

WEB SERVICES FOR MANAGEMENT

WHY WEB SERVICES?

WHAT ARE WEB SERVICES?

EXAMPLE

PERFORMANCE

TOOLS

CONCLUSIONS

WHY WEB SERVICES?

EVOLUTION OF SNMP FAILED

NEW TECHNOLOGIES ARE NEEDED

WEB SERVICES MAY BECOME THE MOST IMPORTANT
MIDDLEWARE TECHNOLOGY

WILL BECOME AVAILABLE ON ALL FUTURE PLATFORMS

WILL BE APPLIED FOR MANY KINDS OF APPLICATIONS

IMPLEMENTATION OF WS APPLICATIONS IS RELATIVELY SIMPLE

MANY SKILLED DEVELOPERS

MANY TOOLS

FUTURE MANAGEMENT EXPERTS
CAN CONCENTRATE ON MANAGEMENT APPLICATIONS
INSTEAD OF MANAGEMENT TECHNOLOGY

WHY WEB SERVICES?

SOME FACTS

MANY PROGRAMMING LANGUAGES HAVE WS LIBRARIES

PART OF DEVELOPMENT PLATFORMS: .NET, SUN-ONE, JBUILDER

WS SUPPORT INCLUDED IN WINDOWS / OFFICE

CALLING A WS FROM EXCEL TAKES 4 LINES OF CODE

COMPARE THIS TO SNMP!

THE KEY TO SUCCESS WILL BE EASE OF USE!

WHY WEB SERVICES?

THE HYPE

IRTF-NMRG

Network Management Research Group

OASIS

Web Services Distributed Management

OGSI

Open Grid Services Infrastructure Working Group

PARLAY GROUP

Parley-X

MANY RESEARCH GROUPS

OVERVIEW

WHY WEB SERVICES?

WHAT ARE WEB SERVICES?

EXAMPLE

PERFORMANCE

TOOLS

CONCLUSIONS

WHAT ARE WEB SERVICES?

