

Understanding JDBC and Configuring Data Sources

Objectives

- After completing this lesson, you should be able to:
 - Configure JDBC and JDBC data sources
 - Configure data source scope
 - Contrast two-tier and multi-tier JDBC architecture
 - Configure a connection pool
 - List the benefits of connection pools
 - Describe how data sources are used
 - Deploy JDBC resources to a target
 - View the server JNDI tree
 - Complete a connection pool checklist
 - Explain the components of JDBC URLs
 - Monitor and test a data source
-

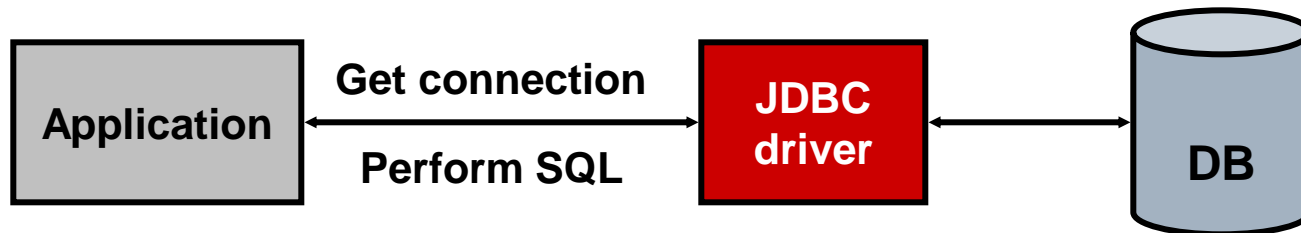
Road Map

- Overview of JDBC
 - High-level architecture of JDBC and the driver model
 - Design of a multi-tier architecture
- Data sources
- Monitoring and testing data sources



JDBC Review

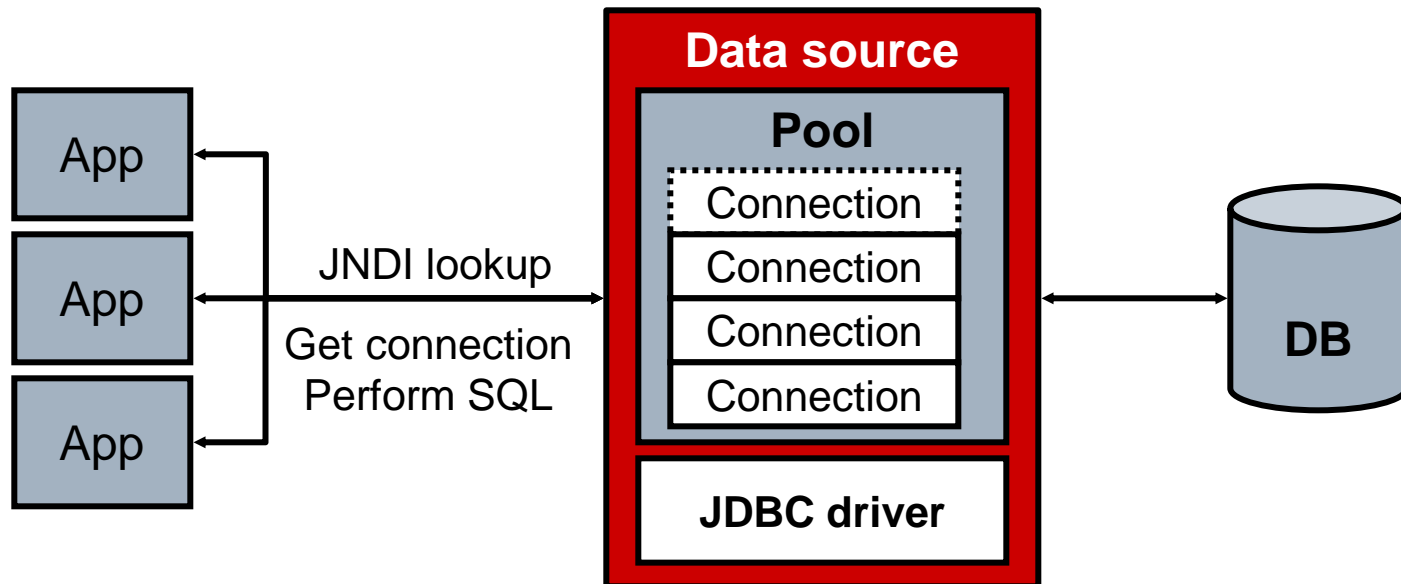
- The Java Database Connectivity (JDBC) specification:
 - Is a platform- and vendor-independent mechanism for accessing and updating a database
 - Provides transparency from proprietary vendor issues
 - Requires the use of a *driver*
- JDBC drivers are supplied by your database vendor.



