RESEARCH METHODOLOGY

Main purpose of my research is to implement the setup of banking application in Middleware technology by using weblogic application server including the main feature of for banking application. Once the setup is completed, explain the security feature provided by weblogic application server because security is the major concern in banking sector. [12]

6.1 Objective

This chapter will explain the methodologies used, including the hardware, software, design and analysis and validation methodology.

Complete analysis that have been done previously to adopting the three tier architecture technology and what was the disadvantage and created a new testing environment through the middleware technology weblogic

Scalability: The key feature of 3-tier benefit is improved scalability since the application servers can be deployed on many machines. Also, the database no longer requires a connection from every client; it only requires connections from a smaller number of application servers. In addition, TP monitors or ORBs can be used to balance loads and dynamically manage the number of application server(s) available.

Better Re-use: The same logic can be initiated from many clients or applications. If an object standard like COM/DCOM or CORBA is employed (as discussed in tool dependence), then the specific language implementation of the middle tier can be made transparent.

Improved Data Integrity: since all updates go through the middle tier, the middle tier can ensure that only valid data is allowed to be updated in the database and the risk of a rogue client application corrupting data is removed.

Improved Security: Security is improved since it can be implemented at multiple levels (not just the database). Security can be granted on a service-by-service basis. Since the client does not have direct access to the database, it is more difficult for a client to obtain unauthorized data. Business logic is generally more secure since it is placed on a more secure central server.

Reduced Distribution: Changes to business logic only need to be updated on the application servers and do not have to be distributed to all the clients.

Improved Availability: mission-critical applications can make use of redundant application servers and redundant database servers. With redundant servers, it is possible to architect an application so that it can recover from network or server failures. [1][4][21]

Hidden Database Structure: since the actual structure of the database is hidden from the caller, it is possible that many database changes can be made transparently. Therefore, a service in the middle tier
that exchanges information/data with other applications could retain its original interface while the underlying database structure was enhanced during a new application release.

### 6.2 Cluster environment with high available and scalable mechanism

As per above figure the admin server will control all the resources in the domain. Admin server contain all the configuration information about the cluster, in above figure there are three managed are in cluster and an Admin server. When the servers in cluster are running, there is no effect on admin server with load-balancing and failover capabilities of the cluster. In live business environment, the administration server is not configured as part of a WebLogic cluster because it should be available for monitoring all other WebLogic instances and hence should not be overloaded with client requests. As per the above figure client will sent the request through load balancer and it route to managed server according the load on different server, if one server will already handling the load then request will route to another weblogic instances. And through admin consoler we can control, manage and monitor all the servers in cluster therefore admin console is single point of control to all resources in weblogic
domain and node manager should be there to control managed server like start and stop etc. As per admin view each WebLogic cluster is managed as a single logical entity in the domain. [36][2]

6.3 SSL Feature and implementation for better security

SSL (Secure Sockets Layer) is the main feature to secure any application and it is a commonly used protocol for managing security of transmission of a message on the Internet. There are two types of SSL as below.

One Way SSL

One way SSL authentication enables an application to authenticate or identify itself to the user (client).

Internet banking site is one of the best example of one way SSL authentication. When we open these types of sites it’s generally popup a warning. This popup shows us to certificate and these certificates are the authentication of the applications which we are going to access. If there is a lock sign at the right bottom corner of the status bar of the browser then double click on it get the details about the certificates. The certificates have expiry date and issuing authority details, which will confirm you the identification of the application. Means, the application which we are going to connect is SSL server and the browser which will connect to the application is a SSL client. SSL client initiates a contact with a SSL server. The SSL server presents a signed certificate (public key) to the SSL client. SSL client verifies the identity of the server with the private key of the server stored in it (client) and the authentication is complete.

![Figure 4 (One way SSL)](image)

Two way SSL

As the name indicate two way SSL, mean authentication happen from both sites. Means client authenticate to server and server also authenticate to client also, here in both server and client, certificates are present and client application verifies the identity of server application and server application also verifies the identity of the client application.
As I have discussed, here not only the server authenticates to the client, also client authenticates itself to the server, that’s why two way SSL is also referred to as client authentication.

Example of two way SSL would be applications which deals with sensitive and confidential data which is intended for a particular recipient. Thus the client who is having the certificates to authenticate itself to the server will only be able to access the application.

