

XML-based Network and Service Management

School of Electronics and Information
Kyung Hee University

Choong Seon HONG
cshong@khu.ac.kr

Contents

- ❑ Introduction to XML
- ❑ Use of XML for Network and Service Management
 - ❑ Related Activities
 - ❑ Information Model for Interaction between DSL SPs
 - ❑ Web-based network management
- ❑ XML platforms and DPE
 - ❑ XML based technology interoperability
 - ❑ SOAP, UDDI, WSDL

XML: An Introduction

- ❑ eXtensible Markup Language (XML)
- ❑ A simplified subset of the Standard Generalized Markup Language (SGML)
- ❑ A replacement for HTML
- ❑ Displaying or managing Information ?
- ❑ Features
 - ❑ XML is very simple
 - ❑ XML is self-describing
 - ❑ XML is well documented

XML: An Introduction

□ Representing HTML in XML

```
<?xml version="1.0"?>
<body>
  <center>
    <b>
      An HTML example
    </b>
  </center>
</body>
```

An HTML example

XML: An Introduction

□ A Telecom example

```
<?xml version="1.0" ?>
<?xml:stylesheet type="text/xsl" href="schema.xsl"?>
<n-entityCreation id="1">
  <ems id="TaskCenter">
    <mgdEltR1 id="NE1"
adminState="unlocked" opState="enabled">
      <equipHolder id="1" equipHolderType="Bay">
        <equipHolder id="3" holderType="Shelf">
          <equipHolder id="4" holderType="Slot">
            </equipHolder>
          </equipHolder>
        </equipHolder>
      </mgdEltR1>
    </ems>
  </n-entityCreation>
```

XML Declaration

PI target

PI value

The prolog of the XML document

XML: An Introduction

- ❑ Comparing the Documents
 - ❑ The first (HTML) example contains rendering information, but not semantic information
 - ❑ The second (Telecom) example contains semantic information, not rendering information
 - ❑ XML is agnostic about what the tags and values mean
 - ❑ XML is only a semantic for Document Markup
 - ❑ XML is **not** a semantic for a Document
 - ❑ A document type syntax is defined by a Document Type Definition or an XML Schema

XML: An Introduction

- Document Type Definitions (DTD)
 - Document Type Definitions were required for SGML, optional for XML
 - XML determines the structure of a **well-formed** document
 - Matching tag start/end, correct nesting
 - A DTD determines the structure of a **valid** document
 - Valid structures, permitted attributes and values
 - A DTD consists of definitions for
 - Elements (Things that may contain things and/or data)
 - Attributes (Data about elements)
 - Entities
 - DTDs are being superseded by XML Schema (later)

XML: An Introduction

□ DTD Fragment

```
<!ELEMENT EMPLOYEE (SAL?, FIRST, MI*, LAST) >
<!ELEMENT SAL>
<!ELEMENT FIRST (#PCDATA) >
<!ELEMENT MI (#PCDATA) >
<!ELEMENT LAST (#PCDATA) >
<!ATTLIST SAL salutation (MR|MRS|MS) #IMPLIED>
```

Attribute default

```
<Employee>
  <SAL salutation="MR" />
  <FIRST>EDWARD</FIRST>
  <MI />
  <LAST>JONES</LAST>
</Employee>
```


XML Topics

- ❑ Namespaces
- ❑ XML Schema
- ❑ UML and XML
- ❑ Xlinks
- ❑ Xpath
- ❑ Xinclude
- ❑ XPointers
- ❑ Several APIs
 - ❑ SAX
 - ❑ DOM
 - ❑ JDOM
 - ❑ Parser specific APIs

XML Topics : Namespaces

- ❑ A Namespace is a set of unique names
- ❑ Namespaces are important because the same name is often used in different documents to mean different things

```
<Address>  
  <Street>123 Main St.</Street>  
  <City>Anytown</City>  
  <State>NJ</State>  
</Address>
```

```
<Address>  
192.0.10.138  
</Address>
```

- ❑ XML Namespaces allow different document types to be combined without name conflicts
- ❑ This is essential in industries with many different standards organizations work independently
- ❑ Or where vendors want to extend a standard Schema

XML Topics : Namespaces

- An XML namespace consists of a two-part name
 - URI (Uniform Resource ID) formatted namespace
 - Element or Attribute name

Namespace prefix

attribute

URI

```
<cust:Address xmlns:cust="http://www.telcordia.com/xml/cust">  
  <cust:Street>123 Main St.</Street>  
  <cust:City>Anytown</City>  
  <cust:State>NJ</State>  
</Address>
```



```
<ip:Address xmlns:ip="http://www.telcordia.com/xml/ip">  
192.0.10.138  
</Address>
```

- XML Namespaces allow diverse schema to be mixed in a document without name clashes
- XML Namespaces do not exist in a conventional or static way - they are constructed anew each time the document is parsed

XML Topics : XML Schema

- ❑ Document Type Definitions (DTDs) are the current method to specify a document structure
- ❑ The DTD specifies only Element, Attributes and Entities. It can only specify the structure, but says little about the content
- ❑ XML Schema takes the next step to specify the content more precisely
- ❑ XML Schema has recently (2001-05-02) been approved as a W3C Recommendation

XML Topics : XML Schema (cont'd)

- ❑ XML Schema makes several additions to the DTD specification
 - ❑ Elements and Attributes are now typed
 - ❑ Datatypes are supported (ability to indicate the type of content an element or attribute contains)
 - ❑ All Datatypes can be extended or specialized
 - ❑ Namespaces provide for Schema reuse and specialization

XML Topics : XML Schema, Example (1)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
      Purchase order schema for Example.com. Copyright 2001
      Example.com. All rights reserved.
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="purchaseOrder" type="PurchaseOrderType"/>
  <xsd:element name="comment" type="xsd:string"/>

  <xsd:complexType name="PurchaseOrderType">
    <xsd:element name="shipTo" type="Address"/>
    <xsd:element name="billTo" type="Address"/>
    <xsd:element ref="comment" minOccurs="0"/>
    <xsd:element name="items" type="Items"/>
    <xsd:attribute name="orderDate" type="xsd:date"/>
  </xsd:complexType>
```

XML Topics : XML Schema, Example (2)

```
<xsd:complexType name="Address">
  <xsd:element name="name" type="xsd:string"/>
  <xsd:element name="street" type="xsd:string"/>
  <xsd:element name="city" type="xsd:string"/>
  <xsd:element name="state" type="xsd:string"/>
  <xsd:element name="zip" type="xsd:decimal"/>
  <xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/>
</xsd:complexType>
```

XML Topics : XML Schema, Example (3)

```
<xsd:complexType name="Items">
  <xsd:element name="item" minOccurs="0" maxOccurs="*">
    <xsd:complexType>
      <xsd:element name="productName" type="xsd:string"/>
      <xsd:element name="quantity">
        <xsd:simpleType base="xsd:positive-integer">
          <xsd:maxExclusive value="100"/>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="price" type="xsd:decimal"/>
      <xsd:element ref="comment" minOccurs="0"/>
      <xsd:element name="shipDt" type="xsd:date" minOccurs="0"/>
      <xsd:attribute name="partNum" type="Sku"/>
    </xsd:complexType>
  </xsd:element>
</xsd:complexType>
```


XML Topics : XML Schema, Example (4)

```
<xsd:simpleType name="Sku" base="xsd:string">  
  <xsd:pattern value="/d{3}-[A-Z]{2}"/>  
</xsd:simpleType>  
  
</xsd:schema>
```

XML Topics : UML and XML

- ❑ Unified Modeling Language (UML) is a notation standard for expressing requirements and models of objects, systems, and processes
- ❑ UML tools support the ability to parse UML objects into primitives
- ❑ Using UML primitives, it is possible to map UML objects to XML structures
- ❑ Mappings was made by W3C and OMG
- ❑ The ability to map UML to XML allows UML to be used as the modeling standard for object and interface definition. These standards can then be used to automatically generate XML Schema

XML Topics : XLink (The Extensible Linking Language)

- ❑ Any element can be a link
- ❑ Links can be bi-directional
- ❑ Links can even be multi-directional
- ❑ Links can be separated from the documents they connect

```
<footnote xlink:type="simple"  
xlink:href="footnote7.xml">7</footnote>
```

XML Topics : XPath

- ❑ Providing a syntax for locating specific parts (e.g., attribute values) of an XML document
- ❑ Used in XPointer and XSLT
- ❑ Basis for XML Query Language

descendant::language[position()=2]
/child::spec/child::body/child::*/child::language[2]
/spec/body/*/language[2]

XML Topics : XInclude

- ❑ A means of merging multiple XML document or parts thereof
- ❑ Not yet finished : **XML Inclusions (XInclude) Version 1.0**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE book SYSTEM "book.dtd" > <book
xmlns:xinclude="http://www.w3.org/2001/XInclude">
<title>The Java Developer's Resource</title>
<last_modified>December 3, 2000</last_modified>
<xinclude:include href="getting_started.xml"/>
<xinclude:include href="procedural_java.xml"/>
</book>
```

XML Topics : XPointer

- ❑ A syntax for addressing into an XML document
- ❑ Extend XPath to support non-well-formed points and ranges
- ❑ Used by XLink and XInclude

xpointer(id("ebnf"))

xpointer(descendant::language[position()=2])
ebnf

xpointer(/child::spec/child::body/child::*/child::language[2])

xpointer(/spec/body/*/language[2]) /1/14/2

xpointer(id("ebnf"))xpointer(id("EBNF"))

XML Topics : Several APIs

- ❑ SAX
- ❑ DOM
- ❑ JDOM
- ❑ Parser specific APIs

SAX

- ❑ Simple API for XML
- ❑ Public domain, developed on xml-dev mailing list
- ❑ org.xml.sax package
- ❑ Parser independent; programs can plug in different parsers
- ❑ <http://www.saxproject.org/>
- ❑ Event based; the parser pushes data to your handler
- ❑ Read-only
- ❑ SAX omits DTD declarations

SAX2

- ❑ Adds:
 - ❑ Namespace support
 - ❑ Optional Validation
 - ❑ Optional Lexical events for comments, CDATA sections, entity references
- ❑ A lot more configurable
- ❑ Deprecates a lot of SAX1
- ❑ Adapter classes convert between SAX2 and SAX1 parsers.

DOM(Document Object Model)

- ❑ Defines how XML and HTML documents are represented as objects in programs
- ❑ W3C Standard
- ❑ Defined in IDL; thus language independent
- ❑ HTML as well as XML
- ❑ Writing as well as reading
- ❑ More complete than SAX or JDOM; covers everything except internal and external DTD subsets
- ❑ DOM focuses more on the document; SAX focuses more on the parser.

