



IBM Software Group

IMS17

IMS Java Application Development using JDBC, including Remote Database Services

Christopher Holtz

IMS Development

Silicon Valley Laboratory

IBM Corporation





Overview

- **IMS Java**
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- **What is JDBC?**
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- **Environments**
 - Non-managed
 - IMS
 - CICS
 - DB2
 - Managed
 - WebSphere
- **Details of WebSphere Application Server**
 - DataSource
 - Enterprise Archive
- **Remote Database Services**





What is IMS Java?

- A new feature in IMS v7
- A set of classes that...
 - Offers Java support to access IMS Databases from various environments
 - IMS (in the Java Dependent Regions)
 - DB2 Stored Procedures
 - CICS, using the JCICS API
 - WebSphere Application Server
 - Offers Java support to access IMS message queue and transaction services
 - Enables SQL access through the JDBC interface

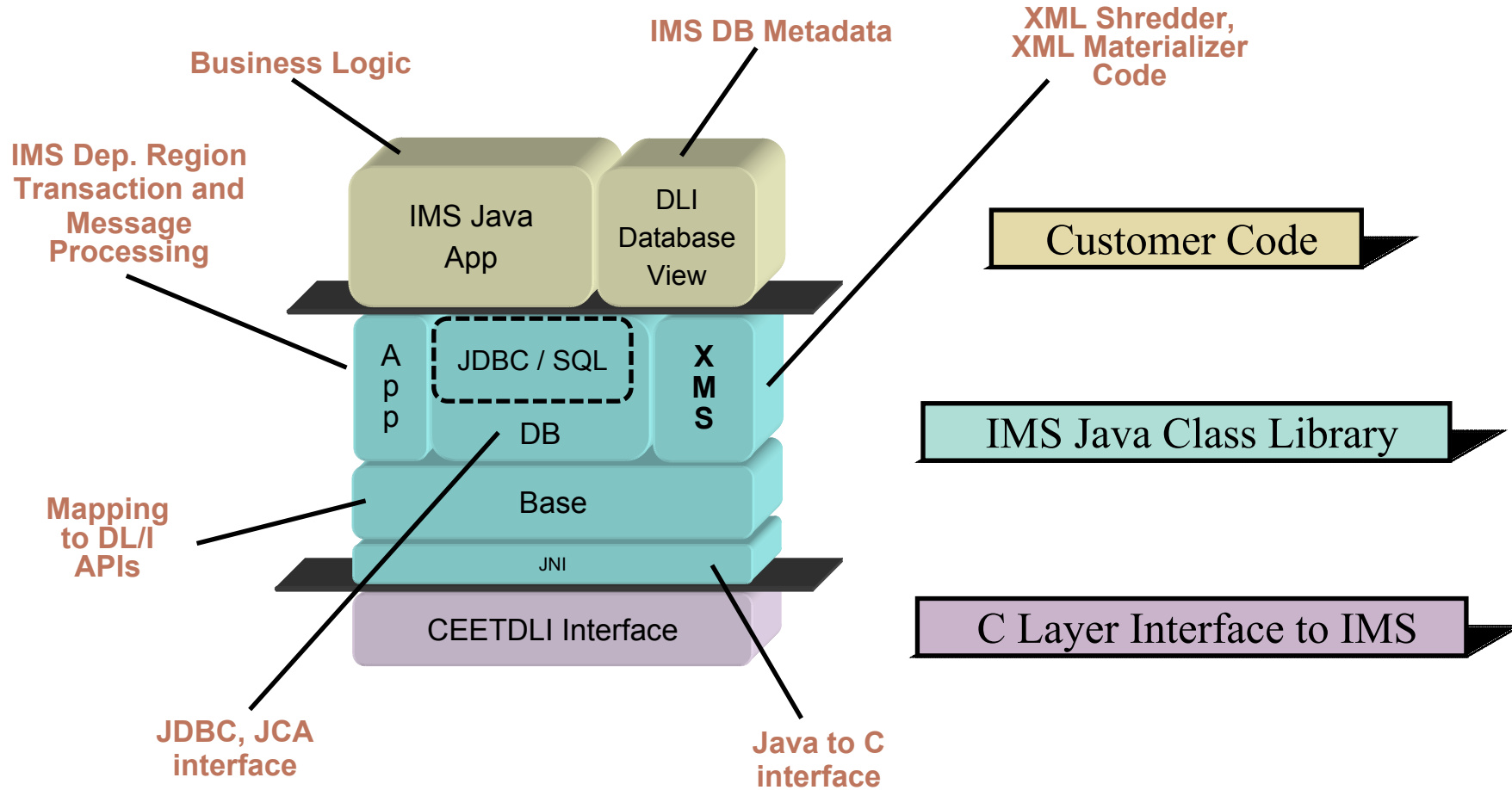




Why IMS Java?

- Rapid Application Development
- Leverage existing marketplace, industry-sanctioned standards - they are the slowest changing and most persistent
 - JDBC and J2EE are standards – help to minimize specific back end knowledge of IMS
- Leverage new and abundant skills in the marketplace and mitigate the loss of 390 skills for customers
- Integrate with other products
 - ☆ Our response is IMS Java, IMS XML-DB, Remote Database Services (RDS), WebSphere support, CICS support, DB2 SP support

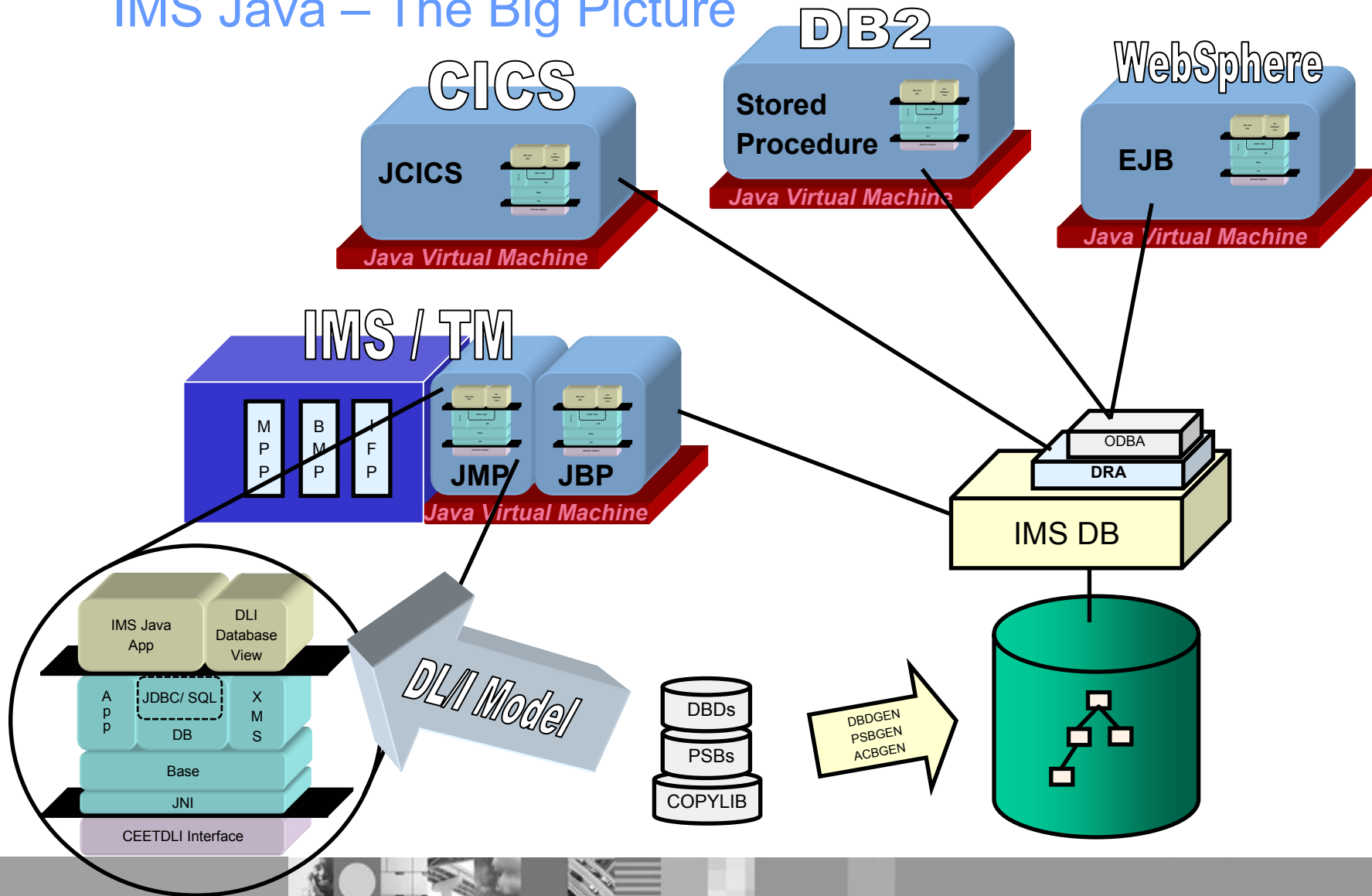
IMS Java Class Library





the world depends on it

IMS Java – The Big Picture





Overview

- **IMS Java**
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- **What is JDBC?**
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- **Environments**
 - Non-managed
 - IMS
 - CICS
 - DB2
 - Managed
 - WebSphere
- **Details of WebSphere Application Server**
 - DataSource
 - Enterprise Archive
- **Remote Database Services**





JDBC Explained

- Defines a standard Java API for accessing relational databases
- Provides an API for sending SQL statements to a database and processing the tabular data returned
- Executing JDBC query statements
 - Establish and open connection to database
 - Execute query and obtain results
 - Process results
 - Commit the results
 - Close connection

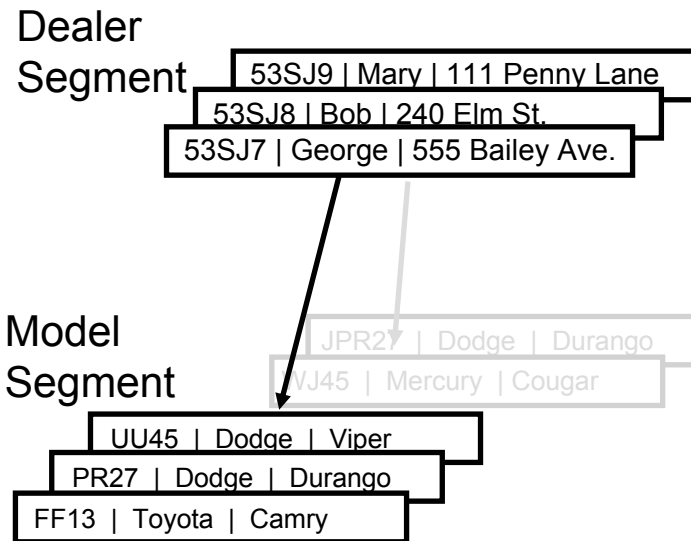


IMS JDBC Obstacles

- IMS uses Segment Search Arguments (SSA) not SQL
 - Internal SQL-to-SSA Parser (with modified SQL syntax)
- No Runtime Metadata Catalog
 - DLIDatabaseView Class
- No Access to DLI Data from Java
 - JNI-to-XXXTDLI Interface
 - CEETDLI in IMS environment
 - AERTDLI in DB2 SP and WebSphere environment
 - AIBTDLI in CICS environment

Hierarchical to Relational Terminology Mapping

Hierarchical Design



Relational Design

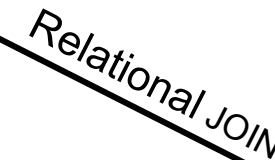
Dealer Table

	DealerID	DealerName	DealerAddress
0	53SJ7	George	555 Bailey Ave.
1	53SJ8	Bob	240 Elm St.
2	53SJ9	Mary	111 Penny Ln.
...

Model Table

ID	Make	Model	Dealer	
UU45	Dodge	Viper	53SJ7	0
PR27	Dodge	Durango	53SJ7	0
FF13	Toyota	Camry	53SJ7	0
JR27	Dodge	Durango	53SJ8	1
WJ45	Mercury	Cougar	53SJ8	1
...

