NET.OBJECTDAYS 2002



Enterprise Messaging Architectures with JMS and XML

Marian Kuffner & Gerald Loeffler

Java Architects

Sun Java Center

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems
We make the net work.

Sun microsystems

Agenda

- Introduction
- JMS Fundamentals
- XML Fundamentals
- JMS Details
- Transporting XML via JMS
- JMS Performance Summary
- JMS Security

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Part I
Introduction

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

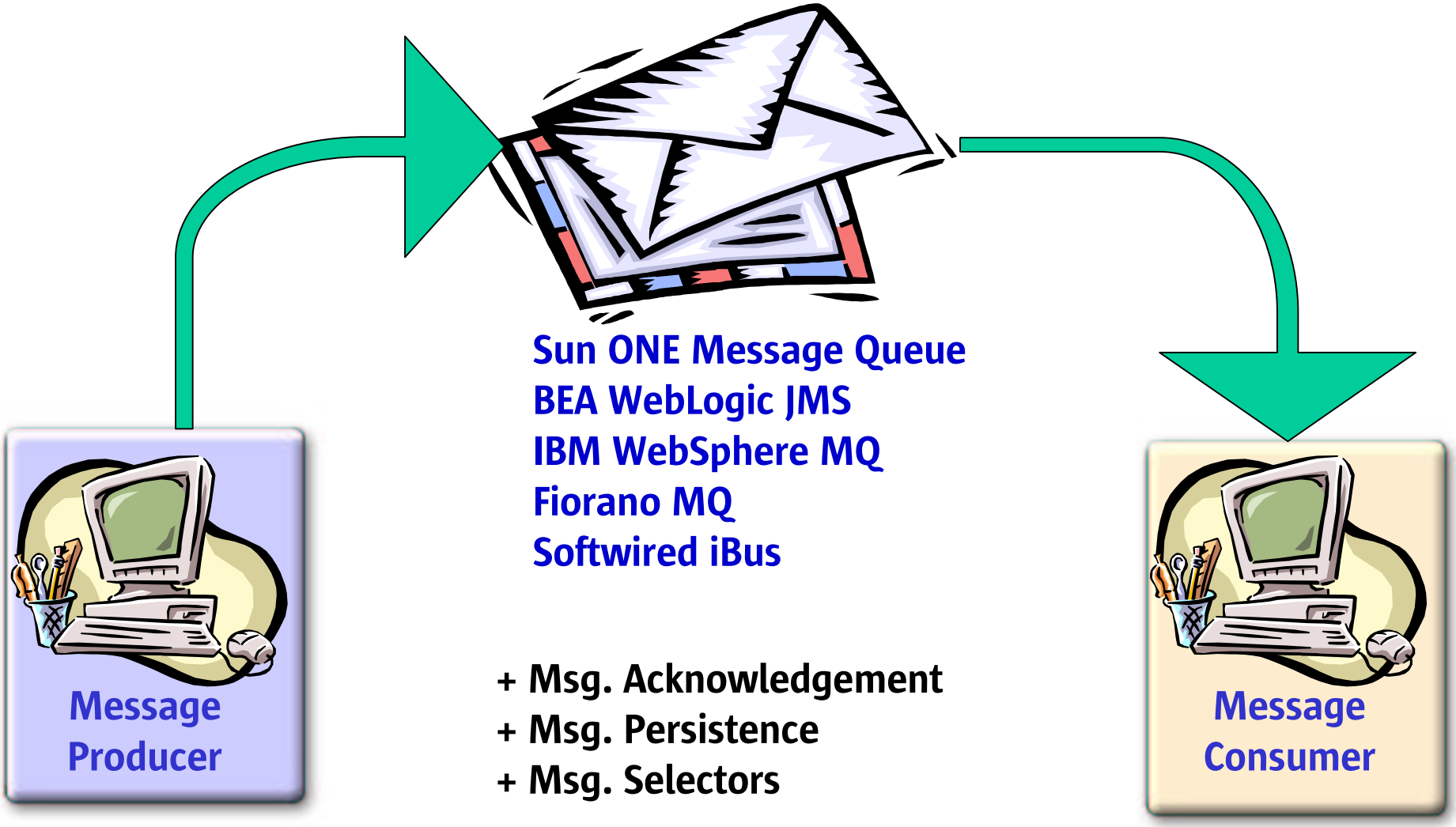
Introduction

- Motivation for Message Oriented Middleware (MOM) and JMS
- Differences from other Distributed Computer Technologies
- JMS History
- J2EE and JMS
- Transporting XML via JMS
- EAI Overview and Scenarios

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Motivation for MOM



Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Message Oriented Middleware

- Allow clients to exchange messages using a common service
- Support lose coupling of systems
- Application Integration over different platforms, mostly using native APIs

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

History

- MOM predates JMS: Tibco, IBM
- Proprietary Java APIs spread
- Standard compatible API needed
- 1st JMS Draft 8/1998
- Current Version 1.0.2b ca. 6/2002
- JMS is part of J2EE

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JMS is Suited to EAI Because…

- MOM is mature and x-platform (provides not only a JMS interface)
- Message format conversion (also custom)
- No need to run systems within single executable image
- Systems remain in their native environment (host, Linux, Windows, ...)
- No need to synchronise execution of systems (but beware of Queue overflow!)

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

NET.OBJECTDAYS 2002



End of Part I – Introduction

Marian.Kuffner@sun.com

Gerald.Loeffler@sun.com

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems

We make the net work.

