

# Setting Up JDBC

**At the end of this module you will be able to:**

- ✓ Describe the high level architecture of JDBC
- ✓ List the four driver types and those provide by WLS
- ✓ Describe and configure Data Sources
- ✓ Use the Administration Console to manage JDBC resources

## 1. Overview of JDBC

- High Level Architecture of JDBC and the Driver Model
- Four Different Driver Types
- Differences Between Two-tier and Multi-tier Models
- Drivers Provided by WebLogic Server

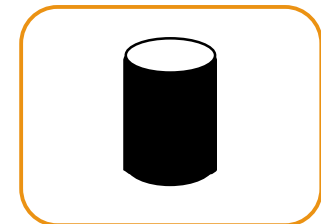
## 2. Data Sources

## 3. Monitoring and Testing Data Sources

# What Is JDBC?

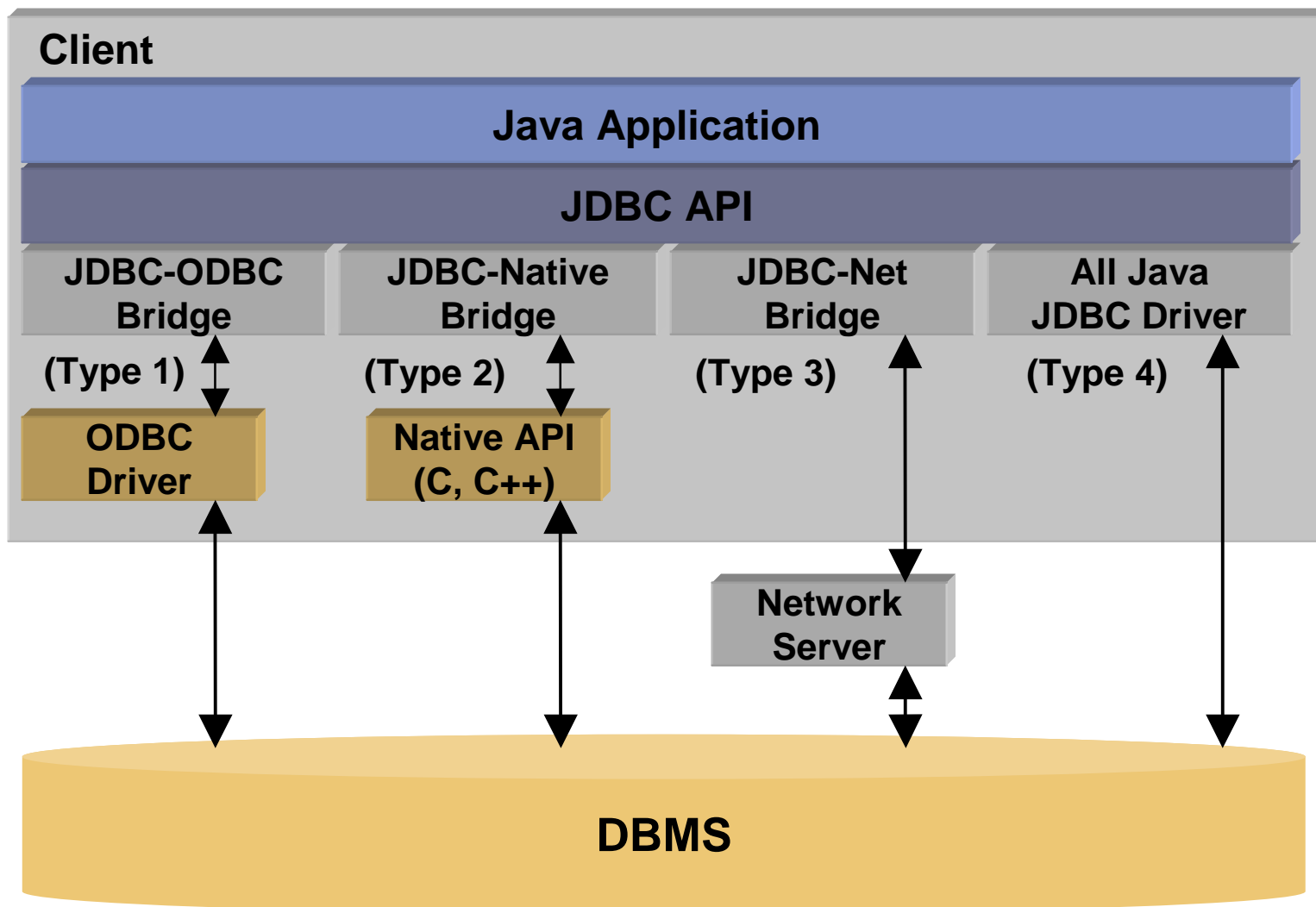


- ▶ JDBC is an API for accessing databases in a uniform way.
- ▶ JDBC provides:
  - Platform independent access to databases
  - Location transparency
  - Transparency to proprietary database issues
  - Support for both two-tier and multi-tier models for database access



