



JavaOne™
Sun's 2003 Worldwide Java Developer Conference

The Other Web Services Paradigm: Document-Style

Michael Leventhal
XML Products Director
Tarari

Sen Zhang
Senior Principal
Dante Consulting

Choose the Right Web Services Technology for b2b Applications

Why: Why Document SOAP is the critical next step in using Web Services between enterprises.

How: How to build a high-performance Document Web Services framework.

So Who Are You Guys Anyway?

- **Michael Leventhal** led the team at Commerce One that created the DocSOAP XDK, an open source Java™ framework for Document Web Services
- Leventhal is currently director of XML Products for Tarari, creating hardware solutions to accelerating XML processing
- **Sen Zhang** was Architect and Technical Lead for the DocSOAP XDK project
- Zhang is currently a Senior Principal for Dante Consulting, helping enterprises to choose the right B2B strategies, design and build high performance Web Services applications

The Fundamental Problem Is Paradigm

Do we understand how to build interoperable, reliable, high-performance Web Services?

- **No**, because the tools, testing and standards work have concentrated on RPC. RPC will not dominate the world of enterprise-to-enterprise Web Services.
- Document Web Services will dominate but currently lag in tools, testing, standards—and understanding

Presentation Agenda

- Whither Web Services—RPC or Document-centric?
- Document Web Services Technologies
- BOF: Issues and Techniques for Building High-Performance Document Web Services
- DocSOAP XDK—an Open-Source Java™ Document Web Services Framework



JavaOne™
Sun's 2003 Worldwide Java Developer Conference

Part 1—Paradigm

Whither Web Services—RPC or Document-Centric

Java™

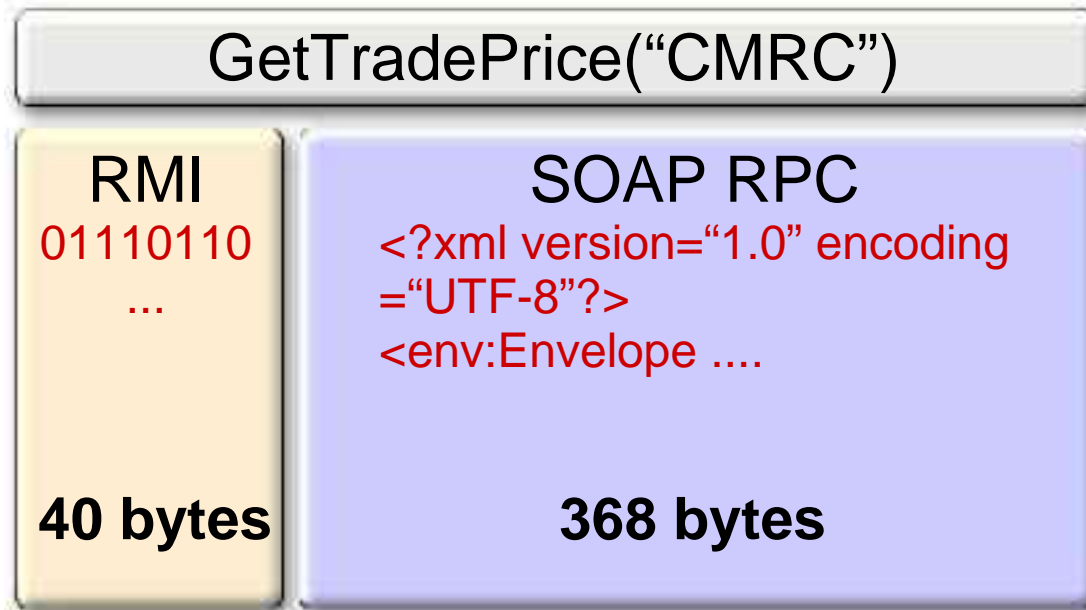
RPC and Document Web Services Defined

RPC-based Web Services	Document Web Services
<ul style="list-style-type: none">•Emulate function calls•Schema is not sufficient to interpret payload•Fine-grained•Tightly coupled•Point-to-point•Synchronous programming model (blocking call)	<ul style="list-style-type: none">•Exchange business documents•XML payload defined by schemas but can carry binary data as well•Coarse-grained•Loosely coupled•End-to-end•Asynchronous messaging

Limitations of SOAP-RPC (1)