Part II
JMS Fundamentals

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

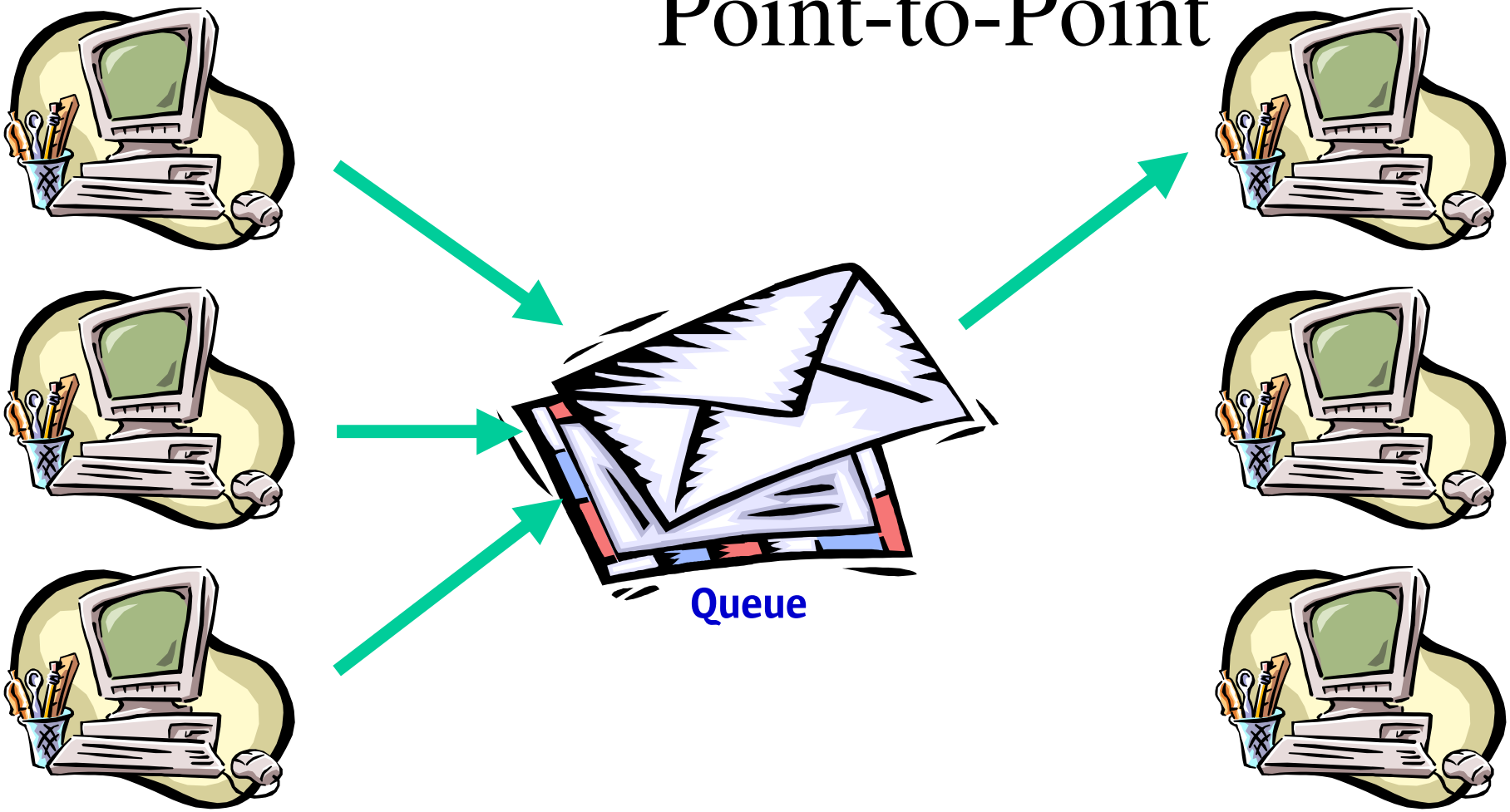
Part II: JMS Fundamentals

- Messaging Domains
- Destinations
- Message Types
- Anatomy of a JMS Message
- JMS Properties
- Message Driven Beans

