

---

# Interaction Translation Methods for XML/SNMP Gateway

**December 13, 2002**

**Yoon-Jung Oh**

bheart@postech.ac.kr

Distributed Processing and Network Management Lab.  
Graduate School for Information Technology, POSTECH

# Contents

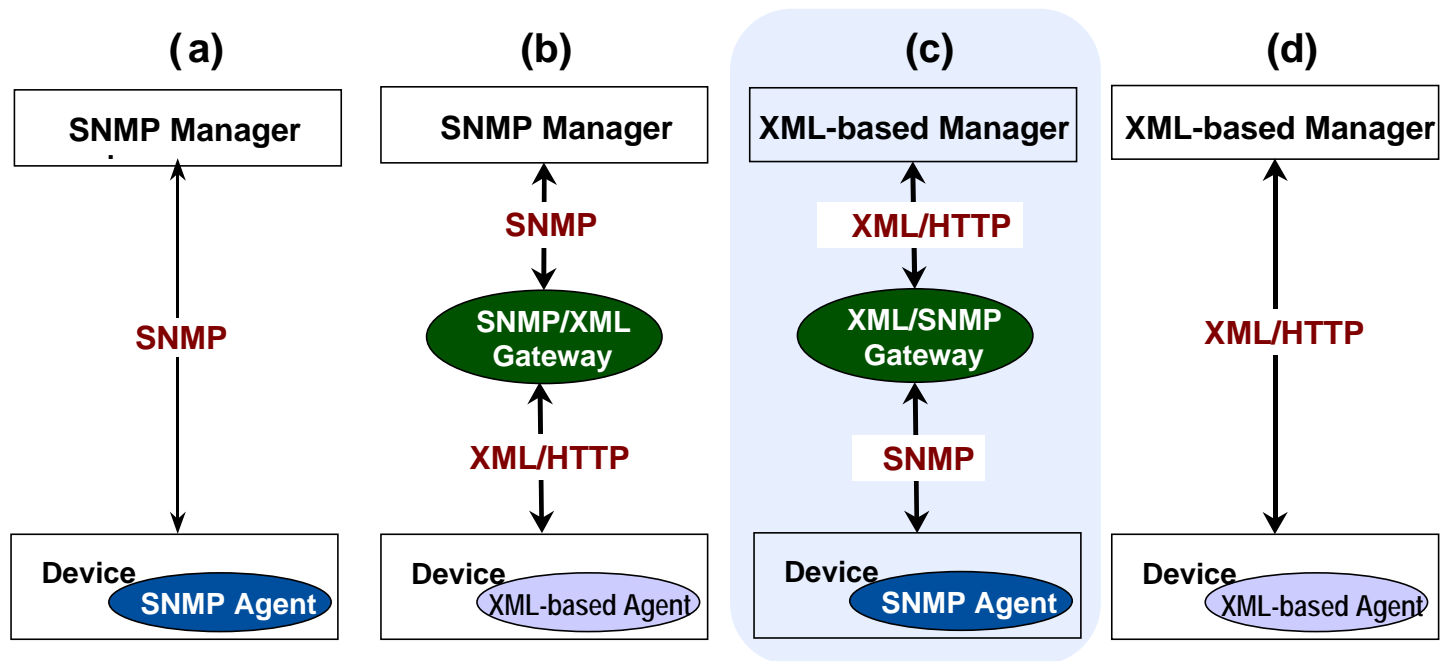
---

- ❖ Introduction
- ❖ Related Work
  - XML Technologies
  - XML-SNMP Integration
- ❖ Architecture of XML/SNMP Gateway
- ❖ Interaction Translation Methods for XML/SNMP Gateway
  1. XML Parser-based Translation
  2. HTTP-based Translation
  3. SOAP-based Translation
- ❖ Comparison of Proposed Methods
- ❖ Implementation
- ❖ Performance Analysis
- ❖ Conclusion and Future Work

# Introduction (1)

## ❖ Motivation

- Currently, **SNMP-based network management** is widely deployed
- **XML-based network management** has been proposed to complement **SNMP**
- **XML/SNMP gateway** is needed for integrated network management



# Introduction (2)

---

## ❖ Motivation (cont'd)

- XML/SNMP gateway must provide both specification translation and interaction translation between the two different management domains, XML and SNMP
- Recent research on SMI to XML Schema specification translation provides the foundation for the XML/SNMP gateway
- No concrete or standard interaction translation method for the gateway exists yet

## ❖ Propose Interaction Translation Methods for XML/SNMP Gateway

1. XML Parser-based Translation between DOM/SAX interfaces and SNMP operations
2. HTTP-based Translation applying XPath, XQuery and XUpdate
3. SOAP-based Translation based on SOAP RPC between manager and gateway

# Related Work (1)

---

## ❖ XML Technologies

- **DOM (Document Object Model)** : Platform- and language-independent interface that allows applications to dynamically access and manipulate the content and structure of documents
- **SAX (Simple API for XML)** : Event-driven and serial-access mechanism for accessing XML documents
- **XPath (XML Path Language)** : Language for addressing specific parts of an XML document
- **XQuery (XML Query Language)** : Query language for XML providing document filtering and contents grouping
- **SOAP (Simple Object Access Protocol)** : Simple and lightweight protocol for exchanging structured and typed information in a distributed environment using XML
- **WSDL (Web Services Description Language)** : XML-based language used to define Web Services and describe how to access them