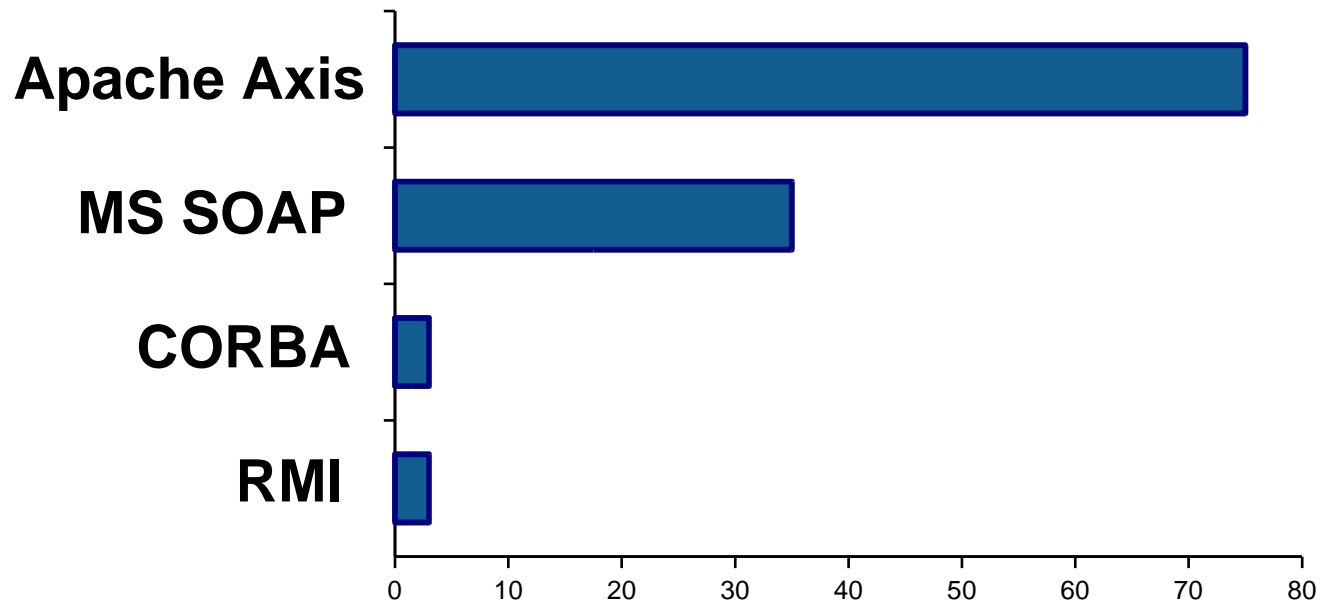
Inefficient Encoding

“XML-RPC ... has the great virtue of being worse than all its competitors.” –David Mertz



The SOAP-RPC Performance Problem (2)

Latency (ms) GetInteger

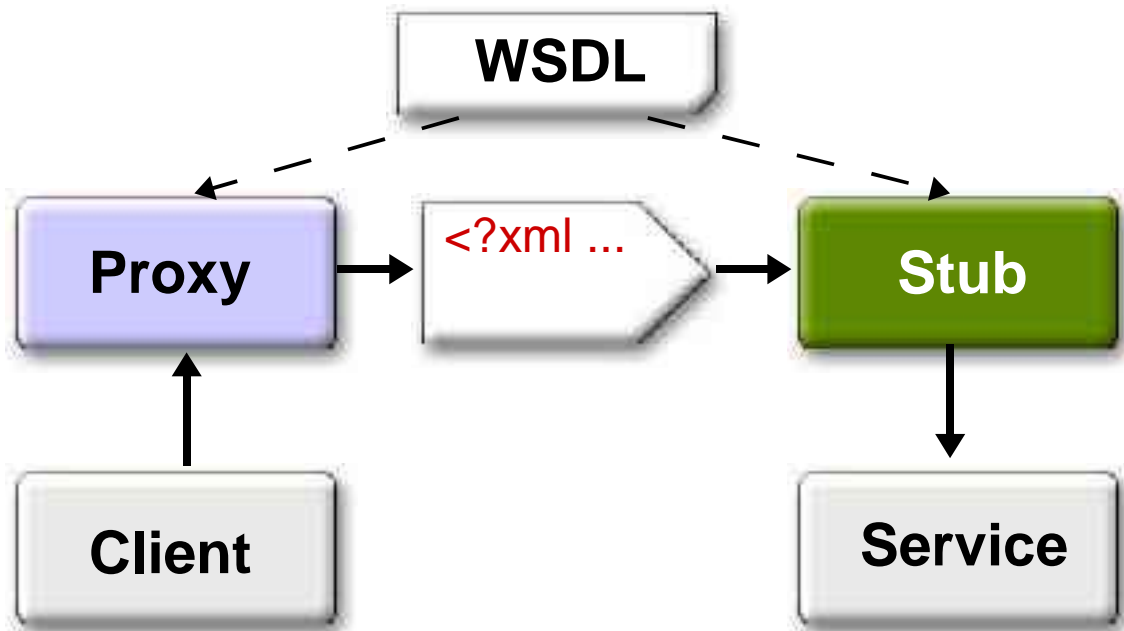


Source: Davis, et.al., *Latency Performance of SOAP Implementations*, IEEE Cluster Computing and the Grid 2002

Limitations of SOAP-RPC (3)

Tight Coupling

WSDL
generated
interfaces
difficult to
evolve



Limitations of SOAP-RPC (4)

Fine-grained

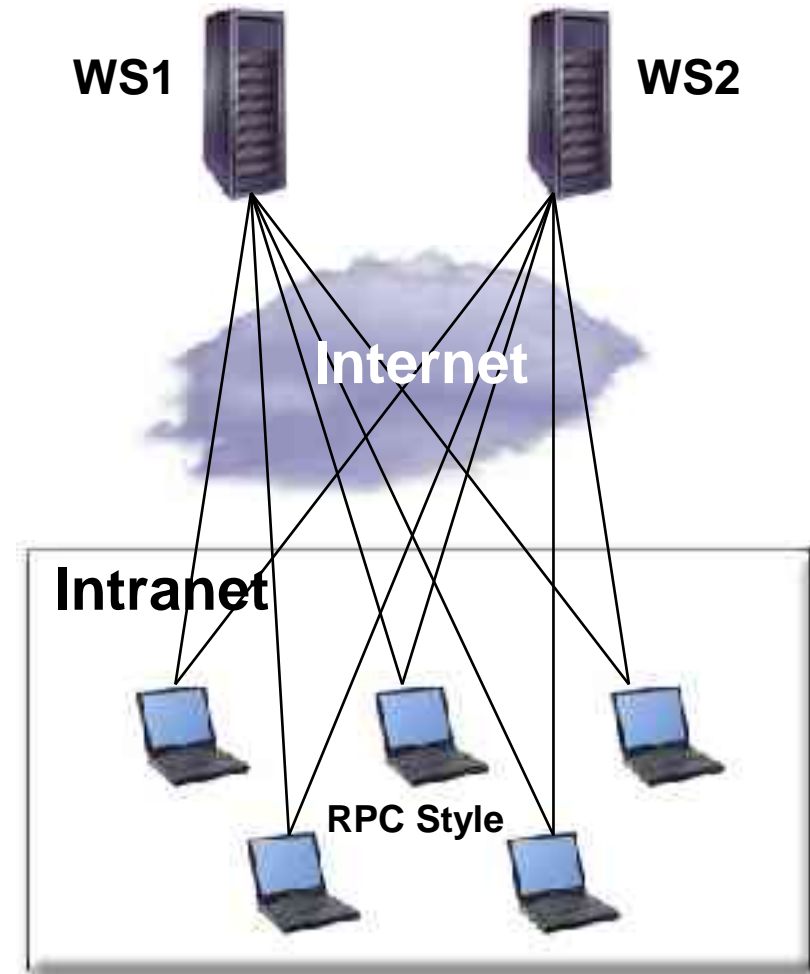
Litters Internet with small, chatty datagrams