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

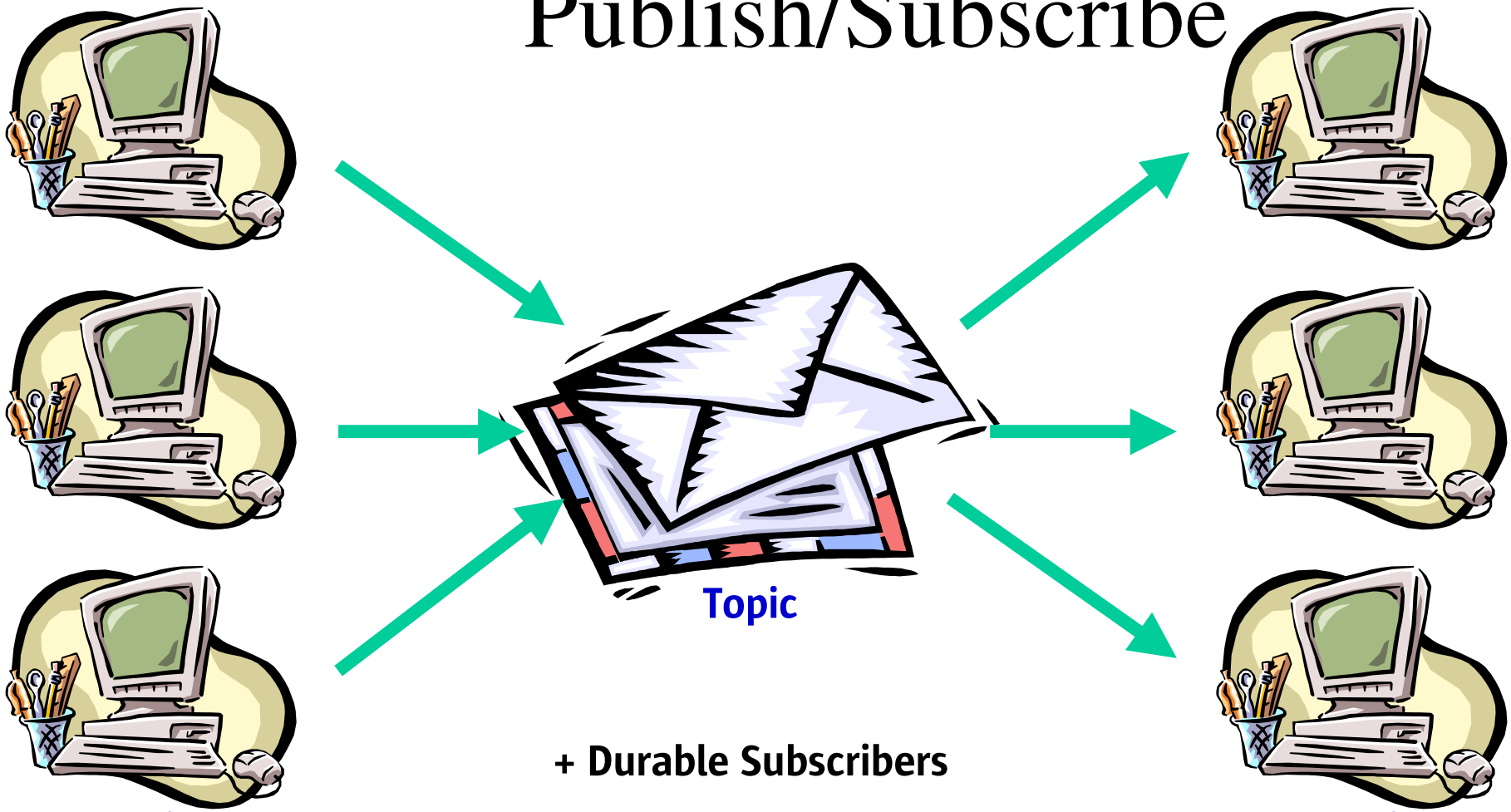
Point-to-Point



Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Publish/Subscribe

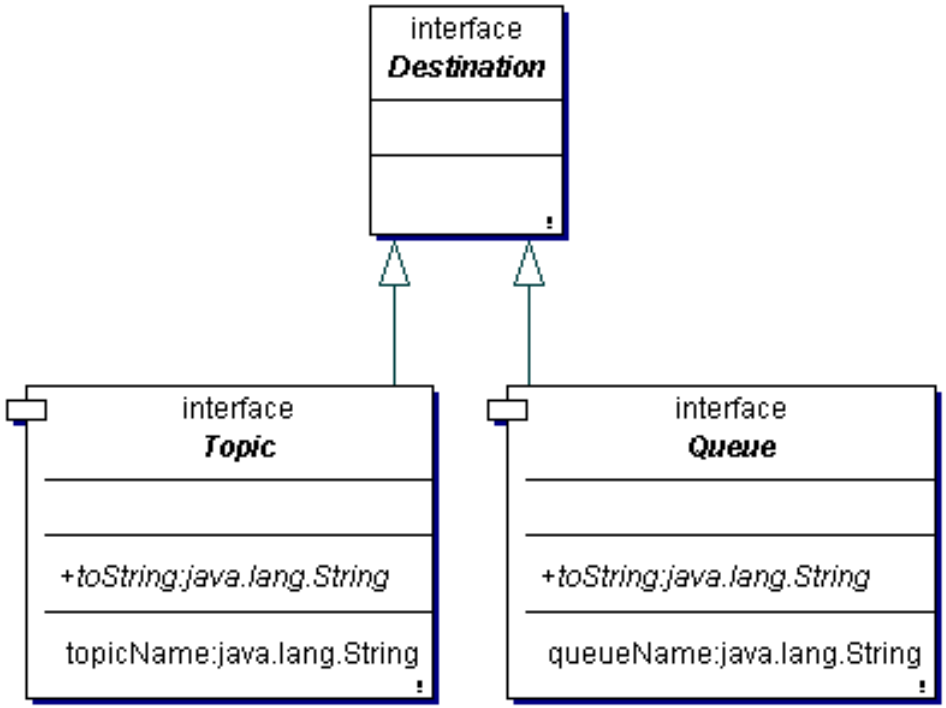


+ Durable Subscribers

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Destinations

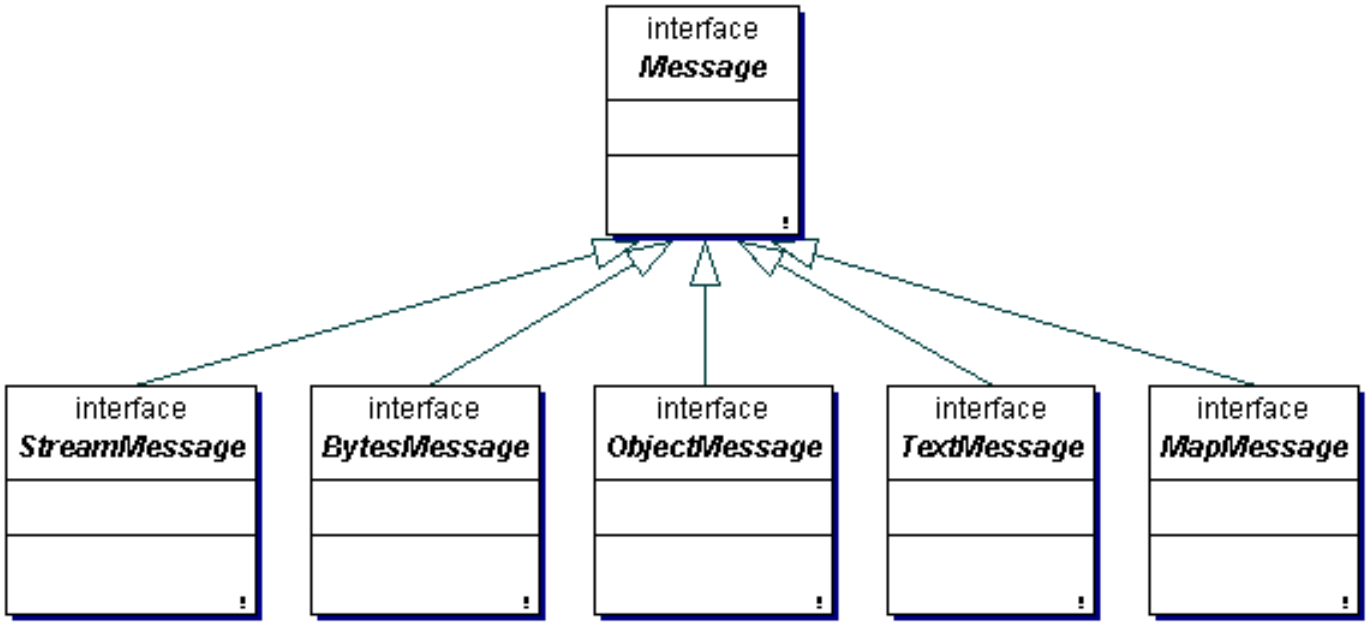


Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Message Types

1



+ Properties
+ Format Conversion

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Message Types 2

- TextMessage
  - java.lang.String (used for XML)
- MapMessage
  - name-value pairs
- StreamMessage
  - stream of primitive Java types
- ObjectMessage
  - serializable Java object
- ByteMessage
  - stream of uniterpreted bytes

Sun microsystems

Anatomy of an JMS Message

A JMS message consists of

header

default properties used for routing and identification purposes

set of properties

application specific properties

provider specific properties

body

the message: text, byte, map, stream, object

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

EJBs on One Slide

- Application Server - EJB Container
- Security - TX Mgmt. - Thread Allocation
- Java Code - Deployment Descriptor(s)
- Packaging - Deployment

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

JMS & J2EE

EJB Spec 1.1

NO asynchronous JMS receiver

EJB Spec 2.0

Message Driven Beans as asynchronous receiver

Message Driven Beans

onMessage() Method

can be called by sending a message to it only

Container acknowledges the message (AUTO)

Parameters are set in the deployment descriptor

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

MDB Characteristics

- Asynch Message Consumer
- Currently JMS-Based...
- Stateless
- Anonymous to Client
- No Home or Component Interface
- Container-Supplied Bean Context

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

MDB

```
public class MessageTraderBean implements MessageDrivenBean, MessageListener {
  private MessageDrivenContext ctx;

  public void ejbCreate() {}
  public void ejbRemove() {}

  public void setMessageDrivenContext(MessageDrivenContext ctx) {
    this.ctx = ctx;
  }

  public void onMessage(Message msg) {
    try {
      String msgText = ((TextMessage) msg).getText();
      System.out.println("Message Received: "+ msgText );
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

EJB Deployment Descriptor

```
<ejb-jar>
  <display-name>HelloMDB</display-name>
  <enterprise-beans>
    <message-driven>
      <display-name>HelloMDB</display-name>
      <ejb-name>HelloMDB</ejb-name>
      <ejb-class>com.sun.jmsworkshop.ex1.HelloMDB</ejb-class>
      <transaction-type>Container</transaction-type>
      <message-driven-destination>
        <destination-type>javax.jms.Queue</destination-type>
      </message-driven-destination>
      <security-identity>
        <run-as>
          <role-name>Sepp</role-name>
        </run-as>
      </security-identity>
    </message-driven>
  </enterprise-beans>
  <assembly-descriptor>
    <container-transaction>
      <method>
        <ejb-name>HelloMDB</ejb-name>
        <method-name>onMessage</method-name>
      </method>
      <trans-attribute>Required</trans-attribute>
    </container-transaction>
  </assembly-descriptor>
</ejb-jar>
```

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

WLS Deployment Descriptor

