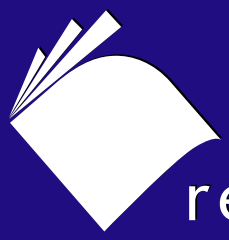

realtimepublishers.com™

The Administrator Shortcut Guide™ To

User Management and Provisioning

abridean

Dave Kearns

Introduction

By Sean Daily, Series Editor

Welcome to *The Administrator Shortcut Guide to User Management and Provisioning*!

The book you are about to read represents an entirely new modality of book publishing and a major first in the publishing industry. The founding concept behind Realtimepublishers.com is the idea of providing readers with high-quality books about today's most critical IT topics—at no cost to the reader. Although this may sound like a somewhat impossible feat to achieve, it is made possible through the vision and generosity of corporate sponsors such as Abridean, who agree to bear the book's production expenses and host the book on its Web site for the benefit of its Web site visitors.

It should be pointed out that the free nature of these books does not in any way diminish their quality. Without reservation, I can tell you that this book is the equivalent of any similar printed book you might find at your local bookstore (with the notable exception that it won't cost you $30 to $80). In addition to the free nature of the books, this publishing model provides other significant benefits. For example, the electronic nature of this eBook makes events such as chapter updates and additions, or the release of a new edition of the book possible to achieve in a far shorter timeframe than is possible with printed books. Because we publish our titles in "real-time"—that is, as chapters are written or revised by the author—you benefit from receiving the information immediately rather than having to wait months or years to receive a complete product.

Finally, I'd like to note that although it is true that the sponsor's Web site is the exclusive online location of the book, this book is by no means a paid advertisement. Realtimepublishers is an independent publishing company and maintains, by written agreement with the sponsor, 100% editorial control over the content of our titles. However, by hosting this information, Abridean has set itself apart from its competitors by providing real value to its customers and transforming its site into a true technical resource library—not just a place to learn about its company and products. It is my opinion that this system of content delivery is not only of immeasurable value to readers, but represents the future of book publishing.

As series editor, it is my raison d'être to locate and work only with the industry's leading authors and editors, and publish books that help IT personnel, IT managers, and users to do their everyday jobs. To that end, I encourage and welcome your feedback on this or any other book in the Realtimepublishers.com series. If you would like to submit a comment, question, or suggestion, please do so by sending an email to feedback@realtimepublishers.com, leaving feedback on our Web site at www.realtimepublishers.com, or calling us at (707) 539-5280.

Thanks for reading, and enjoy!

Sean Daily

Series Editor

Copyright Statement

Chapter 1: Provisioning and the Management of Users

User management is the process of adding, maintaining, changing, and removing user accounts, passwords, authorizations, and attributes from a (usually networked) resource— files, printers, applications, databases, Web sites, and other hardware or software. In today's network, these tasks typically involve using a directory service. However, user management, as you will see, has a longer history than does the enterprise directory system.

Provisioning is the process of insuring that managed users have the requisite information and privileges to enable access to various services, systems, and resources within the enterprise when and where they need it. Further, provisioning allows this access to be modified or removed quickly, efficiently, and automatically whenever the situation changes. Provisioning also usually includes components that are normally considered outside the scope of the Information Technology (IT) organization such as phones, premises access devices, and even company cars.

User management can be thought of as a function of IT; in contrast, provisioning is generally an enterprise function that is facilitated by IT. The difference will become clearer as we delve further into these two topics. Both subjects, by the way, are aspects of the larger arena of Identity Management.

Whenever a particular point refers to both user management and provisioning, I'll reference it as Identity Management; otherwise, I'll indicate whether I'm discussing user management or provisioning.

In this chapter, we'll explore the history of user management and provisioning and take a look at the current impediments blocking implementation in many enterprises. We'll also delve into the realm of directory services, the "plumbing" on which both user management and provisioning are built. There are a number of choices that must be made in the area of directory services for a successful user management and provisioning implementation and this chapter will present the pros and cons of each. Finally, I'll present a few ideas gleaned from forward-looking enterprises (or, at least, those on the bleeding edge) about ways to improve the process and streamline decision making. Chapter 2 will explore user management in depth and Chapter 3 will discuss electronic provisioning in detail.

How Did We Get Here?

User management is actually rather late to the IT party. Back in the days when IT was referred to as Management Information Services (MIS) and "the computer" lived in a big, climate-controlled glass room, the only management of users was making sure they lined up properly waiting for the delivery of printouts. "Authorization" meant identifying yourself to the guard at the door and "access control" meant having a key to the door of the computer room. The only people who actually interfaced with the computer were called "operators" and they all used a single account—actually no account at all.

Timeshare systems, the advent of UNIX-based shared hosts, and especially the proliferation of workgroup, departmental, and enterprise networks of personal computers lead first to the use of user accounts to separate "my stuff" from "your stuff" and later to the concept of shared things, or "our stuff." Rudimentary security, in the form of simple passwords to control access, was fairly prevalent by the 1980s; however, the use of file, folder, and application security didn't really hit its stride until the late 1980s and early 1990s.

UNIX systems allowed a matrix of permissions that was three by three—read, write, and execute permissions for the user, his or her group, and "everybody", whereas Novell, and later Microsoft, introduced a rich array of rights and privileges that could be assigned to individual users or groups of users. The advent of directory services—both those tied to the network operating system (NOS) directories and those that evolved from email address books into global access lists (GALs) and into what are now called enterprise directories—further enriched the opportunities to control, on a granular level, the authorization of users and their access to resources.

The Evolution of Provisioning

Provisioning is both a more recent development than user management and a much older concept. Today's use of the term provisioning (often called *electronic provisioning* to differentiate it from older, manual provisioning) is simply describing the automation of the many manual tasks that Human Resources, Facilities, and IT departments have had to do for employees for many, many years. The term was adopted from the telco environment in which it has been used for decades to cover the process of providing *premises equipment* (for example, a telephone), identification (that is, a phone number), and a dial-tone to subscribers.

However, the telcos didn't create the term; they simply appropriated it from its earlier usage, which goes back centuries. When Christopher Columbus importuned Queen Isabella to pawn her jewels to support his expedition, it wasn't that he needed the money to hire a crew—typically the crew was paid at the end of a voyage with a share of the ship's profit. Mariners needed lots of money to "provision" their ships—provide food, rope, guns and powder, blankets, livestock, trade goods, navigational instruments, and more. Everything the ships might need had to be bought before they sailed because there were no shops along the way. Store keepers, unlike sailors, required "cash on the barrelhead" (you didn't get the barrel until they got the cash) for all purchases. This entire process was and still is called "provisioning the expedition."

Today's usage is surprisingly close to that definition, as electronic provisioning seeks to provide an employee with everything needed to navigate to and use the resources of the enterprise. You might say we've moved from provisioning the world explorer to provisioning the Internet explorer.

What Is Slowing Us Down?

If user management and provisioning have been with us for so long, why are so many enterprises lacking in implementations? Why, in fact, are so few provisioning systems in full production throughout the enterprise? As we'll see in Chapters 2 and 3, the technology is available, so what is holding back the rollouts?

The answer is two-fold: religion and politics. In addition, a third, although minor, factor is that user management and provisioning just aren't as technologically intriguing as Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), and Web Services.

Religion

The religious issue involves true believers—those who think their way is the only right way. This fight is of the "Apple versus Microsoft" and "Linux versus Windows" variety. There are still interminable arguments about the operating system (OS) platform, which only adds to the problem of choosing a directory services platform:

- Microsoft Active Directory (AD)
- eDirectory (formerly Novell Directory Services, formerly NetWare Directory Services)
- Sun Java Services Directory Server (formerly Sun ONE Directory Server, formerly iPlanet directory server formerly Netscape Directory Server)
- OpenLDAP
- IBM SecureWay
- And more than a dozen others

Further up the decision tree, some directory services are based on the Lightweight Directory Access Protocol (LDAP) specifications for a Directory Information Base (DIB—a database of identities), while others support relational databases built on Structured Query Language (SQL) specifications.

SQL vs. LDAP

The argument about whether SQL or LDAP makes the best data repository is actually a misrepresentation. SQL stands for Structured Query Language, a language developed by IBM to speak to relational database management systems (RDBMSs). LDAP, of course, is a protocol developed at the University of Michigan as an easier way to access x.500 (a standard used by international telephone services) directories. Neither actually describes the structure of a data store; however, both have become synonymous with the database systems with which they are closely aligned.

Directory gurus thought they had solved these issues in the late 1990s when a new beast, the *meta directory* was introduced. The meta directory, theoretically, consolidated information from all the other directories and identity storage areas on the network into a single destination data store. Thus, the directory religion fundamentalists could each have the directory of their choice to do their bidding while enterprise systems would speak directly to the meta directory.

Politics

Political issues concern control, power, and "turf." Although directory gurus saw the institution of meta directories as a way to overcome the religious arguments, business managers saw them as a threat to their turf. What directory managers saw as simply consolidating directories into one, large enterprise directory was also seen as removing "ownership" of the data from the business managers who, more often then not, create and maintain the data.

The fact that directory managers rarely get around to removing the data after its usefulness expires is a driving force behind the move to provisioning.

The meta directory leaves that ownership and control with the business manager, simply providing a central place for applications and services to access the data while ensuring that the authoritative data (that is, data created, changed, and removed by the business manager "owner") was synchronized throughout the enterprise.

As it turns out, ownership of the data doesn't seem to be the entire problem. Not only must some business managers own and manipulate the data, but they must also be the *only* conduit for disseminating it. Only they can decide how and when that data can be used. Technology cannot overcome this problem—attitudes will have to be changed.

There is also a second political issue preventing the implementation of user management and provisioning. Users are becoming more aware of how much data about them can be available in the meta directory. No matter how you explain the safeguards on use, and penalties for misuse, of the meta directory data, there will always be some people who will object that because misuse is possible (no matter how unlikely), it will eventually occur. Technology cannot overcome this problem, either.

Solving the Problems

The "religious" issues usually cannot be solved by logical arguments and discussions. It is necessary, most of the time, to enlist the enterprise's executive management to exert their power to force a decision. Arguments based on Total Cost of Ownership (TCO), Return on Investment (ROI), and —most importantly—success, power, and prestige for the executive are the driving forces you should use.

The political issues are also susceptible to a power play that enlists senior executives to force a decision; however, this circumstance will often lead to resentment (and reprisals) by the business managers who perceive that they've lost power.

> ☞ Sandra Harrell, Identity Management guru for DSI Consulting, recommends that every Identity Management project include a facilitator. As she puts it, "The key to a successful project is to have a facilitator who will bring all interested parties together and understand the unique needs of each. This person has to understand psychology, technology, and socio-economics as well as the technology aspects of the project/goal."

One More Consideration

User management and provisioning are generally thought of as global projects because they affect the entire enterprise. Your enterprise might consist of only 50 employees working from a single office, but your Identity Management project can still be called global in scope.

However, problems can arise, if your project is truly global—the more political boundaries (not office politics, in this case, but state, province, and country borders) the project crosses, the more regulations you will need to satisfy. For example, you might have to comply with the United States' Health Insurance Portability and Accountability Act (HIPAA), Europe's EC 95/46 Directive, and Japan's HPB 517. There may very well be conflicts among two or more of the jurisdictions that your project will extend into.

A project limited to a single country might not need to be reviewed by someone familiar with international compliance standards, there still might be differences from one state or province to another. In the United States, California has rules covering your responsibility should identity information of California citizens be compromised. Victoria province in Australia has its "Rosetta" project that includes privacy safeguards that might not apply in other areas of the country. The European Union has very strict privacy regulations that may or may not apply in European countries outside the EU.

Don't Lose Heart

Don't let these factors frightened you off of an Identity Management project. I discussed these considerations only to make you aware of the non-technical aspects of undertaking a user management and provisioning implementation. This book will present the technology you'll need and how to use it, but I wanted to make you aware of possible pitfalls that you can avoid by taking these considerations seriously. Your hard work will be rewarded.

The first step in the project is to decide upon the platform you want to use. There are many and no universal best choice. To determine which platform is ideal for your environment, let's explore the options.

Directory Services: The Platform for the Technology

The directory is where most identity information is stored for use by your Identity Management project. Although some identity data might be kept in a relational database, Windows registry, a text file, or anywhere else within your network (both on local machines and network server boxes and host boxes), a directory should be the primary storage area for the identity information your project is using. A directory service is simply a directory along with the services, tools, and utilities to create, maintain, and remove information. The directory service might also include reporting tools as well as management and monitoring tools for the directory itself (as opposed to the data that resides in the directory). The first challenge you'll run into is that very few organizations—especially those big enough to warrant full-fledged Identity Management projects—will have only one directory.

Multiple, Proliferating Directories

A few years ago when I was delivering a series of seminars on provisioning, I always asked each audience how many identity repositories were present in their organization. I vividly remember one person responding, "243, but we're not finished counting." Admittedly, that individual worked for a university, which are notoriously decentralized—every department might have a handful of identity storage places. But if you stop to think a moment, you'll probably realize that there are quite a few within your organization also:

- There is most likely a network login account and separate passwords for financial packages, Human Resource packages, and CRM packages
- Switches, routers, and other network devices might require login and might store personalization data.
- Anything that stores personalization data is an identity information repository.
- Don't forget intranet portals and Web-based applications and services.

Each of these requires user management of some degree. Total provisioning would require that all of these factors be joined together so that there is only one authoritative source for each of those hundreds, or even thousands, of bits of data about each managed user.

If you are thinking that you should start a project to consolidate all of the data into one enterprise directory before beginning your user management and provisioning implementation project, think again—you will never get to the user management and provisioning project. For the foreseeable future and likely beyond, there will be multiple locations for most identity data.

Strategies for Integrating Identity Data for New Applications

The major reason there are so many identity repositories scattered around your network is the existence of legacy systems. Legacy systems, created before most networks had centralized directory services, couldn't count on any particular directory store being available—there might not be any directory store at all. Thus, each system needed to find a way to keep its authentication and personalization data somewhere that could be accessed by the program or service when needed.

Windows programmers faced a similar situation pre-1995 (and the advent of Windows 95 with the beginnings of the desktop registry). They kept their personalization data in text files, usually so-called INI files (because they had the extension .ini):

- Some programs used their own INI file in the program's folder.
- Some programs put an INI file in the \Windows folder or the \Windows\System folder.
- Other applications used the existing WINDOWS.INI or SYSTEM.INI files.

Even after Windows 95 was released (in fact, even after Windows 98 was released) some applications continued to use INI files for identity information storage.

But you probably aren't dealing with a homogeneous Windows network. Most likely, your network is made up of numerous flavors of Windows, as well as UNIX, Linux, NetWare, MacOS, and SunOS systems—lots of complications, lots of places to store stuff.

When looking at new applications, and especially when creating new in-house applications, look for directory-enabled software using either an LDAP interface or written to something more specific, such as Microsoft's AD or Novell's eDirectory. The more things you can directory-enable, the easier it will be to consolidate information.

But where should you consolidate data? There are three possibilities:

- Enterprise systems
- Meta directories
- Virtual directories

We'll examine each of these options in a moment. First, let's look at the underlying database technology and why it might matter to you.

SQL vs. LDAP

In simple terms, those who come to Identity Management from database programming tend to favor RDBMSs—usually called SQL databases—as the repository for identity data. Those who come to Identity Management from network administration tend to favor simpler structures such as those used with network OSs to store username and password information and generally referred to as LDAP directory data stores (LDDSs). Each system has strengths; each has weaknesses.

