

Medical Billing Software: Types of Architecture

Searce Healthcare

Mihir Karia

Hardik Parekh

Shamik Parikh

October 04

TABLE OF CONTENTS

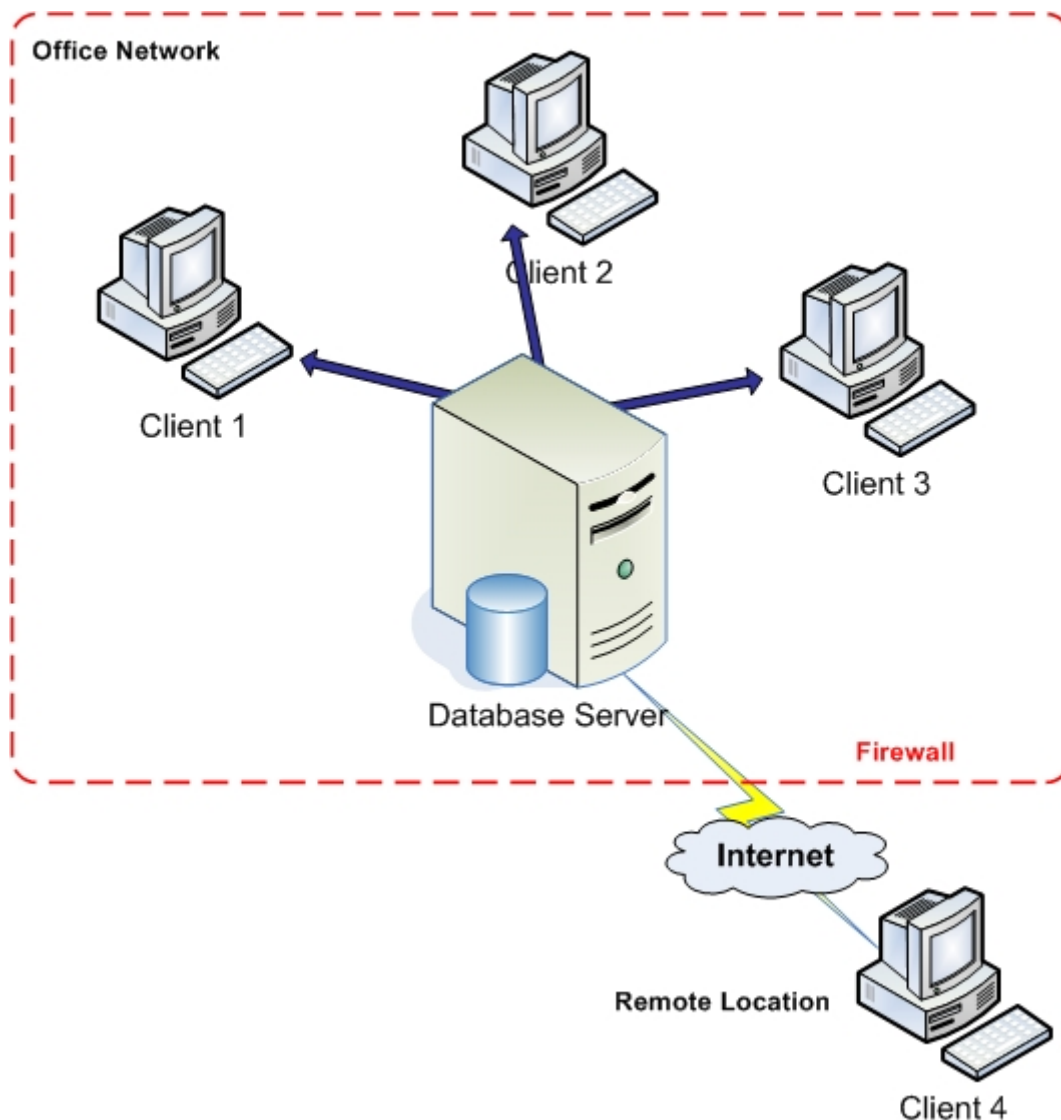
Introduction	3
1. Client / Server	3
2. ASP HTTP/Web Based Application	4
3. Hosted Data with TCP/IP Sockets Based Client Access	6
4. Peer-to-Peer	7
Conclusions	9

Introduction

There are three predominant software architectures found in the medical billing software industry. This document provides an overview of each and outlines some advantages and disadvantages inherent in using a particular architecture. It is not meant to be an in-depth discussion of the available architectures; rather, it seeks to provide a high-level baseline for further analysis and evaluation criteria.