In the two way SSL applications, SSL client initiates a connection to a SSL server and server is set to use two way SSL client authentication. The SSL server presents it certificate [which is a public key of the server] to the client for verification. The SSL client verifies it with the private key store of the server. Then SSL server requests SSL client to send its public key to the SSL server to verify with the private key of the SSL client stored in the SSL server.

![Two Way SSL Implementation](image)

**Figure 5 (Two way SSL)**

### 6.4 SSL cycle

While internet banking, security is the major concern. While doing transaction our information should be secure and safe that means information should be transferred securely. There are various methods in internet banking to secure the data; one of them is monitor the integrity and security of the system.

Security of internet banking application is noted at different level. First concern is security of customer information because it’s sent from his PC to browser. Second is the security of the environment in which customer information resides in database and internet banking server. At last security measures are in place to prevent unauthorized users from attempting to log into the online banking section of the Web site.
Basically SSL (Secure Sockets Layer) control the data security between the customer browser and Webserver. SSL provides server authentication, data encryption and message integrity for an Internet connection. Apart from that SSL provides a security mechanism called handshake is used to establish the connection, once successfully handshake will done then client and server agreeing on the level of security they will use and fulfill authentication requirements for the connection. And it’s depending on browser how much level of encryption is there like 128 bit or higher.

After that the request passed from web browser to internet banking server. And this internet banking server is designed by three tier architecture and all the things like application server and application is deployed in application server however multi tier architecture provides a double firewall, completely isolating the Web server from the customer information SQL database.

Internet banking server send the information to database and security analyzer constantly monitors login attempts and recognizes failures that could indicate a possible unauthorized attempt to log into an account. When such trends are observed, steps will be taken automatically to prevent that account from being used.

Security concerns have been noted from every aspect within the architecture of the Internet banking application. Implementation of the SSL security protocol on the Web server and customer browser ensures authenticated data has been received from the customer. The three-tiered approach of the Internet banking application creates a double firewall which performs information requests over dedicated networks designed to handle specific functions. Placing all business logic and event logging within the Internet banking server creates a controlled environment which allows quick incorporation of Internet security technologies as they evolve. Finally, the security analyzer monitors login attempts in order to prevent unauthorized logins.
6.5 Software’s use for testing

Below software’s and mechanism I have used for complete setup

1. Installation of Weblogic.
2. Weblogic Domain Creation.
3. Database Installation.
4. Manager server configuration in weblogic Domain.
5. Nodemanager configuration.
6. Developing test application.
7. Deployment of test application in single server.
8. Deployment of test application in clustering and also test the failover concept.
9. Database user creation, permission granting and table creation for my testing application.
10. Security implementation

6.5.1 Version of the Software’s

Database

Oracle 11.2.0.2: Database is the main important part of any business environment. It’s used to the Physical data of any organization. I have used oracle database version 11.2.0.2.

OHS (HTTP) Server

Oracle OHS server: It’s an oracle web server just like attaché http server

Application Server

Oracle Weblogic Server 11G: Weblogic server is the Java J2EE based application servers.

Development Application

Application is created in J2EE and deployed in application server.

During my testing I have used Standard X86 Oracle Linux servers because they are globally used by all of the organizations in the market as well as fully certified by the Oracle itself for any FMW set up with weblogic and for other component also.
6.5.2 Testing Environment setup

Install weblogic, database OHS http server and created an application as per the above mentioned version.

After installing the weblogic in both machines (machine 1 and machine 2), create a weblogic domain only with admin server in first machine then copy the compete weblogic domain to machine 2 , for it below steps are executed.

1] Created a tar file in machine 1 with complete domain

2] Then transfer the tar file to machine 2

3] Then untar the tar file in machine 2

Once domain is created, started admin server from machine 1. Once the admin server is up, through console I have added managed server, cluster and deploy the application. As per my testing there are two managed and Admin server are in machine 1 and two managed server are there in machine2 and node manager is also configured on both machine, we can control all managed server through the admin console.

