

Application Architecture for Mobile Thin-Client Internet Applications

Most Internet applications follow a 3 or 4 tier physical architecture as shown in Figure 1. The web and applications servers may be deployed as a single or two separate layers. It may even be reduced to 2-tier if the database server is deployed together with the web and application server.

The correspondence of this physical architecture is closely related to the logical architecture and is presented Section 3.3 of this document.

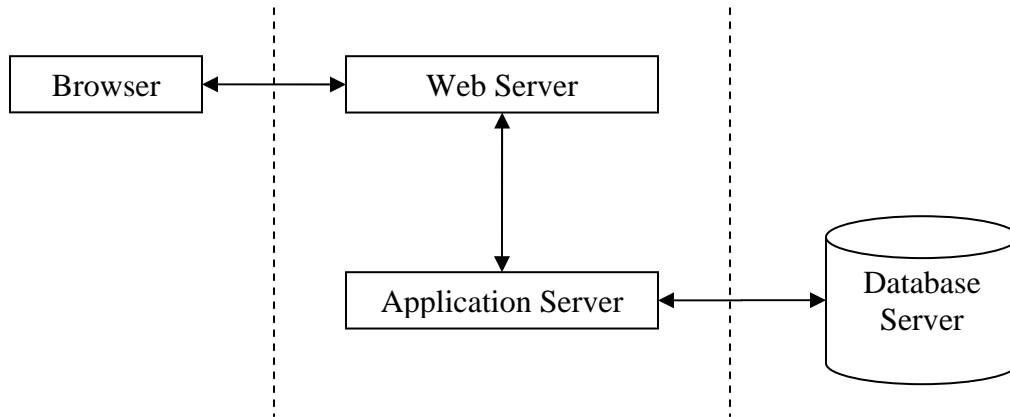


Figure 1: Typical 3 or 4 Tier Physical Architecture

Some advantages of using the internet architecture for mobile devices:

- Minimal or zero software deployment/cost: At most a small plug-in for example the Flash Player ActiveX for Pocket PC will be required for the applications developed by me
- Updates delivery not required: Upgrades to applications can be enjoyed by users upon their next visit. Installation of updates on clients not required
- Extension of Internet computing model: A new interface mechanism for existing computing infrastructure
- Familiar User Interface: The Internet Explorer is familiar to almost all Internet users and the familiar look and feel help new users
- Security: Data is securely maintained and backed up on servers, user do not need to worry about creating backups

Some disadvantages of this model:

- Wireless connectivity: Wireless connectivity is required for application use and this may not be feasible in the current scenarios
- Cost: Additionally wireless data access costs may not permit frequent use
- Security: Data is maintained on servers not controlled by users, privacy and security needs of users must be fulfilled

- Performance: Application performance may suffer due to network latency and traffic volumes. This may degrade user experience that would erroneously be attributed to the application

The Thin-Client refers to the browser and in this study for mobile devices, it is generally a 'micro-browser' supporting only a subset of standard features available on traditional desktop systems. Many of these critical issues with limitations of mobile device browsers and design issues specific to them are discussed in subsequent sections.

1. Mobile Thin-Clients

A thin-client relies on the server to perform most of the CPU and Data intensive tasks. By itself the thin-client is typically a browser that sends user requests to the server and displays the responses. Additionally, often there are special plug-ins that may be installed on a thin-client to perform specific functions or display certain special data formats. Mobile thin-clients follow this definition; in addition they are smaller, even less powerful and have more restrictions as may be expected.

Some important characteristics of mobile thin-clients for this study are:

- Small memory
- Less powerful CPU
- Wireless connectivity for data communication
- Limited display space
- Internet browser (Pocket IE on Windows Mobile 2003)
- Support for Cascading Style Sheets (CSS)

2. Backend Systems

The web, application and database servers run behind a firewall and all communication to the clients is over the standard HTTP+SSL protocols. Dedicated or shared servers can be used to deploy the various components

At Virginia Tech the servers used for my implementation are running Redhat Linux AS 4.0 on 32-bit, Intel x86 architecture. They are clustered with failover systems. MySQL database and Object Oriented PHP was used for data storage and applications development respectively.

3. Distributed System Architecture

Load Balancers are used to manage and distribute the load between redundant Web, Application and Database servers. This makes the system fault tolerant and provides for fail-over systems. This will also give better performance and shorter response time.

3.1 Architecture Objective

- Provide a scalable solution
- Address Security and Privacy concerns for all operations
- Provide high-performance and optimized operations
- Provide resilient, high-availability support
- Manageable, allowing easy deployment, monitoring and maintenance
- Interoperable with multiple devices

The distributed architecture proposed above has the following benefits:

- Multiple redundant servers
- Load balancers to manage and distribute the load
- Fault tolerant with fail-over systems
- Multiple security features
- Scalable solution and servers can be replicated to meet growing demand

3.2 Physical Architecture

A physical distribution of resources is proposed in this 3-tier architecture (Figure 2). Note the wireless access points providing 802.11 for LAN devices or data services (e.g. GPRS) by telecommunication providers.