JDOM

- ❑ More Java-like tree-based API
- ❑ Parser independent classes sit on top of parsers and other APIs



Use of XML for Network and Service Management

Use of XML for Network and Service Management

- ❑ XML's use in system and network management tools promises to more richly integrate management data from disparate sources
- ❑ XML increases flexibility in creating links between management applications and managed objects, and enables management applications from different vendors to interoperate
- ❑ XML provides a method of representing and sharing information between applications, network services, and network elements

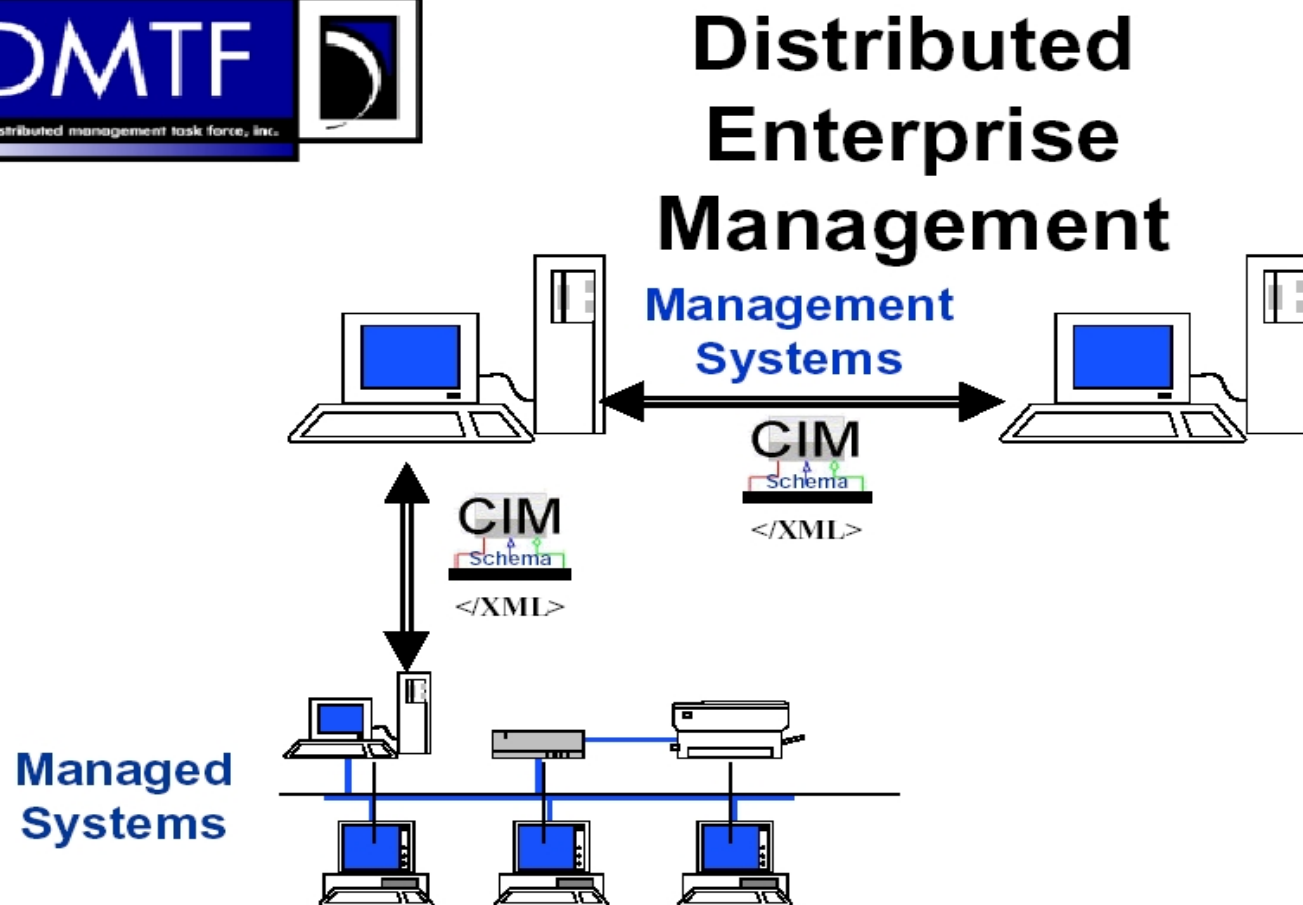
Related Activities to XML

- ❑ DMTF (Distributed Management Task Force)
- ❑ T1M1
- ❑ TMF (TeleManagement Forum)
- ❑ IETF

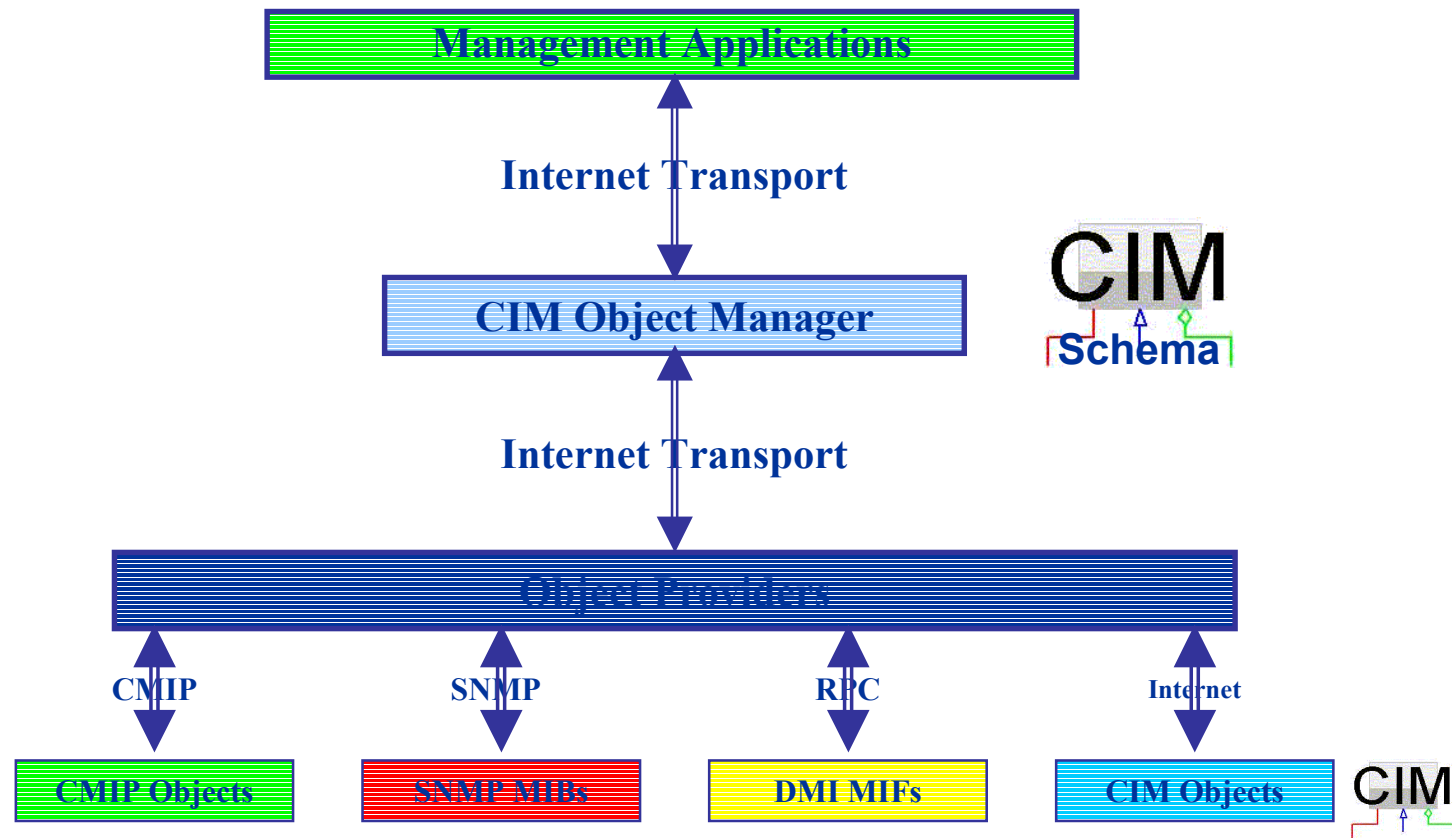
Related Activities : DMTF

- ❑ A Common Model for Management Data
 - ❑ The DMTF developed the Common Information Model (CIM) in 1996 to provide a common way to share management information enterprise wide
 - ❑ The CIM specification is the language and methodology for describing management data
 - ❑ The CIM schema provides models for various implementations to describe management data in a standard format
 - ❑ The CIM is being widely adopted

Related Activities : DMTF (2)



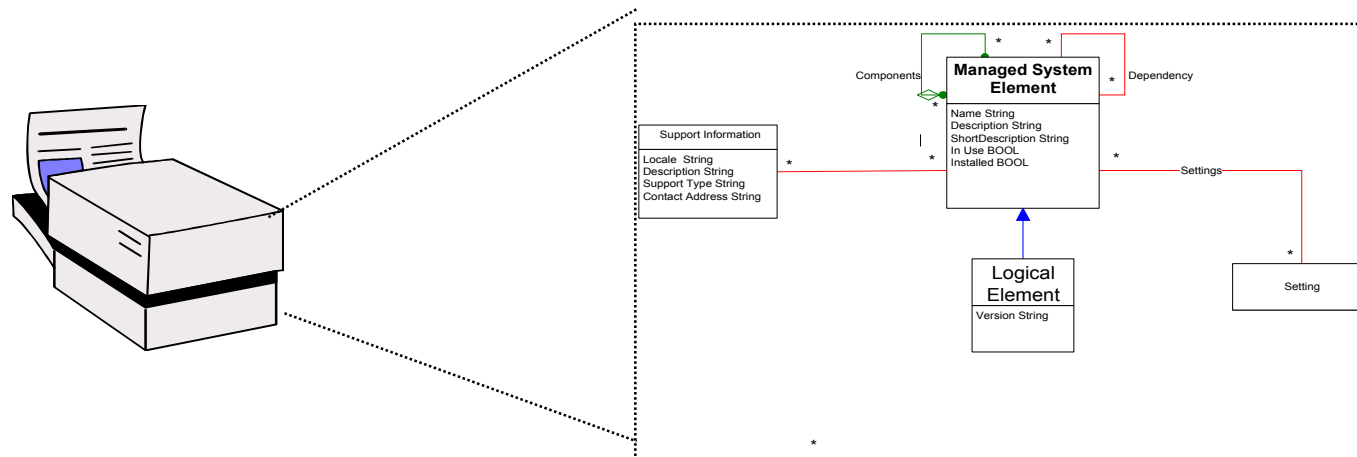
CIM Based Management System



DMI : Distributed Management Interface
MIF : Management Information Format

Information Model of DMTF

- CIM(Common Information Model)
 - The Common Information Model (CIM) defines the schemas used to represent the real-world objects being managed.
 - CIM uses an object-oriented paradigm, where manageable objects are modeled using the concepts of classes and instances.



Information Model of DMTF

- CIM is an implementation neutral schema for describing overall management information
 - CIM facilitates the common understanding of management data across different management systems
 - CIM facilitates the integration of management information from different sources
 - CIM today is a data model not an implementation
 - MOF syntax supports sharing information across management systems
 - CIM provides models for both instrumentation and management

Information Model of DMTF: CIM Management Schema

- ❑ Core model
an information model that captures notions that are applicable to all areas of management.
- ❑ Common model
an information model that captures notions that are common to particular management areas, but independent of a particular technology or implementation.
- ❑ Extension schemas
represent technology-specific extensions of the Common model.