# Related Work (2)

---

## ❖ XML/SNMP Integration

### – Specification Translation

- J.P. Martin-Flatin's **management information model**
- F. Strauss's libsmi, converting **SMI MIBs to XML Schema** (or DTD) definitions
- Our work on MIB to XML translation, validated by developing an **XML-based SNMP MIB browser**

### – XML/SNMP Gateway

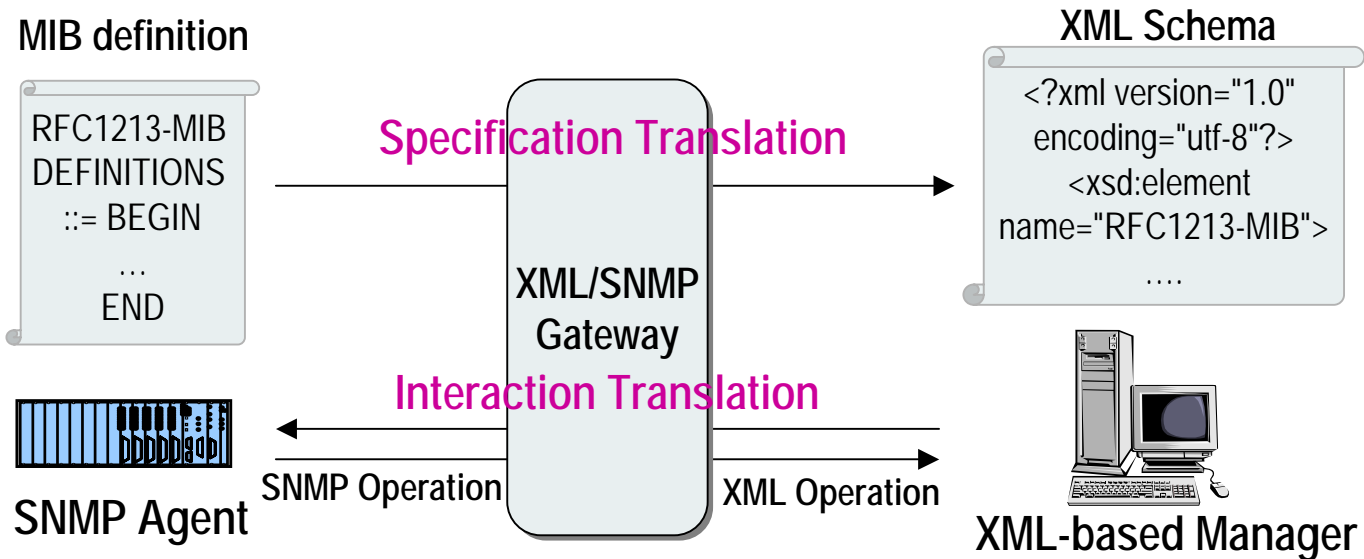
- F. Strauss's gateway, implemented using **mibdump**
- Avaya Labs research on XML-based management interface for SNMP enabled devices, currently implementing **a gateway using XML-RPC**

