

White Paper:
**The advantages and tradeoffs of the
Peer to Peer Client/Server networking
model to the property management
enterprise**

copyright 2006 by Just So Software Inc., all rights
reserved

Table of Contents

Introduction and Scope	3
System Organization	3
Description of the PPCS Mechanism	3
Capability of a web client vs a PPCS client	4
Capability of an SQL “Smart Client” vs a PPCS client	5
Comparison of Networking Technologies and Implications of PPCS	6
Comparison of security issues between Web-based systems and the PPCS system	8
Consideration of SQL Security and Advantages	10
Conclusion	11

Introduction and Scope

This document will describe to the property manager or IT specialist the capabilities and advantages of the Peer to Peer Client/Server (PPCS) networking model, as employed by The Property Manager Release 5, over any web-based or SQL-based networking model when deployed for the property management enterprise. We assume the reader is at least casually familiar with web-based and SQL-based network organizations, so we won't spend much time describing those. Instead, we'll focus mostly on how PPCS compares and contrasts with those.

We show that a web-based system is seriously lacking in capability compared to the PPCS system, and we further show that there is absolutely no relevant advantage and a number of relevant disadvantages to using an SQL-based system instead of a PPCS system, when the goal is to integrate the property management enterprise.

System Organization

The Property Manager is an application that is built on top of a commercial database product (Superbase). Superbase incorporates a unique peer to peer client server networking capability, which has been deployed as part of The Property Manager to provide a package that has unique capability. This provides The Property Manager with the capability to deploy a virtual private network, providing very high security and reliability.

The networking mechanism is defined as "peer to peer" because it runs on any workstation - no server operating system is required - hence the computers are peers. It is Client/Server because the organization of the communication is Client-Server, with one workstation functioning as a server for the particular transaction and the other functioning as the client, and with the communication occurring via a proprietary encrypted protocol that runs on UDP. Given that the workstations are peers, who is a client and who is a server can change dynamically from one transaction to the next.

As a consequence of this unique capability, The Property Manager may be operated in three different modes; as a standalone desktop database product, as a networked application using standard Windows NETBEUI/NetBIOS capability, or as a networked application using PPCS. In the latter case, which is employed by the Enterprise Edition and by the Hosted Edition, networking can occur without regard to geography, can be distributed across multiple diversely located computers, and no dedicated server operating system is required.

Description of the PPCS mechanism

A web based application uses some type of web server with a server operating system, and the end user accesses the property management software package by using a web browser. This mechanism keeps the network bandwidth requirement down, but makes it possible to network a property management application only with a number of limitations and restrictions. Typically, the user (client) is restricted in the capability that she has available; commonly, for instance, the type of reports that the user can generate is strictly limited, and the capability to generate

communications with tenants automatically is both limited and inflexible. An SQL-based solution may relieve the “limited capability” issue but usually will have a larger number of network transactions than the web-based solution, with a consequent (though often slight) speed impact.

Conversely, a typical desktop system has full capability locally. It may work well across a LAN, but only at the price of heavy LAN bandwidth usage. It does not work well across the internet because of the network latencies and the high bandwidth requirements.

The PPCS mechanism follows a middle ground between these two extremes. Basically, we split the database engine between the portion that services the user and manipulates the data and the portion that accesses the data and filters it. So, the portion that services the user (the client) will communicate across the network with the portion that accesses and filters the data (the server) to request information. The server portion will then send only the data requested to the client, which will then perform any post processing and display it for the user, or store it, or whatever else is required.

This mechanism limits the amount of data shipped across the network to levels similar to that seen with a web-based solution, while providing a "fat client" with the complete capability that one would expect to see in a desktop system.

Functionally and syntactically, the PPCS connection resembles that of an SQL environment in that a client is querying a database server which then sends the requested data. Here the comparison ends though because PPCS uses UDP instead of TCP, is both stateless and connectionless, and has a full database on both ends of the connection. Because PPCS is so tightly coupled on both ends (both ends must be a Superbase module), the connectionless nature of UDP is a substantial advantage, both reducing network overhead and enhancing security, while the unreliable nature of UDP is not a factor; if the data doesn't arrive the client just asks the server again.