Reads, Writes, and Replication

Although the RDBMS is relatively quicker for write operations, it's generally slower for reads. The RDBMS will typically offer multiple indices with a lower overhead than the LDDS (if the LDDS even offers a way to construct a second index or more). Most LDDS are able to be partitioned and replicated, thus making them pervasive and ubiquitous—they are always available and available from anywhere. Although a replicated and even partitioned RDBMS is possible, it's far from the typical installation, and with the LDDS these things are highly recommended, if not a requirement.

Accuracy and Timeliness

Accuracy and timeliness are provided in different ways by the two systems. Generally, the LDDS is a "loosely-coupled" system—updates and replications are not instantaneous to all instances. With an RDBMS, most changes are done with a transaction—either the entire transaction succeeds or the entire transaction fails. The LDDS could also be layered with a transaction tracking system, but it typically is not. Replications are done on a set schedule and are done on a one-to-one basis so that different instances of the data store may not contain 100 percent of the same information at any given point in time.

Auditing

Auditing is often handled through the transaction logs with the RDBMS but is usually an interrupt-driven "event notification" system for the LDDS. In either case, it is usually something outside of the system that monitors the auditing.

Choose the System that Works for Your Environment

Neither system is ideal for every environment. You must decide which benefits are necessary for your project, then choose the system that provides the greatest number of those benefits. Unless a particular strength or weakness is germane to your project, this decision will generally be a preference issue rather then one that is resolved technologically.

SQL databases are considered more robust, while LDDS databases are considered more efficient and responsive. Still, if it appears that someone with the power to cripple your project is digging in their heels in favor of one system or the other, don't waste time in fighting it out. All SQL-based directory services include an LDAP programming interface, so be sure to use the LDAP standard or one based on it such as the Organization for the Advancement of Structured Information Standards (OASIS) Directory Services Markup Language (DSML) for all your programming efforts in the project. This way, you can quickly move from SQL to LDAP (or vice versa) should a "religious conversion" occur.

> 🕮 For more information about OASIS, see http://www.oasis-open.org/.

Enterprise System, Meta Directory, or Virtual Engine

As I mentioned earlier, there are really only three possibilities for a consolidated identity information storage solution:

- Enterprise directory service
- Meta directory
- Virtual directory

Let's look at the pros and cons of each. Figure 1.1 illustrates the hub-and-spoke nature of all three solutions.

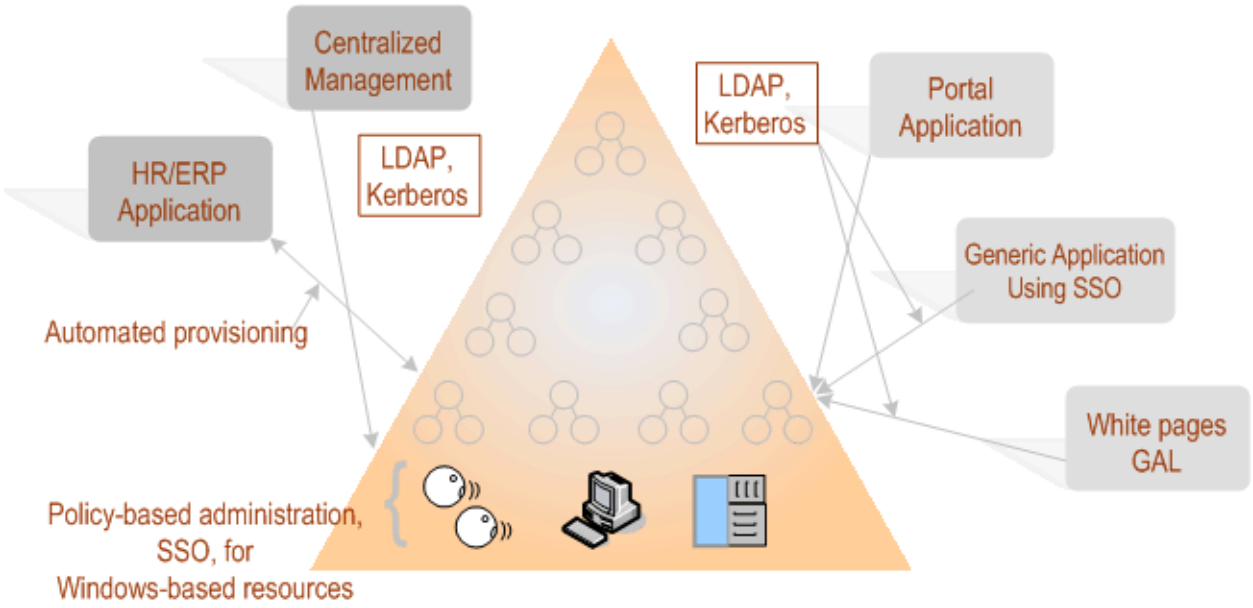

Figure 1.1: The hub-and-spoke architecture of enterprise directory services, meta directories, and virtual directories.

Enterprise Directory Service

If you could have all of your identity information stored in a single directory system—available at all times and in all places on your network both behind the firewall as well as over the public Internet—most of your problems would be solved. User management would be handled by the enterprise directory service, and provisioning would all be handled in one place.

However, as I stated before, "For the foreseeable future and likely beyond there will be multiple locations for identity data." Thus, the chances of you getting everyone to agree on using a single data repository, then agreeing on a particular repository, are somewhat lesser than your chance of becoming the next president of Microsoft. You might have an enterprise directory service, in fact, you should, but it won't be the only place identity data is stored.

Meta Directory

A meta directory is very similar to an enterprise directory with one major difference: although identity data is stored in the meta directory, it is not removed from the data store that contains it today. Applications and services can read data from the meta directory, but the original owners of the data still control what is added, modified, and removed. Thus, meta directories solve many of the political problems that I discussed at the beginning of the chapter (although it is still a very good idea to have a social facilitator on your project team).

The meta directory—a good example of which is Microsoft's Identity Integration Server (MIIS)—uses small applications called connectors or drivers to gather data from the myriad identity sources on your network into the meta directory, which then acts as a single directory. Although initially this setup might be a one-way street, with data flowing only from the existing identity sources to the meta directory, it is also possible for the flow to be two-way. Thus, data can be synchronized among the various storage locations, enabling any number of Identity Management applications and services including enterprise-wide user management and provisioning.

Meta directories usually offer all the robust qualities of a relational database such as transaction tracking, rollback and roll forward, replication, and two-phase commit. These features can guarantee the integrity of the data. There are some drawbacks, however:

- Meta directories generally require you to add yet another directory service to what is likely an already overburdened network.
- A meta directory will require quite a bit of storage and most likely its own server platform.
- There is a significant tradeoff between performance and accuracy. For the meta directory to be completely accurate, it would need to be updated each time something changed in one of the connected identity information sources. This requirement could lead to a lot of extra network traffic bogging down more important tasks. However, if you stagger updates so that they happen less frequently, even holding them until slow times on the network, performance can be improved—with the tradeoff that the data in the meta directory could be outdated and stale when it's read.
- Generally, you cannot query a meta directory via LDAP; to improve this situation, you can use a virtual directory.

Virtual Directory

A virtual directory is, in reality, the meta directory's synchronization engine without the overhead of storing the data (see Figure .1.2). Instead, the virtual directory uses a database to store pointers to the source of the data. When a service or application wants to read identity information, it contacts the virtual directory, which, in turn, goes out and reads the data from the original source and passes it on to the calling application. Think of a virtual directory as an identity proxy service.

The virtual directory uses the same technology as the meta directory—drivers and connectors—to join with and find the authoritative information it needs to serve. Novell's Nsure Identity Manager, formerly called DirXML, and Radiant Logic's RadiantOne Virtual Directory Server are good examples of virtual directories.

The biggest drawback to virtual directories is that reading data from a virtual directory might take longer than from a meta directory because the virtual product needs to go out to the source identity repository and the meta directory doesn't. Of course, the virtual directory is almost guaranteed to have 100 percent accurate information.

Although a virtual directory does usually store its information (pointers and such) in a database, this database is often not a fully robust relational system. Thus, the virtual directory is subject to loss in the event of a disaster. Of course, all of the original data is still available, it would just need to be re-synchronized through the virtual engine.

> Novell's solution, mentioned earlier, actually uses eDirectory to store its pointers and data and profits from that system's robust qualities. Other virtual directories, including the product from Radiant Logic, use their own database layers.

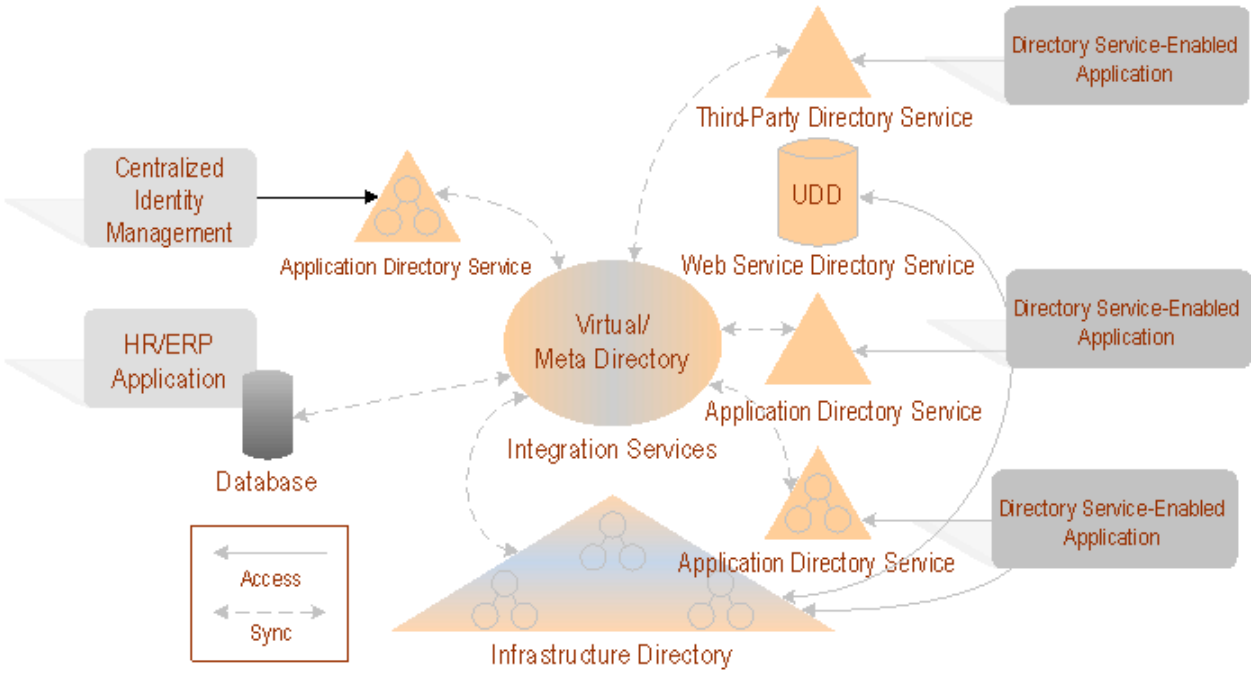

Figure 1.2: Virtual and meta directory services.

The Right System for Your Organization

Which of these options is right for your organization depends on your environment. If you can convince everyone in your organization (or, at least, those that control identity data) to store all of the identity information in an enterprise directory service, then that would, indeed, be the best choice. However, as this option is highly unlikely, you'll most likely have to choose between meta directories versus virtual directories.

Speed or Accuracy

First, decide on the dichotomy of speed versus accuracy. There are some applications for which speed is of the essence—such as verifying logins for a commercial Web site that hosts hundreds of thousands of customers per hour. However, some applications require absolute accuracy, such as verifying balances for a bank account during an ATM withdrawal. Most applications and services fall somewhere in between.

For applications and services that absolutely require speed or accuracy, make your choice accordingly. Otherwise, you might do just as well by avoiding a direct choice and instead looking at the connectors and drivers as well as the application programming interface (API) that the various vendors offer. Choose an API so that you and your team can create other drivers and connectors as needed. The system that provides the most connectors for the applications and services you use and offers an easy way for you (or a third-party) to create drivers as needed should be the one you choose.

Deciding where to store the rules and policies used by the joint engine of both a meta directory and a virtual directory once again raises the debate between RDBMSs and LDDSs. The arguments for this decision are somewhat different than those used when deciding on where to store the actual identity data. The rules and policies will be read many times more than their written, favoring an LDDS. But you can be more efficient with the triggers and stored procedures of an RDBMS. Don't get bogged down in this argument. Choose between a meta directory and virtual directory setup according to your needs and concerns about identity data. Pick your system, then use whatever it requires to store its rules and policies. Once again, don't expend "political capital" where you don't need to.

Project Planning Considerations

You're thinking about an Identity Management project, most likely user management or provisioning (or both). The next two chapters will focus on the specifics for each activity, but there are some general best practices that can help you properly start this project as well as ensure a successful road to an Identity Management—user management and provisioning—project. We'll explore these best practices in greater detail in the next two chapters.

In this chapter:

- Know why you're embarking on an Identity Management project
- Take a phased, modular approach
- Adhere to standards; monitor emerging standards
- Select the most appropriate directory strategy for your environment
- Implement best practices from other successful Identity Management projects

In Chapter 2:

- Define how you will measure ROI
- Take a phased, modular approach
- Choose an appropriate administrative approach
- Secure the user management process
- Ensure that all activity is logged and audited
- Consider Human Resources as the "trigger" for your system

In Chapter 3:

- Define how you will measure ROI
- Take a phased, modular approach
- Take a policy-based approach to provisioning that includes roles and rules
- Ensure that all activity is logged and audited
- Consider Human Resources as the "trigger" for your system
- Consider de-provisioning as important as provisioning—maybe more so

As you can see, there is overlap among the concepts covered in each chapter, as these considerations apply to more than one aspect of an effective user management and provisioning implementation. Let's take a look at the considerations that pertain to beginning as well as maintaining a successful Identity Management project.

Know Why You Are Embarking on an Identity Management Project

You are most likely not embarking on a project that could

- Affect your entire organization,
- Take weeks if not months or years to complete, and
- Costs quite a bit of money

simply because someone in your organization thinks Identity Management is the hot button topic of the moment. Most projects have one of three things providing the impetus:

- A quest to reduce costs through increased efficiency and automation
- A desire to improve security while maintaining as much user-friendliness as possible
- A need to comply with governmental or organizational requirements.

Let's examine each in turn.

Reduce Costs

It has been estimated that it costs a company $50 each time the Help desk has to reset a user's password. Other studies have indicated that the average midsized company with four to eight applications consuming identity information spends .83 hours of Help desk time per user per year managing passwords. For an organization with 10,000 users and an average Help desk support staff pay rate of $20 per hour, this translates into $166,000 annually for managing passwords at the Help desk—almost all of which could be eliminated through an Identity Management project to provide self-service password-reset for users.

📖 I'll explore this user management scenario in more detail in Chapter 2.

Improve Security

Consider the answers to the following questions:

- How many passwords do you have for all of the various applications and systems you access?
- What about your users, clients, and partners?
- Do you have a way to insure that users don't use their own names or the names of spouses, children, or pets as passwords?
- Do you require that passwords be changed periodically and that the new password be a major change from the old one (for example, the user can't replace "channel2" with "channel3")?

Walk around your building and see how many people have passwords written on notes taped to their monitors. Ask people whether they use the same passwords for multiple systems. These are all security shortcomings that a reasonable user management and provisioning project could overcome through the use of single sign-on (SSO) technology, password checking and verification, or the use of other authentication methods such as smartcards or biometrics. Any of these would greatly improve your security and could possible offer the added benefit of reducing costs.

> 🕮 I'll discuss improved security and reduced costs as a benefit of user management and provisioning in Chapters 2 and 3.