Alternative: Coarse-grained function calls break programming paradigm

Blocking Calls

Unpredictable latency

Alternative: Asynchronous programming is complex



Where Do RPC and Document Web Services Excel?

RPC-based Web Services	Document Web Services
<p data-bbox="220 548 826 602">Enterprise computing</p> <ul data-bbox="220 672 977 1146" style="list-style-type: none"><li data-bbox="220 672 639 726">•High bandwidth<li data-bbox="220 796 542 851">•Low latency<li data-bbox="220 921 977 1026">•Well-controlled environment, trusted partnerships<li data-bbox="220 1096 807 1150">•Tight system coupling	<p data-bbox="1070 548 1750 602">Enterprise-to-Enterprise</p> <ul data-bbox="1070 672 1682 1146" style="list-style-type: none"><li data-bbox="1070 672 1630 726">•Unreliable bandwidth<li data-bbox="1070 796 1406 851">•High latency<li data-bbox="1070 921 1682 1026">•Unknown environment, “blind” trust<li data-bbox="1070 1096 1489 1150">•Loose Coupling



JavaOne™
Sun's 2003 Worldwide Java Developer Conference

Part 2—Technology Basics

Document Web
Services
Technologies

Document-Style WSDL

- WSDL provides a document-style service contract between sender and receiver
- Abstract Message Description
 - Provides name for each part: PARTNAME
 - Provides type of each message part (e.g., schema for XML parts)
- Binding Description
 - Provides messaging packaging format
- **Design WSDL first!**

