Chapter 5

Business in the Cloud

Advances in computer technology and the Internet have changed the way America works, learns, and communicates. The Internet has become an integral part of America's economic, political, and social life.

– President Bill Clinton

In This Chapter

Technology is fine, but its deployment (or not) is a business decision that must be made using the same sort of hard-headed business criteria as are applied to other business issues. In this chapter we'll learn about some of the criteria that come into play, strategies that companies apply in deploying cloud-based applications, and what a cloud application can mean for your organization. We'll discuss:

- Can you even use a cloud?—We've talked a bit about regulatory issues, but what are the other issues, and is this really the next step?
- Do you have enough Internet feed into your organization to use clouds instead of local infrastructure?—Moving to cloud desktops might sound great, but you don't get something for nothing. We'll look at where the costs might potentially shift.

- Load balancing—What is it? How is it going to help us? How does it work with clouds?
- Global load balancing and auto provisioning—How can you apply global load balancing to use clouds for on-demand capacity?
- Computing on demand—Do you really have to upgrade your computing infrastructure for that special project, only to let it rot after that project is done? Why not use the cloud for special projects instead of building more asset liability?
- Clouds as the DMZ for partnerships—Why are clouds becoming the neutral territory for a growing number of businesses? Why did the authors decide against setting up a server to host our writing efforts?
- Federation—Are clouds going to be the key technology that finally makes federated computing a reality? Why does it make sense, and are we already starting to see the beginnings?

Business Concerns About IT

Let's begin with a quick review of the basic concerns of business about IT. It's all about return on investment (ROI) and the black hole that is a data center as a huge corporate investment. The care and feeding of a modern data center is a nontrivial affair with a decision-making process akin to dancing a polka through a minefield. While the business concern is about ROI, the biggest fights tend to be over control: who gets it and who wants it.

The data centers and switch closets of companies are filled with departmental servers that are there just because a couple of personalities argued about things such as remote access, operating system support (or lack thereof), who has root access, who can add/edit/delete users, and so on. It's often just easier to buy an additional server than to fight these battles up and down through the organization. For those who do decide to battle it out, it can feel like fight night at every budget meeting; meanwhile, those servers suck up power, add heat to the office, add noise to the office, and prevent facilities from being able to shut down the office on holidays.

On the flip side, new environments lead to new IT training and personnel costs, and with shrinking budgets, saying "No!" has become a fashionable knee-jerk reaction. So, while on-demand clouds or clouds in general might sound like a magic solution, business decision processes demand that we know just where the hidden costs lie.

That's the environment in which cloud computing is being considered and in which decisions are being made. Is the cloud decision just about numbers, or are there issues to be considered that are more difficult to quantify? What kinds of numbers are you going to need to consider making cloud decisions?

Can Your Business Cloud?

The first question is the most basic: Can you use a cloud? This is far from being a technology-only question. In some cases, regulatory issues mandate that your data stay within a particular country; with today's global load balancing, that can't always be put into a service agreement. "It's 10:00—Do you know where your data is?" isn't just a clever take on an old TV ad. The abstraction layers that were so exciting when we were talking about the technology can be incredibly complicating when it comes to policy. We know that the federal courts wanted to use some of the emerging cloud backup solutions, but proxied Internet access combined with out-of-country storage prevented at least one try at adoption.

Second, does the cloud service support your existing applications, or are you looking at migration costs on top of IT retooling costs? The phrase "total cost of ownership" has been greatly abused in the last decade, but when considering a substantial shift in technology customers, you must think about training costs, temporary productivity disruptions, and support costs in excess of normal run-rate expenses. You also have to remember to extend your search for app support all the way out to the edge and in some cases out to your business partners. Consider a company like Walmart, for example: Some of their applications directly affect communications paths with their supply chain. If they were forced to push a major process like supply chain into the cloud, would they also be forcing their suppliers to upgrade similarly? The answer is almost certainly "Yes," and while Walmart has the market muscle to ensure that suppliers follow along, most companies don't have that much clout. Understanding how far the ramifications of a shift to the cloud will spread is another key consideration for executives pondering the change.

A commonly overlooked application with organization-wide ramifications is the email and calendaring combo, especially as they connect to the enterprise directory infrastructure. When we reviewed Microsoft's online services, some of the key questions were about the costs and mechanisms required for the migration. We looked at whether it was better to migrate completely or to try to make a cloud application platform coexist with a large legacy active-directory infrastructure. Microsoft's online services had migration tools for active-directory infrastructure, but other cloud service providers may not.

