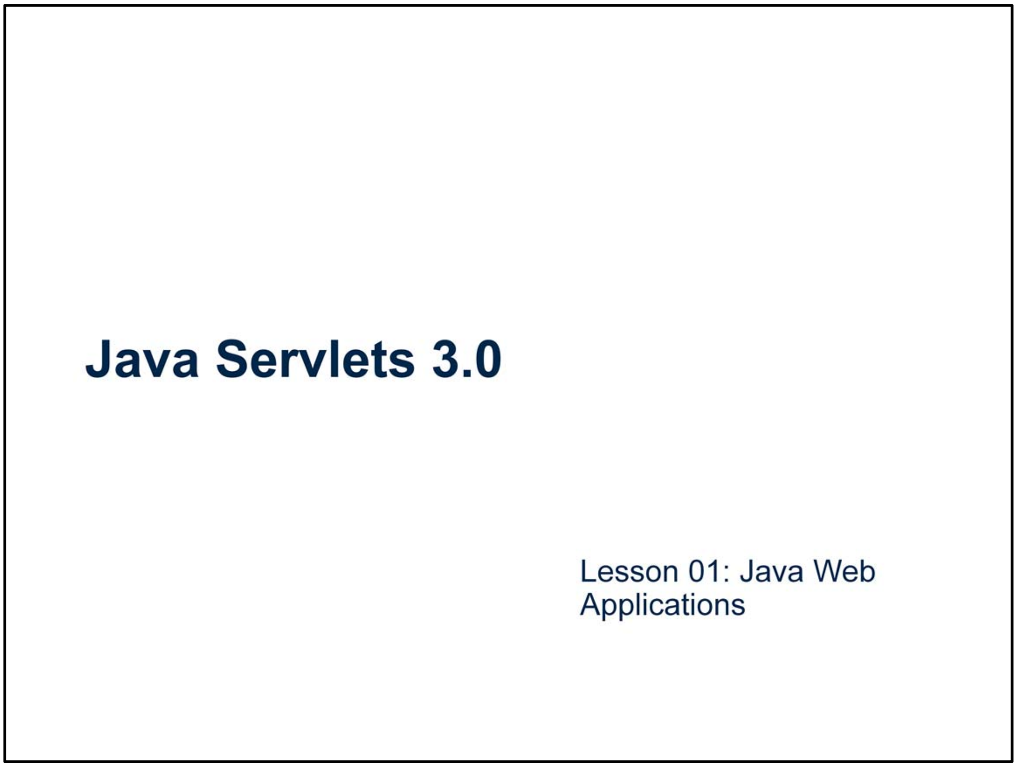
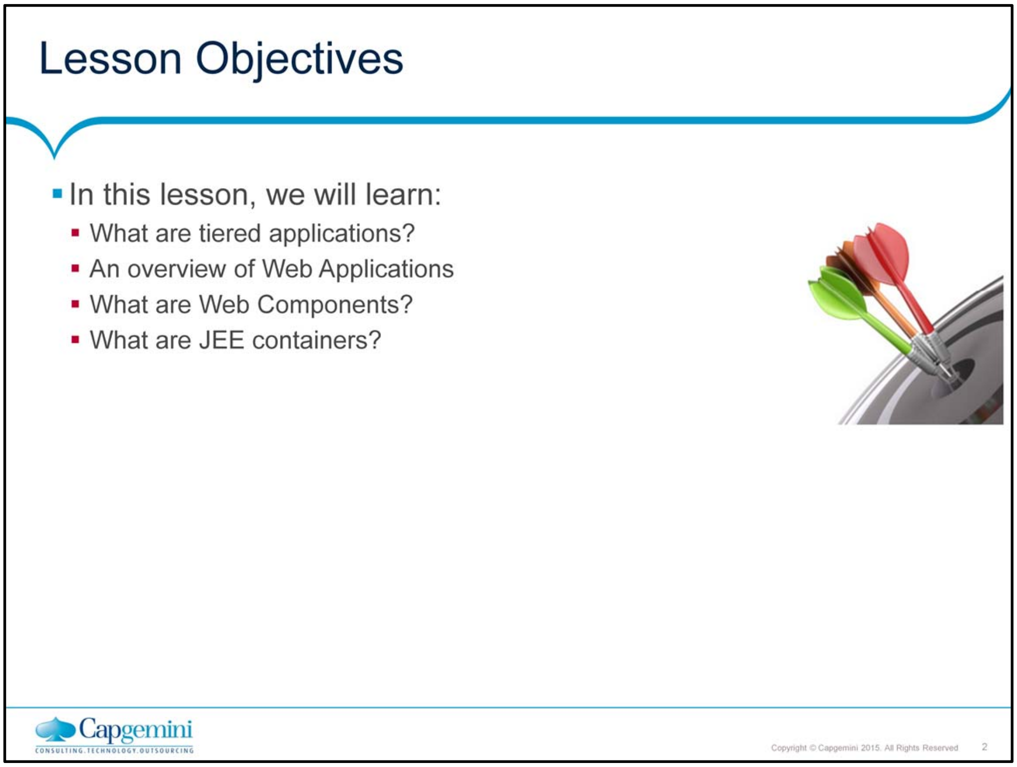


