

Unit 7

Enterprise Application Architecture

Intro

- Java is used to build two types of applications; standalone and internet based applications.
- All the Java programs we learned and developed until now are standalone applications.
- These applications are either console based applications that spit the output to the console, or GUI based applications like word processors, paint tools etc where funky windows show up to display information.
- These applications are like independent applications that are used by several users and every user will have a copy of the application.
- Such Java applications are no different from applications built using other languages like C, C++ etc, except being platform independent.

- This is one side of Java that we all know about.
- The other side of Java is that it can be used build Internet based applications that can be used by millions of users at the same time.
- This is the true power of Java.
- Such internet applications are popularly known as enterprise level applications, as they form the faces of today's modern enterprises on the internet.
- So, from now onwards, we will use Java to build enterprise level applications that run on internet.

What is Enterprise Application?

- An enterprise or e-commerce application is an application that an enterprise or an organization uses to do business using Internet.
- The advent of internet has completely changed the world of communication. With the widespread usage of internet, businesses quickly realized a whole new market had opened up to exploit and started taking advantage through e-commerce.
- With almost all the businesses going online, the information assets of a company became more and more valuable.
- In order to sustain the competition, adoption of sophisticated technologies has become the key factor in exploiting the information assets of a business.
- J2EE is one such technology that helps businesses to build better and flexible e-commerce applications by securing the critical business data.

Key challenges of enterprise application

- **Performance:** Since the application is now exposed to millions of users, its very important that it responds faster enough to user requests. This is one of the most important aspects that play a key role in the success of a business.
- **Reliability:** The application must be reliable in terms of processing the business transactions successfully and accurately.
- **Availability:** It's very important that the application be up and running all the times with almost zero downtime. Few seconds of downtime can result in losing thousands of \$\$.
- **Security:** The application must be able to provide a secure environment during the

Core J2EE Technologies

- J2EE technologies are standard helper technologies that we use to build enterprise applications.
- Here is list of important technologies that every

Technology Name	Description
JDBC	Used for easy interaction with various Databases
Servlets/JSP	Used for building dynamic web applications in Java
JNDI	Used for interacting with Naming and Directory Services
JMS	Used for interacting with Messaging Systems
EJB	Component technology for building distributed components
Javamail	Used for sending Emails

Enterprise Application Architectures

- Since we are building enterprise applications in Java, we need to know the architecture of a typical enterprise application.
- Before we look into it, let's learn something about the various enterprise architectures that we currently have

2-Tier Architecture

- In this architecture, all the business logic and presentation logic of the application is embedded in the clients computer itself.
- Therefore if my application has 100 clients, I need to install the application on all the 100 client computers.
- However the database will still reside on a separate computer that will be shared by all the clients.