Relational JOIN

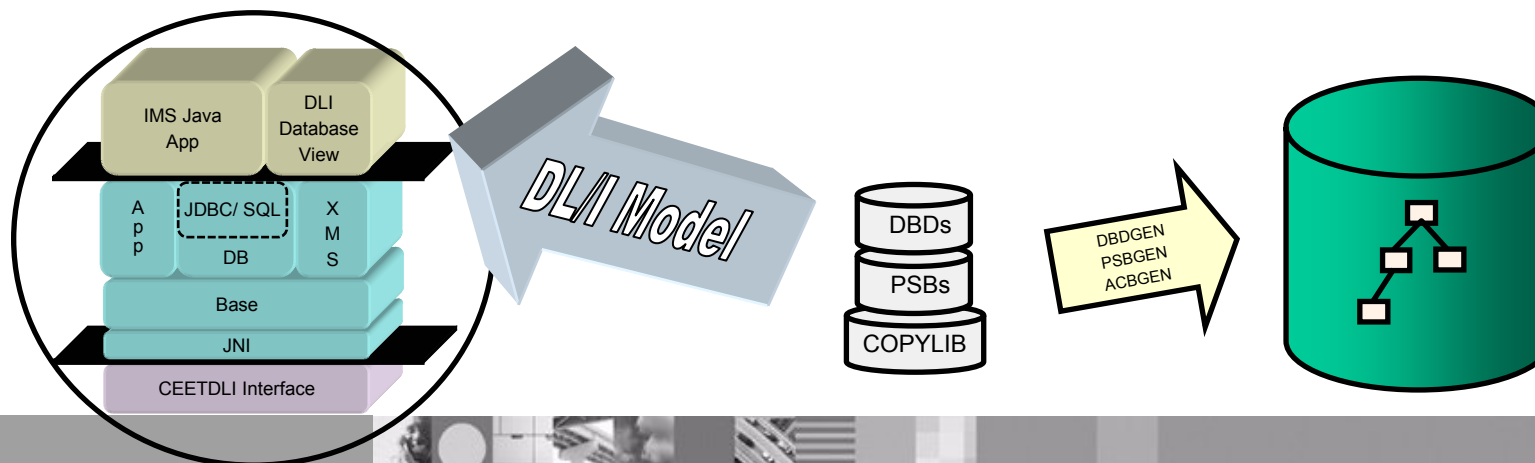


Note: Segment Names ~ Table Names
 Segment Instances ~ Table Rows
 Field Names ~ Column Names



DL/I Model Utility

- **Parse DBD, PSB and Control Statements (COBOL Copylib)**
 - Optionally import COBOL XMI (generated from COBOL copybooks) to define segments
- **Produce XMI to act as a standard form of IMS Metadata**
- **Generate the IMS Java metadata (DLIDatabaseView) from the XMI**
 - Also generate IMS Java Report – a programmer friendly view of the database (hierarchy, segments, fields, etc)

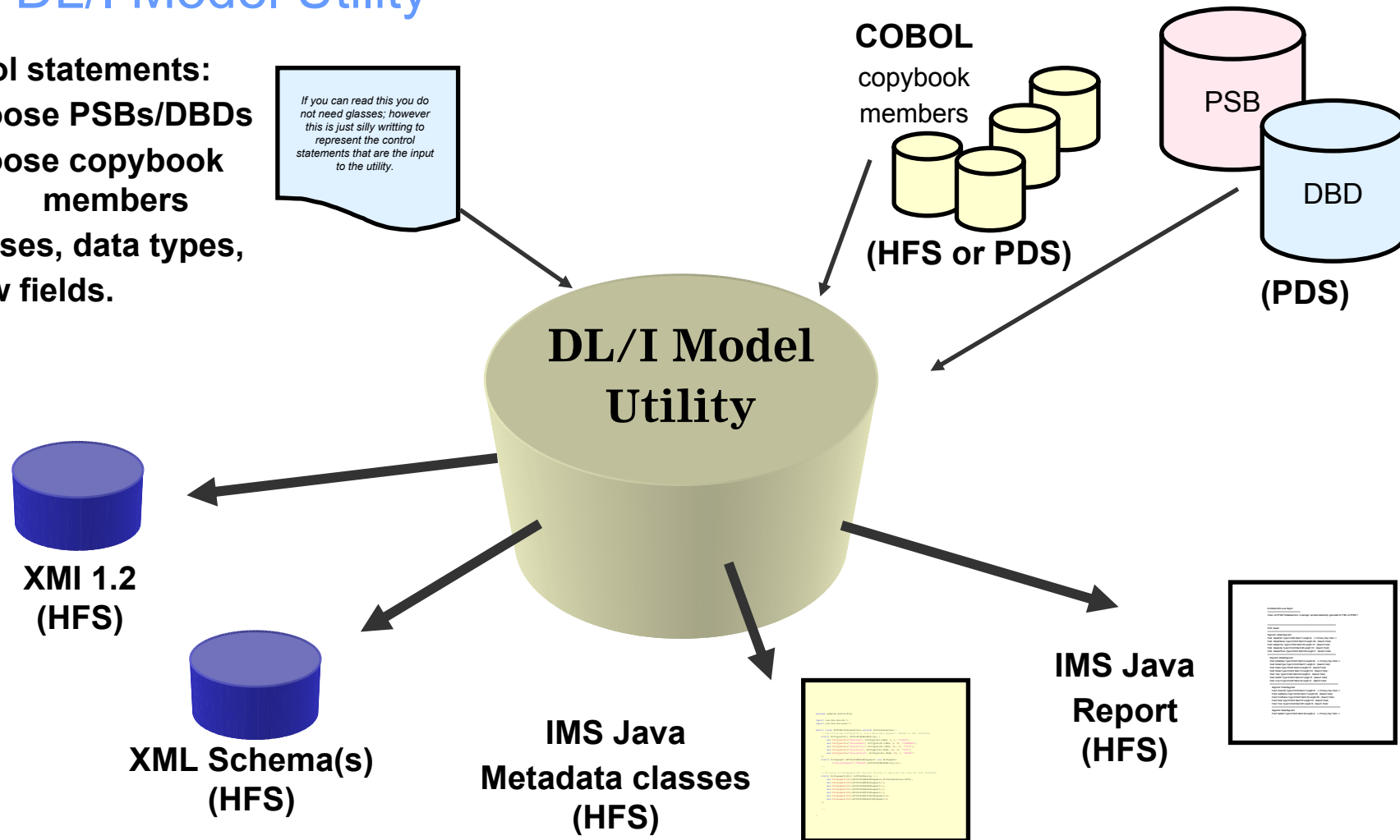


DL/I Model Utility

Control statements:

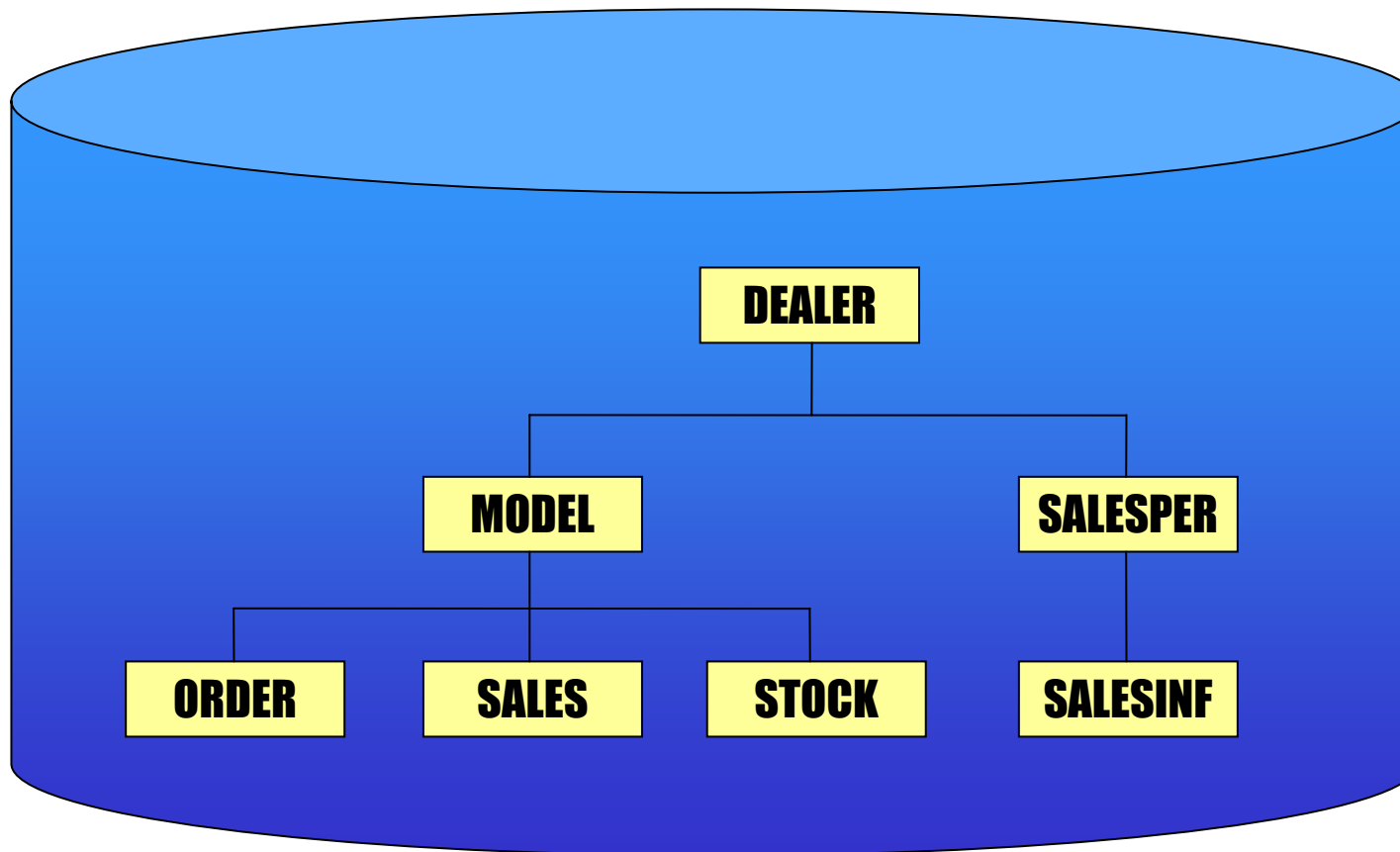
- 1) Choose PSBs/DBDs
- 2) Choose copybook members
- 3) Aliases, data types, new fields.

If you can read this you do not need glasses; however this is just silly writing to represent the control statements that are the input to the utility.





Sample Database for SQL Example





SQL Parsing

SQL

```
SELECT Dealer.Name, Dealer.Phone, Order.LastName  
FROM SomePCB.Order  
WHERE Model.MSRP > '50000'  
      AND Order.Date >= '5/1/2003'  
      AND Order.Date <= '5/31/2003'
```

SSA List





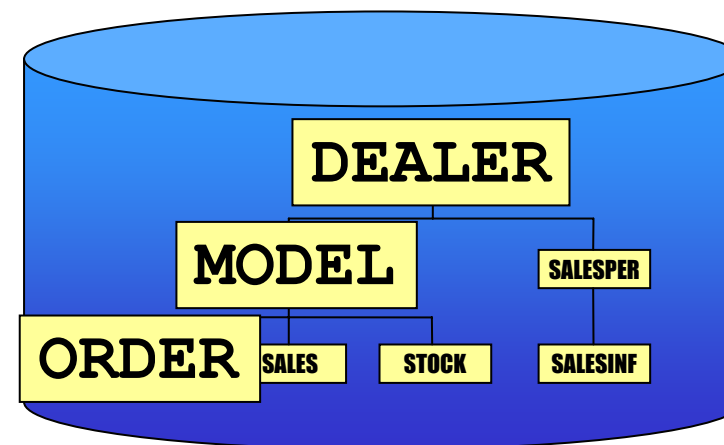
SQL Parsing

SQL

```
SELECT Dealer.Name, Dealer.Phone, Order.LastName
FROM SomePCB.Order
WHERE Model.MSRP > '50000'
      AND Order.Date >= '5/1/2003'
      AND Order.Date <= '5/31/2003'
```

SSA List

DEALER
MODEL
ORDER





SQL Parsing

SQL

```
SELECT Dealer.Name, Dealer.Phone, Order.LastName  
FROM SomePCB.Order  
WHERE Model.MSRP > '50000'  
      AND Order.Date >= '5/1/2003'  
      AND Order.Date <= '5/31/2003'
```

SSA List

DEALER

MODEL (MSRP GT50000)

ORDER (DATE GE20030501 &

DATE LE20030531



SQL Parsing

SQL

```
SELECT Dealer.Name, Dealer.Phone, Order.LastName  
FROM SomePCB.Order  
WHERE Model.MSRP > '50000'  
      AND Order.Date >= '5/1/2003'  
      AND Order.Date <= '5/31/2003'
```

SSA List

```
DEALER    *D  
MODEL     (MSRP      GT50000)  
ORDER     (DATE      GE20030501 &  
          DATE      LE20030531)
```

SQL Parsing

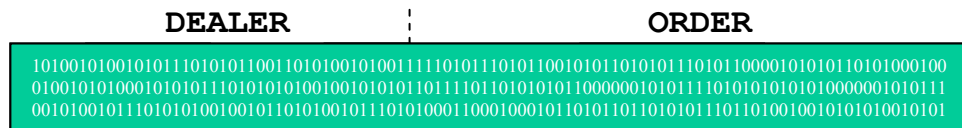
SQL

SELECT Dealer.Name, Dealer.Phone, Order.LastName

SSA List

```
DEALER *D  
MODEL (MSRP GT50000)  
ORDER (DATE GE20030501&  
DATE LE20030531)
```

IOArea



ResultSet



ROW



Non-SQL DB Access

- IMS DL/I API capable of functionality that SQL cannot provide
 - Command codes on SSAs, as an example
- IMS Java provides a Java API to provide this function
 - Objects representative of SSA lists and Segment Search Arguments are provided
 - Full support of SSA functionality
 - All command codes are supported



