I think a nice way to start this book is talk about in brief what components are required to build a Virtual Desktop Infrastructure and discuss the concept of VDI. In later chapters we may go to more detail around the components that make VDI but for now we will try to keep it as high level as possible. Following are the subjects covered in this chapter:

- What is Virtual Desktop Infrastructure
- Connection Brokers
- Why use VDI vs. the Alternatives
- Desktop Pools
- Provisioning
- Which Connection Broker
- Hypervisor
- Blades
- Thin Clients
- Remote Desktop Protocols
- Secure Channel
What is Virtual Desktop Infrastructure

Virtual Desktop Infrastructure aka VDI aka Desktop Virtualisation is a concept in which a solution based on a server based computing model that is not so different from the traditional terminal server centralised computing model used to deliver applications to remote users. It centres on the idea that companies can virtualise their desktop operating systems like Windows XP or Vista using platforms like VMware ESX or Xenserver and run said desktops from within the secured datacentre. So instead of maintaining full blown desktop PC computers on desks, the end-user would be issued a thin client device to establish a remote connection to the centralised desktop. Thin Client devices are mentioned in more detail later in this chapter. This is achievable by default because Windows XP and Windows Vista both have an integrated service that allow remote users to take remote control over the operating system over the LAN or WAN. This service is known as RDC Remote Desktop Connection and is facilitated using RDP the Remote Desktop Protocol. A very similar mechanism is used with RDP in a terminal server model. We will come back to RDP and other desktop protocols later in this chapter. Figure 2.1 demonstrates this idea:

![Diagram](image)

**Figure 2.1**

In this example the user makes use of a thin client device to communicate using an RDP tunnel with his/hers assigned virtualised desktop which is running safely in the secure datacentre traversing through a hardened firewall solution. The result is the user is able to run the applications installed on his/her virtual desktop and access data local to the datacentre.
Chapter 2
What is Virtual Desktop Infrastructure

Connection Brokers

To make this solution viable for use with Small Medium Businesses and larger Enterprises there needed to be a layer of management between the end-user and the virtualised desktop to establish and secure possible remote connections. Initially there wasn’t a solution available on the market and some companies took it upon themselves to use scripting to allow this model to work for them. Soon after a few pioneering software vendors released a similar solutions that would be the glue to hold this concept together, which we now refer to as the Connection Broker. The connection brokers we will focus on this book are as follows:

- Leostream Connection Broker
- VMware vView (aka VDM Virtual Desktop Manager)
- Citrix XenDesktop
- Provision Networks VAS

These connection brokers are discussed in more detail in separate chapters but for now we need to focus on what the role of the connection broker is. I mentioned before the connection broker is the glue to make VDI viable but what does that really mean? Well if we go back to figure 2.1 for the end-user to make an RDP connection to a virtualised desktop the thin client device would have to be programmed with the IP or FQDN (Fully Qualified Domain Name) of the virtual desktop to establish an RDP session. So in theory you could instruct your end-user with this information to make a manual connection without the intervention from any kind of management layer, but instantly you should be able to see the potential issues with this approach:

1. This approach is painful to manage in large environments. With this approach the system administrator would have to manage individual connections for potentially large numbers of users that means statically assigning user to virtual desktop
2. In addition to manually managing the assignment of virtual desktops without some form of management layer the system administrator would also have to manually manage the desktop protocol sessions, the provisioning of virtual desktops and the user experience which in large environments would be an administration nightmare.

To alleviate these management issues from a VDI solution a connection broker is employed and from reading this book you will learn that different connection brokers have similar goals but deliver a whole plethora of different features. So next why don’t we dip our toes in the water and try to understand what is ultimately required from a connection broker in today’s terms?

A connection broker is initially required to place a user on a virtual desktop. Which desktop? That’s another kettle of fish which we’ll discuss later. But for now we need to get our end-user on to a virtualised desktop. Most connection brokers will first authenticate a user to find out who the user is and what permissions that user has to remote control desktops. Using this information a policy can be utilised to position the user on the correct virtual desktop. Figure 2.2 demonstrates this idea.
In this scenario the user need not know information about the virtual desktop but only the connection broker. The system administrator only has to relay the IP or FQDN of the connection broker back to the user instead of maintaining a list of virtual desktop assigned to users. This information is generated using policies within the connection broker. So for example the user authenticates against the connection broker which processes those user credentials locally or against a form of directory services like Microsoft Active Directory depending on the connection broker. Information about the user is requested and policy can be used to position the user on a specific virtual desktop or from a pool of desktops (pools are discussed later) depending on Active Directory group membership. So the system administrator could program the connection broker to connect Joe from sales to a virtual desktop that is assigned to the sales team. Sounds simple doesn’t it? Well on the whole it is but there is whole load of design considerations we need to take into account and additionally each connection broker delivers different features and functionality.

Why use VDI vs. the alternatives?

So why use VDI? Before we answer that question let’s recap on some alternative computing models and why you as a company may decided VDI is a better alternative. One thing to note here is I do not want to go into too much detail about the alternative computing models as you can find numerous books on the market that cover the following concepts but it’s important to talk about in brief to relate them to VDI.