WEB SERVICES COMPONENTS

PROTOCOL STACK

MAIN W3C SPECIFICATIONS

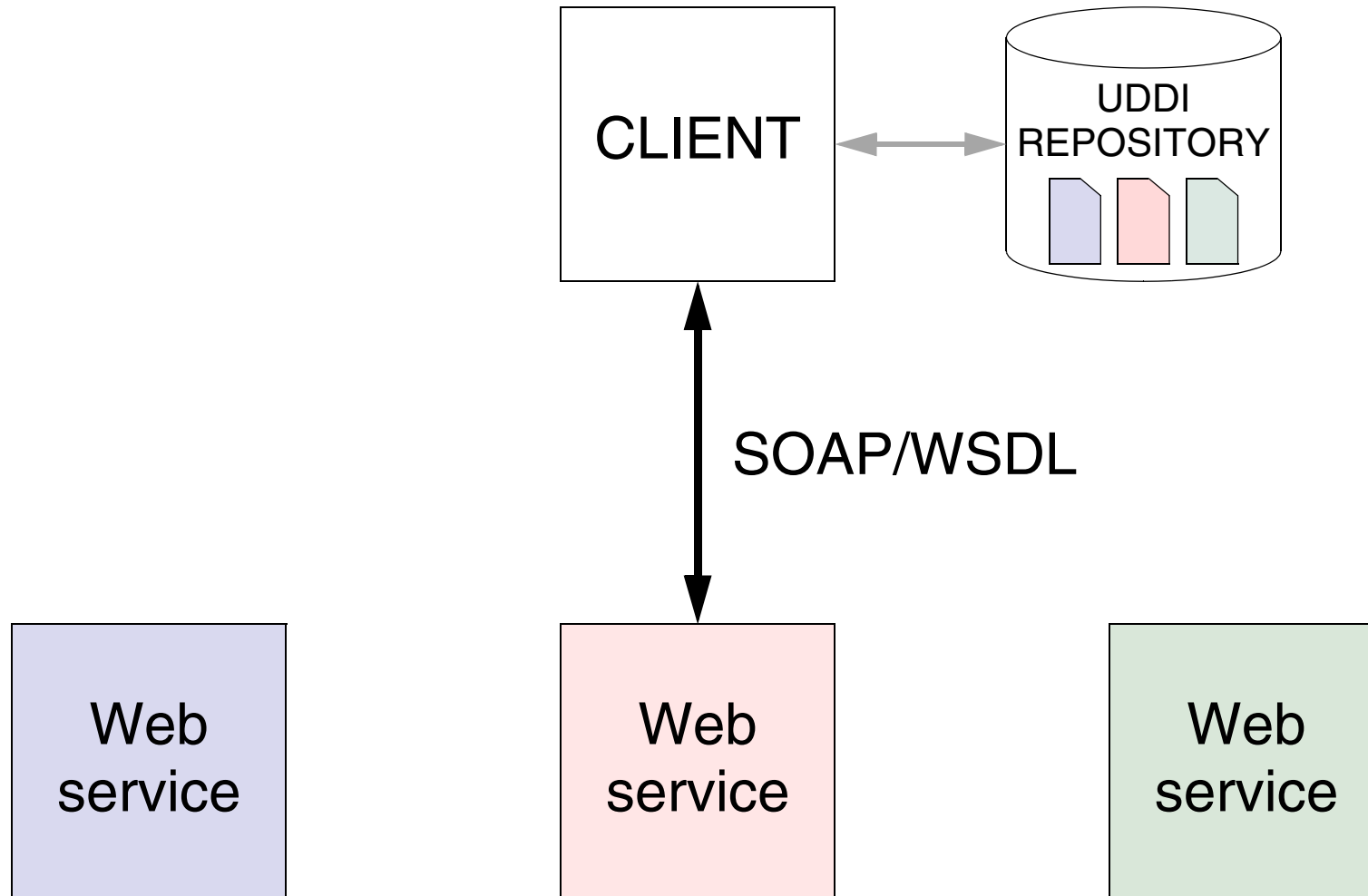
STRUCTURE WSDL DEFINITION

OPERATION STRUCTURE

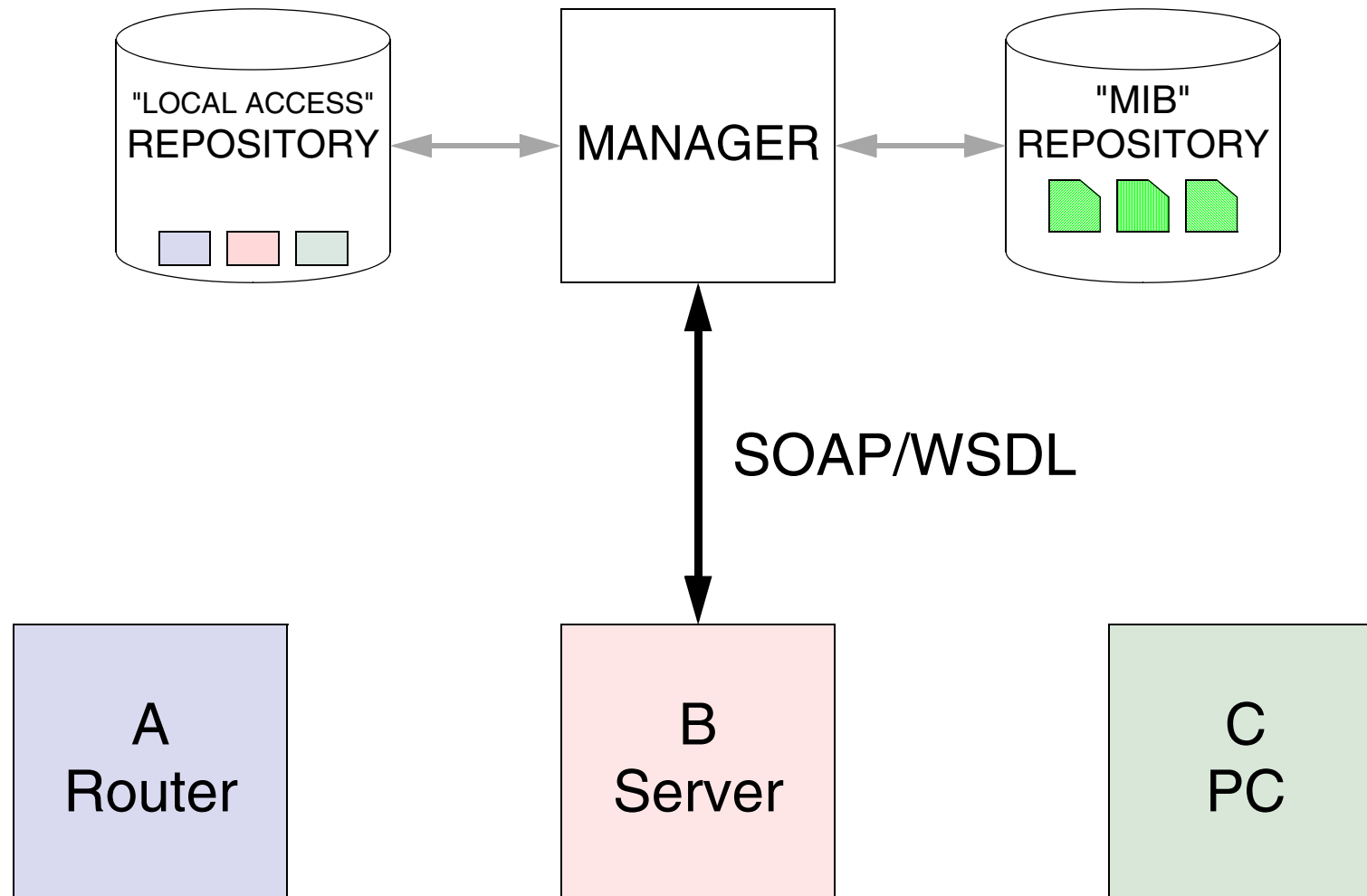
DATA TYPES

ADVANCED FEATURES

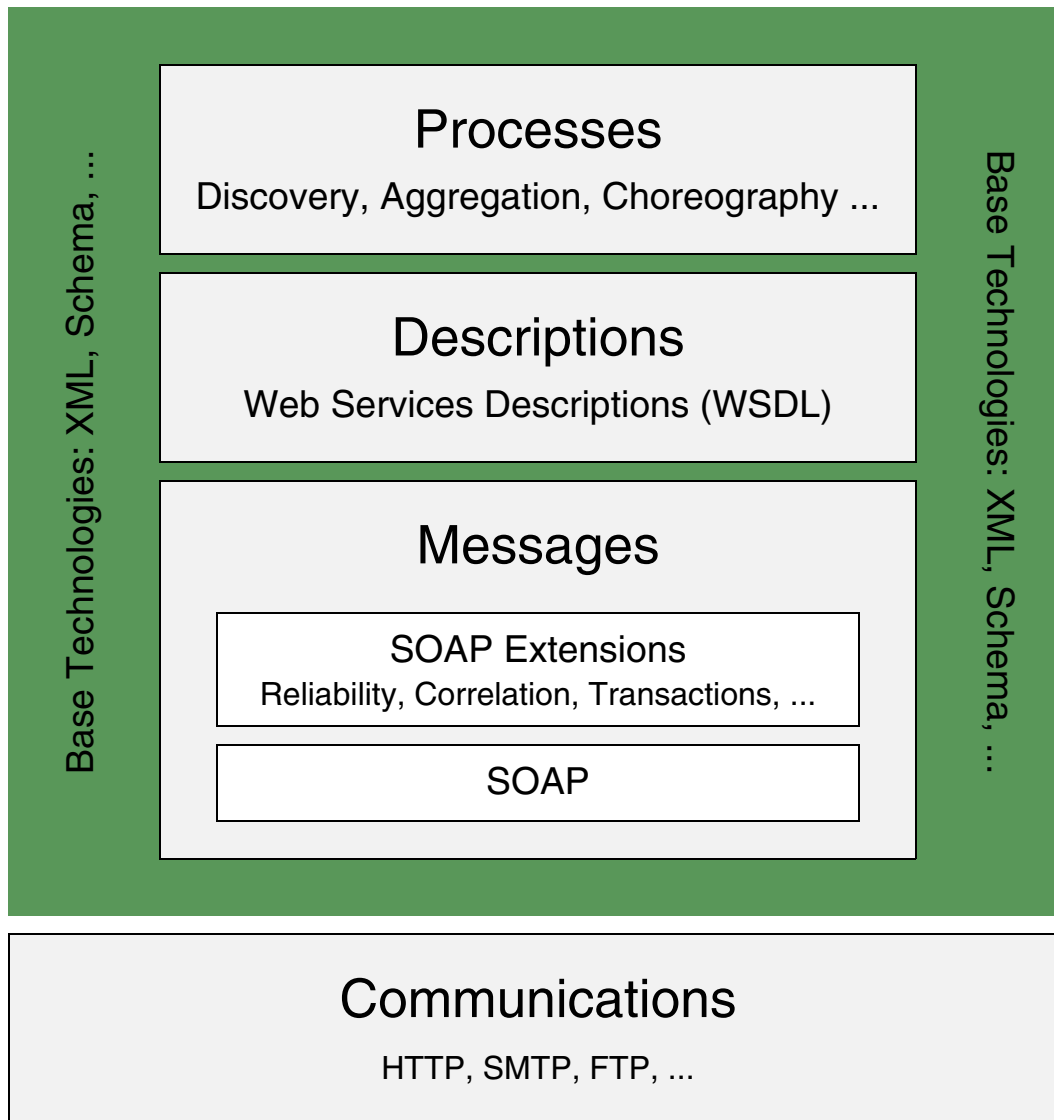
WEB SERVICES COMPONENTS



WEB SERVICES COMPONENTS FOR MANAGEMENT



STACK DIAGRAM



MAIN W3C DOCUMENTS

Web Services Description Language (WSDL)

Working Drafts - Version 2.0 - 2003

- Core Language
- Message Patterns
 - Bindings
 - Requirements
- Usage Scenarios

SOAP

Version 1.2 - W3C Recommendation - June 2003

- Part 0: Primer
- Part 1: Messaging Framework
- Part 2: Adjuncts

XML Schema

W3C Recommendation - May 2001

- Part 0: Primer
- Part 1: Structures
- Part 2: Datatypes

STRUCTURE WSDL DEFINITION

ABSTRACT INTERFACE TO THE WEB SERVICE

Independent of a specific
transport protocol
and Web address

BINDING

To associate the abstract interface
with a transport protocol

SERVICE

To associate the abstract interface
with a Web address

STRUCTURE WSDL DEFINITION

ABSTRACT INTERFACE - EXAMPLE

```
<message name="getflnOctetsRequest">
  <part name="community" type="xsd:string"/>
  <part name="index" type="xsd:unsignedInt"/>
</message>

<message name="getflnOctetsResponse">
  <part name="iflnOctets" type="xsd:unsignedInt"/>
</message>

<interface name="IfDataServiceInterface">
  <operation name="getflnOctets">
    <input message="mysns:getflnOctetsRequest"/>