In Chapter 4 we talked about the analogy of using the cloud like a rental car, and taking the technology for a test drive before buying something you'll have to live with for years. If you're serious about considering Microsoft Exchange for your business, take it for a test drive using Microsoft Office Online services for a representative segment of your user community. Live with it, learn it, and make sure you find all the warts. While the trial is going on, make sure someone keeps track of the hidden costs are. How much time is it taking to manage? Did someone have to go out and buy a whole bunch of books to learn how the pieces fit together? Can you realistically support this if you decide to move forward? Just think to all the pieces you already have to fund, and imagine the increase or decrease in support cost when/if the program is expanded.

It should also be reiterated that clouds are great because they're normally pretty easy to walk away from. Instead of holding the pink slip on a new data center, you can just walk away if the project turns out to be a bust.

Bandwidth and Business Limits

Next under the microscope is the question of external versus internal bandwidth. A decade ago some people thought we were about to enter an era in which bandwidth would be the cheapest possible commodity. In 2009, bandwidth costs were carefully watched and considered by every company. Moving application bandwidth from LAN links that aren't metered to WAN links that are is another of those costs that must be carefully considered when a move to the cloud is proposed. In addition to the dollars to move bits, there are the dollars represented by application performance to consider. Those critical enterprise applications that were so snappy when they had to travel only through internal gigabit pathways now have to make it through to a cloud, a pathway that includes the corporate firewall and the rest of the security infrastructure. Now, the list of factors to take into account includes pieces of the network infrastructure. Is that firewall even capable of handling the new aggregate throughput of shoving that application into the cloud? Is your external Internet feed even big enough for your internal users? The impact of network bandwidth and infrastructure is dramatic, but it is only one of the technology issues that need to be taken into account when working toward the decision to expand enterprise applications into the cloud.

Testing for Clouds

Determining whether you have the necessary bandwidth can run the gamut from simple to extremely complex, though as the complexity increases, so does the accuracy of the model. On the simple side, you can use a site such as Speedtest.net and choose a server target that's fairly close to your cloud provider. Speedtest.net will toss a bunch of files back and forth to give you a thumbnail of the throughput possible between your two sites. However, this simplistic view of the world uses fixed packet sizes over a short duration, and it measures the throughput at only a single point in time. You might consider using Iperf, where you can vary the packet size and duration of the throughput test. Although it has the ability to run under Linux or Windows, iPerf is still fairly simplistic, but at least it considers the fact that network traffic isn't all made up of single-sized packets. At the complex end of the spectrum, Ixia Communications is now the owner of the Chariot application throughput test tool. This piece of software consists of endpoints and a management console. The management console allows you to set up synthetic traffic patterns between the endpoints that can consist of varying amounts of different traffic types. For instance, you use a protocol analyzer and a network tap to look at the traffic exiting your firewall. You find a mix of HTML, SSL, IMAP, POP, SMTP, FTP, and some miscellaneous stuff. The Chariot console can set up synthetic data streams that simulate a variable number of users doing different types of network functions. Since Chariot typically has access to all the resources on those endpoints, a single modern computer can easily simulate several users' worth of data. This gives you the ability to run after-hour's simulation of your entire company. What kinds of synthetic traffic you can toss around includes a pretty big collection, with data streams such as

- YouTube video
- Skype VoIP traffic
- Real streaming video

- SIP trunks
- SIP conversations
- Web traffic
- SNMP
- And many others

The power of this system is the ability to put endpoints on just about any workstation or server technology on the market and even some switch blades from various network equipment manufacturers. The ability to do “what if” scenarios on your network during off-hours is an extremely powerful tool, easy enough that you could run a bunch of “what ifs”: “If I moved my key applications to the cloud, would I have enough bandwidth for those specific applications?” “If there is enough bandwidth, is the link jitter and latency low enough to support voice-over-IP?”

Let’s assume you’ve done a bunch of testing, and so far it seems the answer is a slow migration to the cloud. The first step is to get a handle on what’s out there and exactly where it is.

Remote Access and the Long March to the Clouds

Not long ago, IT expansion meant more racks in the data center, more power to feed those racks, more air conditioning to cool them, expanding backbone connections to link them, and perhaps more IT staff for the care and feeding of those new servers. Those new racks meant capital expenses, physical assets, human resources, and recurring costs, all of which affect the bottom line. The question we’ve heard from CFOs around the world has always revolved around, “Is there a way to make that data center cost less?” The question has never been asked with more urgency than in the most recent two or three years, and the answers have never been more critical to the health of the organization. Cloud computing seems to offer an ideal way of reducing the capital costs and many of the recurring expenses, though we’ve seen that there are other costs that may limit the immediate impact of a migration into the cloud. While we’re still thinking about the costs of cloud computing, we should consider a few additional items that can weigh on the pro or con side of the decision.