**Traditional Client/Server Computing Model**

For many years most companies that employ a traditional client/server model will take the following approach:

- Place a medium to expensive PC computer on the end-users desk
- Allow the user to access data across a network on server computers located in the a datacentre
- Secure this environment but only from the public perimeter
Chapter 2
What is Virtual Desktop Infrastructure

Now let’s break down the previous bullet points into more detail.

What most companies tend to do is provide each static user with a PC computer. Then the system administrator has to spend time and money provisioning which includes OS installation/Configuration and Application Installation/Configuration. These PC computers tend to fail frequently because of the moving parts included in the Hard Drive, CD/DVD drives and only have a life expectancy of about 2 years. This solution involves maintaining these devices and the software installed locally and often incurs trips out to the end-user by the support team. This is fine if the company maintaining this model has the staff to maintain such an infrastructure. The problems starts when organisations out grow the workload their internal support team is able to cope with.

Additionally for many years now, computers have been required to allow users to share files amongst themselves. The need for copying files from one PC to another using such devices as floppy disks has nearly been wiped out and superseded by the use of network file access protocols like SMB/CIFS or NFS which allows a user to access the contents of another computer using the network. Apart from enabling file sharing this had other benefits. Using network file sharing system administrator can now centralised mission critical data located on dedicated computers which we now refer to as file servers. The advantage of doing this is the system administrator doesn’t have to go far to protect the data from potential problems like deletion or corruption and back up the data centrally.

A lot of companies are still only securing data from the public perimeter although some companies see the benefits of securing data internally as well. So what this means is that some companies only employ a security mechanism to protect against intruders at the public perimeter (at the point where the public can use the internet to gain access to your company network) for example a firewall, while other organisations have highlighted the potential problem for intruders that already exist internally and would see a potential security floor in allowing users to access data from their local PC computers which could mean they could make hard copies to be removed physically offsite.

By now you could be asking the question: what does this all have to do with VDI as an alternative to traditional client/server computing? We’ll shortly discuss how using VDI by default could be a better way of computing to help combat some of the inherent problems highlighted above because of the nature of the solution and how it fits together. For now we’ll focus on another computing model very similar in function to VDI which is we will be referring to as the terminal server based computing model.

Terminal Server based Computing Model

In a terminal server based computing model multiple users would remote connect to a desktop session running from a single centralised computer known as a terminal server. All applications would run from this single server and share the same operating system, registry, memory and processor resources. Users would initiate a remote desktop session using desktop protocols like RDP in a similar way as discussed previously with the VDI solution. The difference is the terminal server will be configured with windows component known as terminal services. Figure 2.3 demonstrates this concept.
Figure 2.3

Terminal Services is a Microsoft Windows server component originally developed by Citrix and licensed to Microsoft. It was first introduced in Windows NT 4.0 Terminal Server Edition. In addition Terminal Services has been improved in each subsequent versions of Windows Server operating system 2000, 2003 and 2008 since NT 4.0 Terminal Server Edition.

The advantages of using such a solution to an organisation is very much the same as the VDI concept and are highlighted as follows:

- Centralised shared data
- Datacentre containment of data
- Centralised managed application
- Remote Access to internal systems and data

With this solution system administrators have the ability contain data with the datacentre. This is achieved because the applications used to manipulate said data also runs within the datacentre and remote users connect to the applications using a remote session. So to simplify the applications run on the terminal server while the graphics displayed by the application is rendered on the remote connection facilitated by the end-point device usually a Thin Client terminal. The data never reaches the end-point device unless allowed by the administrator.
Chapter 2
What is Virtual Desktop Infrastructure

Also the applications are maintained centrally on the terminal server and would mean the system administrator only have to maintain single instances of the applications. This means life is made easy for the administrator by not having to maintain multiple instances of applications on the end-point devices which could be out somewhere on the public internet or branch office.

So next let’s highlight what are that advantages of VDI over the two methods you have just read about and we should be starting to build a picture of what VDI does actually offer us. First of all we should look at how VDI could be a better alternative to using the Client/Server model.

VDI advantages over Client/Server model

- The use of inexpensive thin client devices
- Centralised management of applications
- Secured data containment
- Easier to support
- Desktops now run on resilient hardware

With a VDI solution we can make use of thin client devices which do not use components with moving parts, tend to have a life expectancy of 7 years and are much cheaper then desktop computers. Applications are maintain in the data centre and not at the end-point devices which makes life easier for the system administrator for not having to travel very far to support end-user applications.

As mentioned previously: because the data is contained in the datacentre we can ensure sensitive data is not removed of site. Something we’ll discuss later is the idea that we use policies within connection brokers to dynamically create new desktops for use instantly which could negate the need for an administrator to spend many ours refreshing remote desktops because of software corruption and miss use. The user could a have fresh, pristine new desktop every time they login without the administrator lifting his/her finger.

In addition due to the fact that these virtualised desktops now run in the datacentre on top of a virtualisation platform there’s a good chance that the desktop infrastructure is now by default running on much more resilient hardware then they would have normally while running on a standard desktop PC. Virtualisation platforms like VMware ESX or Citrix Xenserver (also referred to as hypervisors) require the use of server class hardware. Server class hardware is often protected against failure or catastrophe be employing redundant hardware. So for example servers would normally use 2 power supplies in case one failed the other is waiting on standby or 2 network cards or 2 storage controllers and multiple redundant disks.
Now let’s look at how VDI can be advantages over a terminal server based computing model. This one is little bit more difficult to realise as these 2 solutions are very similar in concept and in a lot of instances you may argue that terminal server based computing can be more cost effective than a VDI solution. You know what if the glove fits i.e. terminal server based computing does what you need it to do then I cannot see many disadvantages to this approach.