Capability of a Web client versus a PPCS Client

The essential characteristic of the Web is that it permits a server to build a page and display that page to a client. The client may interact with that page in certain limited ways, such as entering data, or clicking on links. As the web has grown, other capabilities have been added through the use of plugins, but often these additional capabilities are unsatisfactory for a number of reasons. This is certainly true of all web-based property management applications; even the best of them have very limited capability compared to a full-up desktop application.

This is because, in spite of the add-ons, the essential feature remains: the web server builds a page and displays it to the client. Almost all the work is done by the server, and the client can specify work for the server to do only in specific pre-defined ways. This is inherently limiting.

In the PPCS environment, the client IS a full-up desktop application. The only thing it does differently is that it goes out on the internet to obtain its data. The PPCS server does not build a page and display it to the client; it responds to a client's query and ships data to the client, which the client then manipulates exactly as if the client had obtained that data locally.

The consequence of this is that the PPCS client has all the performance capability and flexibility of a desktop application while having all the networking capability of a web-based application. Also, the server does a lot less work and the client does a lot more work in the PPCS environment, therefore you are making better use of your resources.

Finally, the client is not absolutely dependent on the server, the way a web-based application is. Local caching using the same database format as that used on the server is easy to implement; if the server goes down or the network connection is lost, the client does not have to stop.

The result is more power, more flexibility, more capability, more reliability, and much higher security than is available with a web-based product

Capability of a Dedicated SQL “Smart Client” versus a PPCS Client

A handful of property management software packages employ a “smart client” that connects to an SQL server on another computer to both read and write data. Such a system can be programmed to look a lot more like the PPCS environment than the web-based system and can in principal have some similarity of advantages. It does provide the advantage of a “fat client” with full capability, just as the PPCS environment provides.

Such clients, however, can’t cache locally using database technology unless they have a copy of an SQL server also installed on the workstation, and therefore they remain vulnerable to being unusable if the network or server goes down. There are licensing issues and substantial costs associated with deploying more copies of an SQL server; this author is not aware that any of the property management packages available have deployed such a caching capability.

Because these clients don’t cache locally, they are more dependent upon the network for collecting data, and this inevitably makes them slower than a PPCS client that does local caching, especially as the day advances and the local cache contains more and more information. An SQL server running on fast server hardware may very well provide the data faster than a Superbase server running on much more modest hardware, but the primary factor in internet communications is not usually speed of data access, but time spent transiting the network. More transactions equals slower overall response.

Further, the network connection between such a smart client occurs at a higher level in the programming environment than that which occurs in the PPCS environment. The consequence of this is that programming errors or configuration problems are far more likely to lead to security issues since each end of the connection is handled by a different vendor (the author of the SQL server on one end and the author of the smart client on the other end). Because the PPCS connection is totally handled on both ends by one vendor, while the protocols which create the application are all that is necessary for the application vendor to write, you can see that the PPCS connection is far less likely to have security or reliability issues. Essentially, the PPCS connection is handled as internal to one application unlike the SQL connection which is a connection between two different applications.

Finally, SQL servers use TCP. This protocol can't hide itself because it requires the establishment of a connection. Thus, when an attempt to connect is made, TCP responds - which announces to a bad guy what kind of software is behind that open port and immediately suggests a means of attack.

Comparison of Networking Technologies and Implications of PPCS

In a Windows networking environment using NetBEUI or NetBIOS, most of the work is done by the client; the server merely serves files and the communications are tightly integrated, with network shared resources appearing as local resources on the client machine and functioning nearly identically to true local resources. While this provides an easy capability to network applications and a near-transparency of the network to the user, it results in high network traffic and requires substantial bandwidth. In a busy office environment, this can result in substantial delays in receiving data and in completing operations. Further, this type of networking is nearly useless over the internet because a combination of high bandwidth requirements, long network latencies, and heavy traffic results in very poor performance.