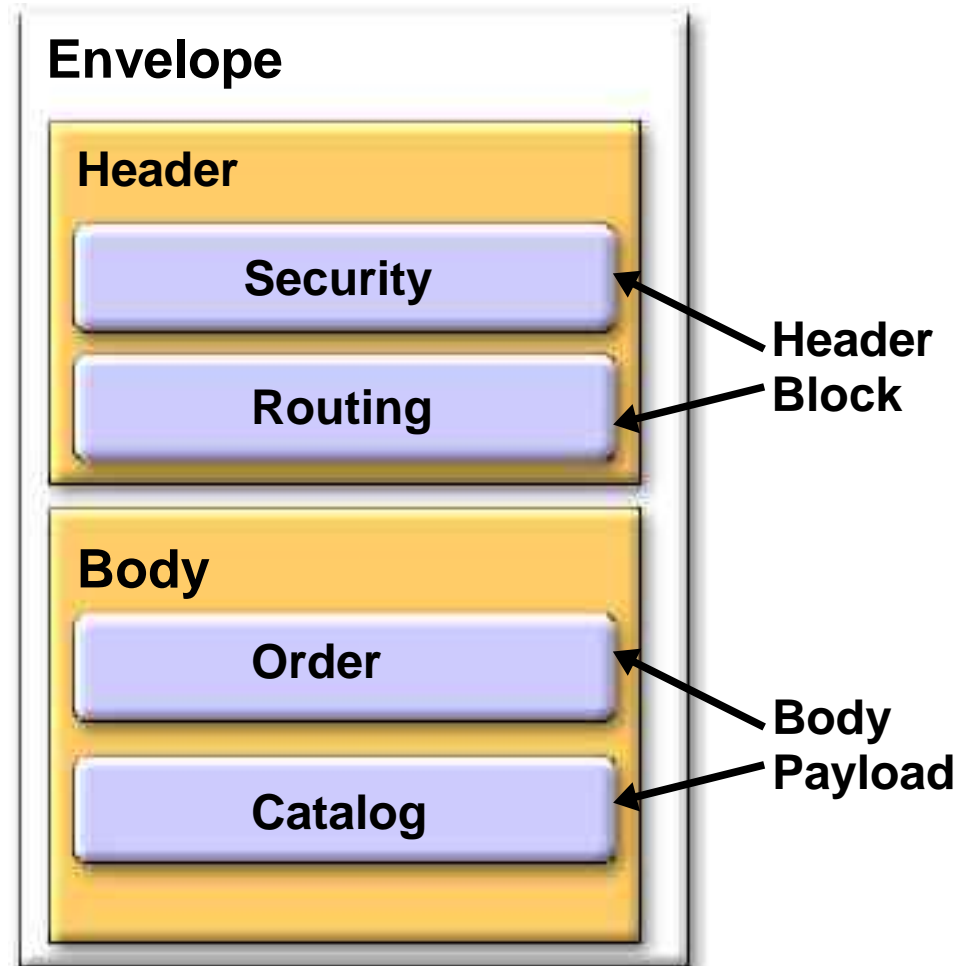
Document-Style SOAP

Header

- Specifies message-level services

Payload

- Opaque
- Schema-defined
- Large
- Complex



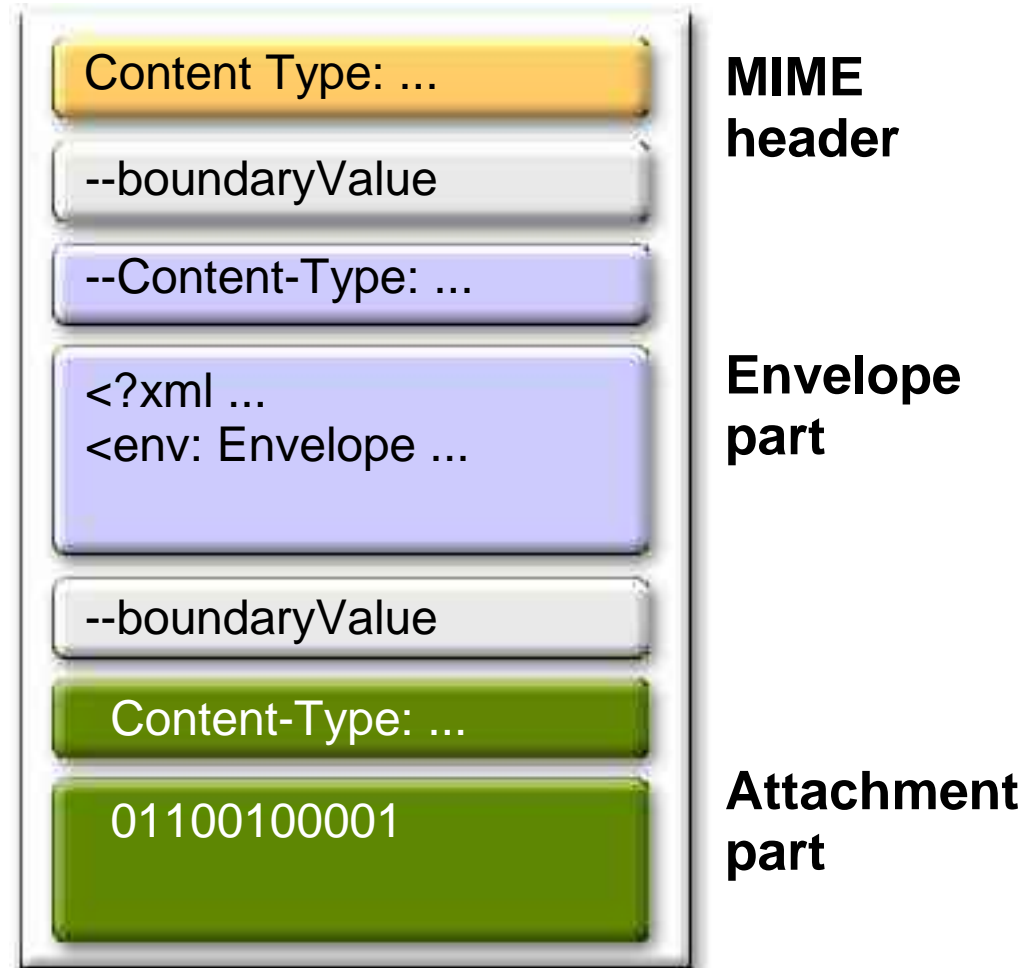
SOAP Packaging—Binary Payload

- In Document-centric SOAP, binary payloads become significant
- Binary payloads tend to be large in size
- XML-embedding solution: Base64 or Hex binary encoding:
 - Pro: maintains pure XML format
 - Con: inflates data size
 - Con: high encoding/decoding overhead

SOAP Packaging—SwA

SOAP With Attachments

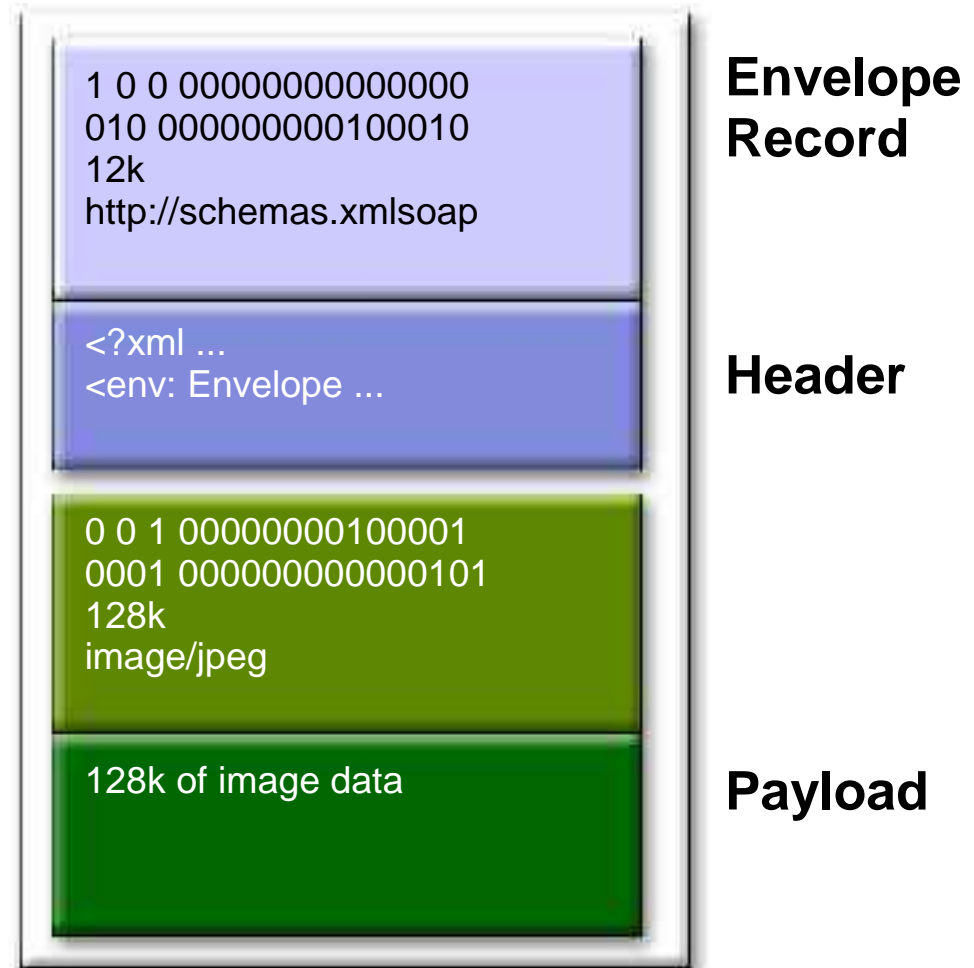
- Multipart MIME
- Reuse existing messaging infrastructure



