Considering all these issues, one thing should be clear: although an e-commerce application is a client-server application, it has major differences with a typical client-server application. In case of an e-commerce application, the server has to essentially carry out most of the application logic and leave just the user interface portion and basic validations (such as whether a field in a form is numeric) to the client. There is an additional important issue here. As anyone, literally anyone in the world could connect to the Web site and become a client, it would be quite impractical to expect each one of them to develop programs for the GUI as well as local validations. Therefore, even these programs (in JSP, ASP, etc.) had to be developed and stored on the server and dispatched to the client PC as and when needed. This was a very important precondition. There are two approaches to distribute the load between the client and server:

1. Use the less thick client model for only user interface, without having any application logic at the client side. This means that: (a) The client would be responsible for GUI-based interactions with the user and performing simple validations, and (b) The server would be very thick and would contain all the application logic, handle database processing and perform transaction management.
2. The other approach paved way for a **3-tier architecture**. Here, the application logic is taken from both the client and the server and clubbed into a new tier. The new tier is sandwiched between the existing two tiers. This is shown in Figure 8.4. The new tier is called **application server**, since it hosts all the application logic; business rules and performs transaction management. The first tier, as before, is a thick client with GUI features and performs basic validations. The last tier is now a **database server**, which is solely responsible for database processing and is independent of the application logic.

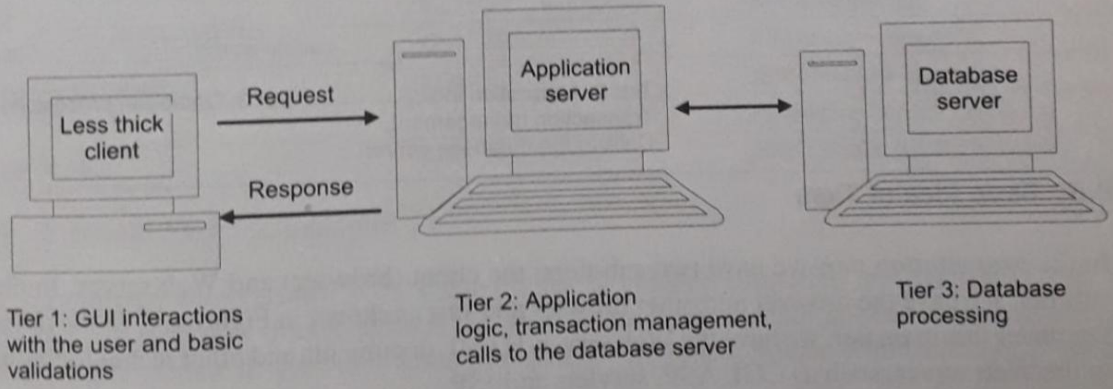
Less thick client
Request
Response
Application server
Database server
Tier 1: GUI interactions with the user and basic validations
Tier 2: Application logic, transaction management, calls to the database server
Tier 3: Database processing

Fig. 8.4 A 3-tier Application

Most modern e-commerce applications are based on this 3-tier architecture. In fact, the trend now is to try and extend this concept to an ***n*-tier architecture**, where *n* is 3 or more. In an n-tier architecture, there can be more than one application servers, running different applications.

8.5 A COMPARISON OF MICROSOFT AND JAVA TECHNOLOGIES

8.5.1 Introduction

As we have discussed, Web technology solutions are these days classified into three **tiers** or **layers**, **presentation tier, business tier** and **database tier**. *Presentation tier* shows screens to the users (which was possible with HTML also), takes user inputs (for placing an order, etc. which is not possible with

HTML.) It then invokes an application in the *business tier*, which in turn invokes a *database tier* whenever there is a database interaction, etc. In the case of Web technologies, what the client sees on the screen is actually sent by the server (HTTP/HTML) or/and it may involve a little client-based processing. However, it may be necessary to run a program on the server, have the output of the program converted into HTML format and sent to the client for the browser to interpret and finally for the user to see. The basic idea is depited in Figure 8.5.