For work across the internet, a client-server model is generally employed, where a client application (most commonly a web browser) interfaces with a server application (most commonly a web server on a server operating system) to get work done. The integration is very loose and the user knows he is working across a network. Usually the client system is little more than a viewer; the vast majority of the work is done on the server. Communications are stateless if possible, and if state information must be maintained it is often necessary to ship that state information back and forth with every transaction. The major advantage of this approach is that it vastly reduces network traffic compared to the NetBEUI type of environment, thus shifting the bottleneck off of the communications medium and onto the server, where it can be managed by the use of fast hardware. The disadvantage is that the client (which usually is a very capable system) doesn't have much work to do (thus wasting capacity) and there are usually substantial overhead penalties levied on the server by this type of software

This technology is adequate for many cases, and most large property management organizations use it for lack of an alternative. The downside is that client-side applications have very limited capability because a browser environment just doesn't support many of the capabilities that are useful in a property management office environment. Furthermore, the server environment requires a server operating system, and usually dedicated support personnel, and often onerous licensing requirements. Also, the server represents a significant single point of failure, that can take down the entire business if it or the network connections to it fail. Finally, web servers are widely targeted by malicious people with malicious software, and maintaining security can be a very large problem.

If the client is a "Smart SQL Client" then the local capability is reminiscent of that capability of a desktop application, which eliminates the "lack of capability" argument that applies to a web browser. But this client still doesn't get around the need to have a dedicated server operating system, including an (often expensive) SQL Server package in place. Further, unless the client

goes to the additional programming effort and cost of establishing local caching, the single point of failure represented by the server remains. SQL Servers can implement automatic streaming to accomplish “on the fly” server backup, but implementing this capability requires yet another dedicated server, again raising costs. Many property management organizations simply don’t spend the money to implement and support the capability.

The PPCS environment used by The Property Manager manages the performance bottleneck issues differently than the typical web-client/web-server application, and somewhat differently than the “Smart SQL Client”/SQL Server application. PPCS provides a peer to peer networking environment using a client-server model, where both the clients and the servers have substantial capability. In the PPCS environment, most of the work is done by the client just as in a desktop system. The client consults the server only to obtain data that is needed for the client to do its job - and much of this data will then be locally cached. The advantage is that client side applications can be as fully featured and flexible as straight desktop applications. In fact, The Property Manager uses the exact same module as the desktop app and as the PPCS client, with the choice of which mode to use made by a single software switch.

The PPCS environment also features extreme flexibility. The Property Manager can make multiple simultaneous connections to geographically diverse data servers in order to obtain data. This is an advantage in that it removes the single point of failure that the server operating system represents, and a disadvantage in that it can make network setup more complicated. Furthermore, workstations hosting The Property Manager can be simultaneously a client AND a server. There are many potential advantages to this, and the essential disadvantage is that this requires the typical workstation to have fast hardware in order to avoid being a bottleneck itself. In a modern computing environment fast hardware is de rigeur, and therefore this disadvantage really isn’t one.

Presuming the system serving the data is not overworked, performance in the PPCS environment is governed very strongly by network latencies. When working across a LAN, the performance of a PPCS-networked The Property Manager is very close to the performance of an ordinary desktop version for most purposes, though some big reports are noticeably slower than the desktop version. Generally, the user on a LAN won’t notice delays in serving data that are sufficient to cause irritation or frustration. Further, if the existence of the property management software package has been the major cause of delays on the LAN, deploying The Property Manager in PPCS mode will eliminate that bottleneck.

On a LAN, ordinary applications such as entering rents, checking tenant payment history, or reviewing who owes money happen quickly enough that the user won’t notice the difference at all. Browsing through the buildings list happens with no noticeable delay, though populating the tenant list box may involve a slight delay (up to about a second) depending on server speed and workload.

In an internet environment, The Property Manager’s performance will, for most common day to day functions, appear to the user to be similar to that of a web-based application hosted on a fast server. This is because the dominant factor in using PPCS across the internet is network latencies; bandwidth is not usually an issue if the client and server both have broadband

connections. Some functions, such as generating a monthly building report, will be noticeably slower than a LAN or desktop version if done via PPCS across the internet. This is because this report involves extensive database searching and filtering and even though the report has been optimized to minimize network communications there are still a lot of communications required.