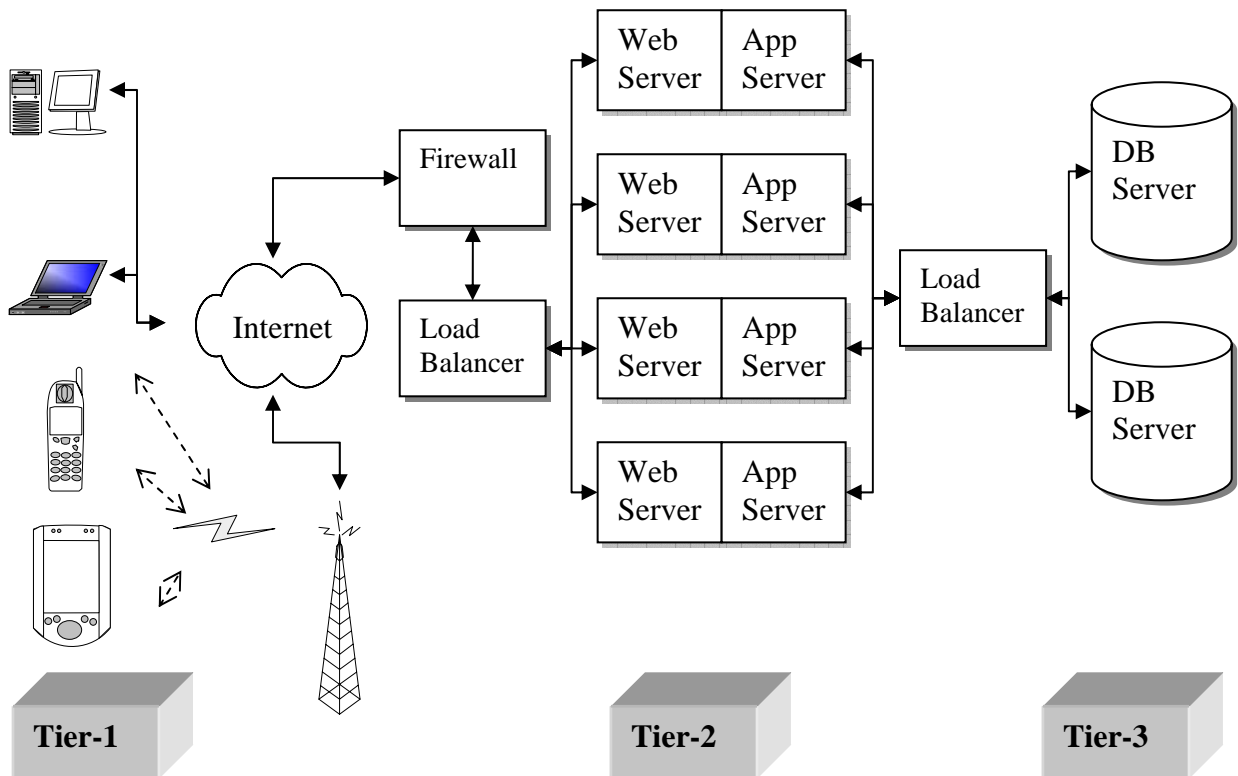


Figure 2: Physical distribution of resources

3.3 Logical Architecture

The architecture is organized in a typical 5 layer framework^{1,2,3}. The idea is to make the whole application highly modular. This will make it easy to maintain and also scale the application. For example the MySQL database could be easily replaced with Oracle or MS-SQL as the need arises.

Each layer is described in detail:

Layer 1: Data Source

Data storage and its reliability are critical to any application and this tier is an essential part of the system. Basically, it is the Database Management System (DBMS). Some examples include MS-SQL Server, MS Access, Oracle, MySQL, DB2, flat files (binary or text, including XML) etc. These products show a range of complexity available, from query optimization, indexing, etc to simple flat files. This layer is intended to deal only with the storage and retrieval of information. It is not concerned with the business logic or processes that use this information. This also should include your stored procedures.

Layer 2: Data Mapping Layer

This layer contains methods that provide a generic interface to the data storage. This abstracts out the data storage details from the rest of the layers. For example changes are required only at this tier if replacing a database technology, for example methods for creating and opening a connection to the data source. It therefore provides a reusable interface to the database.