```
<weblogic-ejb-jar>
  <weblogic-enterprise-bean>
    <ejb-name>HelloMDB</ejb-name>
    <jndi-name>HelloMDB</jndi-name>
    <message-driven-descriptor>
      <destination-jndi-name>ex1Queue</destination-jndi-name>
      <connection-factory-jndi-name>QueueConnectionFactory</connection-factory-jndi-name>
    </message-driven-descriptor>
  </weblogic-enterprise-bean>
</weblogic-ejb-jar>
```

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

NET.OBJECTDAYS 2002



End of Part II – JMS Fundamentals

Marian.Kuffner@sun.com

Gerald.Loeffler@sun.com

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems

We make the net work.

Sun microsystems

Exercise Session I

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Exercise Session I

- Installation of JDK and J2EE 1.3 compliant application server
- Configuration of JMS Destinations
- Hello JMS World
- Hello MDB World

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Part III
XML Fundamentals

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

XML Fundamentals

- XML and Java
- JAXP Overview
- Parsing Technologies (SAX, DOM)
- Transformations (XSLT)
- Best Practices and Pitfalls

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

What is JAXP ?

- JAXP = **J**ava **A**PI for **X**ML **P**rocessing
  - Was done under the Java Community Process℠ initiative, JSR-63
  - Supports processing of XML documents using DOM, SAX and XSLT

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

What is JAXP ?

- A lightweight API for parsing XML documents
  - Allows for pluggable parsers
  - API for transformation, referred to as TrAX
  - Allows for pluggable XSLT engines
  - Allows processing of XML using:
    - Callback Driven Parser (SAX)
    - Tree Based Parser (DOM)
    - XSL Transformation

Sun microsystems

Introduction to JAXP 1.2 API 1

- Feature list JAXP 1.2
  - Support for W3C XML Schemas
  - SAX 2.0 incl. SAX 2 extensions
  - DOM Level 2 core interfaces
  - XSLT 1.0
  - XSLTC
  - Performance improvements
  - Changed deployment packaging !!!

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

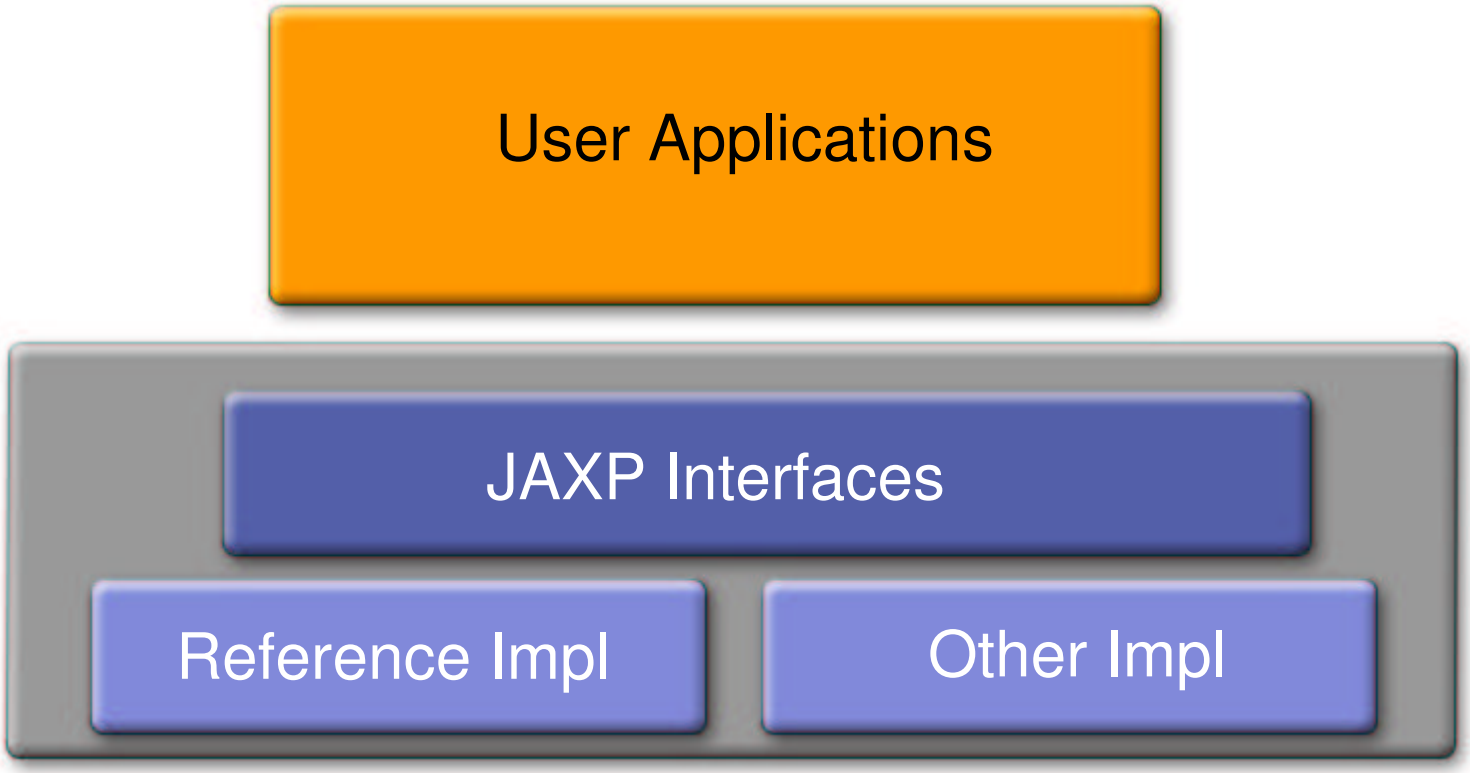
Introduction to JAXP 1.2 API 2

- Production Quality RI
  - Apache community
  - Xerces-2 and Xalan-2
  - Xalan XSLTC

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Architecture Overview



Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

What is JAXP ?

- Different Implementations
- Parsers
  - Crimson
  - Xerces
  - Xerces 2
- XSLT
  - Xalan

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

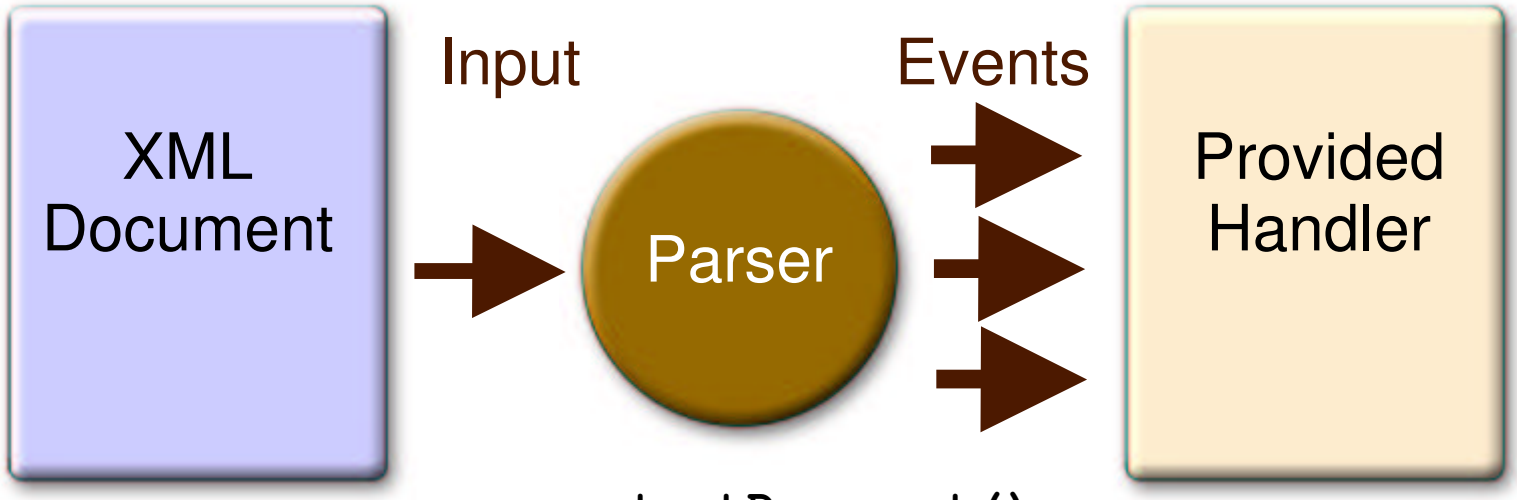
JAXP – SAX

- SAX stands for Simple API for XML
  - An event-driven API
  - The parser tells you when certain things happen, then it's up to you to do something about it
  - Fast and Lightweight
  - Harder to Program (state-machine)
  - Sequential Access Only
  - **`org.xml.sax.*`**

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JAXP – SAX



- `startDocument()`
- `endDocument()`
- `startElement()`
- `endElement()`
- `characters()`

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JAXP – SAX


