

[www.mojtabanow.info](http://www.mojtabanow.info)

# Defining Java Enterprise Edition Terminology and Architecture



ORACLE

# Objectives

After completing this lesson, you should be able to:

- Explain the motivation behind distributed systems
- List the major components of the Java Platform Enterprise Edition 5 (Java EE) specification

# Distributed Systems

- Distributed systems divide the work among several independent modules.
- The failure of a single module has less impact on the overall system, which makes the system more:
  - Available
  - Scalable
  - Maintainable



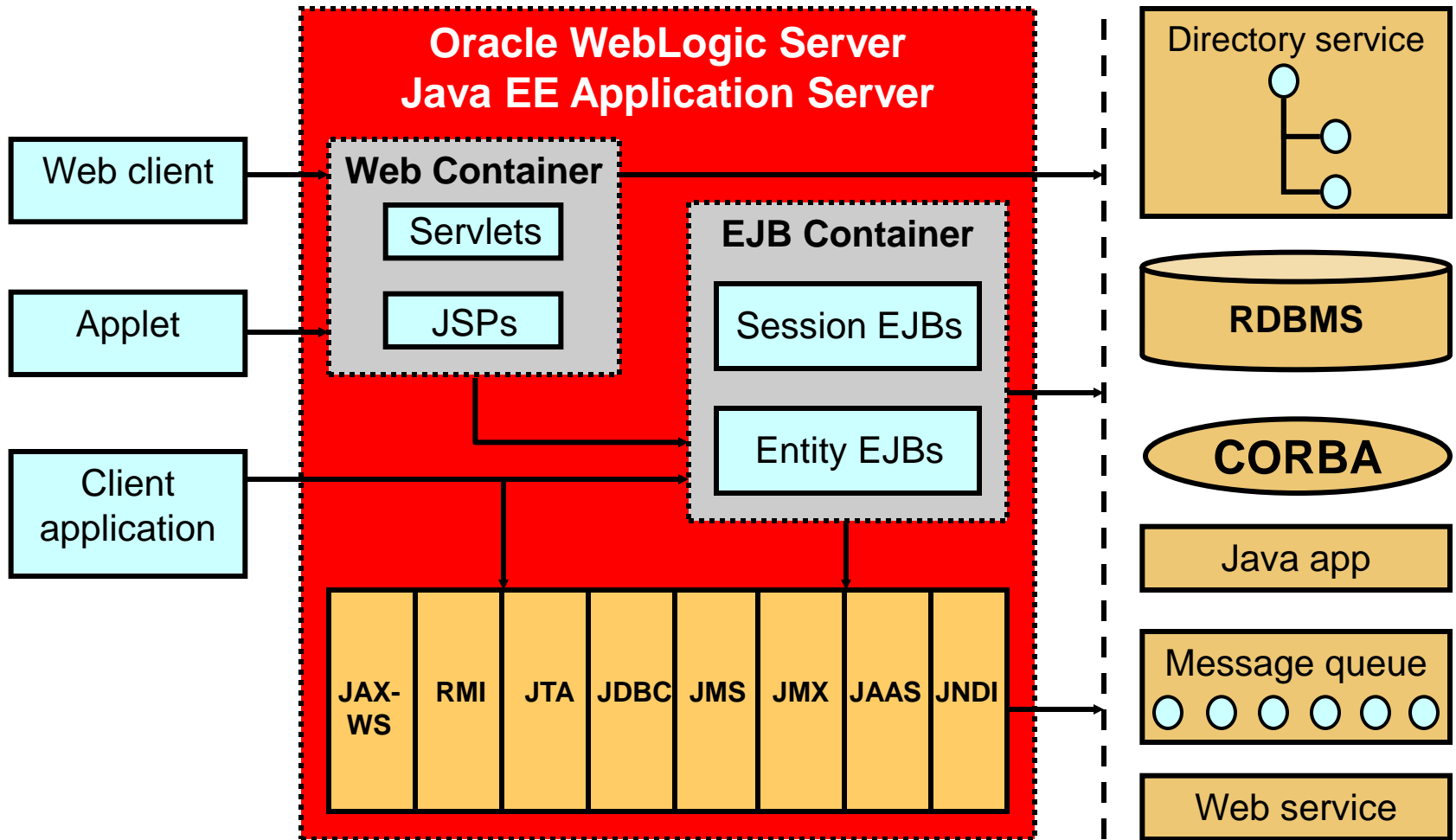
# How Standards Help

- Many advantages of distributed systems come from standards.
- Standards allow:
  - Modularization of complex hardware and software
  - A larger portion of the project costs to go toward solving business software needs

# Java EE Standard

- Java Platform Enterprise Edition 5 (Java EE) helps you to overcome distribution liabilities.
- Applications deployed with Java EE technologies are:
  - Standardized
  - Adherent to specification guidelines
  - Written in Java
  - Deployable in any compliant application server
- Java Community Process (JCP) is the oversight (custodial) process for moderating Java's future direction.
  - <http://jcp.org/en/home/index>
  - <http://jcp.org/en/introduction/faq>

# Java EE Architecture



# Java Servlets

- A servlet is a Java program that executes on the server, accepting client requests and generating dynamic responses.
- The most prevalent type of servlet is an HttpServlet that accepts HTTP requests and generates HTTP responses.
- Servlets:
  - Do not just generate HTML
  - Can also be used to generate other MIME types, such as images

# SimplestServlet.java

Creates HTML

```
package mypackage;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SimplestServlet extends HttpServlet {
    public void service(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<HTML><BODY>");
        out.println("<H1>Hello, World!</H1>");
        out.println("</BODY></HTML>");
    }
}
```



# JavaServer Pages (JSPs)

- Are HTML documents that are interwoven with Java
- Provide a dynamic response that is based on the client's request
- Provide for the separation of responsibilities between the Web presentation and the dynamic content
- Are portable (write once, run anywhere)
- Compile and run as servlets
- May include JavaServer Faces tags



# realsimple.jsp

Creates HTML

```
<!-- this is a comment -->
<HTML>
  <HEAD>
    <TITLE>My title</TITLE>
  </HEAD>
  <BODY>
    <H1>A big heading</H1>
    <P>Blah blah blah blah blah.</P>
    <% for (int i=0; i<3; i++) { %>
      <H3>Say it again, Sam.</H3>
    <% } %>
  </BODY>
</HTML>
```