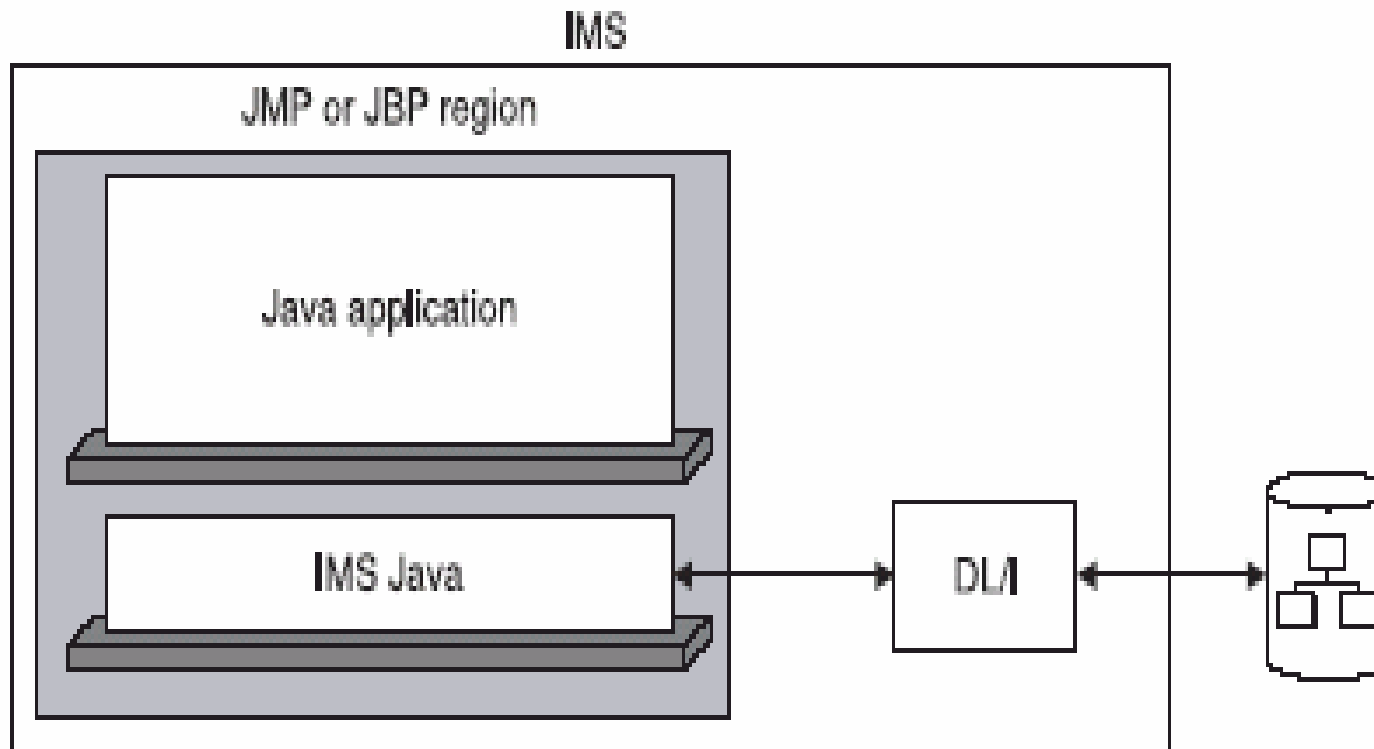
Overview

- **IMS Java**
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- **What is JDBC?**
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- **Environments**
 - Non-managed
 - IMS
 - CICS
 - DB2
 - Managed
 - WebSphere
- **Details of WebSphere Application Server**
 - DataSource
 - Enterprise Archive
- **Remote Database Services**





IMS TM Java Access to IMS Data - Overview





IMS TM Java Access to IMS Data

- Runs in Java Virtual Machine
 - IMS 7.1 or later
 - JDK 1.3+ (strongly suggest \geq JDK 1.4.2)
 - JDBC 2.1
- Two new IMS Dependent Regions for Java processing
 - JMP
 - Analogous to an MPP region
 - JBP
 - Analogous to a non-message driven BMP
- Applications can access only the PSB they were defined to
 - Similar to any other application in an IMS Dependent Region
- Syncpoints done by calls to application package
 - Must be explicitly called by an application





Persistent Reusable JVM

- Speeds up the processing of Java applications in transaction processing environments on z/OS systems
 - Run multiple JVMs within a z/OS address space
 - Ability to process any number of transactions using the same JVM
 - JVM can be reset (back to its initial state) between transactions or whenever necessary
- JVMSet
 - One per address space
 - Consists of one master JVM and a set of worker JVMs
 - Master JVM
 - » Does not participate in work. Used for JVMSet initialization
 - Worker JVM
 - » Applications run under this JVM
- Application isolation still maintained
 - Only one transaction run at a time in a JVM
 - Each JVM created in its own Language Environment (LE) enclave





IMS Message Queue and Syncpoint Access

- IMS Java provides an API to access the IMS Message Queues
 - Read a message from the IMS Message Queue (GU)
 - Read a multi-segmented message from the queue (GN)
 - Insert a message to the IMS Message Queue
 - Program switch
 - Insert a message to an alternate IO PCB (message queue)
- IMS Java also provides an API to commit/rollback a transaction
 - Drives IMS Syncpoint Manager.
 - If there are RRS RMs then RRS is the coordinator.





DB2 Interoperability for Java Dependent Regions

- **Run in Java Dependent Regions**
 - **Access IMS through IMS JDBC driver**
 - **Access DB2 through DB2 JDBC/SQLJ drivers**
- **RRS handles syncpoint**



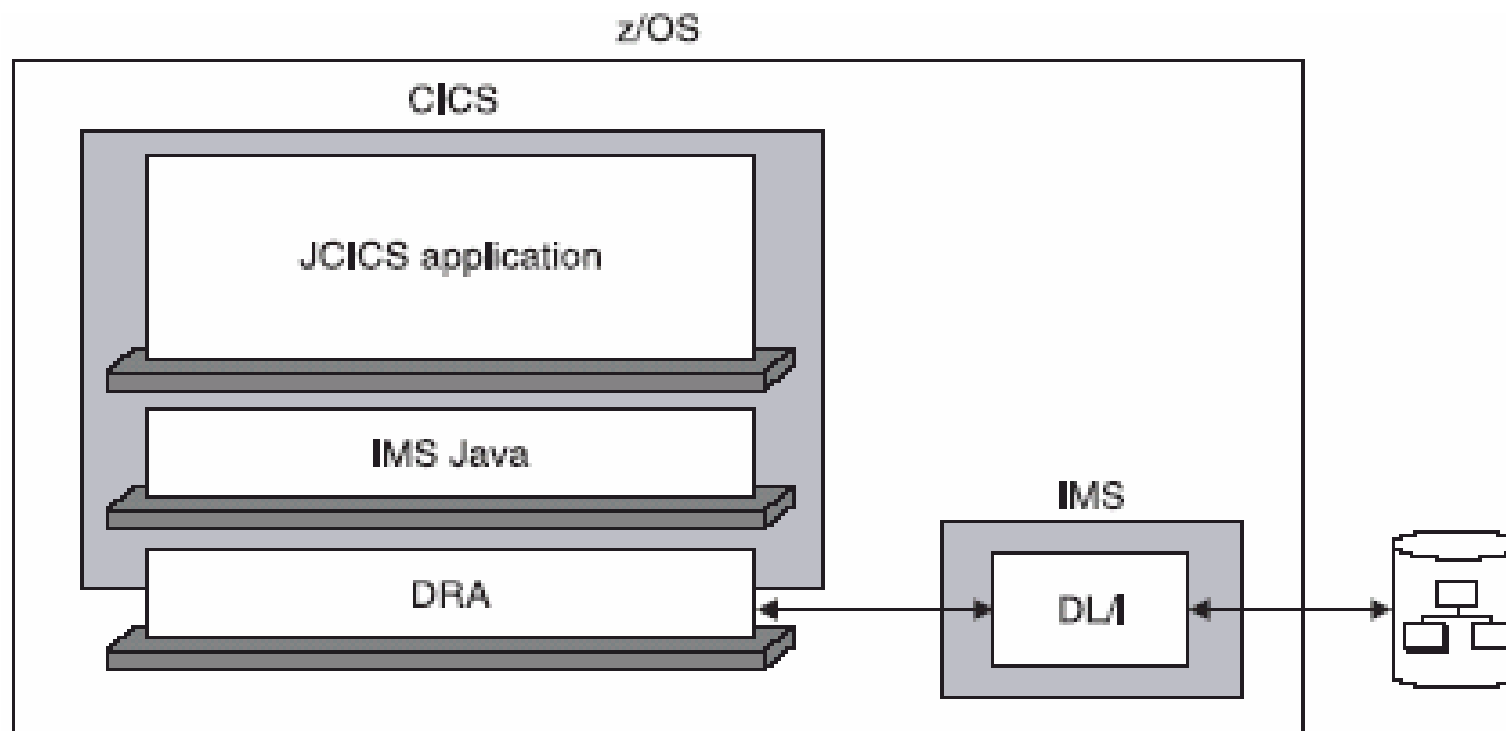
Overview

- **IMS Java**
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- **What is JDBC?**
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- **Environments**
 - Non-managed
 - IMS
 - **CICS**
 - DB2
 - Managed
 - WebSphere
- **Details of WebSphere Application Server**
 - DataSource
 - Enterprise Archive
- **Remote Database Services**





CICS Java Access to IMS Data - Overview





CICS Java Access to IMS Data

- Runs in Java Virtual Machine
 - CICS TS 2.2
 - JCICS API (Java version of the CICS API)
 - JDK 1.3+ (strongly suggest \geq JDK 1.4.2)
 - JDBC 2.1
- Can only have **one** allocated PSB at a time
 - Only one open connection
 - Can open a new connection after all others are closed
- Syncpoint done implicitly when a PSB is deallocated
 - When a Connection is closed



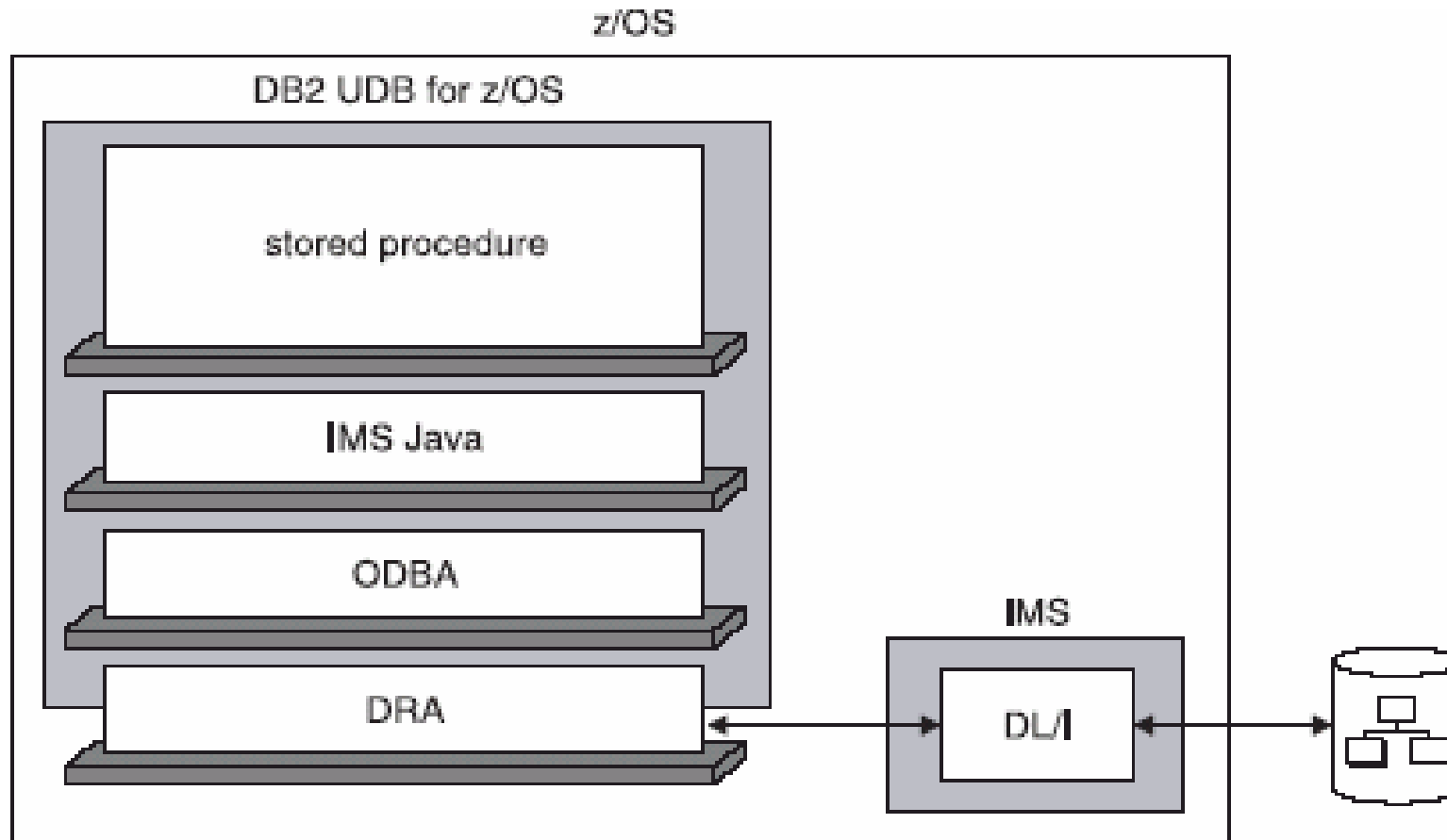
Overview

- **IMS Java**
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- **What is JDBC?**
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- **Environments**
 - Non-managed
 - IMS
 - CICS
 - **DB2**
 - Managed
 - WebSphere
- **Details of WebSphere Application Server**
 - DataSource
 - Enterprise Archive
- **Remote Database Services**