# XML/SNMP Gateway

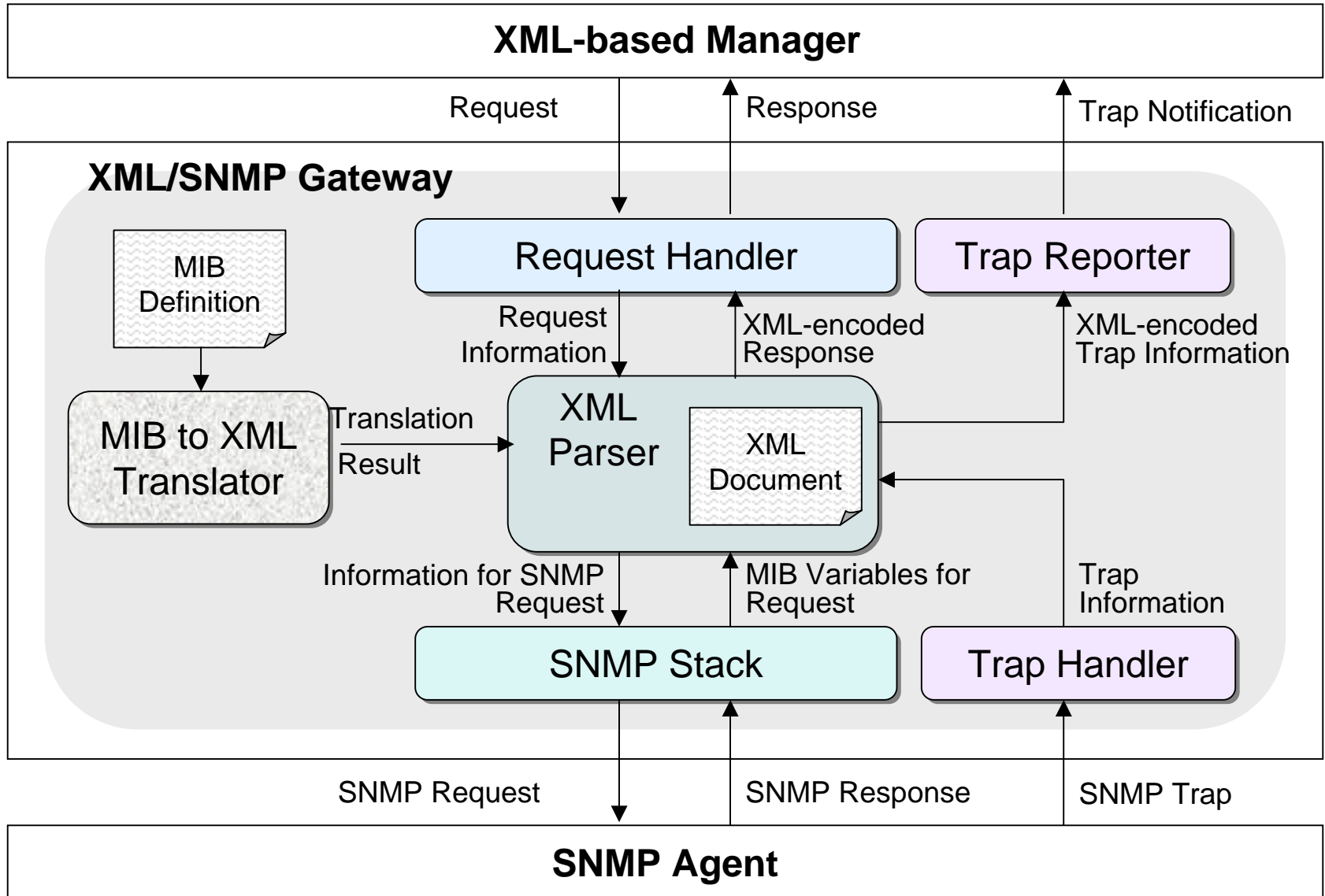
## ❖ What is XML/SNMP Gateway?

- Provides a method to manage networks equipped with SNMP agents using XML-based manager
- Relays management information and operations between XML-based manager and SNMP agent

## ❖ Basic requirements for the gateway



# Architecture of XML/SNMP Gateway





# 1. XML Parser-based Translation (1)

---

## ❖ Specification Translation Results

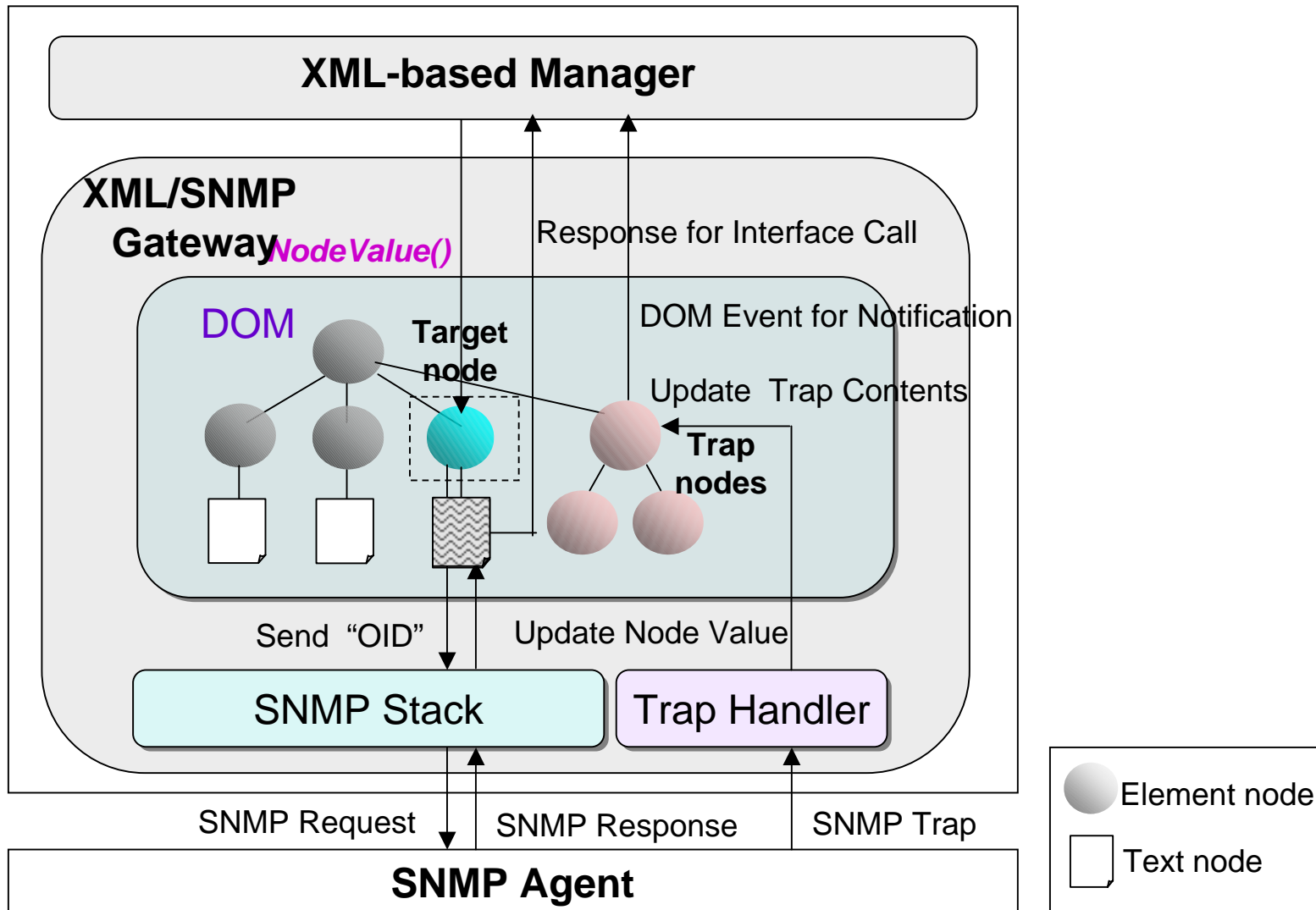
- MIB module → *Node::Document*
- MIB node → *Node::Element*
- Node value → *Node::Text*