**VDI advantages over terminal server based computing**

- No application compatibility issues
- Desktop session is a desktop operating
- Potential Performance control
- Easier to maintain against miss use
- Provisioning

Environments like terminal server are sometimes prone to situations where applications will not reside together on the same instance of an operating system which they would have to traditionally while using terminal server. For example you have one application that requires areas of the operating system that is already occupied by another application. Companies like Citrix have management solutions to combat this issue like *application isolation environment* but incurs an overhead on performance just to facilitate this feature. Some applications are not multiuser operating system aware i.e. they just don’t like being installed on a system that is used by multiple users at the same time.

We have a saying in the VDI community if it works on a desktop it works on a virtual desktop. The reason for this, the desktop operating system used in a VDI virtualised desktop is the same operating system traditionally used on a desktop computer.

In a virtual environment by the nature of what a virtual machine is, we have more control over the compute power and allow some virtualised desktops to perform much better than others especially when using VMware ESX. VMware ESX has the ability to redirect compute power to virtual machine where needed dynamically at many layer of the virtual infrastructure. This means that we can have power users opposed to light users on the same system. In a terminal server solution the compute power is shared between all users residing on that server and one rogue user could potentially bring performance to a grinding halt for everyone else sharing the terminal server.

In a terminal server environment system administrators may find it more of a challenge to refresh the terminal server after software corruption or miss use. For example the terminal server is infected with malware the administrator would have to take multiple uses offline, sometimes hundreds to clear the problem. With VDI as mentioned earlier a user to can always logon to a fresh and pristine desktop so it wouldn’t matter if the single instance was compromised. We can provide a pristine desktop each time the user logs on usually because the connection broker will have a feature to allow for rapid provisioning of new desktops.
Chapter 2
What is Virtual Desktop Infrastructure

Desktop Pools

In traditional client/server computing there is a good chance that you as an administrator maintain similar images for your desktop estate. If you maintain large numbers of desktops it could be advantages to develop a standard image for the operating system of a desktop and may be have groups of images where you need to deviate. Also there is a good chance that you allow hot-desking in which multiple users may use the same desktop. You do the above to minimise the effort needed to build desktops and also saving money on software licensing as multiple users will use the same licensed software.

Well guess what? Nothing has to change when utilising a VDI environment. In fact most of the connection brokers will aid in achieving the above tasks and extend what is achievable when managing mass amounts of desktops. Probably the most important function that is expected from a connection broker is the ability to pool desktops together that hold a common purpose. The different vendors may use different terminology when it comes to desktop pools:

- On-Demand Virtual Desktops
- Desktop Pooling
- Flexible Provisioning

But they really refer to the same concept that through a enforcing policies when a user logs in, the connection broker governs what desktop the user is presented from a pool of desktops. This could be seen to provide for potentially three different scenarios:

1. A one to one mapping is established between the user and a static virtual desktop
2. A mapping is established from the user to a free virtual desktop that is picked at random from a pool of identical virtual desktops.
3. The user is given a choice of multiple desktops.

So an example of using a desktop pool is as follows:

An administrator knows that the sales team all have identical needs when it comes to applications, resources and is aware that only 50% of the sales team need access to the internal system concurrently. Using desktop pooling the administrator could automate the creation and the pooling of virtual desktops to maintain a number of desktops in the sales pool which equates to approximately 50% of the sales team. So when John from the sales team logs in he is presented with a virtual desktop with the security and applications required by the sales team. He logs out and then Sue logs in and is present the same virtual desktop. An Active Directory Group Policy is used to provide Sue with a different profile than John giving the illusion the Sue is on a different machine. Active Directory Group Policies are discussed in more detail in a later chapter.
The advantages of pooling desktops in this manor are as follows:

- Consolidate the amount of desktop operating system licenses required by 50%
- Reduce the amount of application licenses required by 50%
- Life is made easier due to only having to maintain one desktop image for the sales team

Just for now that’s all you need to understand about desktop pooling as I don’t want to give the game away too much before we dive into each connection broker in later chapters.

---

**Provisioning Desktops**

In this section we discuss how we can make use of different features and technology to aid us in the provisioning of desktops and also how to optimise disk space required by VDI desktops.

**Rapid Provisioning**

Virtualisation platforms like VMware VI4 allow for the rapid provisioning of virtual desktops. Traditionally desktops could take hours to provision. The reason for this is the steps that are required to provision a new physical desktop:

1. Order the desktop from supplier
2. Unpack and test the desktop
3. Configure the Operating system and applications

Because VMware have included features that automate the creation of virtual machines and are able to make said virtual machines unique. Connection broker vendors have made use of this feature to allow virtual desktops to be provisioned on the fly. So this means when a user logs in and uses up the last virtual desktop within a pool, some connection brokers have the ability with a little help from VMware to create an additional virtual desktop to add to the pool dynamically. This is achieved in minutes and not hours through imaging.

To make a virtual desktop unique which has been provisioned from an image VMware decided not to reinvent the wheel and made use of a freely available tool which comes with most Microsoft operating systems called sysprep.