```
<?xml version="1.0"?>
<root_element>
  <child_element>a child element</child_element>
  some text
</root_element>
```

Sequential SAX Events

```
startDocument()
  startElement():      root_element
    startElement():    child_element
      characters():    "a child element"
    endElement():      child_element
    characters():      "some text"
  endElement():        root_element
endDocument()
```

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JAXP – SAX

- Best Practices - use SAX when:
  - **You only need to go through a document once**
  - **You're only looking for a few items**
  - **You're not too concerned with the document structure or the context of the elements you're looking for**
  - **You don't have much memory to work with**

JAXP – SAX

Best Practices (cont.)

- **Reusing and pooling**
  - **Reuse parser and reasign handlers for processing different documents successively**
- **Partial parsing with SAX**
  - **If the information that should be extracted is at the beginning of the document (e.g. Envelope), end the parsing process by throwing an exception:**
  - **throw new EndOfProcessingException("Done.");**

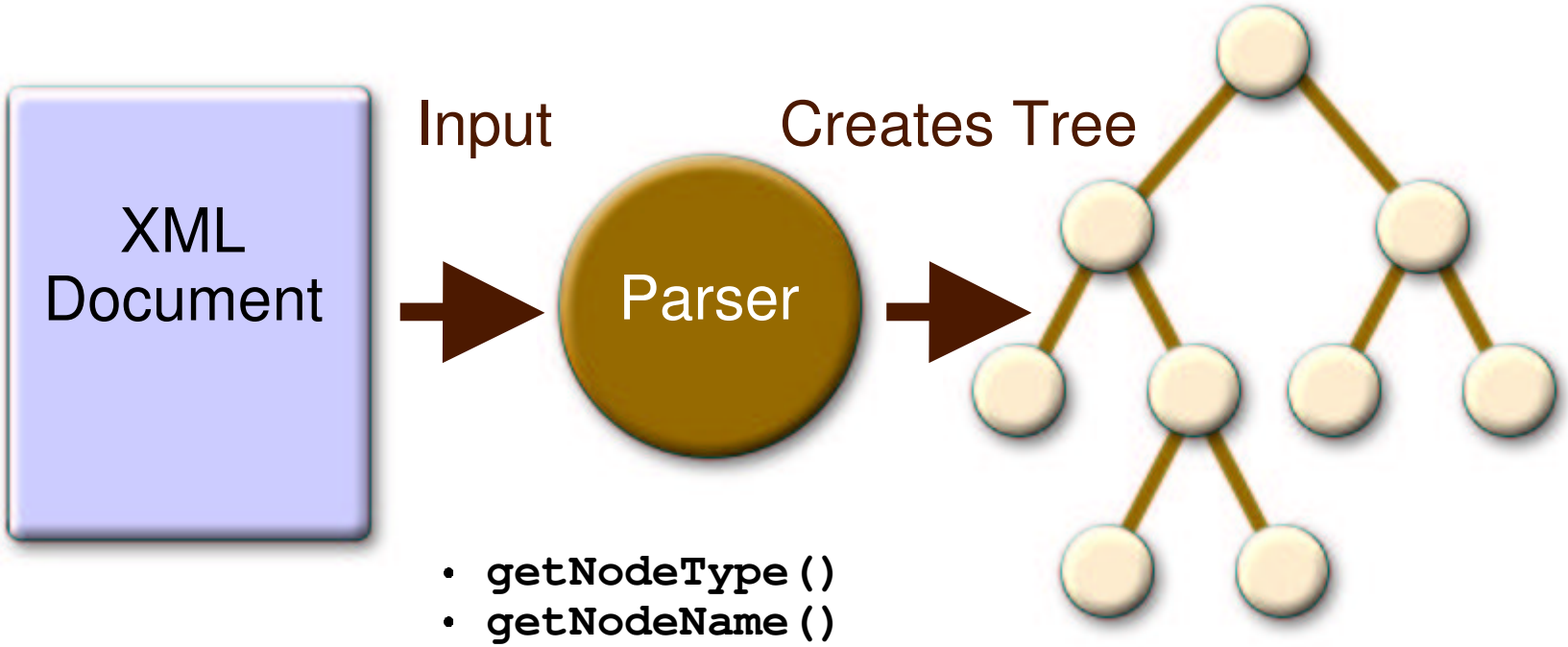
Sun microsystems

JAXP – DOM

- DOM = Document Object Model
  - Access XML document via a tree structure
  - Composed of element nodes and text nodes
  - Can “walk” the tree back and forth
  - Larger memory requirements
  - Designed to be portable from one parser implementation to another
  - **`org.w3c.dom.*`**

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

JAXP – DOM

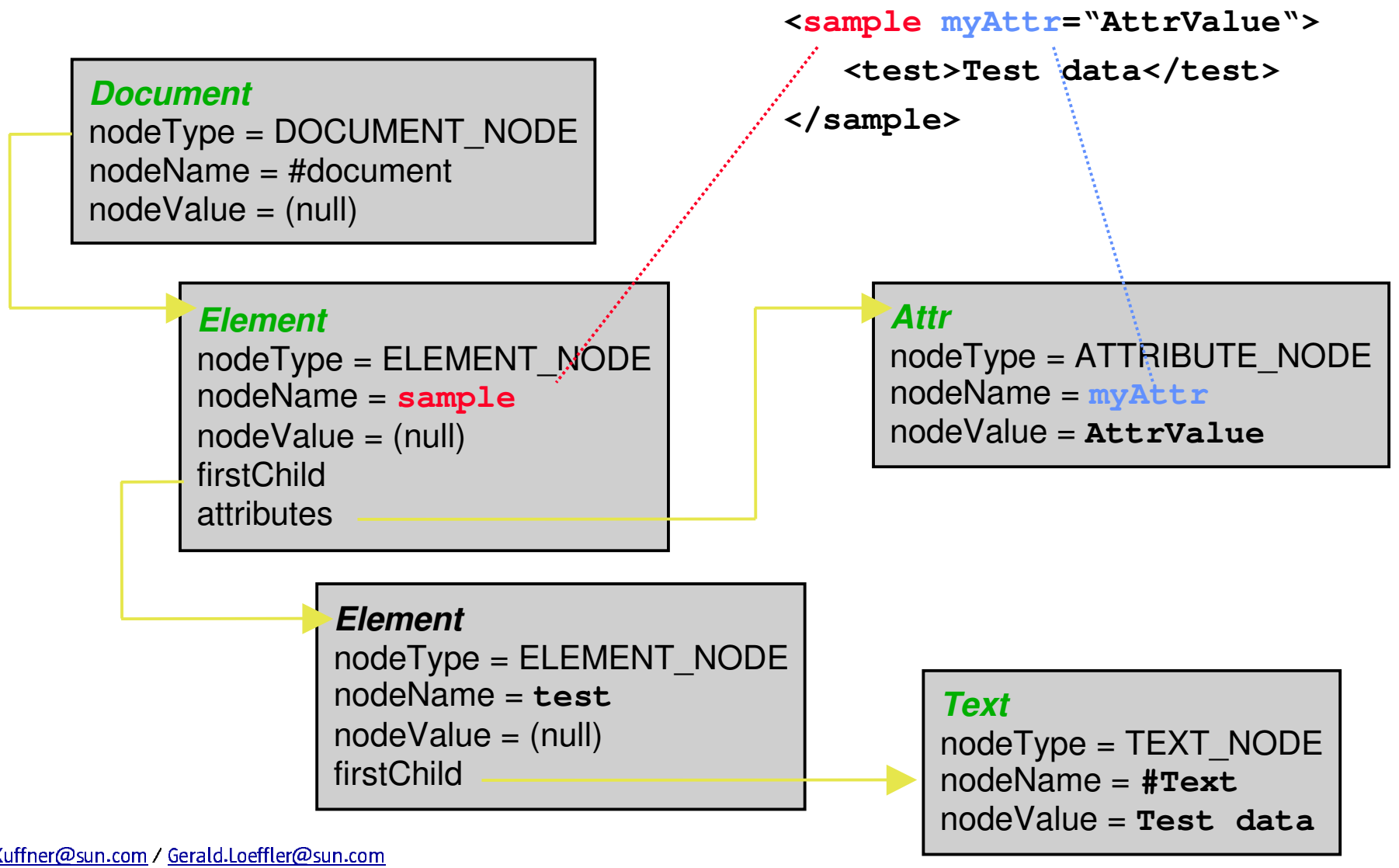


- `getNodeType()`
- `getNodeName()`
- `getNodeValue()`
- `getAttributes()`
- `getParentNode()`
- `getChildNodes()`

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JAXP – DOM



Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JAXP – DOM