    <output message="mysns:getflnOctetsResponse"/>
  </operation>
</interface>
```

STRUCTURE WSDL DEFINITION

BINDING TO A PROTOCOL - EXAMPLE

```
<binding name="ifDataServiceBinding"
  interface="mysns:IfDataServiceInterface">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getInOctets">
    <soap:operation soapAction=""/>
    <input>
      <soap:body use="encoded" namespace="urn:..."
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="encoded" namespace="urn:..."
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>
```

STRUCTURE WSDL DEFINITION

SERVICE AT A WEB ADDRESS - EXAMPLE

```
<service name="ifDataService" interface="myns:IfDataServiceInterface">  
  <endpoint name="ifDataServiceEndpoint"  
    binding="myns:ifDataServiceBinding"  
    <soap:address location="http://my.webservice.com/ifData"/>  
  </endpoint>  
</service>
```

MODULAR WSDL STRUCTURE

ABSTRACT
INTERFACES

IF MODULE

```
<message ...  
<operation ...  
  getIfTable
```

IP MODULE

```
<message ...  
<operation ...  
  getRouteTable
```

IF BINDING

```
<import IF MODULE  
<binding ...  
  SOAP
```

IP BINDING

```
<import IP MODULE  
<binding ...  
  SOAP
```

STANDARDIZED

SITE SPECIFIC

MY MGT. SERVICE

```
<import IF BINDING  
<import IP BINDING  
<service  
  http://...
```

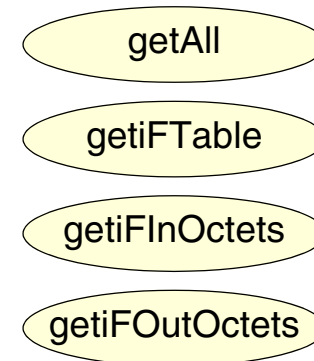
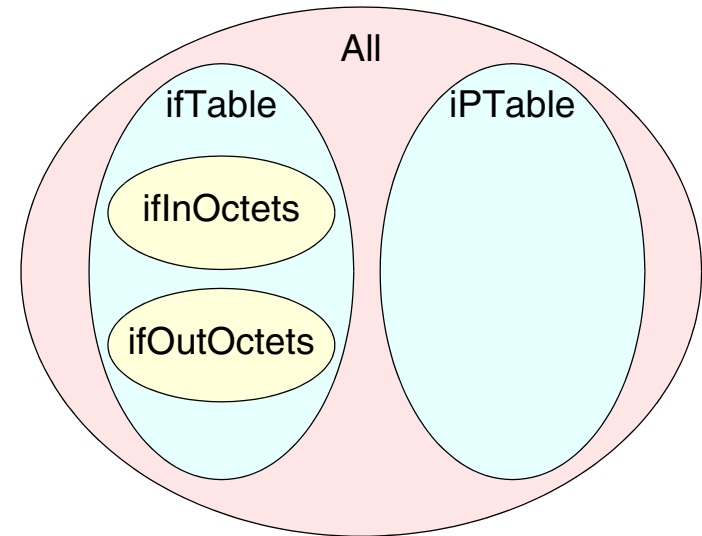
POSSIBLE MESSAGE STRUCTURE

COARSE

- get(OID, instance, ...)
- set (OID, instance, ...)
- ...