Industry Effort for CIM (1)

- ❑ Microsoft's WMI (Windows Mgmt Instrumentation) using CDM (Common Diagnostics Model) based on CIM
- ❑ Sun's WBEM Services
- ❑ Cisco's CiscoWorks2000 and LDAP mapping in CNS
- ❑ Coordinated development with IETF
 - ❑ Policy Framework
 - ❑ IPsec Policy (used for the specification of device-level configuration policy for Internet Key Exchange protocols)
- ❑ The Open Group
 - ❑ Pegasus (Manageability Services Broker) open source
 - ❑ CIM Object Manager

Industry Effort for CIM (2)

- ❑ **SNIA (Storage Networking Industry Association)**
 - ❑ Fibre Channel and Tape / Storage Library modeling and media management
 - ❑ Disk Resource Management - CIM prototype
 - ❑ CIM Object Manager open source

http://www.snia.org/English/Resources/Code/CIM_OM.html
- ❑ **Lutris**
 - ❑ Major contributor/initial developer of SNIA open source
- ❑ **(Newly formed) WBEMSource (open source)**
initiative (to promote the widespread usage of the DMTF's management technologies)
- ❑ **Applications and instrumentation for the various "object managers"**

Industry Effort for CIM (3)

- DEN(Directory-Enabled Network)
 - Initial proposal (DEN 1.0 specification) by Microsoft and Cisco in September 1997
 - Submitted to DMTF in September 1998
 - DMTF merged DEN and CIM development

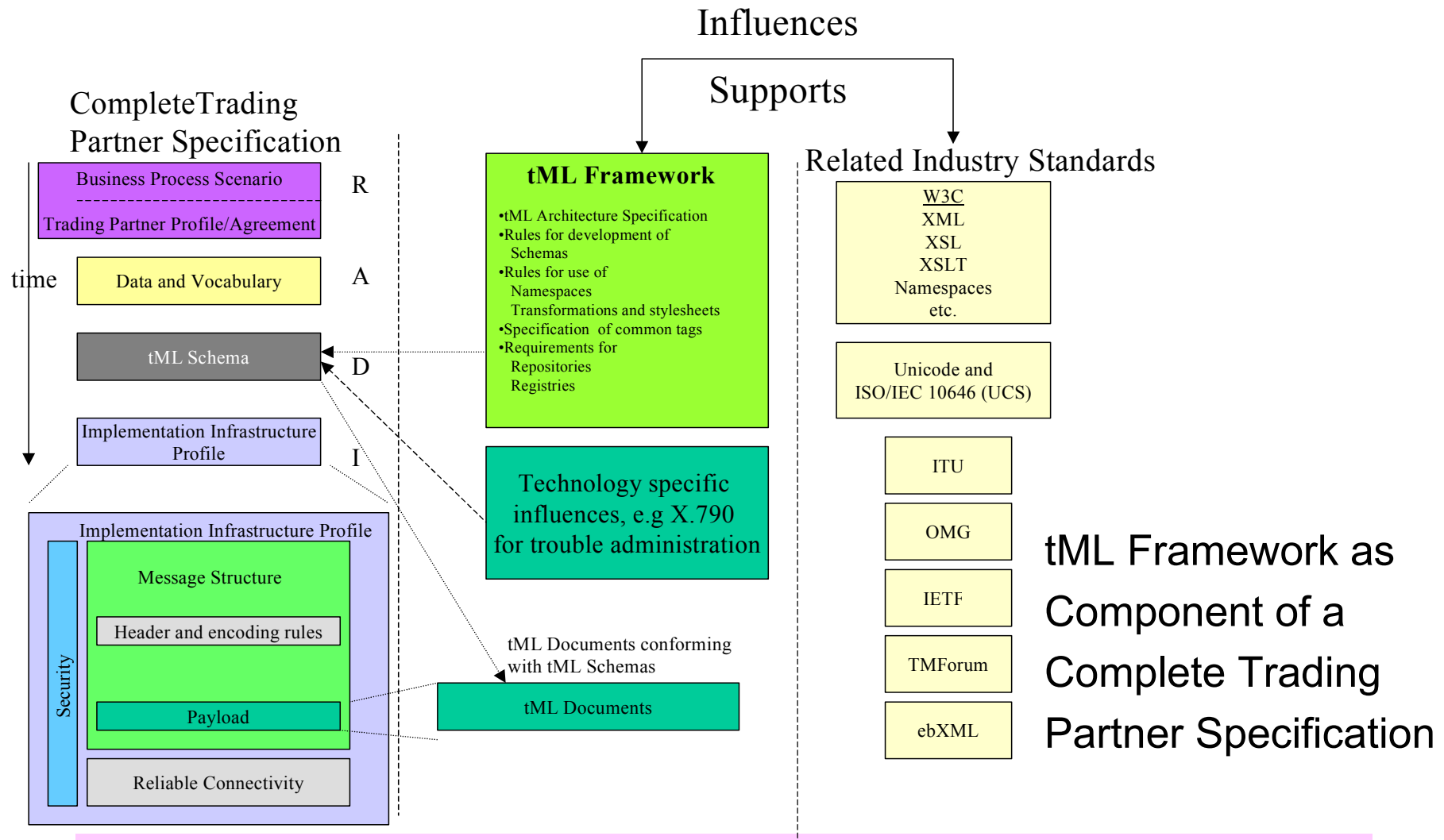
Related Activities : T1M1

- ❑ Development of tML(Telecommunication Markup Language)
- ❑ The purpose of the tML Framework
 - ❑ Provides a standard definition for the development of interoperable interfaces based on the use of the XML, within the TMN context.
- ❑ The goal for use of the Framework
 - ❑ guides the development of tML schemas and vocabularies
 - ❑ provides a common method for the definition of tML data to be exchanged and to provide a mapping to existing standards to promote re-use whenever possible.

Related Activities : T1M1

- ❑ Scope of tML Framework
 - ❑ Provides a standard definition for the development of interoperable interface based on the use of XML, within TMN context
 - ❑ Specification of common tML tags, namespaces and URIs
 - ❑ A set of rules and objectives to be applied in developing standardized schemas based on existing standard models
 - ❑ Mapping procedures from existing TMN ASN.1 syntax to tML schemas and vice versa

Related Activities: T1M1



Related Activities: T1M1

Primary use of XML	Applied to TMN
Application-to-application integration within an enterprise	Q interface (e.g., application-to-application; OS-NE; OS-OS)
Business-to-business applications	X interface (e.g., CLEC/ILEC transaction processing)
User interfaces	User interfaces in the TMN
Metadata/metaprocess descriptions	Descriptions of models and methods
Document publishing	rules for drafting standards documents

Related Activities :TMF

- Consideration of XML with CORBA and J2EE for implementation of NGOSS
 - Component method call using XML protocol such as XML-RPC, SOAP, etc.
 - Contract store in XML format

Related Activities : IETF

- ❑ XML BoF and Application protocol BoF
- ❑ OMML: OPES Meta-data Markup Language, [draft-maciocco-opes-omml-00.txt](#)
- ❑ SOAP for Open Pluggable Edge Services (OPES)
- ❑ Extensible Provisioning Protocol (EPP) version 1.0, an XML text protocol that permits multiple service providers to perform object provisioning operations using a shared central object repository, [draft-ietf-provreg-epp-04.txt](#)
- ❑ Transmission of SOAP 1.1 envelopes using a BEEP profile, [draft-etal-beep-soap-03.txt](#)
- ❑ The eXtensible Directory Access Protocol (XDAP), [draft-newton-xdap-01.txt](#)
- ❑ The Device Message Protocol (DMP), for conveying information pertaining to control, query and event subscription and notification functionality to networked appliances, [draft-khurana-dmp-appliances-00.txt](#)

Related Activities : Others

- ❑ Eurescom/TMF : Carrier Pre-selection (CPS)
Ordering between European operators
- ❑ WBEM : xml/CIM
- ❑ UN/CEFACT (jointly with OASIS) :
working on ebXML initiative
- ❑ ITU-T—Study Group 7

Information Model for Interworking between DSL SPs

- ❑ Without information model, we can not exchange consistency data between SPs and recover the network fault instantly
- ❑ A mapping should be defined between information model and an XML vocabulary for representing management data
- ❑ Interaction of user information for work order between ISPs and NSPs
- ❑ Defining XML Schema

Usage Examples of XML Schema

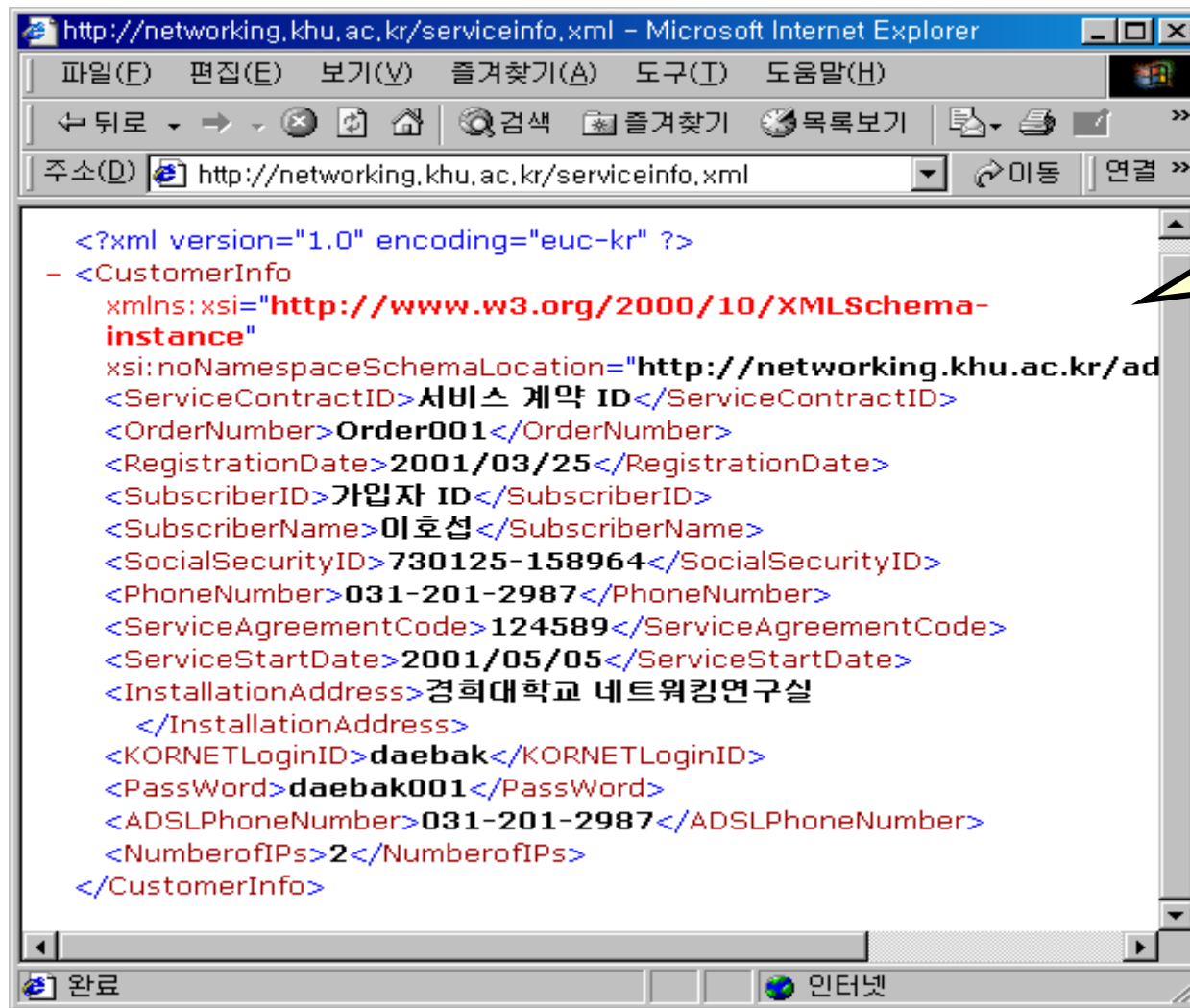
Namespace1 - ADSLCommon

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema
  xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
  <xsd:element name="SubscriberID" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element name="SubscriberName"
    type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="SocialSecurityID"
    type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="PhoneNumber" type="xsd:string"
    minOccurs="1" maxOccurs="3" />
</xsd:schema>
```

Namespace2 - ADSLSchema

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
  xmlns:Common="http://networking.khu.ac.kr/adslCommon.xsd">
- <xsd:element name="CustomerInfo">
- <xsd:complexType>
- <xsd:sequence>
  <xsd:element name="ServiceContractID" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element name="OrderNumber" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element name="RegistrationDate" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element ref="Common:SubscriberID" minOccurs="1"
    maxOccurs="1" />
  <xsd:element ref="Common:SubscriberName" minOccurs="1"
    maxOccurs="1" />
  <xsd:element ref="Common:SocialSecurityID" minOccurs="1"
    maxOccurs="1" />
  <xsd:element ref="Common:PhoneNumber" minOccurs="1"
    maxOccurs="3" />
  <xsd:element name="ServiceAgreementCode"
    type="xsd:string" minOccurs="1" maxOccurs="1" />
  <xsd:element name="ServiceStartDate" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element name="InstallationAddress" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element name="KORNETLoginID" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element name="PassWord" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element name="ADSLPhoneNumber" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
  <xsd:element name="NumberOfIPs" type="xsd:string"
    minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```


Usage Examples of XML Schema (cont'd)



Instance
document
: Customer
information

Usage Examples of XML Schema (cont'd)