Java Servlets 3.0

Lesson 01: Java Web
Applications



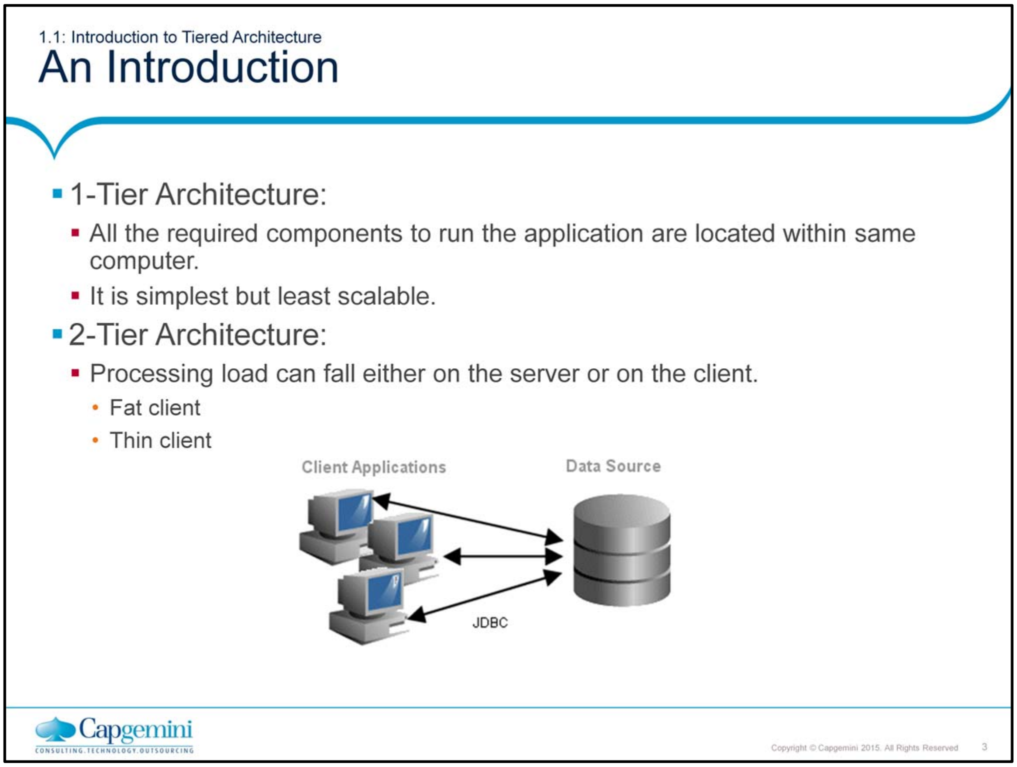
Lesson Objectives

- In this lesson, we will learn:
  - What are tiered applications?
  - An overview of Web Applications
  - What are Web Components?
  - What are JEE containers?

2

Lesson Objectives:
This lesson introduces Web application concepts. The lesson contents are:
Lesson 01: Java Web Applications

1.1: Introduction to tiered architecture
1.2: Web applications – an overview
1.3: What are Web components?
1.4: What are JEE containers?