FINE

- getAll(...)
- getIfTable(...)
- getIfInOctets(index, ...)
- getIfOutOctets(index, ...)
- ...



POSSIBLE MESSAGE PARAMETERS

NON-TRANSPARENT

getflnOctets(index, amount)

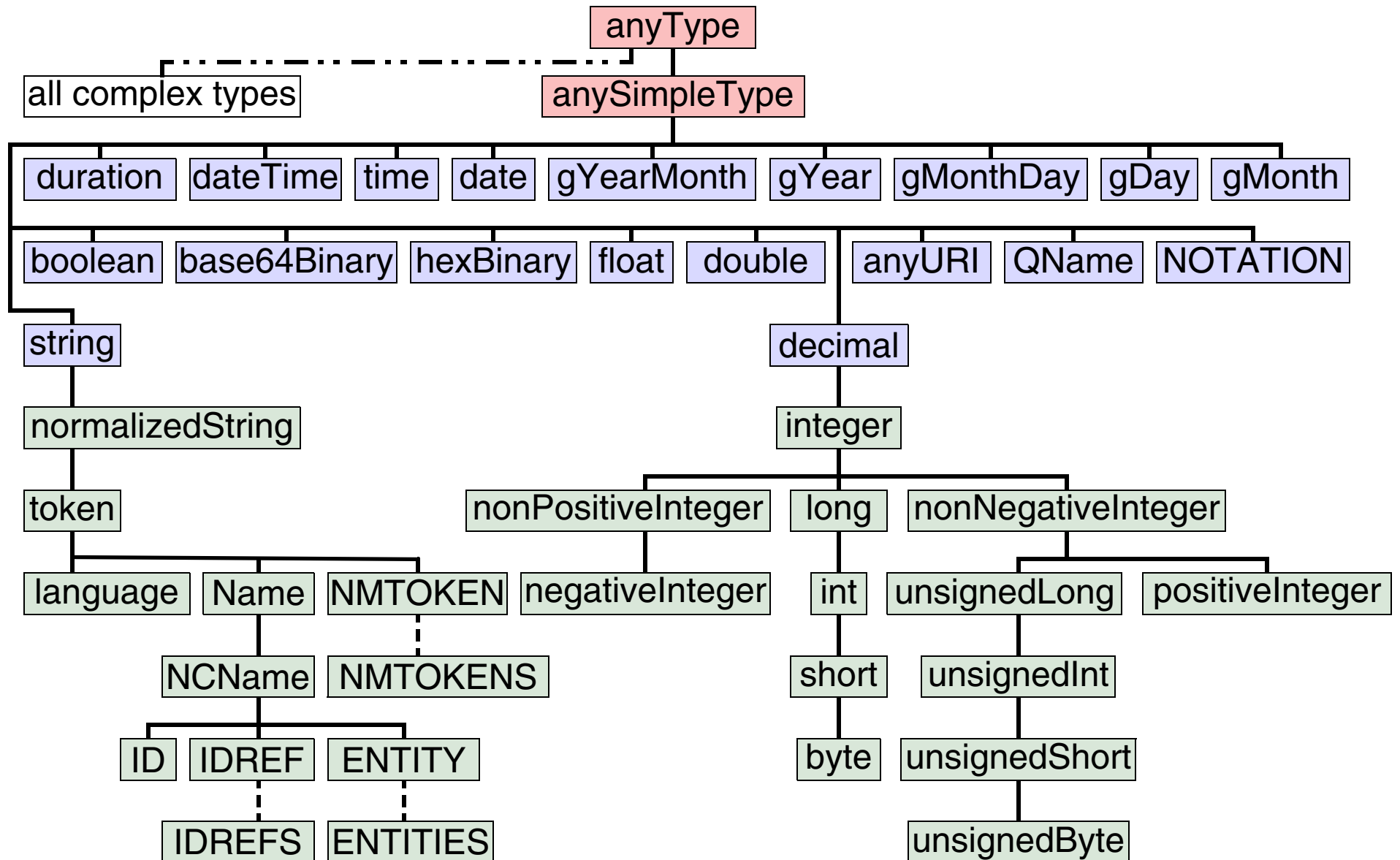
- Data parsed at WSDL level
- One level of standards: WSDL
 - Less flexible
- Easy integration with standard applications
 - Simple users (home environments)

TRANSPARENT

getflnOctets(string)

- Data parsed by higher level application
 - Data could be XML encoded
- Two levels of standards: WSDL operation & XML data
 - Powerful (e.g. XPATH / XQUERY)
 - Harder to use (professional operators)

DATA TYPES



ADVANCED FEATURES

TRANSACTIONS

- Business Transaction Protocol (OASIS)
- WS-Coordination + WS-Transaction (BEA, IBM, MS)
- WS-Composite Application Framework (Arjuna, Fujitsu, IONA, Oracle, Sun)

SECURITY

- WS-Security (IBM, OASIS)

CHOREOGRAPHY / ORCHESTRATION

- XLANG (MS), WSFL (IBM)
- BPEL4WS (IBM, MS, BEA)
 - WSCI (SUN, ...)
 - W3C

OVERVIEW

WHY WEB SERVICES?

WHAT ARE WEB SERVICES?