A key advantage of The Property Manager is the graceful degradation that it undergoes in an internet environment. Should network communications fail, or should a central server go down, The Property Manager will automatically enter fallback mode and will continue to work using locally cached data. Naturally, capability will be somewhat restricted because the server is not available, but usually the user will be able to perform the most common day-to-day functions, even absent the server. When using a web-based solution, if the network goes down all work stops until it comes back up. In the event of a significant server issue, this could take days

Furthermore, with The Property Manager backups and preparation of alternate servers to accommodate problems and outages becomes simple; any workstation can be a server. Thus if dual writing is enabled, a “hot backup” is always available and, if necessary, that remote location can quickly take over operations. While this capability can be prepared and maintained in the usual web-based server environment or in the SQL environment, most property management organizations do not do it because of its relatively high incremental cost. With The Property Manager, this incremental cost approaches zero.

Comparison of security issues between Web-based systems and the PPCS system

The security of internet transactions is a problem of considerable magnitude and a source of considerable debate in today’s world. It seems like every day there are news reports about compromises in security leading to the leaking of personally identifiable information about individuals.

In the property management environment, this danger must be taken very seriously because the property management organization inevitably collects massive amounts of information about tenants, certainly sufficient information to facilitate identity theft. Aside from the moral and ethical obligations that the property management organization has to safeguard this data, there are the legal implications; the organization that fails to safeguard the information can find itself being sued and losing as a consequence.

Web-based property management systems are inherently insecure. This can be seen by the fact that web servers are the most commonly attacked systems, and the enormous quantity of viruses, trojan horses, spyware, and assorted other malware that so commonly attacks web browsers (most notably Microsoft’s Internet Explorer). Even when SSL connections are used, web browsers remain vulnerable; in the past weaknesses (programming errors) have been identified in Microsoft’s implementation of SSL and it is entirely possible that further weaknesses will be identified in the future. Thus, when you use a web-based property management system, you are using a protocol and an environment that is widespread, historically recognized as insecure, and continually under attack. When you employ this protocol to transmit sensitive information, you are playing russian roulette.

Beyond that, web-based property management systems employ plugins on your web browser in order to provide functionality on your side of the connection. Plugins are notorious for reliability issues and for opening security holes into your system, either due to programming flaws in the plugin or due to the fact that the need for the plugin causes you to lower security in your web browser so that it can run the plugin - and the browser certainly has security holes; this has been seen over and over.

Further, the ubiquity of web browsers means that any of your employees can sit down in an internet café or a public library and access your property management system. This is an enormous security breach; using an unknown computer (whose security is probably compromised) to access sensitive information pertaining to your business and your tenants.

There are a variety of ways that web-based systems try to secure themselves (such as the challenge-response mechanisms employed by banks and brokerages) but these mechanisms are somewhat painful to use and encourage the unmotivated user to find ways to compromise the security in order to make her life easier. Further, there is absolutely no assurance that these mechanisms will be successful in all cases.

The best way to avoid the security issues associated with the web is to avoid the web, and of course to employ appropriate “lock down” procedures on your servers and workstations.

Our PPCS mechanism does not use the web. It establishes a virtual private network between the client and the server, and uses an assortment of mechanisms that are intended to maintain security. The PPCS server does not respond to any requests for information, unless that request is properly formatted. Thus, it gives nothing away to an attacker who might be trying to identify the application that has the port open on your server.

The PPCS server expects to only talk to other PPCS systems; the protocol is proprietary, and therefore only a properly configured PPCS client can talk to the PPCS server. Further, by virtue of our authentication mechanisms, only pre-approved specific computers can connect with a given PPCS server, thus locking out any “bad guy” who tries to access from any computer that has not been approved, even if he has a properly configured copy of The Property Manager on his system.