```
<?xml version="1.0" ?>
- <xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
+ <xsl:template match="/">
- <xsl:template match="CustomerInfo">
  - <head>
    <title>Customer Information</title>
  </head>
  - <body bgcolor="blue">
    <h1 align="center">Customer Information</h1>
    - <table border="1" align="center" bgcolor="yellow">
      - <xsl:for-each select="ServiceContractID">
        - <tr>
          <td align="center">ServiceContractID</td>
          + <td>
            </td>
          </tr>
        </xsl:for-each>
        + <xsl:for-each select="OrderNumber">
          + <xsl:for-each select="RegistrationDate">
          + <xsl:for-each select="SubscriberID">
          + <xsl:for-each select="SubscriberName">
          + <xsl:for-each select="SocialSecurityID">
          + <xsl:for-each select="PhoneNumber">
          + <xsl:for-each select="ServiceAgreementCode">
          + <xsl:for-each select="Service-StartDate">
          + <xsl:for-each select="InstallationAddress">
          + <xsl:for-each select="KORNETLoginID">
          + <xsl:for-each select="PassWord">
          + <xsl:for-each select="ADSLPhoneNumber">
          + <xsl:for-each select="NumberofIPs">
        </table>
```

XSLT Document

ServiceContractID	서비스 계약 ID
OrderNumber	Order001
RegistrationDate	2001/03/25
SubscriberID	가입자 ID
SubscriberName	이호섭
SocialSecurityID	730125-158964
PhoneNumber	031-201-2987
ServiceAgreementCode	124589
Service-StartDate	2001/05/05
InstallationAddress	경희대학교 네트워킹연구실
KORNETLoginID	daebak
PassWord	daebak001
ADSLPhoneNumber	031-201-2987
NumberofIPs	2

Benefits of XML for representing Management information

- ❑ Rendering and Transformations
 - ❑ XSL can be used to transform XML documents to other representation
 - ❑ Map the XML into a MOF or into HTML
- ❑ Heterogeneous Application
 - ❑ XML is particularly suitable as a data representation mechanism for use in heterogeneous environments
 - ❑ An ideal standard medium for transferring data between heterogeneous platforms

Benefits of XML for representing Management information (2)

- XML as a Source of Management Data
 - Can also make a useful source of management data
- Validation
 - The “validation” of an XML document is carried out by checking that the XML document conforms to a definition given by a Document Type Definition(DTD) and XML schema

Mapping Managed Data (CIM) to XML

- Two fundamentally different models for mapping CIM into XML(DMTF)
 - *A Schema Mapping* is one in which the XML schema is used to describe the CIM classes, and CIM Instances are mapped to valid XML documents for that schema
 - CIM classes → XML element names
 - *A Metaschema Mapping* is one in which the XML schema is used to describe the CIM metaschema, and both CIM classes and instances are valid XML documents for that schema.
 - CIM classes → XML attributes or XML element values

Mapping Managed Data (CIM) to XML (cont'd)

The left window, titled "Cim_ManagedSystemElement.mof - 메모장", displays the following MOF class definition:

```
[Abstract]
class CIM_ManagedSystemElement
{
    [MaxLen(64)] string Caption;
    string Description;
    [MappingString{"MIF.DMTF|CompentID|001.5"}]
    datetime InstallDate;
    string Name;
    [Values{"OK","Error","Degraded","Unknown"}] string Status;
}
```

The right window, titled "D:\Documents and Settings\Administrator\바탕 화면\cim_ManagedSystemElement.x...", displays the corresponding XML representation:

```
<?xml version="1.0" ?>
<!DOCTYPE CIM (View Source for full doctype...)>
- <CIM CIMVERSION="2.0" DTDVERSION="1.0">
- <CLASS>
  <CLASSPATH>
    - <NAMESPACEPATH>
      <HOST>WBEM_TECRA_2</HOST>
    - <NAMESPACE>
      <NAMESPACENODE>ROOT</NAMESPACENODE>
    - <NAMESPACE>
      <NAMESPACENODE>CIMV2</NAMESPACENODE>
    </NAMESPACE>
  </NAMESPACEPATH>
  <CLASSNAME>CIM_ManagedSystemElement</CLASSNAME>
</CLASSPATH>
+ <QUALIFIER NAME="Abstract" LOCAL="true" TYPE="boolean"
  OVERRIDABLE="EnableOverride" TOSUBCLASS="Restricted"
  TRANSLATABLE="false" PROPAGATED="false"
  TOINSTANCE="false">
+ <PROPERTY NAME="Caption"
  CLASSORIGIN="CIM_ManagedSystemElement" LOCAL="true"
  TYPE="string" PROPAGATED="false">
+ <PROPERTY NAME="Description"
  CLASSORIGIN="CIM_ManagedSystemElement" LOCAL="true"
  TYPE="string" PROPAGATED="false">
+ <PROPERTY NAME="InstallDate"
  CLASSORIGIN="CIM_ManagedSystemElement" LOCAL="true"
  TYPE="datetime" PROPAGATED="false">
+ <PROPERTY NAME="Name"
  CLASSORIGIN="CIM_ManagedSystemElement" LOCAL="true"
  TYPE="string" PROPAGATED="false">
+ <PROPERTY NAME="Status"
  CLASSORIGIN="CIM_ManagedSystemElement" LOCAL="true"
  TYPE="string" PROPAGATED="false">
</CLASS>
</CIM>
```

Mapping Managed Data(MIB) to XML

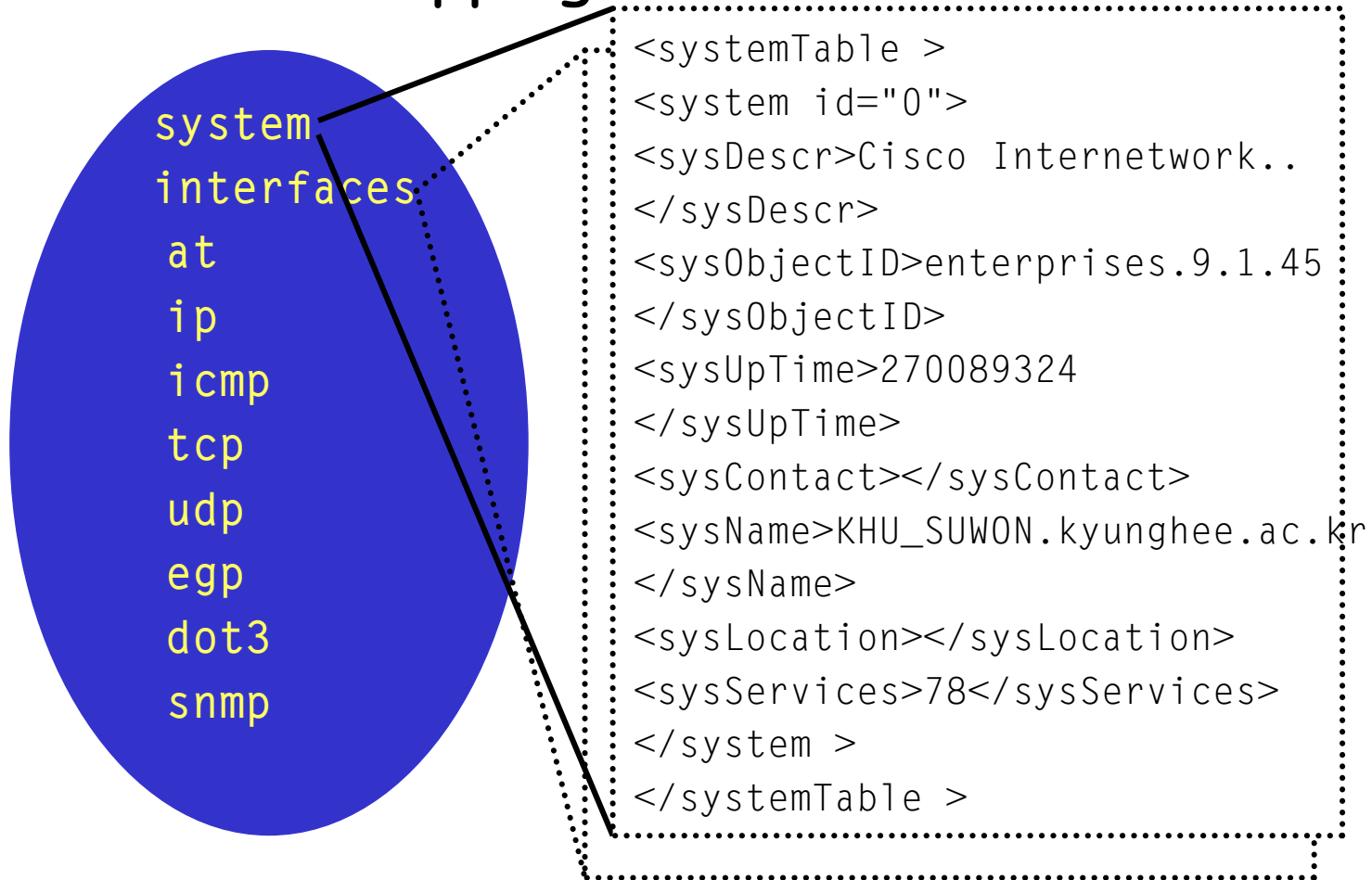
- Two fundamentally different models for mapping MIB into XML(J.P martin-Flatin)
 - **Model-level mapping (Schema mapping) :**
The XML Schema is used to describe the MIB elements, and MIB elements are mapped to valid XML Documents for that schema.
 - **Metamodel-level mapping (Metaschema mapping) :**
The XML schema is used to describe the XML metaschema

```
<system>  
<sysDescr type = "DisplayString">  
Cisco xx</sysDescr>  
</system>
```