Just what, for example, is the life cycle of the project you’re considering? Using the *New York Times* indexing project described in Chapter 4 as an example (http://open.blogs.nytimes.com/tag/aws), the *Times* was looking at several racks of blades, server licenses, Adobe Acrobat licenses, power, cooling, and personnel for a project that more than likely would have to be done only once. Then all those assets would either have to be sold, or re-tasked within the organization. This is where our CFO asks how much of our original investment can be recovered if we can return or sell these temporary assets. “Can’t you just rent that gear?” is a CFO war cry heard all over the world. What cloud computing gives us is the ability to give it all back, for a small fraction of the long-term asset cost.

With all the issues we’ve provided to think about, it’s possible that we’ve not yet considered the most important question: How, precisely, will you use the cloud? To begin answering this question, it’s useful to think in terms of models.

One of the models we most often hear about is “local prototyping, remote production.” This model had its roots in behavior that started in software development groups before cloud computing began. Programmers began installing VMWare or Virtual Server onto their workstations simply to provide prototyping for new systems. The reasons were fairly straightforward: Virtual machines were far less expensive than actual banks of new computers, and virtual operating system images that are hosting still-buggy applications in development can be blown away and regenerated much more quickly than similar images running on dedicated hardware.

So far we’ve talked only about savings on physical infrastructure. How about demand-based expansion? An application or set of information that is unavailable because the server can’t keep up with demand is just as useless as an application that is bug-ridden. While excess demand can be considered a “high-class problem,” it is a problem, and it can come from a variety of sources. Depending on your target market, your company might be SlashDot’ed or covered by CNN and get a massive surge in Web traffic. If you were to have enough foresight to try to plan for this, what would it cost you? We’ll look next at a couple of ways to implement a strategy for this situation.

Traditional Server Load Balancing

The first server load balancing systems were simple: They just divided up the incoming Web requests among several physical servers. They did this based on simple algorithms that depended on basic round-robin scheduling or elementary demand-feedback routines. These load balancers had the advantage of also allowing for maintenance of a server by shifting its load to the other servers in the group. These Layer 4 (referring to the ISO seven-layer networking model) devices had a single public IP address for each service (FTP, HTTP, etc.) and were configured to split the incoming traffic up among two or more physical servers behind it. Coyote Point has a great demonstration on their website: http://support.coyotepoint.com/docs/dropin_nav.htm.

A typical load balancer configuration would go something like this:

1. The DNS name for the server cluster is set up to point to the outside or public address for the load balancer.
2. Inside or private addresses are assigned to various servers behind the load balancer.
3. The load balancer is then told which private addresses are serving what type of network service (i.e., Web, ftp, email) and whether a weight should be assigned to larger, faster servers in the collection.
4. Then a choice is made as to what kind of load balancing should be used: round-robin, Gaussian distribution, weighted average, etc.
5. If a machine needs servicing of some sort, the system administrator declares a machine to be out of service, and the load balancer shifts load to the remaining servers.

Key to this whole arrangement working is that each collection of servers has access to some sort of common storage system (i.e., NFS). Load balancing in many cases came in the back door as a method to extend the backup window for many critical services. By shifting the load off a primary server, it could be frozen in time and have a full backup done without worries about open files and such. In many cases backups were taking longer than the system administrator's window of opportunity, forcing the migration to some sort of load balancing.

The downside to this plan was that adding servers to respond to larger than anticipated loads was a long and expensive process, and the process was inherently reactive: In most cases, capacity couldn't be added until after the traffic surge had passed. More critically, the servers that were added were static, dedicated to a single purpose when deployed. Load balancing wasn't really dynamic in that FTP servers, for example, couldn't be reallocated to handle HTTP traffic without large amounts of human intervention. There's a way to balance in genuinely dynamic ways, but financial officers don't like it.

The workaround is to deploy a series of new servers, put all the necessary services for all applications on each server, but not route traffic to them until needed. This way a system administrator can quickly alter the load balancer's configuration to add additional Web servers to handle an unanticipated load spike. Once again, though, this requires encumbered resources sitting idle (and sucking up power and cooling) to handle a load spike that may never occur. This was all a guessing game played by IT groups all over the world, and a boon to hardware and software vendors worldwide. Now, this was not necessarily a bad thing, since backup facilities of some sort are part of everyone's business continuity plans. With virtualization and cloud computing, though, there may be a better way.