1. Client / Server

This is the traditional client/server model. The client/server software architecture is a versatile, message-based and modular infrastructure that is intended to improve usability, flexibility, interoperability, and scalability as compared to centralized, mainframe, time sharing computing. A client is defined as a requester of services and a server is defined as the provider of services. A single machine can be both a client and a server depending on the software configuration.



The user hosts and stores the data. There is usually an associated and dedicated database containing patient records/statements and claims data. Application users access the data via client applications installed on individual workstations. Client applications are often sold under separate license. For remote users, either employees or Business Associates, a broadband connection is required for data transfer from the server to the remote workstations. Each client communicates with the server application, which regulates and controls access to the database.

Pros:

- Complete control of billing software and hardware
- Might be faster to access depending on how the clients communicate with the server
- Hosting organization controls scalability to handle growth

Cons:

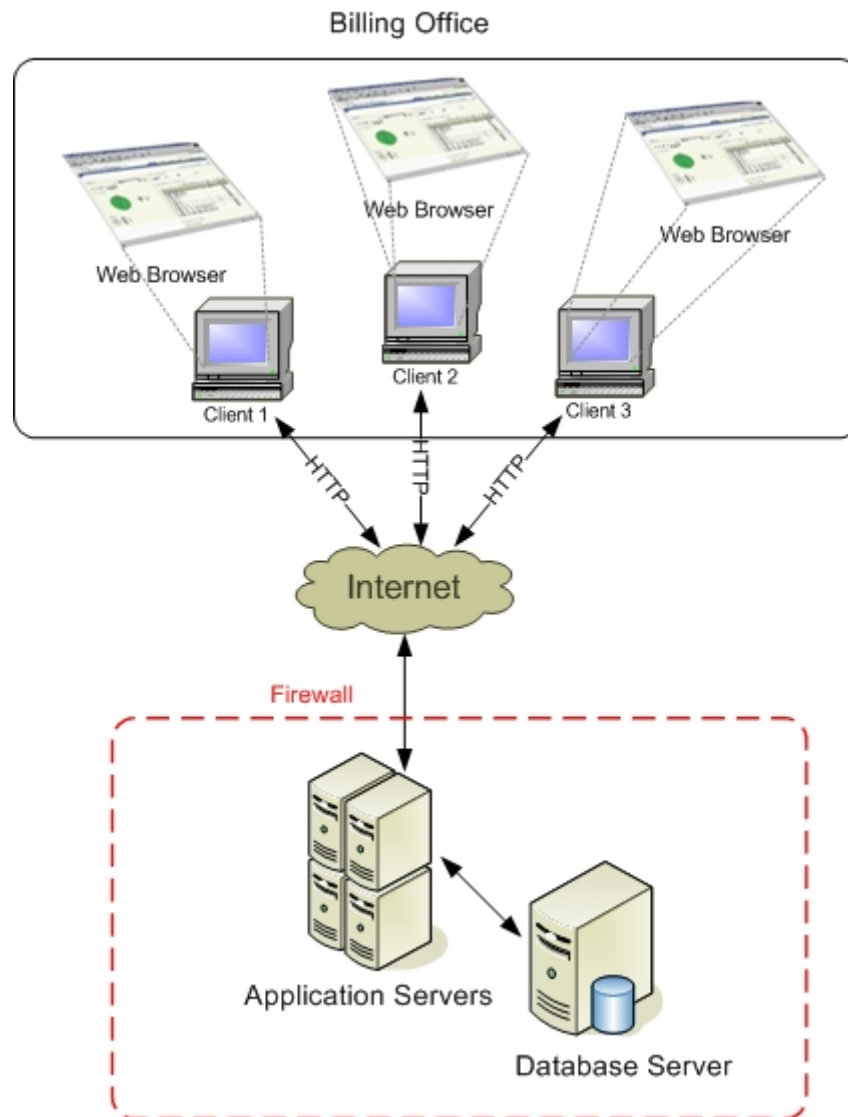
- Higher up-front cost in software, hardware
- User is responsible for administering software & platform
- User is responsible for database management and backup
- Software upgrades could be cumbersome and costly
- While scalability is in the control of the hosting organization, additional hardware may be required and servers may need to be upgraded to accommodate scalability.
- Maintaining uptime, redundancy and high-availability could get costly
- Many vendors rely on Windows Remote Access Service for client/server communications. RAS is not a very reliable or industrial strength solution.