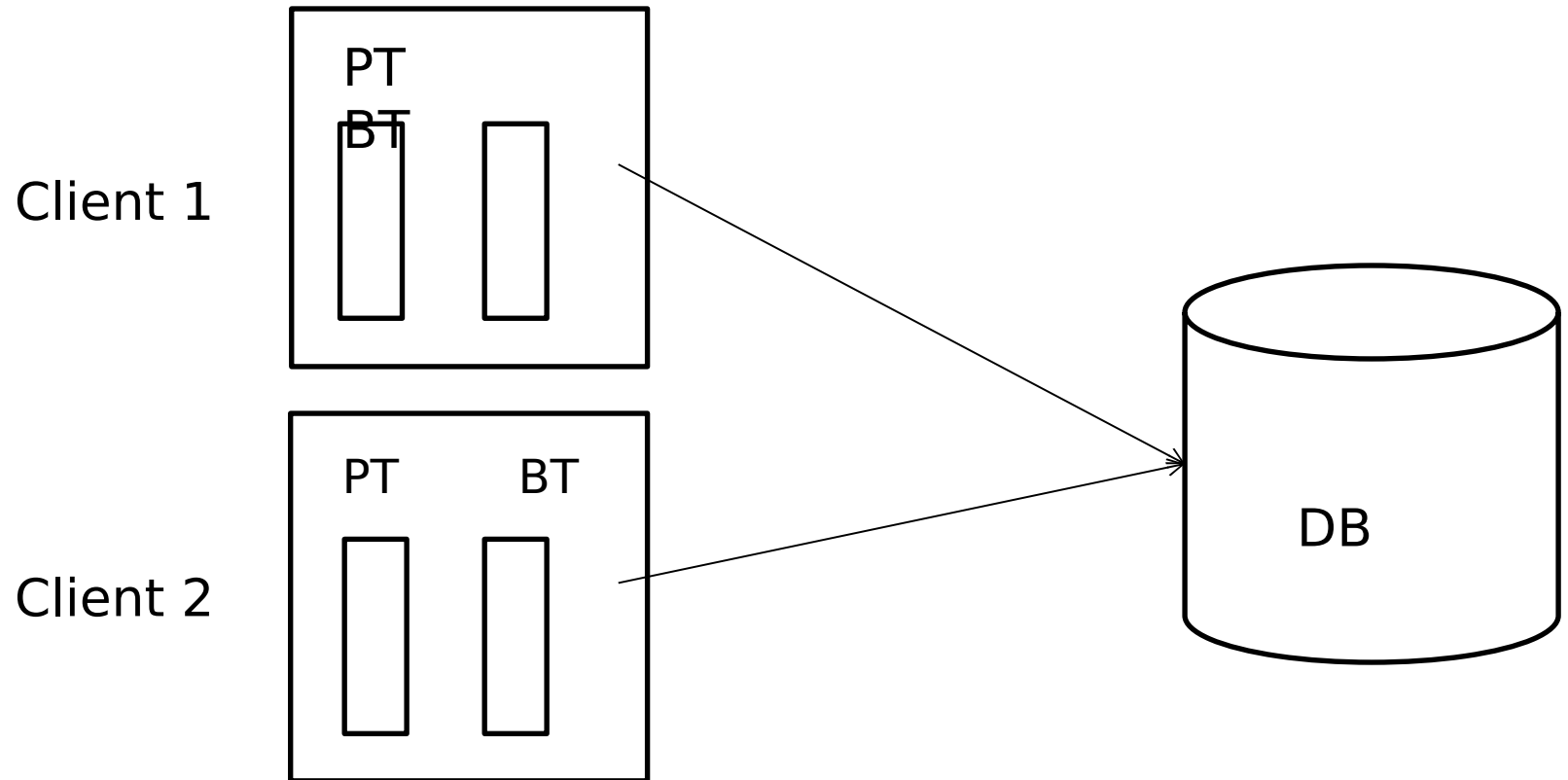


Fig: Typical 2-Tier Architecture

Advantages

- Simple and easy to build
- Low cost

Disadvantages

- Even a small change to the application requires a reinstall of the application on all the client computers.
- If some clients are not ready to take the changes, then multiple versions of the application will prevail, thereby causing maintenance nightmares.
- The performance of the application is dependent on the performance of client computer.
- Heavy network traffic due to multiple requests to the database from all the client computers.
- Scalability
- Low Security

The disadvantages of this architecture outweigh the few advantages making it less useful.

3-Tier Architecture

- In this architecture, the presentation logic, business logic and data are logically distributed in three tiers.
- The main difference is that the business logic is permanently isolated from all the client systems and moved to centralized middle tier.
- The presentation logic on the client systems will query the business logic in the middle tier which in turn accesses the data from the database.
- This architecture overcomes all the cons of 2-tier system.