EXAMPLE

PERFORMANCE

TOOLS

CONCLUSIONS

EXAMPLE

PROTOTYPE

- ifTable
 - GetIfCell
 - GetIfColumn
 - GetIfRow
 - GetIfTable
- gSOAP (2.3.8)
- Net-SNMP (V5.0.x) Data retrieval functions
- Debian Linux, kernel v2.4.22, 800 Mhz Pentium

EXAMPLE

```
<complexType name="GetIfTableResponse">  
  <sequence>  
    <element name="ifEntry" type="utMon:ifEntry" minOccurs="1" maxOccurs="unbounded"/>  
  </sequence>  
</complexType>
```

```
<message name="GetIfTableRequest">  
  <part name="commuity" type="xsd:string"/>  
</message>
```

```
<message name="GetIfTableResponse">  
  <part name="-sizeTable" type="xsd:int"/>  
  <part name="ifEntry" type="utMon:ifEntry"/>  
</message>
```

```
<portType name="GetIfTableServicePortType">  
  <operation name="GetIfTable">  
    <documentation>Service definition of function utMon__GetIfTable</documentation>  
    <input message="tns:GetIfTableRequest"/>  
    <output message="tns:GetIfTableResponse"/>  
  </operation>  
</portType>
```

EXAMPLE

```
<complexType name="ifEntry">
  <sequence>
    <element name="ifIndex" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifDescr" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    <element name="ifType" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifMtu" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifSpeed" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifPhysAddress" type="xsd:string" minOccurs="1" maxOccurs="1" nillable="true"/>
    <element name="ifAdminStatus" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifOperStatus" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifLastChange" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifInOctets" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifInUcastPkts" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifInDiscards" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifInErrors" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifInUnknownProtos" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifOutOctets" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifOutUcastPkts" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
    <element name="ifOutErrors" type="xsd:unsignedInt" minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>
```

OVERVIEW

WHY WEB SERVICES?

WHAT ARE WEB SERVICES?

