JAVA DAYS ´99



Java Transaction API (JTA) and Java Transaction Service (JTS)

Ralph Galantine
Senior Product Manager
Sun Microsystems

Sun microsystems

Agenda

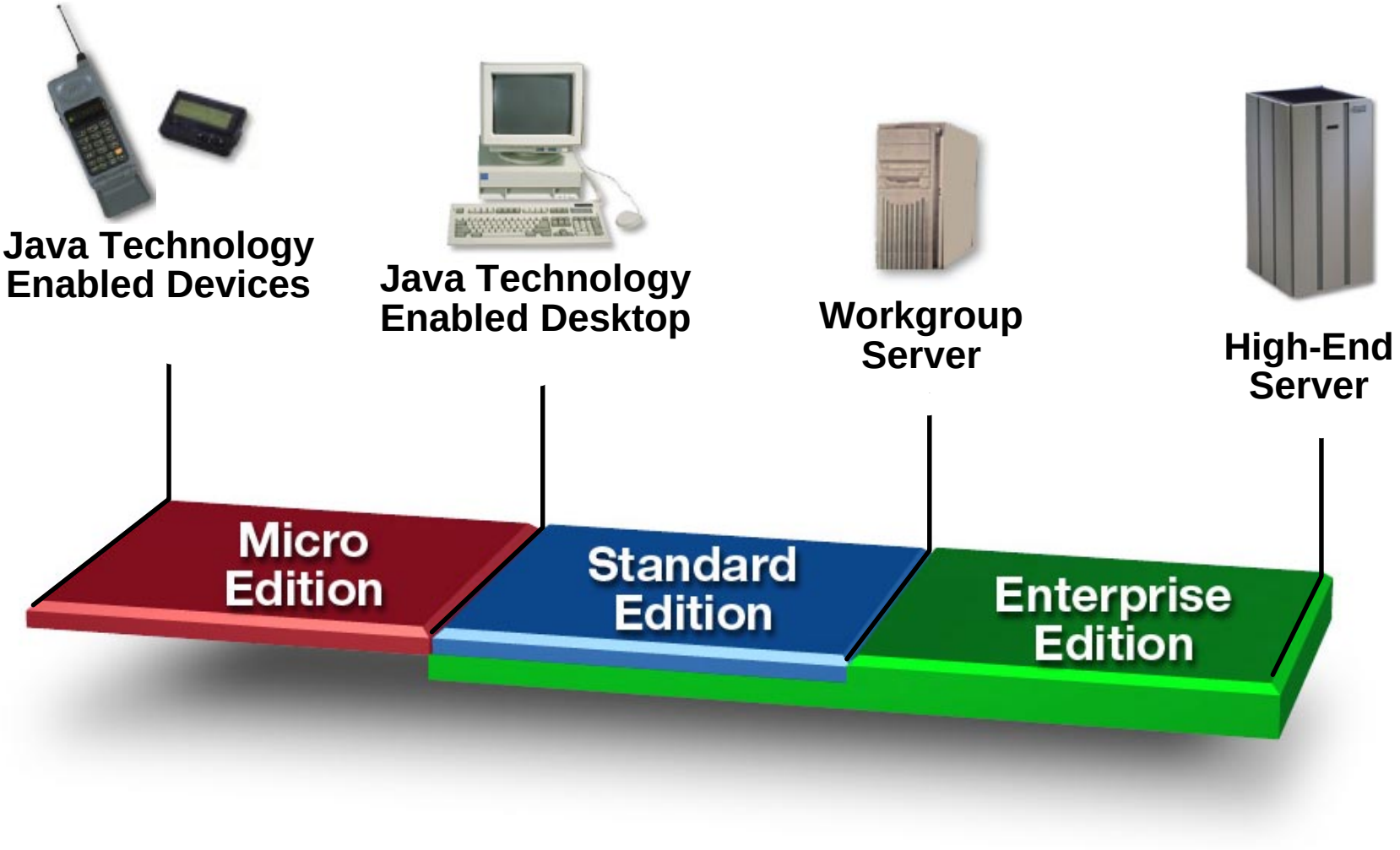


- J2EE and Transactions
- Java Transaction API
- Java Transaction Service
- Transaction Support in Java 2 Enterprise Edition Reference Implementation
- Questions

© 1999 Sun Microsystems GmbH

Sun microsystems

Java 2 Platform Devices



© 1999 Sun Microsystems GmbH

Java Platform



Where Have We Been, Where Are We Going

Year 1

- JDK All Things to Everyone

Year 2

- JDK "All Things to the Enterprise"
- Early Development of Consumer Java
- Early Development of "Enterprise" APIs"