- Best Practices – use DOM when:
  - **You need to manipulate lots of things in the document**
  - **The structure of the document and the context of the elements is a concern**
  - **Memory is not an issue**

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

JAXP – Parsing Best Practices

- Be aware of differences in the implementations
  - Don't use implementation specific functionality !
- Reduce Validation Cost
- Reduce the cost of referencing external entities

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Validation against Schemas 1

- http://java.sun.com/xml/jaxp/properties/schemaLanguage
  - This property specifies the schema language to be used to validate the document
  - Value must be a URI of the schema language specification being used
  - Must support the W3C XML Schema specification

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Validation against Schemas 2

- http://java.sun.com/xml/jaxp/properties/schemaSource
  - Defines the source from which the schema needs to be read
  - Must accept values
    - String that points to the URI of the schema
    - InputStream with the content of the schema
    - SAX InputSource
    - File object
    - An array of Objects with the content being one of the types defined above

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

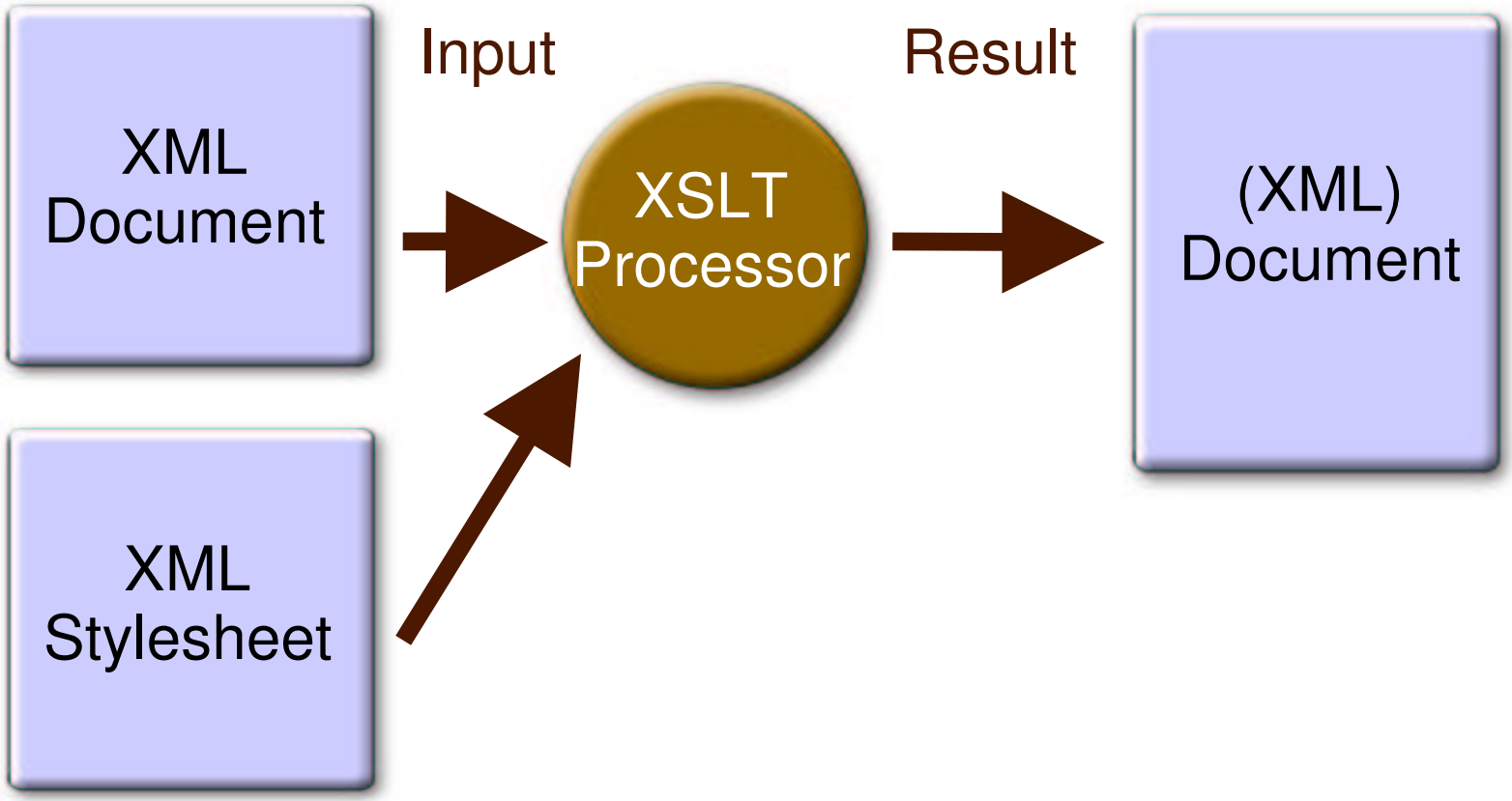
JAXP – XSLT

- XSLT = Extensible Stylesheet Language for Transformation
  - Syntax and semantics for transforming xml documents into other xml documents or any other format
  - Expressed as a well-formed xml document
  - Recommended processing instruction to apply to an xml document
  - **`javax.xml.transform.*`**

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JAXP – XSLT



Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JAXP – XSLT