1.1: Introduction to Tiered Architecture

An Introduction

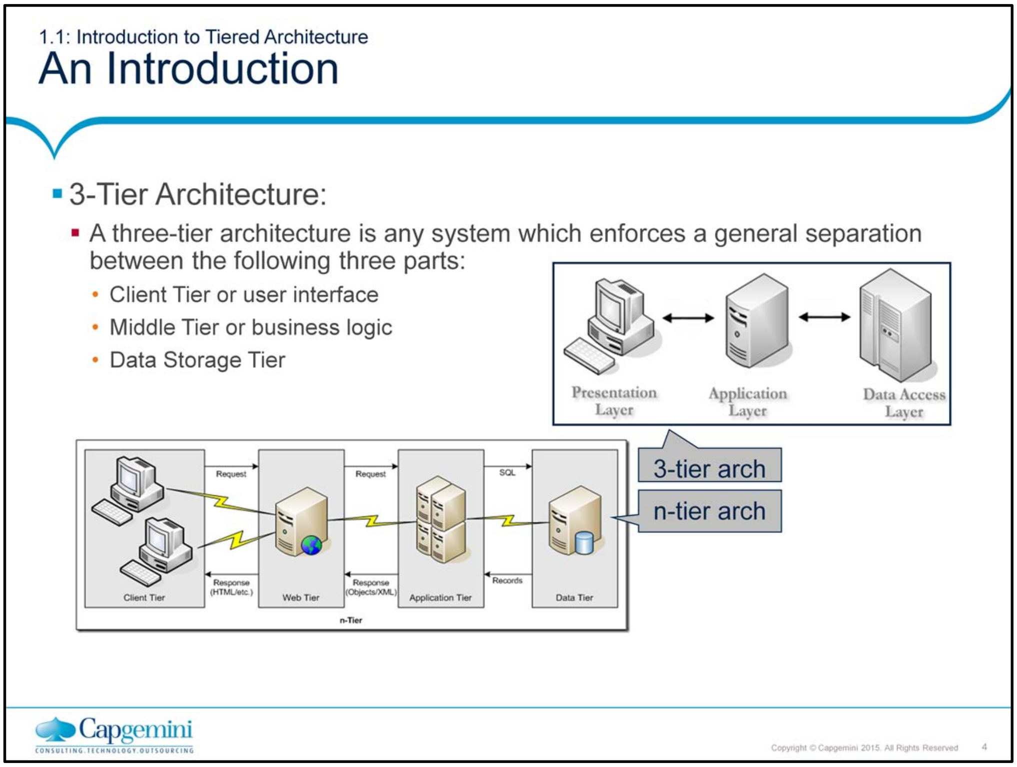
- 1-Tier Architecture:
  - All the required components to run the application are located within same computer.
  - It is simplest but least scalable.
- 2-Tier Architecture:
  - Processing load can fall either on the server or on the client.
    - Fat client
    - Thin client

Client Applications
Data Source
JDBC

Introduction to Tiered Architecture:
Tiered architecture can be classified as 1-Tier, 2-Tier, and 3-Tier or n-tier.
1-Tier Architecture is the simplest, single tier on single user, and is the equivalent of running an application on a personal computer. All the required components, that is User interface, business logic, and data storage to run the application are located within the computer. They are the easiest to design, but the least scalable. This is because they are not part of a network – they are not useful for designing web applications.
2-Tier application or architecture is a client server application where the processing load can fall either on the server or on the client.

> When the processing load falls on the client, the server simply acts as a controller between the data and the client. Such a client is called as a fat client and imposes a lot of memory and bandwidth on the client's machine. Thus in this type of architecture the business logic and the presentation layer is located on the client machine and the data layer is on the server machine.
> Problems in this approach? As the number of clients connecting to the server increases, there will be multiple requests to the server which will be time consuming. Also client machine should have sufficient processing power.
> Another approach is that the client is a thin client and the business logic and the data layer are located on the server machine.
> Problems in this approach? Multiple clients connecting to the sever may overload the server and this will make processing each client request very slow.
> To overcome all these problems one should plan a 3-tier architecture.

The figure in the above slide shows a typical client/server application where Java programs access database server for executing SQL calls.