**Sysprep**

Sysprep is an application that is used to remove anything about an operating system that makes it unique for repackaging. Things like the computer name, network configuration and serial number can all be removed using sysprep.
Chapter 2
What is Virtual Desktop Infrastructure

The clever bit is VMware have scripted and automated the use of sysprep within their management console VMware Virtual Centre as normally using sysprep would be a fairly a manual process.

This allows the creation of virtual machines to very dynamic as the virtual machines identity is given to it automatically at creation. Its computer name, IP address, its domain membership, passwords etc are all injected on the fly.

As of writing this section of this chapter VMware are currently way ahead with this kind of functionality and definitely more feature rich than their closest competitor Citrix in this space. But Citrix do have an ace up their sleeve which we’ll discuss next.

On-Demand Image Provisioning
Citrix not so long ago acquired a technology referred to as Citrix Provisioning Server from a company called Ardence. Provisioning Server is quite cool as it allows multiple Virtual Desktops to PXE boot an operating system across a network from a single image of the operating system.

**PXE aka Preboot Execution** Environment is a technology that allows computers to boot their operating systems stored on a network resource opposed to a disk subsystem like the local hard drive

So the first question that might come to mind “Won’t that put tremendous load on the network?” Well this is a possibility as you could imagine lots of virtual desktops all trying to boot from the same image across a network....But if you way up the advantages of using this technology and follow some rules then this may be a viable option.

First of all let’s take the alternative namely booting each virtual desktop from a separate virtual disk containing the operating system. It does not take a rocket scientist to work out the amount of disk storage needed for this solution. Let me give you an example:

If you have 1000 virtual desktops all with an average of 4 Gigabyte disk space used to contain the virtual desktops operating system. 4 Terabytes worth of disk storage is required. You need shared storage i.e. a SAN storage array for this solution and 4 Terabytes worth of SAN storage doesn’t come cheap. With Citrix Provisioning Server potentially all 1000 virtual desktops could boot from the same image and that image would only require 4 Gigabyte of storage. But then again you may prefer to through money at your solution to ensure performance isn’t an issue.
**Linked Clones**

In the next incantation of VMware’s own connection broker VDM, VMware will port over a concept from a different VMware product (VMware Workstation) referred to as linked clones. The idea behind linked clones is that VDI desktops will all point to a single disk image located on shared storage to boot from (similar type of concept as Citrix Provisioning Server) and maintain a delta file where disk writes created by individual virtual desktops are stored. The key difference between this idea and the Citrix concept is that the single instance of the disk image is boot from while located on disk storage and not a network resource but the advantage is the same: major cost savings on the amount of disk space required opposed to using a single virtual disk per virtual desktop.

**De-Duplication**

Storage is always on the mind of the system administrator and even more so in a VDI implementation. Help maybe at hand. Some storage vendors maintain a feature referred to as Data deduplication. Dedupe as it's also known offers major savings in disk usage. The concept behind this technology is simple to explain:

When data is stored on a disk subsystem, it’s divided up in to chunks of data known as blocks. If a storage system uses a 64k block size file/data is broken up into 64k blocks of data. This means the file system will read & write 64k worth of data to/from the disk at one time. Figure 2.4 demonstrates that a 100k file resides on disk and therefore consumes 2 blocks of data.

![Figure 2.4](image)

Note – in the above example file.doc consumes 2 blocks of disk storage. File.doc is 100k in size which means 28k of the second block is waste as the next file will start on disk at the beginning of the next block. This may seem like being wasteful but has benefits when it comes to performance. Especially for applications that have been optimised to read & write in block format.

But how does help us? Well Storage vendors have a great appreciation for what disks are being used for and they know there is a good chance that on a disk subsystem there will be many instances where a block of data is replicated throughout the disk subsystem. An example of why this may happen is because of the nature of some operating systems. If you have 10 windows 2003 operating
system located on the same disk subsystem then there is a very good chance that area of the disk contains the same data blocks. Out of the thousands of files that make up the operating system there will be large amounts of block area that is the same.

What Dedupe does is uses an algorithm to work out what blocks are duplicated and instead of writing the same block multiple times on disk the technology uses redirection and pointers to divert operating systems to a single instance of the block. Figure 2.5 demonstrates this concept.

![Diagram of Dedupe concept](image)

**Figure 2.5**

So from this example you can see that there has been a saving in storage as we have saved 2 blocks of disk space when ordinarily we needed 3.

There is an overhead with this technology though. First of all the storage array spends a lot of compute power monitoring for duplication. Also when a block is changed by the owning component i.e. the operating system or an application within, the storage array has to perform a copy on write. This means a copy of the single instance block is duplicated to a different part of the disk subsystem for use by the component that changed it. This would change our example as demonstrated in figure 2.6:
Dedupe is traditionally used with disk based backup solutions. It reduces the amount of disk space required for copious amounts of data in which the overhead used with this concept is less important. Dedupe is also raising its head within the VDI community for the same reasons. Will there still be a need for it when VMware introduce linked clones maybe not but storage vendors may argue that the concept of linked clones also introduces its own overhead.

This feature is sometimes also referred to as single instance storage and for those of you familiar with Microsoft Exchange you could also compare this to single instance emails.