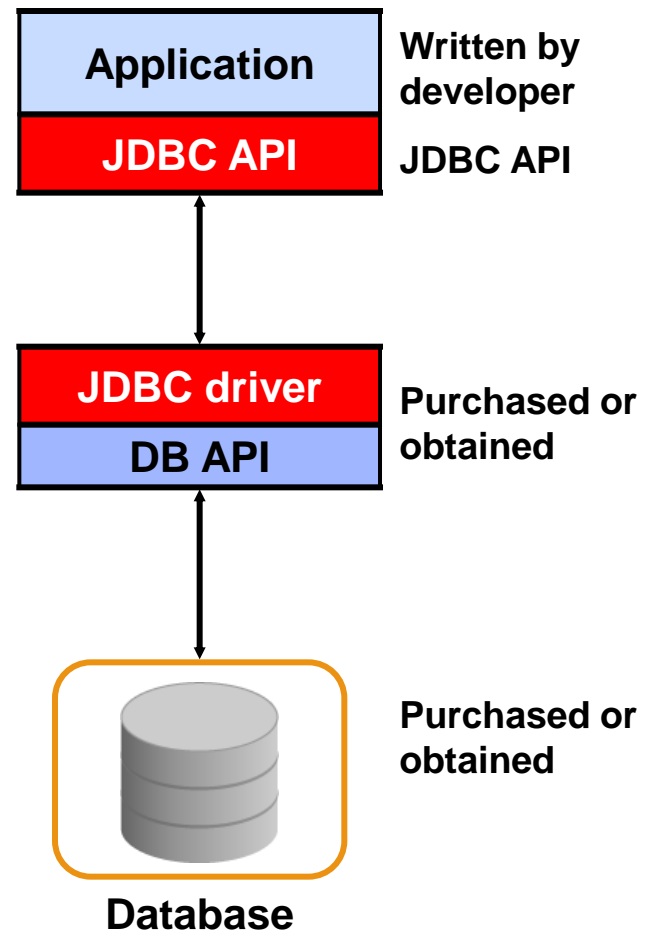
# Enterprise JavaBeans (EJBs)

- Are distributed components written in the Java programming language
- Provide distributable and deployable business services (logic) to clients
- Have well-defined interfaces
- Are reusable across application servers
- Execute within a container that provides management and control services
  - Oracle WebLogic Server 10.3.1 supports the EJB 3.0 specification.



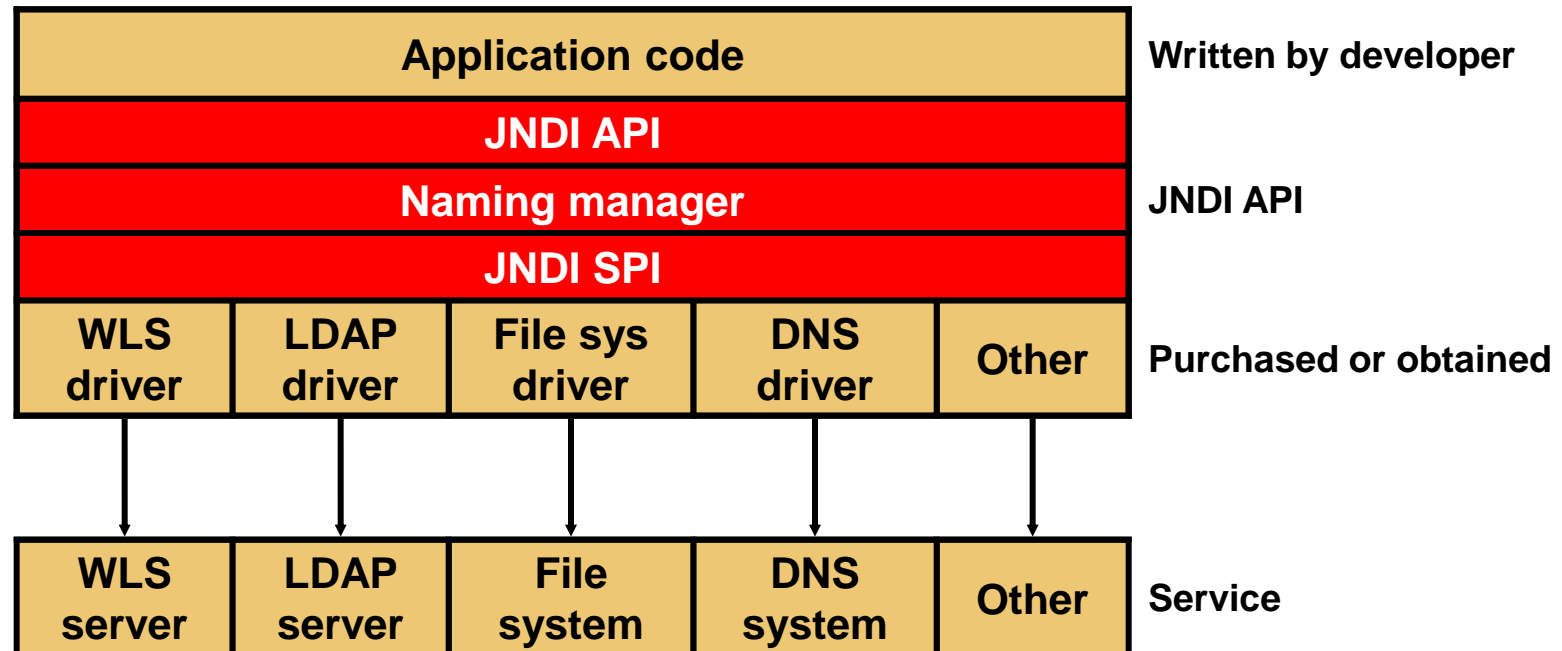
# Java Database Connectivity (JDBC)

- The standard Java interface for accessing heterogeneous databases
- The specification that defines four driver types for connecting to databases