SOAP Packaging—DIME

Direct Internet Message Encapsulation (DIME)

- Binary format
- Fixed header fields
- Efficient message parsing



XML Document Processing software

Document-centric Web Services requires robust XML processing capabilities

- Full support of XML Schema
 - Adopt standards
 - Design your own schemas
- Validation
- Handling of XML document fragments
- XML inter-document references



JavaOne™
Sun's 2003 Worldwide Java Developer Conference

Part 3—Techniques and Issues

Approaches and Issues for High- Performance Document Web Services

Challenges Facing Document Web Services

- Large data handling
- Binary payload
- Strong API support
- Strong document support
- XML validation
- Referential integrity



JavaOne™
Sun's 2003 Worldwide Java Developer Conference

Part 4 – Case Study

DocSOAP XDK – an Open-Source Java™ Document Web Services Framework

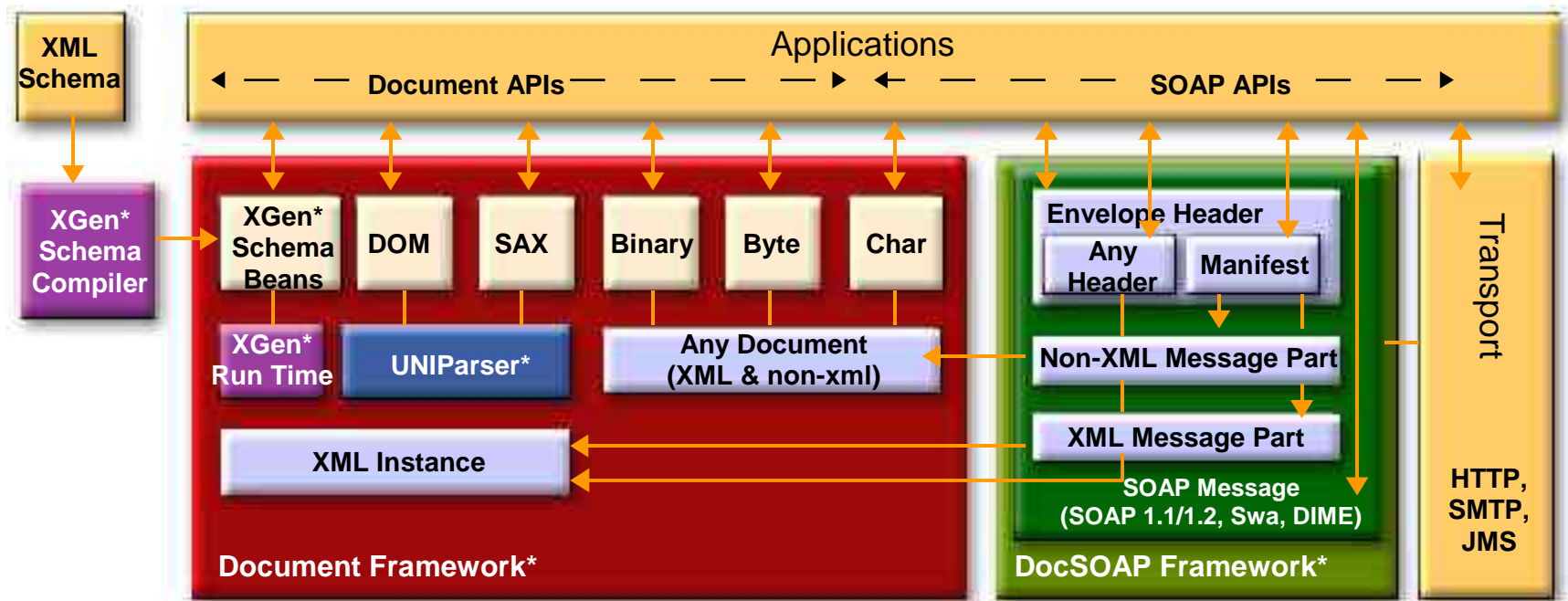
DocSOAP XDK Features

- The only toolkit expressly designed for document-centric Web Services
- Open-source, free, Java™, J2EE™ technology compliant
- SOAP, Document, and XML APIs
- Designed for:
 - Protection of message integrity
 - High-performance on messages bearing typical XML business documents
 - Ease of programming for document SOAP and XML payload handling
- 2X performance compared to other toolkits on document SOAP messages

DocSOAP XDK Components

- DocSOAP Framework—API for document-centric SOAP Processing
- Document Framework—Unified, extensible API for handling documents in binary, text, and XML formats
- XGen—Generates Java™ APIs from XML Schemas and converts between XML and Java™ technologies at runtime (JAXB equivalent)
- Uniparser—XML Parser and Specialized Chained Entity Resolution

DocSOAP XDK Architecture



* Conductor DocSOAP XDK Component

Key Innovations

The DocSOAP XDK Team invented new approaches to SOAP and XML processing in order to implement high-performance, reliable, and easy-to-implement document Web Services:

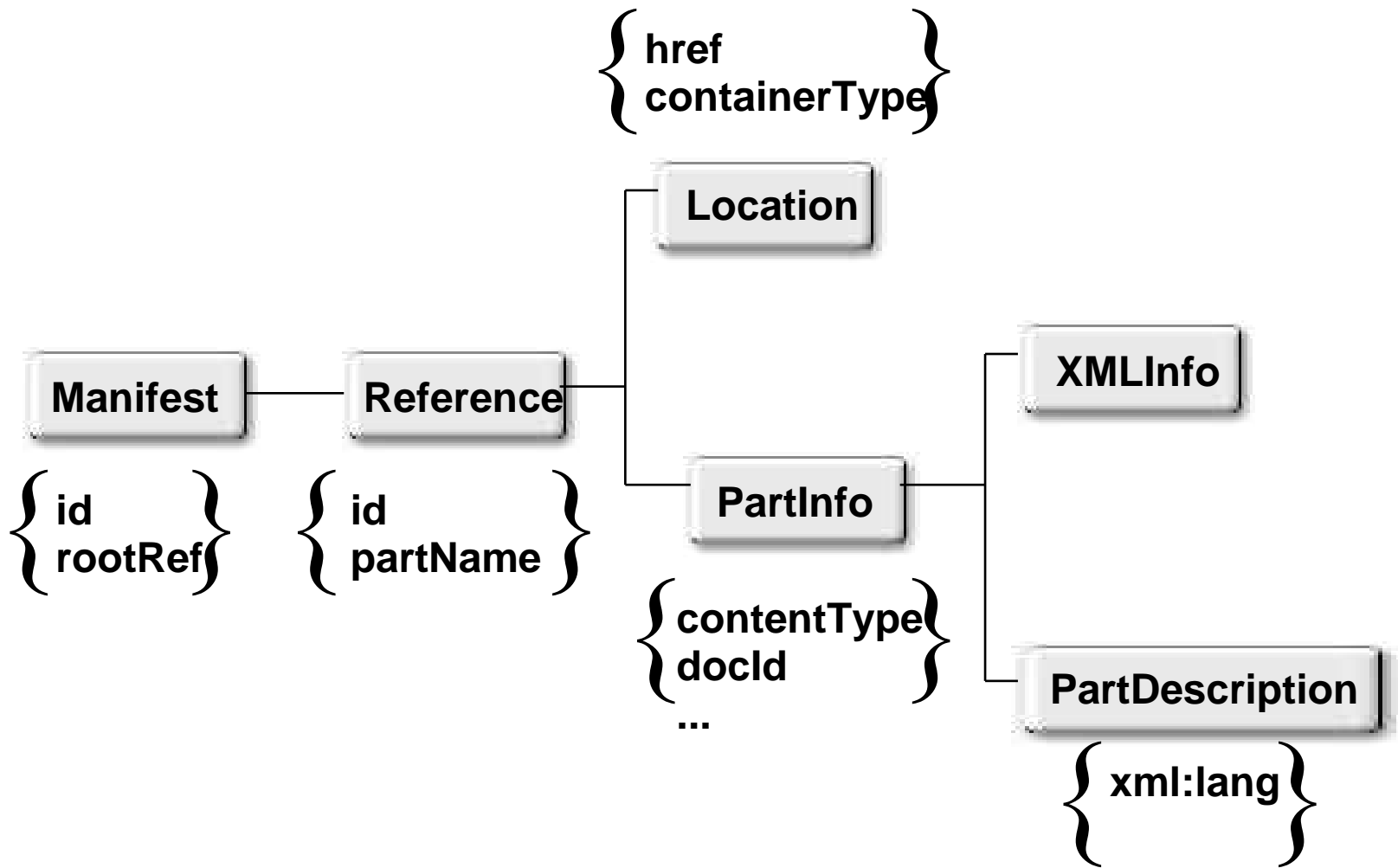
- Manifest Abstraction
- Document Framework

Key Innovations—DocSOAP Manifest Abstraction

Two complementary aspects of the implementation:

Manifest API	Manifest SOAP Header
Manifest-mediated access to parts ensures message integrity	Header allows knowledge of contents to be shared between sender and receiver and SOAP pipeline nodes

Key Innovations—Manifest SOAP Header Structure



Key Innovations—Manifest SOAP Header Features (1)

- Catalog of message
- Meta-data about each message part, XML processing info
- Packaging independent
- Only way to link WSDL PARTNAME to message parts

Key Innovations—Manifest SOAP Header Features (2)

- Indirect intra-message references through manifest preserves referential integrity
- Only way to support remote or external parts
- Manifest is materialized if not provided by sender