To control the size of the mailbox stores, Exchange supports single instance message storage. This means that when a email is sent to more than one mailbox in the same store, only one instance of that email is stored, in one mailbox. The other mailboxes contain pointers to the stored message.
Chapter 2
What is Virtual Desktop Infrastructure

Thin-Provisioning

Staying with the theme of optimising storage it is also worth mentioning a feature of some storage technologies like Datacore, Lefthand, Compellent and 3Par referred to as Thin-Provisioning aka Auto-Provisioning aka Dynamic Capacity. Thin-Provisioned storage is where the actually size of a LUN/Volume of storage that is presented to a node device e.g. a server is much less than what is visible to the server. So for example a server is presented with what it thinks is a 100GB volume of storage but it reality the true physical size of the volume is 20GB. At the point where the server reaches the 20GB mark in usage the storage array is intelligent enough to allow the volume to grow in small chunks until it reaches the 100GB limit. This way we only provision what we use.

Is it worth doing? Well if you consider the fact that most storage is over provisioned due to the fact when system administrator request storage they usually over provision storage to a. Future proof the requirement and b. To make sure they haven’t under provisioned the storage. So for those reason a lot of storage tends to be wasted up to 80% in some scenarios. Figure 2.7 demonstrates visually the difference between not using this feature and optimising storage usage. As you can see with a Thin-Provisioned volume the used area is also unallocated for use elsewhere:

![Traditional Storage Volume vs Thin-Provisioned Volume](image)

**Figure 2.7**

Of course you could get to a point where you have provisioned more storage than you actually have but most storage vendors that provide this feature also allow you to set threshold of when to notify the storage administrator that storage is low. At which point the storage administrator has a choice of whether he/she needs to acquire more storage or consolidate the existing storage.
Which Connection Broker is best for you

Which connection broker is best for you? Well that’s a tricky question to answer at this stage of the book. Why it’s tricky to answer is because the best connection broker for you is heavily dependent on the requirements of solution being implemented. All the connection brokers listed in this book have the same purpose in life which is to facilitate a remote connection session from and end-point device to a VDI virtual machine. How they achieve this goal differs from connection broker to broker. Also things like having a requirement to access this infrastructure across the internet or slow WAN links could ultimately be the deciding factor on which choice of connection broker. Or the need to automate dynamically this environment could sway the decision in a different direction. Or needing multimedia capabilities could leave standing one clear winner. You get the idea. I’m going to be as agnostic as possible because in reality there’s no clear winner each connection broker displays a plethora of features and its going to be one of those features that make your mind for you.

In later chapters of this book we will go into each connection broker in more detail so you will gain an understanding of the difference between the features. But for now we still need to focus on some of the other components that will or may be required for a VDI solution. The next two sections will cover a choice of another fundamental component required in a VDI implementation. A VDI session can reside on different platforms and so far we’ve discussed the idea the sessions would be facilitated using virtual machines but we could also use Physical machine to do the same thing. The desktop operating system used in a VDI virtual desktop is the same operating system used on a physical desktop. If a physical solution is required then the traditional platform used is normally Blade Systems. So the following sections will relate in brief the technology that used to facilitate the virtual infrastructure and Blade systems used with a physical scenario, the Hypervisor.

Hypervisor

We are not going to go into too much detail about virtualisation hypervisors as the there is a level of expectation that you as the reader of this book should be familiar with virtualisation. In addition there is a plethora of books and courses available to teach you about virtualisation.

So what is a hypervisor?

Virtualisation vendors like VMware and Citrix develop software that allow multiple operating systems like Microsoft Windows and Linux to be installed in self contained environments and shared compute resources on the same PC computer. This so called “software” is referred to with different labels: Middleware, Virtualisation layer and even Hypervisor. Traditionally operating systems would install directly on top of the PC architecture and you could even say that the operating system was the middleware between the hardware and the applications that reside on the above layer. Figure 2.8 demonstrates visually this idea:
Chapter 2
What is Virtual Desktop Infrastructure

Figure 2.8

With the introduction of virtualisation the picture looks slightly different:

Figure 2.9

The hypervisors that may be referenced in this book will be Citrix XenServer and VMware ESX. These are the two mainstream hypervisors of choice and the two vendors are in a race head to head to the finish line. While writing this book I’m trying to be as agnostic as possible but I have my favourite.

😊
With this knowledge you can easily see why virtualisation platforms like VMware are ideal for VDI. Maintaining numerous virtual machines on a single PC platform using virtualisation has many benefits that you expect to see like consolidation, rapid provisioning, business continuity and etc. In some scenarios virtual machine may not be ideal. There may be some instances where a solution requires that the VDI session be established on a high performance alternative to virtual machines. The mainstream alternative to using virtual machines with VDI is by deploying a Blade server which we cover next.

Blade Servers

Using a Blade server (aka high-density server) is a physical alternative to using virtual machines for VDI sessions. The idea is each VDI user is positioned on a single physical blade server. The mechanism for the user making the connection from an end-point device is no different from the VDI solution mentioned early. User authenticates through a connection broker – connection broker creates a remote session to VDI desktop which happens to be a physical Blade server. Figure 2.10 demonstrates this idea:

![Blade Chassis](image)

Before we talk further about the advantages of using Blades let’s go through what this technology is and how it works. The idea behind blade systems is you take a large server box known as a blade chassis that is capable of housing 10 encased server motherboards known as blades which plug into a backplane of the blade chassis. Following is a front view of an IBM blade chassis:
Chapter 2
What is Virtual Desktop Infrastructure

Figure 2.11

Each blade typically has two CPU sockets which could contain single, dual or quad core processors. Each blade would typically be capable of utilising at least 12 gigabytes or RAM memory and each blade could have one or two local harddrives.