Comply with Regulations

If Sarbox, GLB, HIPAA, or CFR are more than just scrambled letters to you, an Identity Management project could be in your future. These mixed-up letters represent the Sarbanes-Oxley Act, Gramm-Leach-Bliley Act, Health Insurance Portability and Accountability Act, and The Food and Drug Administration's 21CFR Part 11 regulations—all of which either require Identity Management or are simply impossible to implement without Identity Management.

These regulations are just a few that apply within the United States—other countries, including the European Union, Canada, and Australia, require compliance to additional regulations. Check with your legal or compliance departments to determine which regulations are going to require a change in the way you manage identity information.

There is no reason why your project can't encompass two of these drivers or even all three, just be sure you understand the primary reason for your project and don't allow yourself to be sidetracked into non-productive sub-projects.

Take a Phased, Modular Approach

I mentioned earlier that your project might take weeks, months, or even years. Your team, your management, and your users can't be expected to maintain focus for that long. To ensure success, break the project into manageable steps with identifiable goals for each.

Trying to roll out a provisioning solution that provides users with absolutely everything they need on day one of their employment might one day be possible, but right now you would most likely get bogged down trying to enable one application or service that has little impact on the organization but is proving recalcitrant and unmanageable. Instead, select the applications and services most important to the organization and do them first. Alternatively, look for business functions you can automate first, such as password management and group management. Do them one at a time if necessary so that you can show steady progress and periodic milestones.

In this way, finishing a sub-project then becomes a goal in itself and allows for multiple success events. Quite a few people have managed to stir up interest in Identity Management projects by first creating a GAL, which is a relatively simple project (compared with provisioning) but one with high visibility.

Adhere to Standards and Monitor Emerging Standards

There is always a temptation with homegrown projects to create ad-hoc or shortcut solutions that work for your organization but aren't capable of being extrapolated to others. The use of standards might mean that you have to include extraneous code, policies, and even whole applications that aren't particularly useful to your organization simply to satisfy the standard.

However, Identity Management is becoming important to many large, influential organizations. Application and service vendors are listening, and they're enabling their applications and services to work with standards such as:

- Security Assertion Markup Language (SAML)
- Services Provisioning Markup Language (SPML)
- Liberty Alliance specifications
- Web Services Initiative specifications (WS-*)

Watch activities in the standards setting groups:

- OASIS
- Internet Engineering Task Force (IETF)
- World Wide Web Consortium (W3C)
- Open Group

Adopting these standards for your own work will actually save you time and effort in the long run. As more applications and services comply with these standards and specifications, your ad-hoc solutions may no longer work but those written to the specs will work even better.

Select the Most Appropriate Directory Strategy for Your Environment

Earlier in this chapter, we looked at the various issues surrounding LDAP versus SQL data storage for your directory as well as the pros and cons of meta directories and virtual directories. What is important here is not the specific choice you make—that choice will be dictated by the needs and requirements of your organization. Rather, you need to insure that your project takes advantage of the strengths of the chosen directory system while minimizing its drawbacks.

Implement Best Practices from Other Successful Identity Management Projects

Everyone thinks their own project is unique. Although such is true to some degree, most parts of your project have already been attempted by others. Rather than work in a vacuum:

- Ask your vendors to provide case studies and contacts at other organizations.
- Read about and talk to people who have already gone through what you're about to go through.

Provisioning is a fairly recent discipline, and there is not a lot of literature available on the topic yet. However, directory implementations, data migrations, and secure user management each have been practiced for a number of years (even decades, when it comes to migrations).

If you are taking a phased approach as I recommended earlier, get information about each stage and each sub-project you're undertaking. Implement first those that are well documented and that you feel comfortable undertaking. Also, be sure to document everything you do, not only to help others but also to ensure that by the end of the project, you're still able to recall why you made the decisions you did early on.

Summary

This chapter provided a basic understanding of what to look forward to in an Identity Management project designed around user management and provisioning. We started by tracing the history of user management from early mainframe computing through timeshare systems, local area networks (LANs), and up to today's seemingly always connected machines. We also found that provisioning as a concept is centuries old but has consistently been defined as providing all that is necessary for someone to do a task.

We then examined the reasons why, if user management and provisioning have such long histories, Identity Management projects are very slow in developing. We saw that technology wasn't the problem, but that social issues—what we referred to as "religious" and "political" issues—are the major reasons for the slow progress. Government regulation has been both a prod to get projects started as well as an impediment to getting them finished, especially when different regulations conflict.

We next took a look at the myriad choices available for directory services and systems. The directory (or, at least, a directory) lies at the heart of any Identity Management project. Making the right choice initially—the choice that works best for your organization, can save time, money, and vexation.

Finally, we identified best practices and considerations for a successful user management and provisioning project, then went into depth on those that apply to the beginning of a user management and provisioning project. Those best practices that are specific to user management will be discussed in Chapter 2, and the considerations specific to provisioning will be discussed in Chapter 3. A few will be discussed in both chapters because there is a difference in how the best practice is applied in each situation.

Coming up in Chapter 2, along with details of the best practices that I outlined for this chapter, we'll look at user management techniques and technologies, value propositions, and business drivers. We'll identify which technologies are currently available and which are on the horizon.

There are multiple methods of administering user management, and no one way is right in all instances. We'll examine the various administration approaches:

- Automated
- Delegated
- Self-service
- A combination of these approaches

The opportunities and challenges include:

- Security versus user-friendliness
- Monitoring versus auditing
- The role of TCO and ROI

There is also the question of whether simply outsourcing user management to a third party is right for your organization. Finally, we'll explore the available technologies and consider the role of standards as well as contrast the possibility of integrated suites of tools as against creating a "best of breed" solution. We'll wrap up with a gaze into the crystal ball to see what's beyond the horizon but close enough so that we can plan for it.

Chapter 2: User Management

Many of you might secretly hope that a chapter about user management will include subheadings such as "Cattle Prods" and "Holding Pens." In fact, user management really isn't about users very much at all. Rather, it's about the accounts of users on enterprise networks. For example, a discussion about a user's authorizations—the rights and privileges that the user has—is actually an exploration of the rights controlled by a particular account. Should the physical user log on to a different account, it's likely that the authorizations—the access privileges—would be different. Thus, although user management isn't a completely accurate term, it is embedded in the Identity Management discussion, so we will use it throughout this guide. Simply remember that it isn't the physical users we're talking about, but their digital presence on the network.

A Working Definition

User management is sometimes called user life cycle management, which is sometimes shortened to life cycle management. It is the existence of the user account on the network that is being defined. There are five aspects to this process:

- Creating an account for a user
- Giving the account the right access roles
- Replacing a forgotten password
- Moving a user from one location to another
- Retiring the account when it is no longer required

In this chapter, we'll take a look at each aspect in turn, then review the factors presented in Chapter 1 for a successful user management and provisioning project. These considerations should be followed to plan and implement a successful user management project.

📖 For more information about considerations for a successful user management and provisioning project, see Chapter 1.

Creating an Account

You might expect that the process of creating an account for a user is governed by the parameters of the service within which you are creating the account than by any general or "best practice" issues. However, although you cannot perform tasks that are forbidden by the application or service's rules, there is still a great deal of leeway within which you can work. Two major areas of flexibility are the formation of a username for the account and the rules for passwords.

Usernames

Almost all databases are indexed (that is, sorted) according to a field called the primary key, which needs to contain a value unique within that database. Many databases holding identity information, such as network directories like AD (Windows servers) or eDirectory (for example, NetWare servers), require usernames to be unique either for the entire directory or for the namespace within the directory (such as a domain, container, or organizational unit—OU) in which that user's data will be stored. You might think, then, that the username is also the primary key. In fact, the primary key is usually a number, which no one ever sees, that can be called the serial number, the User Identification (UID) number, or some other indicator. A unique name is required to avoid confusion—although it serves a useful purpose when the username is also the email name, which is required to be unique.

To make the account creation task easier, adopt a template for usernames. Two widely used templates are first initial plus last name (for example, John Doe would be known as jdoe) or first name/last name with a separator such as a dot or a dash (for example, John Doe would become john.doe or john-doe).

15 or 20 years ago, the custom was to use only the first seven letters of the last name along with the first initial (rendering Ruth Bresnahan as rbresnah) so that a home directory—what we now call a folder—could be created using the username while holding to the DOS limit of eight characters for a file or folder name. Such is no longer a requirement with modern Microsoft OSs and was never required for UNIX, Linux, and Macintosh systems. Still, some organizations cling to this template. If at all possible, create a template that employs all the letters of a person's name in the username. Doing so will speed searches and be easier for people to remember.

Collisions are caused when a template creates the same username for two or more people. John Smith, Jane Smith, and Joe Smith would all become jsmith in a first initial plus last name scenario. Assume that there will, at some point, be a collision, and that as the organization grows, the number of collisions will increase dramatically. Plan for this scenario by developing a secondary template to differentiate users. Perhaps Jane Smith become jsmith1 and Joe Smith become jsmith2. Alternatively, you might use middle initials (for example, jsmith, jasmith, jrsmith) or even entire middle names. Consider ease of use for the users involved but also minimize the confusion for others within and outside of the organization who might be attempting to find an email address.

> ☞ Modern network directory systems and email directories, also known as address books, often allow for a user's whole name and other information (for example, "Jane Smith [acctng]") to be shown and searched. This type of *display name*, which is an alias to the username, makes life easier for everyone.

Passwords

Passwords should be secret and should be known only to the user. However, forgotten passwords trigger more Help desk calls than any other single problem. Nevertheless, avoid the urge to maintain a list of passwords that would enable you to quickly look up a user's forgotten password. Many of the authentication systems in use today—both those included with servers and applications as well as third-party add-on products—allow you to specify rules for passwords that make the passwords hardened against attacks by intruders. Rules can be designed to cover some or all of these situations: length, character set, case, commonality, reuse, and familiar terms.

Password Length

On one hand, the longer the password, the more difficult it is for an intruder to guess or otherwise discover the password. On the other hand, the longer the password, the easier it is for the user to mistype or forget it. To balance these needs, require a minimum length of 6 to 12 characters. Sites with easy public access or with highly sensitive information should use longer passwords, of course.

Character Set

Almost all authentication systems allow the use of the 26 letters of the English alphabet. Most also allow the digits 0 through 9. Some allow certain punctuation or special marks such as dashes, underscores, addition signs, and exclamation points. Generally, the more characters allowed, the stronger the password will be. "Password" is an easily guessed password. But if you use a substitution method, such as the example method that Table 2.1 shows, to change a few characters, "password" becomes p@s5w0rd, which is much more difficult to guess.

Letter	Substitute
A	@
B	8
b	6
E	3
g	9
i	!
l	1
o	0
S	5
t	7
y	4
z	%

Table 2.1: Substitute numbers and symbols for letters to help obfuscate passwords.

Whether a familiar word spelled with unfamiliar characters will be more easily remembered by users than a random selection of familiar characters (for example, Ch1c@9o rather than dxfgjfgj) will vary depending on your users. Users who are familiar with texting—sending short text messages on cell phones and other wireless devices—will be more familiar with the use of odd spellings. You will need to weigh password ease of use against the improved security provided by difficult-to-remember passwords (and the level of security needed) to determine which password policy is best for your environment.

Case of Characters

Another option is to use a password in mixed case, such as "PasSwORd"; however, many systems ignore case. Given a choice, choose an authentication system (or an add-on) that differentiates between upper and lower case characters in passwords. Doing so immediately doubles the number of characters available for a password. And as we determine earlier, the more characters allowed, the stronger the password will be.

Common Words and Phrases

Add-on security packages for UNIX systems have long had the ability to block users from using common dictionary words (for example, dog, cat, breakfast) as passwords. This feature is gradually becoming available for more systems. You should look for an authentication system that provides this capability. A favorite ploy of intruders attempting to guess passwords is to use what's called a *dictionary attack*, which simply rotates through the words in a standard dictionary attempting to guess passwords. I'm not suggesting you attempt to block the use of the more than 500,000 words in the Oxford English Dictionary, which would, in any case, take far too long to do. However, blocking the 5000 or so most common words would be a big step forward.

> ☞ If your system allows mixed case as well as an extended character set, common words can be allowed if spelled using mixed case. You could block "cartoon" while allowing "cARtOoN."

Password Changing and Reuse

The longer a password is in use, the easier it becomes to guess. Require that users change their passwords periodically—at least every 90 days in a low-security environment and perhaps weekly in a high-security area. No matter how often they are required to change their passwords, you should also prohibit re-use of a password within a set period of time. Users who must change passwords monthly, for example, should not be allowed to repeat passwords within a 6-month period (or longer). Not all authentication services will allow you to set rules on reuse, but many do and the best of them will provide additional checks to ensure that users aren't subverting the rule (changing "password" to "password1", "password2", and so on).

Different systems have different rules; however, it is necessary to create and enforce one password policy that is consistent across the organization regardless of the system. Consider replacing systems that don't allow setting rules for passwords or, at least, acquiring add-ons that enable these systems to enforce password rules.

> 🕮 In Chapter 3, I'll explore Single Sign-On (SSO) and synchronized systems, which make it easier to enforce the same rules across multiple systems.

Familiar Terms

Users feel comfortable choosing easily remembered passwords. If you've blocked common words, users will fall back on names (spouses, kids, pets), words written on their equipment (Dell, Epson, and so on), and similar weak passwords. Although blocking the use of familiar terms is one of the most difficult rules to implement, these passwords offer very fertile ground for intruders. Blocking use of these words requires you to spend time gathering this information, but if effective security is your intent, that research is time well spent. As with previous password rules, you must reach a balance between secure passwords and those that are too complex for users to remember without writing them down on a sticky displayed on their monitors.

Granting Account Access

One of the most security-conscious changes to network accounts over the past 20 years is the move from allowing everything except that which was specifically prohibited to denying everything except what is explicitly permitted. The industry has gone from a situation in which all of the access points were open by default (and stayed that way until administrators remembered to close them) to having most access points closed by default. As Figure 2.1 illustrates, there are three principal ways of assigning access: by individual user, via group assignments, and through role-based access.

> ✎ Although discussions generally talk about granting access to the user in terms of the user account, the access is most likely controlled by an access control list (ACL) that is a property of the resource. Granting a user—or a group or a role—access actually means that you're adding the user—or group or role—to the ACL for that resource.

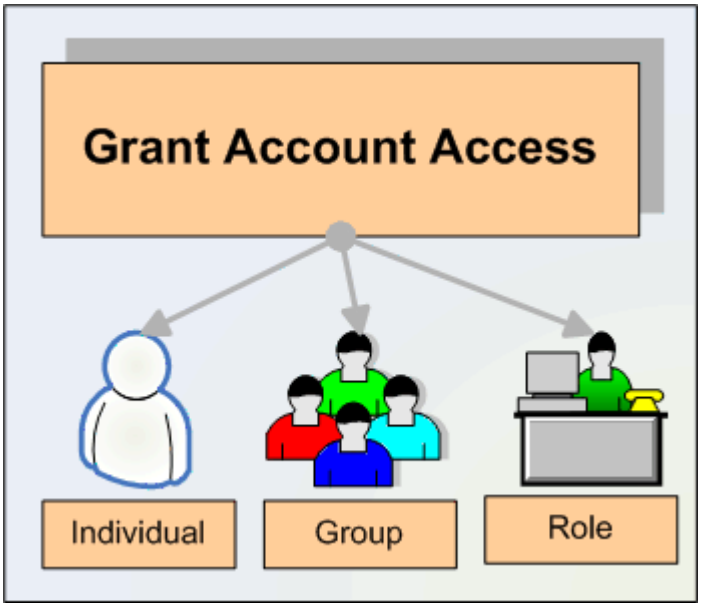

Figure 2.1: Account access can be granted to individual user accounts, based on group membership, or based on organizational role.