Install the database and created necessary user, schema and give the privileges as per requirement and finally created the table also. Then finally deployed a banking application on application server and access the application in different scenario and validate all the things through logs and from DB also.
7. CONCLUSION AND FUTURE PERSPECTIVE

The IT system that is currently used in banking requires a review and extension at time to time. That means the technology that we are using, should be proofed to suit integration of existing and future development platforms. It does not meaning like we need to replace the existing core system. Within a bank’s legacy systems reside vital components of the organization’s competitive edge, the mission-critical processes and systems that form the heart of the enterprise. They may require rationalization, documentation and better understanding, for the benefit of extracting more value, but nevertheless
they exist, and have been bought and paid for, and have proven reliability; processing billions of transactions per day across the globe.

In this regard core system should be identified individually and can be usable for future, so this is the way in that system development is easier and faster, and implementation and maintenance costs will be reduced. Middleware technology, SOA (Service Oriented Architecture) is a key technology through which we can re use the technology and avoids the extreme cost and risk of complete systems replacement.

A service-oriented architecture can provide a bank with the robust, resilient IT architecture it needs to grow, achieve speed-to-market and optimize customer service. It also provides a platform to help meet compliance requirements and assure security and integrity of information assets.

Now a day, IT market including banking and payment sector want to use the services that covering the business process are registered for usage across the enterprise. This way we can achieve the replacement of legacy application and can provide the common services to various applications for multiple bank or business processes around the globe. If customer uses all products with part of an overlapping service, the exact same handling in each product offering. Apart from that improved customer, this approach could also help decrease costs through shared usage.

All large organization is thinking about to implement the SOA technology to reduce the cost, faster and easier work. The ability to build applications faster can be easier to launch in market quickly and improve their value time to time. More specific to banking and financial services is the ability SOA brings to quickly integrate business units and companies.

SOA enable integration of core legacy services with future development platforms essentially a two-stage process. First, the legacy system must be converted into re-usable core business services. This means peeling away the original operator interface from the underlying ‘services’ in the applications that perform useful business functions. These core services are redeployed to their host platform in such a way that they can be invoked though some well chosen SOA-based middleware (e.g. web services). Second, these services are ‘published’ to the development teams building new client offerings, real time transfer and settlement systems, business reporting and approval processes, all using established international standards of communication. Published services appear as lists of fully production-hardened entities such as web services.

While most legacy systems will eventually need to replace, SOA enables banks to leverage what they have in the short-term without being forced to replace everything at once. In essence, SOA lengthens the life of legacy systems by enabling older systems to communicate with the core and work together.

SOA eliminates the need for new lines of code to be written to connect applications. Instead, standard protocols, such as Web services, can be used. Since new code will not have to be written by employees, their time can be better utilized by performing tasks that more directly affect the bottom line.
SOA allows for the elimination of many of the high costs that are often associated with integrating solutions; this leads to a greater ROI for applications. Perhaps the greatest benefit of SOA is the ability to integrate disparate solutions more easily. Therefore, banks are not limited in their vendor or solution searches and can deploy the applications that best meet their needs. This is especially important in a competitive marketplace where many banks are looking for best-of-breed solutions. While technology vendors are increasingly broadening their product portfolios as a result of mergers and acquisitions, banks do not want to be limited to the offerings of a single provider. SOA allows for greater flexibility in choices.

SOA security implements securities as a service through a hardware-based gateway and XML proxy that can parse, filter, validate schema, decrypt, verify signatures, access-control, transform, and sign and encrypt XML message flows. The security appliances is a server side security gateway that allows for all keys and tokens used to provide integrity and confidentiality to services exposed through the gateway to be managed at one point, the appliance. That means that clients invoking any number of services exposed through this appliance need only be configured to trust keys or tokens from this single gateway, rather than keys and tokens from each service. The appliance needs to be configured to trust keys and tokens from each client. Security appliances should implement the following features:

- Comprehensive security, All XML and web services security functions in one device
- Web services access control via new technologies (like SAML, XACML, WS-Security) and existing systems (like LDAP and SSO) to control access to applications
- Centralized configuration and policy management
- Performance, purpose, built to secure without degradation
- XML-based agility - future-proof for changing standards, policies, partners
- Appliance-based - drop-in device secures multiple applications at once
- Easy integration - inter-operates with and augments existing security systems

So, the conclusion is that current SOA technology is most popular in private banks and they are started to work on this technology but still in Government sector or especially in Government bank this technology need to implement.