The advantages of using such a solution is even though each server maintains separate resources like disk, CPU and memory they also share a fault tolerant power supply which is fed into the backplane of the blade chassis reducing the amount of electricity used if you were to have 10 conventional servers. They also share resources like network IO, fibre channel IO, CDrom and floppy disk drive.

Typically each blade would not have a port to plug in a keyboard, mouse nor monitor but access to the console of each blade server is facilitated using a lights out management interface.

*Lights out management interface* – this is very usual component included with most servers and is isolated from the operating system running on the server. What this component allows for is the full remote control of the server over a network connection. You are able to take control over the keyboard, mouse, video, cdrom, floppy disk and even control the power button of each server. This means administrator is able to do things like install operating systems on servers without standing in front of the server in a noisy datacentre.

Staying with the theme of consolidation and reducing the total cost of ownership the advantage of using blades in a VDI solution is take make use of server performance. In scenarios where some VDI users are classed as power users and need a high performing VDI session blade VDI solutions allow a user to have access to isolated resources almost like they were positioned on a physical desktop but with the added advantage of the VDI management layer.
You could argue that using virtual machines in a VMware ESX environment produces the same results. The control you have as an administrator over the resources allows you to isolate resources for use with individual virtual machines and if a VDI session calls for having 2 processor cores and 12 gigabytes of RAM memory is possible. When using Blades you get a slice of both worlds: The centralisation of your desktops with the performance of a physical equivalent.

**Thin Clients**

This next section we will discuss Thin Clients the hardware end-point devices that are typically used instead of desktop PC when using a VDI solution. Thin Client devices have really been ported from the server based computing model where they have seen a lot of usage especially with terminal server and Citrix.

A thin client is very similar to desktop PC in the sense that you plug in peripherals devices like keyboard, mouse and monitor. In addition it contains an operating system and you are able to access network resources. The big difference between the two is thin client devices do not use components with moving parts i.e. hard-drives, the operating system is a cut down light operating system and integrates just a handful of applications. The focus of these integrated applications is to provide remote access to servers or even access to the internet and that’s a about it. These devices don’t even have a CPU fan because the CPU needed for such medial work doesn’t produce a lot of heat. In this environment applications that require processing power would not schedule CPU workload on the CPU within the thin client but rather is processed on the central server whether it is a terminal server or mainframe. The operating system maintained on these devices is so cut down that a hard-drive is not needed and the operating system is installed on solid state memory like flash memory.

The thin client devices allow for some configuration, they maintain an IP stack which allows the devices to communicate on the network and allow for minor configuration changes like specifying DNS servers, FTP Servers etc etc. In addition you do things like specify screen size, audio settings etc etc.

In addition the parameters listed these devices also have integrated in them the software code to enable these devices to connect to Citrix XenAPP, Citrix XenDesktop, VMware VDM, Leostream Connection Broker and Provision Networks.

Also unrelated but some thin client devices also contain applications to allow you to connect to mainframe or UNIX environments using terminal emulation like VT100 etc.
Chapter 2
What is Virtual Desktop Infrastructure

As for external devices you’d expect to connect the follow devices:

- Keyboard
- Mouse
- Monitor
- Network
- Audio
- Printer
- PDA
- Scanner

Whether these devices work within the remote session is dependent on the technology that facilitates the session. For example Citrix supports PDAs, Bi-directional audio and scanners where some other environments might not.

There is a plethora of thin client devices on the market and the mainstream vendors are:

- Wyse
- ChipPC
- Neoware
- HP

These thin client vendors maintain 4 different types of operating:

- Windows XP Embedded
- Linux
- ThinOS (Wyse Only)
- Win CE

*Windows XP Embedded* is very cut version of Microsoft windows XP. Cut down just enough to do what you need it to, i.e. run a RDP client to connect to your server or virtual desktop. The advantage of using XP embedded is at the end of the day it’s still XP and it’s possible to integrate third-party applications or drivers. So opposed to using ThinOS and Win CE with a XP embedded the administrator is able to modify the device work and fit in within specific environment. *Win CE* and ThinOS are pretty much set in stone and do not have the capability to install additional drivers etc unless its flashed within the firmware of the device.

Using *Linux* as alternative carries similar advantages. The thin client vendor may prefer to use Linux over XP embedded as it’s at zero cost. Where the vendor may incur a license fee for XP embedded.
I didn’t want to make this book into an advertisement for thin client devices but it’s worth mentioning a few. Wyse is probably the market leader due to their feature range. The two most popular models are the S90 and the V90. Figure 2.11 is a picture of a S90 and Figure 2.12 is a picture of a V90.
Chapter 2
What is Virtual Desktop Infrastructure

*ChipPC* is another vendor which is worth mentioning. ChipPC have some innovative designs some may say gimmicky but never the else they are pretty cool. First of all they have a range of the smallest thin client which is very appealing as part of the idea of using a thin client device is that they are small enough to be out of the way. Figure 2.13 is picture of a ChipPC Xtreme PC:

![Figure 2.13](image)