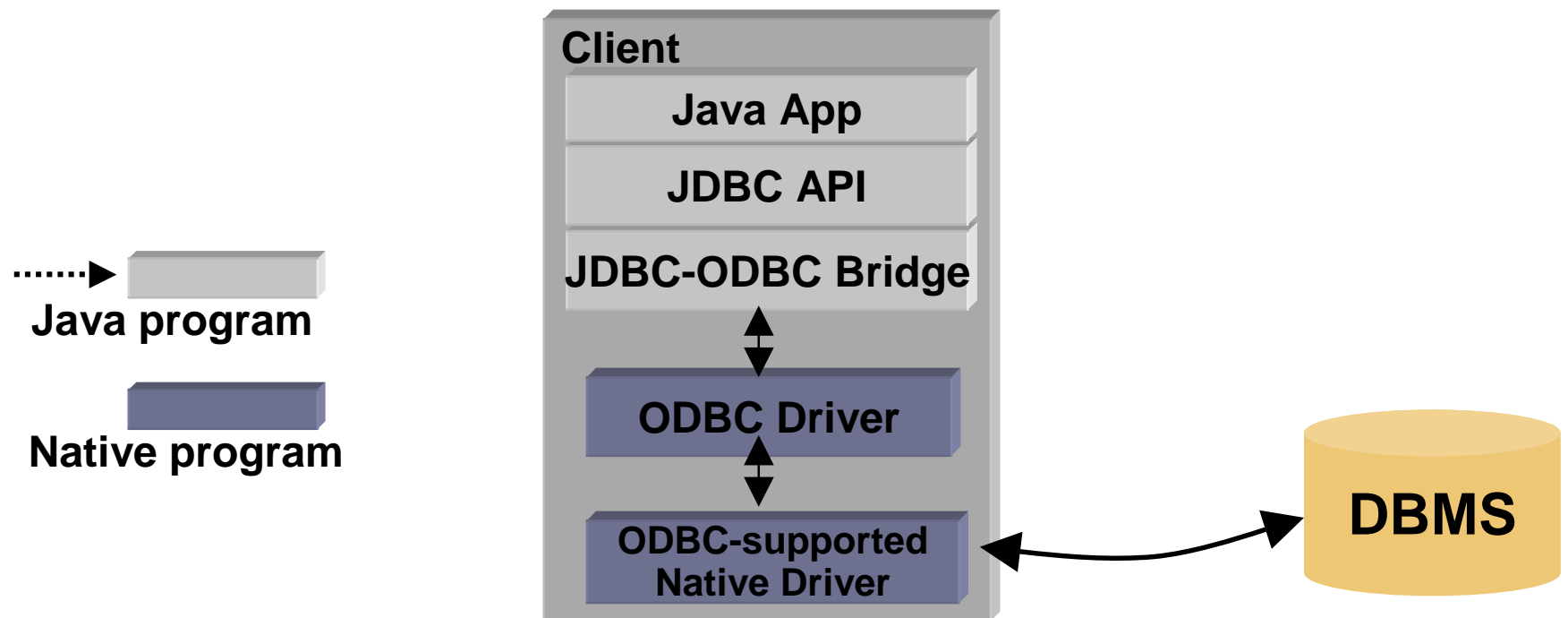
**Data**

# JDBC Architecture



# Type 1 Driver

- ▶ The Type 1 driver:
  - Is a JDBC-ODBC bridge
  - Usually runs on Windows
  - Requires ODBC driver to be installed on client machine