2. ASP HTTP/Web Based Application

Application service providers (ASPs) are a new type of service providers that make the whole spectrum from low-end applications like e-mail systems to high-end applications like enterprise resource planning (ERP) management systems available to customers over the Internet through browser-based user interfaces (application hosting). This relieves the customers from installing and managing the applications themselves in their own data center. Instead, the customers pay a fee for accessing the application over the Internet.

An Application Service Provider works as a third party organization to provide access to applications/software to multiple customers over private networks or the public internet. These applications are provided via rental or lease contracts. An ASP typically deploys, manages and controls application software, system hardware, and networking at a centralized facility on behalf of the customers.

Server and database are both web based. The browser is the client and makes HTTP based requests which are fulfilled by the server within an ASP (Application Service Provider) web hosted environment.



Pros:

- Provides better uptime and redundancy as well as 24x7 support. Cost effectiveness in achieving predictable and accountable availability is very difficult to duplicate in-house.
- Platform independent
- No investment in server hardware and dedicated broadband. Organizations may subscribe to inexpensive internet access from cable companies or DSL lines from local phone companies.
- Application Server & Data are both managed by ASP. This includes regular data backups.
- The pay-as-you-go model is often significantly less expensive for all but the most frequent users of the service.
- Eliminates specialized IT infrastructure for the application as well as supporting applications.

Cons:

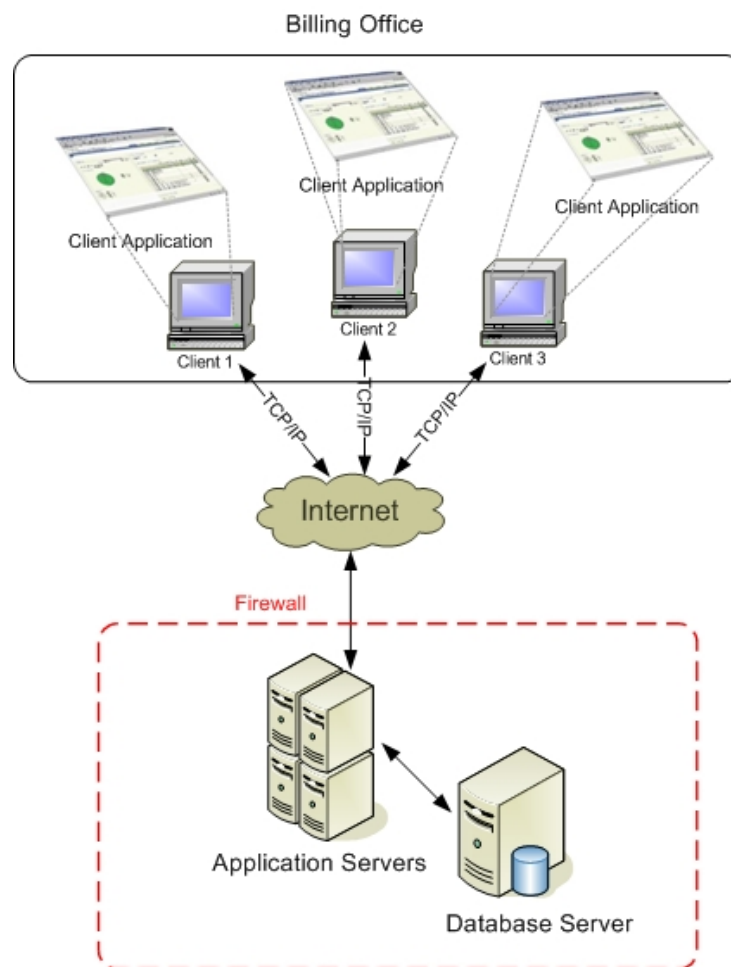
- Less control over Mission-Critical Software Applications (applications considered to be vital to core functions of the business)

- Browser-based user interface is limited and cumbersome to use
- HTTP protocol is stateless. Session and state management becomes complex. Generally, this means that a user needs to complete a particular task/screen before going on to another. This can slow down the claims process and increase costs.
- Difficult to customize the application to your unique or specific needs. Some ASPs may provide customized solutions at a higher cost.