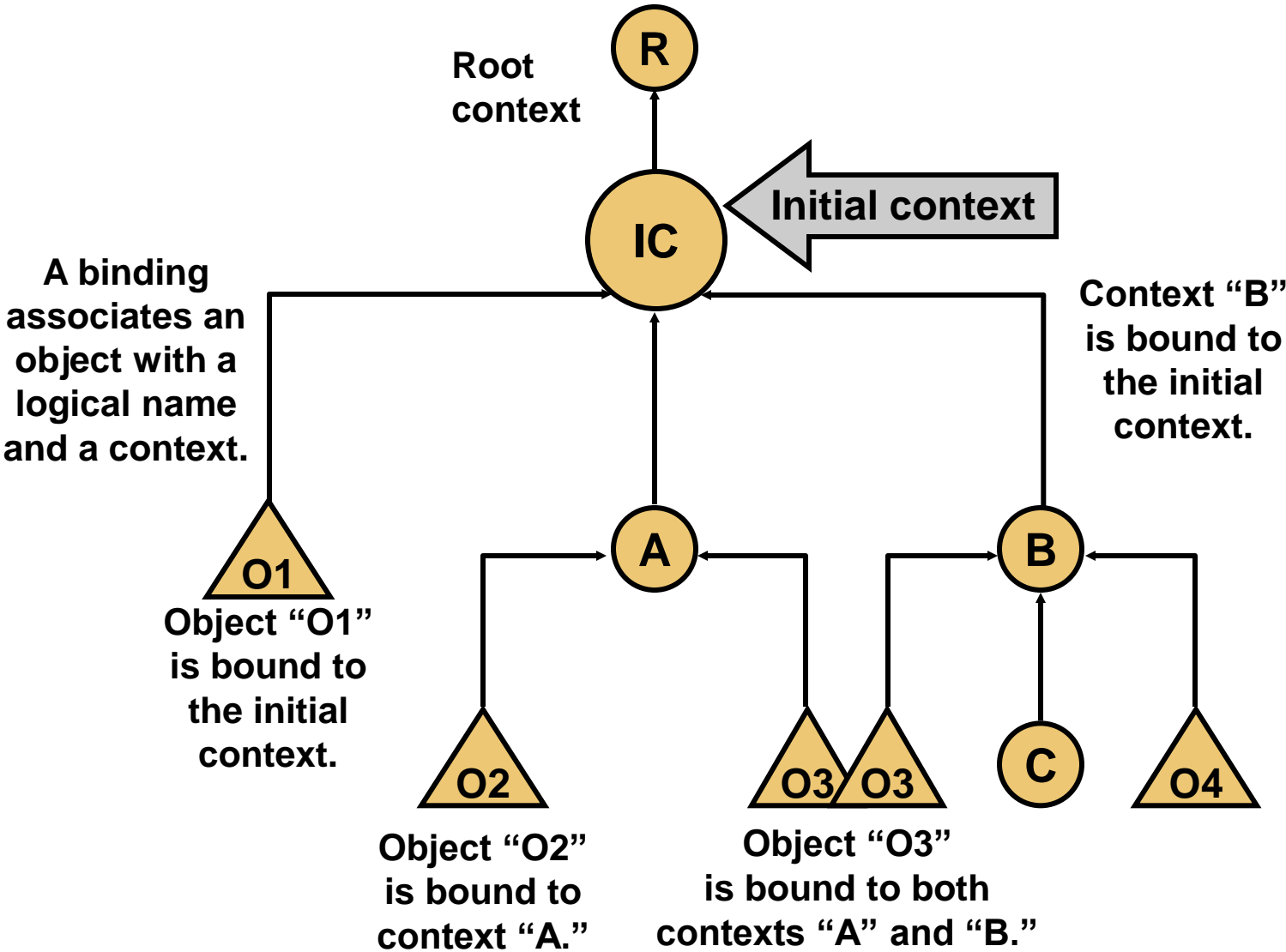


# Java Naming and Directory Interface (JNDI)

- Java API for accessing naming and directory services
- Built as a layer over DNS, LDAP, and so on



# JNDI Tree



# JNDI Contexts and Subcontexts

- Subcontexts are referenced through dot delimiters (.).
- Subcontexts must be created before objects are placed into them.
- Typically, when objects are bound to a JNDI tree, subcontexts are automatically created based on the JNDI name.

If the following context exists:

```
com.oracle.examples
```

Then you cannot bind:

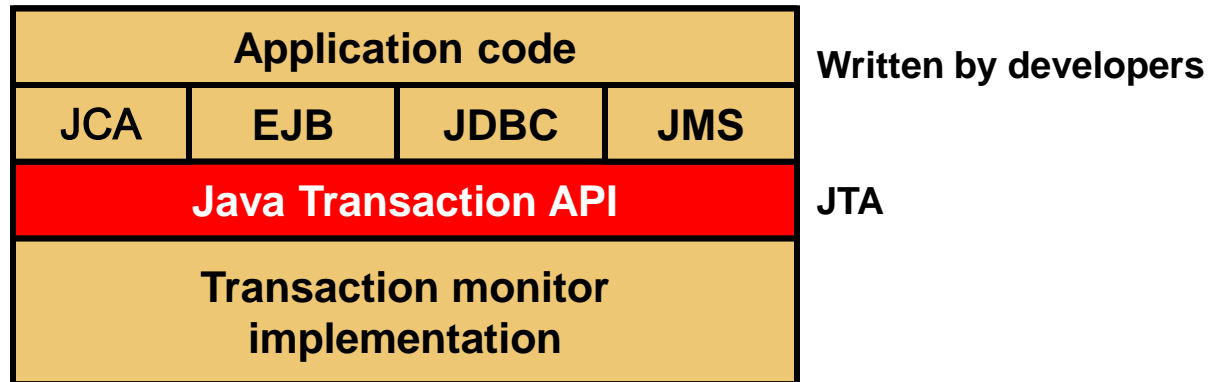
```
com.oracle.examples.ejb.SomeObject
```