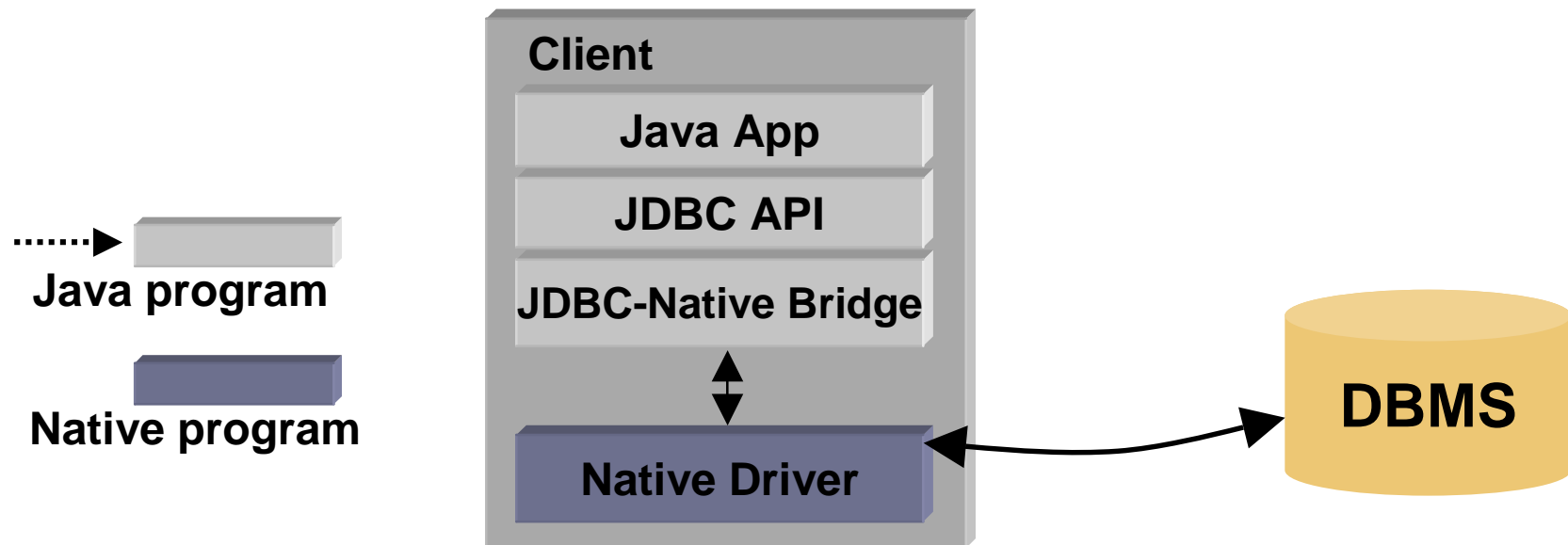


# Type 2 Drivers

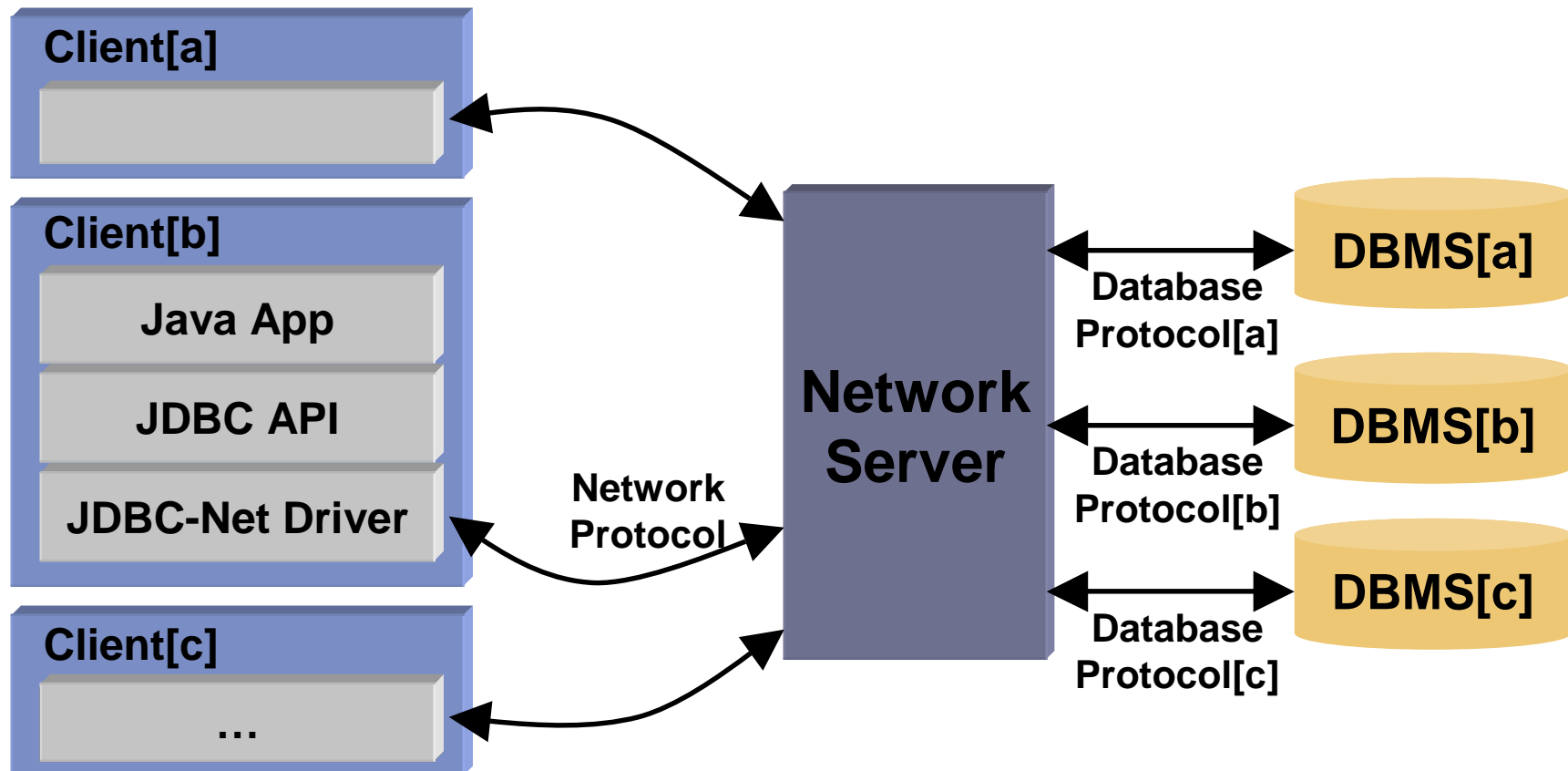


## ► The Type 2 driver:

- Requires a native driver to be already installed on the client machine
- The driver converts JDBC calls to native API calls of the database