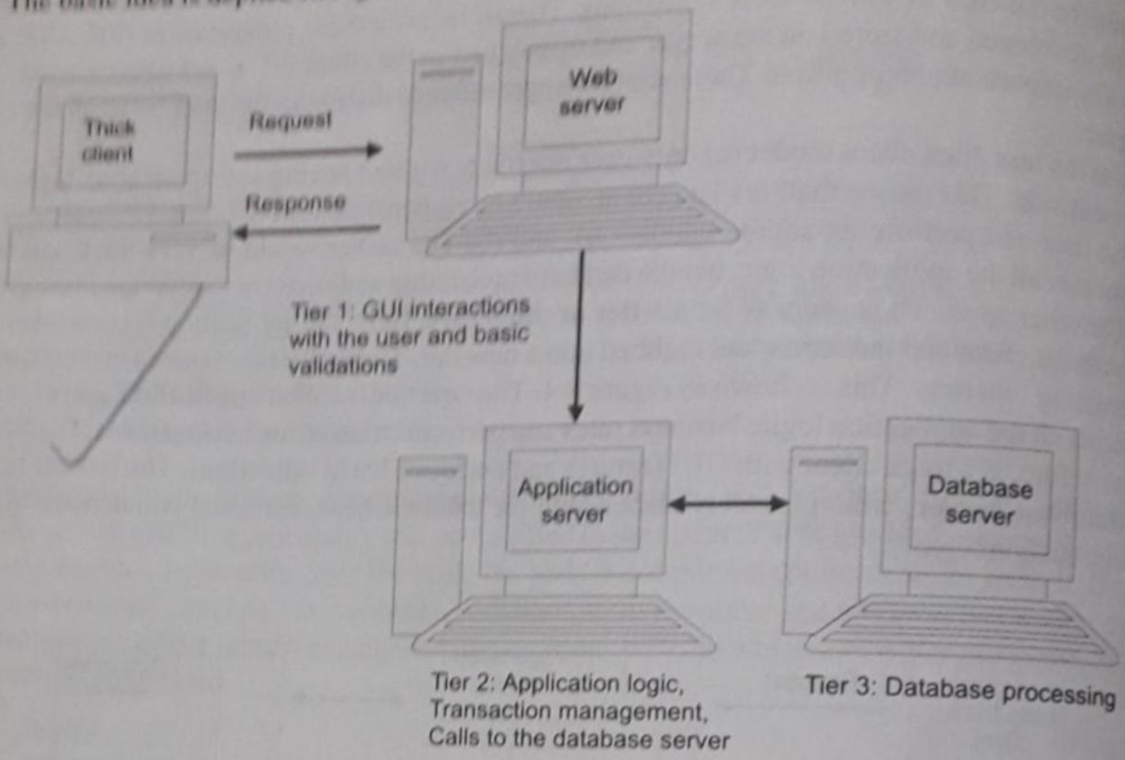
Thick client
Request
Response
Web server
Tier 1: GUI interactions with the user and basic validations
Application server
Database server
Tier 2: Application logic, Transaction management, Calls to the database server
Tier 3: Database processing

Fig. 8.5 Basic Idea of Tiers

- In the *presentation tier*, we have two sub-tiers: the client (browser) and Web server. In the first sub-tier, we have the browser and other browser add-ons as shown in Figure 8.6. In the other sub-tier under this main tier, we have the Web server, HTML documents and other technologies related to the Web server, such as CGI, ASP, servlets and JSP.
- In the *business tier*, the main technologies are MTS, MSMQ and EJB.
- In the *database tier*, the database servers such as Oracle, SQL server, etc. exist. They are not shown here since they are not specific to the Web platform. Instead, the existing database servers could be used in Web technology, and therefore, there is nothing special to them from the context of Web technology.

8.5.2 Presentation Tier Technologies

Figure 8.6 depicts a summary of the presentation tier technologies offered/supported by Microsoft and Sun. Note that we have used the word *supported*, which means that even if Sun has not developed some of these technologies, its Java-based systems support them. For instance, CORBA is a standard developed by a committee and supported by Sun in their J2EE platform.

At client or server?	Property	Microsoft Technology	Sun Technology
Client	HTTP (Web) based	HTML browser (Internet Explorer)	HTML browser (Netscape Navigator)
		ActiveX controls	Java applets
	Non-HTTP based	COM clients	CORBA clients
	Communication Protocol between client and server	DCOM	RMI, IIOP
Server	For creating dynamic Web pages	ISAPI, ASP	NSAPI, Servlets, JSP
	Other pages	HTML, XML	HTML, XML

Fig. 8.6 Presentation Tier Technologies

8.5.3 Business Tier Technologies

Figure 8.7 depicts a summary of the business tier technologies offered/supported by Microsoft and Sun.