The Virtualization Load Response

Scyld Software (part of Penguin Computing) was the first company we know of to deliver products that saw computing clusters change from Beowulf scientific-style cluster computing to business clusters. In the Scyld system, a virtualization kernel was installed on each server in the cluster and applications were distributed across these. The distinctive feature of Scyld's software wasn't in the virtualization cluster, though, but in how this system could detect incoming application loads and apply business rules to the problem of how to handle unanticipated loads. The example the company gives was how they handled a massive spike in Web traffic. Their system would move the Apache Web server from a shared system (multiple applications all sharing a single physical server) to a dedicated server. If the setup was configured correctly, this happened automatically. An added benefit was that it was *not* bound to a single type or model or server, but rather could be run on a heterogeneous collection of boxes with weight assigned to them to vary the load.

A few years later, VMWare started offering a system called VMotion (www.vmware.com/products/vi/vc/vmotion.html), which took this idea quite a bit further. The VMotion concept was to have a collection of servers all running the VMWare infrastructure. Under normal circumstances, machine #1 could be running a collection of virtual servers that might consist of Apache Web servers and email services. Machine #2 might be running SugarCRM, and machine #3 might be running billing software. Let's imagine a case in which Company X has decided that if a huge surge in Web traffic occurs, the business won't be hurt if billing is delayed by a day. So their IT group has set up business rules that allow VMotion to shift the Apache Web server to a dedicated server if a huge load starts up. When the load disappears, the Web server will move back to a shared server and billing will be resumed. Those rules could be modified to also handle automatic migration of running servers to another physical server if a hardware failure should occur. This takes virtualization much of the way to the scenario that might exist when a company deploys a "private cloud." What's missing from this current scenario is how to detect when an application like Apache has crashed even if the virtual server is still up. Previously, IT professionals would write custom scripts for UniCenter or OpenView that would periodically probe to see if applications were running on the target machine and, if not, send a reset script to the system in question. Early efforts were more of a "Hail Mary" in that they would keep sending the reset over and over again if the application had crashed badly and restarting the system wasn't fixing it. More sophisticated scripts started appearing, and as the Microsoft Power Shell interface documentation became widely known, testing at the application level and then restarting more intelligently became commonplace.

Taking this knowledge base quite a bit further, Coyote Point has extended its application load balancer into the VMWare world to the extent that rules can be set up for spawning additional machines from prestored images. This generation of load balancers is able to probe higher in the ISO stack, and it has the ability to detect if a Layer 7 application like Apache has crashed and then do something about it. According to Sergey Katsev, Engineering Project Manager at Coyote Point Systems:

> Actually, we have a few customers who have a few applications "in the cloud" and still have a minimal datacenter "since they have control of it." Either way, app load balancing is needed since otherwise you don't know when your application has failed. . . . Amazon or whatever will guarantee that the "server" remains up, but they have no way of guaranteeing that your Apache Web server hasn't crashed.

With technology and deployment moving toward cloud capability, the next big question is where the servers and applications will live. This is the point at which the cloud begins to separate from simple virtualization, and decisions we've discussed earlier—decisions about bandwidth and networking infrastructure—are joined with business strategy concerns to determine whether it's time to move data and apps out of the local network. Now an IT professional has the choice to have apps live both in a local data center and in the cloud. It isn't a hard stretch to imagine that most of the time a key e-commerce app will live in a small but adequate data center in Corporation Y. Suppose, however, that a CNN reporter stumbles across their newest widget and highlights it every half-hour all over the world. Suddenly the Web load on this tiny little e-commerce app skyrockets, and if nothing is done the server in question will die a horrible death. However, preplanning has paid off, the meat-and-potatoes apps have already been set up in the clouds, and the load balancer is spinning up the cloud apps in a big hurry. Now, with the business surge spread across the entire North American continent (and the small but adequate data center), Corporation Y can reap the benefits of the CNN report.

Computing on Demand as a Business Strategy

Deploying applications or moving data to the cloud is rarely an all-or-nothing proposition. Instead, internal versus external computing is a balance whose formula is unique for every corporation. Using the criteria we've discussed in this chapter, you can build your own decision-making spreadsheet to aid in the process of deciding whether to try moving to the cloud. In later chapters we'll look at particular clouds and the impact they can have on your applications. In the rest of this chapter, we'll look at more general answers to questions about cloud strategies. Most of the answers will start with the assumption that you've already committed to move at least some of your data infrastructure to the cloud.

Regardless of which cloud you do choose, you should always keep in mind what mechanisms are in place to move data back to your internal data processing infrastructure if you decide not to continue the project, or if you decide that the initial balance of data or applications inside and outside the cloud should be changed.