# Type 3 Drivers...

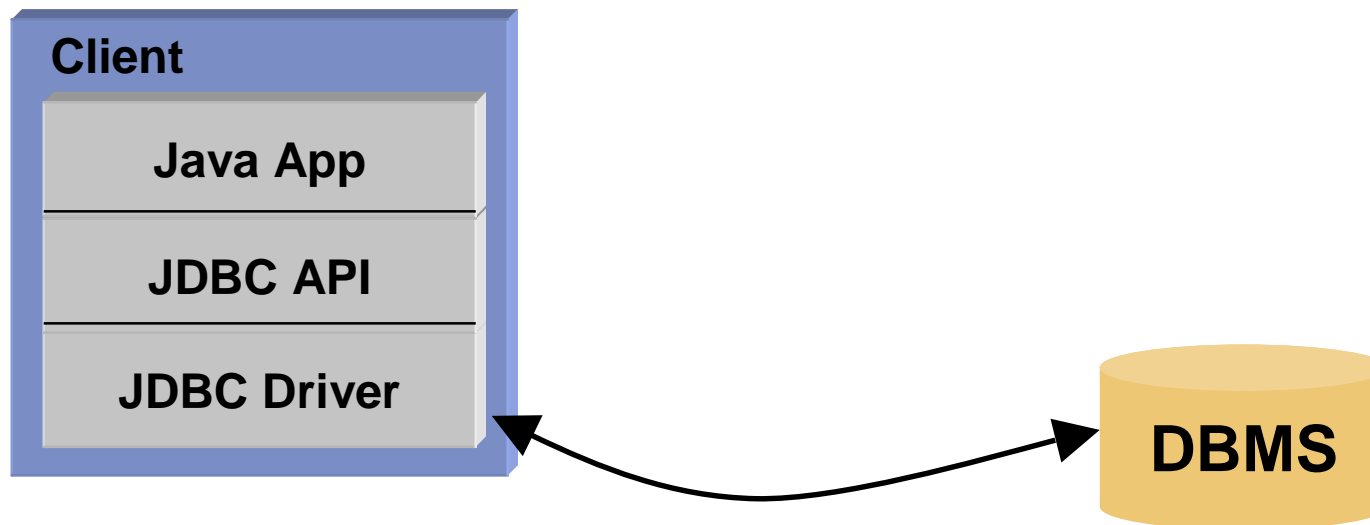


- 
- Caching
- getConnection()
- Free Connection
- Connection in Use
- DBMS



# Type 4 Drivers

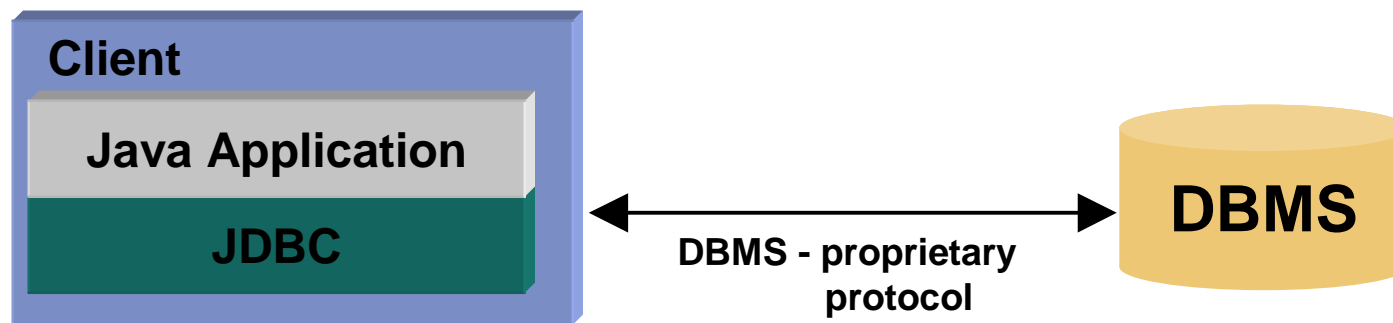
- ▶ Type 4 drivers are ‘all-Java’ driver implementations that *do not* require client side configuration.



# Two-Tier Architecture



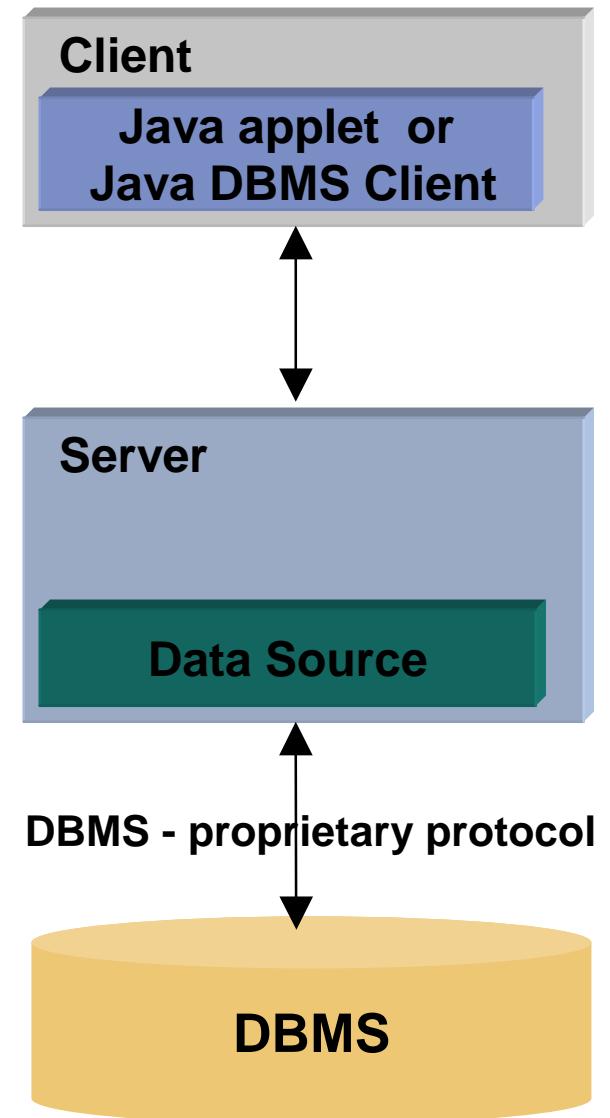
- ▶ In the two-tier model a Java application communicates directly with the DBMS.
- ▶ A JDBC driver is needed that can communicate directly with the DBMS.
- ▶ This is a client/server configuration.



# Multi-Tier Architecture



- ▶ In the multi-tier model, commands are sent to a "middle tier" of services, which then sends the commands to the DBMS.
- ▶ The DBMS processes the commands and sends the results back to the middle tier, which then sends them to the client.



# WebLogic Server Two-Tier Drivers



- The following table shows the two-tier drivers for vendor specific platforms that WebLogic Server provides.

Driver	Type
DB2	4
Informix	4
SQL Server	4
Oracle	4
Sybase	4
MySQL	4

# Choosing the Correct Driver



- ▶ Choosing the correct driver can have significant impact on performance.
- ▶ For two-tier applications use the type 1, 2 or 4 driver specific to the DBMS you are using.
- ▶ For multi-tier applications use:
  - Data Source lookup in a client class
  - A type 1, 2 or 4 driver on the server, specific to the DBMS you are using
  - XA driver where transaction support is required

# Section Review



## In this section we discussed:

- ✓ JDBC high level architecture
- ✓ WebLogic Server driver types and support
- ✓ Two-tier versus multi-tier architectures
- ✓ Selecting a driver type



# Road Map



## 1. Overview of JDBC

## 2. Data Sources

- Describe a Data Source and How It Works
- Use the Administration Console to Create a Data Source

## 3. Monitoring and Logging

# What Is a Data Source?



- ▶ A Data Source object provides a way for a JDBC client to obtain a database connection from a connection pool.
- ▶ A Data Source:
  - Is stored in the WLS JNDI tree
  - Can support transactions
  - Is associated with a connection pool