```
<mibs name = "system">  
<attrib name = "sysDescr"  
type="DisplayString">Cisco xx  
</attrib></mibs>
```

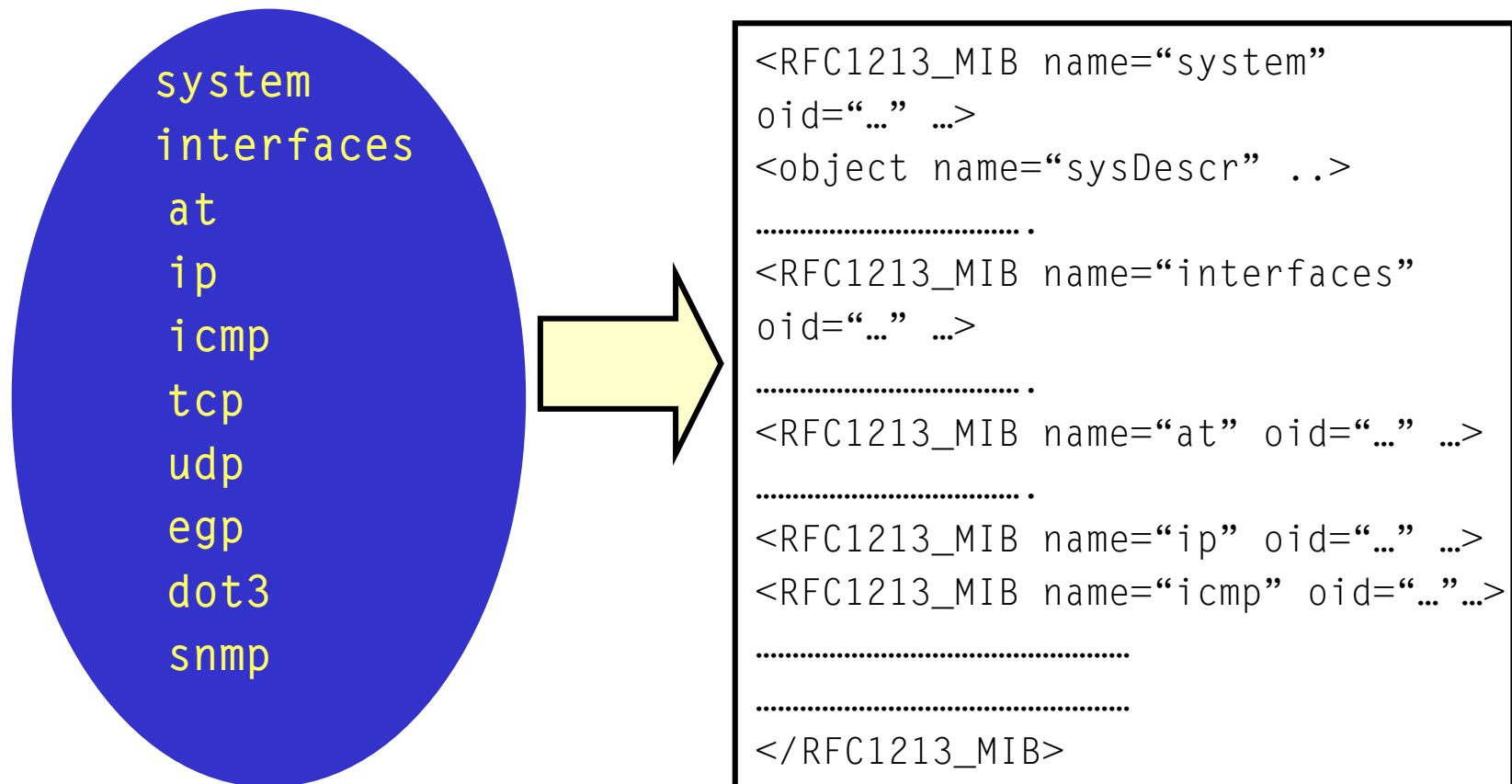
Mapping Managed Data (MIB) to XML(cont'd)

□ Model-level mapping

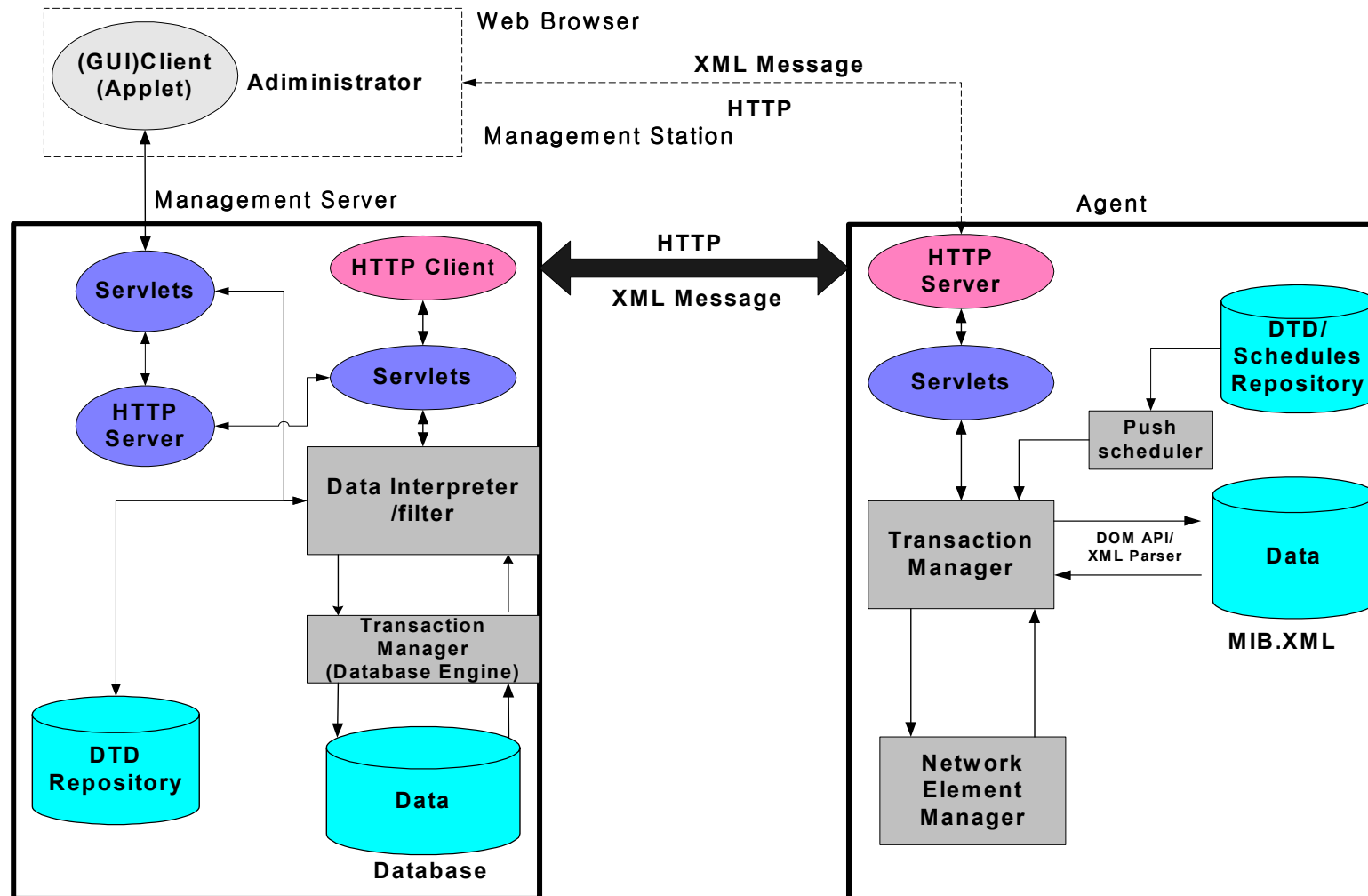


Mapping Managed Data (MIB) to XML(cont'd)

□ Metamodel-level mapping



A Web-based Network Management Architecture Using XML in MIB



A Web-based Network Management Architecture Using XML in MIB (cont'd)

- ❑ It provides flexible and dynamic way to control the MIB.
- ❑ The communication between agents and manager is accomplished by interaction between Java applets and servlets using HTTP/TCP.
- ❑ It provides a web interface to manager for adding, deleting, and monitoring variables to the agent.
- ❑ The MIB data as XML message is transmitted to manager.
- ❑ It uses the push model and the publish-subscription paradigm.
- ❑ It consists of three parts :
 - ❑ The management station
 - ❑ The management server
 - ❑ The agent

A Web-based Network Management Architecture Using XML in MIB (cont'd)

- ❑ Management station
 - ❑ It is the desktop of the administrator.
 - ❑ It can be a machine which runs Web browser supporting Java
- ❑ Agent configuration applet
 - ❑ The applet communicates directly with the agent.
 - ❑ It provides a web interface for manager to send commands for adding, deleting, and monitoring variables to the agent
- ❑ Data analysis applet
 - ❑ Analyzes the network traffic.
 - ❑ Retrieves some values stored in a database by a day, a weekly, a month, a year, etc.

A Web-based Network Management Architecture Using XML in MIB (cont'd)

- Management Server
 - It runs two servlets : the pushed data subscription and the data analyzing manager.
 - The database server could be separated in a different machine
- The servlet for pushed data subscription
 - The servlet connects to the agent at startup time.
 - This servlet communicates with the data filter/interpreter
- The servlet for data analysis manager
 - This servlet connects to another servlet and receives the results that are analyzed by the data interpreter.
 - Administrator can retrieve some values in specific period and analyze this data

A Web-based Network Management Architecture Using XML in MIB (cont'd)

- ❑ Transaction Manager