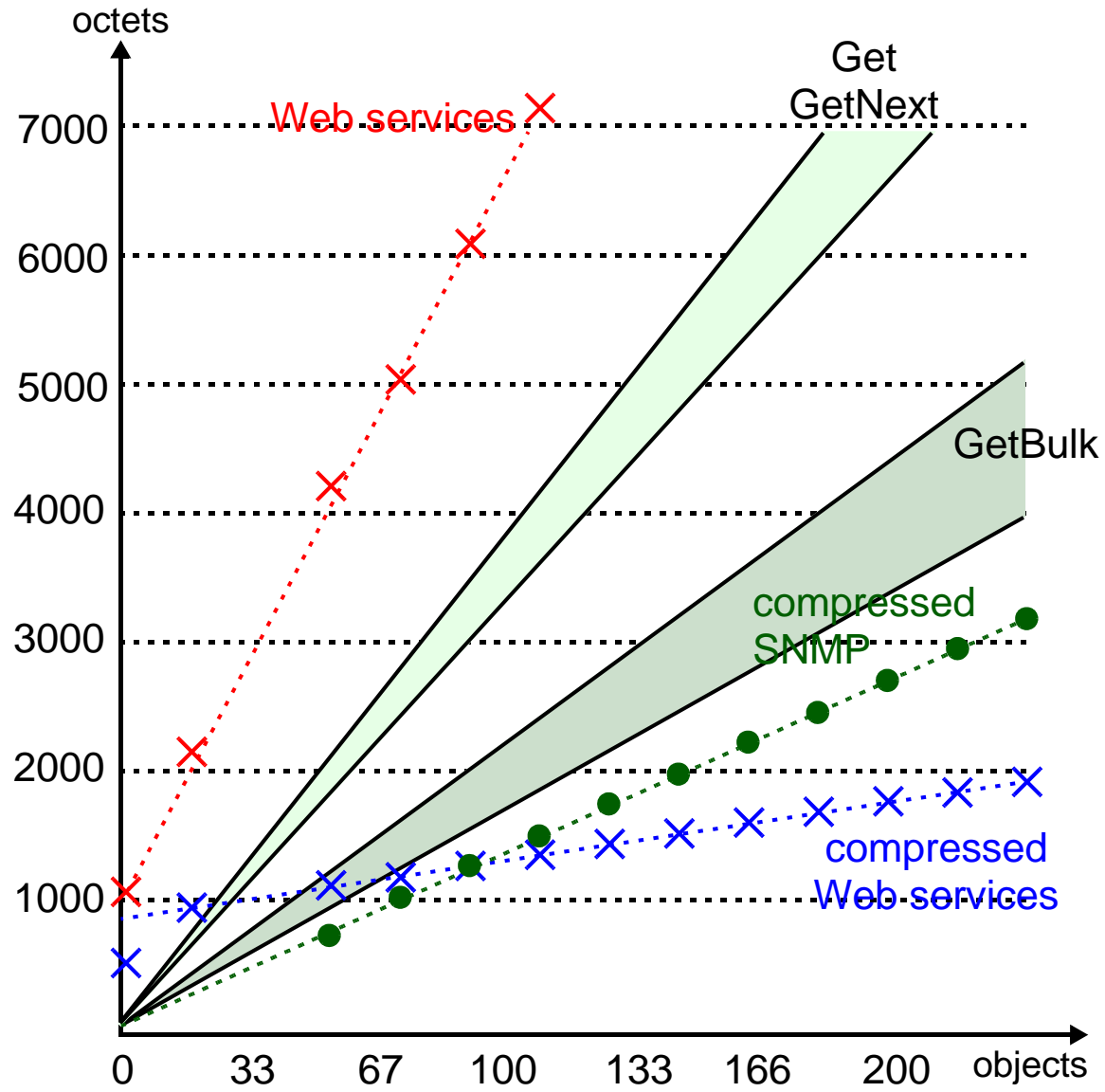
EXAMPLE

PERFORMANCE

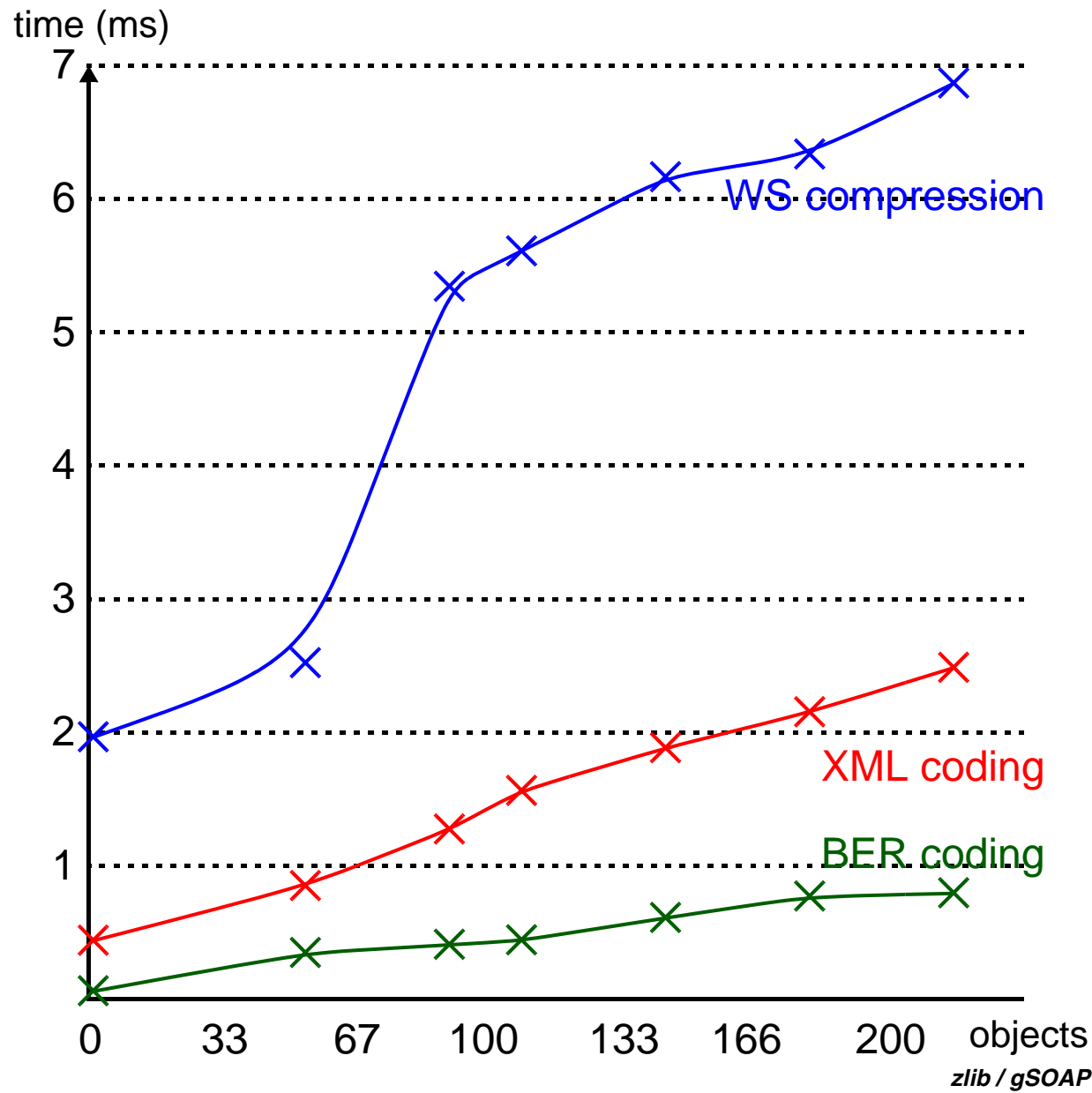
TOOLS

CONCLUSIONS

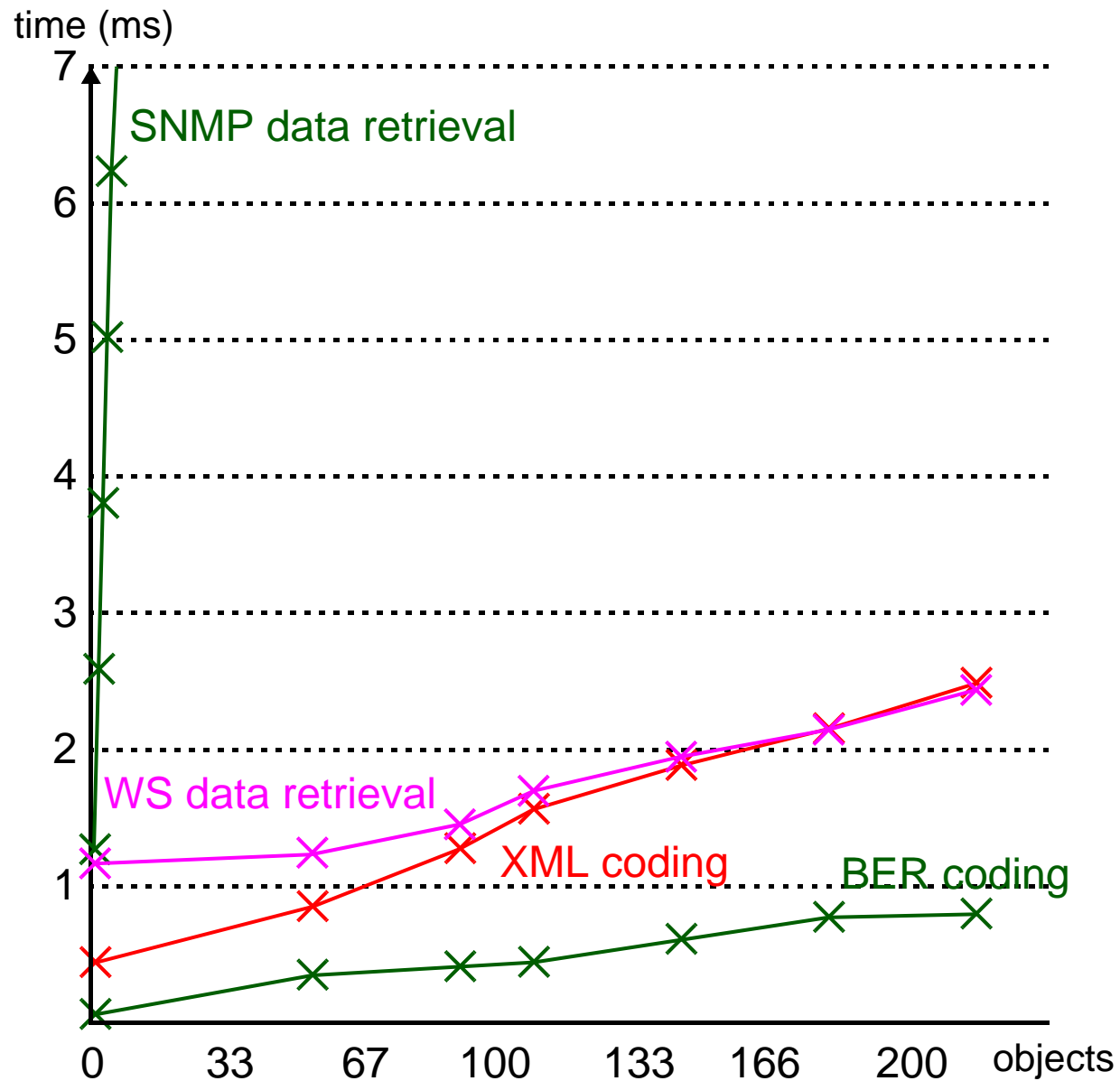
PERFORMANCE - BANDWIDTH



PERFORMANCE - CPU TIME - CODING & COMPRESSION