Individual Access

Individual access simply grants access on a user-by-user, resource-by-resource basis. It's tedious—just think of all the resources you have access to, then multiply by the total number of users to determine the number of access grants you would have to perform. It's seems even more tedious when compared with group-based and role-based access because any changes—such as the location of a resource—require you to make changes to every user account needing access to that resource. Because the access is actually a property of the resource, you might need to search through all resources to find every instance of that user. Thus, for authentication systems that offer group-based and role-based access controls, always use these methods rather than granting individual access—except for those items (folders, files, personal desktop computers, directly attached printers or scanners, and so on) that are primarily or solely used by the individual user.

Group Access

If your authentication system allows it, assign users to groups based on a shared need to access a particular resource in the same way. A simple example is a departmental printer: create a group that includes everyone in the department, then grant the group access to the printer. Should any changes be needed, such as a new driver, a new location for the printer, or a new model—one change to the group configuration should be all you need to ensure that everyone still has access. Similarly, applications, such as word processors, that are used by a large number of users can be best administered by using groups to grant access. You can then update the application, move it, or even replace it with only a minimal amount of administrative activity. Using groups also eases the administrative burden when adding users to or removing them from your network: simply putting them into—or removing them from—the requisite groups eliminates the need to remember every right and privilege that needs to be granted (or revoked).

A few authentication systems now support the concept of *dynamic groups*. A dynamic group contains those users that have a common value for a particular attribute. For example, everyone who has the value Accounting for the attribute Department is a member of the ACCT-DYN group. As with a static group, a dynamic group can be assigned access rights. The difference between a static and dynamic group is that when a user attempts to access a resource, the authorization system checks the attribute for the required value and only grants access if the proper value is found. Thus, if you change the user's Department attribute to Marketing, and the user will no longer be able to access resources as part of ACCT-DYN.

Role-Based Access

Granting access based on roles is similar to using groups to control authorization; for systems that don't support the concept of roles, groups can serve a similar purpose. The difference between a role and a group is not great. Usually, a group is organized around a particular resource (for example, a word processor group), business organization (the marketing group), or activity (the carpool group), while a role is typically based on job function (buyer, administrative assistant, Help desk technician). You would rarely set up a group for less than three or four users because the overhead of group creation and maintenance wouldn't save you any time or effort. However, a role-based account is useful even if no one is in the role at a given time. For example, you could create a role called VP of Marketing so that you could rapidly move a new user in (or an existing user out) of all of the access the user would need to do the job.

Most authentication systems that understand the concept of "role" allow you to create one using much the same process as creating a user account. You can't put in location data or personal information (birth date, home address, and so on), but you can indicate groups that the role should be a part of and any access rights the role should automatically acquire.

Which Access is Correct?

Different systems (network OSs, desktop OSs, database applications, Web services) offer a range of rights and privileges. Traditional UNIX systems offer only Read, Write and Execute, while Novell's NetWare allows a rich mix including Supervisor, Read, Write, Create, Erase, Modify, File Scan, and Access Control. Also with NetWare, each of these rights differs in effect depending on whether they refer to a file or a folder. The system you're working with may support as few rights as UNIX supports, as many as NetWare, or somewhere in between. It should be self-evident that a user should have full access to his or her data files, which are stored in folders assigned by the administrator to only that user.

Shared folders, however, present a problem. If all users of the shared folder have full access, one user's file will often overwrite another user's file of the same name. Granting, for example, only the Create, File Scan, and Read rights allows each user to read and place files in the shared folder without overwriting another user's files. Unfortunately, this setup doesn't allow the creator of the file (or anyone else without more rights) to modify the contents or remove a file. One solution to this problem is to grant one user Supervisor rights to the folder and have that user perform all maintenance. Another solution is to employ a *versioning* (also called *librarying*) application that allows users to create, register, and store new documents and read existing ones while allowing the document's owner (or creator) to modify or remove it. In effect, the versioning package acts as the Supervisory user.

Applications deserve their own consideration when granting user rights. After all, applications are only files that have a special purpose stored on the system. In general, you should group applications together in a folder structure away from data files. Grant users only the Read and File Scan rights to the applications (and the Execute right for those systems that require or support it). No ordinary user should need to modify or remove an application file; that is something that viruses, worms, and Trojan Horses try to do. Good security practice demands not allowing your users to inadvertently cause damage or destruction to your systems. For more information about your authentication system's available user rights and their security implications, consult an administrator's guide for the system(s) that you are supporting.

Forgotten Passwords

As I mentioned earlier, Help desk staff spends more time resetting passwords than on any other activity. It's been estimated by the GIGA Group that password resets cost a typical organization $105 per user per year. If you have 1000 users, you're spending more than $100,000 per year on password resets!

One option for saving almost all of this expense is self-service password reset; however, this feature is not yet available for every system. Self-service password reset allows users who forget their passwords to reset the passwords through a Web service from any browser available to them. Typically, the user must answer three to five questions about personal information that should only be known to that user. Elementary systems have stock questions, such as Where were you born?, for which the user needs to supply answers. The best systems also allow the user to create the security questions (for example, What was your nickname in first grade?).

Of course, the typical user has more than one password to remember. There is the network password, the workstation password, the database password (or passwords for multiple applications), all the Intranet Web site passwords, and so on. To address this potential password-reset problem, in Chapter 3, we'll explore SSO services, which hide the multiple passwords and enhance the value of self-service reset. It isn't necessary to implement a full-featured electronic provisioning solution to implement SSO. In fact, many organizations install SSO as the first step towards an electronic provisioning implementation.

Moving a User from One Location to Another

Some vendors and experts refer to user management as life cycle maintenance. This term reflects the fact that creating an account is far from the end of the work you'll need to do for that user. As people move around the organization, both literally (to a different office, building, city, country) and figuratively (through promotion and transfer), their user accounts will need to be modified. There are three major topics in this area: change in responsibility, change in location, and change in organization. The latter refers to mergers and divestitures and is more of a function of the provisioning technology that we'll delve into in Chapter 3.We'll explore change in responsibility and change in location in the following sections.

Change in Responsibility

Everyone hopes to work their way up the corporate ladder, and even with the high turnover that many organizations face today, there are many people who remain with an enterprise for 10, 20, and even 30 years. Over that time, users' duties and responsibilities will change. The corporate resources they need to access will also change. Your job will be to ensure that users have the access they need to efficiently perform their jobs as well as to see that access they no longer need is removed. This function is often overlooked, but as we become more security conscious, it grows in importance.

Traditionally, the way these changes were handled would be for a user to request access to resources needed for a new position. After getting approval from the user's manager, the resource administrator would grant the access. Invariably, if the administrator attempted to remove access from resources the user should no longer need, the user would claim that a transition period was necessary to finish work with the old resources and train a replacement in working with those resources. The overworked administrator would hope to remember to remove the access in a few week's time, but often wouldn't. Even in the rare instance in which the administrator did remember, *all* of the unneeded rights were very rarely removed.

Role-based access controls can be a real boon to the administrator faced with users who change corporate responsibilities. The use of roles saves the administrator from hunting and pecking to find the correct privileges that the user needs for a new position as well as those that should be removed as a result of the move from the old position.

The transition period can be handled by modern tools that allow you to set an expiration data on the user's use of the role. It's also much easier to spot inappropriate role assignments for a user than it is to spot inappropriate rights assignments to folders, files, and other resources during periodic auditing of your system.

📖 We'll look at auditing needs towards the end of this chapter.

Change in Location

Users also seem to move around a lot—from one cubicle to another, to a different office, a different floor, a different building, a different city, state, or country. The users in Australia shouldn't be accessing the word processor on a server in Switzerland, nor should the users in San Francisco print on a printer in Chicago. Group-based access control can preserve your sanity in dealing with user location changes.

A large part of group assignment is based on geography. All the users on the third floor of building 27 on the South San Francisco campus use the same printer. A group called, perhaps, SSF27-3pr would handle all the rights needed to use that printer as well as be useful in distributing drivers and online manuals for that printer. If a user moves to the fifth floor of the building at 227 State Street in Chicago, you would simply move the user from the SSF27-3pr group to one perhaps called CHI227SS-5pr. You could just as easily call the groups PR-SSF27-3 and PR- CHI227SS-5 if the resource, rather than the location, is the primary way you want to sort the groups.

Automating Change

Using roles and groups certainly speeds the application of changes; scripting will automate the process even more. Imagine clicking on a user object, then dragging it to a different location. Based on the user's roles and groups, the service automatically makes the necessary changes so that not only is the user productive from his or her first day in the new position or location but also the no-longer-needed access rights and privileges are removed.

📖 We'll explore this type of functionality in more detail in Chapter 3.

Retiring the Account When it Is No Longer Required

I've mentioned that it's important to remove no-longer-needed access rights when a user moves to a different position or location. But even more important, from a security perspective, is the need to remove all access rights from a user who has left the organization.

A Cautionary Tale

In the spring of 1993, I walked into the monthly staff meeting at the company (a manufacturer of computer products) of which I was IT director. About halfway through the meeting, I realized that the person who usually represented the design department wasn't there. When I asked about him later, I was told that he had left the company 3 weeks earlier and was now working for a major competitor. This person had access to all of our proprietary design information and, as a senior staff member, had dial-in access to the network. I immediately disabled his account and sent out memos warning everyone about the danger of not disabling the accounts of terminated staff. Two years later, when I left the company, my account (which had Supervisory access to all of the company's resources) remained active for a month—until I phoned my successor and told him to disable it!

You should periodically—on a weekly basis, for example—review the login/logout logs for any anomalies: people not logging on at all or only accessing resources at odd times or from non-typical workstations or locations. Investigate the anomalies to be sure they aren't simply a result of someone on vacation, temporary assignment, or sick leave, but be ready to disable the account if you don't discover a satisfactory reason for the changed behavior.

Better still, cultivate the Human Resources department. Create an automatic tool that performs the function or formalize a way for Human Resources staff to inform you when someone is:

- On leave
- In legal dispute
- No longer with the company

Many of the Human Resources applications used today are built on standard relational database platforms that can be programmed to take an action whenever the contents of a data field changes. Most have a data field for "last day" or "final day" or "employee status" or an alternative that indicates that the employee has left the company. If at all possible, have a change in that field trigger a message to the appropriate administrator (or administrator's group) to remove access rights for that user.

Rather than remove the individual, group, or role-based rights, however, the preferred action is to disable the account. It's possible that at some future time, someone will need to know what the user had access to, which files and folders were created by that user, and which resources the user accessed and when. If the user account is deleted, this information might be lost.

However, even removing the access rights (as well as removing the user from groups and roles) can lead to a loss of information because you can never be sure which resources the user might have been able to reach. The increasing government interest in and regulation of digital data and access—such as the Health Insurance Portability and Accountability Act (HIPAA)—often requires that you be able to produce records showing when, where, and by whom a bit of data was created, retrieved, modified, or removed. Disabling the account and logging the data and time you do so should protect you and your organization if legal proceedings were to ensue (see Figure 2.2).

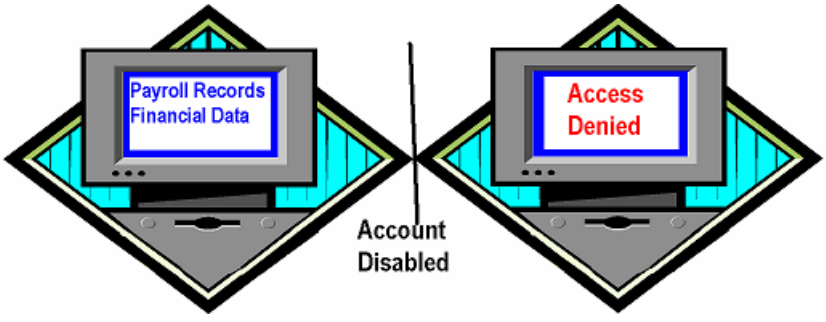

Figure 2.2: With many authentication systems, the user is immediately denied access when the account is disabled.

Considerations for a Successful User Management Project

In Chapter 1, I introduced considerations for a successful user management and provisioning project. In the following sections, we'll explore which of these factors apply specifically to a user management project:

- Define how you will measure ROI
- Take a phased, modular approach
- Choose an appropriate administrative approach
- Secure the user management process
- Ensure that all activity is logged and audited
- Consider Human Resources as the "trigger" for your system

Define How You Will Measure ROI

According to the Gartner Group, "A broad range of factors—including the demands of enterprise resource planning implementation, regulatory compliance issues, and the pressure to contain costs—are intensifying the focus on how enterprises manage the processes associated with granting users access to business information. Identity and access management solutions, which can offer 3-year return on investment in the triple-digit-percent range, are becoming essential tools for effective management of user account and access rights information across heterogeneous IT environments, for Web and non-Web applications" (Source: "ROI Drives Identity and Access Management Implementation," Gartner Group). But how will you measure ROI?

Reduction in Help desk calls—through self-service password reset, for example—is quantifiable. So too is the reduction in administrator time spent in assigning access rights when group-based and role-based access is implemented. Determining the ROI on security improvements, however, is more problematic. The following list highlights suggestions for doing so:

- Perform a manual audit of accounts and access rights, then tally the number that are in need of change or removal. Calculate the amount of time necessary to perform the changes one at a time compared with using roles and groups.
- Carry out a thorough risk assessment (or hire a security consultant to do so) that includes costs for damage and repairs. Perform both a before and after scenario that highlights the changes your user management project will implement.
- Remember that a security breach can have a multiplier effect: the costs of a breach include the costs of validating the integrity of other data—data that is based or relies on the data that has been compromised, the cost of adding security, and the loss of reputation.

Take a Phased, Modular Approach

In the first chapter, I mentioned that a phased approach is a good practice for any Identity Management project. The basic idea is that your total project might take months and even years to implement, but people want to see results in a shorter period of time. By breaking the total project into smaller sub-projects and implementing them one at a time over time, you can keep the implementation team's interest from flagging and provide opportunities for non-team members to be aware of and re-commit their approval of the entire project.

For example, implementing self-service password reset could be fairly easily accomplished as long as the authentication system you use supports it either directly or through third-party services. This implementation will be visible to almost everyone in the organization. In addition, it is a sub-project that has an easily demonstrable ROI that results from the reduction in Help desk calls.

Creating an automated notification system from the Human Resources software's database to your Administrators Group whenever someone leaves the company is a project that won't be visible to most people but is a sure winner with upper management—especially those most concerned with security. Depending on the Human Resources system you use as well as your in-house programming expertise, this task could also be an easy sub-project to implement.

There's also the sub-project I suggested in Chapter 1: a GAL. If your identity storage system allows you to access its data through LDAP, there are many applications and services available (some at no cost) that will allow your users to query the name, location, phone number, and email address of others in the system (provided, of course, that you have that data in the system). This implementation has proven time and again to be a big hit with most users and can usually be implemented in less than a day at little or no cost. This type of initiative is ideal in helping you to sell the larger user management project.

Choose an Appropriate Administrative Approach

Although moving to group- and role-based access methods will reduce the amount of time you and other administrators must spend creating, maintaining, and removing user accounts, managing users still remains one of the least desirable activities in your job description.

At one time, user administration was an all or nothing activity. Either you had full control of the user object or you had none. Today's authentication, authorization, and user management tools allow for much greater granularity of control. It is possible in most systems, for example, to create a role of Password Administrator—a person who can reset a user's password but otherwise has no control over the user's account or access privileges. Of course, the ultimate delegation is self-service password reset—you delegate the administration to the user! Almost all of the duties associated with user management can now be delegated along with controls to ensure that the management is not abused nor causes unwanted security or privacy risks.