Key Innovations— Manifest-Mediated SOAP API

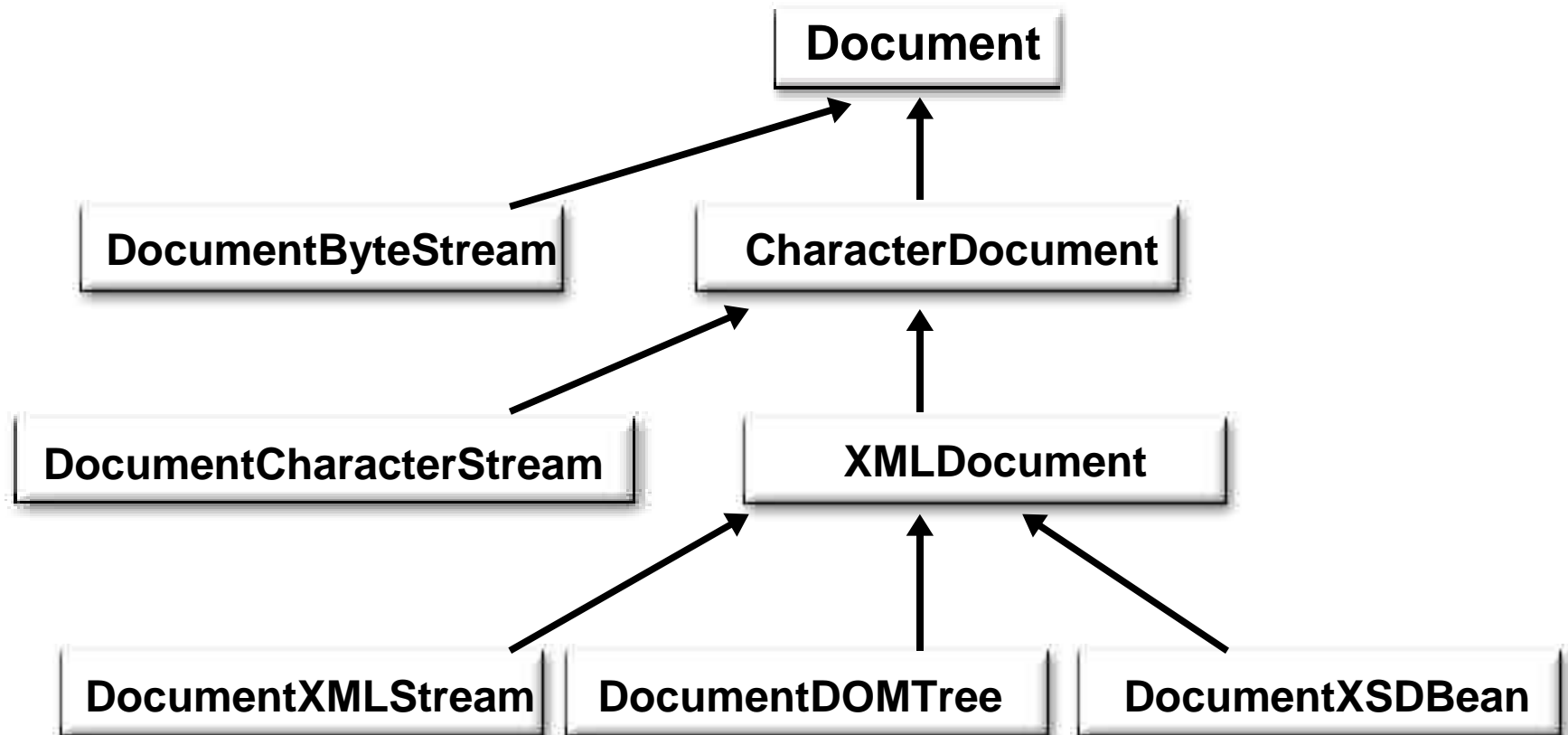
Abstraction over packaging details Simplifies SOAP Programming

- Message descriptors are part “handles”
- Cannot manipulate SOAP message parts (add, remove, inspect, set properties) except through descriptors
- Guaranteed synchronization between manifest and message
- Location field is protected so user cannot misset. User does not need to know packaging rules.
- Can query for parts over descriptor properties

Key Innovations—Document Framework Abstraction

- Single class library for handling all types of documents: binary, text and XML
- Single-rooted, extensible deep hierarchy of self-contained, self-describing document types
- Unites and interrelates all major programming interfaces to XML documents: SAX, DOM, and schema-generated Java™ classes

Key Innovations—Document Framework Abstraction



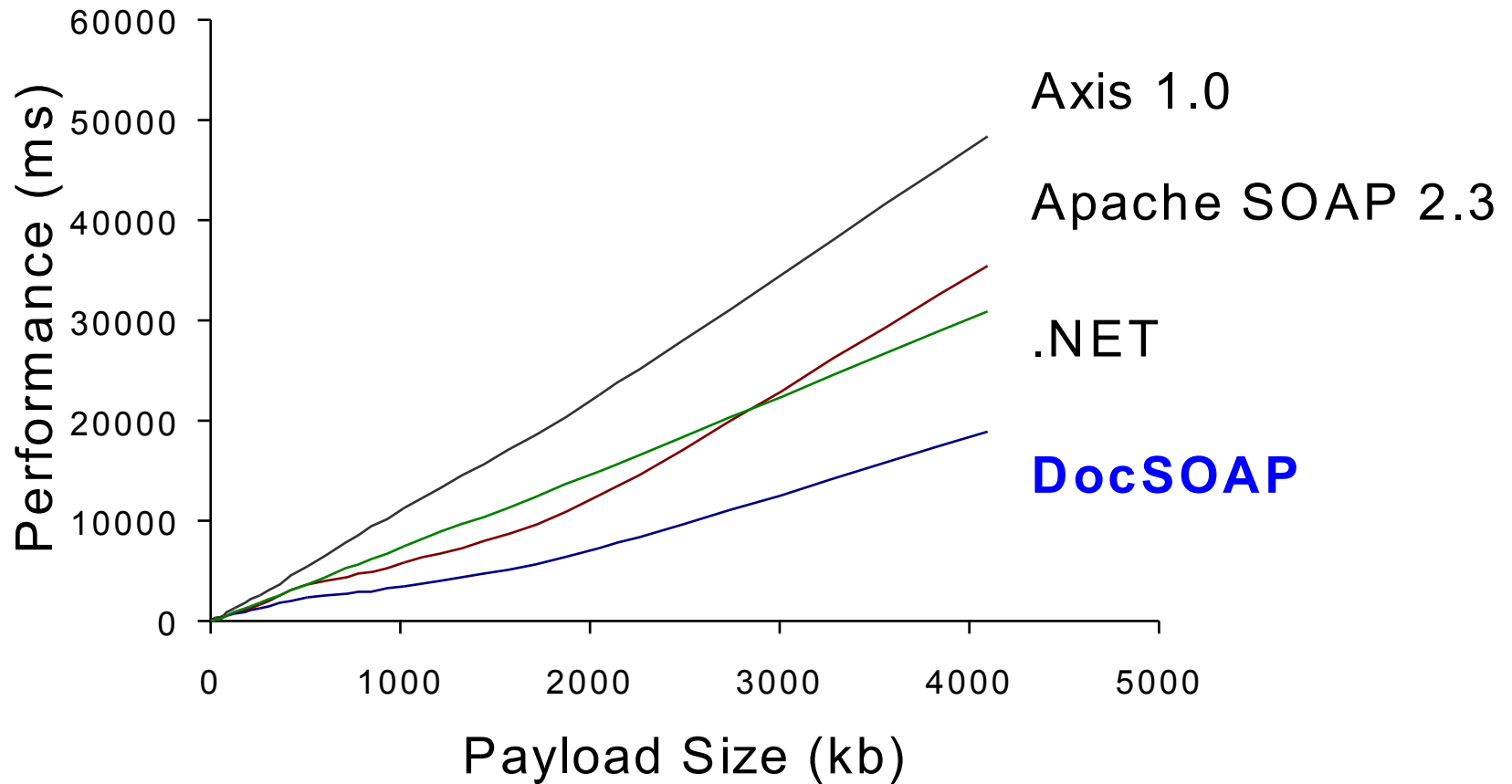
Key Innovations—Document Framework Performance