Purpose	Microsoft Technology	Sun Technology
Transaction handling, Business objects	COM, MTS	EJB (Session beans)
Queuing and Messaging	MSMQ	IBM's MQSeries, Java Messaging Service (JMS)
Database access	ADO, OLE, ODBC	JDBC, J/SQL (via Entity beans)

Fig. 8.7 Business Tier Technologies

Having summarized the various Web technologies developed/supported by Microsoft and Sun Microsystems, we shall now draw diagrams that show where these technologies fit in the overall Web-based architecture. This will give us a good idea graphically about which Web technology fits in at which tier.

Figure 8.8 shows Microsoft's Web-based technologies at a broad level. Figure 8.9 shows a parallel view for the Sun technologies.

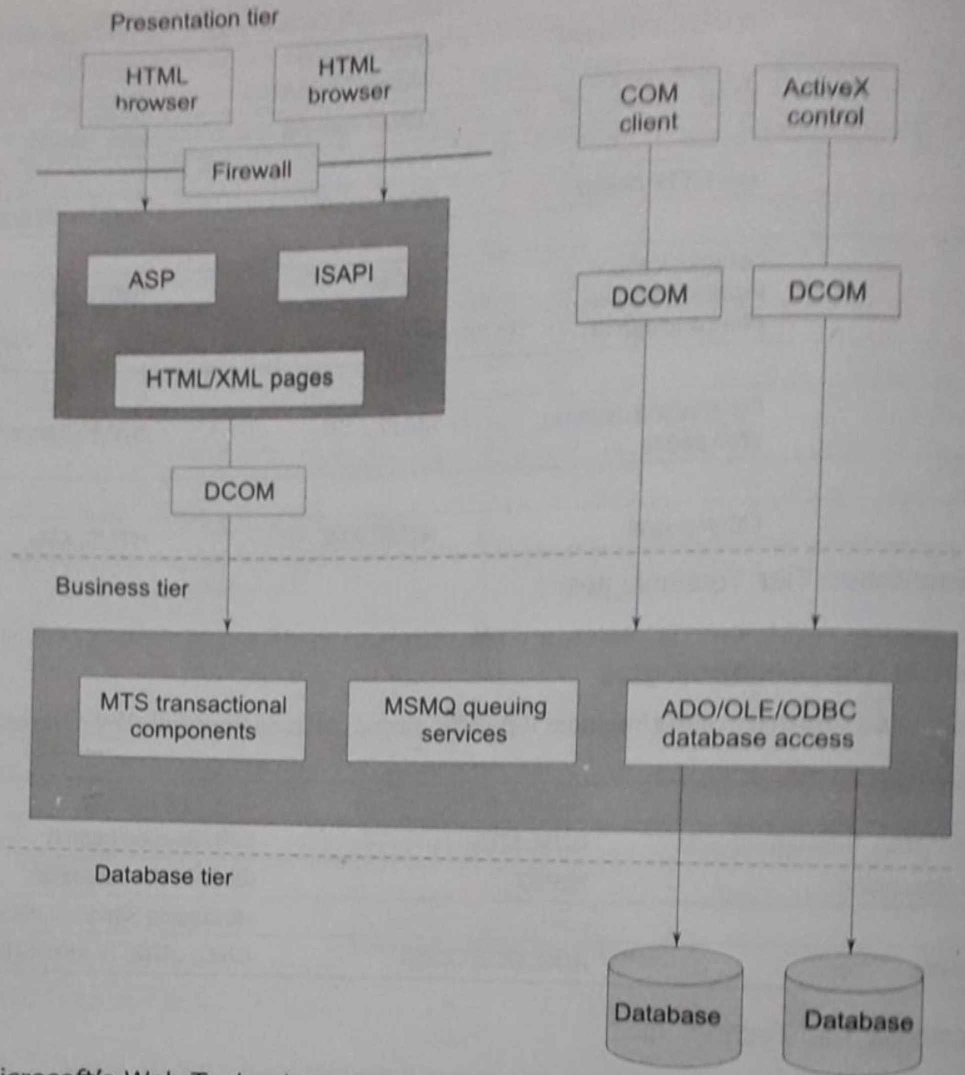

Fig. 8.8 Microsoft's Web Technologies

You can refer again to the tables and diagrams in this chapter when we discuss these technologies in detail in later chapters. This would give you an overall perspective of the various technologies available in the market, thus helping in understanding of what fits in where.