JDBC Data Sources

□ Data sources:

- Enable database connectivity to be managed by the application server
- Are obtained by applications from the server's JNDI tree
- Use a dynamic pool of reusable database connections

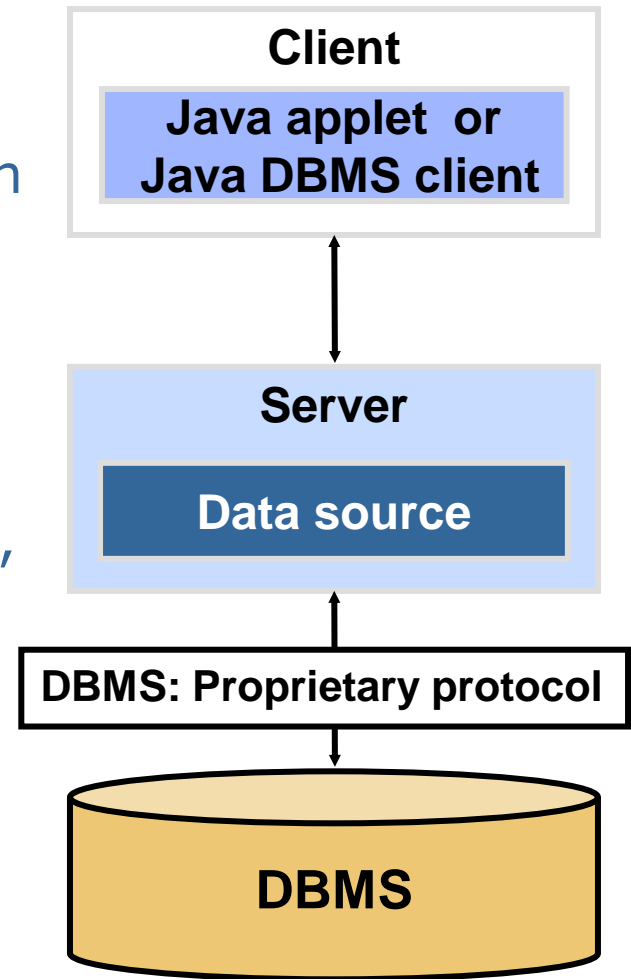


Data Source Scope

- Each data source configuration or “module” is persisted as a separate XML document.
 - Application-specific modules are:
 - Deployed as part of Java Platform, Enterprise Edition (Java EE) enterprise applications
 - Accessible only by the containing application
-