DB2 Stored Procedure Support - Overview





DB2 Stored Procedure Support

- DB2 Version 7
 - DB2 UDB for z/OS V7 APARs PQ46673 (Stored Procedure support) & PQ50443 (access to IMS from a DB2 Stored Procedure)
 - DB2 for z/OS SQLJ/JDBC driver with APAR PQ48383 or the DB2 Universal JDBC driver
- Stored Procedures that access IMS Databases
 - User-written SQL programs that are stored at the DB2 server and can be invoked by a client application
- DB2 handles syncpoint (not stored procedure)
 - Client that invokes the SP can commit/rollback
- DRA table and name required





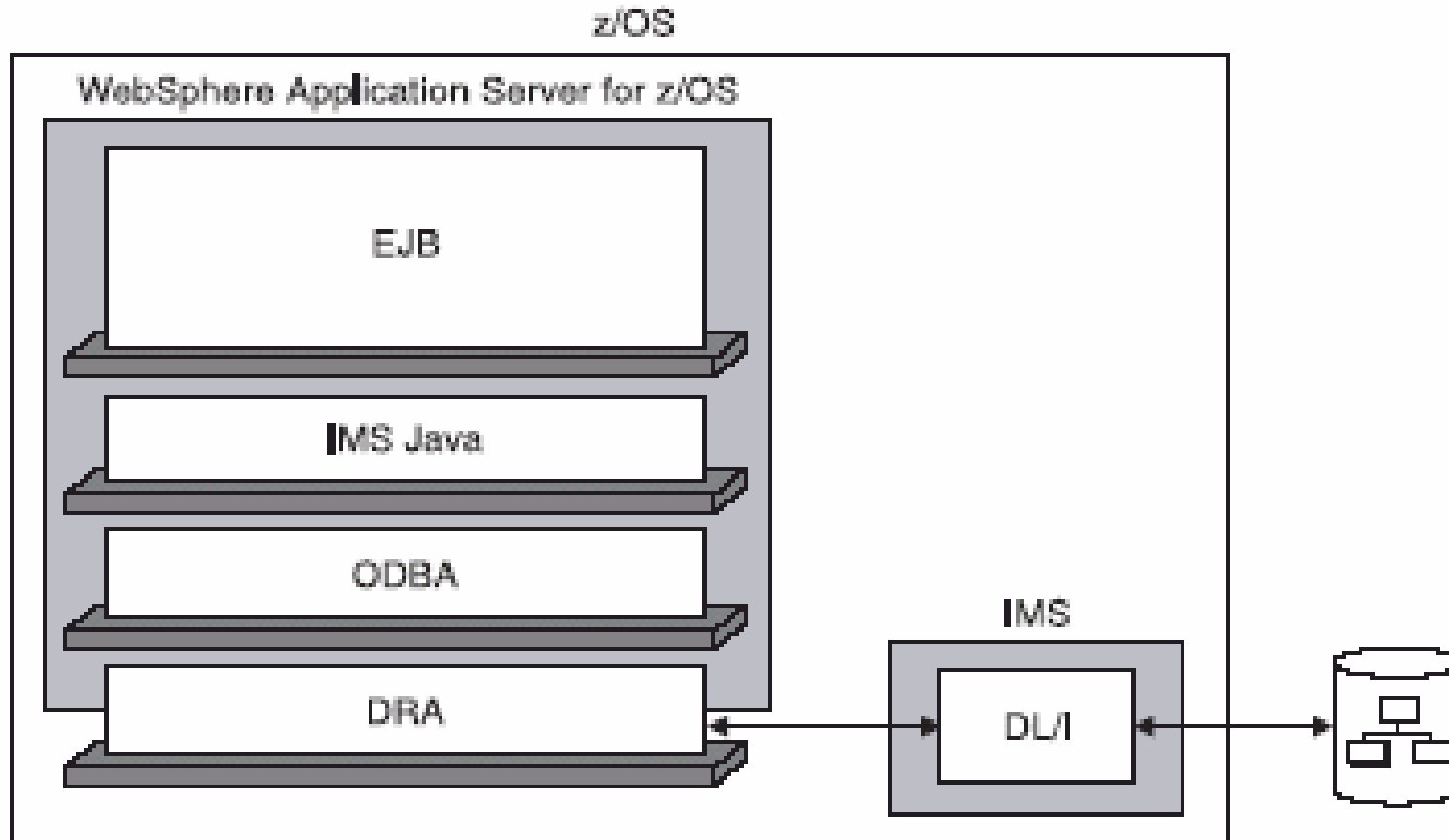
Overview

- IMS Java
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- What is JDBC?
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- Environments
 - Non-managed
 - IMS
 - CICS
 - DB2
 - Managed
 - WebSphere
- Details of WebSphere Application Server
 - DataSource
 - Enterprise Archive
- Remote Database Services





WebSphere Application Server Support - Overview





WebSphere Application Server Support

- Runs in Java Virtual Machine
 - JDK 1.3+ (Strongly suggest \geq JDK 1.4.2)
 - JDBC 2.1
- IMS JDBC provides a Resource Adapter to operate in this environment which interfaces with our class libraries
- Applications run as Enterprise Java Beans (EJBs)
 - Support for both Bean managed and Container managed session beans
- Access IMS databases through ODBA/DRA
- J2EE Connection Architecture
 - Managed Environment
 - Connections are managed by application server



Overview

- IMS Java
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- What is JDBC?
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- Environments
 - Non-managed
 - IMS
 - CICS
 - DB2
 - Managed
 - WebSphere
- Details of WebSphere Application Server
 - DataSource
 - Enterprise Archive
- Remote Database Services

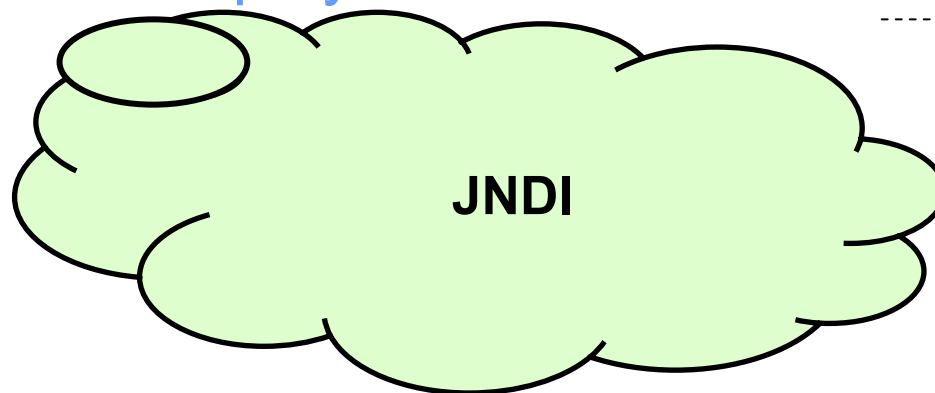


DataSource

- Factory for connections to a physical data source
- Replacement to the DriverManager facility
 - Required when running in a managed environment (WebSphere)
- Typically registered with a naming service based on the Java™ Naming and Directory (JNDI) API.
 - Names are associated with objects and objects are found based on their names.
- DataSource objects have properties that can be modified when necessary
 - Code accessing the data source does not need to be changed
 - (Properties include: DRA Name, DLIDatabaseView)



DataSource Deployment



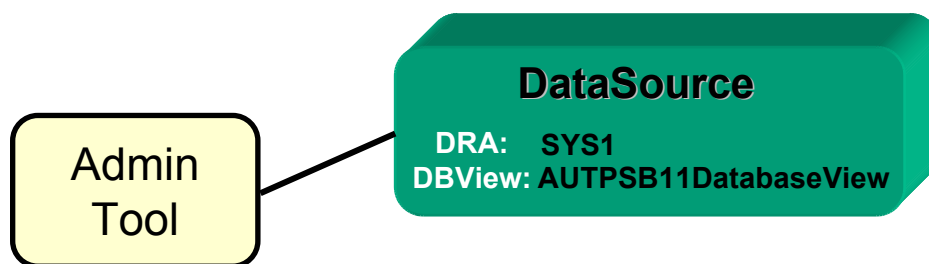
AbcdefDB

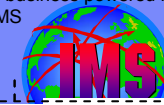
the way

DataSource
DRA: SYS3
DBView: TestAppDatabaseView

EmployeeDB

DataSource
DRA: SYS1
DBView: TestApp2DatabaseView





DataSource Deployment

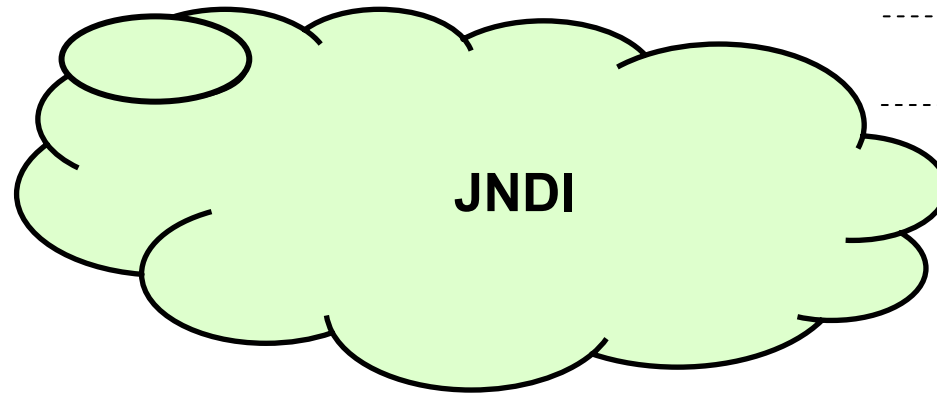
AbcdefDB

the way

DataSource
DRA: SYS3
DBView: TestAppDatabaseView

EmployeeDB

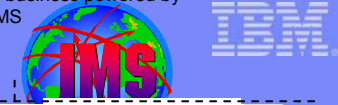
DataSource
DRA: SYS1
DBView: TestApp2DatabaseView



"DealershipDB"

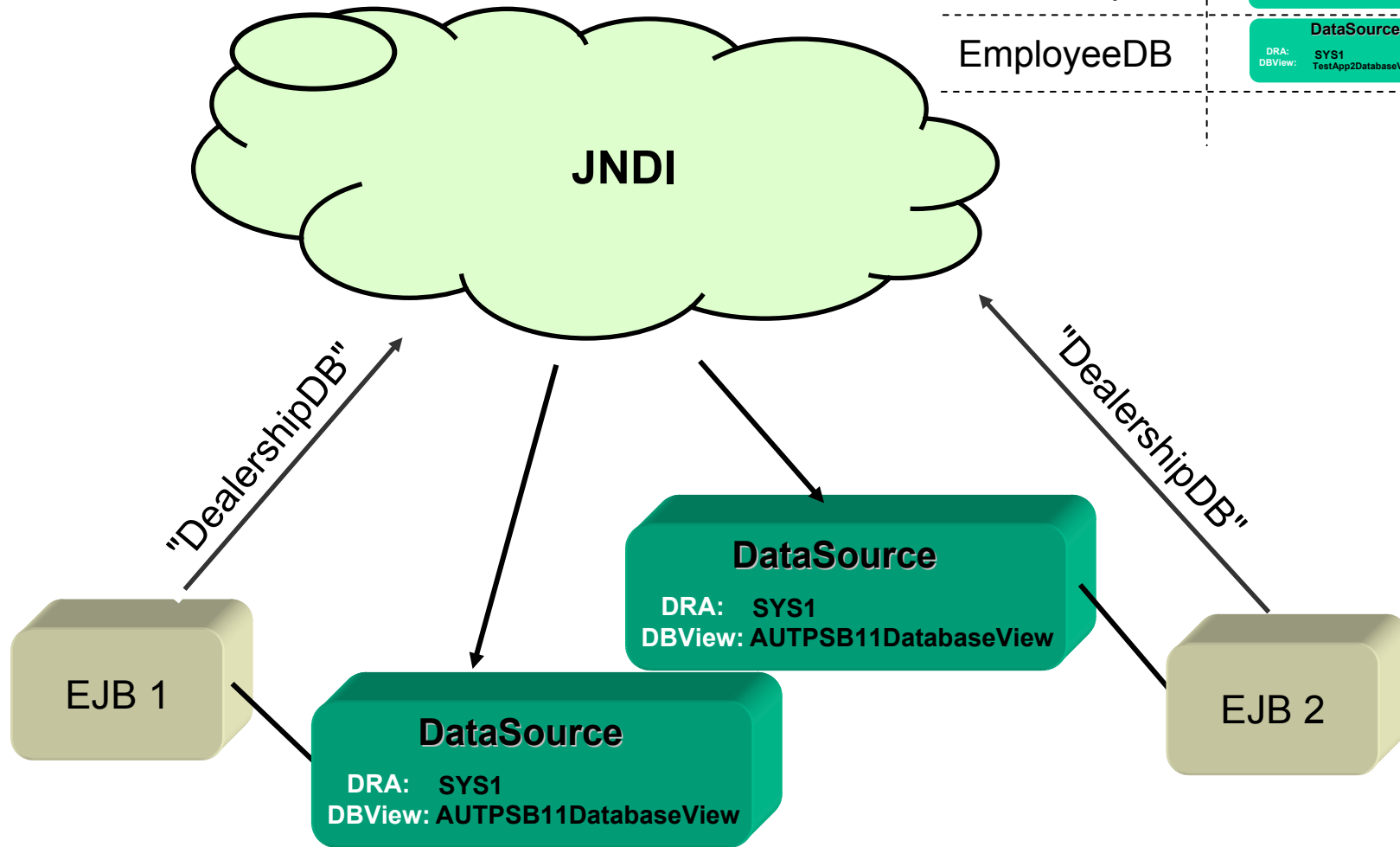
Admin
Tool

DataSource
DRA: SYS1
DBView: AUTPSB11DatabaseView



DataSource Retrieval

AbcdefDB	<i>the way</i> DataSource DRA: SYS3 DBView: TestAppDatabaseView
DealershipDB	DataSource DRA: SYS1 DBView: AUTPSB11DatabaseView
EmployeeDB	DataSource DRA: SYS1 DBView: TestApp2DatabaseView