Without first creating:

```
com.oracle.examples.ejb
```

# Java Transaction API (JTA)

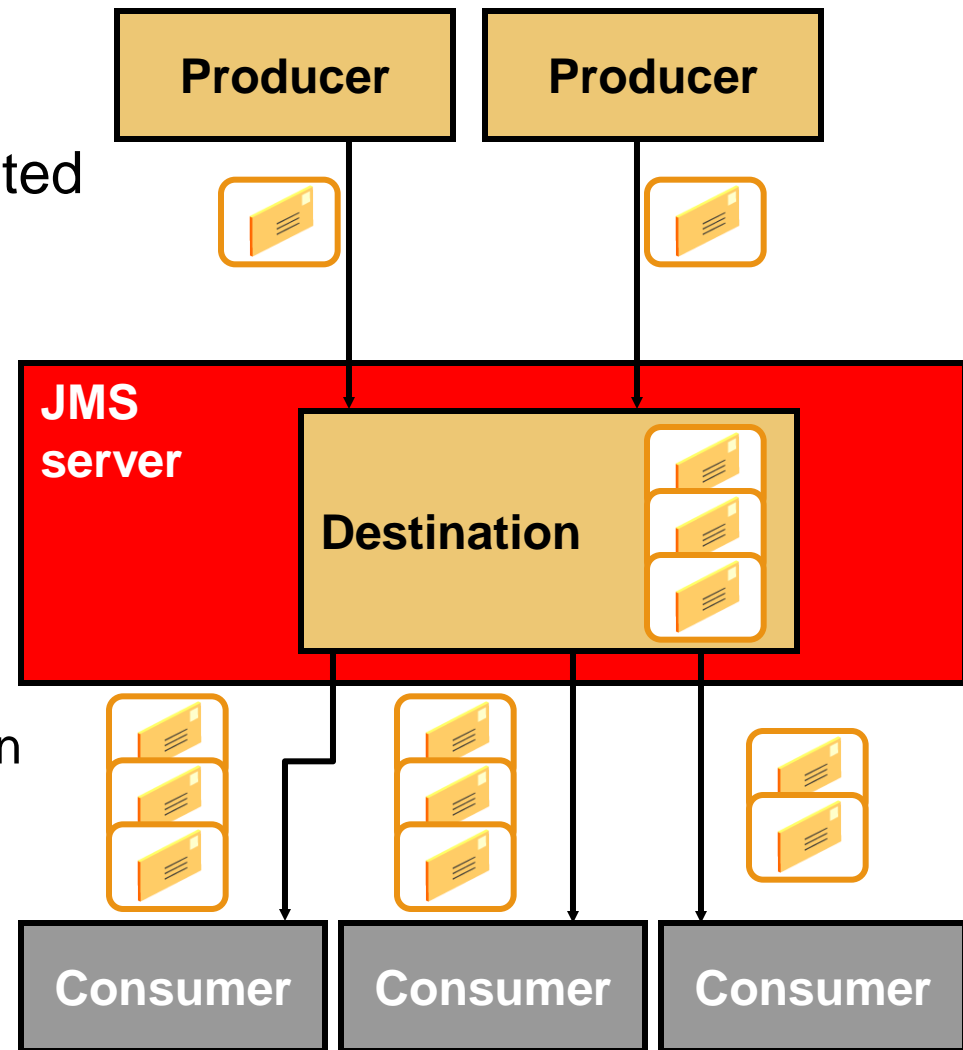
JTA is a standard Java API for demarcating transactions within a program.





# Java Message Service (JMS)

- JMS is a Java API for accessing message-oriented middleware.
- The interface supports:
  - Point-to-point domain
  - Publish/subscribe (“pub/sub”) domain
  - Guaranteed message delivery
  - Transactional participation
  - Dynamically configurable services
  - Application- or system-scoped resources
  - Interoperability with other messaging systems

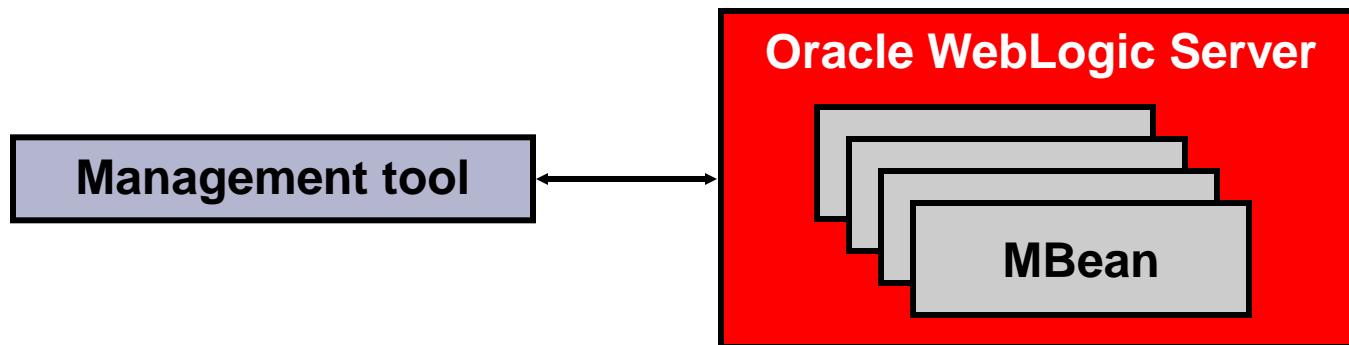


# Java Authentication and Authorization (JAAS)

- Java Authentication and Authorization Service (JAAS) is a Java-based security management framework.
- JAAS supports:
  - Single sign-on
  - A Pluggable Authentication Module (PAM)
- JAAS enables flexible control over authorization whether it is based on:
  - Users
  - Groups
  - Roles