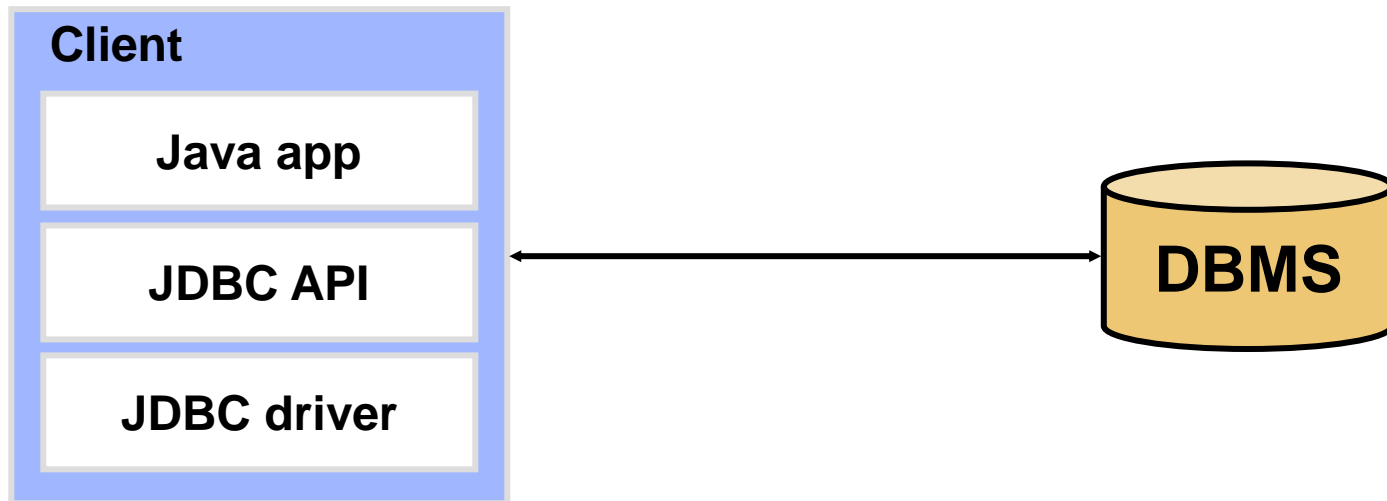
Multi-Tier Architecture

- 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.



Type 4 Drivers

- Type 4 drivers are “all-Java” driver implementations that do not require client-side configuration.