1.1: Introduction to Tiered Architecture

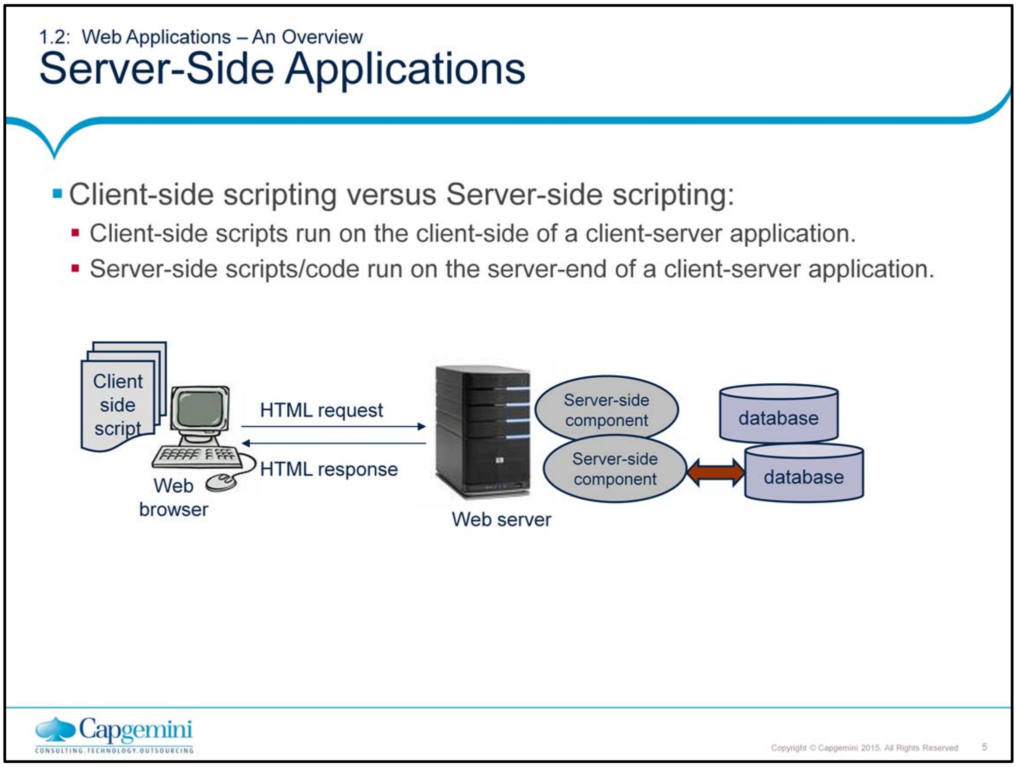
An Introduction

- 3-Tier Architecture:
  - A three-tier architecture is any system which enforces a general separation between the following three parts:
    - Client Tier or user interface
    - Middle Tier or business logic
    - Data Storage Tier

Introduction to Tiered Architecture:
3-Tier Architecture: 3-Tier Architecture is most commonly used to build web applications. In this model, the browser acts like a client. Middleware or an application server contains the business logic, and database servers handle data functions. This approach separates business logic from display and data. The figure in the above slide shows a 3-tier application where Presentation, Business logic, and Data management is distributed across three separate tiers.

Middle Layer acts as a bridge between the clients and the server, handling all the requests that come in from the client, routing appropriate requests to database, and taking care of the application business logic. The middle layer typically is an application server such as JBOSS, Websphere Application Server, Weblogic, WildFly
N-Tier Architecture: The 3-tier architecture does not specialize functional layers. It is fine for prototypical or very simple web applications, but it does not measure up to the complexity demanded of web applications. The application server is still too broad, with too many functions grouped together. This reduces flexibility and scalability. N-Tier Architectures provide finer granularity, which provides more modules to choose from as the application is separated into smaller functions.

Usually N-Tier Architecture begins as a 3-Tier model and is expanded. Here the application logic components can be divided and made available on multiple middle layers. Various layers of multi-tier architecture may or may not be distributed across various machines.



1.2: Web Applications – An Overview

Server-Side Applications

- Client-side scripting versus Server-side scripting:
  - Client-side scripts run on the client-side of a client-server application.
  - Server-side scripts/code run on the server-end of a client-server application.

Web Applications – An Overview:

Desktop applications are those that have stand-alone applications running on them. They are maintenance heavy.

Server-side applications on the other hand run on the server side of a client-server system. The server is then able to access server side resources like databases, server components, and provide these services to multiple clients concurrently.

Client-side scripting versus Server-side scripting:

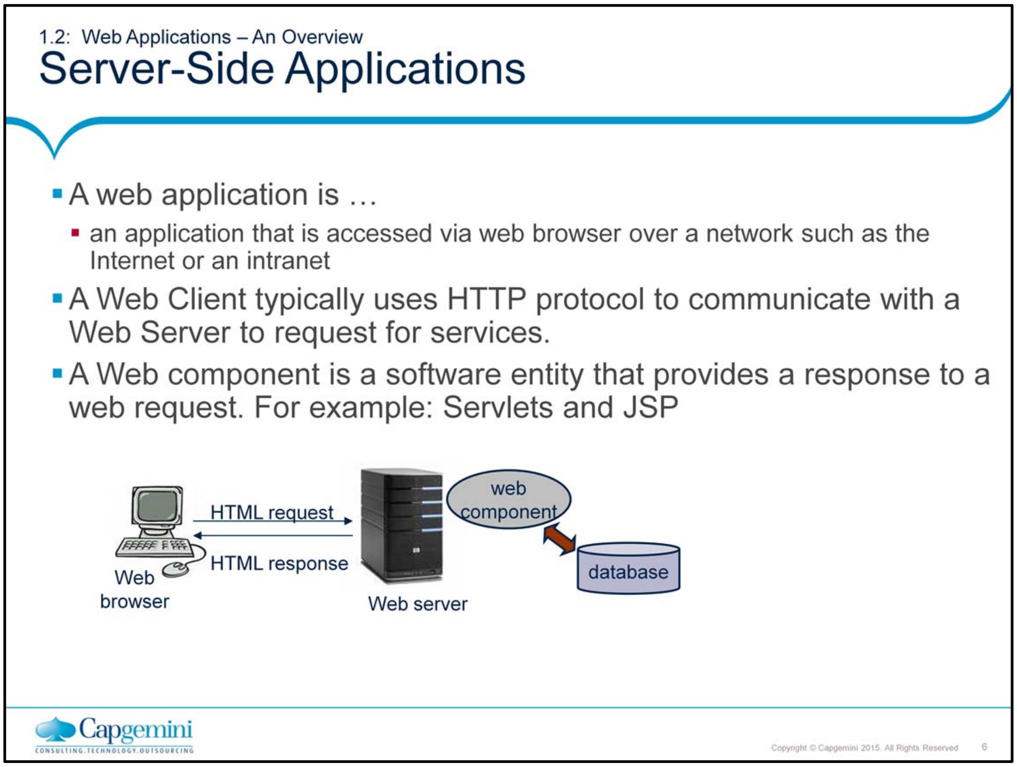
Client-side scripts run on the client-side of a client-server application. For example, in a web application, we have written Javascript that runs on the browser to perform presentation validation!