# What Is a Connection Pool?



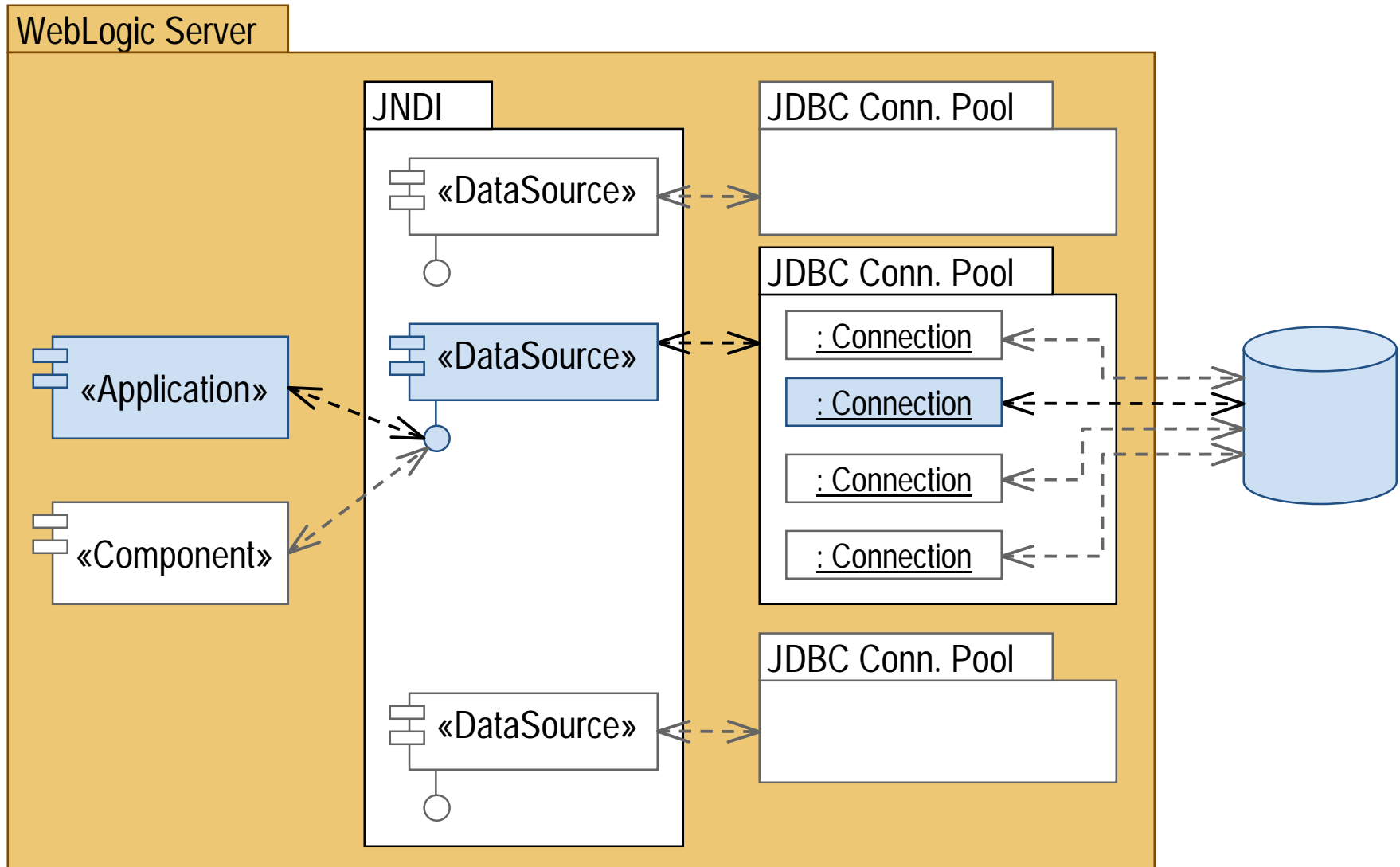
- ▶ A connection pool is a group of ready-to-use database connections associated with a Data Source.
- ▶ Connection pools:
  - Are created at WebLogic Server startup
  - Can be administered using the Administration Console
  - Can be dynamically resized to accommodate increasing load

# Benefits of Data Sources + Connection Pools



- ▶ Some advantages of this approach are:
  - Time and overhead are saved by using an existing database connection
  - Connection information is managed in one location in the Administration Console
  - The number of connections to a database can be controlled
  - The DBMS can be changed without the application developer having to modify underlying code
- ▶ A connection pool allows an application to “borrow” a DBMS connection.

# JDBC Data Source Architecture



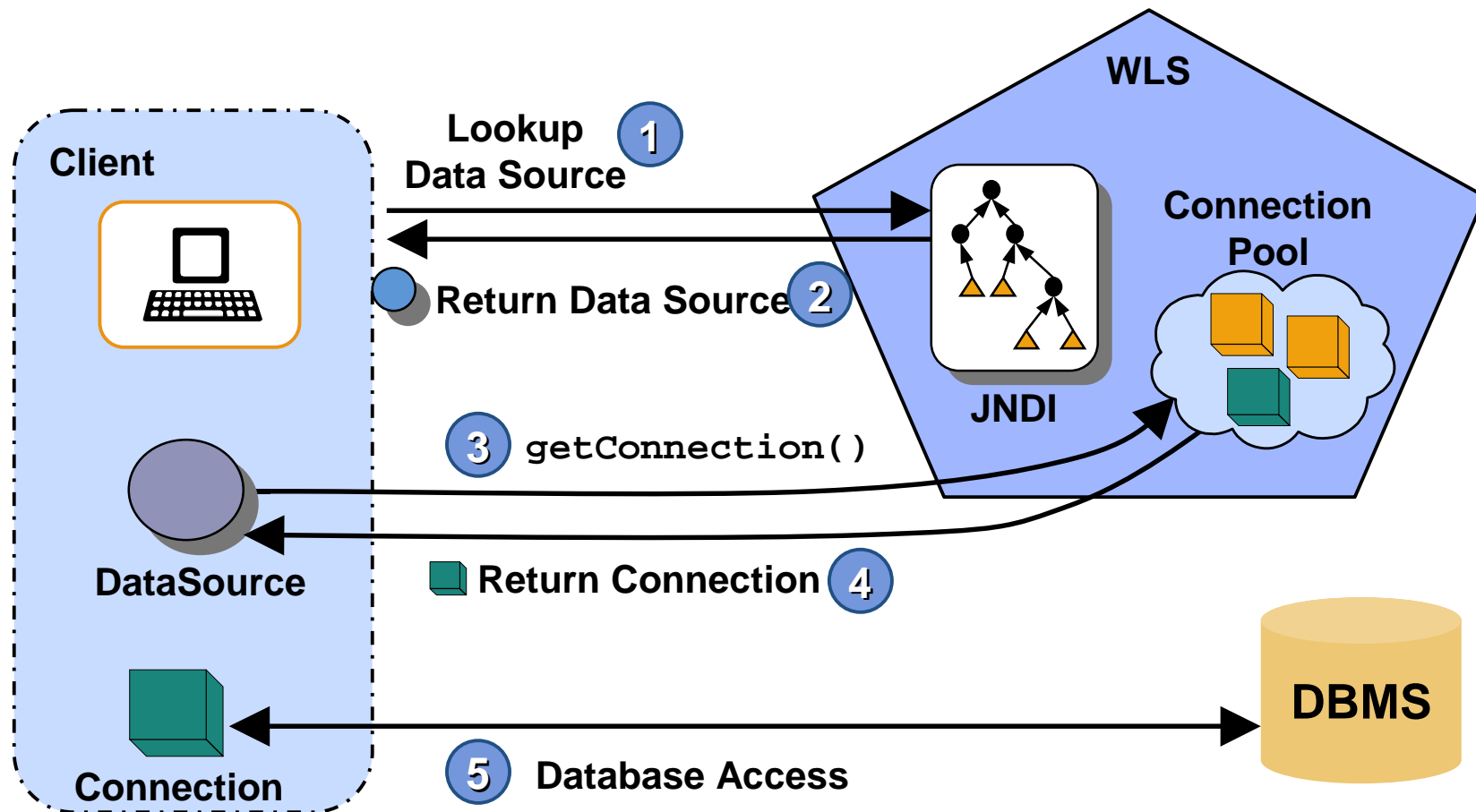
# Modular Configuration and Deployment of JDBC Resources