Related to state management, normally, caching is turned off using html Meta tags. The browser is forced to load all elements of a screen, including images, for every request. This hampers productivity.

3. Hosted Data with TCP/IP Sockets Based Client Access

The service provider hosts and stores the data and manages the same. Users access the data using client applications installed on individual workstations. Client applications access the hosted data over the network using TCP/IP sockets.



Applications built using such an architecture are referred to as internet based applications. These are distinctly different from "web" based applications. Web applications use the HTTP protocol while internet applications use proprietary protocols and do not suffer the state and contextual limitations of the HTTP protocol.

Typically, the client application handles all the business rules processing while the database server is dedicated to managing queries and updates to the database. Network efficiencies are achieved since only data gets transmitted both ways. There is no transmission of user interface elements or images since these are embedded within the client applications.

Pros:

- All advantages of web based applications apply
- All disadvantages of web based applications are mitigated

Cons:

- Client software needs to be installed on individual workstations
- Software upgrades could be cumbersome depending upon how the vendor implements this.

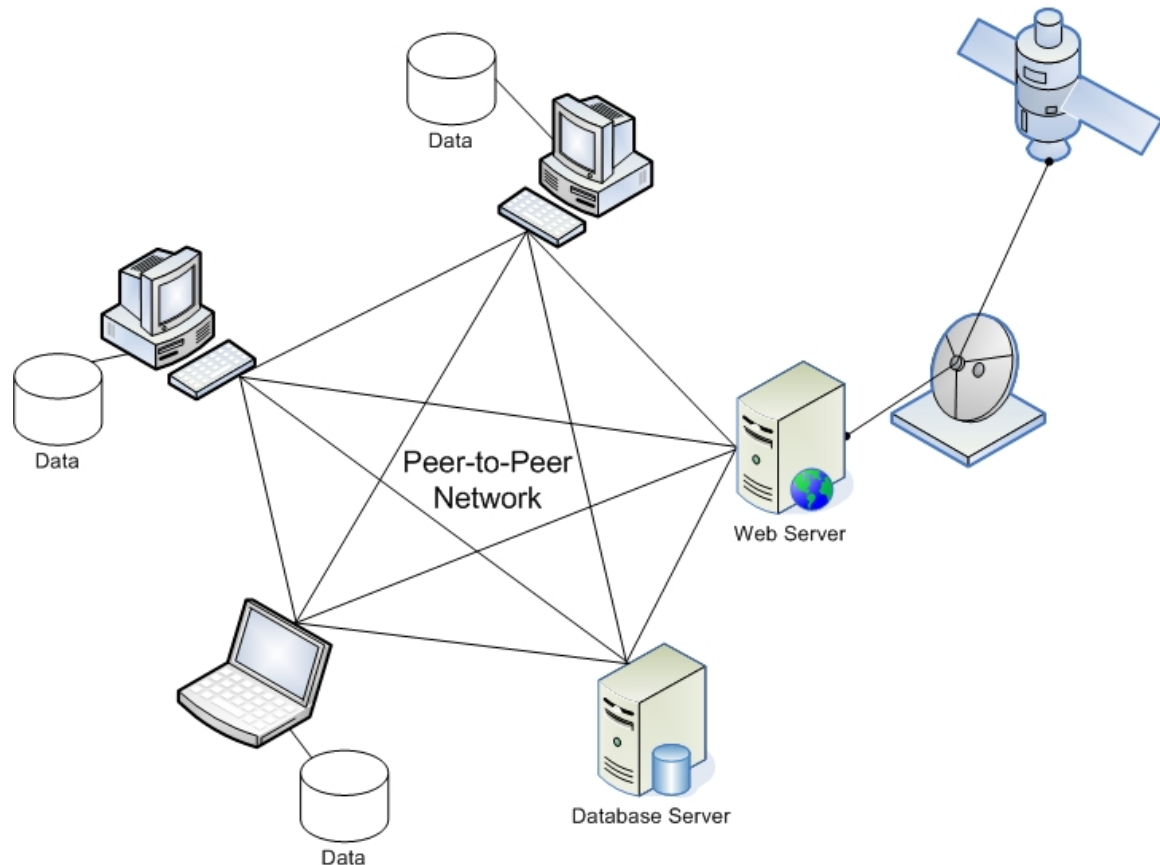
4. Peer-to-Peer

Peer-to-Peer is a distributed network architecture where the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers etc). These shared resources are necessary to provide the service and content offered by the network including file sharing or shared workspaces for collaboration. Each machine is accessible by other peers directly, without passing through intermediary entities. The participants of such a network are thus resource (Service and content) providers as well as resource (Service and content) requestors (Servent-concept).