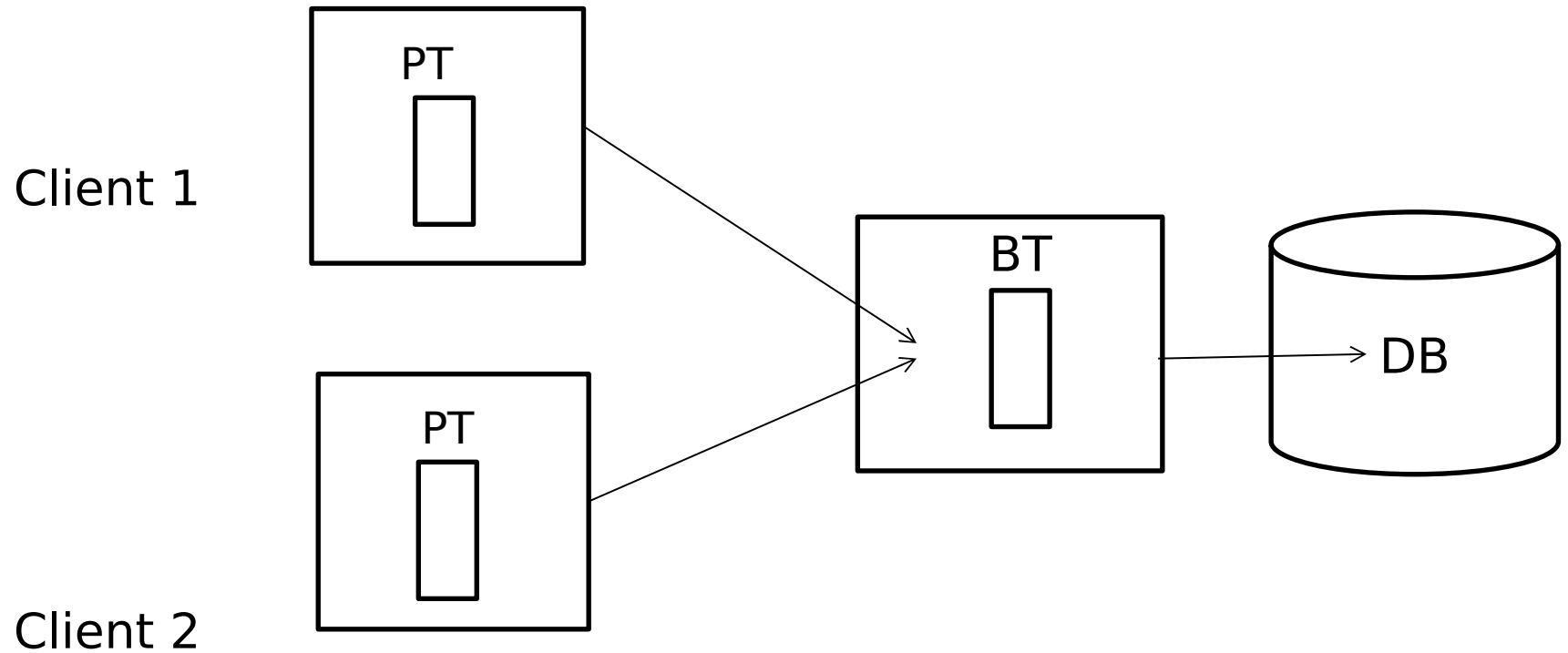


Fig: Typical 3-Tier Architecture

Advantages

- Centralized business logic will offer more flexibility. Business logic is only required to be changed at one place there by eliminating the installation process of the application on client systems.
- Less network traffic, thereby improving the performance of the application.
- Application performance is no longer dependent on client computer due to the business logic isolation.
- No more maintenance nightmares.

Disadvantages

- Any update to the business logic must be accepted by all the clients even if they are not ready for updates.

N-Tier Architecture

- In this type of architecture, the application logic is divided based on the functionality rather than physically like in 2-tier and 3-tier architectures.
- A typical n-tier architecture contains the following elements:
- User Interface: This is something like a web browser that handles the client interactions.
- Presentation Logic: This defines format using which the information will be displayed.
- Business Logic: Encapsulates all the business rules by interacting with data sources.
- Infrastructure Services: These are utility services that the presentation and business logic makes use of.