PERFORMANCE - CPU TIME - CODING & DATA RETRIEVAL



PERFORMANCE - MEMORY USAGE

	instructions	data	
		static	dynamic
SNMP	1972 KB	128 KB	70 - 160 KB
Web services	580 KB	470 B	4 KB

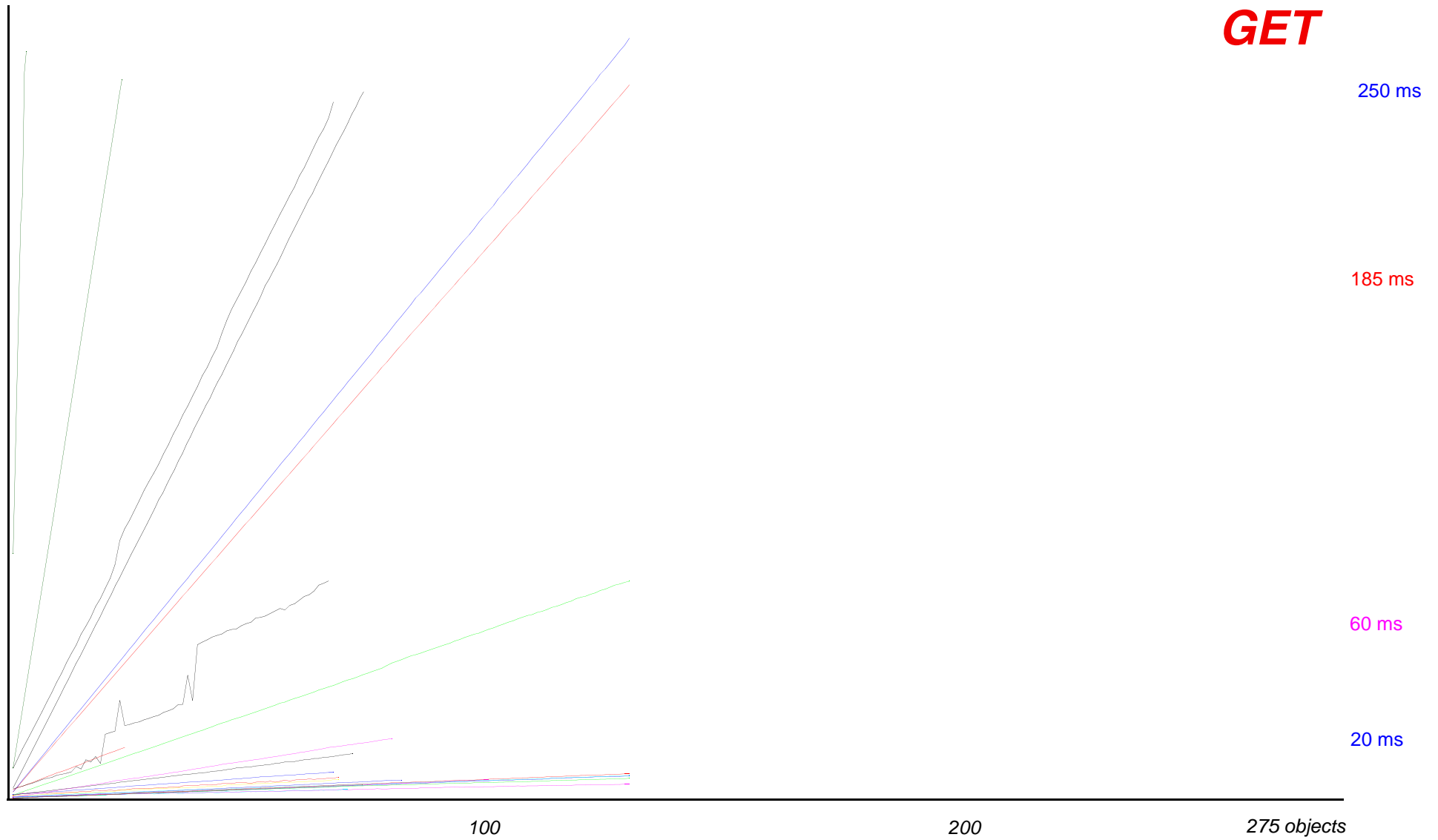
Note: zlib / gSOAP (V2.3.8) / Net-SNMP (5.0.9)