The ideal situation is to delegate as much as possible to automated services. In a recent report, the Burton Group noted that "…manually administered environments require a Full Time Equivalent (FTE) for approximately every 500 to 1,000 users, while automated environments can manage 5,000 or more users per administrative FTE" (Source: "User Management," the Burton Group). A reduction in administrative costs by a factor of 10 to 1 could certainly improve the ROI of your project.

Also consider moving parts of the process to non-IT departments. If your identity system supports rules-based administration so that you can implement rules governing the creation of user accounts, for example, you should be able to have the Human Resources department, through the action of their Human Resources software, initiate the creation of user accounts. Although this type of delegation is more prevalent in provisioning services (see Chapter 3), it could be done by having the Human Resources database start a process that both creates the user account and notifies an IT administrator who can then tweak or modify that account. Delegation of these responsibilities doesn't decrease your control; it frees your time to do those parts of your job that you might consider more interesting.

Secure the User Management Process

I encourage you to delegate as much of the user management process as possible, but I must also urge you to ensure the security and integrity of the process at the same time. The more people who have access to a resource either as users or administrators, the greater the risk that the resource will be compromised in some way (either purposely or accidentally). There are three very important aspects to the security implications of your user management project:

- Restricting and authenticating access to the user management system
- Restricting administrators' purview to only those users, groups, and objects for which they are responsible
- Automating and simplifying the process of user management

Restricting and Authenticating Access to the User Management System

In the previous section, I encouraged you to enlist as many people as possible as delegated administrators in order to ease the burden on you and your immediate staff. This step doesn't contradict that recommendation, but it is important to consider in the light of delegated administration. You'll need to institute rules governing the use of the user management tools that restrict them to only authorized users. To enforce those rules, you'll need to be vigilant and audit all access to the user management system. Any breach of the rules should be dealt with quickly and appropriately. Anyone deliberately misusing the system should at least have those privileges revoked. If there is evidence of a security breach, the offending person should be subject to immediate dismissal—a chain of events that should be included in a corporate policy. The security and integrity of your user management system should be paramount.

Restricting Administrators Access

People are curious. People will look at, probe, poke, and manipulate anything they can get their hands (or eyes) on. Telling delegated administrators that although they can easily manipulate a user's access privileges that doing so is forbidden, might encourage them to give in to the temptation to see what might happen. Fortunately, most identity data repositories have rights and privileges associated with the objects they contain.

Just as you can assign rights to folders and files, you can also assign rights to the identity system's objects, which also have access control lists (ACLs). It is this function of having an ACL for each object and property that allows you to create the Password Administrators I suggested in the delegation section earlier.

If your authentication system allows, create delegated administrators for particular functions (account creation, printer administration, password reset, and so on). Be sure that these administrators have access only to the objects they need to see and have the ability to make only the modifications you want them to be able to do.

Automating and Simplifying the Process of User Management

Many of the problems that occur are not caused by someone with malicious intent but by an administrator who isn't giving full attention to the activity he or she is involved with. The creation and maintenance of user accounts is generally a tedious operation. Delegating this task doesn't make it less tedious for those who are performing the process—mistakes will happen. Automation is a solution.

Scripted processes that are initiated by a change—for example, in the Human Resources' database—are the best way to mitigate the problems of administrator error. Of course, if the Human Resources clerk spells a new employee's name wrong or wrongfully indicates someone is being terminated, the result might be embarrassment but rarely will it engender a security breach.

Where you can't fully automate a process, try to eliminate as many data entry steps as possible by having the application or service read that data from an existing computer-based source. If your administrators will need to handle user data in a number of services or applications, try to minimize the differences between the administrative tools they will need to use. The goal is to eliminate the avoidable mistakes that an administrator might make.

Ensure that all Activity is Logged and Audited

Perhaps the most important consideration: whenever the user identity storage is accessed; whenever a user account is created, modified, or removed; and whenever a user's access is granted or revoked an entry should be automatically made in a log file. The log file itself should only be able to be modified (or removed) by an auditor who is not the same person as the administrator whose activities are being logged. Ideally, the auditor will be someone with no connection to the user management process.

Auditing is now required—or at least strongly recommended—by many, if not all, of the recent government-imposed regulatory environments (such as HIPAA and the Sarbanes-Oxley Act). The auditing process itself should be under the strictest security controls while following the recommendation to automate and simplify the process. Most modern user management services allow for a completely automated and transparent audit process. You should implement it immediately when you begin using the service. For older systems, there are third-party auditing packages that can help you ensure that full audit trails are available for all user management activities.

I encouraged you to delegate administration to others and to automated processes as much as possible. Frequently, the ability to make the changes you've delegated rests not with the individual user to whom you've given the administrative role, but to a service that performs the actual task. The delegation simply involves giving the user access rights to the service. When the activity is logged, then, it is the service that is identified as doing the activity and not the user who is running the service. If your delegation works in this manner, be sure to associate the user management functions performed by the service with the identification information for the user who is running the service at that particular time.

Thoroughly test in a non-production environment the auditing functions and capabilities of the system you are using. The time to discover problems is before going into production, not after you've received a subpoena from court.

Consider Human Resources as the "Trigger" for Your System

In the section on delegating administration, I suggested that you cultivate contacts in the Human Resources department and that, where possible, you automate the user management function through the use of the Human Resources database and applications. Doing so will almost be a requirement when you want to move to a full provisioning project (as we will discuss in Chapter 3) and is very desirable for your user management project. By having the user account creation, maintenance, and removal initiated by activity within the Human Resources database, you go a long way towards delegating administration as well as automating and simplifying the process (and, as I mentioned earlier, preventing manual mistakes).

The best way to enlist the help and buy-in of the Human Resources department depends on your environment—office politics often play a significant role in IT implementations. To aid in the justification of this type of implementation, highlight how the process will minimize the work that must be done by the Human Resources staff—either through automation or simplification. If, for example, the trigger you place in the Human Resources database to initiate creation of the user account could also notify the payroll department—with, say, the user name, social security number, employee number, and pay rate—will save a step for the Human Resources staff, your project will be looked upon favorably (see Figure 2.3).

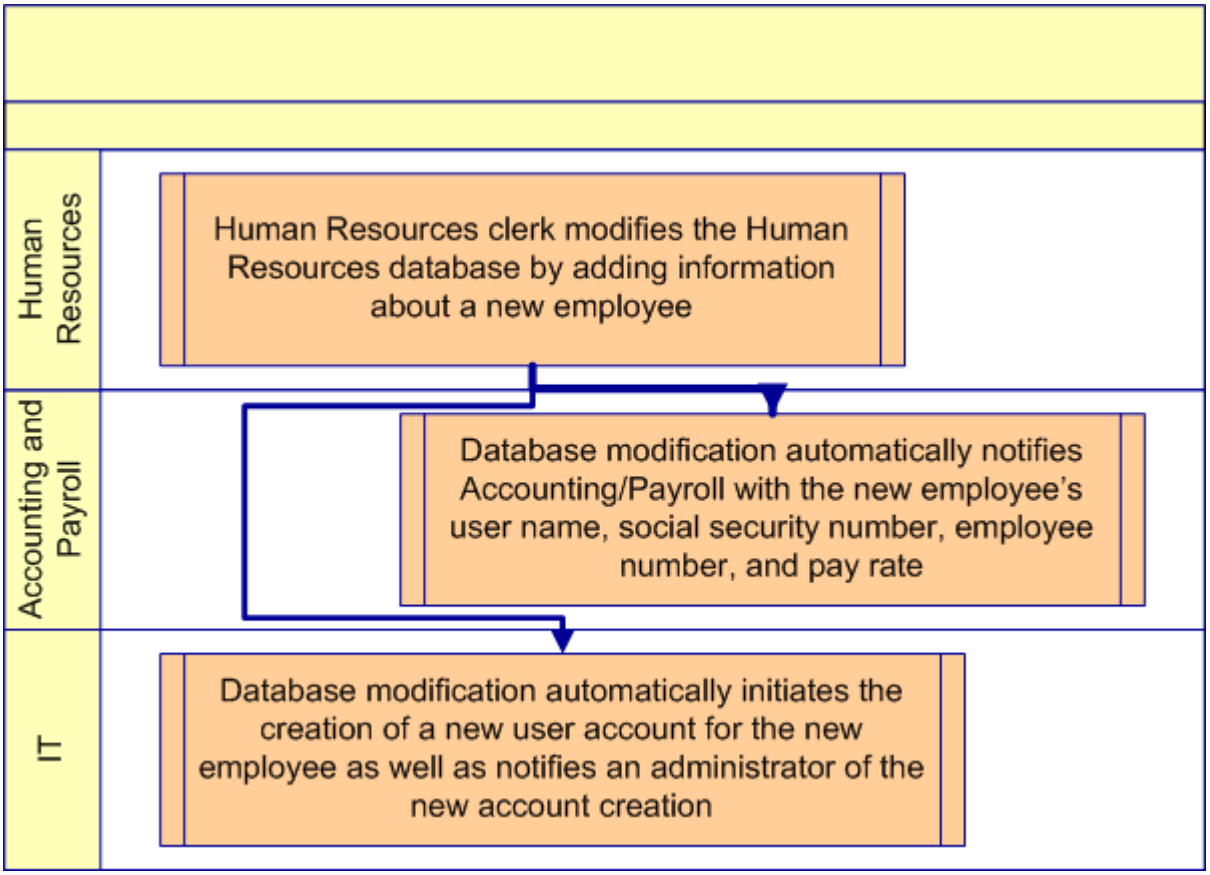

Figure 2.3: Implementing an automated Human Resources "trigger" eases user management as well as the workload of the Human Resources staff.

Outsourcing

The ultimate in delegated administration is to move the entire process out of your company to some third-party resource that will handle all of the tedious details for you. As with just about every choice you must make, there are pros and cons to be considered.

Small organizations—certainly those that employ less than 100 users (but more than 20 users) and those that employ as many as 500 users—should seriously consider outsourcing. These are organizations with only a one- or two-person IT department (when the entire department isn't outsourced) who must be generalists rather than specialists in security or user management. Care must be taken by small organizations in regulated industries—healthcare, securities, banking, and law— to ensure that all required safeguards are taken to protect the security and integrity of the user data even if the user management is outsourced.

Larger organizations might well have all the required expertise on hand, but might choose to outsource parts of the user management process. Password resets as part of an outsourced Help desk are one example of outsourcing by service. Alternatively, you might outsource by geography. It might be cost advantageous for smaller, more remote or isolated offices to have their users managed independently by a third-party organization. When dealing with multi-national organizations, the expertise of a local outsourcing company might be very useful in determining the correct procedures to comply with local government requirements. Privacy requirements, for example, vary widely by country but need to be enforced for the workers in that country. Using a user management provider in that location could be necessary in order to be sure of compliance.

No matter what reason you have to choose a third-party organization to provide outsourced user management, you need to have detailed, binding, written service agreements in place. When a user needs access but the user management provider isn't answering the phone, it is you and the IT department that will be held accountable. There is nothing inherently wrong with using a small organization for your outsourcing partner—it is, after all, specialized knowledge that you're licensing—as long as you have guarantees that they will do the job in an accurate and timely manner.

Summary

A user management project can be a tremendous asset to both your IT organization and to the enterprise as a whole. Improved security, better access control, lower cost of account maintenance, and greater user satisfaction can go a long way towards improving the image of the IT department.

In this chapter, we explored all aspects of user management categorized by the basic aspects of the user account life cycle:

- Creating an account
- Granting the account the correct access
- Resetting forgotten passwords
- Moving a user and maintaining the account
- Retiring the account when it is no longer required

We've also seen which of the successful user management and provisioning project considerations apply specifically to user management and examined the best practices they suggest.

In the next chapter, we'll take user management, which we've been looking at in a system-by-system method, and extrapolate it to the entire organization. Electronic provisioning, in theory, allows you to automate every aspect of user management and the user account life cycle.

Chapter 3: Applying the Technology: The Details

When you stop to consider the scope of what provisioning applications are designed to do, you can fully appreciate why deploying one is neither simple nor easy. By implementing provisioning, you are trying to automate IT processes via corporate policy that will ensure that the people who are using your enterprise resources have all the necessary access to systems and information—and, yet, have only the type of access that they are supposed to have.

In addition, this capability must work both ways—automatically providing the user accounts and access rights where appropriate and automatically disabling the same user accounts and access rights once they are no longer needed. Yet, the provisioning process, while rooted in IT operations, extends into enterprise operations beyond managing users and the information they access.

A Provisioning Approach to Identity Information Management

Automating the creation of user accounts and assignment of the necessary access—and only the necessary access—is at the core of provisioning, yet provisioning is about much more than just user accounts and passwords. In many aspects, provisioning also functions to help manage the security of information within the enterprise while controlling access to enterprise information resources.

One of the most common uses of provisioning technology is in the so-called hire-fire scenario, in which the provisioning application is responsible for establishing user accounts and access rights within the enterprise network operating systems (OSs), messaging applications, essential databases, customer relationship management applications, and Line-Of-Business (LOB) applications. As Figure 3.1 illustrates, the hire-and-fire scenario use of a provisioning application supplies a company with a critical capability—the automated addition of a user to all enterprise applications and IT systems upon hiring, and the automatic disabling of that same user account (and all of the user's access rights and permissions) at the point that the user is no longer with the company.

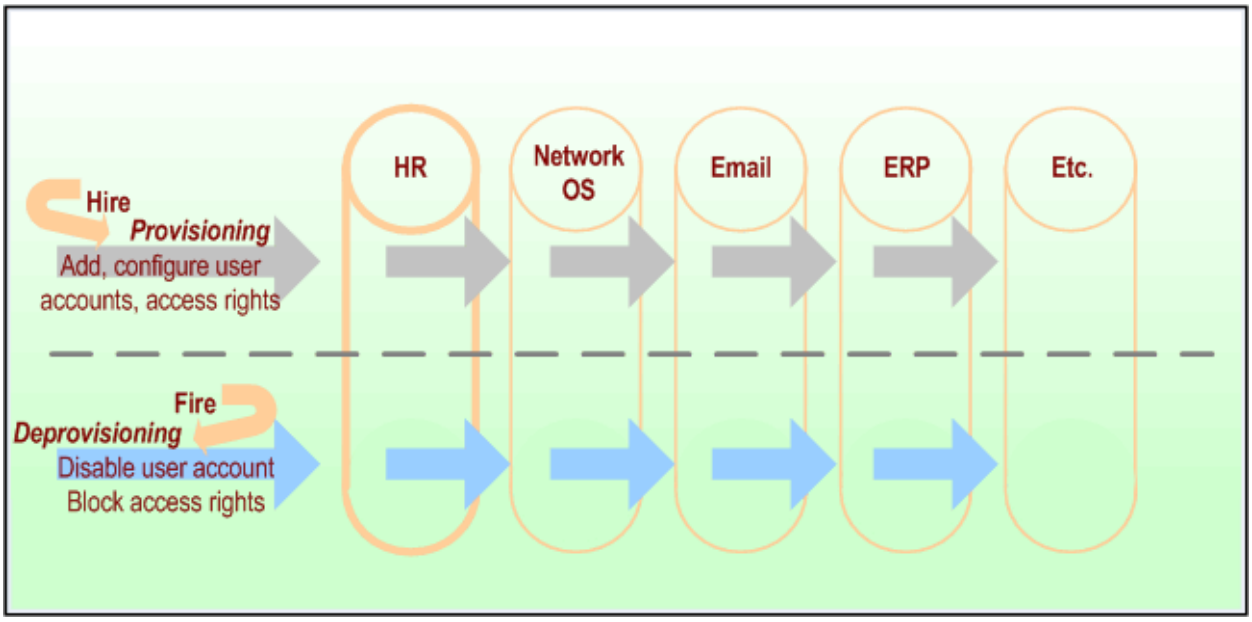

Figure 3.1: The process of provisioning and deprovisioning spans core identity information repositories.