- Hierarchy of document types gives the programmer a choice of using lowest to highest “cost” representations
- Java™ Activation Framework (JAF) Data Source used for lazy processing – keeps document in source location until actually needed
- Uses fast, low-memory stream processing wherever possible

DocSOAP XDK Implementation Experience

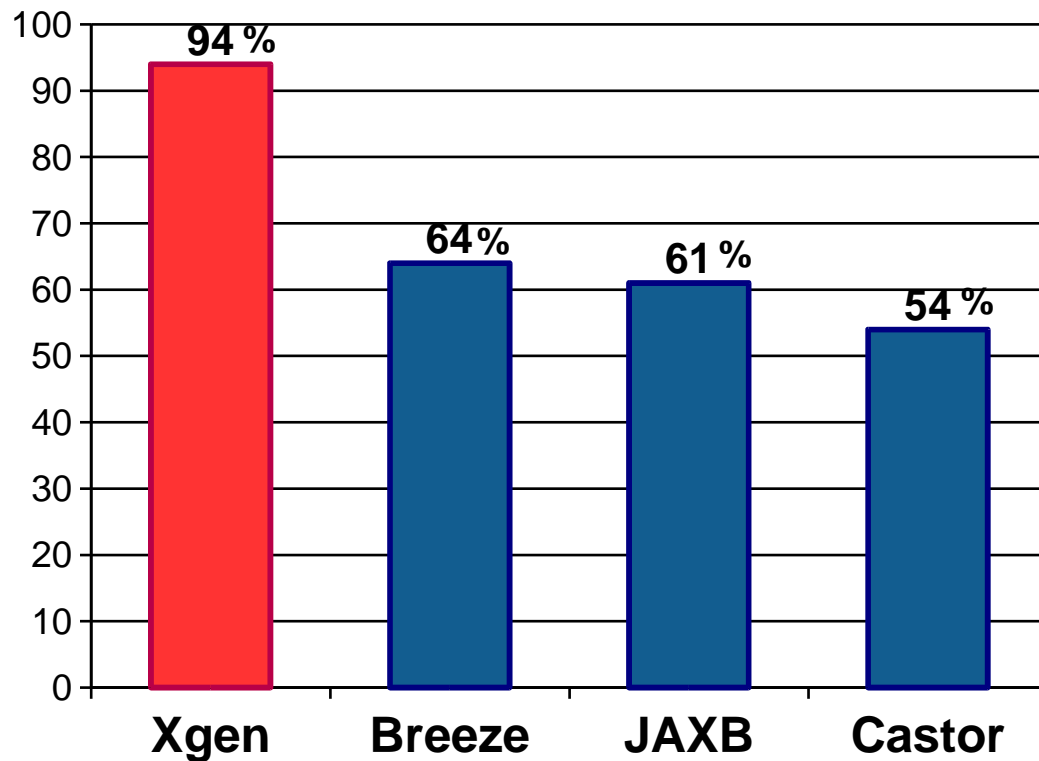
- Core infrastructure used in Commerce One's Web Services Platform Conductor
- Enabled rapid development of a Web Services platform by hundreds of developers
- Extremely robust and error-free
- Interoperable with wide range of SOAP toolkits and Web Services including .NET
- Tested with xCBL, one of the largest b2b XML document libraries with an average size of .5Mb per document
- Used in approximately one dozen early adopter implementations of Commerce One Conductor

Benchmarks—DocSOAP Performance



Benchmarks—XGen (Schema to Java™) Schema Feature Coverage

XML Schema support



DocSOAP XDK Download

The DocSOAP XDK was released as free, open-source software with an unrestricted development and distribution license (OSI model) in order to encourage the adoption of document-centric Web Services. It is available at:



<http://www.commerceone.com/developers>

Questions and Comments:

Michael.Leventhal@tarari.com

Sen.Zhang@danteconsulting.com



JavaOne™
Sun's 2003 Worldwide Java Developer Conference

Conclusion



Summary

- Document-centric and RPC Web Services fill quite different niches. Document Web Services are more suitable for most enterprise-to-enterprise interaction over the Internet.
- Document Web Services pose special problems and demand particular solutions to achieve reliability, high-performance and ease-of-use
- Tools such as the DocSOAP XDK are needed to enable document Web Services

The Future

- Web Services will prove to be the next revolution in Internet computing.
- Document-centric Web Services will be the dominant paradigm for b2b.
- Still much to do in tool support and standards.

BOF Tonight!

Issues and Techniques for Building High-Performance Document Web Services. **8:30pm–9:30pm Esplanade 301**

Q&A

Java™



JavaOneSM

Sun's 2003 Worldwide Java Developer Conference

JavaSM

java.sun.com/javaone/sf