An example may be useful here. While few would dispute that Salesforce.com is a great customer relationship management (CRM) system, the cost per seat is a key decision point in adoption for most organizations. One solution that we keep hearing about from different companies is about reducing the total seat costs by using Salesforce.com for the front-line salespeople, but something like SugarCRM for the call centers. Using two separate cloud applications isn't unusual, but it does lead to the question of where the data is stored and how it is moved from one application to another. One company had a middleware data mashup product from Apatar.com periodically moving data back and forth to make sure the call center knew about recent outside sales activity. This little company with roots in the old Soviet Republics also has offices in Boston, and is addressing the huge data conversation market. It's not hard to imagine a sales manager looking at the huge cost per seat for something like Salesforce, yet wanting to populate a hundred seats in a call center. This solution is tailor-made for this exact situation: The sales manager can download a free copy of Apatar and drop connectors onto the Apatar workspace. Each connector has a set of credentials for the data source, and has connector nubs on them for tools. Easiest are straight field conversions, where one program uses "firstname" and the other "fname"; harder are the items where one separates the first and last names and another uses only fullname, or where one program uses department codes and the other uses names. All this type of data manipulation is simple with this type of tool. Considering that we've heard of all kinds of companies paying some pretty big bucks for this type of data migration, it's no wonder that this tiny little company has gotten so much attention. Although it is certainly not the only tool of this type, this drag-and-drop data mashup tool is certainly worthy of attention.

While cloud computing has begun to take hold at the opposite ends of the computing spectrum, we're also seeing clouds gaining traction in the small-to-medium-size business (SMB) market. As the SMB world seeks to use Internet presence to compete with companies both larger and more agile, we've seen a shift in how they're starting to use cloudlike applications to leverage their Internet presence, allowing them to provide considerably more services to their customers than with traditional data processing methods.

As one example, we're seeing more and more Web designers taking responsibility for maintaining Internet servers. On the one hand, smaller organizations don't have the resources to dedicate workers to a single IT task. On the other hand, historically it has been these situations, where IT workers are required to perform multiple tasks, where systems administrators become less vigilant and attackers are able to exploit security weaknesses to turn those weakened servers into illegal download sites or zombies in a "botnet" army. This liability seems to be a new driving force for SMB organizations to look at clouds, to sidestep the potential liability of maintaining a server farm. However, this trend has some unintended consequences if we look further down the IT support chain.

Considering just how much Web design talent there is out in the world, it just makes sense to leverage this talent pool for special or new projects. Traditionally, you had to spin up a new server, customize it, add users, do penetration testing, fix the holes, load the application development environment, and then invite the contractors in to play. But all you're really after is some cool clean Web code for your smoking-hot new site. So why not spin up a site in the clouds, and get up and running on this new project in significantly less time? Since any cloud vendor worth anything has already done the patching, securing, and penetration testing, you can probably spin up a development site faster than you can steam a latté.

Clouds may sound like a do-all strategy, but that silver lining also is a stormy scenario for value-added resellers (VARs). Countless small and medium-sized companies look to VARs to provide the application development and IT support that they cannot supply from internal sources. What we question is whether the outsourcing trend is becoming a crutch. VARs aren't always going to look out for the best interests of the customer as they look to increase their profits. What we can't tell is whether this trend toward cookie-cutter solutions is also going to stifle the creativity that has made the Internet such a great resource. Or will this trend toward standardization make it even easier to migrate to generic clouds? The successful VARs that we've seen are the ones that have used hardware sales only as a service to their customers; and instead are using the outsourcing trend to provide high-profit services. We've especially seen this as giants such as HP, Dell, and IBM carve up the computing hardware market and somehow survive on tiny profit margins. The trend over the past decade has been toward services, and we just have to believe that those services are eventually going to live in the clouds.

A saving grace is that cloud vendors are working with many VARs to develop new profit models for this part of the industry, and the same vendors are looking to build direct partnerships with customers—direct partnerships that some say will reduce the need for SMB customers to rely on VARs for the bulk of their IT support. We maintain that with any paradigm shift in the IT industry, there will always be some pain as we see the adoption of the new technology. Some of the retooling examples we've seen are from mini-computers to PCs, from PCs to the Internet, from paper to intranets and the Internet, and from 800 telephone numbers to websites. Each technology shift has been a double-edged sword, with ramifications both seen and unseen. Said a different way, there will always be some fallout as the next disruptive technology appears, but the survivors will be those who plan for the change and manage it, rather than hiding from it.

It's difficult to forecast with any accuracy precisely how all the economic pieces of a major technology shift will work out. It's certain, though, that cloud computing is bringing about shifts in the way companies think about the allocation of costs in IT. Part of those shifts deal with recurring costs, and many of those recurring costs are built around partnerships with cloud vendors and with VARs. We're also predicting that as the comfort level sets in with clouds, the finance folks will start to get used to the concept of the rent-a-data center attitude that clouds can provide. If you look at the processes that went on during the *New York Times* indexing project, you can easily see how the temporary nature of cloud computing has really started to catch fire.

Let's now look a little more deeply at the cloud's impact on partnerships.

