPossesion>
```

```
    </AuthUserInfo>
```

```
  </AuthInfo>
```

```
  <Respond>
```

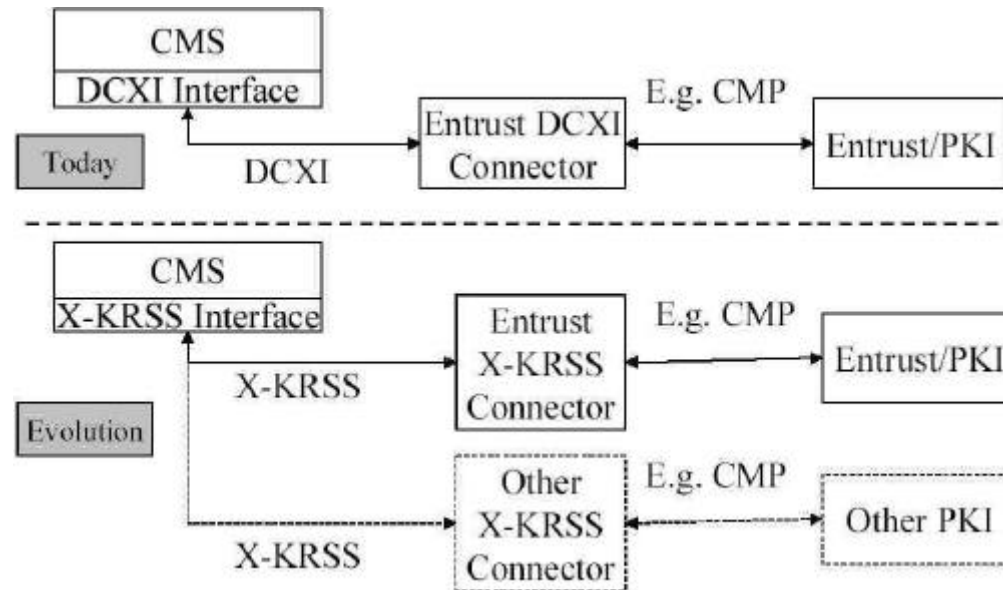
```
    <string>KeyName</string>
```

```
  </Respond>
```

```
</Request>
```

XKMS Extension

→ X-KRSS et CMS



→ X-BULK (Baltimore)

- gestion des clés pour des cartes SIM, des modems Cable, ...
- Motivations : batches

Web services vendors offering

- Hewlett-Packard's *e-Speak* and Netaction
- Microsoft's *.NET* Framework and *.NET MyServices*
- IBM's *Web Services Architecture*
- Oracle's *Dynamic Services*
- Sun's *Sun ONE*
- BEA *Web Services architecture* based on WebLogic E-Business Platform
- Iona
- SilverStream *SilverStream eXtend* (runs on Websphere, Weblogic and Silverstream)
- Systinet *WASP* (Web Applications and Services Platform) (available for Apache)
- CapeClear *CapeConnect*

Java ONE (Open Net Environment)

→ **Annoncé le 5/2/2001**

→ **Ajout des technologies Web Services dans J2EE**

➤ SOAP, WSDL, UDDI, ebXML, ...

→ **Nouvelles API (JSR)**

➤ **Context API**

● Interface vers les données contextuelles d'un service Web

➤ **JAX/RPC (Java API for XML based RPC)**

● Interface aux protocoles de transport de message XML de type RPC (SOAP, W3C XP, ...)

➤ **JAXB (Java API for XML Data Binding)**

● Correspondance automatique entre des objets Java et des données XML

➤ **JAXM (Java API for XML Messaging)**

● Interface aux protocoles de transport de messages XML : SOAP, ebXML Message Service

➤ **JAXP (Java API for XML Processing)**

● Gestion des arborescence DOM

➤ **JAXR (Java API for XML Registries)**

● Interface aux annuaires ebXML Registry/Repository et UDDI Business Registry

➤ **JAXTX (XML Transactioning API for Java)**

● Interface pour l'encapsulation et le transport de contextes transactionnels ACID et avancées dans SOAP, ...

Web services and Open source arguments

- **Thesis: shift towards Web services has reduced the attractiveness of the current generation of Open Source Web products**
- **Microsoft's Webserver has gained 5% in market share at the expense of Apache**
 - They have turned the "good enough" argument against us. IIS on .NET is a Web services delivery platform, but it's also a "good enough" Webserver, if that's all the customer wants today
- **What we need today is no longer a Web Server, but a Web services delivery platform**
- **What we need to do now is add Open Source implementations of Web services standards (WSDL, UDDI, ebXML) to open source J2EE platforms**

Actually, the choice is between either a Java-Linux world, or a .NET world

Open source Web services projects(1)

→ *Apache XML project*

- includes a *Soap project* that is an implementation of the draft W3C SOAP protocol. It is based on, and supersedes, the IBM *SOAP4J* implementation

→ *jUDDI (bowstreet)*

- is an open source Java-based implementation of a UDDI registry and a toolkit for developers to build access to UDDI registries within their own applications. **jUDDI has been architected to allow it to act as the UDDI front-end on top of existing directories and databases.**

→ *UDDI4J (IBM)*

- is a Java class library that provides an API to interact with a UDDI registry. It can be used to easily develop a Java application that can publish and find services in a UDDI registry.

Open source Web services projects(2)

→ *WSDL4Py*

- an open-source Python library for WSDL 1.1 from IBM

→ *WSDL4J*

- for Java Toolkit (IBM) allows the creation, representation, and manipulation of WSDL documents describing services

→ *Enhydra kSOAP*

- *kSOAP* is an SOAP API suitable for the Java 2 MicroEdition, based on *kXML* (a **lean XML API with optional WAP support -WBXML/WML-**), and *kUDDI*

Benefits

→ Multiple languages, platforms, vendors

- Java RMI requires java
- DCOM requires Windows

→ Independent standard organization (W3C)

- Unlike Java RMI (Sun) and DCOM (Microsoft)

→ Simple and interoperable (what CORBA is not...)

- XML is a widely understood and accepted technology
- Easy to map from any language
- Suitable for large and small systems:
 - PDAs, Cell Phones, Consumer Electronics, Home Portals, ...

Comparaison

	RMI	RPC	DCOM	CORBA	SOAP
Qui	SUN	SUN/OSF	MicroSoft	OMG	W3C
Plate-formes	Multi	Multi	Win32	Multi	Multi
Langages de Programmation	Java	C, C++, ...	C++, VB, VJ, OPascal, ...	Multi	Multi
Langages de Définition de Service	Java	RPCGEN	ODL	IDL	WSDL
Réseau	TCP, HTTP, IIOP customisable	TCP, UDP	IP/IPX	GIOP, IIOP, Pluggable Transport Layer	HTTP, HTTPR SMTP, MOM
Firewall	Tunneling HTTP				HTTP
Nommage	RMI, Courtage JINI	IP+Port	IP+Nom	Sv Nom, SvCourtage	IP+Port, URL Courtage UDDI
Transaction	Non	Non	MTS	OTS, XA	Extension applicative dans le header (BTP)
Sécurité	SSL, JAAS	Non	??	SSL	SSL, XKMS, XACML, SAML
Extra	Chargement dynamique des classes			Services Communs Services Sectoriels	