- ▶ JDBC configurations in WLS are stored in XML documents:
  - All JDBC configurations must conform to the new `weblogic-jdbc.xsd` schema.
  - IDEs and other tools can validate JDBC modules based on the schema.
- ▶ You create and manage JDBC resources either as system modules or as application modules.
- ▶ JDBC *application modules* are a WLS-specific extension of J2EE modules and can be deployed either within a J2EE application or as stand-alone modules.

# How Data Sources Are Used

- ▶ A client retrieves a Data Source through JNDI look up and uses it to obtain a database connection.



# Creating a JDBC Data Source...



**WEBLOGIC SERVER**  
ADMINISTRATION CONSOLE

**Change Center**  
  
View changes and restarts  
  
Pending changes exist. They must be activated to take effect.  
  
**Activate Changes**  
  
Undo All Changes

**Domain Structure**  
  
wl\_server

- Environment
- Deployments
- Services
  - Messaging
  - JDBC**
    - Data Sources** **1**
      - Multi Data Sources
      - Data Source Factories
    - Persistent Stores
    - Path Services
    - Foreign JNDI Providers
    - Work Contexts
    - XML Registries
    - XML Entity Caches
    - jCOM
    - Mail Sessions
    - File T3
    - JTA
  - Security Realms
- Interoperability
- Diagnostics

**How do I...**  
  

- Create JDBC data sources
- Delete JDBC data sources

Welcome, weblogic

Connected to: wl\_server

[Home](#) [Log Out](#) [Preferences](#) [Help](#) [Ask](#)

Home > ???Summary of Services: JDBC??? > **Summary of JDBC Data Sources**

**Summary of JDBC Data Sources**

A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool of JDBC connections. Applications can look up a data source on the JNDI tree and then borrow a database connection from a data source.

This page summarizes the JDBC data source objects that have been created in this domain.

[Customize this table](#)

**2**

**Data Sources**

New Delete

Showing 1 - 4 of 4 Previous | Next

<input type="checkbox"/>	Name	JNDI Name	Targets
<input type="checkbox"/>	examples-demo	examples-dataSource-demoPool	examplesServer
<input type="checkbox"/>	examples-demoXA	examples-dataSource-demoXAPool	examplesServer
<input type="checkbox"/>	examples-demoXA-2	examples-demoXA-2	examplesServer
<input type="checkbox"/>	examples-oracleXA	examples-dataSource-oracleXAPool	

New Delete

Showing 1 - 4 of 4 Previous | Next

Setting up JDBC-22

© 2006 BEA Systems, Inc.

374

# Creating a Data Source: Properties



## Create a New JDBC Data Source

Back

Next

Finish

Cancel

### JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

What would you like to name your new JDBC data source?

3

Name:

JDBC Data Source-0

What JNDI name would you like to assign to your new JDBC Data Source?

4

JNDI Name:

jdbc\_datasource-0

What database type would you like to select?

5

Database Type:

PointBase

What database driver would you like to use to create database connections?

6

Database Driver:

\*PointBase's Driver (Type 4) Versions:4.X,5.X

7

Back

Next

Finish

Cancel

# Creating a Data Source: XA Options



## Create a New JDBC Data Source

Back

Next

Finish

Cancel

### Transaction Options

You have selected non-XA JDBC driver to create database connection in your new data source.

Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.

8

☒ **Supports Global Transactions**

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the Logging Last Resource (LLR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.

9

☐ **Logging Last Resource**

Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application can tolerate heuristic conditions.

10

☐ **Emulate Two-Phase Commit**

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the one-phase commit transaction processing. With this option, no other resources can participate in the global transaction.

11

☒ **One-Phase Commit**

Back

Next

Finish

Cancel

12



# Creating a Data Source: Connection Properties



## Create a New JDBC Data Source

Back

Next

Finish

Cancel

### Connection Properties

Define Connection Properties.

What is the name of database you would like to connect to?

13

Database Name:

What is the name or IP address of the database server?

14

Host Name:

What is the port on the database server used to connect to the database?

15

Port:

What database account user name do you want to use to create database connections?

16

Database User Name:

What is the database account password to use to create database connections?

17

Password:

Confirm Password:

Back

Next

Finish

Cancel

18

# Creating a Data Source: Test Database Connections & Select Targets



**19** {

### Create a New JDBC Data Source

Test Configuration | Back | **Next** | Finish | Cancel

#### Test Database Connection **20**

Test the database availability and the connection properties you provided.

What is the full package name of JDBC driver class used to create database connections in the connection pool? (Note that this driver class must be in the classpath of any server to which it is deployed.)

**Driver Class Name:**

What is the URL of the database to connect to? The format of the URL varies by JDBC driver.

**URL:**

**21** {

### Create a New JDBC Data Source

Back | Next | **Finish** | Cancel

#### Select Targets **22**

You can select one or more targets to deploy your new JDBC data source. If you don't select a target, the data source will be created but not deployed. You will need to deploy the data source at a later time.