Year 3

- Micro Edition
- Standard Edition
- Enterprise Edition



© 1999 Sun Microsystems GmbH

Sun microsystems

Java® 2 Enterprise Edition

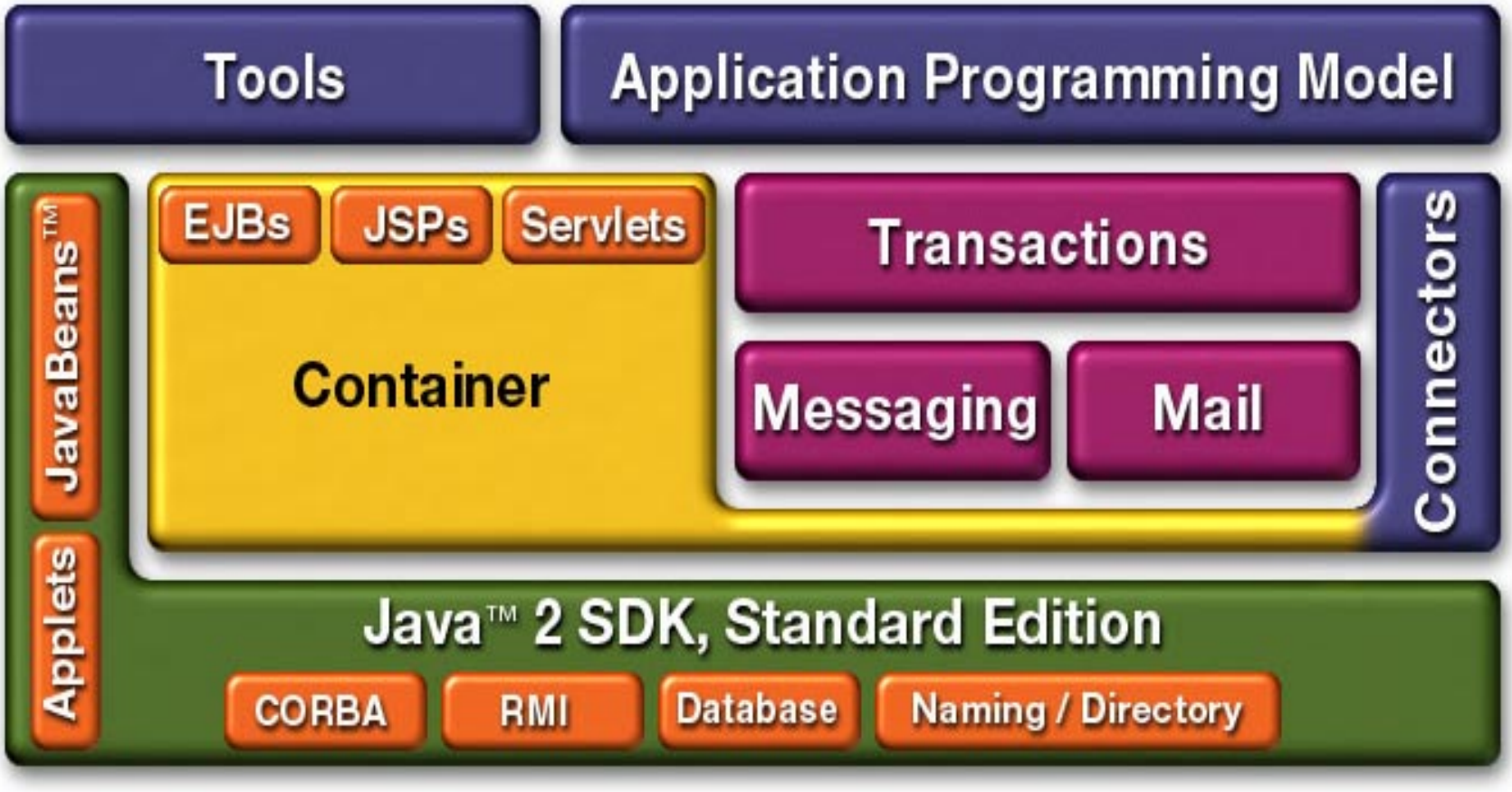


Database	Objects	Mail
Transac-tions	Messaging	Directory



© 1999 Sun Microsystems GmbH

J2EE Architecture



© 1999 Sun Microsystems GmbH

Enterprise JavaBeans and Transactions



- Container managed transaction demarcation

  Transaction attributes defined in the deployment descriptor.