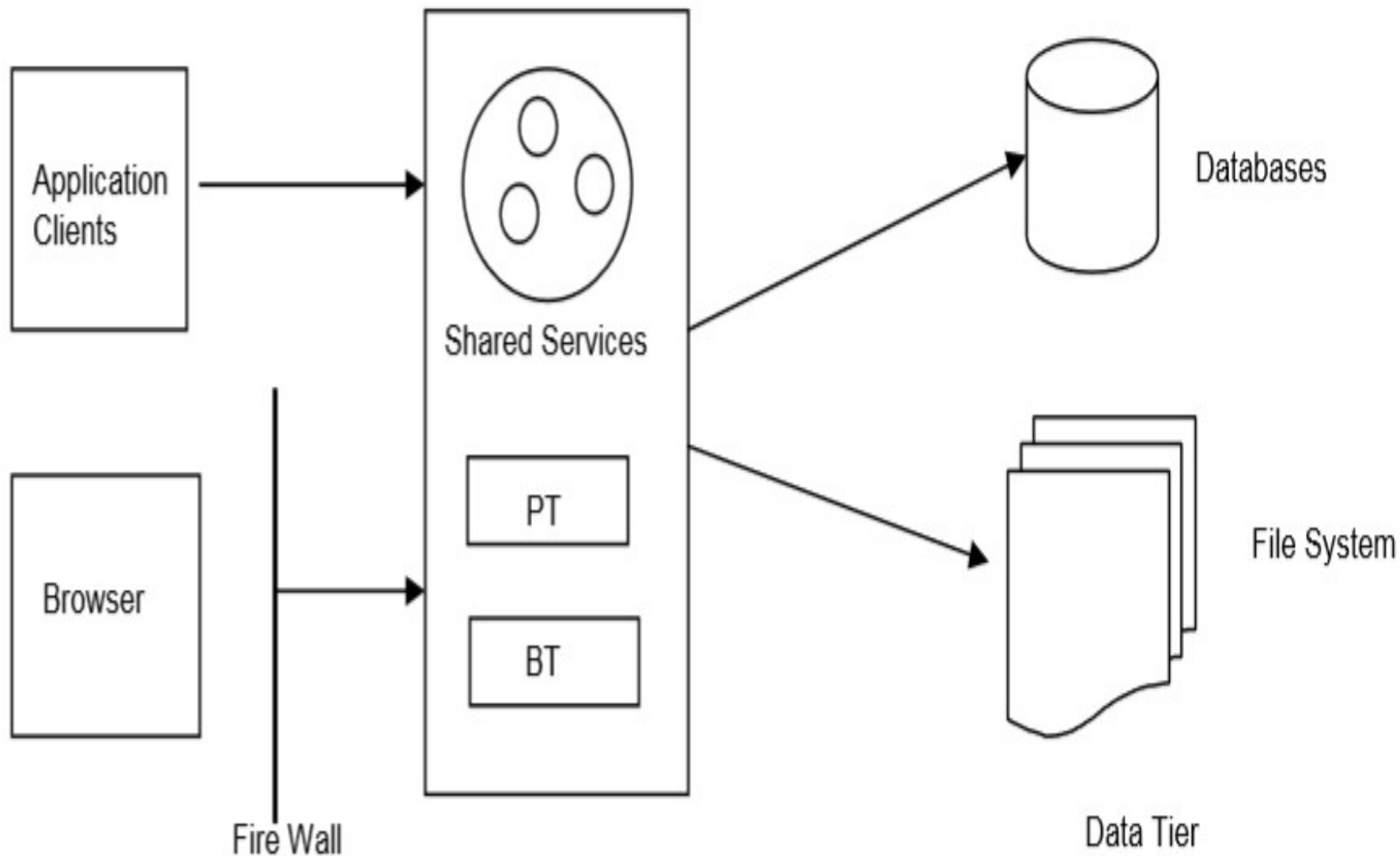


Fig 10.4 Typical n-tier architecture

- Breaking the application logic based on functionality offers several benefits like flexibility, better maintenance, improved performance, reusability of code and may more.
- This architecture is also referred to as Model-View-Controller (MVC) architecture.

Enterprise Architecture

- An N-tier architecture is applicable to a single application in an enterprise.
- However, an enterprise application is a collection of several applications within the enterprise with all the applications working in tandem and interacting with each other through well defined interfaces as shown in the following figure.