Servers
<input checked="" type="checkbox"/> examplesServer

Back | Next | Finish | Cancel

# Configuring a Connection Pool...



## Data Sources

New	Delete	Showing 1 - 5 of 5 Previous   Next	
<input type="checkbox"/>	Name	JNDI Name	Targets
<input type="checkbox"/>	examples-demo	examples-dataSource-demoPool	examplesServer
<input type="checkbox"/>	examples-demoXA	examples-dataSource-demoXAPool	examplesServer
<input type="checkbox"/>	examples-demoXA-2	examples-demoXA-2	examplesServer
<input type="checkbox"/>	examples-oracleXA	examples-dataSource-oracleXAPool	
<input type="checkbox"/>	JDBC Data Source-0	jdbc_datasource-0	examplesServer
New	Delete	Showing 1 - 5 of 5 Previous   Next	

## Settings for examples-demoXA

Configuration	Targets	Monitoring	Control	Security	Notes
---------------	---------	------------	---------	----------	-------

General

Connection Pool

Save

Applications get a database connection from a data source by looking up the data source on the Java Naming and Directory Interface (JNDI) tree and then requesting a connection. The data source provides the connection to the application from its pool of database connections.

This page enables you to define general configuration options for this JDBC data source.

# Configuring Settings of a Connection Pool...



Configuration **Targets** Monitoring Control Security Notes

General **Connection Pool**

Save

The connection pool within a JDBC data source contains a group of JDBC connections that applications reserve, use, and then return to the pool. The connection pool and the connections within it are created when the connection pool is registered, usually when starting up WebLogic Server or when deploying the data source to a new target.

Use this page to define the configuration for this data source's connection pool.



**URL:**

`jdbc:pointbase:server://localhost/demo`

The URL of the database to connect to. The format of the URL varies by JDBC driver. [More Info...](#)



**Driver Class Name:**

`com.pointbase.xa.xaDataSource`

The full package name of JDBC driver class used to create the physical database connections in the connection pool. (Note that this driver class must be in the classpath of any server to which it is deployed.) [More Info...](#)



**Properties:**






```
user=examples
DatabaseName=jdbc:pointbase:server://localh
```



The list of properties passed to the JDBC driver that are used to create physical database connections. For example: `server=dbserver1`. List each property=value pair on a separate line. [More Info...](#)

# ...Configuring Settings of a Connection Pool



 Password:	<input type="password" value="XXXXXXXXXXXXXXXXXXXX"/>	The password attribute passed to the JDBC driver when creating physical database connections. <a href="#">More Info...</a>
Confirm Password:	<input type="password" value="XXXXXXXXXXXXXXXXXXXX"/>	Confirm your Password. <a href="#">More Info...</a>
Initial Capacity:	<input type="text" value="2"/>	The number of physical connections to create when creating the connection pool. <a href="#">More Info...</a>
Maximum Capacity:	<input type="text" value="10"/>	The maximum number of physical connections that this connection pool can contain. <a href="#">More Info...</a>
Capacity Increment:	<input type="text" value="1"/>	The number of connections created when new connections are added to the connection pool. <a href="#">More Info...</a>
 Statement Cache Type:	<input type="text" value="LRU"/> 	The algorithm used for maintaining the prepared statements stored in the statement cache. <a href="#">More Info...</a>
Statement Cache Size:	<input type="text" value="10"/>	The number of prepared and callable statements stored in the cache. (This may increase server performance.) <a href="#">More Info...</a>
 Advanced		
<input type="button" value="Save"/> 		

# Connection Pool Checklist



- ▶ You can modify a connection pool after the Data Source has been created
- ▶ Before modifying a connection pool, you should understand:
  - The JDBC URL of the database
  - The connection properties used to authenticate a user or optionally configure the driver
- ▶ Ask your DBA for:
  - The maximum number of connections your application will be allowed

# JDBC URLs



- Database locations are specified using a JDBC Uniform Resource Locator (URL).

The syntax for a JDBC URL is:

```
jdbc:<subprotocol>:<subname>
```

Where:

**<subprotocol>** identifies the database connectivity mechanism

**<subname>** identifies the data source. The subname can vary, depending on the subprotocol



# JDBC URL Examples

**Example 1.** This URL could be used to access a Pointbase database:

```
jdbc:pointbase:server://dbhost:9092/HRDATABASE
```

The subprotocol is “pointbase:server”, the subname is a location of Pointbase database named “HRDATABASE”.



**Example 2.** This URL specifies that the “oracle:thin” subprotocol should be used to connect to an Oracle database:

```
jdbc:oracle:thin:@dbhost:1521:SALESINFO
```





# Connection Properties...



- ▶ Connection properties are:
  - Key/value pairs
  - Used to configure JDBC connections and are passed to the driver during connection setup

**For a complete list see your driver documentation!**

## ...Connection Properties



- ▶ A partial list of connection properties for the supplied drivers:

Driver	Some Connection Properties
Oracle	User, Password, ServerName, ServiceName, PortNumber
Sybase	User, Password, ServerName, DatabaseName, PortNumber
MSSQL	User, Password, ServerName, DatabaseName, PortNumber
Informix	User, Password, ServerName, DatabaseName, PortNumber
Pointbase*	cache.size, crypto. communication, database.home, database.pagesize