The Cloud Model for Partnerships

"There is no way I'm going to give Company X a log-in to my server!" We've all heard this before. It might be personalities, it might be regulations, or it might be just plain paranoia, but all the same, we often run into situations where it could save Company X a huge amount of money if Company Y's buyer could just log-in and check inventory for a fast-selling widget, yet Company X can't seem to loosen its corporate controls enough to let it happen. The problem in this case is where we put neutral territory that both companies can access and control without exposing their internal IT infrastructure. It's not an unreasonable position, really. Security surveys during the last couple of years have indicated that partners are a huge, legitimate security threat for most companies. If we assume that allowing access to certain "inside the firewall" information is a legitimate business need, how can we make it happen without unnecessarily endangering the inner workings of our corporate network?

The answer for some has been to use a cloud service as a common area for cooperative processing. Instead of spending the time and money building a neutral zone, why not use a service? After all, it wouldn't be hard to have several images to work from, with one running for Company Y and another running for Company Z. The common files, and common network access points, are outside either company's security perimeter, allowing the files to be shared without requiring that security protocols be breached. All data transfer can take place over secure VPN tunnels; policies and procedures can be put into place to govern precisely which files can be synchronized to cloud storage.

Let's look at a scenario where all the public-facing systems for Company X live in the cloud, but finance and human resources live in the company's local data center. However, finance certainly needs to get data off the public e-commerce site. Depending on what consultant you ask, the answer is most likely going to be some sort of proxy. The whole idea is to hide from the outside world the details of what the inside of the company looks like.

On a more personal scale, the authors used the Microsoft SkyDrive cloud to write this book. Instead of going through all the hassles of setting up a DMZ server in either of our facilities, we found it much easier to use a cloud service to store drafts of the book along with support material, images, and notes to ourselves. We could have easily built a system on a spare server, but that would have taken a machine away from our testing infrastructure and someone would have had to maintain it. This way, we can always get to our material, it's backed up by someone else, we aren't paying utility bills, and we didn't spend all the time to bring up a content management system. We've heard the same story from countless others who needed a common storage area for a project, but who couldn't or wouldn't open the firewall for the other party.

Going a bit further into Microsoft's cloud offerings, the folks in Redmond didn't leave SkyDrive to handle all the cloud file storage chores; a separate service called Live Mesh automates the process of synchronizing files between a computer (or a series of computers) and the cloud. Of course, Microsoft is far from the only provider of services like these. Dropbox, for example, is a popular file synchronization service that provides cross-platform automated updating. Media Fire is one of the many cloud services that allows you to share files with any number of people with whatever level of security suits you best. Of course if you're using a Mac, you've practically had Mobile.Me rammed down your throat.

What systems like these provide are a fertile ground for customized connections that provide data synchronization and a place for applications to more easily exchange information. Amazon's S3 storage system is a frequently used platform for development, and we've started to hear about developers writing wrappers for the system that will allow multiple parties to mount a common storage area with full read/write privileges. So we can easily imagine a special directory on both Company X and Company Y servers that are a common area. In this example, neither company is exposing its entire infrastructure, but both companies are able to access a shared directory. One provider of just such a solution for Linux servers is SubCloud.com (www.subcloud.com), where an application installed either in the cloud or locally extends the server's capabilities to share the S3 storage. A good analogy is how an income tax preparer uses a special set of forms to convey your income tax information to the Internal Revenue Service. Formerly, the common data transmission medium was the U.S. Postal Service. Now, those same forms are electronic, so the tax preparer sees an image that is very familiar—just like the old forms—but the IRS sees tagged information fields transmitted to a public proxy and eventually input into the IRS processing system. The point is that the data can enter a DMZ in one format and exit in another. It can also scrutinized at several levels, so that a certain level of trust can be established. Perhaps you could call your proxy "Checkpoint Charlie"?

At the workstation level, there is a cross-platform solution by Bucket Explorer (www.bucketexplorer.com) that utilizes a file explorer-like interface to provide team folders on the Amazon S3 system. That has a direct analog from both Microsoft and Apple. The point is that data can be input on a Mac, examined by a Linux machine, and then perhaps a Windows machine could be the SQL host that stores all the transactions.