## ❖ Example Mappings of DOM Interfaces into SNMP Operations

### – *Node::NodeValue*

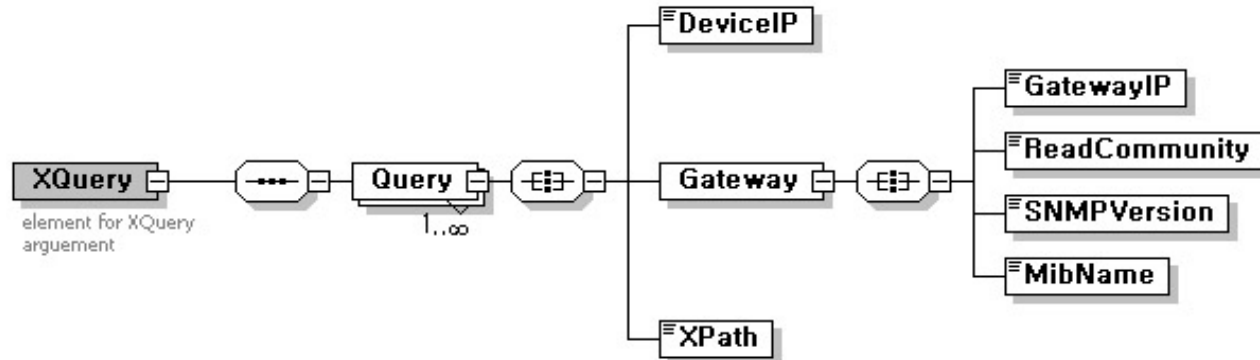
- Retrieval from the manager : translated into an **SNMP GET** request
- Modification from the manager : translated into an **SNMP SET** request
- Modification on trap nodes : translated into a notification message through event handler

# XML Parser-based Translation (2)

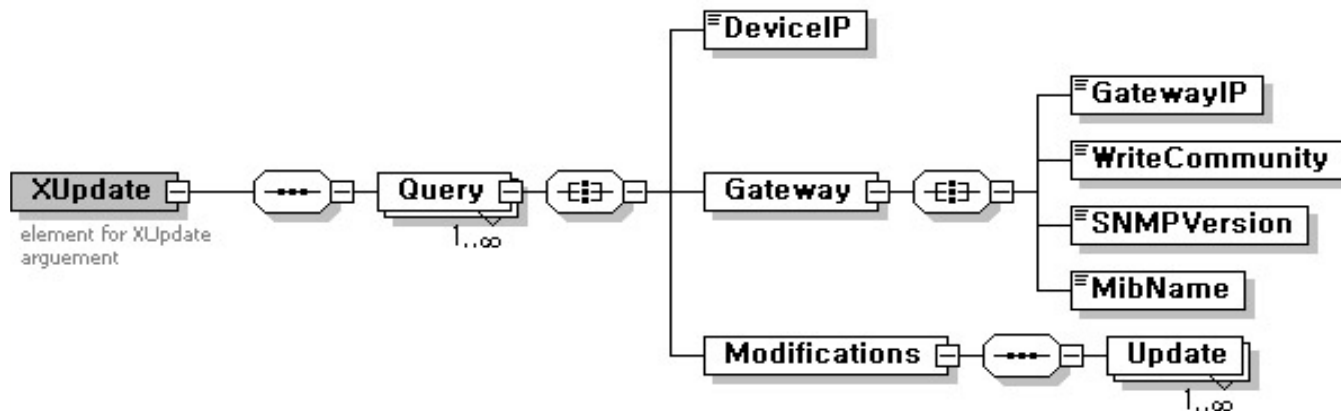


# 2. HTTP-based Translation (1)

## ❖ HTTP GET Method with *XQuery* → SNMP GET Request



## ❖ HTTP POST Method with *XUpdate* → SNMP SET Request



# 2. HTTP-based Translation (2)

## ❖ Examples of XPath, XQuery and XUpdate in HTTP

### Example of using XPath

```
http://hostname/gateway?XQuery=<XQuery>  
... <XPath>device[@type="server"]</XPath> ...<XQuery>
```