**Figure 2.13**

The Xtreme PC has all the features you’d expect to see from a thin client device but is small enough to fit in your pocket. For those security paranoid ChipPC have another device that’s almost impossible to steal as its fixed to the wall in a standard network port junction box. This model is the JackPC and is demonstrated in figure 2.14:

![Figure 2.14](image)

**Figure 2.14**

Cool hey! Well you guessed it; this device has all the features you’d expect from a thin client
Thin Client Tip

Not all thin clients support VDI. This is not to say that the model you have cannot be updated. The first thing to do is check with your manufacturer to see if the firmware that’s currently in your unit supports the specific connection broker you are planning to use. If not find out if there is an updated version which will support the connection broker and follow the manufacturer’s instructions to flash the new firmware to make it compatible. With XP embedded and Linux thin client devices this is not so much a problem as the idea with these to platforms is can either integrate the VDI client yourself or use a web browser equivalent of the client.

Following is a table of Wyse thin Client devices that support VDI:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Linux</th>
<th>Xpe</th>
<th>CE</th>
<th>ThinOS</th>
<th>VDI Ready</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wyse S10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wsys S30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wsys S50</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wsys S90</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wsys V10L</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Wsys V30L/V30LE</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Wsys V50L/V50LE</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Wsys V90L/V90LE</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Wsys X90/X90e/X90L/X90Le</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.15

Following is a table of ChipPC thin Client devices that support VDI:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Linux</th>
<th>Xpe</th>
<th>CE</th>
<th>ThinOS</th>
<th>VDI Ready</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xtreme PC NG-6600</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtreme PC NG-6010</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtreme PC NG-6050</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtreme PC NG-6052</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtreme PC NG-6400</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtreme PC NG-6450</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtreme PC NG-6452</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtreme PC NG-6500</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xtreme PC NG-6552</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jack PC EFI-6700</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jack PC EFI-6800</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jack PC EFI-6900</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.16
Chapter 2
What is Virtual Desktop Infrastructure

Following is a table of iGel thin Client devices that support VDI:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Linux</th>
<th>Xpe</th>
<th>CE</th>
<th>ThinOS</th>
<th>VDI Ready</th>
</tr>
</thead>
<tbody>
<tr>
<td>Igel 3210 LX</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 3210 CE</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 3210 XP</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 2110 LX</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 2110 CE</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 2110 XP</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 4210 LX</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 4210 CE</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 4610 XP</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 5210 LX</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 5310 LX</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Igel 5610 XP</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Figure 2.17

We just don’t have enough pages in this book to go through every thin client so we should stop there. What I would like you to have an appreciation is what to do with the client device to help authenticate against a connection broker. In this example we will just focus in on the Wyse ThinOS devices:

Power on the device and wait until the devices goes through its boot up procedure and can go no further. Then Click on Desktop > System Setup > Network:

Figure 2.18
Next the Network settings dialog box appears. Click on the Servers tab:

![Network Setup Dialog Box](image)

**Figure 2.19**

You’ll notice that there is a field to input a VDI broker address. Dependant on which connection broker you are planning to use depends what you should input into this field. For example if you were planning to use a Leostream Connection Broker then you could get way with inputting the IP address, hostname or Fully Qualified Domain Name here which you could not do if using VMware VDM. VMware VDM requires you to input the URL of the VDM connection broker i.e. `https://www.vdmConnectionServer.domain`
Chapter 2
What is Virtual Desktop Infrastructure

Remote Desktop Protocols

The purpose of the thin client device in a VDI solution is to facilitate a VDI Client and a remote desktop protocol used to access the VDI session. The 3 most used protocols for VDI implementations are as follows:

- RDP
- ICA
- VNC

**RDP (Remote Desktop Protocol)** is based on, and an extension of, the ITU T.120 family of protocols and as mentioned previously RDP is integrated be default into most windows operating system s. This makes it easy for the connection broker vendors to utilise this existing protocol to create a remote desktop session from the user’s end-point device to the datacentre. *Why reinvent the wheel?*

**RDP** uses multiple virtual channels (up to 64,000) to separate out communications between devices and presentation data. These virtual channels facilitate the following features:

- 32-bit color support. 8-, 15-, 16-, and 24-bit color are also supported.
- 128-bit encryption, using the RC4 encryption.
- Audio Redirection allows users to run an audio program on the remote desktop and have the sound redirected to their local computer.
- File System Redirection allows users to use their local files on a remote desktop within the terminal session.
- Printer Redirection allows users to use their local printer within the terminal session as they would with a locally or network shared printer.
- Port Redirection allows applications running within the terminal session to access local serial and parallel ports directly.
- The clipboard can be shared between the remote computer and the local computer.

The following features were introduced with the release of RDP 6.0 in 2006:

- Remote Programs: Application publishing with client-side file type associations.
- Seamless Windows: Remote applications can run on a client machine that is served by a Remote Desktop connection.
- Terminal Server Gateway: Enables the ability to use a front-end IIS server to accept connections (over port 443) for back-end Terminal Services servers via an https connections.
- Support for remoting of Windows Presentation Foundation applications.
- Rewrite of device redirection to be more general-purpose, allowing a greater variety of devices to be accessed.
- All of Terminal Services will be fully configurable and scriptable via Windows Management Instrumentation.
- Improved bandwidth tuning for RDP clients.
- Support for Transport Layer Security (TLS) 1.0 on both server and client ends (set as default).
- Multiple monitor support. Spread session across two monitors.
RDP will render all display output server side then chop that data into network packets which is sent down the presentation data channel as figure 2.20 demonstrates:

![Diagram of RDP Tunnel with Virtual Channels and Presentation Data Channel]

**Figure 2.20**

Dividing up communication in this manner provides a way of allowing a quality of service or QOS on a per service basis. This means if we want better quality graphics onscreen opposed to the time it takes to print a document we have the capability to do so by restricting the print channel or even to go as far to shut it down therefore freeing up bandwidth for presentation data.