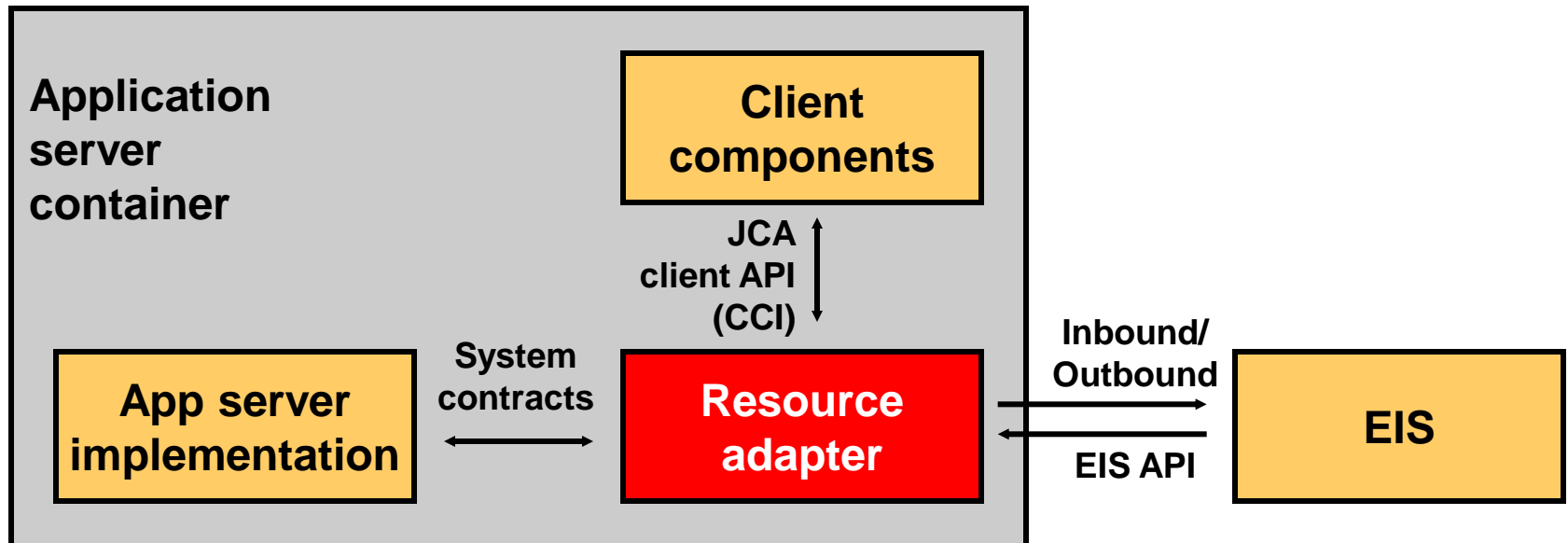
# Java Management Extensions (JMX)

- Java Management Extensions (JMX):
  - Defines a standard infrastructure to manage a device from Java programs
  - Decouples the managed device from the management tools
- The specification describes MBeans, which are the building blocks of JMX.



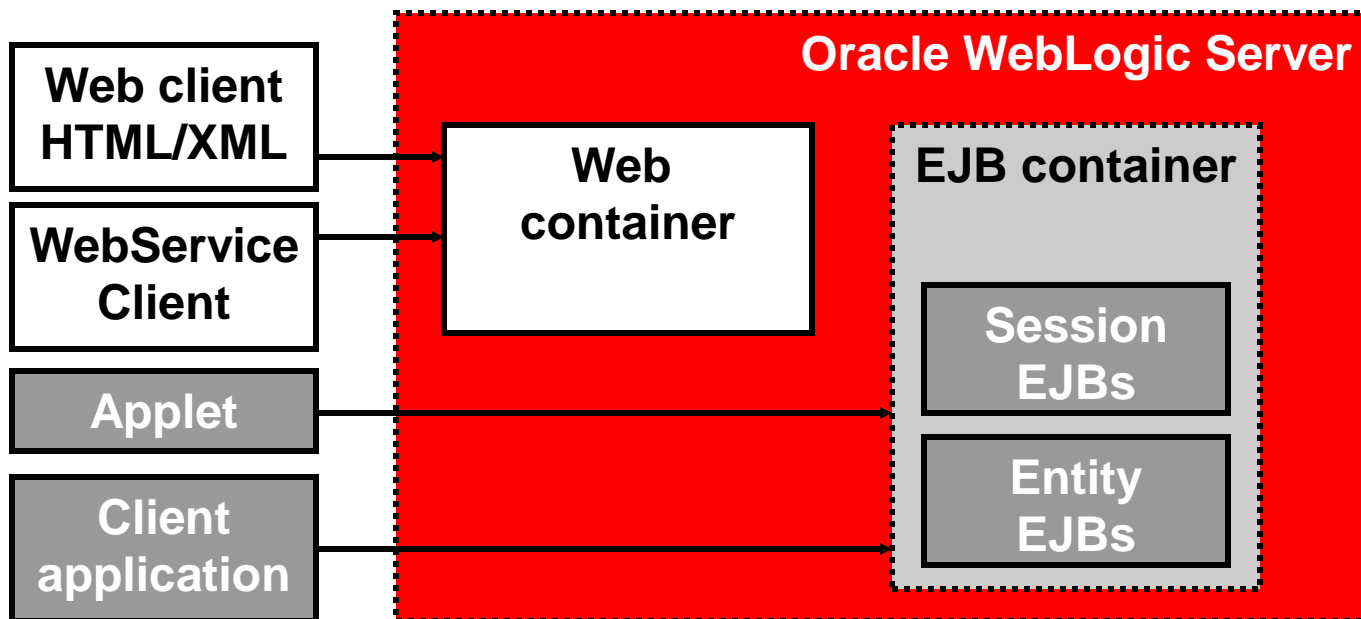
# Java EE Connector Architecture (JCA)

- Connects Enterprise Information Systems (EIS) with resource adapters
- Resource adapters can be deployed in a Resource Adapter Archive (RAR)



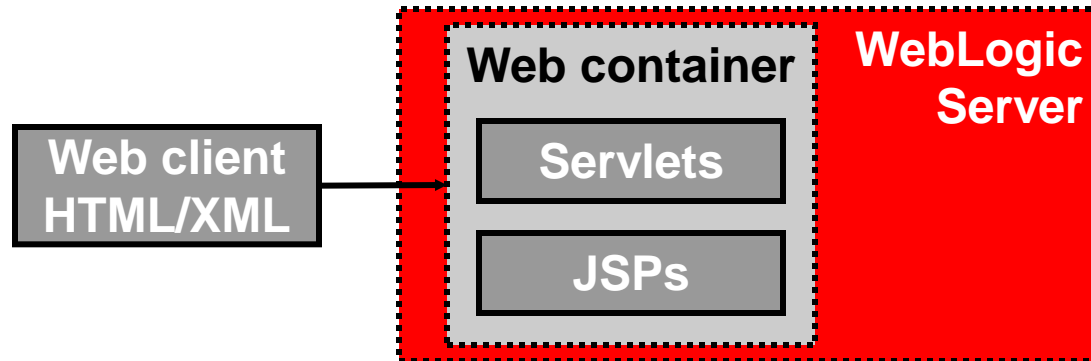
# Client Application

- The client application interacts with WLS through JRMP/T3, IIOP, and jCOM.
- The types of clients include:
  - Stand-alone Java applications
  - Applets within a browser