# Multi Data Sources...

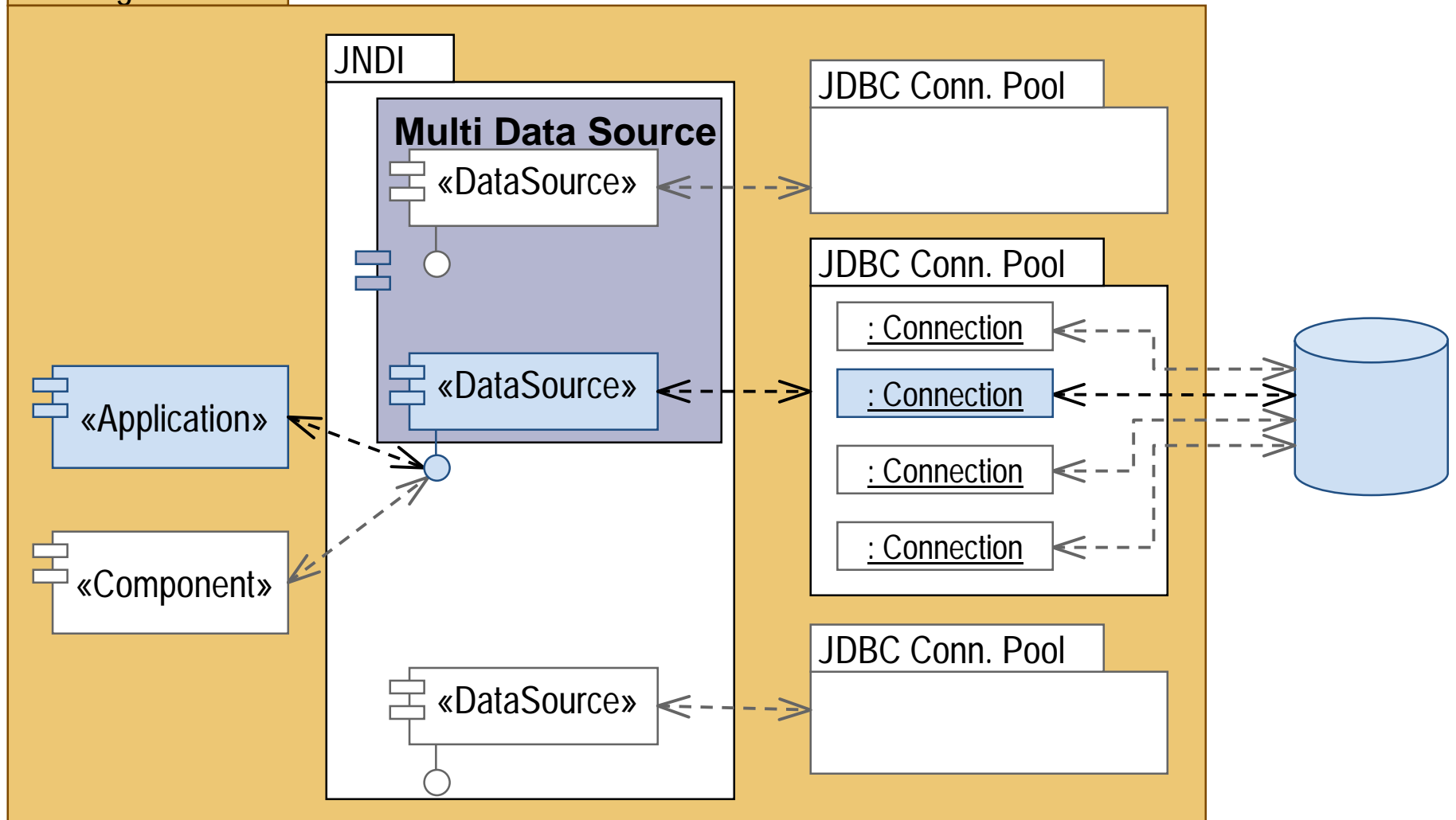


- ▶ Multi Data Source:
  - Is an abstraction around a group of data sources
  - Determines which data source to use to satisfy the request depending on the algorithm selected in the multi data source configuration:
    - Load balancing
    - Or
    - Failover
  - Are bound to the JNDI tree
- ▶ XA Support for Multi Data Sources
  - The WLS JDBC supports using Multi Data Sources in XA transactions.
  - You can configure the data sources contained within the Multi Data Source to use XA JDBC drivers.

# ... Multi Data Sources



WebLogic Server



# Section Review



## In this section we discussed:

- ✓ Data Source definition and workings
- ✓ Data Source management with the Administration Console
- ✓ Multi Data Sources



# Road Map



1. Overview of JDBC
2. Data Sources
3. **Monitoring & Testing Data Sources**
  - Monitoring
  - Testing

# Monitoring Data Sources: Statistics



- The Administration console provides two types of data source monitoring: statistics and testing.

Settings for examples-demoXA

Configuration Targets **Monitoring** Control Security Notes

Statistics Testing

This page displays statistics associated with this JDBC data source. Use this page to monitor the activity of the data source.

Customize this table

Filter

Filter by Column: Server

View

Column Display:

Number of rows displayed per page: 10

Apply Reset

Available

Chosen

Server

Active Connections Average Count

Active Connections Current Count

Active Connections High Count

Connection Delay Time

Connections Total Count

Current Capacity

Curr Capacity High Count

Enabled

Failed Reserve Request Count

Failures To Reconnect Count

Highest Num Available

Leaked Connection Count

Num Available

Num Unavailable

Server	Active Connections Average Count	Active Connections Current Count	Active Connections High Count	Connection Delay Time	Connections Total Count	Current Capacity	Curr Capacity High Count	Enabled	Failed Reserve Request Count	Failures To Reconnect Count	Highest Num Available	Leaked Connection Count	Num Available	Num Unavailable
examplesServer	0	0	1	15	2	2	2	true	0	0	2	0	2	0

# Monitoring Data Sources: Testing



- The Administration console provides a mechanism for manually testing individual data sources.

## Messages

✓ Test JDBC data sources successful.

## Settings for examples-demoXA


[Configuration](#) [Targets](#) [Monitoring](#) [Control](#) [Security](#) [Notes](#)

[Statistics](#) [Testing](#)

Use this page to test database connections in this JDBC data source.

► [Customize this table](#)

### Test Data Source

Test Data Source		Showing 1 - 1 of 1	Previous   Next
Server	State		
 examplesServer	Running		

Test Data Source

Showing 1 - 1 of 1 Previous | Next



# Section Review



## In this section we discussed:

- ✓ Monitoring statistics and the testing of an individual data source.

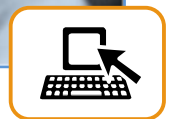


## Configuring Data Sources

- ▶ In this lab you will work with configuring and monitoring Data Sources.
- ▶ Ask the instructor for any clarification.
- ▶ The instructor will determine the stop time.



Lab Exercise



# Module Review



## In this module we discussed:

- ✓ JDBC high level architecture
- ✓ WebLogic Server provided JDBC driver types
- ✓ Data Source definition and workings
- ✓ Connection pool definition and workings
- ✓ Managing JDBC resources with the Administration Console