Road Map

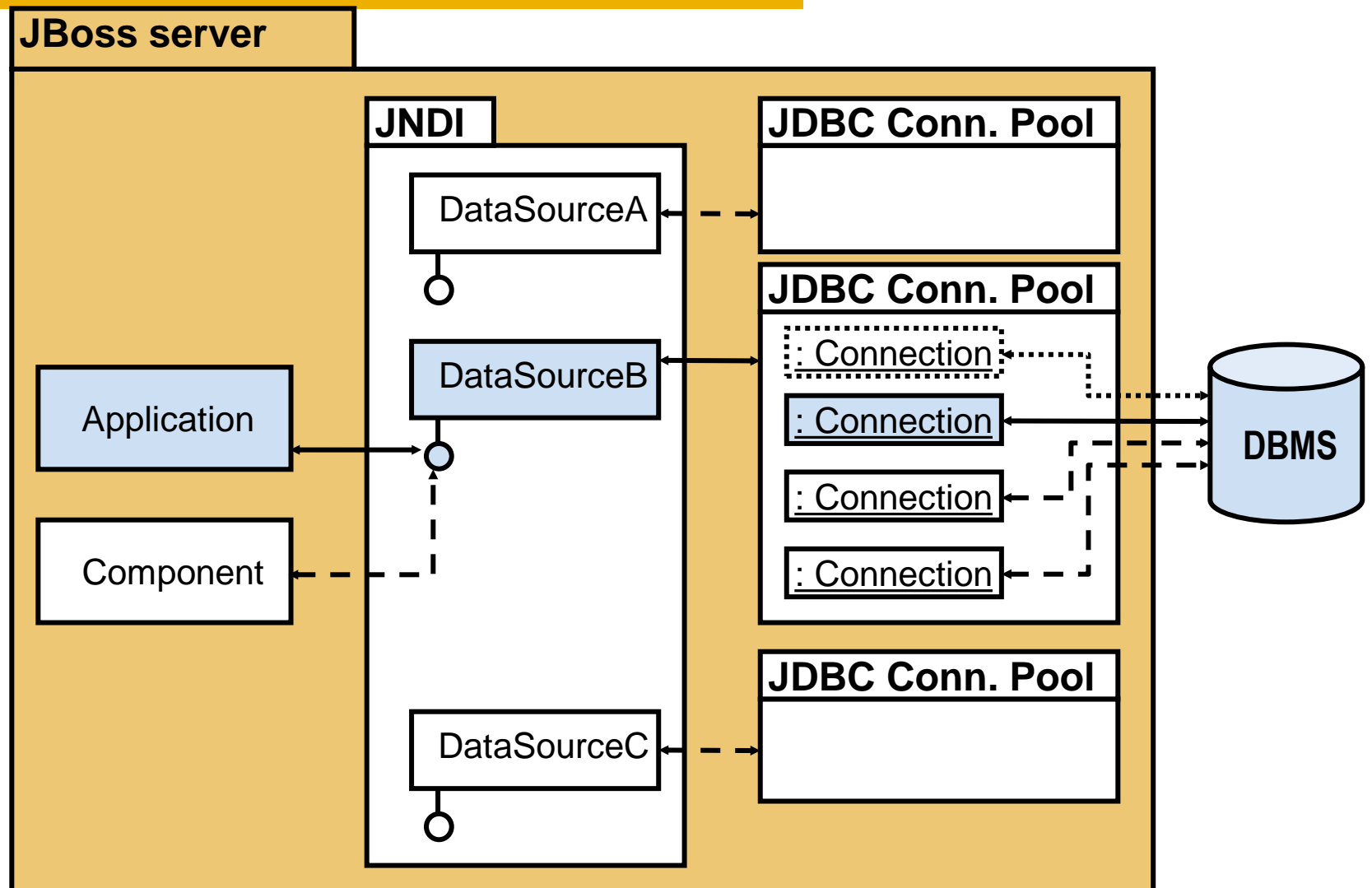
- Overview of JDBC
- Data sources
 - Describing a data source and how it works
 - Using the Administration Console to create a data source
- Monitoring and testing data sources