Layer 3: Domain Business Logic

This is basically the “brains” of the application containing the business rules for user interface control and data flow. For example when completing a survey it determines which question to display to the user and how to store the response. This layer is neither aware of the presentation methods nor the data storage methods. It has no direct access to the database which is mediated via the data mapping layer.

Objected Oriented PHP is used for its implementation in my system and OOP knowledge is required to understand all object instantiations and data flow methods.

Layer 4: Server-Side Presentation Preparation

This layer consist of the standard ASP, JSP, PHP etc documents and provides the model for the user interface layout. It works with the information made available from the

¹ <http://msdn.microsoft.com/architecture/>

² <http://java.sun.com/blueprints/enterprise/index.html>

³ <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/distapp.asp?frame=true>

Business Logic layer and transforms it into meaningful and readable presentation for the end user.

Layer 5: Client Side Presentation Layer

This comprises the end-user presentation. Client requests are forwarded by the browser to the Business Logic Layer which process the request and delivers an appropriate response via the Server Side presentation.

A summary is presented in Table 1 and diagrammatically in Figure 3.

Table 1: Logical Layers

Layer	Description
Layer 5	Client Side Presentation Layer – HTML, JavaScript, Cascading Style Sheets (CSS). The client browser (Pocket Internet Explorer 4.0 or later) with support for CSS. Will display the web forms and content requested by the user.
Layer 4	Server-Side Presentation Preparation – ASP.NET, JSP, PHP
Layer 3	Domain Business Logic – EJBs ⁴ , .NET Managed Components
Layer 2	Data Mapping Layer – JDO ⁵ , ADO.NET Data Access Components. Components can map relational data tables and data from XML web services as well
Layer 1	Data Source – Oracle, MS-SQL Server, MS Access, MySQL, Flat files (text, XML, binary)