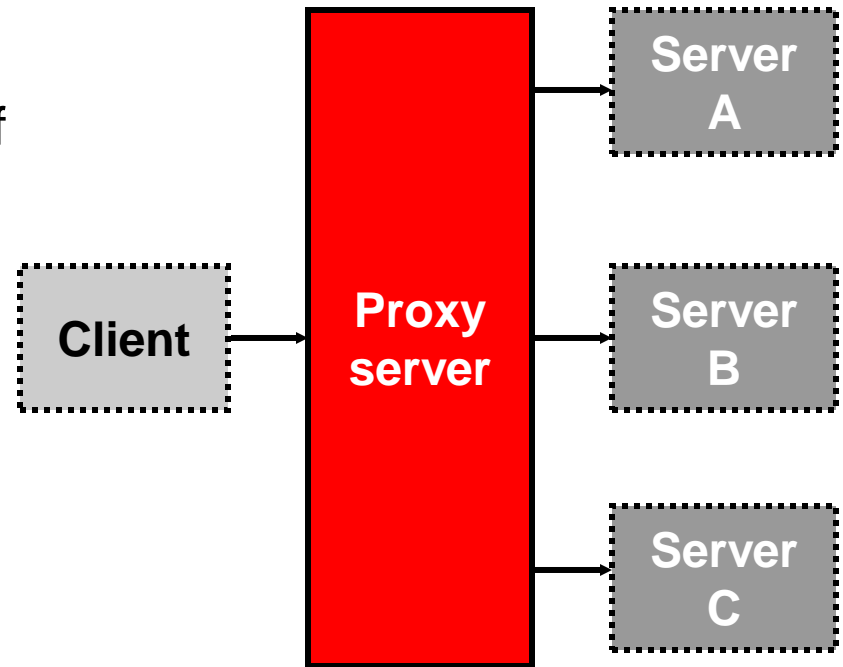
# Web Client

- A Web client interacts with Oracle WebLogic Server via HTTP using servlets or JSPs.
- The types of Web clients include:
  - Browser
  - WebService (SOAP over HTTP)



# Proxy Server

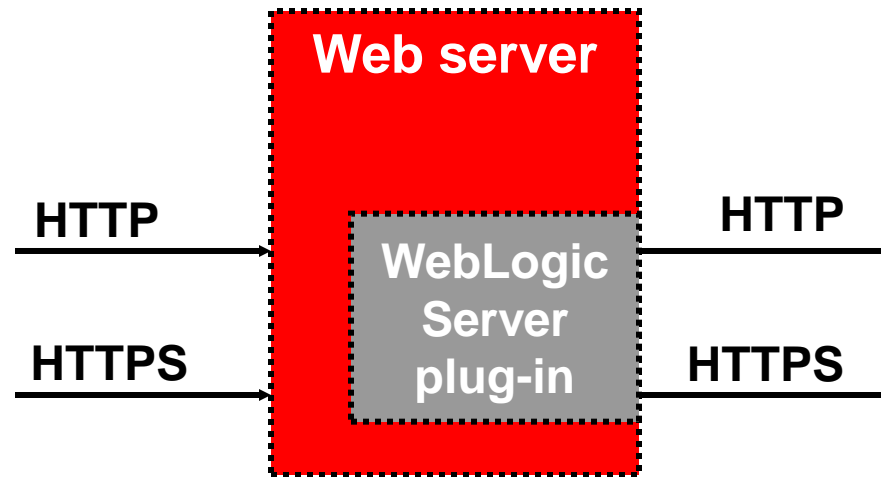
- The proxy server:
  - Forwards requests to other machines
  - Can be used as a level of indirection and security
  - Can be used to load-balance a system
- A reverse proxy is a Web page cache.



# Web Server

Web servers:

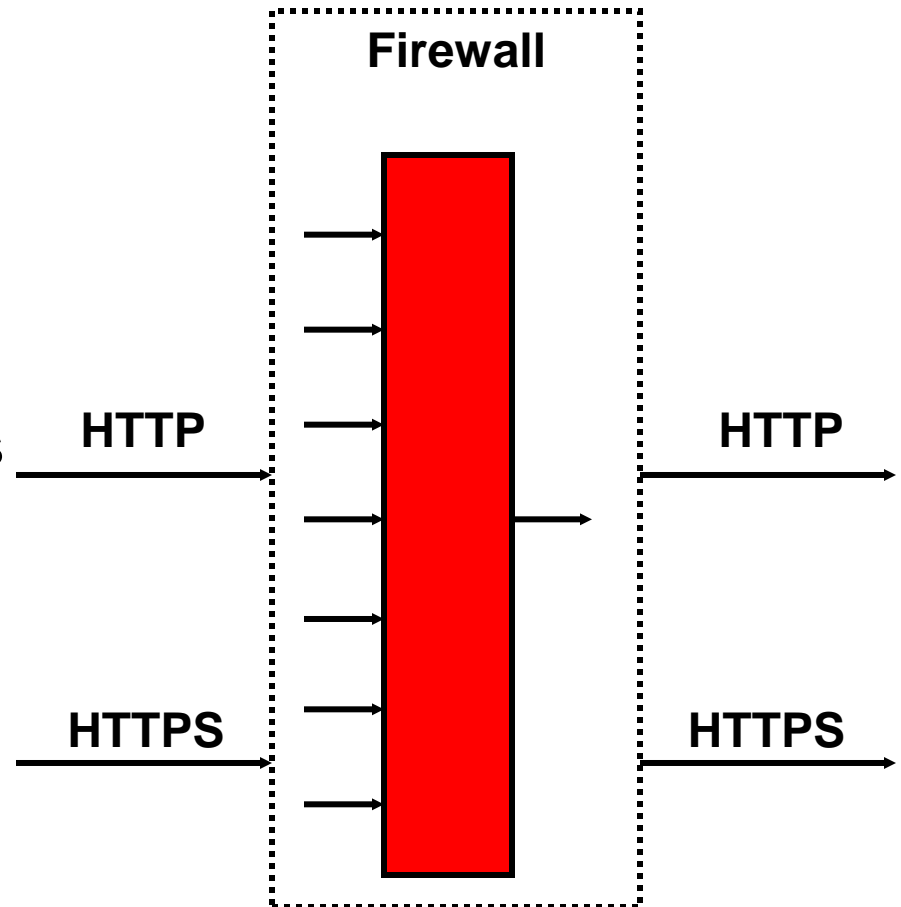
- Provide Web content
- Communicate via HTTP, FTP, and so forth
- Can handle CGI requests
- Proxy some requests to application servers