Providing user accounts with the correct access rights and privileges is not only a security concern but also a serious cost factor—for many companies, this process can take days (or even weeks), can involve the work of multiple administrators, and slows the assimilation of a new employee into the company's workflow. Additionally, there are logistical issues that can add to costs and impede employee assimilation—for example, scheduling meetings with the right managers or administrators (or catching them at their desks) to gain needed approval and provisioning actions.

To a considerable degree, the information and resources involved in the provisioning process and network or application security is the same—user accounts and the related permissions that grant access rights to enterprise information, services, applications, or other network resources are common ground for provisioning applications and security subsystems. Comprehensive provisioning applications might extend beyond the scope of IT security by initiating business processes that facilitate requisitioning of physical materials or facilities (such as an office, desk, chair, telephone, pager). However, much of the core function of automated provisioning is to provide the new employee with access to the enterprise networks and core business applications.

As stated earlier, implementing provisioning is neither an easy nor simple process; rather, it will take significant time, effort, and cost to implement. Despite these efforts and time and monetary costs, provisioning can provide substantial savings over time in the creation and management of user accounts and identity information throughout your enterprise.

Policies Are Essential to Provisioning

Policy-based provisioning allows you to establish sets of rules related to the business roles that new employees will be taking on. For example, although all employees need user accounts within enterprise networks as well as access to common applications, only employees who will be working from remote locations (branch offices, home, and so on) need accounts on remote access servers and the ability to access enterprise resources from locations outside your organization's firewalls.

Provisioning systems assess the business roles and related rules in order to establish profiles for employees. These systems also determine the nature and type of services and access to resources to be allocated and the equipment and facilities needed to support the work that the employee is expected to do. Establishing a set of policies for provisioning will provide a common foundation for administrators, management tools, network OSs, applications, and users, as well as facilitate the standardization of the configurations for user environments.

In fact, a substantial part of the time and effort of implementing automated provisioning will be taken up in establishing the business and security policies that govern user accounts and access rights. This process requires discussion, negotiation, and (with any luck) agreements among business managers throughout the various divisions controlling the identity information repositories and business applications in your enterprise.

Although this process won't be easy, it will be worth the difficulty. Once you've established the policies by which classes of users gain access to information and resources, you will no longer need to discuss it on a per-user basis—the policy will be established and it will be automatically implemented.

There might be exceptions to the policies, so be sure to establish a process for dealing with them. A manual review by administrators knowledgeable of the employee's role in the organization can help to track the handling of policy exceptions as well as errors in initial assignment. How a new employee is assigned organizational roles by policy might not always be exact, and placing them into roles that are the "closest match" might not entirely fit. As a result, it can be useful to have an administrator verify that the assigned rights and appropriate assets—and *only* those rights and assets—have been granted.

Provisioning Software Needs to Be Robust

A provisioning/deprovisioning solution at its core automates the identity information management for the people involved in your business across multiple applications, platforms, and operations. At minimum, it must provide unified control over the user identity information; thus, of necessity, it must have access to most, if not all, the identify information repositories in the enterprise.

Provisioning systems need to be self-correcting to some degree—they should be able to identify problems, redundancies, and inconsistencies in the provisioning process, and have some configurable mechanism for automatically correcting these errors when possible. When an activity requires a person to take action, the notification process should be automated. If an application or database server is down during a provisioning operation, for example, the user account information will not be able to be added. The provisioning software should generate a notification of this failure, and provide some direction for how to resolve it.

You can more easily manage the security and operational concerns if you establish policies that require that all additions are routed through your provisioning application (perhaps as triggered by entries in the Human Resources—HR—database), thus establishing consistent policy-based control over accounts and access.

Yet, to some degree, provisioning systems need to be aware of and responsive to the information that is put into the applications and services that they will be manipulating. If a new email account is directly added to the email server, for example, the provisioning system should be able to cross-reference this email account with user accounts in the network OS and an employee entry in the HR database, and provide some kind of integration between these systems. Although this form of "reverse synchronization" is very useful, it can be difficult to implement across all possible applications in use in your enterprise.

Communicating with Enterprise Data Stores

Provisioning applications need to be able to communicate with many types of data sources, especially directories, databases, and other identity information repositories. To do so, they must support common industry protocols and methods for accessing these data stores:

- Lightweight Directory Access Protocol (LDAP)—Because LDAP is the de facto standard for most directory service (and many email) products, selecting provisioning software that can communicate via LDAP with all directories within your network environment seems necessary.
- Structured query language (SQL)—Similarly, the ability for the provisioning application to use SQL to communicate with the wide range of databases (such as Oracle, MySQL, Microsoft SQL Server, and others) is also a central requirement (many HR, Customer Relationship Management—CRM, and Enterprise Resource Planning—ERP—applications use SQL). A provisioning application should support either the Open Database Connectivity (ODBC) interface or the Java Database Connectivity interface to allow programmatic access to SQL-based databases.
- Extensible markup language (XML)—Support for the import and export of identity information via XML is important (perhaps even critical) to the integration with your current and future enterprise applications.

Provisioning Connects to Many Identity Information Stores

In addition to integrating with an underlying directory service (enterprise, meta, or virtual), provisioning applications must be able to connect to all of the repositories of identity information within your enterprise. To do so, specialized namespace connectors (aka drivers, adapters) must either exist as part of the provisioning application or the application must provide a means for the connectors to be easily developed to meet your specific requirements.

Provisioning applications typically provide connectors for common forms of information exchange between services or applications, such as connectors for SQL to exchange information with databases (and the HR, CRM, ERP applications that use them), LDAP to exchange information with directories (such as network OSs and email applications), and XML to exchange information with most of the newer enterprise applications. In addition to these standardized connectors, it is common to need to communicate with specialized or proprietary applications in enterprise identity management and integration projects; thus, most provisioning systems will support some way to customize connectors. If your organization has legacy applications with data stores that are inaccessible via standard connectors, you can use third-party products such as Novell's DirXML, which supports creation of custom connectors to exchange information with such proprietary data sources.

Choosing a Path

What is the right provisioning solution? The one that works best for your business, of course. One of the aspects of all systems is described as *equifinality*, which asserts that within any system, there are many paths that can lead to the same outcome. Such is also true with provisioning options—you can approach it several ways.

The desired outcome is automated provisioning and deprovisioning of user accounts, information, and resources in a way that is seamlessly integrated with your existing IT infrastructure and business operations. There is more than one right way to do so; however, within the context of your current business and technology environments, certain approaches will be more suitable than others.

Selecting your path to an identity information management solution that uses automated provisioning requires the assessment of the underlying platform that will host the user management data store as well as the specifics of the provisioning application itself. The underlying directory technology and specific provisioning application that you select must be rooted in what exactly you need it to do and in the context of the environment in which you are going to be doing it. Not all approaches to a provisioning solution—even some that technically work—will necessarily yield the results that you are looking for.

In a number of ways, provisioning involves federated identity management—that is, the cross-correlating of a user's identity as it is stored in multiple different applications, services, or even different Internet portals or Web sites; and then, by automated means, adding or removing user accounts and permissions to facilitate or block access to the services and resources provided by each of these. The path that you choose through this technological, political, and religious minefield of information management will, of course, need to be based upon the business and IT requirements of your specific enterprise.

> 🕮 For more information about the technological, political, and religious challenges to user management and provisioning, refer back to Chapters 1 and 2.

Best of Breed vs. Integrated Suite

Provisioning is a technology that, by definition, spans a wide range of enterprise applications, services, and the network software infrastructure. When evaluating a new technology with enterprise-wide application implications, the choice of whether to select an integrated suite of applications or to find the best of each type of application and implement them separately is always subject to debate. Although there are arguments in favor of each approach, what you choose, of course, depends upon your environment and your requirements.

Best of Breed

In brief, choosing the best of breed involves selecting individual products from different vendors based on the products' capabilities, while ensuring interoperability with your current enterprise applications and IT environment. The best of breed approach lets you leverage your existing IT software infrastructure—you can retain your messaging and collaboration applications, network OSs, and all of your business applications.

Approaching your provisioning solution by buying the best of breed software applications enables you to facilitate the automation of provisioning the IT user accounts and related access permissions while leveraging the identity information data stores contained in your existing business and network applications. The best of breed strategy can provide the ability to keep your existing identity information in place without first having to migrate or export the identity information sources.

In employing this strategy, however, you will want to carefully research operational and implementation issues, with particular attention to interoperability with your other applications. A best of breed approach is also likely to take more time to maintain and more effort to troubleshoot problems, as you will have to check with multiple vendors.

> ✎ Applications designated "best of breed" might not always be "best" across the board. Organizational requirements differ, and features in standalone applications might not be better for your environment than those found in an integrated suite. Matching the features of the provisioning applications or modules to your organization's list of functional requirements will highlight which components of a provisioning solution are best for you.

Integrated Suite

An integrated suite of applications to support provisioning addresses the interoperability questions involved in making the provisioning solutions work correctly. Finding an integrated suite that precisely fulfills all your business's IT requirements and operational considerations is unlikely—it is generally far easier to find individual applications that more closely meet your needs. Finding a suite of provisioning-related applications, however, is becoming easier, as big vendors such as Microsoft and Sun Microsystems are now supplying integrated solutions.

Buying a suite of integrated applications for provisioning can provide a more seamless connection between identity data sources as well as allow you to consult a single vendor in order to resolve issues or obtain technical support when needed. Most businesses, however, do not use a single platform for network servers and business applications.

If you want to buy an integrated suite of applications that encompasses the entire scope of the identity information repositories that you use in your enterprise, keep in mind that it entails migrating all of this information from your legacy applications to the new integrated suite of applications supporting the provisioning software. Because of the cost, logistics, and disruption that such a large-scale migration would have for existing business operations, the choice of an integrated suite would seem less practical for most businesses, unless they already have most of that integrated suite deployed, or they are establishing a new business in which no information repositories currently exist.

Develop, Hire Consultants, or Buy Off-Shelf

Whether you decide to develop your provisioning solutions in-house, buy a commercial provisioning application off the shelf, or bring in consultants to develop a provisioning solution for you, you will want to review your provisioning requirements and goals in light of the experience of companies that have already implemented provisioning solutions. Although automated provisioning is still a relatively new technology, many companies of all sizes have already explored this territory; it makes sense to take advantage of what these companies learned in the process.

Developing In-House

There are positives and negatives to developing your provisioning solution in-house. Perhaps the most significant upside is that, by developing it yourself, you can make sure it will meet your business needs and your IT operational requirements precisely. Additionally, your provisioning solution can be dynamically modified as needed to meet changing business requirements and conditions within your IT environment.

One of the downsides to developing your provisioning solution in-house is the upfront costs of the developers, and the allocation of these developers and their expertise away from other projects within the organization. It will also take substantial time to develop the necessary coding and test it in your environment before your custom solution is successful and seamless. Of course, any solution needs to be well tested before deployment regardless of whether you develop your provisioning solution in-house or buy it off the shelf.

Designing, creating, and maintaining custom provisioning software to match your business and operational requirements also necessitates substantial development expertise. You might not have this development talent available within your organization, thus it could require hiring of additional talent to effectively produce a solution in-house.

In addition to the development team required to build the solution, you'll need to include teams to handle documentation, support, and upgrading of your custom provisioning application. As a result of employee turnover and reassignment, documentation of such custom applications is essential to an enterprise in order to be able to continue to maintain it. Be sure you assess the costs of documentation, support, and upgrading efforts alongside the development costs.

Hiring a Consulting Firm

Another approach to achieving the needed functionality of an automated provisioning solution is to hire consultants with experience in developing provisioning solutions in the enterprise environment. Like the approach of developing the provisioning solution in-house, hiring consultants to develop the provisioning application can allow you to make sure that it exactly meets your business and IT requirements. By hiring consultants, you can have the custom provisioning solution developed without having to reassign your own development team away from existing projects or hiring developers if you don't already have this kind of talent in-house.

You should, however, be clear that this method is not an inexpensive approach. When hiring a consulting firm capable of developing custom provisioning software suitable for an enterprise environment, you must assume that the upfront costs will be considerable. In addition to the cost, having a provisioning solution custom developed by a consulting firm is still going to take significant time to build, test, and deploy.

When shopping for a consulting firm to implement a provisioning or other identity management project for your business, look for a consulting team that has extensive experience in identity management, directory services, and development and implementation of directory-enabled applications. Make sure that the firm you hire has extensive experience in the development of namespace connectors for both standard interfaces and custom applications. Verify the consulting team's track record in directory and identity management projects with substantial scalability to ensure that they are clearly capable of handling enterprise-level identity management projects.

Buying an Off-The-Shelf Solution

Another approach is to buy an existing commercial provisioning solution. In most cases, the provisioning solution is tried and tested and has been implemented by other companies before you; thus, unlike an internally developed solution, off-the-shelf products have most of the bugs worked out. Such being the case will allow you to quickly deploy a provisioning solution as opposed to taking the substantial time required to develop the software on your own prior to being able to deploy it.

Because organizations have a largely common set of operations to which they want to apply a provisioning solution (add user accounts, disable user accounts, and so on), an off-the-shelf commercial provisioning application should be able to meet most of your baseline functional requirements. In theory, implementing an off-the-shelf provisioning application (or suite of applications) should allow you to minimize the expense and time to implement the solution and provide for faster return on investment (ROI).

The downside to buying an off-the-shelf provisioning solution is that it might not address all of your provisioning needs. You might have identity information repositories with which it simply cannot exchange information, you might have technical requirements that it cannot address, and it might have an approach to implementation that conflicts with your business operations. Although most commercial provisioning applications allow you to customize drivers or connectors to communicate with information stores that the application does not natively talk to, developing these custom connectors requires additional expertise, time, and expense.

One example of an off-the-shelf provisioning solution is abrideanProvisor, a provisioning application that can integrate your existing identity information repositories, enabling automated user management without substantial changes to your current business applications and IT infrastructure. abrideanProvisor's support for a range of underlying directory architectures and products allows you to leverage your existing databases and directories without having to add another directory layer (such as a metadirectory or enterprise directory) to your IT environment. Using a virtual directory—with products such as RadiantLogic RadiantOne virtual directory—avoids the additional network traffic generated by replication and synchronization of information between your distributed identity information data stores. In this environment, Provisor maintains pointers to all of your core identity information sources and directly manipulates information in its original repositories.

Another off-the-shelf solution is Netegrity IdentityMinder eProvision 4.0, which enables you to establish policies that control user access to enterprise services, applications, and resources. (Netegrity acquired its provisioning technology when the company purchased Business Layers and its DayOne provisioning solution). In addition to creating user accounts and assigning access permissions, this product includes a workflow engine interface that allows drag-and-drop modeling of workflow processes (as opposed to having to develop scripts to accomplish this effect).

In addition to these three basic approaches (in-house, consultants, or commercial application), a hybrid approach can also be used. Through this method, you purchase an off-the-shelf solution and either hire consultants or have in-house developers customize it to fit your requirements.

Leveraging Prebuilt Drivers and Connector Toolkits

Commercial provisioning solutions contain a default set of *drivers*—also referred to as *adapters* or *connectors*—that enable it to exchange information with common identity information repositories (directories, databases, and so on). Some of these provisioning applications also provide a toolkit to let you create custom drivers as needed.

Prebuilt Drivers

Using prebuilt drivers will, probably most significantly, expedite implementation of a provisioning solution. Optimally, the provisioning solution you select will have prebuilt drivers to exchange data with all of the critical identity information repositories used in your enterprise. Common prebuilt drivers include:

- Network OSs
- Directory services
- Email and collaboration services
- Databases

Having prebuilt drivers available makes it much easier, not to mention faster, to implement your provisioning solution and to begin to reap some of the benefits of automating account management processes. Prior to investing in a provisioning solution, carefully consider the impact of a product that does not have a prebuilt driver for a mission-critical or high-visibility application.