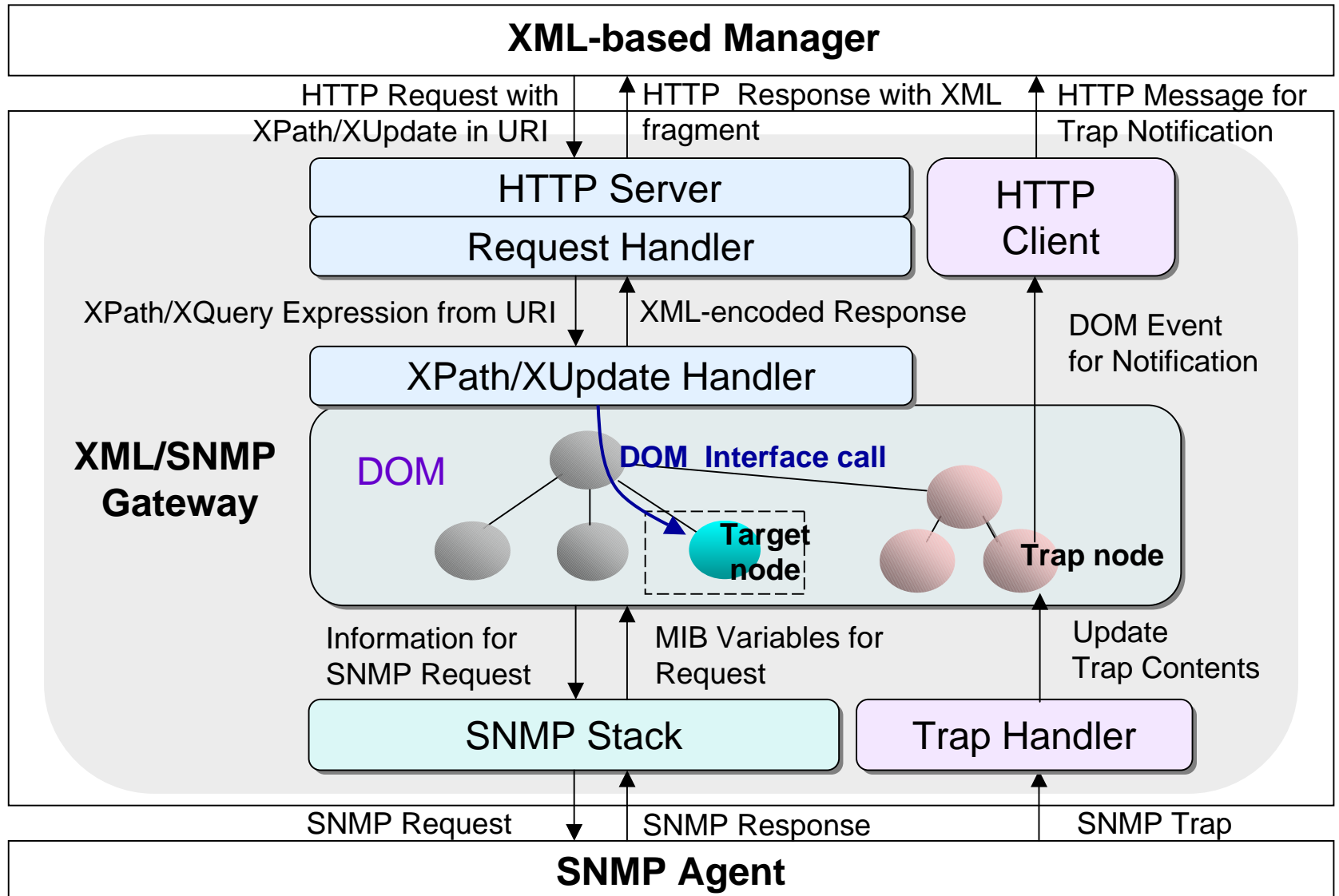
### Example of Using XQuery

```
http://hostname/gateway?XQuery=<XQuery>... <result>  
{ Let $t := input() //ifTable/ifEntry/ ifType[contains( ./text(), "6")]  
RETURN <totalInOutOctets count="{count($t) }">  
<in> { sum($t/ifInOctets/text()) } </in>  
<out> { sum($t/ifOutOctets/text()) } </out>  
</ totalInOutOctets> } </result> ... </XQuery>
```

### Example of Using XUpdate

```
... <Modifications><Update select="//sysContact">admin</Update>  
<Update>...</Update></Modifications> ...
```

# HTTP-based Translation (3)



# 3. SOAP-based Translation (1)

## ❖ SOAP Message Format

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>.....</SOAP-ENV:Header >
<SOAP-ENV:Body>
  XML for advertised methods
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

## ❖ *get* and *set* operations published by the gateway

- *get* operation with *XQuery* → SNMP **GET**
- *set* operation with *XUpdate* → SNMP **SET**