Server-side scripts run on the server-end of a client-server application. For example, when a HTML form is submitted, the data may be persisted on a database at the server side!

From a web application perspective, there are many advantages to this:

The response is in HTML form, so complex code that is executed to generate the response stays at the server-side.

Server is able to access server side resources like databases, server components, which ensures centralized control.a



1.2: Web Applications – An Overview

Server-Side Applications

- A web application is …
  - an application that is accessed via web browser over a network such as the Internet or an intranet
- A Web Client typically uses HTTP protocol to communicate with a Web Server to request for services.
- A Web component is a software entity that provides a response to a web request. For example: Servlets and JSP

Web Applications – An Overview:

A web application relieves the developer of the responsibility of building a client for a specific type of computer or a specific operating system. Since the client runs in a web browser, the user could be using an IBM-compatible or a Mac. They can be running Windows XP or Windows Vista using popular browsers like Internet Explorer, Netscape Navigator, or Firefox.

Web applications commonly use a combination of server-side script (such as ASP, JSP, PHP) and client-side script (such as HTML, JavaScript) to develop the application.

The client-side script deals with the presentation of the information.

The server-side script/code deals with business logic, storing and retrieving the information.

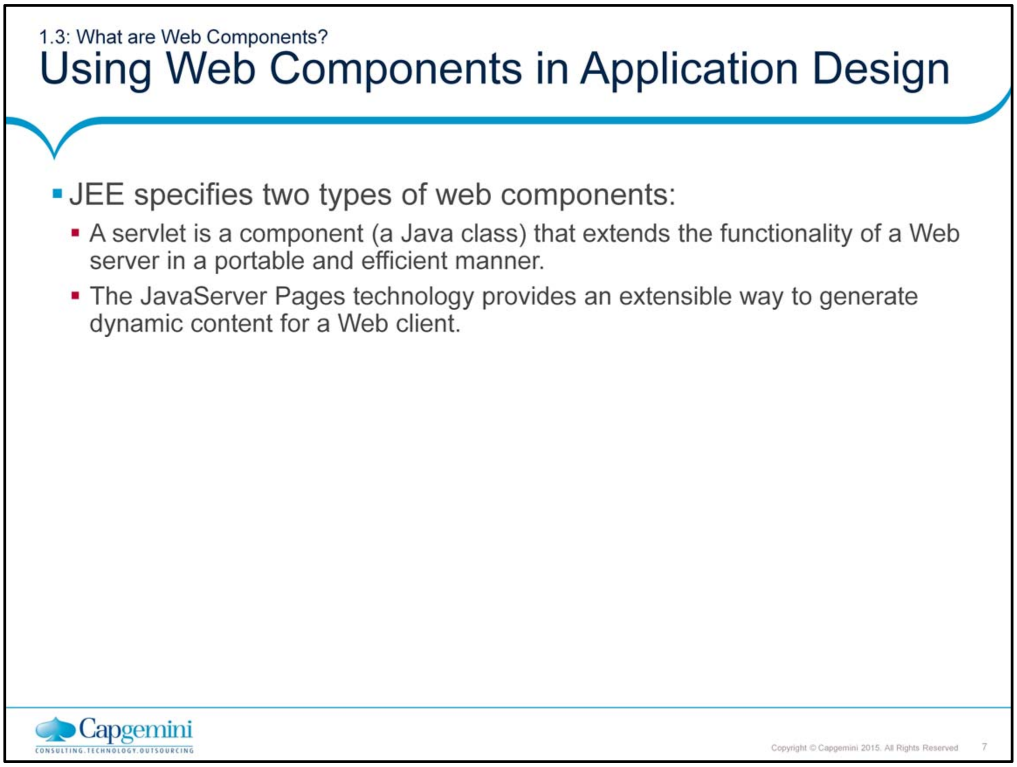
Web clients (browser) typically requests for services to a web server, and the web server either sends a static HTML page or dynamically generates a response to the request.