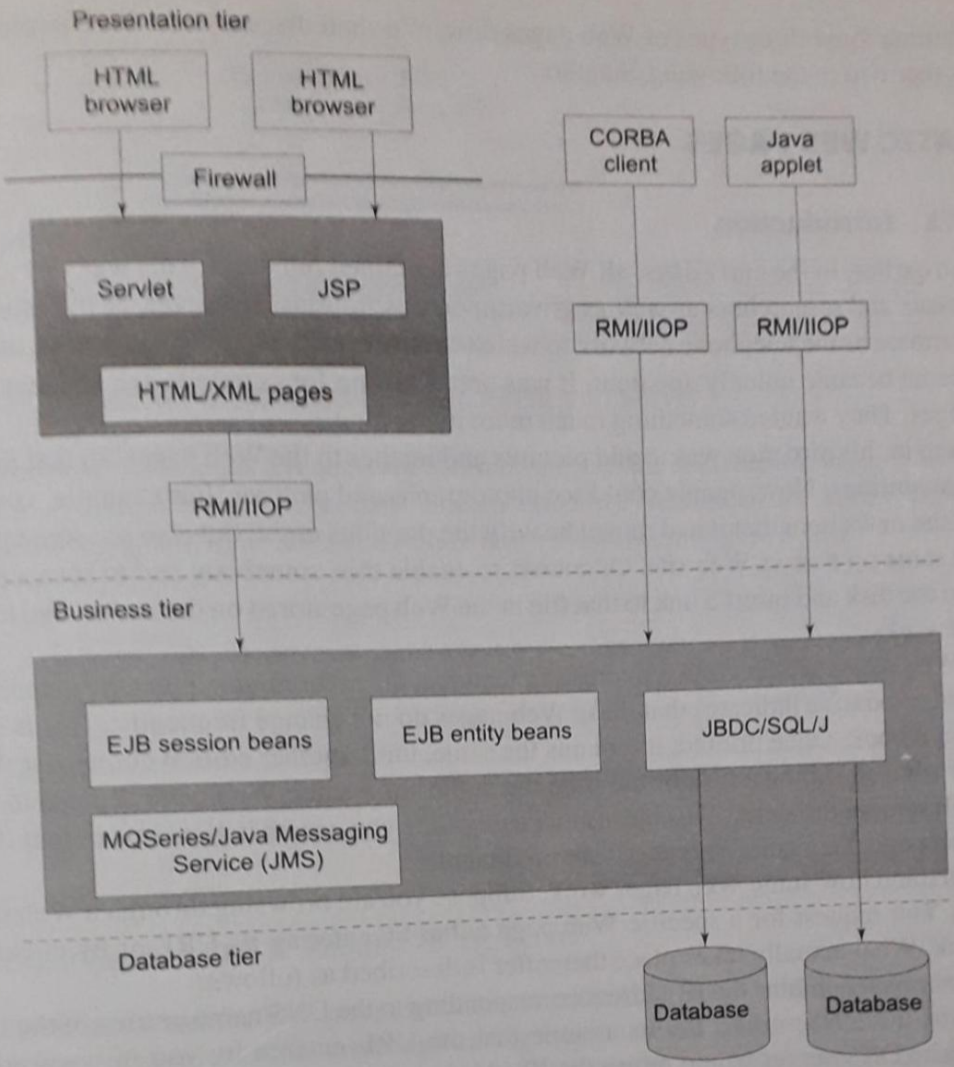

Fig. 8.9 Sun's Web Technologies

8.6 WEB PAGES

At a broad level, the various Web technologies that have emerged over the last few years have caused in the Web pages getting classified into three main categories: **static, dynamic** and **active**. This is shown in Figure 8.10.

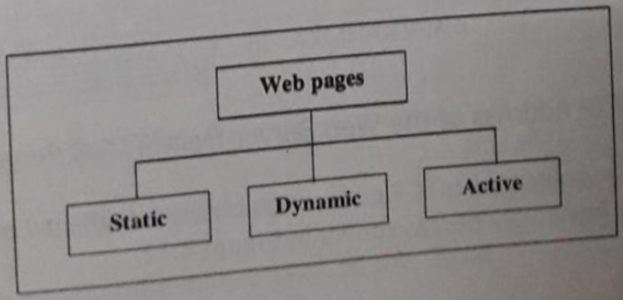

Fig. 8.10 Types of Web Pages