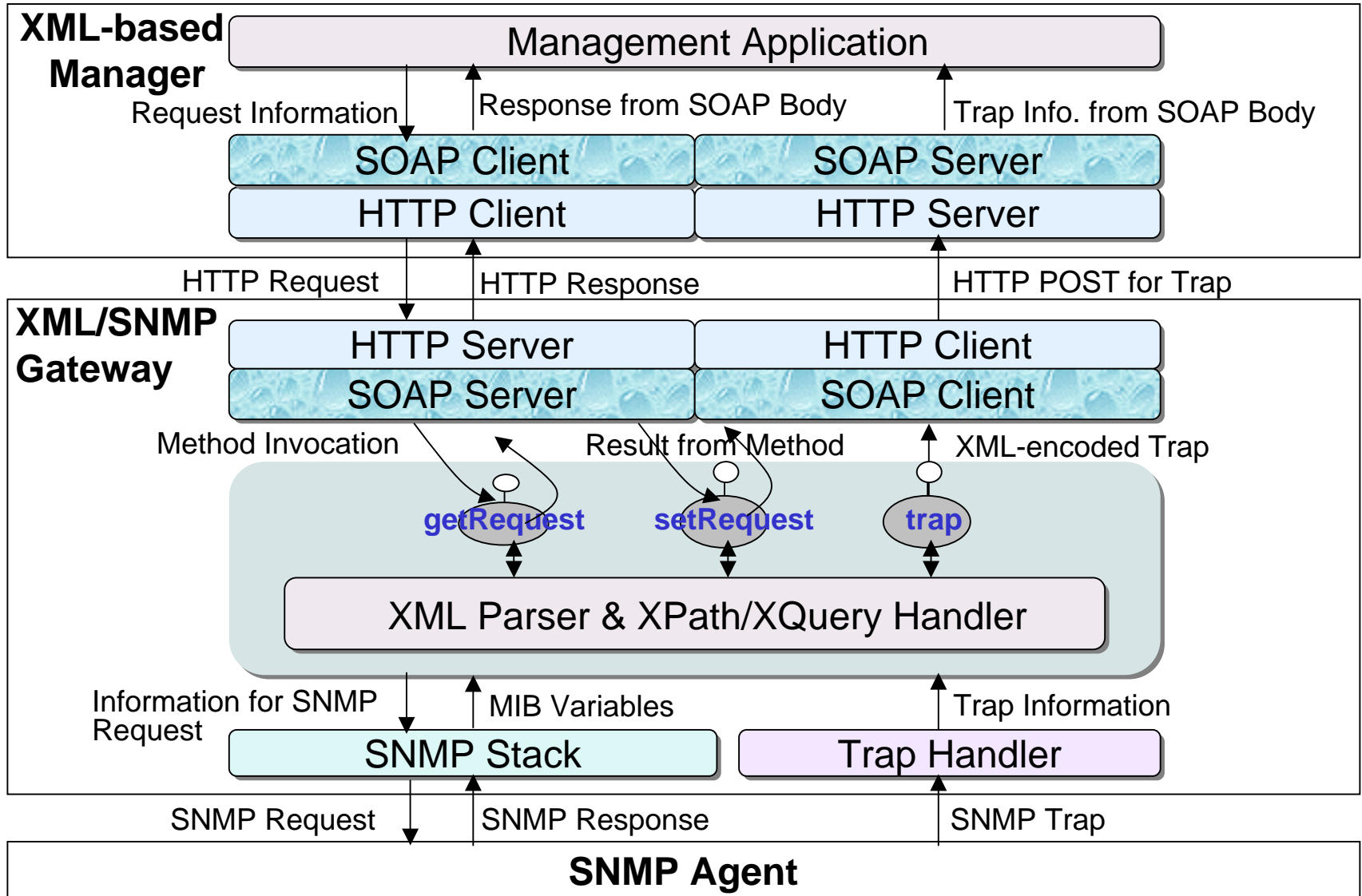
## ❖ *trap* operation published by the manager in order to receive notification message from the gateway

# 3. SOAP-based Translation (2)

## ❖ WSDL Definition of Basic Operations for SOAP RPC

```
<wsdl:definitions (namespace declarations) >
<wsdl:message name="getRequest">
<wsdl:part name="param" type="xsd:string" /></wsdl:message>
<wsdl:message name="getResponse">
<wsdl:part name="getReturn" type="xsd:string" /></wsdl:message>
<wsdl:portType name="SoapInterface">
<wsdl:operation name="get" parameterOrder="param">
  <wsdl:input message="intf:getRequest" name="getRequest" />
  <wsdl:output message="intf:getResponse" name="getResponse" />
</wsdl:operation></wsdl:portType>
<wsdl:binding name="SoapInterfaceSoapBinding" type="intf:SoapInterface">
<wsdlsoap:binding style="rpc"
  transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="get"> ... </wsdl:operation></wsdl:binding>
<wsdl:service name="SoapInterfaceService">
<wsdl:port binding="intf:SoapInterfaceSoapBinding" name="SoapInterface">
<wsdlsoap:address location="http://hostname:8080/axis/SoapInterface.jws" />
</wsdl:port></wsdl:service></wsdl:definitions>
```