With P2P computing, each participating computer, referred to as peer, functions as a client with a layer of server functionality. This allows the peer to act both as a client and as a server within the context of a given application. P2P applications build on such functions as storage, computations, messaging, security, and file distribution, when handled through direct exchanges between peers.

A peer can initiate requests, and it can respond to requests from other peers in the network. The ability to make direct exchanges with other users liberates P2P users from the traditional dependence on central servers. Users have a higher degree of autonomy and control over the services they utilize.

This framework provides the capabilities that allow peers to directly interact. An important characteristic of the direct-interaction capability is that the computing environment becomes decentralized. Many end-user applications become possible through the P2P services, including storage, computations, messaging, security, distribution, and more. What unifies these application types is sharing of resources with some form of collaboration.



This framework provides the capabilities that allow peers to directly interact. An important characteristic of the direct-interaction capability is that the computing environment becomes decentralized. Many end-user applications become possible through the P2P services, including storage, computations, messaging, security, distribution, and more. What unifies these application types is sharing of resources with some form of collaboration.

Peer-to-peer architecture can be distinguished into two separate types – “pure P2P” and “hybrid P2P.”

The term “pure P2P computing” refers to a model, such as Freenet, where all participating computers are peers. No central server is used to control, coordinate, or manage the exchanges among the peers.

In the “hybrid P2P” computing model, the application relies on a central server to perform some of the required functions. The degree of involvement varies with the application. For example, Napster requires the user to first connect to a control server, where the directory of all available files is stored.

Pros:

- Hardware costs are minimized as compared to client server architecture, since no server is required
- P2P provides the opportunity to make use of vast untapped resources that go unused without it. These resources include processing power for large-scale computations and enormous storage potential
- The P2P infrastructure allows direct access and shared space, and this can enable remote maintenance capability

- When P2P is used within the enterprise, it may be able to replace some costly data center functions with distributed services between clients

Cons:

- It is limited in extensibility, tends to overburden user workstations by having them play the role of server to other users
- Unless configured meticulously it is largely unsecured
- Generally used for less than 4-5 users
- User is responsible for database management and backup
- Maintaining uptime, redundancy and high-availability could get costly

Conclusions

While each architecture has its advantages and disadvantages, selection of the right software depends on several factors including the size of the organization, current architecture, future need, etc. The web-based and internet-based software are easy-to-deploy and scale for small businesses. For a small set-up (less than 4 users), peer-to-peer architecture is less expensive as compared to client-server model. It remains to be seen whether the price per transaction/claim sways decisions one way or another, provided that the functionality, features and complexity of comparable products with varying architectures is similar.

From the perspective of HIPAA compliant remote access to the server and the data, the client/server and peer-to-peer model might prove a little more challenging when compared with web-based and internet-based architecture. A true ROI and cost of ownership analysis is beyond the scope of this paper.



Privacy Policy and Disclaimer

All Rights Reserved. No part of this document may be photocopied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means whether, electronic, mechanical, or otherwise without the prior written permission of Searce.

The information contained herein has been taken from sources believed to be reliable. Searce disclaims all warranties as to accuracy, completeness or adequacy of such information. Searce shall have no liability (including liability to any person by reason of negligence) for errors, omissions or inadequacies in the information contained herein or for any interpretations thereof. The reader assumes sole responsibility for selection of these materials to achieve its intended results. Searce reserves the right to change details in this publication without notice.

Copyright © 2004 Searce. All Rights Reserved.

36 Compton Court, Basking Ridge, NJ 07920, USA

■Tel: +1 908-758-0782 ■Fax: + 1 908-349-3339 ■Email: info@searce.com

www.searce.com