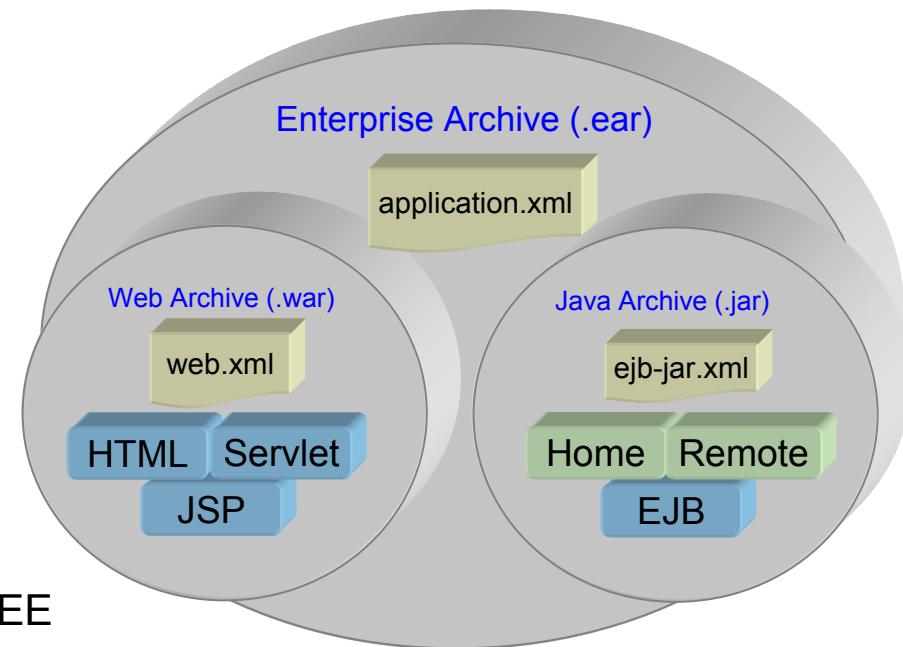
Overview

- IMS Java
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- What is JDBC?
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- Environments
 - Non-managed
 - IMS
 - CICS
 - DB2
 - Managed
 - WebSphere
- Details of WebSphere Application Server
 - DataSource
 - Enterprise Archive
- Remote Database Services

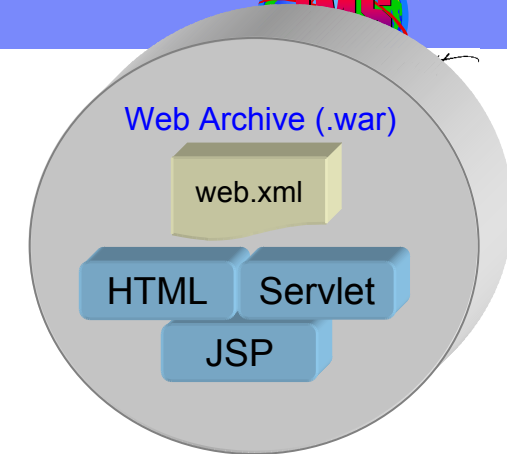


Enterprise Archive (ear)

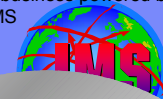
- J2EE Enterprise Application Archive
 - Complete J2EE application
- Must contain
 - One or more J2EE modules (Java Archive)
 - Deployment descriptor (application.xml)
 - Represents a top level view of a J2EE application's contents
- May contain
 - One or more Web modules (Web Archive)
 - Libraries referenced by J2EE modules



Enterprise Archive (Servlet)

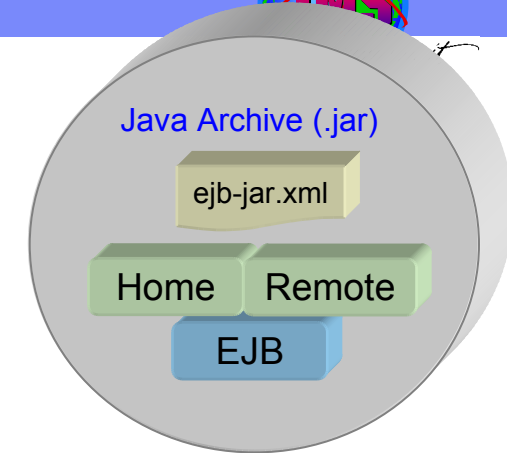


- Looks up the EJB home interface in JNDI
- Using the home interface, creates the EJB remote interface
- Invokes methods on the remote interface
 - Remote interface uses IIOP to communicate to EJB
 - Proxy (Pass-through interface)
- Passes results of method to a Java Server Page (JSP) for displaying on a web browser



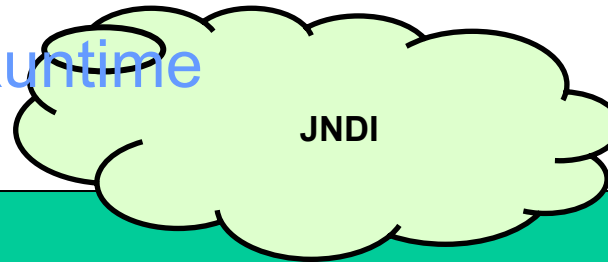
Enterprise Archive (EJB)

- Receives requests from the servlet
- Obtains a connection
 - Looks up the deployed DataSource instance and requests a connection from it
- Accesses the database(s) and performs business logic
- Sends the results back to the servlet to display

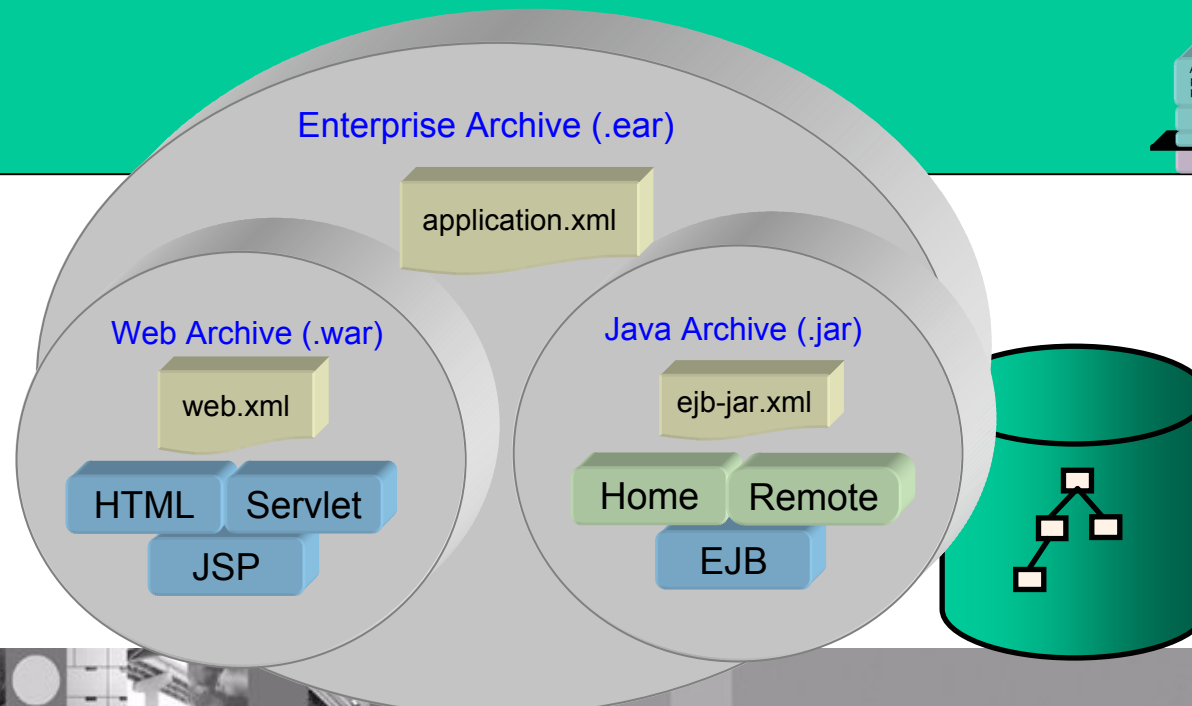
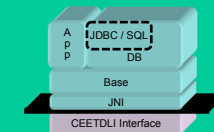