- Bean managed transaction demarcation

  Use javax.transaction.UserTransaction interface to demarcate transactions.

In both cases, the container handles the burden of implementing transaction management

© 1999 Sun Microsystems GmbH

Sun microsystems

Transaction attributes for methods



- Not Supported
  - Suspends current transaction, if one is in progress
- Required
  - starts new transaction, if one is not in progress
- Supports
  - executes whether a transaction is in progress or not

© 1999 Sun Microsystems GmbH

Sun microsystems

Transaction attributes for methods (continued)



- Requires new
  - suspends transaction, if one is in progress
  - starts new transaction
- Mandatory
  - Throws an exception, if no transaction is in progress
- Never
  - Throws an exception, if a transaction is in progress

© 1999 Sun Microsystems GmbH

Sun microsystems

Relationship between JTA and JTS



- JTA
  - API for transaction management
  - Can be implemented by any transaction manager
- JTS
  - An implementation of a transaction manager
  - Java mapping of the OTS 1.1 specification
  - Supports JTA interfaces

© 1999 Sun Microsystems GmbH

Sun microsystems

Java Transaction API Overview



- Specifies interfaces for transaction demarcation, resource enlistment and synchronization
- A layer on top of an existing transaction manager (e.g., Java Transaction Service)
- Transaction manager is responsible for transaction propagation
- Support for nested transactions is not required

© 1999 Sun Microsystems GmbH

Sun microsystems

Java Transaction API Interfaces



- UserTransaction
- TransactionManager
- Transaction
- XAResource
- Synchronization

© 1999 Sun Microsystems GmbH

Sun microsystems

UserTransaction interface



- Enables application components to begin, commit and abort transactions
- Transaction to thread association is maintained by the transaction manager
- Used by EJB components with the TX_BEAN_MANAGED attribute; can also be used by transactional clients

© 1999 Sun Microsystems GmbH

TransactionManager interface



- Enables application servers to demarcate transaction boundaries
- *getTransaction()* method returns the transaction object
- *suspend()* method disassociates the transaction from the calling thread
- *resume()* method re-associates the transaction with the calling thread

© 1999 Sun Microsystems GmbH

Sun microsystems

Transaction interface



- Enables the following operations on the transaction object:
  - resource enlistment
  - transaction synchronization
  - commit or abort the transaction
- Thread independent

© 1999 Sun Microsystems GmbH

Sun microsystems

XAResource interface

- Java mapping of the X/Open's XA interface
- Defines the contract between the transaction manager and the resource manager
- Enables resource managers to participate in global transactions and recovery after a crash

© 1999 Sun Microsystems GmbH

XAResource interface

- Transaction manager uses the 2PC protocol to *prepare*, *commit* or *rollback* transactions

© 1999 Sun Microsystems GmbH

Synchronization interface



- Enables application servers to get notifications from the transaction manager *before* and *after* the transaction completion
- *beforeCompletion()* method is invoked with the same transaction context
- *afterCompletion()* method is invoked after the completion of the transaction

© 1999 Sun Microsystems GmbH

Sun microsystems

Java Transaction Service Overview



- A transaction manager which supports JTA interfaces at the high-level and implements the Java mapping of the OMG's OTS 1.1 specification at the low-level

- Uses IIOP for transactional interoperability

© 1999 Sun Microsystems GmbH

Sun microsystems

Java Transaction Service



- Provides for transaction demarcation
- Resource enlistment
- Transaction synchronization
- Implicit transaction propagation
- Interoperates with other TMs through IIOP
- Supports CORBA's OTS interfaces