 - ❑ It communicates with data filter/interpreter.
 - ❑ It can store pushed data, search and retrieve a specific data in a database
 - ❑ It provides functionality to convert XML message to relational database format
- ❑ Agent
 - ❑ The agent runs a lightweight Java Virtual Machine and one servlet : the XML publisher servlet
 - ❑ The XML publisher servlet
 - ❑ This servlet communicates with several managers.
 - ❑ During the subscription phase, the push scheduler locally stores the subscription information sent by the applet for configuration of agent and data subscription applet.
 - ❑ DTD Repository

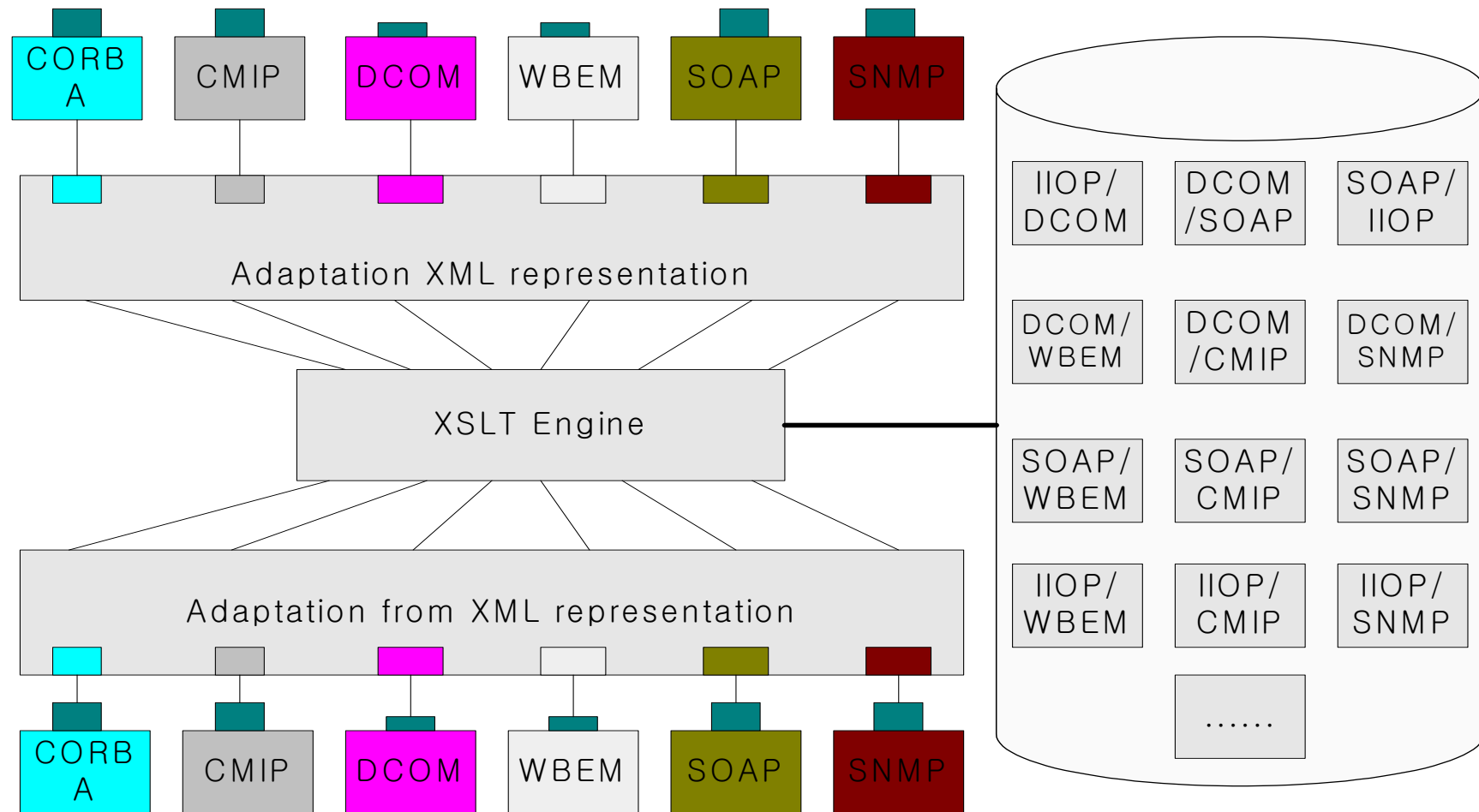


XML Platforms and DPE

XML Based Technology Interoperability

- ❑ Defining a single XML profile for each communication technology
- ❑ Adapters can be developed using XSLT docs and generic transform engines
 - ❑ Can readily prototype new transforms
 - ❑ Can develop dynamically loadable adapters in DPE
- ❑ SOAP, WSDL, UDDI, Web Services

XML Based Technology Interoperability (2)



XML and CORBA Integration

- ❑ Brings together XML and IDL objects
- ❑ Read XML documents and create IDL Valuetypes
- ❑ Save IDL Valuetypes into an XML document
- ❑ Manipulate XML via CORBA IDL
- ❑ Generate IDL Valuetypes from an XML DTD
 - ❑ Static approach
 - ❑ Schemas not yet supported
- ❑ Flexible handling of new XML documents
 - ❑ Dynamic approach
 - ❑ Handles any document

XML and CORBA Integration (cont'd)

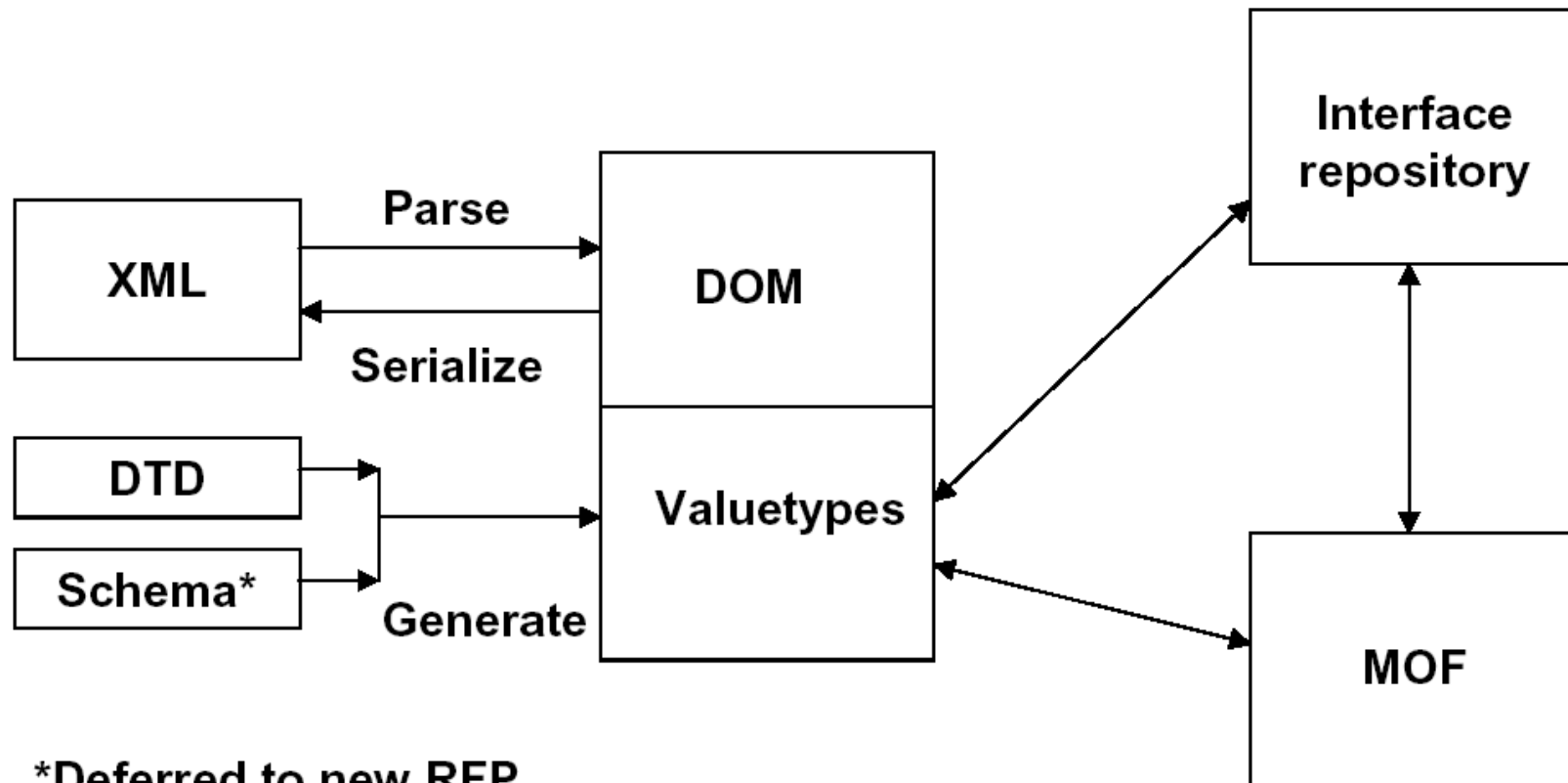
- ❑ Dynamic approach
 - ❑ DOM API is a W3C Standard for parsing XML documents, with or without DTD or Schema
 - ❑ Tree of nodes produced from XML document
 - ❑ Valuetype for each node
 - ❑ Concrete generated node valuetypes
 - ❑ Generic Element node valuetype when a generated valuetype is not present

XML and CORBA Integration (cont'd)

- ❑ Static approach
 - ❑ Generated valuetypes from DTDs and Schemas.
 - ❑ Efficiency to minimize transferred information.
 - ❑ Used when you have knowledge of types.
 - ❑ Valuetypes are integrated with the DOM valuetypes.
 - ❑ Generated valuetypes include accessors - efficient and easy to use.
 - ❑ Bridges the worlds of compiled and adapted.

XML and CORBA Integration (cont'd)

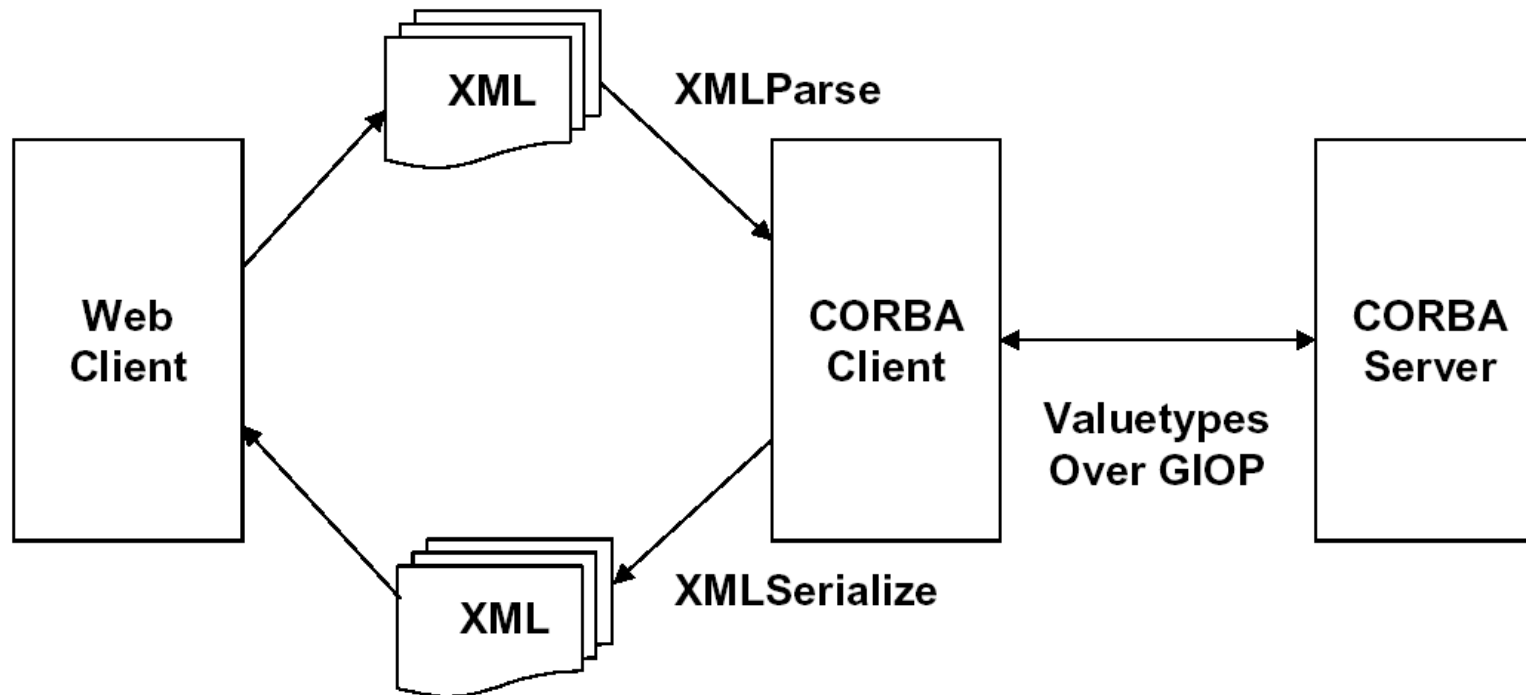
- Architecture - dynamic and static



*Deferred to new RFP