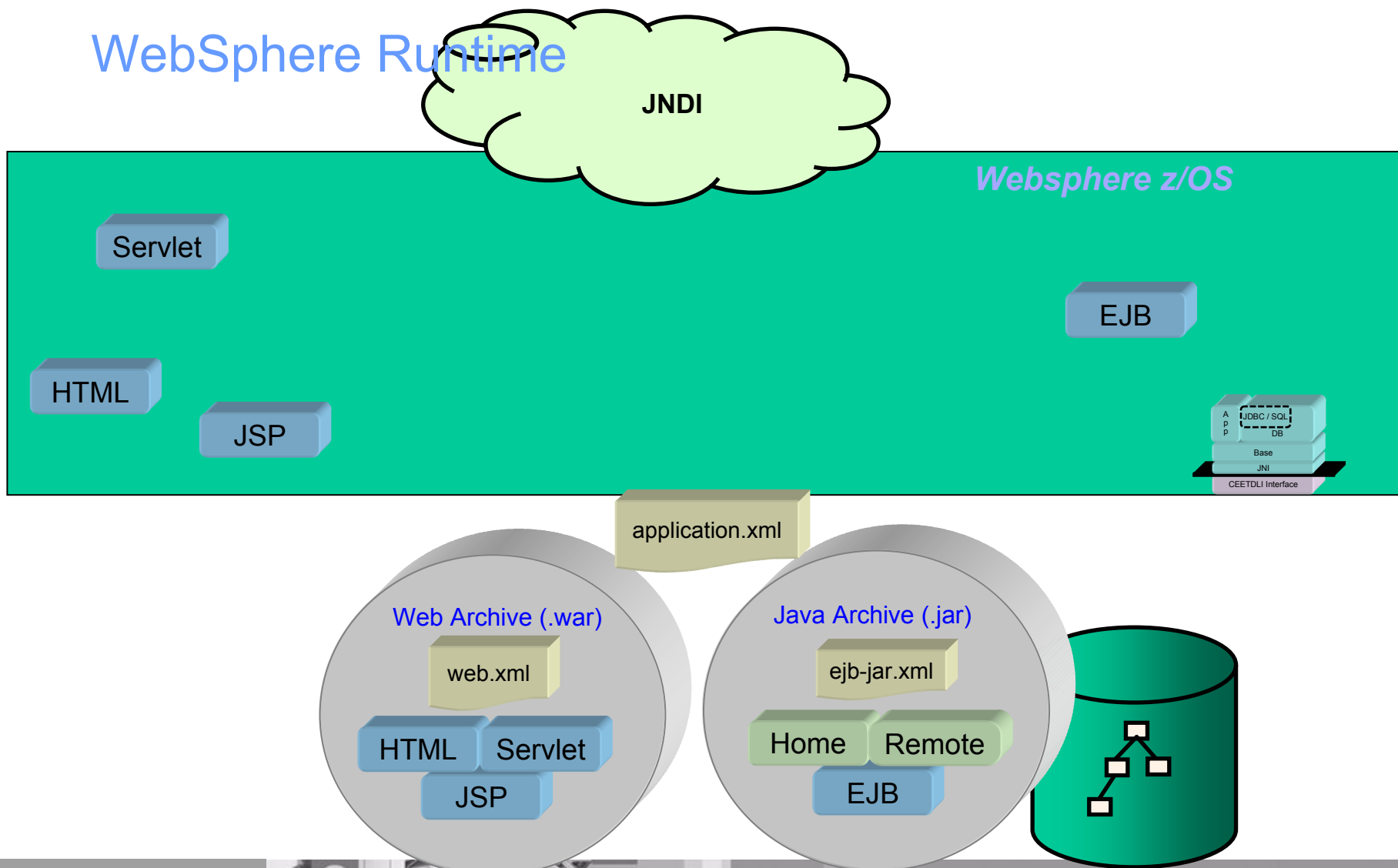
WebSphere Runtime



WebSphere z/OS



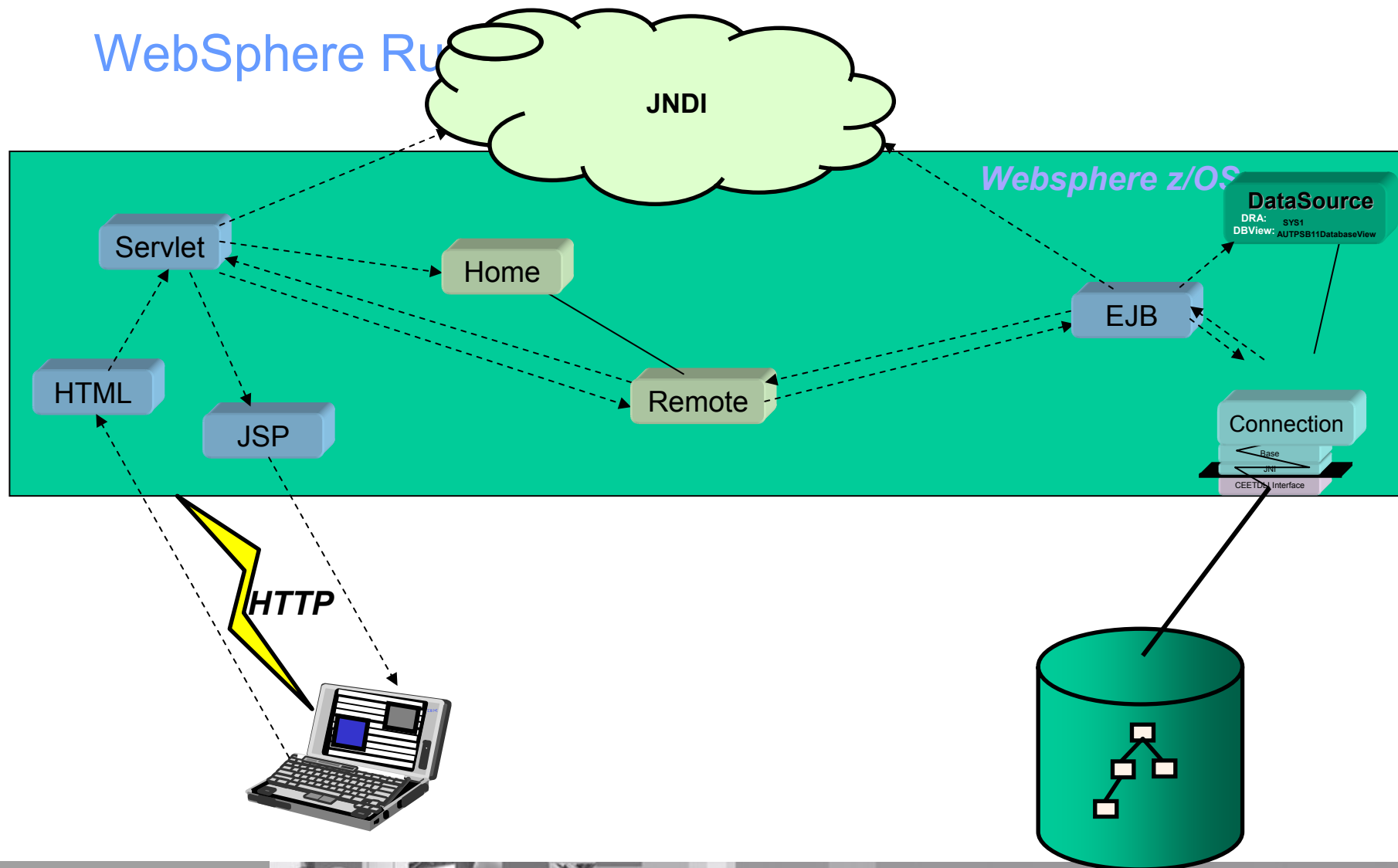
WebSphere Runtime





the world depends on it

WebSphere Runtime





Overview

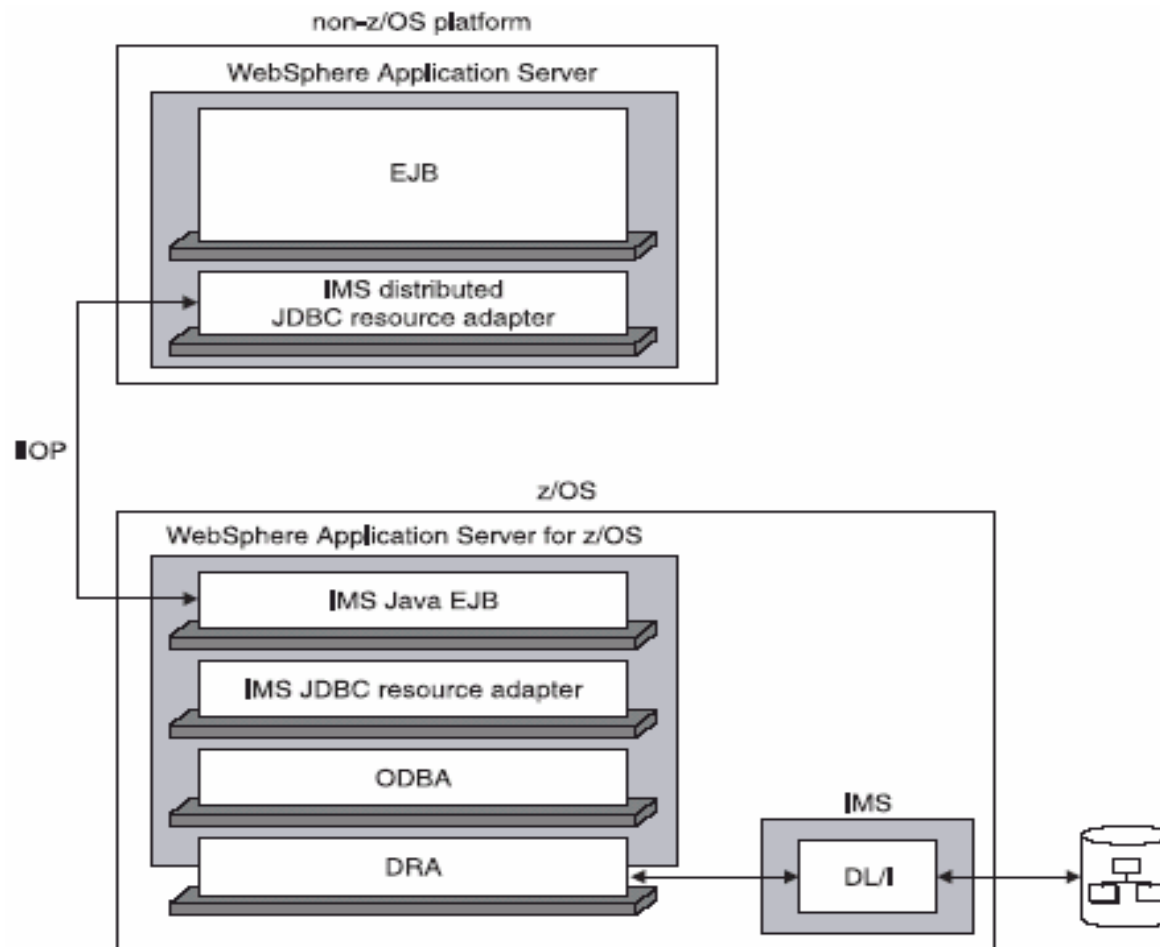
- **IMS Java**
 - What Is IMS Java
 - Why Use IMS Java
 - IMS Java Class Library Architecture
- **What is JDBC?**
 - Problems with JDBC
 - DL/I Model Utility
 - SQL example
- **Environments**
 - Non-managed
 - IMS
 - CICS
 - DB2
 - Managed
 - WebSphere
- **Details of WebSphere Application Server**
 - DataSource
 - Enterprise Archive
- **Remote Database Services**





the world depends on it

Remote Database Services - Overview



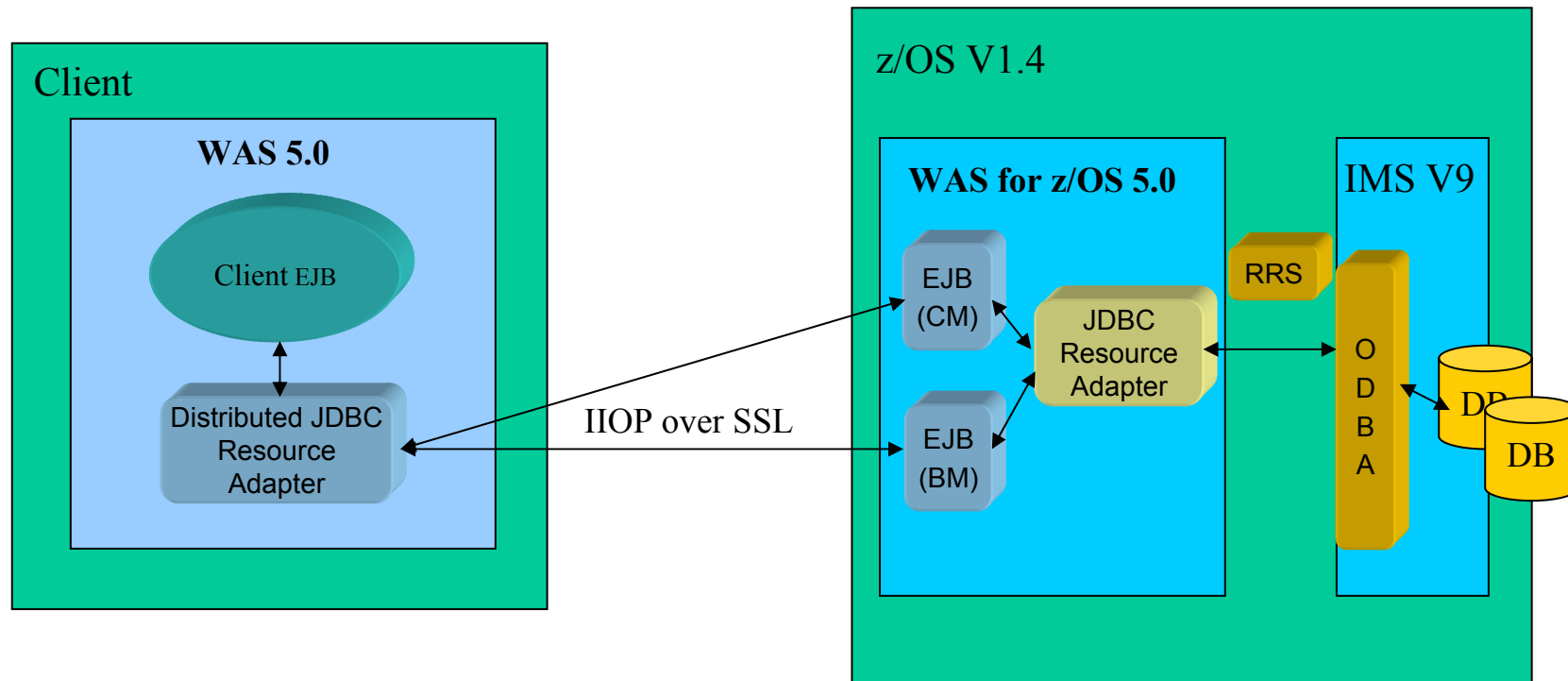


Remote Database Services

- **Remote Database Services (RDS) available IMS Version 9**
- **Ability to access IMS DL/I data from a distributed J2EE server**
 - WebSphere Application Server 5.0.2 for distributed platforms
- **Complete client application deployed on distributed J2EE server**
 - Distributed functionality is transparent to client application
- **All client-server communication is handled by new IMS Java components**
 - IMS JDBC distributed Resource Adapter (client-side RAR)
 - Container Managed EJB (server-side EAR)
 - Bean Managed EJB (server-side EAR)



Remote Database Services



CM – Container Managed

BM – Bean Managed



Remote Database Services – Client Setup

- Client-side
 - Install IMS Java distributed JDBC resource adapter (one-time setup)
 - Deploy J2C Connection Factory (DataSource)
 - Can be done once per IMS Database, or generically (one-time setup) where each property is set by the application at runtime after it looks up the DataSource
 - Properties
 - Host machine name
 - Host IIOp port number
 - DRA name
 - DatabaseView name



Remote Database Services – Server Setup

- Server-side (one-time setup)
 - Similar process as with regular IMS Java JDBC resource adapter
 - Install IMS Java JDBC resource adapter
 - Configure IMS custom service
 - Install Remote Database Services enterprise archive
 - Contains the two EJBs that service client requests
 - The installation of the archive is done from the web based admin tool on the non-z/OS platform
 - Deploy generic J2C Connection Factory (DataSource)
 - No DRA name specified
 - No DatabaseView name specified



Remote Database Services – Transaction Semantics

- Global Transaction Semantics
 - Entire application in one Unit of Work (UOW)
 - All database activity is committed or rolled back at one time
 - Scenarios resulting in global semantics
 - Application is a container-managed EJB
 - Application is a bean-managed EJB and a UserTransaction is started prior to first SQL query
 - Connection.commit() and Connection.rollback() are not allowed as calls imply local transaction semantics
 - If container-managed, container responsible for commit/rollback
 - If bean-managed, application responsible for commit/rollback via UserTransaction object
 - Container-managed EJB on server side is invoked by the IMS Java distributed resource adapter for communication from client to server