© 1999 Sun Microsystems GmbH

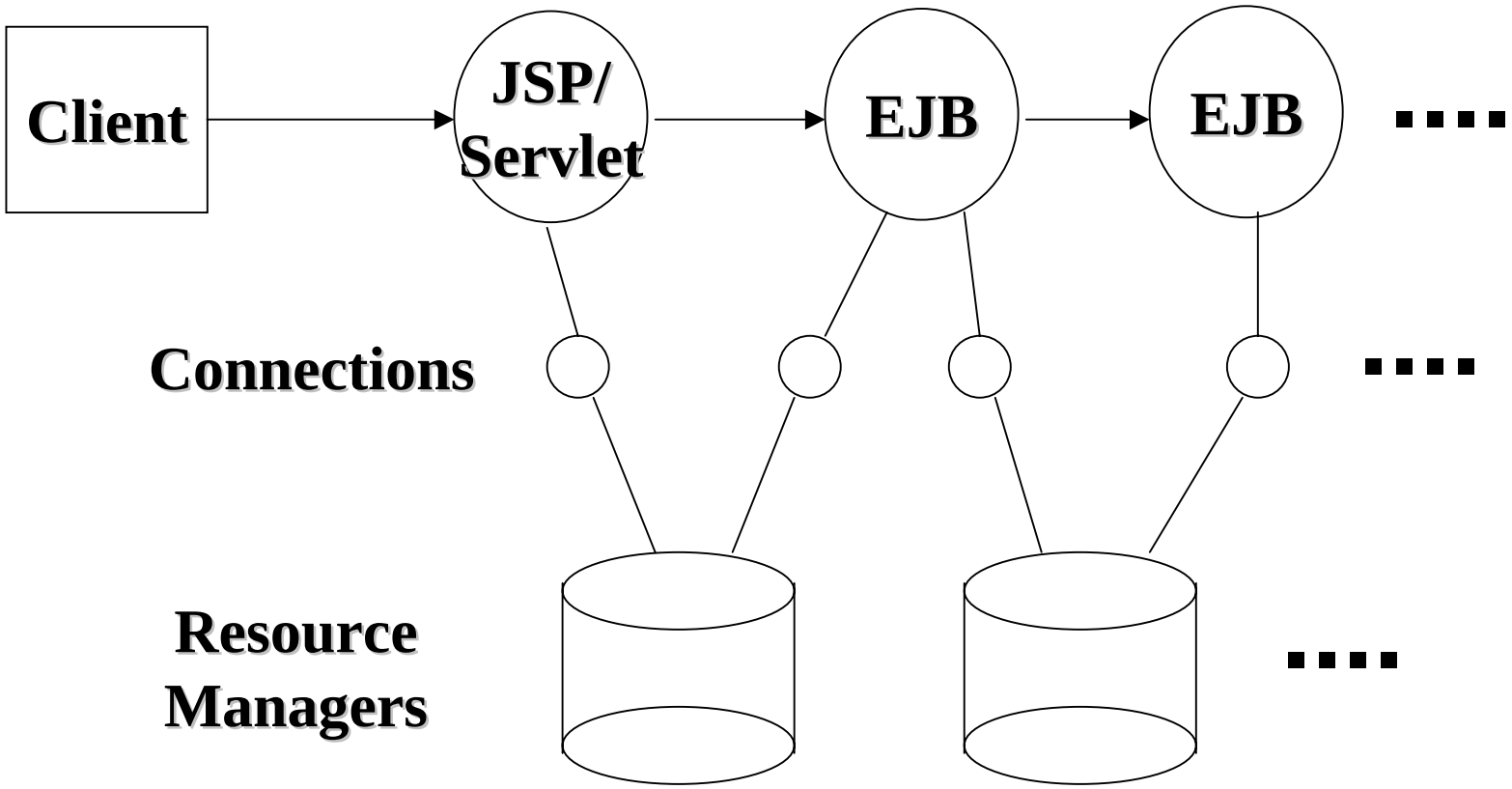
Sun microsystems

Optional JTS functionality

- Support for nested transactions
- Support for native XA resources
- Support for OTS resources