The file names on your server that contain your data will be unique to your system; no other copy of The Property Manager anywhere will have the same file names. Thus, a bad guy with a properly configured copy of The Property Manager who somehow manages to spoof your server in order to connect with it won't know what files to ask for - and The Property Manager server won't reveal those names to any enquiry. And even if he does somehow obtain the file names (which he could do if he managed to compromise your server system by some means) he still can't access the files unless he has the passwords, and the passwords are kept in encrypted files. Further, even if he compromises your server and obtains physical access to your data files, all of those files are encrypted and can't be read without the password - and the password to read those files is NOT the same password that is needed to access them from the internet.

Not only that, but our executables (on both the server and the client) are encrypted so even if the

bad guy gets a copy of the executable he can't disassemble it to obtain information such as passwords that might be buried in it. In fact, we know for a fact that "black hats" have tried exactly this and have failed.

All communications between the client and the server are encrypted, so no one will learn those file names or the authentication information by sniffing the server's network connection. In fact, there is exactly one practical path that we are aware of that could lead to cracking our system, and that requires the bad guy to have physical access to both the server machine and to one client machine. Even then, we consider it unlikely that he would succeed, and we would presume that the property management organization would take those steps necessary to prevent this from happening.

From the user standpoint, all of this is transparent. You don't even have to require your users to set up passwords to access The Property Manager on their workstations (though we very strongly encourage this); all of the security takes care of itself once it is set up originally and doesn't visibly impact your users in any way.

The overall consequence is that our PPCS mechanism provides you with an environment that is as secure as we can make it, when you are working across the internet. Our security is vastly superior to the security of any web based application, since our system was designed with security at the very heart of the design, while security on the web was a distinct afterthought that only came about as the demands on web browsers grew beyond the original intent.

Consideration of SQL Security and Advantages

SQL is widely deployed by very large organizations and security has been a very important issue. A properly secured SQL server installation is generally fairly secure. However, SQL servers have been very badly cracked in the past (remember the Code Red worm, which brought the entire internet to its knees?), and it is likely that this will happen again in the future.

The likelihood of this happening is high if for no other reason than that SQL is widely deployed by very large and high-profile organizations, lots of bad guys devote lots of time to cracking it, and it announces its presence by responding to TCP requests for connection. So long as you maintain your security patches and have not misconfigured the system, your SQL server will probably remain secure unless you happen to be one of the unlucky ones who are found and attacked on "day zero" of a new attack.

The only advantage of an SQL system over a Superbase system is that the SQL system scales very well; organizations that handle terabytes of data daily need a database system that can handle huge datasets. However, even the very largest of property management organizations don't come close to this need for data handling, so that advantage is moot for this case. Superbase in a PPCS environment performs well for hundreds of users, and handles datasets of billions of records - more than adequate for even the largest of property management organizations.

In the property management environment, the disadvantages of an SQL environment compared to

a Superbase PPCS environment vastly outweigh the one advantage; high cost of deployment, high cost of maintenance, greater security implications, and ultimately less capability for the user due to no local caching.

Conclusions

The development of the Web has without doubt profoundly changed the world. It is the “killer app” that made the internet accessible and useable by everyone.

The web browser makes it easy for a casual user to obtain massive quantities of information and entertainment. There is no denying the value of the web browser.

But no technology is perfect; no technology can or should be used everywhere for everything. In the case of the property management enterprise, a web browser and web server environment is seriously deficient for a number of different reasons, as set forth in this paper.

We have shown how the PPCS environment is vastly superior in this environment. It is superior because it can be thoroughly dedicated and completely locked down. In other words, it is superior in this environment because it expects and permits that which the web browser is specifically intended to avoid; the ability to lock down and secure all connections, and the ability to ship data-only back and forth, with highly capable programs running on both ends of the connection to process the data appropriately.

An SQL environment with a smart client can duplicate many of the capabilities of the PPCS environment, but only at much higher cost. Even for the higher cost, the SQL environment can't match the security of the PPCS environment, and won't be as fast because the capability for local caching won't be there.

The Property Manager, with its PPCS environment, permits you to have a dedicated virtual private network, with full capability and high security, and without a dedicated server operating system, in order to do your property management job. In terms of cost, reliability, security, and performance, no other system comes close.