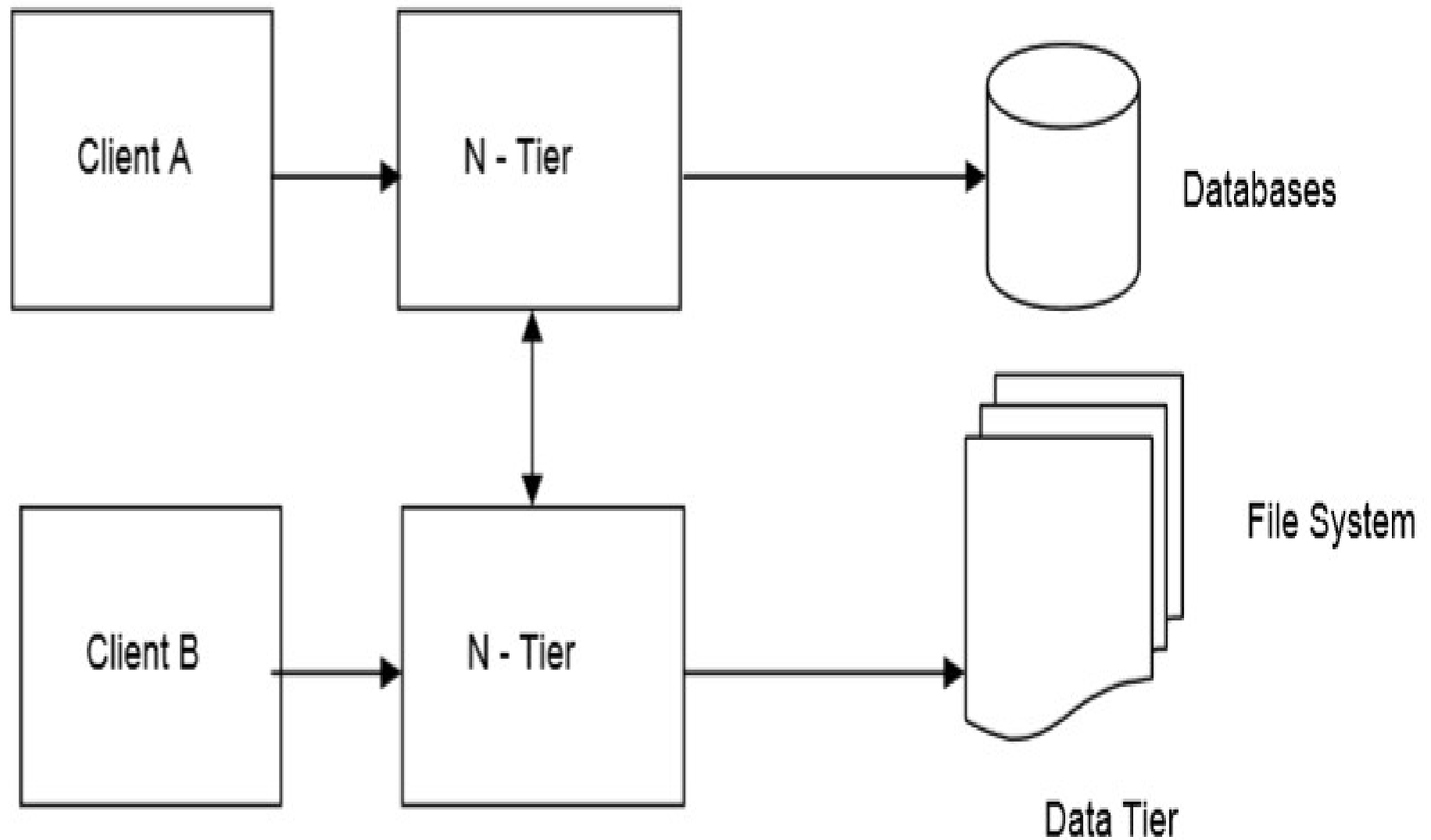


Fig 10.5 Enterprise Architecture

- Simply stating, enterprise architecture is a combination of several n-tier architectures.
- With this basic knowledge of various architectures, let's now try to understand somethings about application servers.

J2EE Application Servers

- Unlike simple Java programs, we don't use java command to execute the J2EE applications.
- We use a special program called application server to run the applications.
- However, we still use the same old javac command to compile the applications.
- An application server is the heart of any J2EE application.
- This is the one that runs the enterprise application on the internet.
- So, we can't even move a step forward without knowing what this is.

- A J2EE application server is a ready made sophisticated application that will host and run all the J2EE applications.
- There are several free and commercial application servers available in the market today that are developed by noted companies like Sun Microsystems Inc, IBM, BEA and many more.
- What is the purpose of all these companies developing the same application server?
- Here is the deal. Sun Microsystems first came up with the notion of J2EE platform that can be used to build large scale enterprise applications in Java. To simplify the development of such applications, they felt the need for the following two things:
 1. A Server that runs the applications
 2. Helper technologies to build the applications.

- They did the hard work and came up with the specifications for the above.
- These are popularly known as J2EE specifications.
- Having just the specifications is not good enough. We need someone to convert the specifications into workable realities.
- Therefore, Sun Microsystems decided to distribute the specifications to companies who can implement them.
- This is how IBM, BEA, Apache and several other companies came into picture.
- These companies used the J2EE specifications from Sun Microsystems, and built the two workable components listed above using their expertise.
- Once they built the application server along with the helper technologies, they started selling it.