Web server is a computer program that delivers (serves) content, such as a web page, using the Hypertext Transfer Protocol. The term web server can also refer to the computer or virtual machine running the program.

Web components present in the Web Server are responsible for generating appropriate dynamic response. The JEE platform specifies two types of Web components:

Servlets

JavaServer Pages (JSP) pages



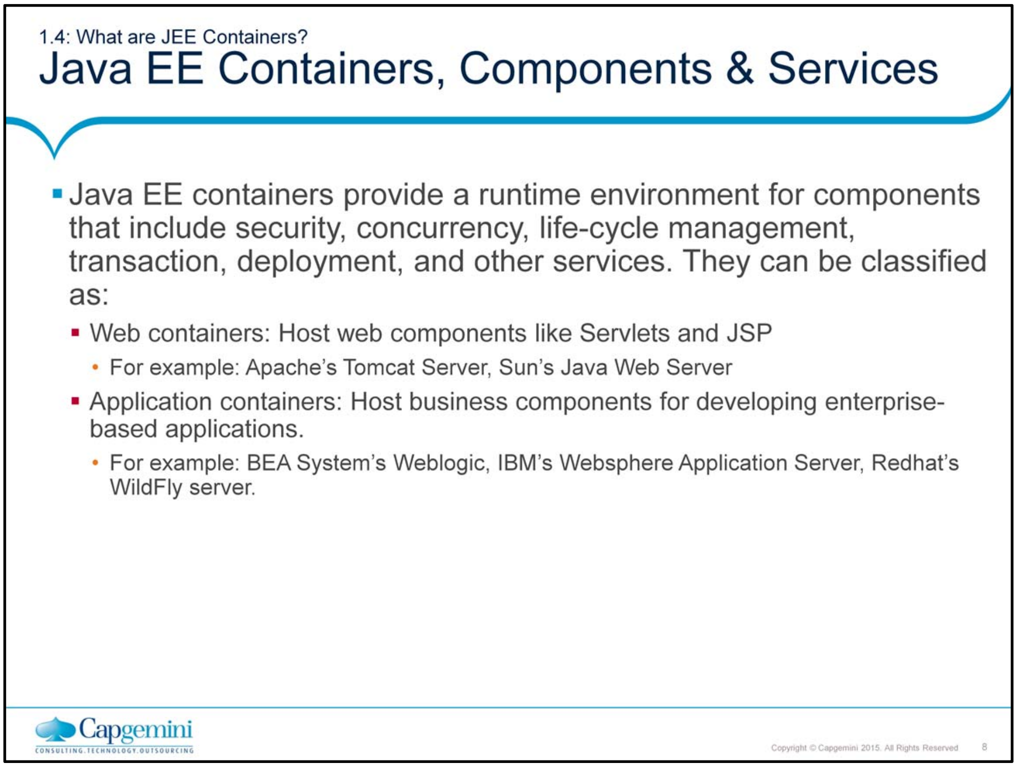
1.3: What are Web Components?

Using Web Components in Application Design

- JEE specifies two types of web components:
  - A servlet is a component (a Java class) that extends the functionality of a Web server in a portable and efficient manner.
  - The JavaServer Pages technology provides an extensible way to generate dynamic content for a Web client.

What are Web Components?
A Web component typically generates the user interface for a Web-based application.
Servlets: A Web server hosts Java servlet classes that execute within a server (web) container. When a servlet receives a request from a client, it generates a response, possibly by invoking business logic in Enterprise Beans or by querying a database directly. It then sends the response – as an HTML or XML document – to the requestor.
Java Server Pages technology (JSP): It uses XML-like tags and scriptlets written in the Java programming language to encapsulate the logic that generates the content for the page.

> By separating the page logic from its design and display, and supporting a reusable component-based design, JSP technology makes it faster and easier than ever to build web-based applications.
> We shall be covering servlets in greater detail in this course and shall be covering JSP as a subsequent course.



1.4: What are JEE Containers?