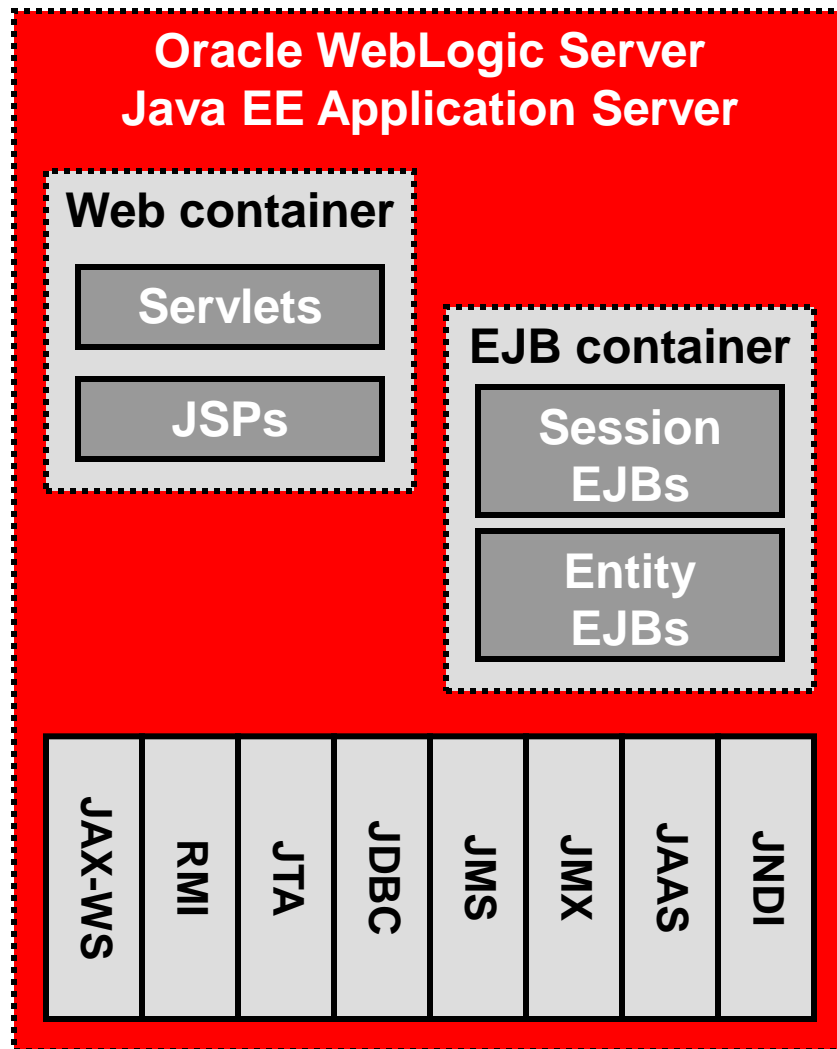
# Firewalls

- Provide filtering, authorization, and authentication services
- Help keep out hackers
- Map port requests
- Can act as proxy servers
- Can decrease back-end network activity