⁴ <http://java.sun.com/products/ejb/>

⁵ <http://java.sun.com/products/jdo/>

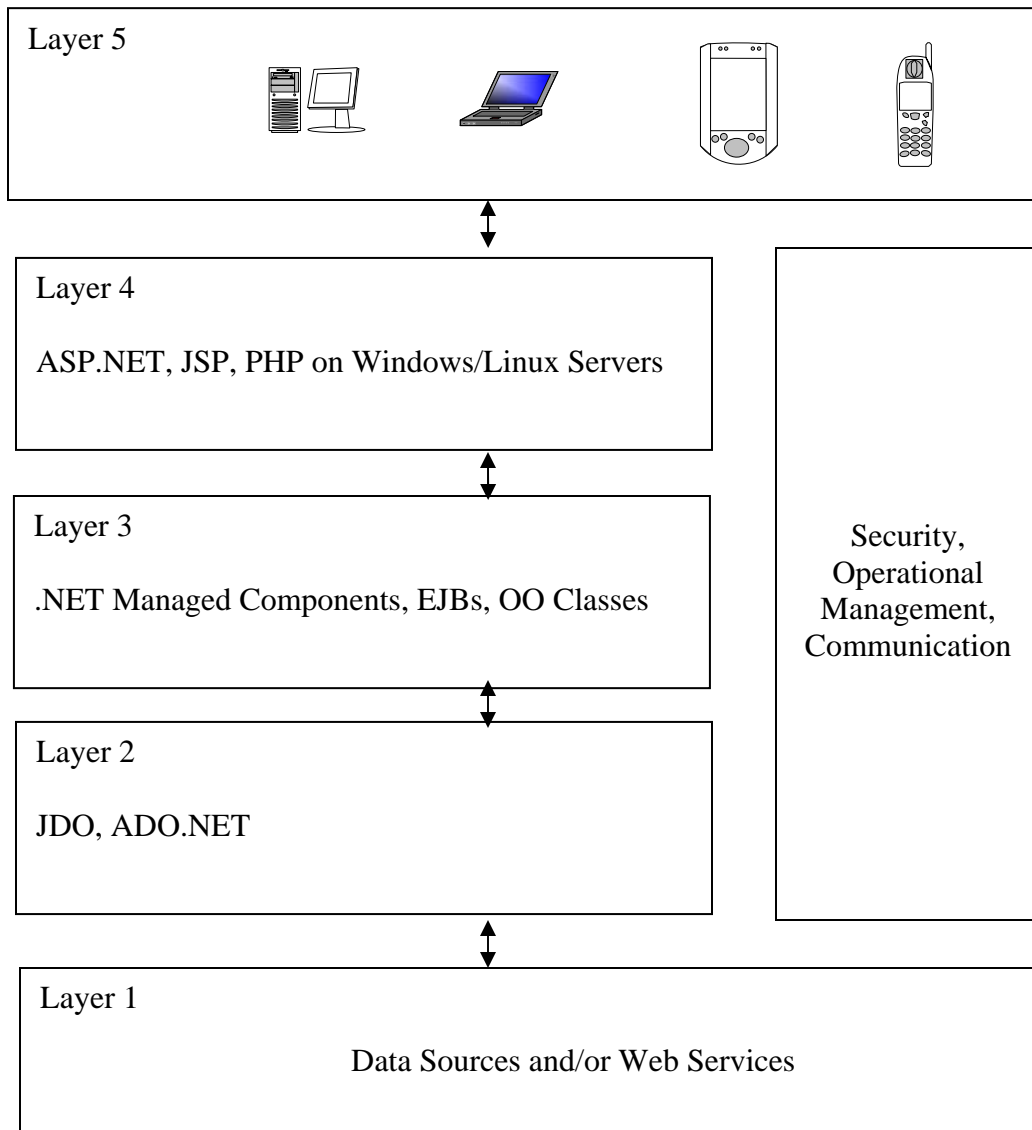


Figure 3: Logical Layers

3.4 Security for Mobile Thin-Client Internet Applications

Wireless users more vulnerable than wired users as discussed and here the specific points of weakness are identified and ways to secure the application are suggested. Figure 4 shows the typical scenario of wireless mobile device deployment using the Internet application model.

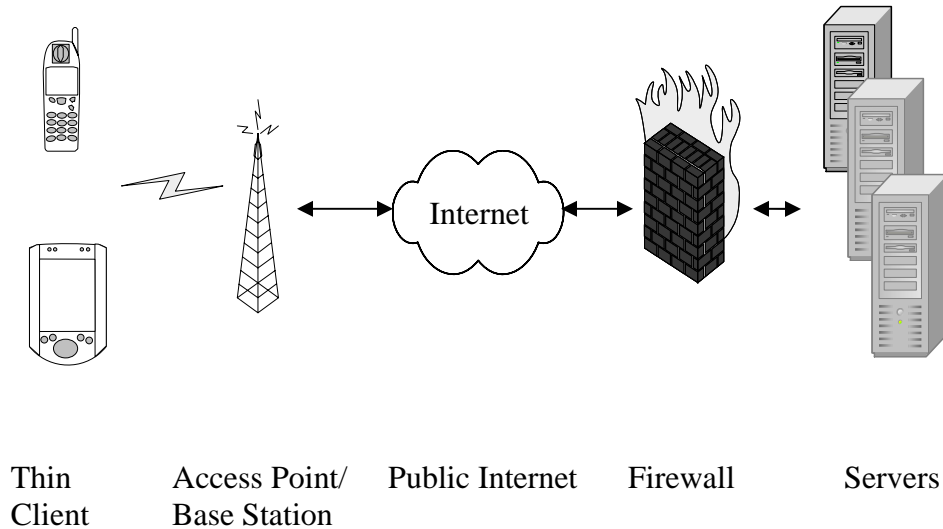


Figure 4: Security Vulnerabilities in Mobile Environments

This suggests the following vulnerable points:

1. **Mobile Device:** The device itself is susceptible for a security breach. It might get stolen and permit access to unauthorized users. Such issues are beyond the scope of this discussion. Additional security concerns might be OS vulnerabilities, viruses, etc that are analogous to desktop computers. These too are beyond the scope of this discussion
2. **Mobile device to access point/base station communication:** This communication over the air is follows protocols discussed under “Wireless Networks for Mobile Devices” and there are serious security concerns as described in “Wireless Communications Security” for 802.11 networks. For Internet applications an additional security layer is provided by using SSL⁶ over HTTP. The applications being developed will use SSL to provide end to end security and HTTP+SSL is a well recognized industry standard
3. **The Access Point/Base station:** This point requires physical safety and administrative control. If controlled by malicious users it can compromise the user data. Such issues are again beyond the scope of this discussion however End-to-End encryption provided by SSL is a good way to ensure that the data is safe in transit.

⁶ http://en.wikipedia.org/wiki/Secure_Sockets_Layer

4. The Public Internet: This public switched networks going over different geographies and provided by multiple carriers on its own provides no security assurances. Again End-to-End encryption provides a required layer of security.
5. Firewall and Servers: These are controlled by the application provider and proper physical safety and configuration is required. The firewall protects the infrastructure by allowing only valid protocol requests. Also ensure that all known OS vulnerabilities on the servers have been patched.

Application Level Security

This security has to be provided by the application developers. For example a user who successfully logs into an online banking site is not shown someone else's account information! For mobile surveys to be conducted the following are some considerations:

- Authenticated users are displayed only the surveys they are required to complete
- Security checks to ensure that malicious GET/POST of control data can be overcome
- Cookies are not used on client devices, instead server sessions are used
- Non persistent connection by mobile devices is common due to network availability or device shutdown due to inactivity. Require authentication for every start of session regardless of time elapsed to prevent session hijacks