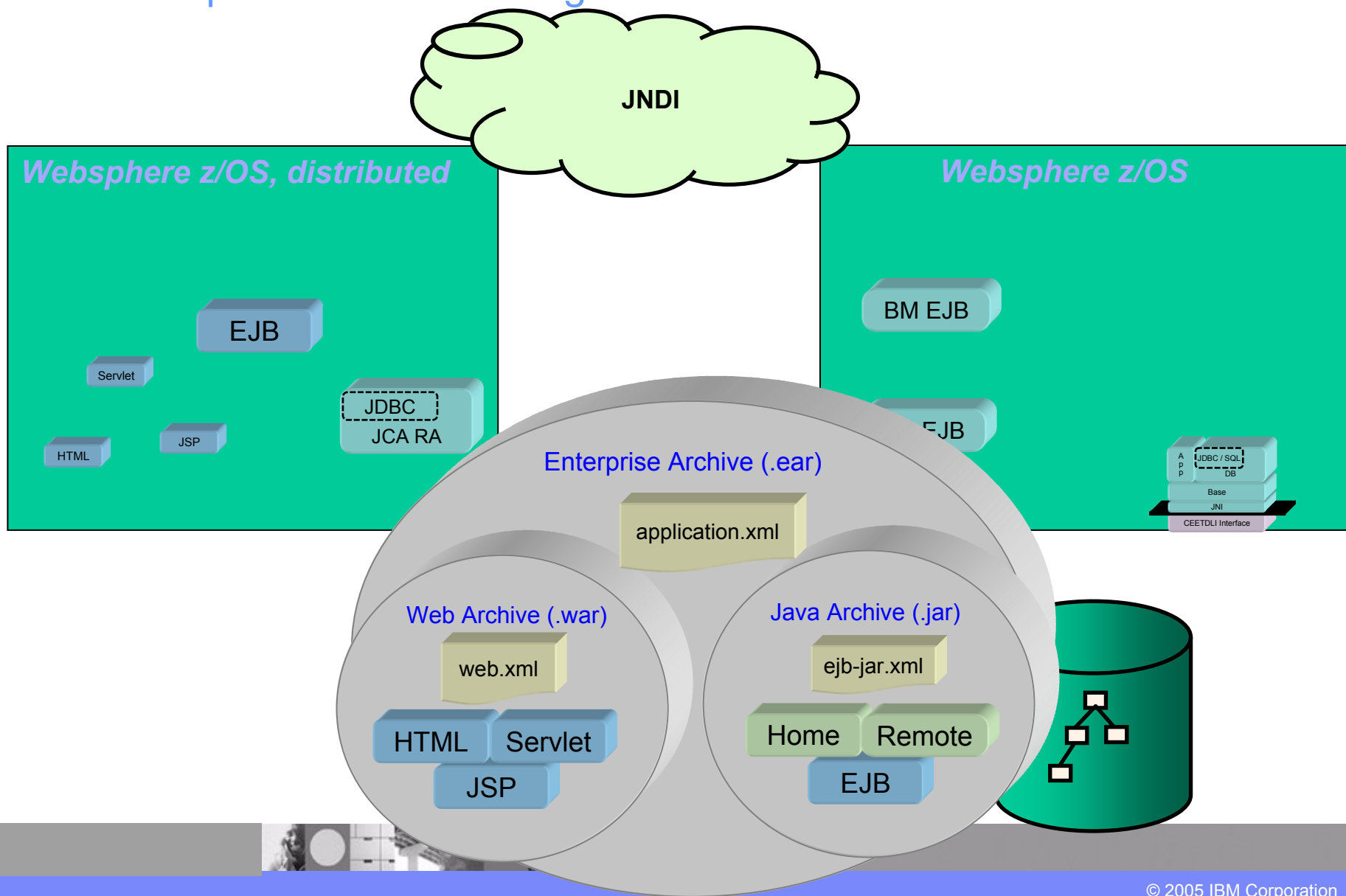




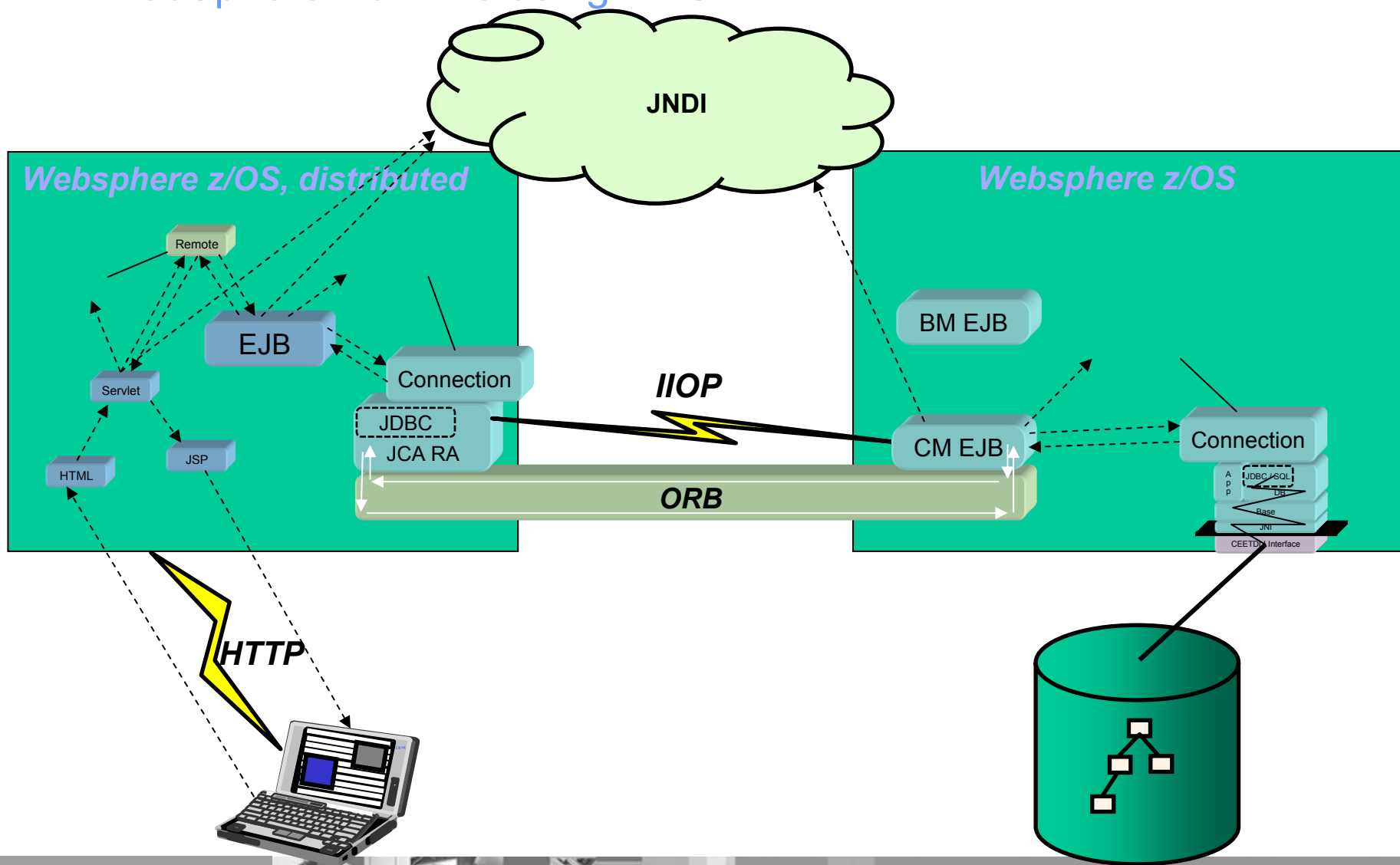
Remote Database Services – Transaction Semantics

- Local Transaction Semantics
 - Each connection to a database under its own Unit of Work
 - Work on a particular connection (database) is committed or rolled back without effecting work on other databases
 - Scenario resulting in local semantics (emulated)
 - Application is a bean-managed EJB and no transaction is started prior to first SQL query
 - Connection.commit() (and rollback) is allowed
 - Bean-managed EJB on server side is invoked by the IMS Java distributed JDBC resource adapter for communication from client to server
 - Transaction started by IMS Java JDBC distributed resource adapter
 - Transactions are still global, but since each client connection has its own reference to the bean-managed EJB, local semantics are emulated

WebSphere Runtime using RDS



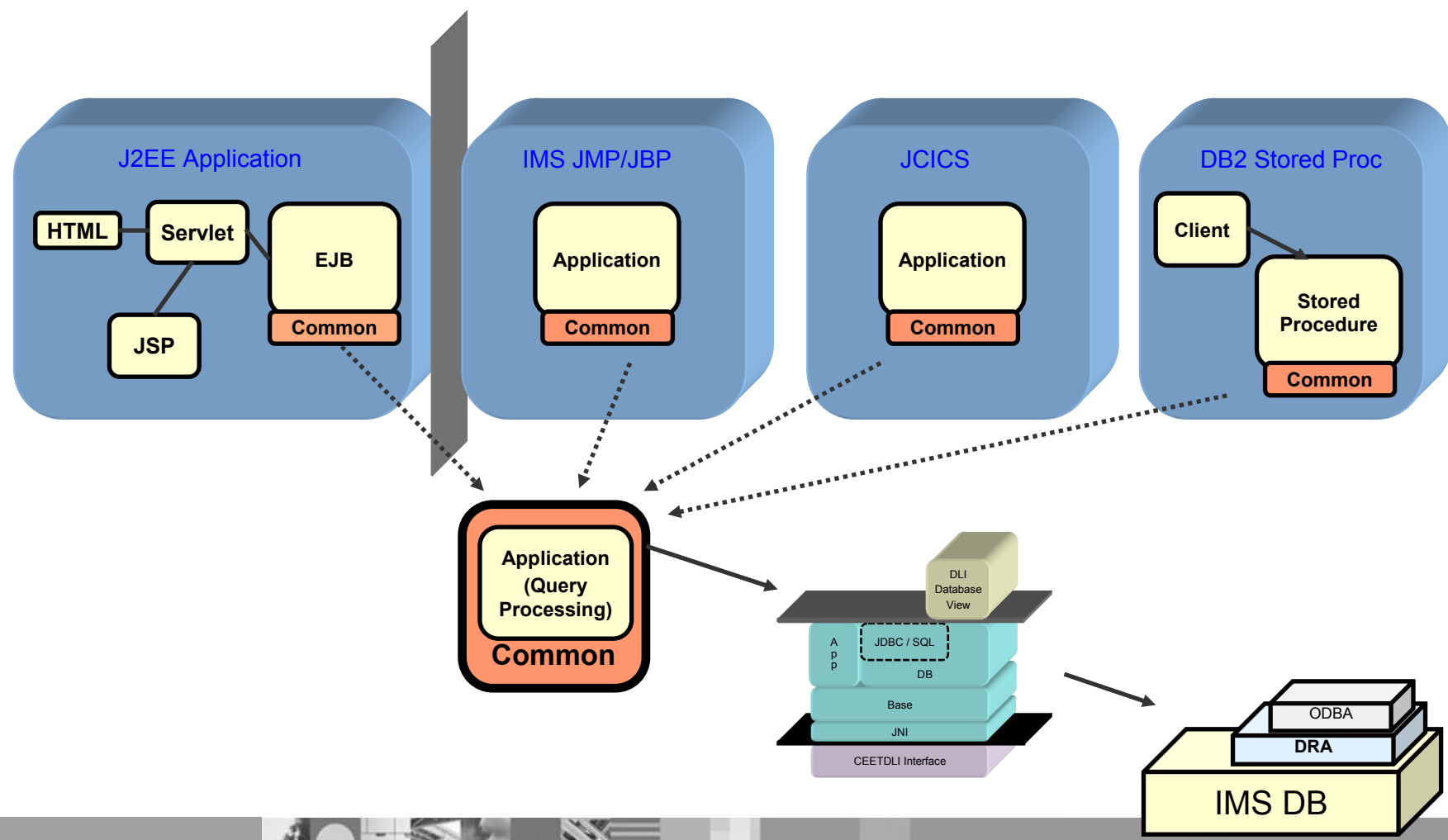
WebSphere Runtime using RDS





the world depends on it

Conclusion





Books/Documents for reference

- **IMS Java Guide & Reference**
 - IMS V9 (SC18-7821)
 - IMS V8 (SC27-1296)
 - IMS V7 (SC27-0832)
- **New IBM Technology featuring Persistent Reusable JVM**
 - SDK 1.3 (SC34-6034)
- **DCF Technotes – “IMS Java: Commonly Asked Questions”**
- **IMS Java web page**
 - Can be accessed from <http://www.ibm.com/ims>



IBM Software Group

Appendix – Environment Setup Details

IMS TM

CICS

DB2 Stored Procedure

DB2 Interoperability (from IMS TM)





Java Dependent Regions

- JMP (Java Message Processing)
 - For message-driven Java applications
 - New IMSJMP JOB that EXECs the new DFSJMP procedure
 - DFSJMP procedure added to IMS.PROCLIB
 - Similar to the DFSMPR procedure for MPPs
 - Couple of new parameters
 - Several DFSMPR parameters not supported
- JBP (Java Batch Processing)
 - For non-message driven Java applications
 - New IMSJBP JOB that EXECs the new DFSJBP procedure
 - DFSJBP procedure added to IMS.PROCLIB
 - Similar to the IMSBATCH procedure for BMPs
 - Couple of new parameters
 - Several IMSBATCH parameters not supported





JMP/JBP Setup

- **JCL PROC**

- Add location of DFSCLIB to STEPLIB
e.g. IMSJAVA.JAVARESL.PDSE

- **DFSJVMAP**

- Alias PSB Name to map to Java Application Name

- **DFSJVMEV**

- Set location of libJavTDLI.so

- **DFSJVMMS (Master JVM)**

- Set middleware classpath to IMS Java Jar (*Java ARchive*)
- Set application classpath to location of app code