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 JBoss Server startup
 - Can be administered using the Administration Console
 - Can be dynamically resized to accommodate increasing or decreasing load

JDBC Connection Pooling

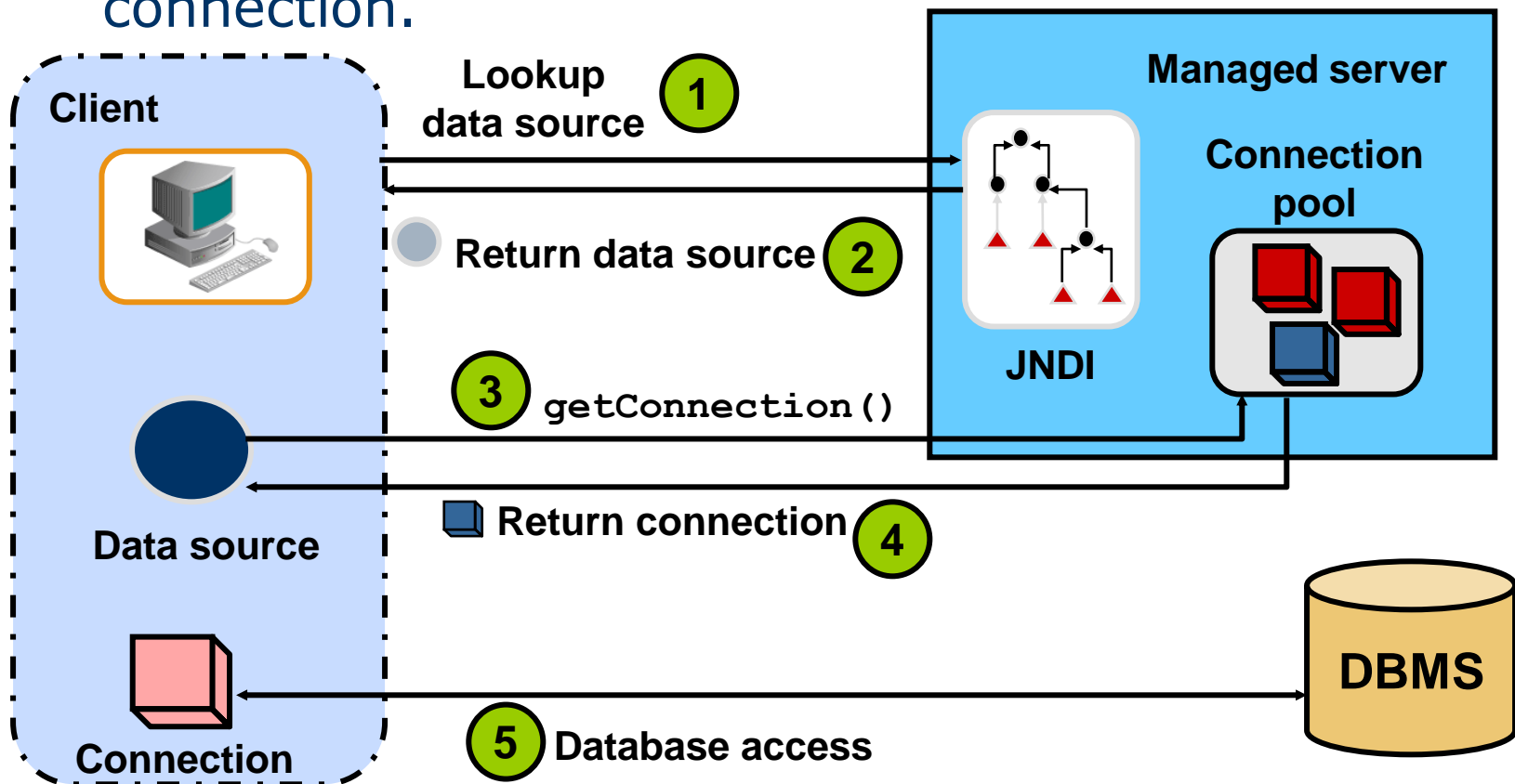


Benefits of Connection Pools

- The following are some advantages of connection pooling:
 - Connection time and overhead are saved by using an existing database connection.
 - It facilitates easier management because connection information is managed in one location.
 - The number of connections to a database can be controlled.
 - The DBMS can be changed without the application developer having to modify the underlying code.
 - A connection pool allows an application to “borrow” a DBMS connection.
-

How Data Source Connection Pools Are Used

- A client retrieves a data source through a JNDI lookup and uses it to obtain a database connection.



JDBC URLs

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

- Example 1:

- This URL specifies that the `:thin` subprotocol should be used to connect to an Database:

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

- Example 2:

- This URL can be used to access a PointBase database:

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

Connection Properties

- Are key/value pairs
- Are used to configure JDBC connections
- Are passed to the driver during connection setup

Specifying Connection Properties

- A partial list of connection properties for the some important drivers:

Driver	Some Connection Properties
	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

Road Map

- Overview of JDBC
- Data sources
- Monitoring and testing data sources
 - Monitoring
 - Testing
 - Suspend/resume



Quiz

- Which of the following is NOT an available configuration attribute for a JDBC data source?
 1. Host name
 2. Queue size
 3. Test frequency
 4. Initial capacity
 5. Capacity increment
-

Quiz

□ Client applications look up data sources from the local server's _____ tree:

1. Application
 2. Web
 3. LDAP directory
 4. JNDI
 5. System
-

Summary

- In this lesson, you should have learned how to:
 - Define JDBC high-level architecture
 - Create data source definitions
 - Create connection pool definitions
 - Manage JDBC resources using the Administration Console
-

Practice 13 Overview: Configuring JDBC Data Sources

- This practice covers the following topics:
 - Creating JDBC modules
 - Deploying JDBC modules
 - Testing JDBC modules
-