**ICA (Independent Computing Architecture)** is a Citrix trademark and is only available with Citrix products like Citrix XenApp and Citrix XenDesktop. It is also seen as the most optimised protocol due to its maturity and tends to outperform the others. ICA works in pretty much the same way as RDP in the sense that it creates virtual channels but uses techniques like better server side compression to provide high performance.

With ICA you get a plethora of features to many to bullet point as we did with RDP so if you are interested Citrix maintains a Client feature list and as of writing this book you could find this list at

**VNC (Virtual Network Computing)** is common throughout the Linux community and there is no reason that we cannot use Linux operating systems as VDI sessions. Some companies will prefer to do that due to cost savings. Linux falls under the GNU agreement and cost nothing to run. Supporting your flavour of Linux is a different story. But for now know that most distributions of Linux have a VNC server included so you can use a VNC client to remote access those sessions.

VNC is based on the RFB protocol, was developed by Olivetti and the original code was open source under the GNU agreement.

Its simple protocol that transmits the keyboard and mouse events from one computer to another, transmitting the rendered screen updates back to the end-point device over the network. Later versions now allow you to do things like transfer files from the end-point to the remote session. The cool thing about VNC is that its platform independent and will also run on Microsoft operating systems.

---

**Secure Channel**

Maybe we should have dedicated a whole chapter to securing your VDI solution for use from the public perimeter, but how you secure this environment will depend on VDI solution you go for. Explanation of the different solutions is described in more detail within each respective connection broker chapter. But for now let’s highlight in brief the options for securing each connection broker.

**VMware VDM** has a built in SSL relay and would not need anything else to secure the communication channel between the end-point and the datacentre and the RDP session would need to traverse through a variant of the connection broker server known as the security server.

**Provision Networks VAS** also maintain as SSL component for securing the communication channel between the end-point and the datacentre.

**Citrix XenDesktop** will make use of an existing Citrix hardware appliance known as the Citrix Access Gateway. The CAG as its also known is a SSL VPN device. Works like a VPN but performs better than a standard VPN device as it uses SSL public/private key encryption.

**Leostream Connection Broker** – Leostream have always stood by their belief that they should not put effort into reinventing the wheel and believe that securing VDI session should be left to the vendors that specialise in this area. So instead what Leostream have done has demonstrated how to integrate with popular SSL VPN and VPN units like Cisco and Juniper.
Chapter Summary

This chapter gave you taste of what will come in more detail later. Some aspects that we discussed are out of scope of this book and you should look to other material for a more detailed view. That being that we still had to discuss all components that make a VDI solution, if not some aspects of this book would not have made sense. What this chapter did was highlight the said components at a high-level.

**What is Virtual Desktop Infrastructure** - So we covered at a high-level what is the concept around VDI as a solution. We compared to the existing industry server based computing model and demonstrated how it differs.

**Connection Brokers** – We have highlighted the connection brokers that are covered in this book but didn’t go into too much detail as detail discussion is saved for later chapters

**Why use VDI vs. the Alternatives** – We discussed how VDI compares to more traditional ways of computing and why might VDI be a better way of life. We now know that VDI in addition to having centralised desktops, VDI is advantageous because we get:

- The use of inexpensive thin client devices
- Centralised management of applications
- Secured data containment
- Easier to support
- Desktops now run on resilient hardware

**Desktop Pools** – Is a way of saving money in respect to the cost of operating systems and software. When you pool desktops you no longer need an individual copy of an operating system and applications per user now we only need an individual copy of an operating system and applications per concurrent users.

**Provisioning** - We discussed ways to save time by using automated ways of creating desktops on the fly. We also talked about the methods we could save on disk storage when it came to homing lots of virtual machines used with VDI.

**Which Connection Broker** - This is a question which cannot be answer yet – you will have to make your own mind up after completing this book. A specific feature from a connection broker may ultimately make the call. So for example if you know that you will need access to this environment over the internet then you may decide that XenDesktop is the way forward.

**Hypervisor** – In this section we just gave you a brief overview of virtualisation platforms. Before you even think about VDI it might be an idea that you go find out about virtualisation first.

**Blades** - In this section we discussed the concept of using a physical alternative with VDI instead of virtual machines and the advantages of doing so.
Chapter 2
What is Virtual Desktop Infrastructure

**Thin Clients** – Thin Client devices will play a big part in the VDI industry and in this section we covered some of the options.

**Remote Desktop Protocols** - These are one of the key components of the VDI architecture. No remote session protocols No VDI. What we did in this sections is gave you an overview of what is possible with these technologies.

**Secure Channel** – In this section we highlighted in brief the need for securing this environment but diverted attention as a more detailed discussion will be presented in the connection broker’s respective chapter.