PERFORMANCE - ROUND-TRIP DELAY - 1

	1	22	66	270
WS	1,7	2,6	10,3	36,5
WS-Comp	3,3	4,3	5,6	11,8
SNMP-1	0,4	1,6	3,9	21,1
SNMP-2	0,4	1,9	5,0	
SNMP-3	0,5	1,6	4,2	
SNMP-4	0,5	1,7	4,4	
SNMP-5	0,5	1,8	4,8	
SNMP-6	0,7	2,2	5,7	
SNMP-7	0,8	1,8	2,9	
SNMP-8	0,9	1,6	3,9	
SNMP-9	0,9	6,6	18,5	
SNMP-10	1,1	1,8	3,4	58,5
SNMP-11	1,2	2,9	6,7	
SNMP-12	1,3	2,7	5,4	
SNMP-13	1,5	14,0	40,1	
SNMP-14	1,6	5,0	15,1	
SNMP-15	1,7	4,2	9,6	
SNMP-16	2,7	44,5	127,6	178,7
SNMP-17	2,7	47	140,4	251,7
SNMP-18	3,5	17,2		
SNMP-19	3,7	24,3	77,9	
SNMP-20	4,1	76,7	100,8	
SNMP-21	11,1	83,7	243,0	
SNMP-22	11,3	238,7	727,6	
SNMP-23	87,7	1822,2		

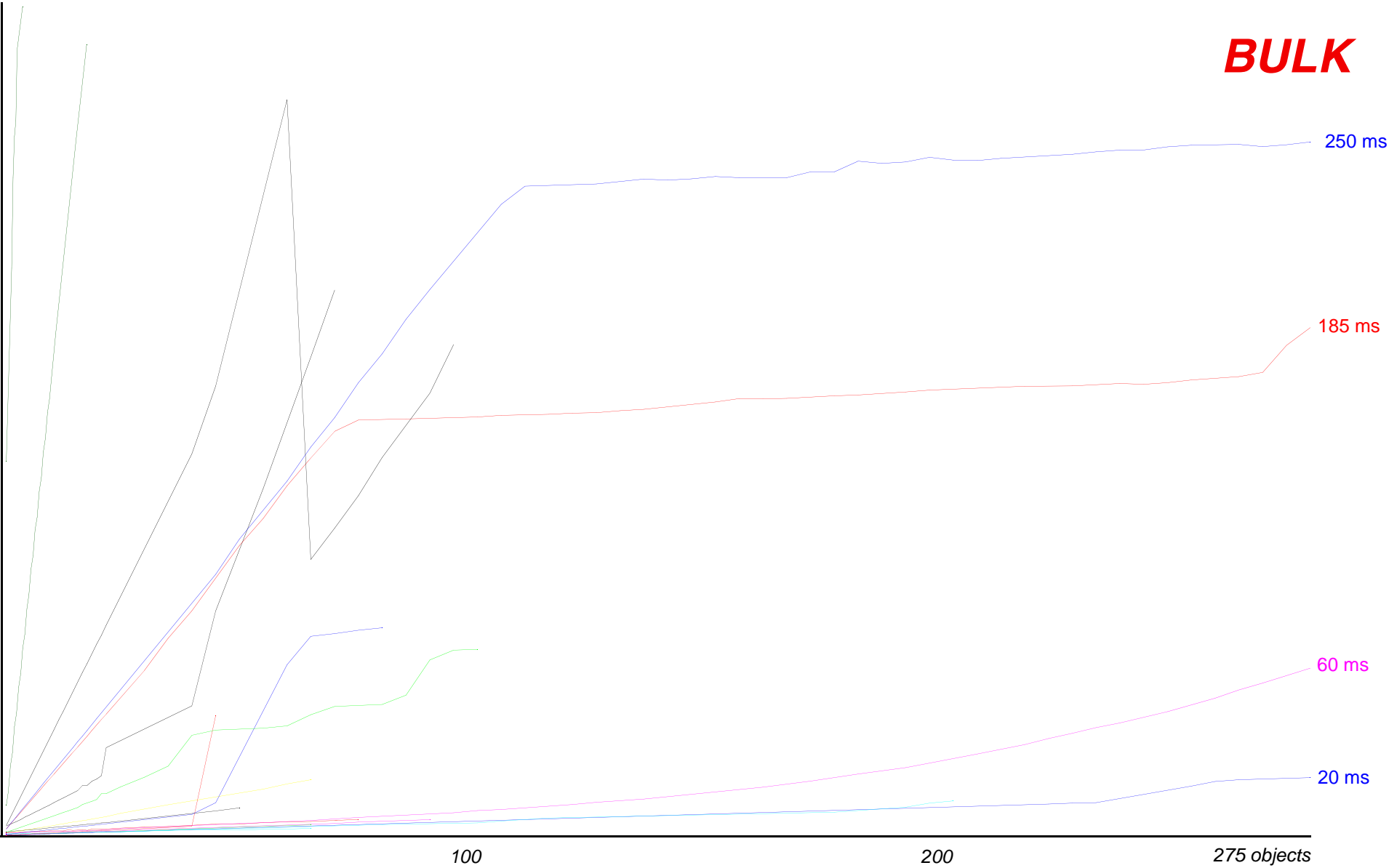
PERFORMANCE - ROUND-TRIP DELAY - SNMP GET

GET



PERFORMANCE - ROUND-TRIP DELAY - SNMP GETBULK

BULK



OVERVIEW

WHY WEB SERVICES?

WHAT ARE WEB SERVICES?

EXAMPLE

PERFORMANCE

TOOLS

CONCLUSIONS

TOOLS

gSOAP

WASP

easySOAP++

.NET

JBuilder

SunOne

OVERVIEW

WHY WEB SERVICES?

WHAT ARE WEB SERVICES?

EXAMPLE

PERFORMANCE

TOOLS

CONCLUSIONS

CONCLUSIONS

EVOLUTION OF SNMP FAILED

WE NEED REVOLUTION

WEB SERVICE IS AN INTERESTING TECHNOLOGY

MANY ISSUES STILL UNCLEAR

TOPIC FOR FUTURE RESEARCH

PERFORMANCE OF WEB SERVICES
MAY NOT BE A PROBLEM