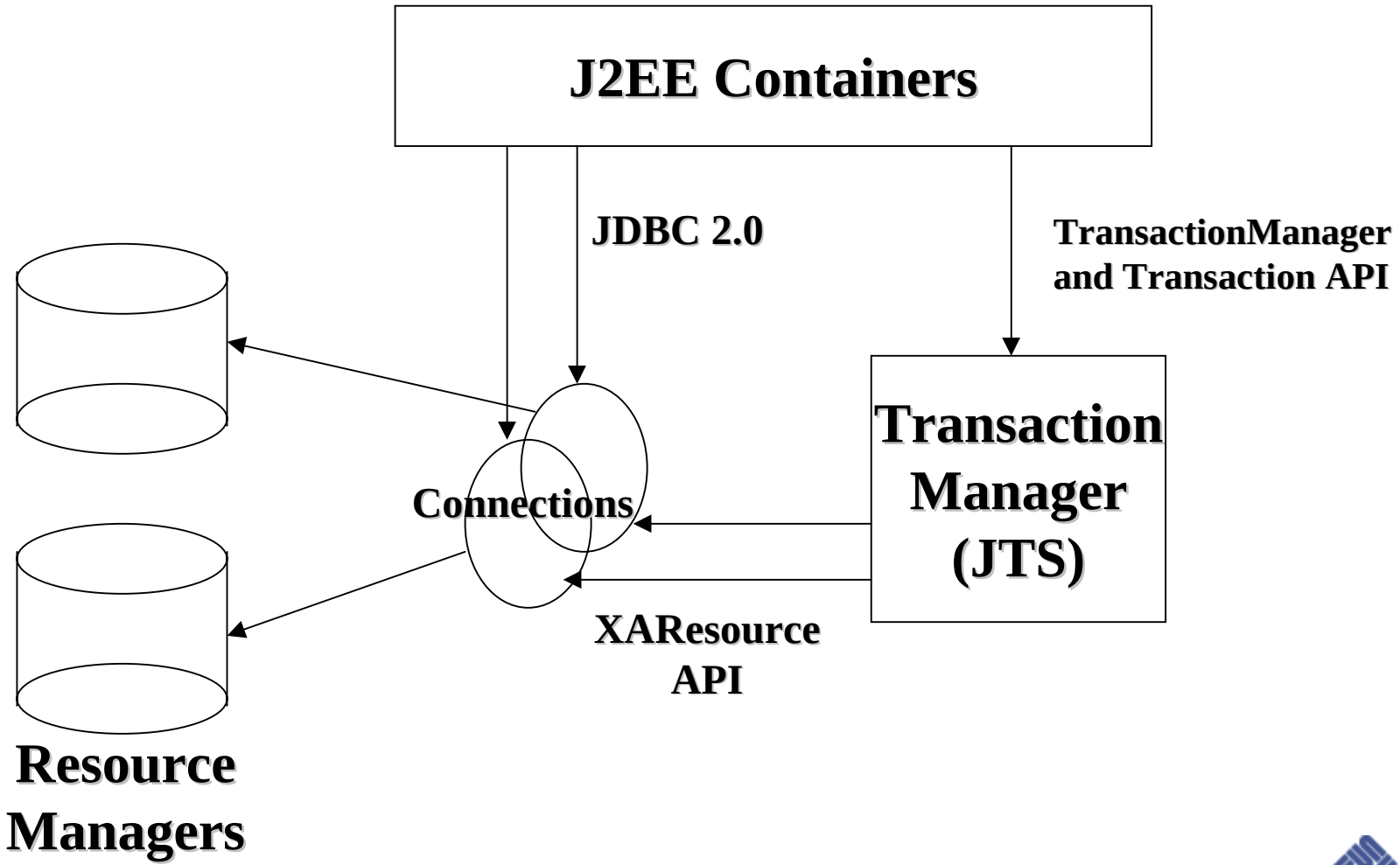
© 1999 Sun Microsystems GmbH

J2EE Requirement on Transactions



© 1999 Sun Microsystems GmbH

J2EE RI and JTS Integration



© 1999 Sun Microsystems GmbH

Transaction Manager

- An implementation of JTS
- Written in Java
- Implements JTA and CORBA OTS
- Supports
  - transaction context propagation (via IIOP)
  - two phase commit
  - JTA XAResources

© 1999 Sun Microsystems GmbH

Container and JTA

- Uses *TransactionManager* interface to demarcate transaction boundaries
- Uses *Transaction* interface to enlist and delist resources
- Obtains *XAResource* objects from resource adapters and passes to TM
- Provides *UserTransaction* implementation to components

© 1999 Sun Microsystems GmbH

EJB Code Sample

```
Public myBusinessMethod() {
    DataSource ds1 = ...  // get DataSource by JNDI
    DataSource ds2 = ...  // get DataSource by JNDI
    Connection con1 = ds1.getConnection();
    Connection con2 = ds2.getConnection();
    // do work on connection 1
    // do work on connection 2
}
```


© 1999 Sun Microsystems GmbH

What Container does Underneath



J2EE App	Container
Public myBuisnessMethod() { …	tm.begin(); Transaction t = tm.getTransaction();
Connection con1 = ds1.getConnection();	t.enlistResource(xares1);

Connection con2 =
ds2.getConnection();
// do work on connection 1 and 2
	t.enlistResource(xares2);

}	t.delistResource(xares1, flags);
	t.delistResource(xares2, flags);
	tm.commit();

© 1999 Sun Microsystems GmbH

Sun microsystems

Status

- Supports JDBC 1.0 and 2.0 drivers
- Transaction propagation across components, processes and machines
- Multiple, heterogeneous databases support with JDBC 2.0 extension drivers

© 1999 Sun Microsystems GmbH

Resources



- http://java.sun.com/j2ee

© 1999 Sun Microsystems GmbH

Sun microsystems


Sun
microsystems