- **DFSJVMWK (Worker JVM)**

- Only used in a JMP region



JMP/JBP Setup - DFSJVMAP

```
* Mapping example for the IMS Java IVP PSB genned as: *
* PSBGEN PSBNAME=DFSIVP37,LANG=JAVA *
* *
* With the actual IVP sample class file inside the samples.jar *
* which is at pathname: *
* /usr/lpp/ims/imsjava91/samples/samples.jar *
* *
* *
```

DFSIVP37=samples/ivp/ims/IMSIVP





JMP/JBP Setup - DFSJVMEV

* LIBPATH environment variable *

*

*

* *Assuming the libJavTDLI.so is at pathname:* *

* */usr/lpp/ims/imsjava91*

*

* *and*

*

* *The IBM SDK is installed at /usr/lpp/java*

*

LIBPATH=/usr/lpp/java/bin/classic:/usr/lpp/java/bin:>

/usr/lpp/ims/imsjava91





JMP/JBP Setup - DFSJVMMS

```
*****
*      JVMOPMAS= member                                *
*****
*
-Dibm.jvm.shareable.application.class.path=>
/usr/lpp/ims/imsjava91/samples/samples.jar
*
-Dibm.jvm.trusted.middleware.class.path=>
/usr/lpp/ims/imsjava91/imsjava.jar
*
-Xinitacsh128k
-Xinitsh128k
-Xmaxf0.6
-Xminf0.3
-Xmx64M
-Xoss400k
```



JMP/JBP Setup - DFSJVMWK

* JVMOPWKR= member *

* The following JVM options are a subset of the options *

* allowed *

-Xmaxf0.6

-Xminf0.3

-Xmx64M

-Xoss400k





CICS Java: Setup & Configuration

To configure CICS for IMS Java, modify the CICS JVM profile DFHJVMPR. (The default PDS where this member is located is DFHJVM)

➤ **Include the IMS Java libraries and optionally the XML parser libraries as part of the CICS trusted middleware:**

✓ Add a TMPREFIX variable. The TMPREFIX adds the libraries at the head of the middleware path.

✓ Point to the location of the IMS Java libraries.

➤ **Update the LIBPATH variable to contain the location of the IMS Java file named libJavTDLI.so.**

✓ For example: LIBPATH=/usr/lpp/ims/imsjava91

➤ **Change the HFS dfjvmpr.props file to set the shareable application classpath (The location of this file is specified by the JVMPROPS variable in the CICS JVM profile). Point this classpath to the locations of the user applications. For example, for the IMS Java IVP for CICS:**

`ibm.jvm.shareable.application.class.path=/usr/lpp/ims/imsjava91/samples/samples.jar`



CICS Java: Setup & Configuration – CICS JCL

- **JCL job to IPL CICS server has a DFHJVM DD card**

- Edit this to point to the dataset containing the DFHJVMPR member

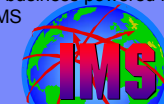
```
//*=====*
```

```
//*          DATASETS FOR JVM SUPPORT          *
```

```
//*=====*
```

```
//DFHCJVM  DD DUMMY
```

```
//DFHJVM   DD DSN=YOURHLQ.TS220.CICS.SDFHENV,DISP=SHR
```



CICS Java: Setup & Configuration - DFHJVMPR

```
WORK_DIR=.
INVOKE_DFHJVMAT=NO
JVMPROPS=/imsjava/cics/dfjjvmpr.props
LIBPATH=/usr/lpp/cicsts/lib:/usr/lpp/cicsts/ctg:\
/usr/lpp/java/J1.3/bin:/usr/lpp/java/J1.3/bin/classic:\
/usr/lpp/ims/imsjava91
#
STDIN=dfhjvmin
STDOUT=/imsjava/tmp/dfhjvmout -generate
STDERR=/imsjava/tmp/dfhjvmerr -generate
#
JAVA_HOME=/usr/lpp/java/J1.3
CICS_DIRECTORY=/usr/lpp/cics
#
TMSUFFIX=/usr/lpp/ims/imsjava91/imsjava.jar
```



CICS Java: Setup & Configuration – dfhjvmpr.props

- **ibm.jvm.shareable.application.class.path=/usr/lpp/ims/imsjava91/samples.jar**





DB2 Stored Procedure Support - Setup

- **Create the JAVAENV sequential dataset with the following attributes:**

- Organization . . . : PS
- Record format . . . : VB
- Record length . . . : 1028
- Block size : 6144

- **Set the following properties inside JAVAENV:**

- CLASSPATH – must point to application location
- DB2_HOME – must point to DB2 installation
- JAVA_HOME – must point to Java installation
- LIBPATH – must point to IMS Java shared library (libJavTDLI.so)
- TMSUFFIX – must point to IMS Java class library (imsjava.jar)



DB2 Stored Procedure Support - Setup

- **Submit the JCL job to startup the WLM-established address space**
 - This is the address where the Stored Procedure will run
- **Submit the JCL job to define the DB2 Stored Procedure to SYSIBM.SYSROUTINES**
- **Create a DB2 plan for the client program**
 - The DB2 plan contains date and time stamps, and all the details about how DB2 data is to be handled by programs
- **Create db2sqljjdbc.properties file that contains the following:**
 - DB2SQLJSSID=yourDB2ID
 - DB2SQLJPLANNAME=DB2APPCL
 - DB2SQLJATTACHTYPE=RRSAF
 - DB2SQLJDBRMLIB=DB2HLQ.SDSNDBRM
- **The above properties file is pointed to by the USS environment variable:**
 - DB2SQLJPROPERTIES
 - This variable has to be manually created



DB2 Interoperability for Java Dependent Regions

- **Create the SSM member.**

- **It is sent in as a parameter during IMS IPL process**

- **This tells IMS to attach to RRSAF subsystem**

- **Name the member as follows: (XXXXYYYY)**

- The first four characters would be the IMS ID

- The next four characters is of your choice

- **The contents of this member are as follows:**

- SST=DB2

- SSN=<db2SubsystemName>

- COORD=RRS

- ✓ As of now, from the same control region applications cannot use both RRS and ESAF





DB2 Interoperability for Java Dependent Regions

- **Set the LIBPATH (in the member DFSJVMEV) to point to the DB2 library (apart from libJavTDLI.so)**

- LIBPATH=/usr/lpp/db2/db270/lib:/usr/lpp/ims/imsjava91

- **Set the trusted middleware classpath to point to the DB2 JDBC driver:**

- **For the old DB2 JDBC driver:**

- Dibm.jvm.trusted.middleware.class.path=<PATH>/db2jdbcclasses.zip:<PATH>
where <PATH> could be /usr/lpp/db2/db2710/class

- **For the new JCC driver:**

- Dibm.jvm.trusted.middleware.class.path=<PATH>/db2jcc.jar:<PATH>/sqlj.zip:<PATH>
>/db2jcc_license_cisuz.jar:<PATH>/db2jcc_javax.jar
where <PATH> could be /usr/lpp/ims/db2/jcc/classes



DB2 Interoperability for Java Dependent Regions

- **Add the following DD card into the Java Dependent Region startup JCL:**

- DFSDB2AF DD DSN=*HLQ*.SDSNLOAD,DISP=SHR
DD DSN=*HLQ*.SDSNLOD2,DISP=SHR

- This is required to set up the IMS environment to interoperate with DB2

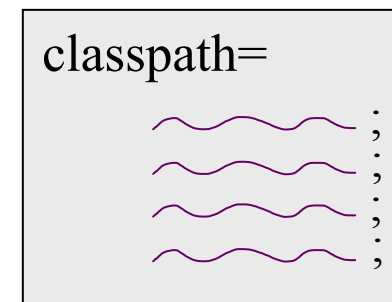
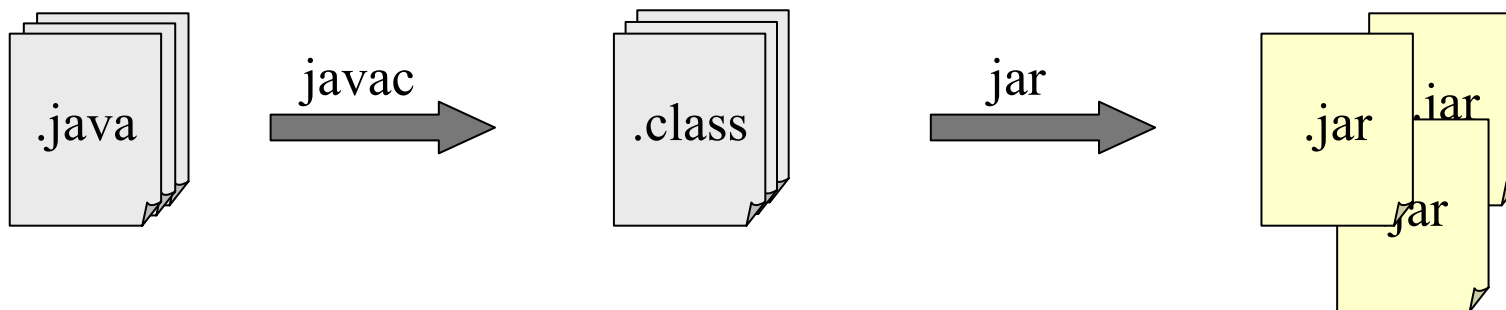
- **During the IMS IPL, make sure RRS is active**

- IPL IMS with Parm1='RRS=Y,SSM=XXXXYYYY'



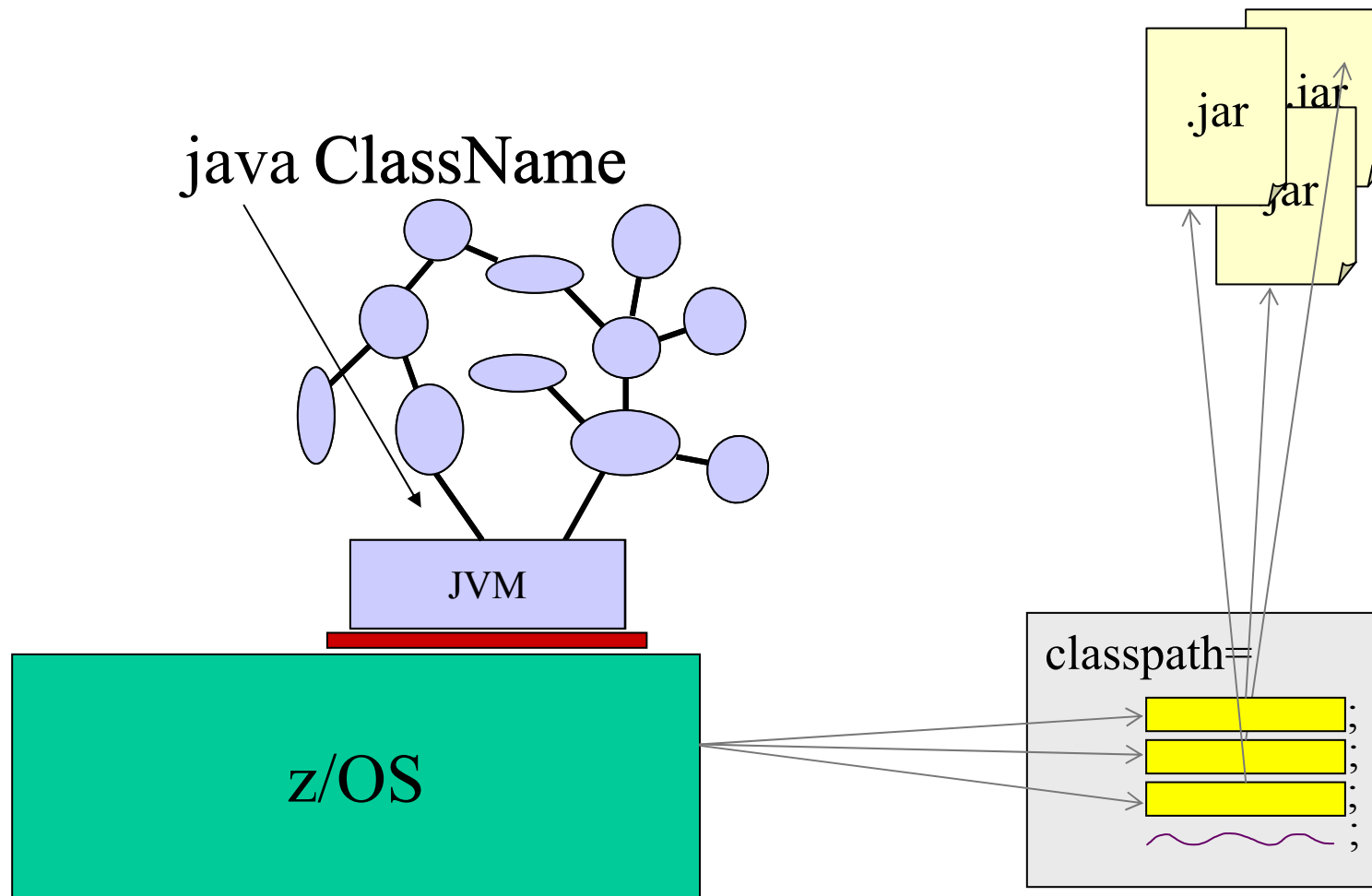


Java Compilation





Java Virtual Machine Runtime





Persistent Reusable JVM Runtime