- A company to build and host enterprise level applications in Java needs a J2EE application server.
- It will therefore hunt for those who sell the J2EE servers and come across several companies like SUN, IBM, BEA, JBoss and many more.
- After serious negotiations with the companies, it finally chooses the company who sells it for less \$\$ and more value.
- Once you have a J2EE application server, you can start building and running enterprise applications.
- All the companies that built the J2EE Application Server gave funky names to their product shown in the following table:

Company	Server Name
SUN	SunONE Studio Application Server
IBM	IBM WebSphere Application Server
BEA	BEA WebLogic Application Server
JBoss	JBoss Application Server

“A J2EE application server runs enterprise internet applications that are built using standard J2EE technologies”.

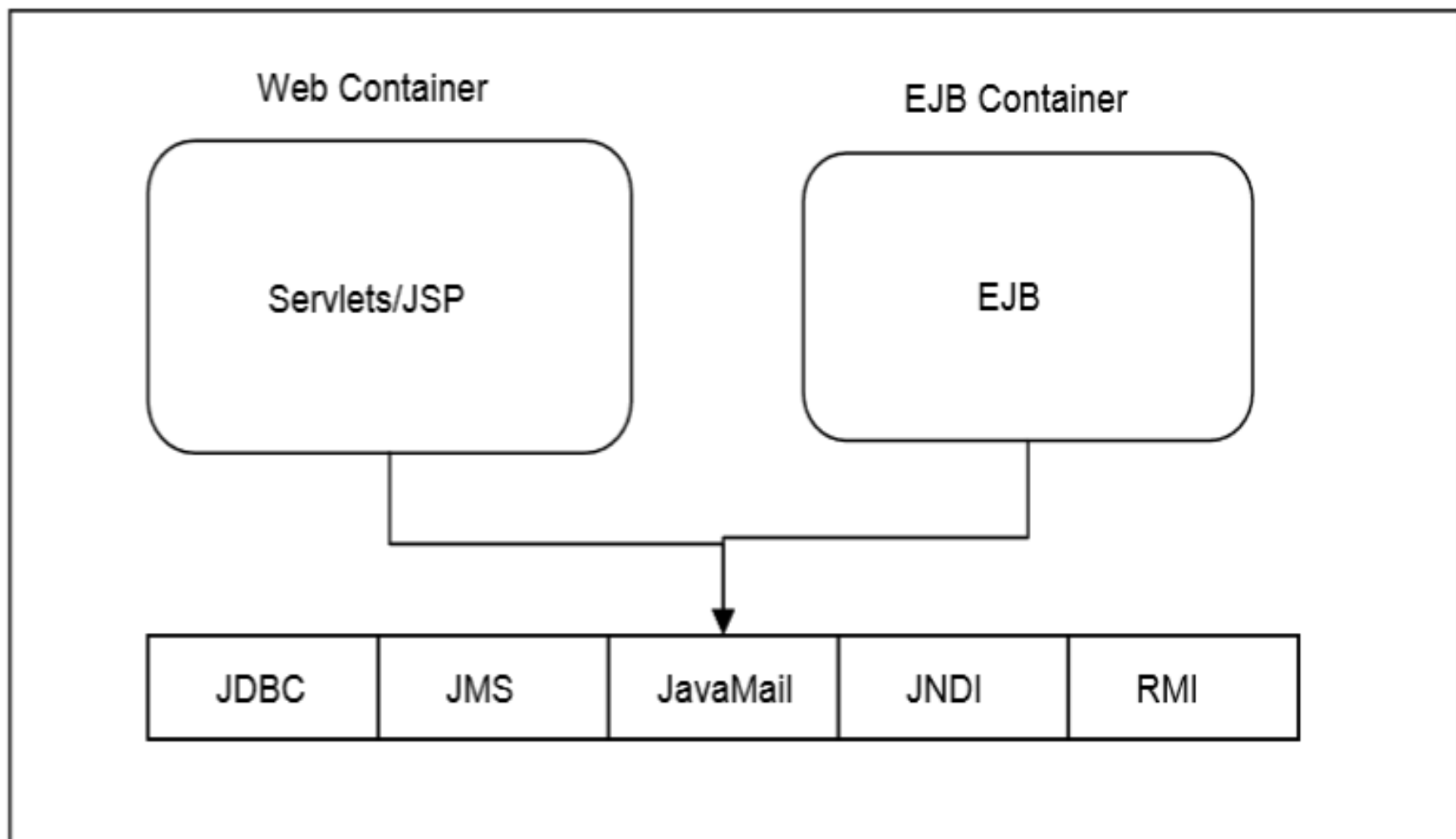


Fig 10.6 J2ee Application Server Components