Java EE Containers, Components & Services

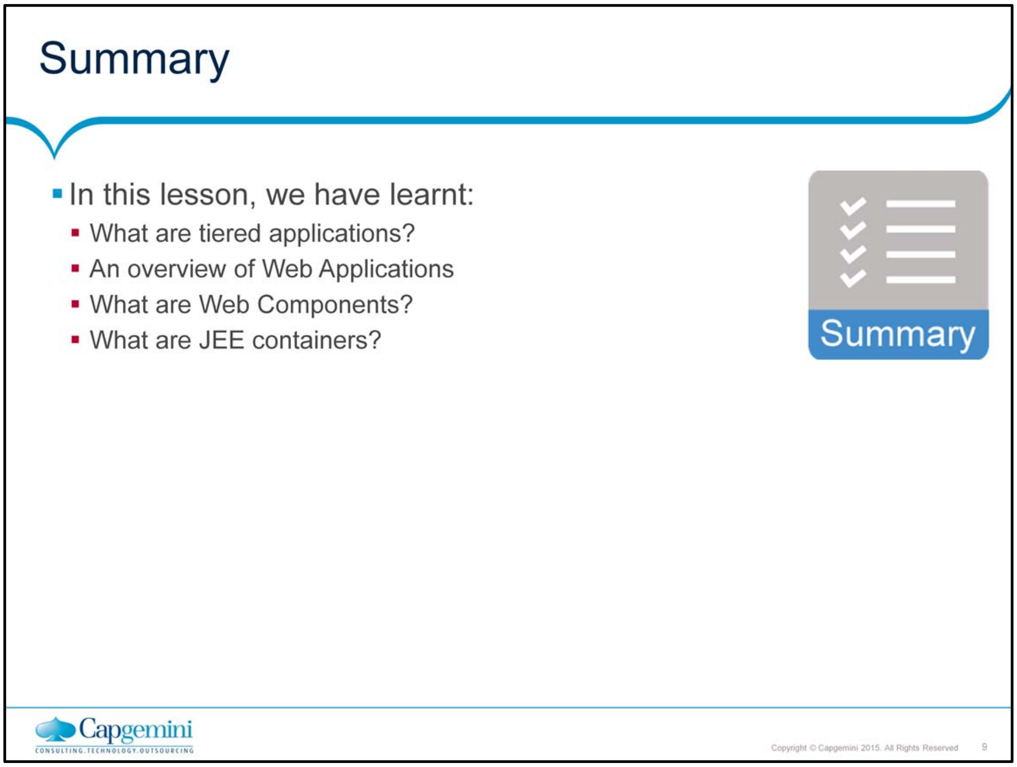
- Java EE containers provide a runtime environment for components that include security, concurrency, life-cycle management, transaction, deployment, and other services. They can be classified as:
  - Web containers: Host web components like Servlets and JSP
    - For example: Apache's Tomcat Server, Sun's Java Web Server
  - Application containers: Host business components for developing enterprise-based applications.
    - For example: BEA System's Weblogic, IBM's Websphere Application Server, Redhat's WildFly server.

8

What are JEE Containers?
Once an application is built, it is packaged into one or more standard units for deployment to any Java EE platform. This JEE unit is now ready to be deployed. Deployment typically involves using a platform's deployment tool to specify location-specific information, such as a list of local users, who can access it, and the name of the local database. Once deployed on a local platform, the application is ready to run.
Containers provide a runtime environment for components that includes security, concurrency, life-cycle management, transaction, deployment, and other services. Since the container handles these services, the developers need not be concerned about taking care of these aspects in the code. They can then focus on the business logic!
JEE Containers are classified as:

Web component containers (servers): Allow us to test and deploy web components. They come in three flavors: Standalone, Add-on, Embeddable. We shall be seeing more on this topic in lesson-2.

Application servers: They offer server-side support for developing enterprise-based applications. Most Java-based application servers support servlets,JSP, and the rest of JEE specification. For example: BEA System's Weblogic, IBM's Websphere Application Server, Redhat's WildFly server