Connector Toolkits

Although communicating with common applications and standard interfaces (such as SQL and LDAP) is essential in a provisioning solution, businesses commonly also have identity information stored in less standard data structures and/or proprietary applications. As a result, the availability of software toolkits that allow you to build connectors to effectively access and manipulate the identity information belonging to disparate and perhaps uncommon (or even one of a kind) applications can be a critical aspect of whether a given provisioning solution will work in your environment.

Connector toolkits provide flexibility yet require in-house or consulting expertise to develop and test. Accordingly, in addition to the availability of connector toolkits, you will need to evaluate whether you have the development expertise to customize connectors in-house. If not, make sure to include in your evaluation a *realistic* assessment of the cost and time of hiring the expertise to build the necessary connectors.

In many ways, if you can buy a provisioning application that has pre-existing drivers to connect to the various applications, directories, databases, and other identity information stores currently in use on your network, it will be faster and more cost-effective to implement than if you have to build custom drivers. However, there is also an advantage to provisioning applications that provide extensive toolkits that enable you to customize the connectors to these various data sources: they provide you with greater flexibility and a wider scope of applications with which you can interface.

Selecting the Type of Underlying Platform

As we discussed in Chapter 2, the most common approaches to the storage and management of identity information employ some kind of an underlying directory service that provides authentication and controls access to information stored within the enterprise network. Why use a directory in provisioning? A directory service provides scalability, reliability, and enables policy-based management of user accounts and the requisite authentication and authorization services.

Provisioning is fundamentally a directory-enabled application that relies upon change events in either the underlying directory or connected applications (such as the HR application) to initiate the provisioning process that will update all related and contingent identity information repositories. Accordingly, the scalability of a provisioning application will be substantially impacted by the selection of its underlying directory and information integration architecture. As a result of the provisioning application using a directory service product, you can leverage common characteristics of the directory to support scalable and robust provisioning operations.

Directory services are designed with a distributed architecture using multiple directory system agent (DSA) servers to ensure robust and reliable operations throughout a distributed enterprise. Directory services are designed with fault tolerance in mind by implementing multiple directory servers that can handle authentication and authorization should one or more directory servers fail.

The underlying directory service technology also facilitates the use of policy-based controls for the provisioning process. A directory service is designed to enable policy-based networking, managing user accounts and identity information, and handling authentication and access control to all network accessible resources (perhaps more realistically, to as many resources as possible, and at the very least, the most important). Thus, a directory serves as a logical platform for provisioning technologies that automate the implementation of user accounts, identity information, and access permissions across diverse applications and networks.

There are three approaches taken in the architecture of the underlying directory service to manage this information—you can use an enterprise directory, a metadirectory service, or a virtual directory service to handle the user identity information. There is no "one size fits all" solution for managing this information in various business and network environments. Each company must evaluate its own business goals, operational requirements, IT infrastructure, and the demands that will be placed upon the provisioning process in order to determine the best underlying platform for their provisioning application.

All provisioning applications do not allow selection of the directory service it will use. If the directory platform matters to you (and it probably does if you have an enterprise directory deployed), you will want to check provisioning product offerings carefully to verify that they will work with the directories that are critical to your environment.

Enterprise Directory Service

Employing an enterprise directory service as the platform for provisioning is theoretically a more effective overall solution in that it unifies and integrates identity information from all the different identity information repositories throughout the enterprise, simplifying the provisioning process (see Figure 3.2). In this scenario, all the identity information and related data is stored, managed, and updated in a single enterprise directory data store, providing a single authoritative source for all identity information in the enterprise.

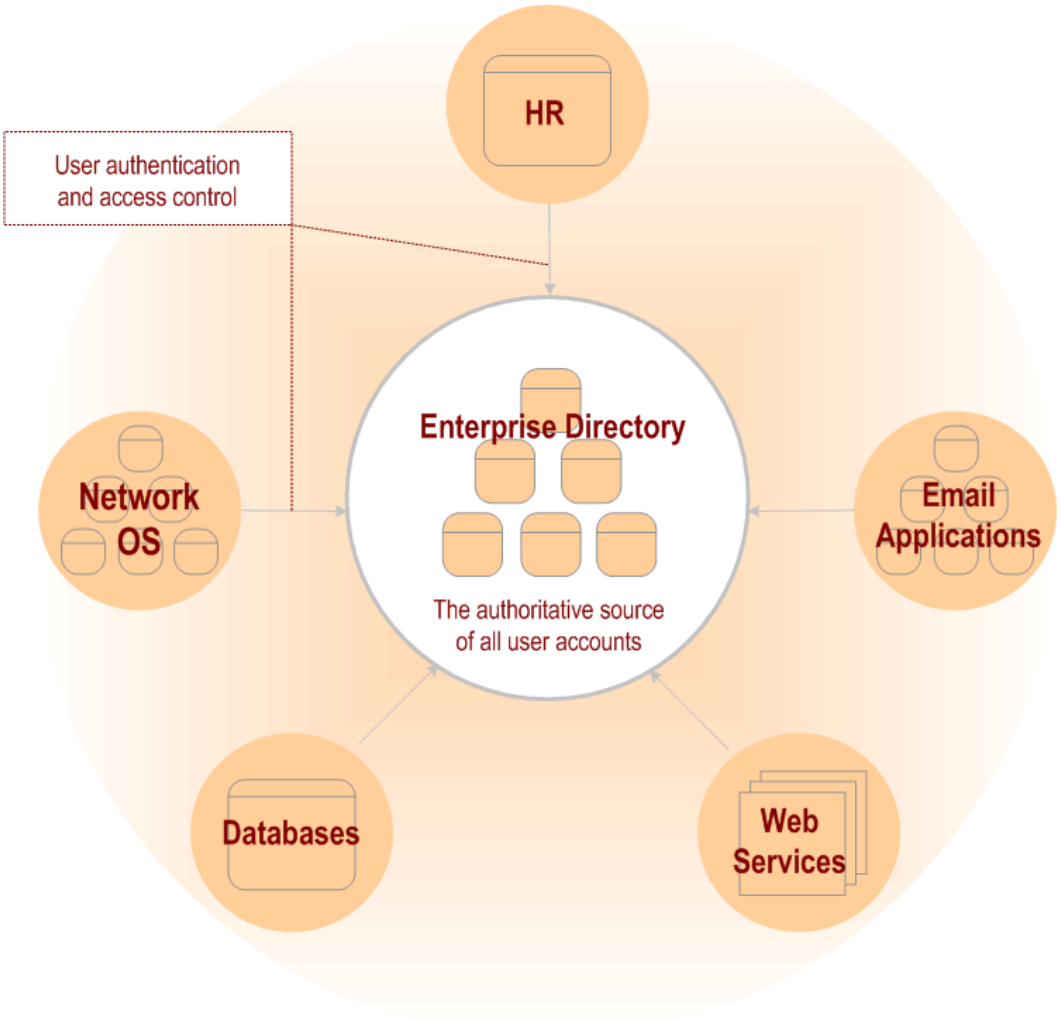

Figure 3.2: An enterprise directory service serves as the core authoritative source of user account information.

Yet, an enterprise directory service is also much less practical to employ in that it requires adherence to a single directory service throughout all divisions of the enterprise. In many cases, managers are responsible for controlling the identity information within their division and are using LOB applications and other identity information repositories that might not lend themselves to enterprise-wide integration. As discussed earlier, the religious and political issues involved in attempting to implement an enterprise directory service can present prohibitive roadblocks to using one as the underlying platform for a provisioning solution.

Metadirectory Service

A metadirectory service provides a more practical option as the underlying platform for a provisioning solution in that it retains the existing identity information data stores. Data from the existing identity information repositories are copied into the metadirectory data store while leaving those repositories intact.

In this scenario, the metadirectory contains all the identity information, yet the metadirectory can bidirectionally synchronize with all the other identity information repositories throughout the enterprise whenever any of that information changes (see Figure 3.3). There are, however, latency issues created by this approach in terms of the amount of time that passes between changes to the information in one repository and those changes being reflected in the metadirectory (and vice versa).

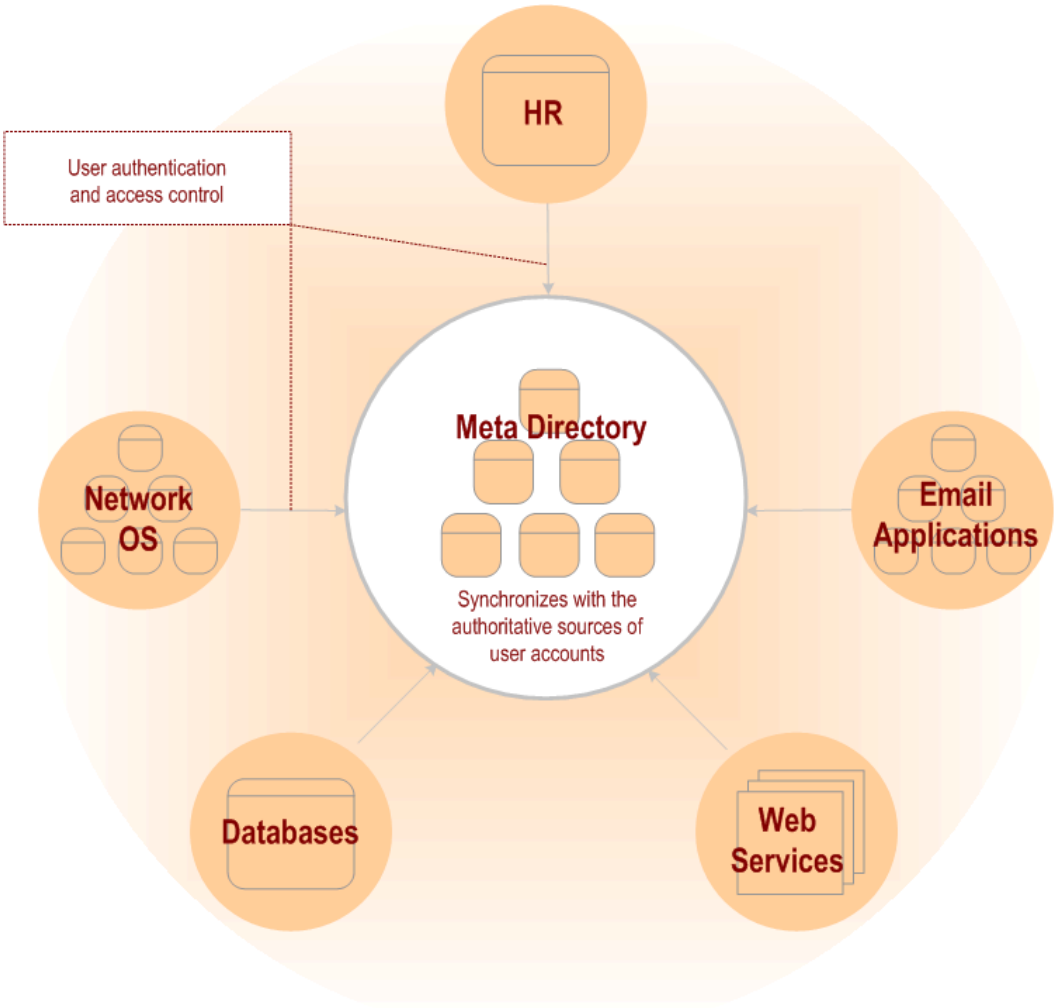

Figure 3.3: A metadirectory synchronizes with other sources of user account information.

In some ways, you still have some of the religious and political issues that affect the enterprise directory scenario in that there still are subdivisions of control over identity information and responsibility for that information. In most cases, the business and network management teams must define which of the information repositories are going to be the authoritative data source for which pieces of information and determine who will have access to the information as well as control over adding or updating the information. Yet, the addition of a metadirectory used only to implement a provisioning solution adds another layer of directory management to your overall IT environment, and can significantly increase the amount of network traffic to support metadirectory operations and synchronization processes.

Virtual Directory Service

For many organizations, an underlying virtual directory service provides the most efficient approach to implementation of a provisioning solution in that it doesn't attempt to duplicate the information already stored within the existing identity information repositories. Instead, a virtual directory service only maintains a set of pointers to where the information already exists and provides a common access point for a provisioning solution to use to access or update the information stored in any of the identity information repositories currently in use (see Figure 3.4).

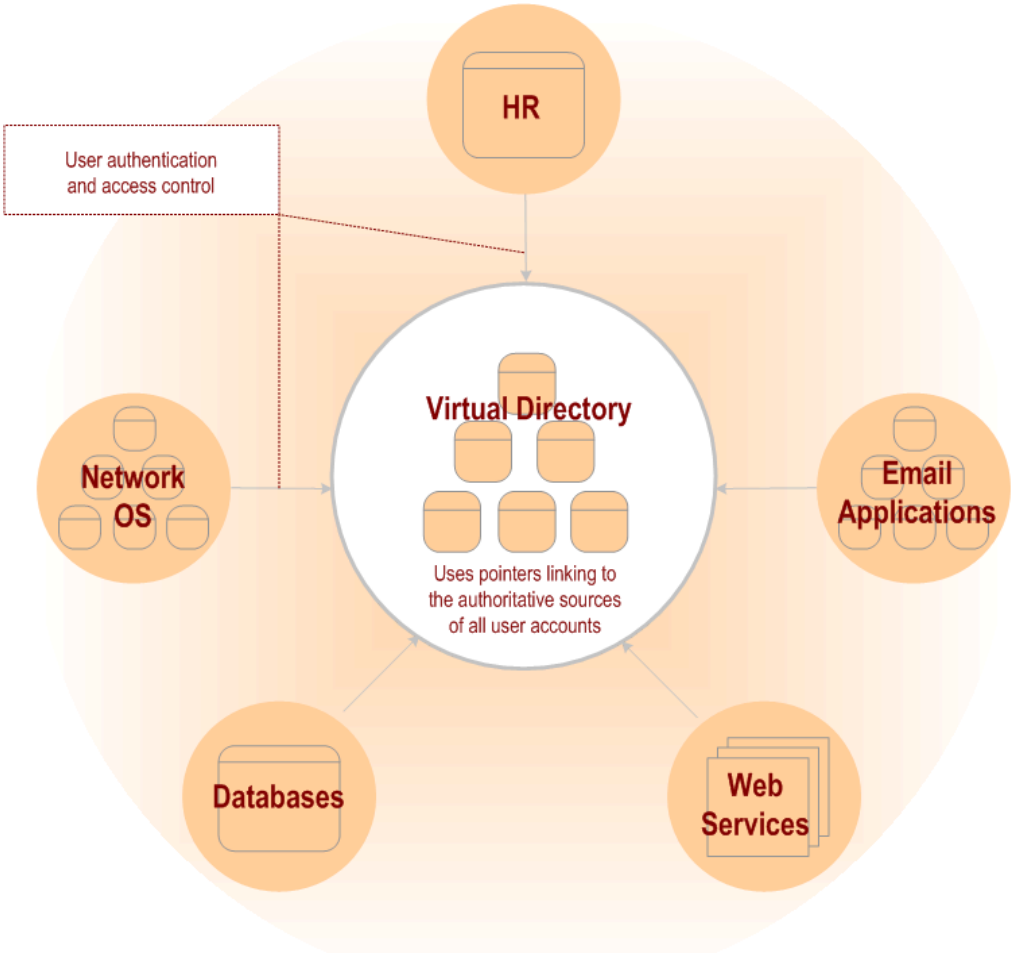

Figure 3.4: A virtual directory service relies upon other repositories as the authoritative source of the user account information.