The issue of interface—how data moves from local network to cloud application or from desktop to cloud server—is one of the issues that differentiates one cloud system from another. There are, if not an infinite number of ways to make these things happen, at least a large number of options. We've already seen the drag-and-drop interface of Skydrive and Media Fire, and the automated synchronization of Mesh, Mobile. Me, and DropBox. There are many others as well, including some with roots in earlier, nonvirtualized operating systems. Some developers have significantly stretched the original intent of the "named pipe" interfaces by having processes on different servers using a shared file system for interprocess communications. The concept is that a Python app running on Amazon EC2 might have a file mount to Amazon S3, but Company Y's Linux server also has that same Amazon S3 share-mounted on its accounting server. With a shared file area, the IT personnel can work cooperatively to implement a named pipe on the shared area so that immediate information on widget orders can be transferred from one company to another without exposing anyone's internal infrastructure. Peter A. Bromberg, while exploring the possibilities on the Egghead Café for .NET programmers, noted:

> The point we're trying to make goes back to the quote from Sir Isaac Newton about standing on the shoulders of giants. Just because the original intent of this was X doesn't mean it can't be extended to do Y. It should also be pointed out that named pipes aren't the only method for inter process communications, just one of the legacy methods commonly found.
>
> (*Source:* www.eggheadcafe.com/articles/20060404.asp.)

Seeding the Clouds of Federation

Before we leave the topic of cloud applications that allow data to be shared among different systems, we should look at ways in which user information—user identity—can be shared in the same way. The concept is called *identity federation,* and it's one of the big ideas that cloud computing is bringing to reality a bit more quickly than might happen if clouds didn't exist. In simple terms, identity federation is a single authenticated user identity that is accepted as valid across a wide variety of systems. While the concept of having a particular type of user identification exist in two organizations might be easy to picture conceptually, the implementation has been fraught with heated arguments in the standards committees. Because the company that owns a customer's directory has a huge advantage in owning the rest of the organization's network infrastructure, vendors tend to want to feature their own solution to the exclusion of all others. With Sun Microsystems pushing LDAP, Novell pushing eDirectory, and Microsoft pushing Active Directory, the battle is a three-way slugfest among some of the biggest IT providers on the planet. Each bases identity management on a directory structure that vaguely resembles the work done in the X.500 standards committee but is tweaked to the individual company's benefit (www.infoworld.com/article/05/10/07/41FEidm_1.html?s=feature).

We'd like to point out that one of the huge roadblocks to federation has been the issue of government regulations. The medical industry's HIPAA rule set has certainly affected consumers by requiring a large number of new forms to sign acknowledging that their medical providers are compliant with HIPAA regulations and that they'll make every effort to protect your personal medical information. What hasn't been said is that HIPAA, Sarbanes-Oxley, and other federal legislation doesn't specify technology, only overall effects. The government doesn't say you must use AES256 encryption, but instead alludes to "secure communication pathways." This fact is creating a new era in the way medical providers share information and communicate with patients. A typical hospital doesn't own its laboratory, but rather provides space to a contractor to provide med tech services. When HIPAA first went into effect, many hospitals reverted to paper records to avoid having to answer privacy questions they really didn't know how to answer. However, as the scare faded and clearer thinking prevailed, medical providers realized that setting up a clearly defined procedure and risk management could provide just as much privacy as paper, perhaps even more. The Japanese have even gone as far as providing an even easier way for patients to identify themselves, so that they can start from a strong position of trust. Fujitsu Limited has produced a whole series of kiosks that scan the blood vessels in the palm, allowing for positive identification but without the resistance faced by other biometric identification systems. The Japanese figured out that if you start your information chain from a strong position of trust, much more can be done with less risk.

Let's clear the air a bit and say that each of the players in the debate about federation does seem to have the common goal of being able to interoperate. Each of the vendors agrees that creating a facility that would allow you to create special-purpose users on your system is a good thing *only* if it doesn't also expose your internal infrastructure to attack. That's it—we're all talking about literally how to implement that simple Venn diagram showing an overlap in authority between two organizations. The fight is really about how you determine trust so that you can more comfortably manage the risk for each transaction.

Suppose that Mary, an employee of Whapapalooza Widget Works, needs to place an order with Fergenschmeir Sprocket Works for 100 dozen size 20 sprockets. She and dozens of other defense contractors do this often enough that the folks at Fergenschmeir have been screaming for three more order-entry people. However, the enlightened IT staff at Whapapalooza and Fergenschmeir have discovered that their two internal IT infrastructures have an agreed-to standard for “federation.” Each IT group has created a special user group that has privileges *only* in specific areas. Each has also assigned a group manager so that personnel changes in one company won’t affect the other. *InfoWorld* magazine did a huge article on just this kind of thing way back in October 2005. The scenario mapped out a merger between two companies and followed the changes to a single employee. In this early comparative review, federation was only a buzzword, but the authors had long conversations with the vendors on just how federation would be implemented. Identity management, security event management, and federation all seem to be intertwined and no longer really exist as stand-alone subjects. All of these are being woven into the base operating system regardless of whether it was designed to be a monolithic system or virtualized. Considering the massive changes made to Windows Server 2008, the borders have certainly blurred.