# SOAP-based Translation (3)





# Analysis of Translation Methods

Translation Methods	Advantages	Disadvantages
XML Parser-based Translation	<ul style="list-style-type: none"> <li>- No need for a request handler between <b>internal gateway</b> and manager</li> <li>- Can be <b>the basis for the other translation methods</b></li> </ul>	<ul style="list-style-type: none"> <li>- <b>Imposes a burden on the manager</b> by invoking a series of interfaces for request processing</li> </ul>
HTTP-based Translation	<ul style="list-style-type: none"> <li>- Simple to implement in <b>XML/HTTP</b>.</li> <li>- Provides <b>an efficient mechanism for querying managed objects</b></li> </ul>	<ul style="list-style-type: none"> <li>- <b>No standard</b> in the use of URI string as a request specifier</li> <li>- Need of <b>XPath, XQuery and XUpdate</b> parsers</li> </ul>
SOAP-based Translation	<ul style="list-style-type: none"> <li>- Simple to implement over <b>HTTP</b>.</li> <li>- Inherits <b>advantages in HTTP-based translation</b></li> <li>- Provides <b>a standard method</b> to implement an <b>RPC</b></li> </ul>	<ul style="list-style-type: none"> <li>- Need of <b>SOAP server/client</b></li> <li>- <b>Overhead of packaging SOAP</b> messages</li> </ul>

# Implementation (1)

---

## ❖ XML/SNMP Gateway Specification

- Linux Server with Pentium-III 800 MHz CPU and 256 MB RAM

## ❖ XML Parser-based Translation

- Apache **Xerces** 1.4.4 for XML parser
- OpenNMS's **joeSNMP** 0.2.6 for SNMP Handler and Trap Handler
- Validated using a simple manager running with the gateway in the same process

## ❖ HTTP-based Translation

- Based on XML parser-based translation
- Apache **Tomcat** 4.0 for HTTP server and Innovation's HTTP Client V0.3
- Apache **Xalan** 2.4.0 for XPath processor
- HTTP GET for MIB Retrieval and POST for MIB modification
- Query messages are XML-encoded using **XPath/XQuery/XUpdate**
- Validated using our **XGEMS**(XML-based Global Element Management System)

# Implementation (2)

---

## ❖ SOAP-based Translation

- Based on XML parser-based translation
- Apache **Axis** 1.0 (SOAP 1.1 and WSDL 1.1 compliant) for SOAP engine
- Used WSDL for SOAP service definition
- Service Deployment Methods
  - Using **Java Web Service(JWS)** provided by Axis
  - Automatic generation of WSDL and proxy/skeleton codes
- Validated using our **XGEMS**

# Performance Analysis: Design (1)

## ❖ Performance Test Items

- Latency Performance of XML Parser-, HTTP- and SOAP-based method for MIB II retrieval using iterative SNMP GetNext
- Performance evaluation of variations of the proposed translation methods