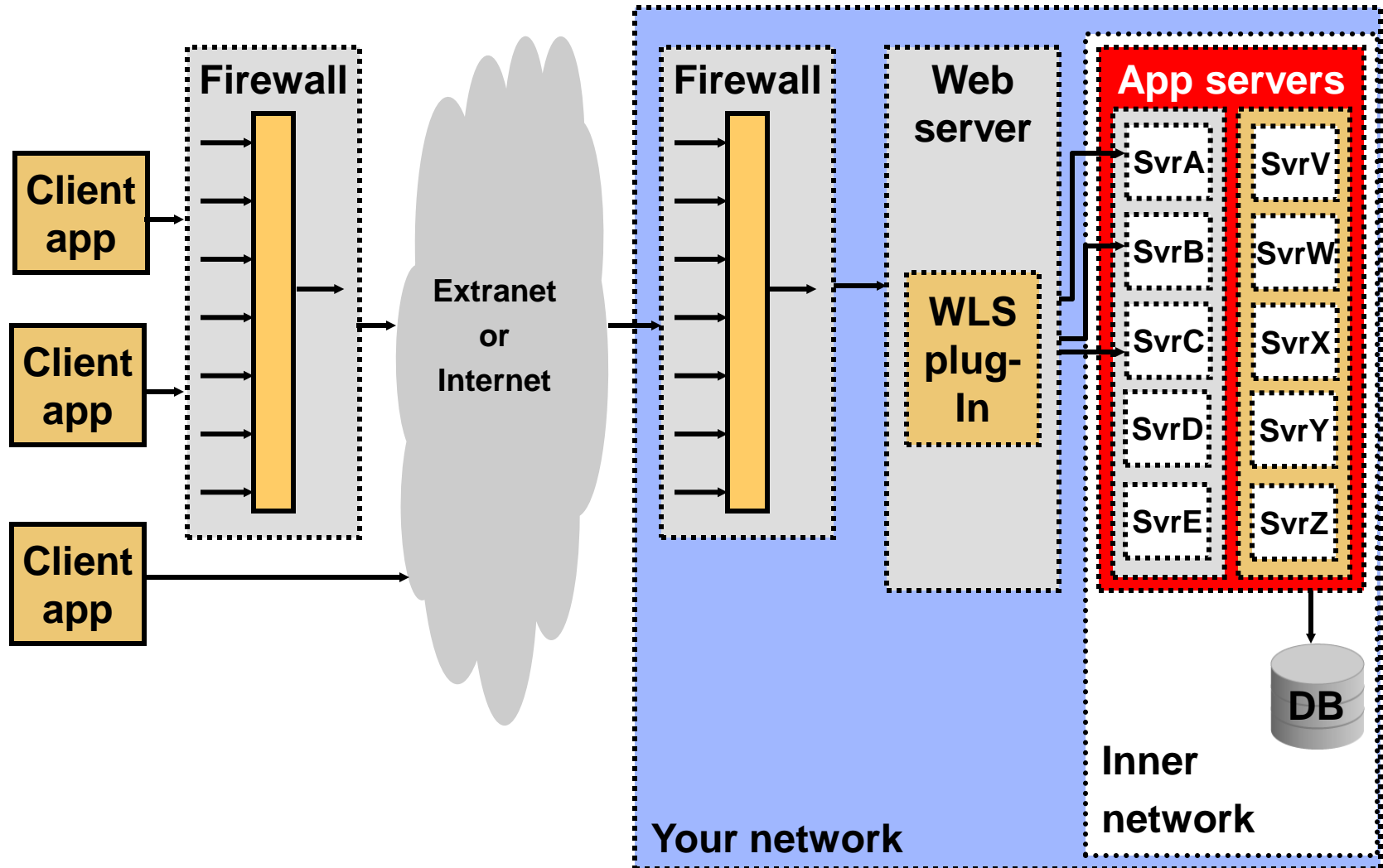


# Application Servers

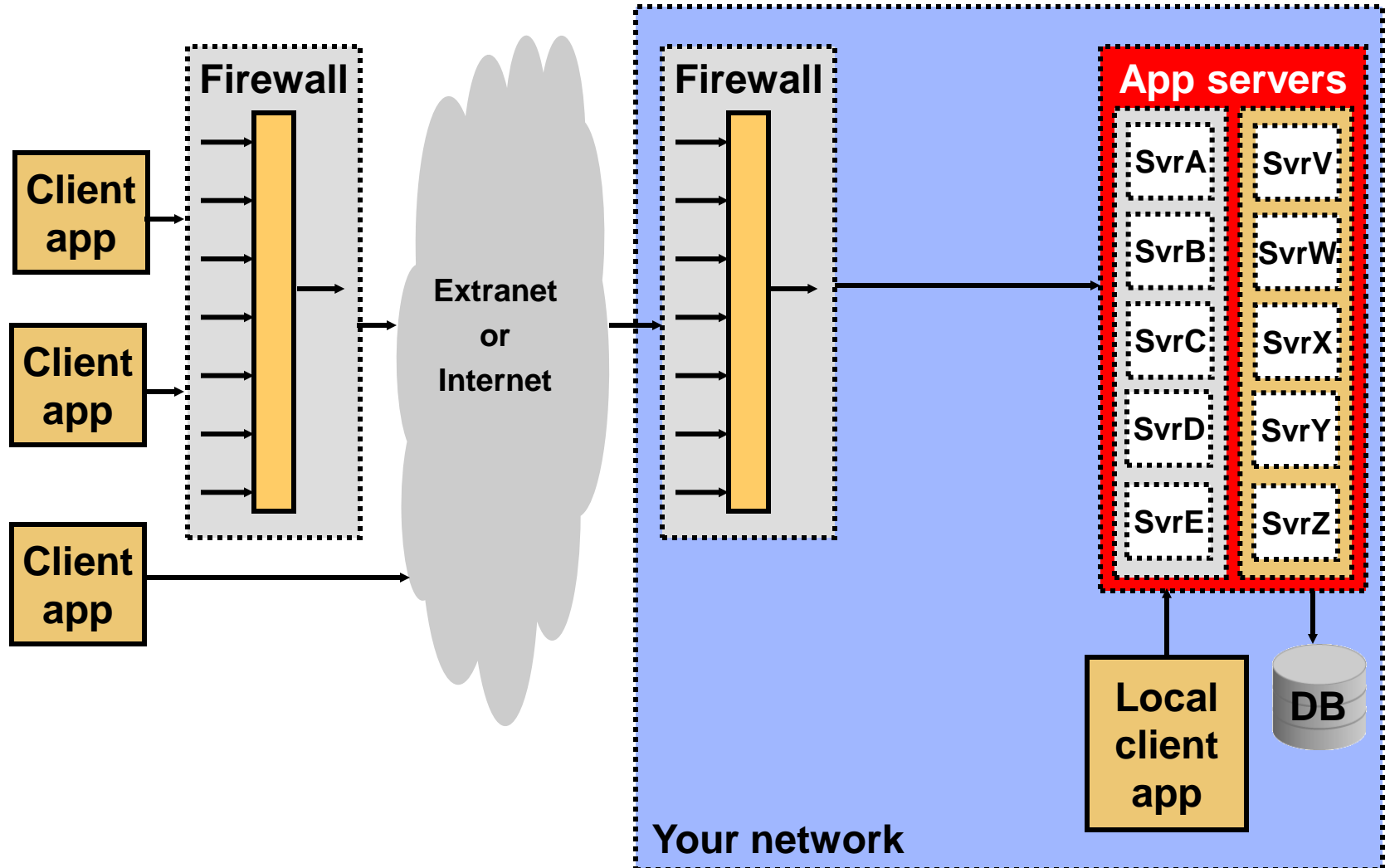
- Provide services that support the execution and availability of deployed applications
- Handle heavier processing chores than Web servers



# Web Application Server Configuration



# Application Server Configuration



# Quiz

Oracle WebLogic Server 10.3.1 is certified with JDK 1.6.

1. True
2. False

# Summary

In this lesson, you should have learned how to:

- Explain the motivation behind distributed systems
- List the major components of the Java EE specification

# **Practice 2 Overview: Defining Terminology and Architecture**

There is no practice for this lesson.