XML and CORBA Integration (cont'd)

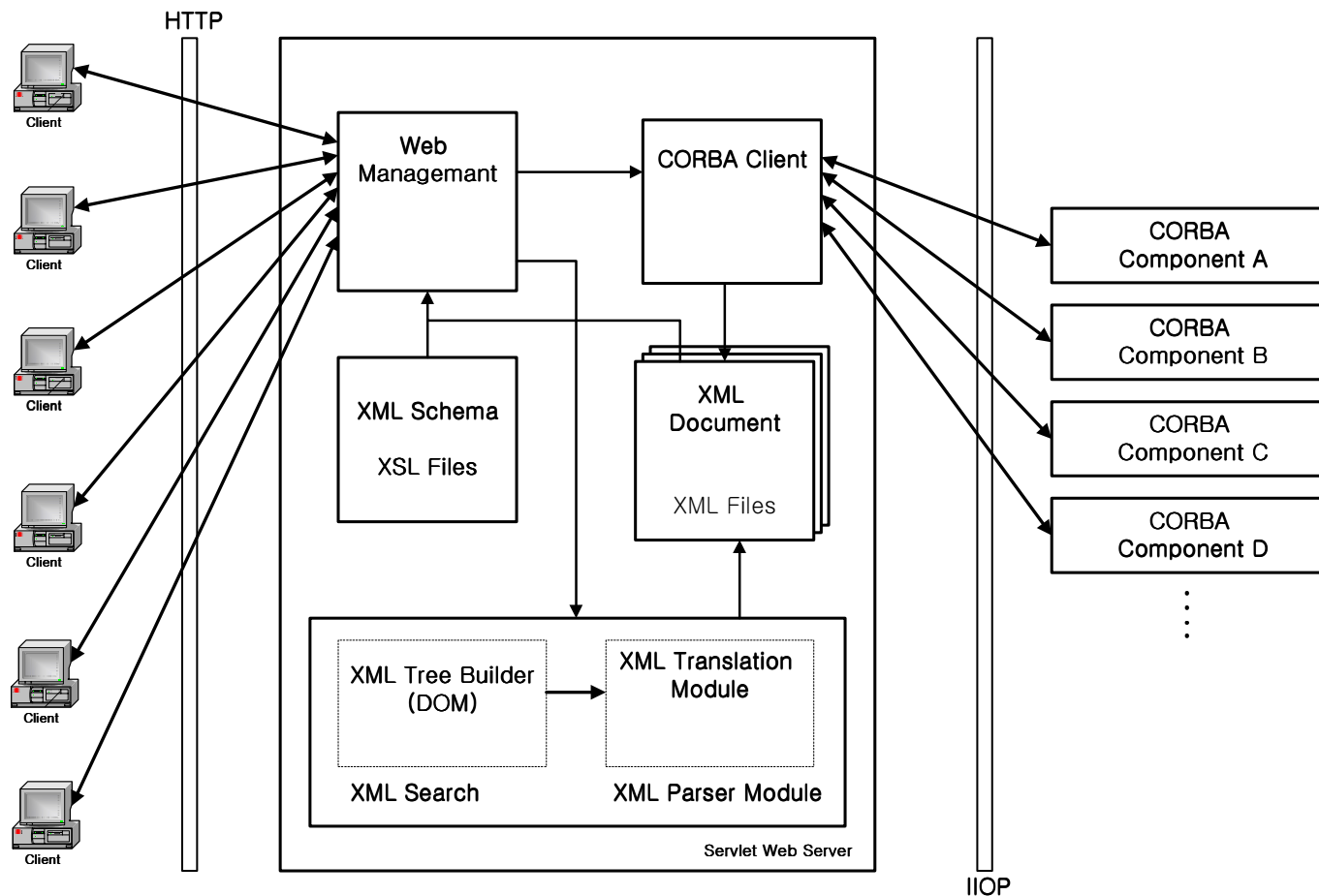
□ Example - document processing



- Static mappings if DTD available
- Dynamic mappings if no DTD, or new elements found

XML and CORBA Integration (cont'd)

□ An Example of Implementation



SOAP(Simple Object Access Protocol)

- ❑ XML-based messaging framework specifically designed for exchanging formatted data across the internet
- ❑ Simple, easy to use, and completely neutral with respect to OS, programming language, or distributed computing platform
- ❑ Made by Ariba, Commerce One, Compaq, DevelopMentor, Hewlett Packard Company, IBM, IONA, Lotus, Microsoft, SAP AG, UserLand Software Inc.
- ❑ Now SOAP 1.2 by W3C

SOAP (cont'd)

- ❑ Lightweight XML protocol for exchanging structured and typed information over the web
 - ❑ Messages are expressed in XML
 - ❑ Message elements are described using XML schema
- ❑ Simple
 - ❑ A minimum of functionality
 - ❑ Reuse of existing technologies
- ❑ Extensible
 - ❑ Does not include application or transport semantics

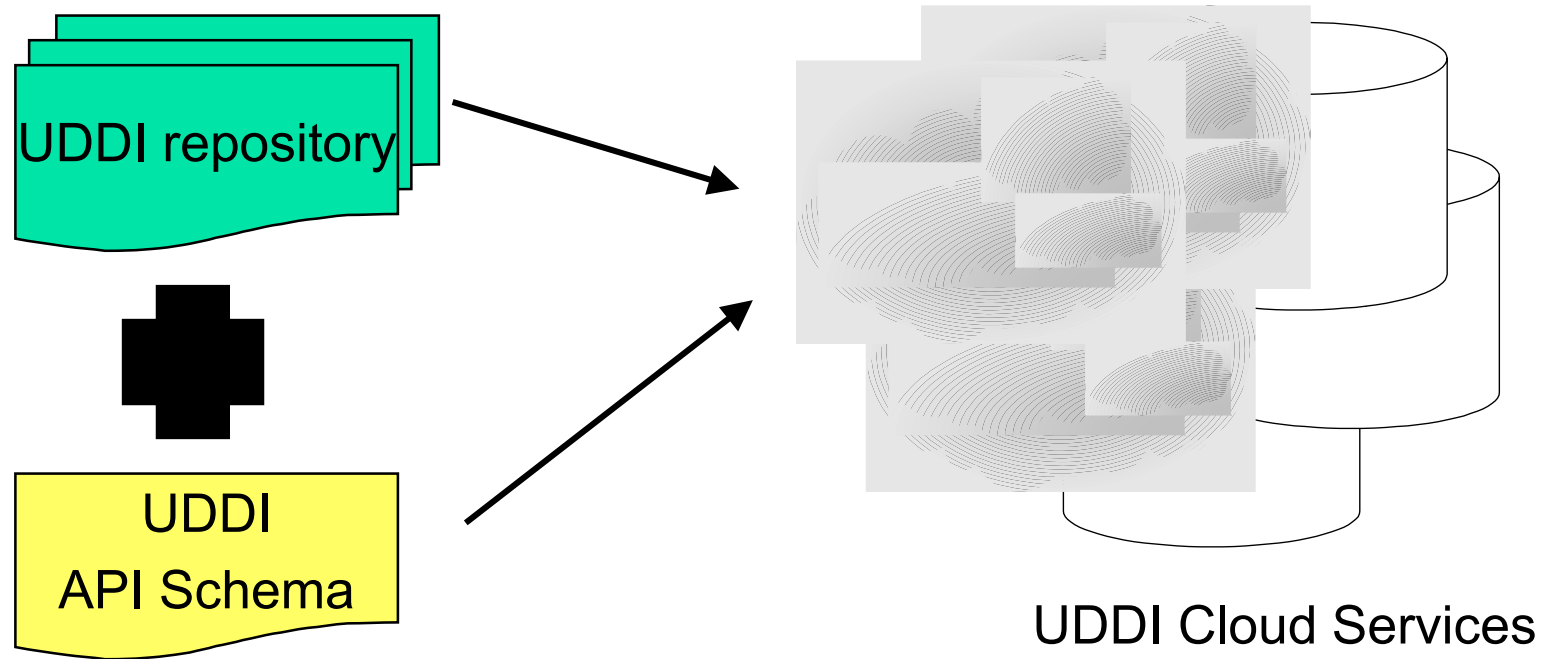
SOAP(cont'd)

- ❑ Speaking SOAP, means exchange of SOAP messages in some fashion
- ❑ SOAP messages are encapsulated in a mandatory envelope which describes
 - ❑ What features and services are represented in the message
 - ❑ Who should deal with them (destination)
 - ❑ Whether they are mandatory or optional
- ❑ RPC-like, with calls and responses

Universal Description, Discovery & Integration(UDDI)

- ❑ Defines a way to publish and discover information about Web services
- ❑ Contains search and contact information for business
- ❑ Consists of an XML schema for SOAP messages, and a description of the UDDI API specification
- ❑ Includes pointers to WSDL and other web service descriptions

Universal Description, Discovery & Integration (UDDI) (cont'd)



- ❑ The UDDI specifications and schema are used to build discovery Services on the internet
- ❑ These discovery services provide a consistent publishing interface and allow programmatic discovery of services

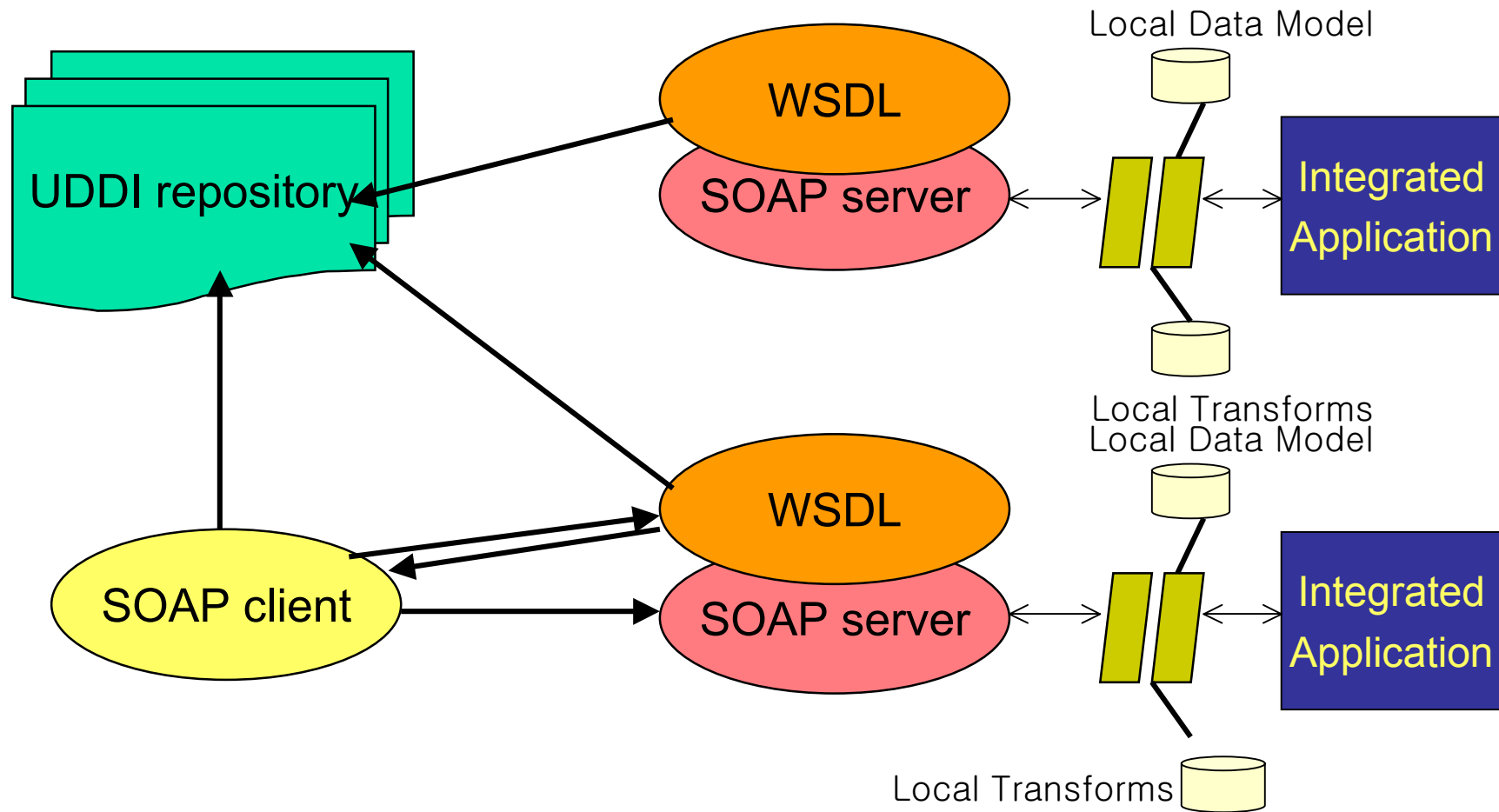
Web Services Description Language (WSDL)

- ❑ A particular form of an XML Schema,
 - ❑ by MS and IBM for the purpose of defining the XML message, operation, and protocol mapping of a web service accessed using SOAP or other XML protocol