## ❖ Performance Testbed



XML-based Manager

Linux server with  
Pentium-III 800 MHz CPU  
256 MB RAM

100 Mbps



XML/SNMP Gateway  
Simple XML-based Manager

Linux server with  
Pentium-III 800 MHz CPU  
256 MB RAM

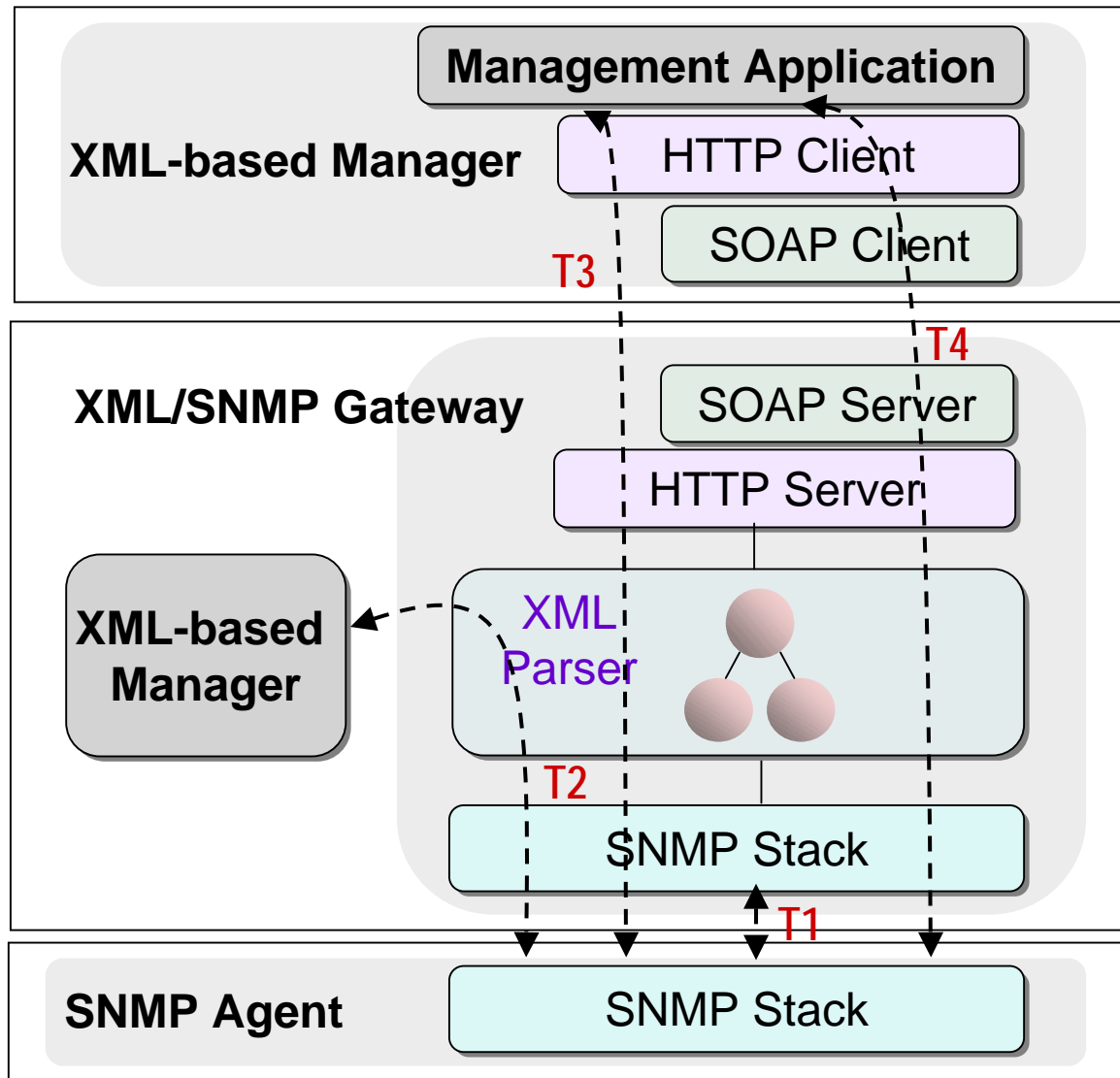
100 Mbps



SNMP Agent

1. L4 switch with 9 ports
2. L2 switch with 26 ports

# Performance Analysis: Design (2)



# Performance Analysis: Results (1)

## ❖ Latency Performance of Proposed Translation Methods

- Time for MIB-II translation and DOM loading : 535 ms
- Connection Setup Time : 0.1 ms for HTTP, 17.4 ms for SOAP

Method	Device (MIB size)	L4 switch (28 KB)	L2 switch (54 KB)
SNMP Stack: T1 (ms)		1307.1	4283.6
XML Parser-based Translation: T2 (ms)		1360.6 (4.1%)	4317.6 (0.8%)
HTTP-based Translation: T3 (ms)		1419.1 (8.6%)	4418.8 (3.2%)
SOAP-based Translation: T4 (ms)		1613.3 (23.4%)	4922.2 (14.9%)

- Each translation method adds approximately the same amount of overhead time to the basic SNMP processing time (T1)
- Additional translation overhead takes small portion of the overall processing time

# Performance Analysis: Results (2)

## ❖ Comparison on Parsing Overhead of XPath

Type	A	B	C	D
Example	<i>//ifTable</i>	<i>//ifInOctets //ifOutOctets</i>	<i>//ifType/following-sibling::ifInOctets   //ifType/following-sibling::ifOutOctets</i>	<i>//ifType[.='6']/following-sibling::ifInOctets   //ifType[.='6']/following-sibling::ifOutOctets</i>
Retrieved Data Size (KB)	12.4	2.1	2.1	1.8
Overhead of XPath Parsing (ms)	0.5	0.6	108.2	1432.2
Overall Processing Time : T2 (ms)	353.8	103.5	212.3	1682.8

Apache Xerces 1.4.4 and Xalan 2.4.0 for XPath Parser

# Performance Analysis: Results (3)

## ❖ Performance Comparison of HTTP GET and POST in HTTP-based Translation

Method	Device (MIB data)	L4 switch		L2 switch	
		<i>system</i> group	MIB-II	<i>system</i> group	MIB-II
HTTP GET (ms)		96.5	1419.1	94.9	4418.8
HTTP POST (ms)		112.2	1440.7	101.2	4623.4

## ❖ Latency of Several Variations of SOAP *get* Operation

Type	<i>get</i> operation using one method with			<i>get</i> operation using	
	1 param (type A)	2 params (type B)	6 params (type C)	One method (type D)	Two methods (type E)
L4 switch(ms)	1613.3	1676.6	1908.8	1613.3	1743.5
L2 switch(ms)	4922.2	4993.1	5298.4	4922.2	5063.3



# Conclusion

---

- ❖ Interaction Translation Methods in XML/SNMP gateway for XML-based Integrated Management
  1. XML parser-based translation provides the manager direct access to management information DOM/SAX interfaces
  2. HTTP-based translation provides efficiency improvement in XML/HTTP communication using XPath, XQuery and XUpdate
  3. SOAP-based translation provides a flexible and standardized method for interaction between the manager and gateway
  
- ❖ Validation of the Proposed Methods
  - Implemented and validated XML/SNMP gateway using XML-based Global Element Management System (XGEMS)
  - Performed several experiments for latency performance of the proposed methods
  - Validated our XML/SNMP gateway provides efficient translation in integrated network management

# Future Work

---

- Performance improvement of the translation mechanism using the **SAX parser** instead of the **DOM parser**
- **Scalability** of the gateway
- Implementation of **Web Services** using **UDDI integration** with our current implementation
- Integration with **CORBA**
- **WBEM(Web-based Enterprise Management) implementation** using CIM to XML mapping and CIM operations over HTTP

# Q & A

---