However, the fight isn’t over yet, and right now there just isn’t a standard for federation in the world of identity management. However, there is a silver lining, and it’s in the cloud. All Whapapalooza and Fergenschmeir really wanted to do was automate the ordering process so that neither company would have to encumber additional personnel to handle intercompany orders. A common area in which to place and acknowledge orders might be set up in any of the cloud services available. In Amazon it might be a virtual DMZ server, or maybe a shared storage area on Amazon S3 for named pipes, or a Python application in the Google App Engine. Like a Swiss Army knife, there are lots of ways to use the tools at hand.

Let’s step back a few years and look at the early days of credit card validation. Although it was not the first, Verifone was founded on the idea of small simple devices that could read the magnetic stripe on the back of the credit card, call a credit bureau to validation the transaction, and then get back some acknowledgment by IC Verify for the transaction. This simple idea was applied to network applications in a simple DOS application that looked for files in a specific directory with a specific file extension. Upon finding those files, it would do something very similar to what Verifone did, but this time with a regular old computer modem. What made this different was how the system would keep the modem link up as long as it kept finding files in that directory. So, in many high-use cases these systems didn't drop the line all day. Credit card clearing houses now exist all over the world, but the concept is still the same. You've acknowledged a level of trust with the clearinghouse that in turn has a level of trust with the banks or credit card companies. Each in turn passes data along in a particular manner, but can't do anything beyond what is agreed on—thus dramatically limiting the potential for mischief. Key to this trust relationship is a third-party validation service called a Certificate Authority. In any typical browser today there exists a list of hosts that are considered trustworthy, and each of those servers takes part in a validation dance that utilizes dual-key encryption technology.

As a historical sidebar, modern encryption systems all spring from work originally done at MIT by mathematicians Ron Rivest, Adi Shamir, and Leonard Adleman (RSA Corporation was named for their initials), who were the first to create a commercially viable encryption system that utilized one encryption key to "lock" the transaction and a completely separate encryption key to "unlock" it. This dual-key encryption became the basis for almost all secure Internet communications today. More important for this discussion is how this same mechanism can be used to authenticate information. The "private key" is used to create a numerical representation of the message. To validate this message, the recipient retrieves the "public key" from a trusted Certificate Authority (all a Certificate Authority does is hold onto public keys for servers). The original work that led to this advance was done in Honolulu, Hawaii, by Wesley Peterson, PhD, in 1964. His paper on the mathematical representation of data for error correction became the basis for all modern data transmission error checking and all modern encryption. Today, Peterson is acknowledged as the father of the cyclical redundancy check used in every data transmission.

Is federation happening now? You bet! Just look at how Amazon's massive Internet sales site can place orders with hundreds of companies all over the world. The sophistication of the federated identity varies widely from organization to organization, but the goal is the same: Provide more services between companies but not at the added expense of human resources. After all, the biggest cost in just about any organization is warm bodies.

Clouds Flight Path for Chapter 5

- *Will government regulations prevent you from using the cloud?* We all know that government regulations play a huge part in how various organizations do business, and clouds have to learn to play along. We've looked at some of the issues you might stumble across as you think about moving into the cloud. Considering that we've had some friends retrieve their files and discover that they came from Italy, it really pays to do your homework and make sure you buy the right options. While the big boys are all offering regulatory options, you must ask for them, and they might very well need to be part of your service-level agreement.
- *To use clouds internally, you really need to examine the size of your Internet pipe.* It doesn't pay to move your internal computing facilities into the cloud if your Internet pipe is tiny. It's all about balance, and about looking at every piece in the puzzle. Remember that some applications are very timing-sensitive and don't lend themselves nicely to being shoved into a cloud. Here is where taking it all for a test drive really makes sense. Don't take the word of the salesperson; test it yourself and make sure it's worth risking your reputation on the move.
- *There are different types of load balancing, and a good load balancer can also provide auto provisioning.* We looked at some big Web surges and how various organizations handle them. Load balancing is a way of life as your audience grows. We mentioned some key factors you should consider, and we discussed why load balancing is making even more sense today, especially because it can actually help you strike a balance between in-house infrastructure and the cloud.
- *You can use a cloud as a DMZ between partners, just as good fences make good neighbors.* Setting up some neutral ground makes a whole lot of sense and limits risk for everyone involved. We're only human, and there is always potential for mistakes. It's said that good fences make good neighbors, and that's certainly the case with business partners using the cloud as neutral territory for exchanging information.
- *The seeds of federation are finally sprouting.* That no man's land might very well finally give federation a chance to bear fruit. Will this be the beginning of the business world coming to some sort of agreement on just how to handle foreign trust relationships, and will clouds become the Switzerland of the computing world?