- ❑ Defines
 - ❑ Message types : Defines types and complex types using XML for use in a WSDL description
 - ❑ Port types (abstract interfaces)
 - ❑ Collection of operations
 - ❑ Ports
 - ❑ Address for binding

Web Services Description Language(WSDL) (cont'd)

- ❑ WSDL defines Web services in terms of "endpoints" that operate on XML message
- ❑ The current WSDL spec describes how to map messages and operations to SOAP1.1, HTTP GET/POST, and MIME
- ❑ WDSL creates web services definitions by mapping a group of endpoints into a logical sequence of operations on XML messages

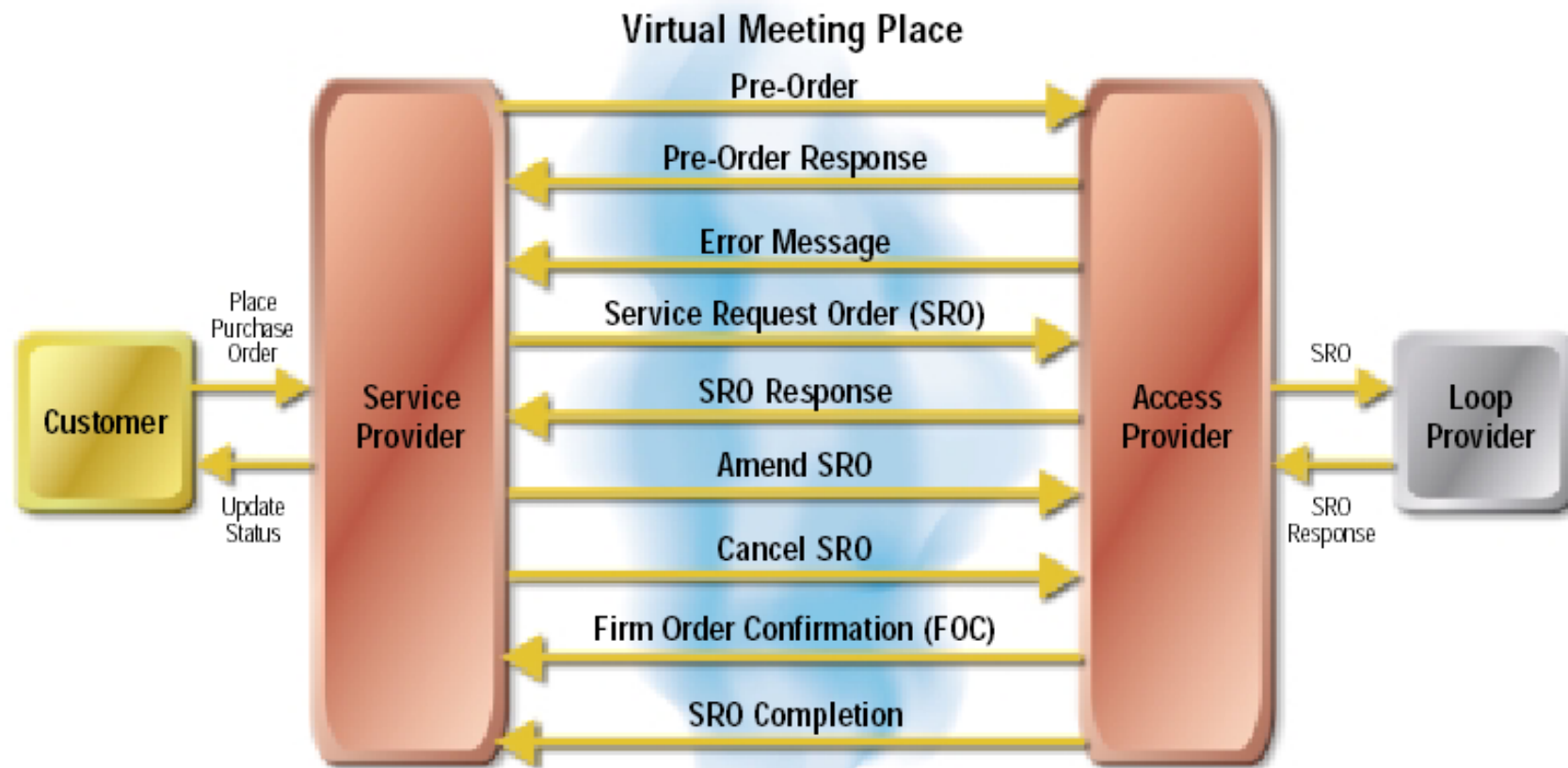
Web Services Discovery Architecture



Example : Catalyst Project

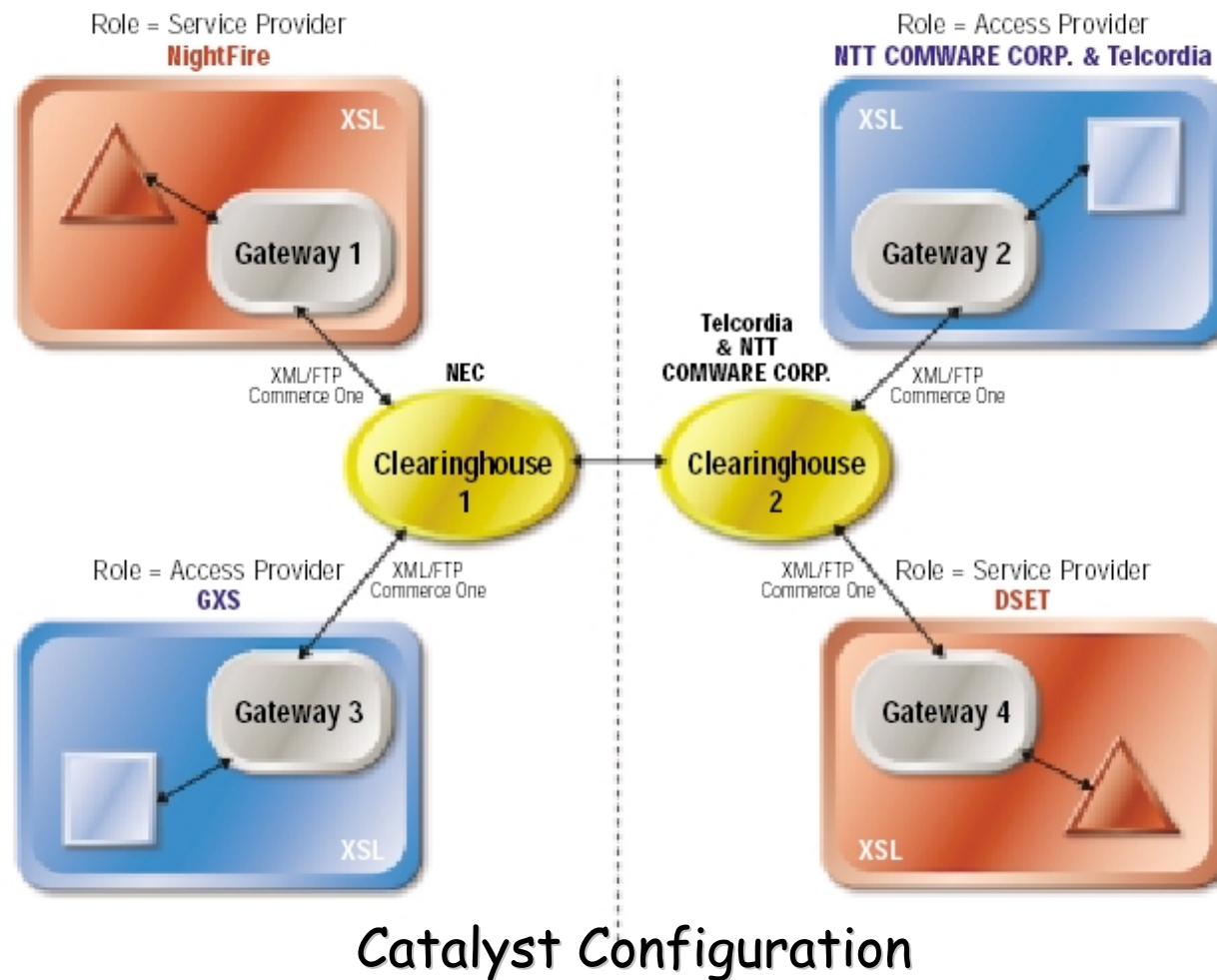
- ❑ Provides SPs with a business model for processing Digital Subscriber Line(DSL) orders based upon the concept of a *Virtual Meeting Place*
- ❑ VMP enables providers to improve supply chain management by converging available technology and e-business applications through the services of an interconnection clearinghouse

Example : Catalyst Project (2)



DSL Order Flows

Example : Catalyst Project (3)



References

CIM <http://www.dmtf.org/spec/cims.html>

IETF Policy Framework WG <http://www.ietf.org/html.charters/policy-charter.html>

UML <http://www.rational.com/uml/resources/documentation/index.jtmpl>

Solaris WBEM APIs

[http://docs.sun.com/ab2/coll.453.1/WBEMPROG/@Ab2TocView?Ab2Lang=C&Ab2Enc=i
so-8859-1](http://docs.sun.com/ab2/coll.453.1/WBEMPROG/@Ab2TocView?Ab2Lang=C&Ab2Enc=i
so-8859-1)

WBEM <http://www.dmtf.org/spec/wbem.html>

WMI CORE 1.5

[http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=/MSD
N-FILES/027/001/576/msdncompositedoc.xml](http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=/MSD
N-FILES/027/001/576/msdncompositedoc.xml)

XML <http://www.w3.org/XML/>

SNIA – CIM Object Manager open source

http://www.snia.org/English/Products/Products_FS.html ->

technical activities -> code

Open Group – Pegasus (CIM Manageability Services Broker)

<http://www.opengroup.org/management/pegasus/>

Open Group – WBEMSource Initiative <http://www.opengroup.org/wbemsource/>

T1M1 <http://www.t1.org/t1m1/t1m1.htm>

TMF <http://www.tmforum.org/>

XML Deitel et al, XML How to Program, Prentice Hall, 2001

Questions