```
<?xml version="1.0"?>
<people>
  <person born="1912" died="1954">
    <name>
      <first_name>Alan</first_name>
      <last_name>Turing</last_name>
    </name>
    <profession>mathematician</profession>
  </person>
  <person born="1918" died="1988">
    <name>
      <first_name>Richard</first_name>
      <last_name>Feynman</last_name>
    </name>
    <profession>physicist</profession>
  </person>
</people>
```

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="person">
A Person
</xsl:template>
</xsl:stylesheet>
```

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

Marian.Kuffner@

JAXP – XSLT

- Best Practices - use XSLT when:
  - **You need to perform the same transformation many times**
  - **Do XSLT-Processor reusing and pooling to improve transformation performance**

Sun microsystems

What is inside JAXP 1.2 RI ? 1

- jaxp-api.jar
  - The javax.xml.parsers and javax.xml.transform components of JAXP. These packages contain the APIs that give applications a consistent way to obtain instances of XML processing implementations.
- sax.jar
  - The APIs and helper classes for the Simple API for XML (SAX), used for serial access to XML data.
- dom.jar
  - The APIs and helper classes for the Document Object Model (DOM), used to create an in-memory tree structure from the XML data.

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

What is inside JAXP 1.2 RI ? 2

- xercesImpl.jar
  - The implementation classes for the SAX and DOM parsers, as well as Xerces-specific implementations of the JAXP APIs.
- xalan.jar
  - The "classic" Xalan XSLT processor.
- xsltc.jar
  - The Xalan Compiling XSLT processor.

Sun microsystems

JAXP – Take care of these:

- Be aware of the different versions of JAXP and JAXP-implementations
- Follow the link below, on how to correctly setup your environment (JDK 1.4, JWDSP, Tomcat, ...)
  - **http://xml.apache.org/~edwingo/jaxp-ri-1.2.0-fcs/docs/ReleaseNotes.html**
- Latest notes can be found in the FAQ (see References)

Sun microsystems

JAXP – References

- Part of the Java XML Pack
  - http://java.sun.com/xml/javaxmlpack.html
- JAXP
  - http://java.sun.com/xml/jaxp/index.html
- Feedback aliases
  - jsr63-comments@eng.sun.com
  - xml-feedback@java.sun.com

Sun microsystems

JAXP – References 2

- FAQ
  - http://java.sun.com/xml/jaxp/faq.html
- RI docs
  - http://xml.apache.org/~edwingo/jaxp-ri-1.2.0-fcs/docs/
- SAX
  - http://www.saxproject.org

NET.OBJECTDAYS 2002



End of Part III – XML Fundamentals

Marian.Kuffner@sun.com

Gerald.Loeffler@sun.com

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems

We make the net work.

Part IV
JMS Details

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JMS Details

- Provider Model
- Persistence
- Message Properties and Filters
- The Role of JNDI
- Synchronous vs. Asynchronous Message Consumers
- Queue Browser
- Message Driven Beans Subleties
- Acknowledgement, Recovery and Transaction
- Distributed Transactions

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Provider Model

- JMS comprises mostly Interfaces
- Program against JMS Interfaces only
- Factories return provider impls
- JMS defines API - not transport
- JMS is Java-only - but MOM product can be x-platform

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Decoupling Producer-Consumer

- In Space - distributed computing
- In Platform and Language
- In Time - asynchronous
- In Java Object Types (potentially)
- In TX-context

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Persistent vs. Non-Persistent

- Reliability in case of Provider failure
- Provider needs Persistent Store configured
- Tradeoff in performance (and space)
- Administered for Connection Factory
- Coded in MessageProducer (DeliveryMode)

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Message Types

- TextMessage: String
- BytesMessage: Unaltered, raw bytes
- ObjectMessage: Serializable (Java-only)
- MapMessage: String->primitive value
- StreamMessage: Ordered primitives
- Message: Empty Message (only props)

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Message Properties

- Name-value pairs
- Confirm availability outside JMS!
- Filterable
- Message Headers vs. Optional Props
- Headers: Dedicated accessors in Message
- Optional Props: Type-safe accessors
- JMSX-prefix for names reserved
- Goodie: Type-conversion of values

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Message Properties 2

Header Field	Set by	Default
JMSDestination	Client, for each Message Producer	
JMSDeliveryMode	Client, for each Message Producer	Persistent
JMSExpiration	Client, for each MessageProducer	0 (no expiration)
JMSPriority	Client, for each MessageProducer	4 (normal)
JMSMessageID	Provider, automatically	
JMSTimestamp	Provider, automatically	
JMSRedelivered	Provider, automatically	
JMSCorrelationID	Application, for each message	
JMSReplyTo	Application, for each message	
JMSType	Application, for each message	

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Message Headers

- Read-only (set-on-call):
  - JMSDestination, JMSDeliveryMode, JMSExpiration, JMSPriority, JMSMessageID (disable?), JMSTimeStamp (disable?), JMSRedelivered
- Writable:
  - JMSCorrelationID (-> JMSMessageID), JMSReplyTo, JMSType (set!),

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JNDI

- JNDI used to specify JMS Provider
  - Hashtable env = new Hashtable();
    env.put(Context.INITIAL_CONTEXT_FACTORY,
    "weblogic.jndi.WLInitialContextFactory");
    env.put(Context.PROVIDER_URL, url);
    InitialContext ctx = new InitialContext(env);
    ctx.lookup("conn factory or destination name");
- Multiple Providers possible

Sun microsystems

JMS Factories

- ConnectionFactory -> Connection
  - QueueConnectionFactory, TopicConnectionFactory
- Connection -> Session
  - QueueConnection, TopicConnection
- Session -> Msg Producer / Consumer
  - QueueSession, TopicSession
- Session -> Message