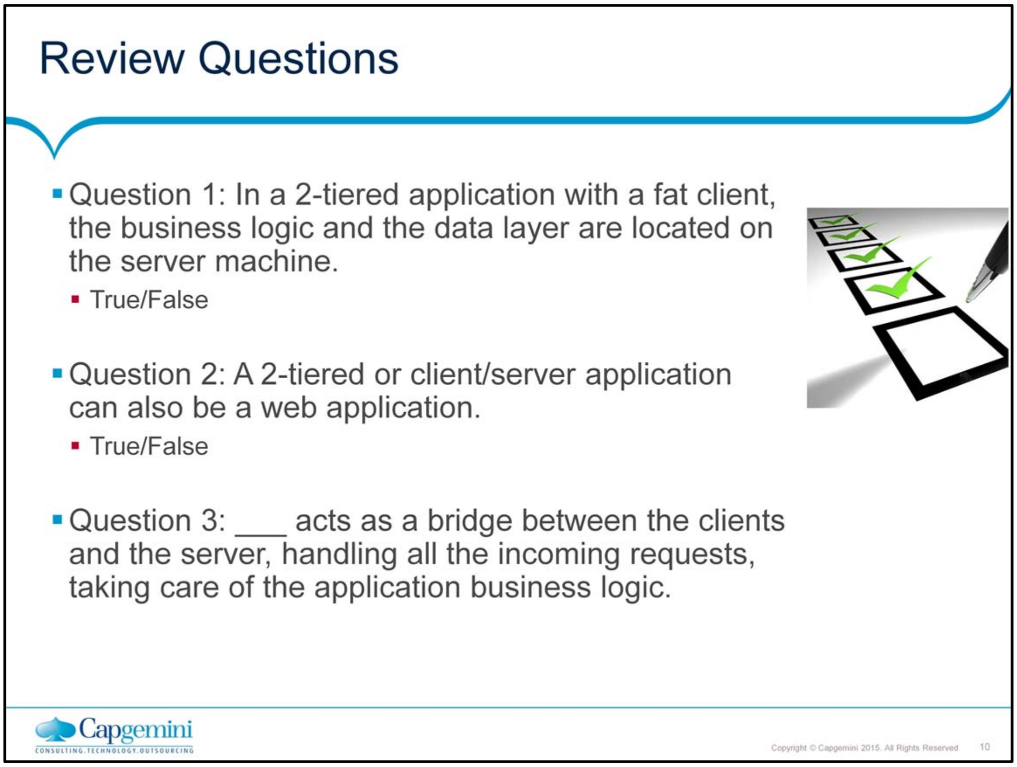


Summary

- In this lesson, we have learnt:
  - What are tiered applications?
  - An overview of Web Applications
  - What are Web Components?
  - What are JEE containers?

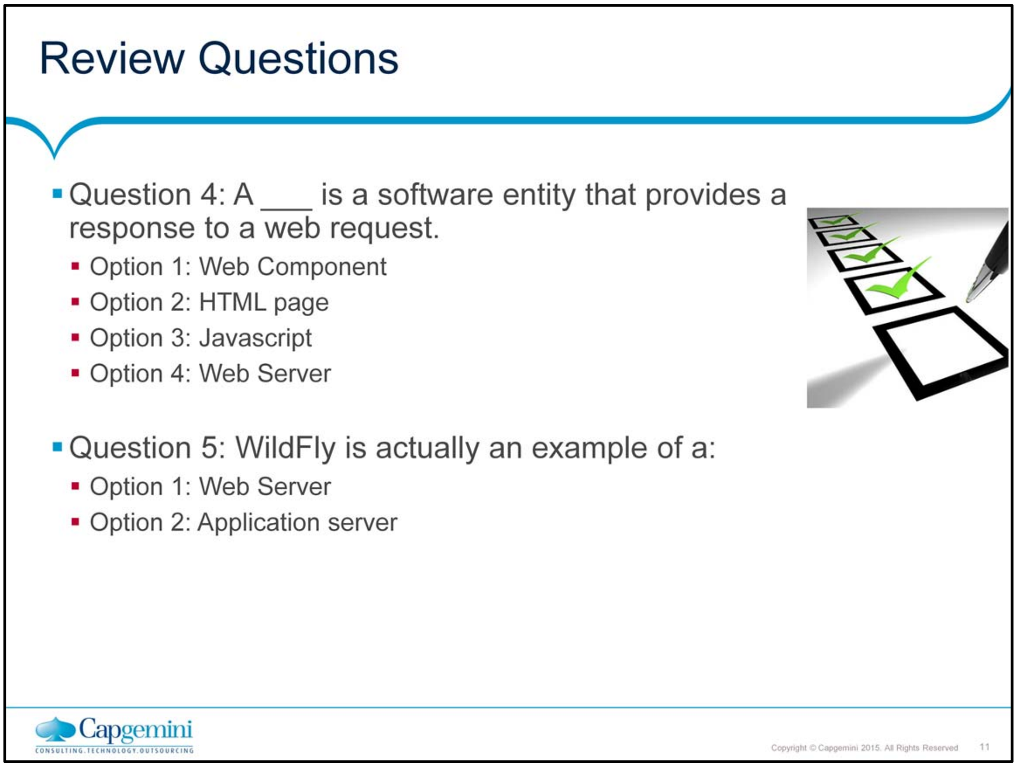
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

9



Review Questions

- Question 1: In a 2-tiered application with a fat client, the business logic and the data layer are located on the server machine.
  - True/False
- Question 2: A 2-tiered or client/server application can also be a web application.
  - True/False
- Question 3: ___ acts as a bridge between the clients and the server, handling all the incoming requests, taking care of the application business logic.



Review Questions

- Question 4: A ___ is a software entity that provides a response to a web request.
  - Option 1: Web Component
  - Option 2: HTML page
  - Option 3: Javascript
  - Option 4: Web Server
- Question 5: WildFly is actually an example of a:
  - Option 1: Web Server
  - Option 2: Application server

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

11