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

JMS Factories

Parent	Point-to-Point	Pub/Sub
ConnectionFactory	QueueConnectionFactory	TopicConnectionFactory
Connection	QueueConnection	TopicConnection
Session	QueueSession	TopicSession
MessageProducer	QueueSender	TopicPublisher
MessageConsumer	QueueReceiver	TopicSubscriber

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Sessions

- Only one Listener thread!
  - All calls to Session's Consumer's onMessage() are serialized
- Basis for TX
- Basis for Acknowledgement mode
- Provides Recovery via recover()

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Destinations

- JNDI-Lookup - don't create dynamically
- Ignore Session's factory methods!
- Temporary Destinations
  - Valid for Connection
  - Created via Session's factory methods
    - QueueSession, TopicSession
  - Request-Reply (but: Requestor helpers)

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Message Producers

- Created for Destination or "Unidentified"
- Default Delivery Properties administered in Connection Factory
- Set Delivery Properties (disabled?)
  - For all Messages sent through Producer
  - Delivery Mode, Expiration, Priority (TTL)
- Send Message
  - Possibly overriding Delivery Properties

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Message Consumers

- Created for Destination
- Need to start() Connection
- Handle Messages Synch or Asynch
  - But always delivered to client asynch!
- In practice, don't do blocking receive()
- Only one Listener per Consumer
- Don't share Listeners between Sessions!
- Messages received read-only

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Durable Topic Subscribers

- Bridge Topics to Queues
- Provider keeps Messages for un-connected Durable Subscribers
- Needs ClientID String for Connection
  - Admin and/or programmatically
- Special factory method in TopicSession
  - Requires subscription identifier unique within ClientID
- Needs Msg expiration of some sort
- A real resource hog - use with care!

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Multiple Queue Receivers

- Unspecified distribution semantics
  - First-come first-serve
  - Round--robin
  - Client-fail-over
  - Enforce one receiver

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Message Filters

- SQL where-clause using Properties
- Case-sensitive Property names
- Confirm availability outside JMS!
- Specify at Message Consumer creation
  - In QueueSession, TopicSession
- Hope for server-side filtering with Queues and fear client-side filtering with Topics

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

QueueBrowser

- Peeks at Messages in Queue
- Created via QueueSession like Consumer
- May apply Filters
- Undefined: static vs. dynamic image

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Requestors

- Implement Request/Reply
- QueueRequestor and TopicRequestor
  - The only conrete classes in JMS API
- Use Temporary Destinations and JMSReplyTo
- Blocking receive - limited usefulness
- Deadlock when used within one TX!

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

MDB Subtleties

- Acknowledgement and TX -> see later
- MDB instances not reentrant
  - onMessage()-calls serialised for each instance
- No seriality in order-processing whatsoever
- MDB listens on only one Destination
- No available Connection or Session
- No use in throwing RuntimeException from within onMessage() -> see TX discussion
- Don't use EJBContext.*Caller*()

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

NET.OBJECTDAYS 2002



End of Part IV – JMS Details

Marian.Kuffner@sun.com

Gerald.Loeffler@sun.com

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems

We make the net work.

Sun microsystems

Exercise Session II

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Exercise Session II

- Administering Connection Factories
- Discovering the Semantics of the Queue Browser Implementation
- Testing Simple JMS Transactions
- Distributed Transactions
- Message Acknowledgement vs. Session Recovery vs. Transaction Rollback

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Part V
Transporting XML via JMS

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

XML Messages

- Do you need it? - JMS is x-platform!
- (Un)Marshalling costs!
- Message size!
- Use TextMessage
- XML-Filters provided by proprietary XMLMessage

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

NET.OBJECTDAYS 2002



End of Part VI – Transporting XML via JMS

Marian.Kuffner@sun.com

Gerald.Loeffler@sun.com

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems

We make the net work.

Part VI
JMS Performance Summary

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Performance Tips 1

- Topics (if broadcast-based) faster and more scalable than Queues (if connection-based)
- SSL is connection-based (sockets) and outrules broadcasts
- Beware of Durable Topic Subscribers
- Turn off what you don't need:
  - Generation of Timestamps
  - Generation of Message IDs

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Performance Tips 2

- Be conscious about TX requirements
  - Use distributed TX only if you have to
  - Use JMS-local TX only if you have to
- Use DUPS_OK_ACKNOWLEDGE
  - ...and deal with duplicate messages
- Complete onMessage() quickly

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Performance Tips 3

- Beware of Message Selectors
  - Avoid them if evaluated on client (which often is the case for Topics (at least if broadcast-based))
- Increase parallelism by using more than one Session
  - ...and deal with out-of-order Messages and complicated TX handling
- Use XML Messages only where necessary
- Have a good DBA

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

NET.OBJECTDAYS 2002



End of Part VI – JMS Performance Summary

Marian.Kuffner@sun.com

Gerald.Loeffler@sun.com

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems
We make the net work.

Sun microsystems

Part VII
JMS Security

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Security Considerations 1

- Authentication for JNDI access
  - Security principal and security credentials supplied in constructor for InitialContext
- Access control for JNDI lookup on per-object basis
- Other means to get Administered Objects, bypassing JNDI access control
  - API to create Connection Factories and Destinations
  - Re-use objects from other installation

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Security Considerations 2

- Access control on per-Destination basis
  - Browse / send / receive permissions granted to individual user or groups
- Message encryption to prevent wire sniffing
  - By all-Java application using ObjectMessage and Java Cryptography Extensions (body only)
  - By JMS Provider via SSL, HTTPS, T3S, etc.
  - Messaging inside VPN

Security Considerations 3

- Signing Messages to prevent Message Faking
  - Ensures that Message was sent by who claims to have it sent
  - Message is left unaltered, with secure hash appended

NET.OBJECTDAYS 2002



End of Part VII – JMS Security

Marian.Kuffner@sun.com

Gerald.Loeffler@sun.com

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems

We make the net work.

Sun microsystems

Exercise Session III

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Exercise Session III

- Designing Common JMS Architecture Patterns
  - job scheduling
  - Backend connectivity
- Designing an Enterprise JMS Architecture

Marian.Kuffner@sun.com / Gerald.Loeffler@sun.com

Sun microsystems

Resources

- http://java.sun.com/products/jms
- "Java Message Service Specification, Version 1.0.2b", Sun Microsystems
- "Java Message Service API Tutorial", Kim Haase, Sun Microsystems
- "Enterprise JMS Programming", Shaun Terry, M&T Books

Sun microsystems

Resources

- "JMS – API Overview"
  Article in Java Magazin
  http://www.javamagazin.de – 05/2002
- iMQ iPlanet Message Queue for Java 2.0
  http://www.iplanet.com/products/java_message/
- Java Naming and Directory Interface
  http://java.sun.com/products/jndi/
- Java 2 Enterprise Edition
  http://java.sun.com/products/j2ee/



Marian.Kuffner@sun.com

Gerald.Loeffler@sun.com

http://www.sun.com/service/sunps/jdc/index.html

JAVA™ CENTER

Sun microsystems
We make the net work.