A virtual directory service also minimizes the amount of synchronization and related network traffic, as it only has to update its pointers when the actual location of its information changes—as opposed to having to synchronize each time the information itself changes (as would be the case in using a metadirectory platform for provisioning).

Application-Specific Considerations

Automating provisioning and deprovisioning of user accounts and related identity information will present both challenges and opportunities through the process of integrating and cross-mapping user identity information between disparate systems. To begin with, user identity information will not necessarily be referenced in the same way in different systems (for instance, the names of users might be constructed using differing formats), presenting technical challenges in exchanging and updating information. Additionally, the ownership and responsibility for each of these different identity repositories is likely to be under the control of different divisions and management within the enterprise, creating both political and religious questions of control and desired approach to implementing the provisioning solution.

Although each application presents a distinct set of issues and concerns, there are a few categories that are both common and critical to most businesses—specifically, HR, messaging and collaboration, and portals and content delivery systems.

HR Applications

At minimum, enterprise provisioning solutions will need to be able to leverage the information in your HR application and to use changes in that information to generate enterprise-wide automatic updates of user account information and status. HR applications are the logical trigger points for provisioning in multiple scenarios—hiring and firing, change of position, change of location, and so on. Many provisioning solutions are designed to work exactly that way—using the addition of an employee as the trigger to cause that user's information to be populated to the network OS and email and core business applications as well as to assign access rights and privileges to enterprise resources.

One consideration when looking for a provisioning solution is whether it provides the right level of access to the HR data and provides the needed level of controls over the limits to the access to that information. HR data is commonly considered the authoritative data source of user (that is, employee) information from an enterprise perspective, so the provisioning software should be able to leverage this HR data without modifying it. Stringent auditing controls for access to and dissemination of this information should be another core feature of the provisioning application.

Provisioning user accounts throughout your enterprise applications based on changes to the HR database also provides an opportunity to improve overall security in the enterprise IT infrastructure by automatically setting user access and permissions based on their job functions, expected roles, and location. Similarly, the ability to deprovision user accounts based upon a status change in the HR database (such as firing or being laid off) will also improve security by automatically disabling user accounts and prohibiting access to sensitive enterprise information and resources.

Messaging and Collaboration

Messaging and collaboration services such as Microsoft Exchange Server and SharePoint Portal Server are applications that present an obvious scenario for user provisioning. Everyone, virtually irrespective of their job description, needs an email account, and setting up email accounts requires far more information than just a username and password:

- Distribution group membership
- Physical location at which the new employee will be working
- Addressing information, phone numbers, and other key location or job function data

As an example of how provisioning applications can help in the management and security of messaging systems, consider Microsoft Exchange Server 2000/2003 and its integration with Active Directory (AD). This integration supplies enhanced administrative rights to Exchange administrators, and thus puts overall network security and delegation of Exchange-related tasks somewhat at odds. Provisioning applications can mitigate the security vulnerabilities by automatically limiting Exchange administrators to only the rights that are necessary to carry out the management of Exchange. Furthermore, provisioning can assist with controlling major projects such as Exchange migration, ensuring the security of the process, and providing an audit trail.

Portals and Content Delivery Systems

Content delivery systems of all sorts are facing a balancing act between facilitating personalization of the content to which a user wants access and the ability to maintain the privacy and security of the user's identity information. To facilitate personalization, companies must obtain and collate information about a person's preferences and patterns of information access and usage. Yet, to do so, and to bring this information to bear in the context of the user's Web site experience, companies must monitor users' activities and associate the results of the monitoring with the users' profiles.

Additionally, the different services available via content delivery systems, even at the same site, frequently require multiple logons and multiple user profiles that are storing much of the same identity information. Especially in the context of the Internet—where the user may go from site to site seeking information and services at various portals and content delivery systems—having to identify and authenticate themselves to each site can render the user experience frustrating at best.

Given these constraints, implementing some sort of single sign-on (SSO) or federated identity management system can not only provide a more seamless user experience but also give the companies developing these Web services, portals, and content delivery systems more reliable information about their visitors. Yet in this context of cross-enterprise federation of users (and provisioning of access), the strength of the security measures within the identity management system is even more significant.

Emerging Technologies and Standards

When considering any provisioning and user management solution, you must be aware of how changes to the industry standards and enhancements in technologies can affect the future of your IT and business operations. Problems that exist in current technological initiatives or platforms, or which deter ongoing business workflow operations, might be issues that can be alleviated by application of technologies that have been recently introduced (or are still emerging). As a result, paying attention to developments in the industry could potentially help you solve intransigent problems, and save you time, money, or frustration.

Industry-leading vendors of directory and provisioning software are integrating their identity management solutions into applications and product suites that leverage key emerging technology standards such as XML and Web Services. For example, Microsoft is integrating namespace federation services into AD as part of the Windows Server 2003 update currently scheduled for 2005. This new Active Directory Federation Service (ADFS) supports mapping user accounts between not only different identity information data stores within an enterprise but also between different companies. This service will provide an underlying mechanism to enable cross-company authentication and authorization of user accounts. This capability will facilitate integration of corporate partnering efforts, providing partners with needed access to enterprise resources as well as support portal-to-portal user recognition and access control. This new federation service for identity information will be used by Microsoft to enable SSO functionality via Web Services and related protocols.

Considering Markup Languages: XML, SPML, and SAML

XML is becoming the de facto industry standard for information exchange between applications, services, and information repositories of all sorts. XML provides a structured format used for exporting or importing any set of data, which allows XML-compliant software to integrate data from any source. Industry vendors of a wide range of applications, services, directories, databases, and virtually every other kind of software have adopted XML as the global method of exchanging information.

With XML, businesses increasingly can be assured that their provisioning applications will be able to exchange and update identity information contained in most enterprise applications. Interoperability between the provisioning software and your legacy directories and databases is enhanced with XML, enabling the integration of all the identity and information that needs to be accessible to support a fully automated user management and provisioning solution.

Support for XML in the provisioning solution should be considered a central element—as an emerging standard with widespread industry support, interoperability between the provisioning solution and future enterprise applications may well be contingent upon XML functionality. In addition, multiple XML-based specifications supporting security and provisioning are gaining acceptance in the industry, and vendors are beginning to integrate these specifications into their new product offerings.

SPML

The Service Provisioning Markup Language (SPML), for example, is rooted in XML and is designed to provide a framework for dealing with system resource allocation—defining the provisioning of user accounts and permissions for networks, services, applications, and systems within an enterprise as well as between organizations. The SPML standard is approved by the Organization for the Advancement of Structured Information Standards (OASIS).

SPML, as an XML-based specification, provides the functionality and support for provisioning operations across different platforms. This capability will allow administrators to automate user account provisioning for both internal and external enterprise networks, applications, services, and resources. The use of SPML could allow the replacement of vendor-specific and proprietary namespace connectors/adapters with open-standardized XML schema for the exchange of provisioning-related information, enhancing the overall interoperability and freeing provisioning applications from vendor-specific constraints.

In addition to easing provisioning deployments, SPML-compliant services will facilitate authenticated access to diverse network resources, reduce administrative overhead in providing access to these enterprise resources, support two-factor authentication, and create a comprehensive audit trail. The design of SPML allows it to interoperate with the latest versions of Security Assertion Markup Language (SAML) and the WS-Security standards supported by OASIS as well as the Simple Object Access Protocol (SOAP) standards supported by W3C (in version 1.2 of SOAP).

SPML has been gathering substantial vendor support—from such companies as Sun Microsystems, Novell, PeopleSoft, BEA Systems, and many others—who see it as a means of making it faster and cheaper to deploy Web Services-based provisioning solutions and simplify management of these applications.

SAML

Another OASIS standard, SAML manages the exchange of authentication and access control data between different organizations and works with SPML. SAML provides an open methodology for handling user identity authentication that is independent of vendor-specific authentication schemes. SAML is gaining widespread industry support as a vendor-neutral cross-platform authentication interface and is supported by vendors such as Sun Microsystems, IBM, Microsoft, Novell, and RSA Security. SPML and SAML work together to enable organizations to automatically create and authenticate user accounts, providing a foundation for Web Services-based provisioning and SSO applications.

Assessing Web Services

Web Services are slowly coming to the forefront as a communication and commerce platform, providing unique opportunities to integrate diverse business and technological objectives and operations. From a business perspective, enhanced methods of working with partners and sharing information, services, and technological resources enable inter-business synergy that was previously difficult at best. With Web Services-based identity management functionality, business can allow personnel from new business partners to access resources that were previously only available to internal staff. Similarly, businesses that have developed in-house applications and tools can now expose the capabilities those tools provide via a Web Services interface, leveraging existing functionality to add value to new customer markets via the Internet.

Web Services are fundamentally rooted in XML, using document type definitions (DTDs) to set the structure of the XML documents being manipulated for the purposes of providing the Web Services. Using XML allows for communication with a wide range of applications and data sources; XML is, by design, an extensible data format easing interoperability across platforms. Web Services leverage the XML Infoset, XML schema, and XML Namespaces as key underlying elements within the Web Services architecture (see Figure 3.5).

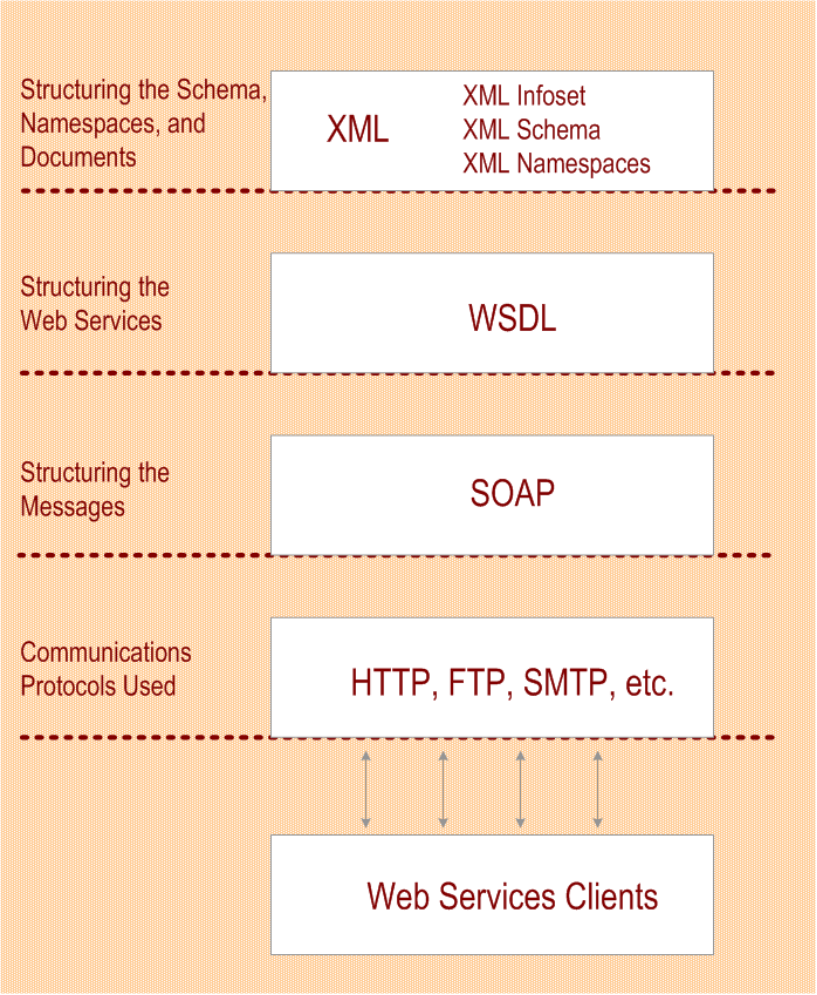

Figure 3.5: XML is the foundation for structuring Web Services and messaging, and uses common transport protocols to communicate with clients.

A core standard used for specifying how XML documents define specific Web Services are structured is the Web Service Description Language (WSDL). Web Services handle communications with clients, applications, and other service via SOAP. Fundamentally, Web Services interoperate with client applications via messages that are constructed using SOAP and are transmitted via one of several Internet communication protocols such as HTTP, FTP, SMTP, and others.

To support the range of operations needed in this new Web Services paradigm, additional Web Services protocols and standards are being developed, including a range of security standards and protocols such as WS-Security, WS-Secure Conversation, WS-Federation, WS-Trust, WS-Policy, WS-Authorization, and WS-Privacy; These protocols and standards provide secure communication, cross-enterprise policy support, authentication and authorization management, and identity federation.

For more information about these emerging Web Service standards for security and federation, check out the OASIS Web site at http://www.oasis-open.org.

The support for these protocols as well as XML and its operational derivatives (such as SAML) is an essential part of providing Web-based support for the SSO functionality so desperately needed for inter-business identity federation—the mapping of user accounts and the underlying identities they represent between the Web sites, portals, and Web services. Provisioning user accounts in the Web environment and authenticating identities across platforms, sites, and services will be greatly enhanced by these Web Services developments.

Evaluating the Liberty Alliance

The Liberty Alliance is a cross-company identity information management and federation initiative driven by Sun Microsystems and supported by a wide range of industry vendors. The Liberty Alliance is dedicated to developing and implementing an identity management system that will effectively provide users and businesses with an SSO capability, such that all users could connect to any of the Web sites, services, or portals subscribing to the Liberty Alliance specifications and only have to log on once. This potential capability is a far cry from the existing state of affairs in which virtually every Web site or portal requires users to log on to each site independently.

The identity management service being proposed by the Liberty Alliance is a federated model, allowing users credentials to be recognized by each of the independent Web sites irrespective of how the corresponding user identity data is stored at the destination site. In this model, which is similar to how automated teller machines (ATMs) work, only the information necessary to authenticate who you are (your identity) and the appropriate degree of information access (your authorization) is shared. This setup differs significantly from the model employed by Microsoft's Passport system, which involves the user identity information being stored in a single structured format on Microsoft servers, requiring each vendor to authenticate incoming user traffic against this data store.

Regulatory Requirements

New federal regulations have driven the need for identity management solutions capable of providing a high level of security for sensitive information and resources. Provisioning solutions can greatly assist in managing information security by automating the assignment or removal of permissions and access controls for users in the enterprise.

Three recent pieces of legislation play a foremost role in the new demands for information security in the enterprise: the Health Information Portability and Accountability Act (HIPAA), the Sarbanes-Oxley Act, and the Graham Leach Bliley Act. Each of these laws places additional responsibilities on corporate executives and IT departments to control, monitor, and audit their information systems. Because of the civil and criminal penalties that can be enforced against companies and their executives for lack of compliance, user management and provisioning initiatives go a long way toward helping organizations meet these requirements.

HIPAA

HIPAA is legislation that is intended to protect the confidentiality of personal medical information, both in terms of its usage while providing healthcare services as well as in the transmission and handling of patient information. This information must be protected even during the insurance review of the services provided and in the accounting related to the healthcare services.

HIPAA includes regulations that cover privacy of information as well as lay out security requirements for managing the information, respectively known as the Privacy Rules and the Security Rules. Although delineated separately and having different start dates for enforcement, these two sets of rules are interconnected and interdependent regulations.

The HIPAA regulations define *covered entities* as businesses that are responsible for controlling and managing the medical information they handle (such as businesses that provide medical services, insurance companies that review and process medical information, and so on). HIPPA specifies that these entities must safeguard this data—described as Personal Health Information (PHI)—from disclosure (unintentional as well as intentional) or misuse in any way that violates the HIPAA standards or implementation specifications. By now, if you are one of these covered entities, your organization has had many meetings involving your executive, legal, HR, and IT departments to discuss the HIPAA regulations and the implications for your business.

One of the requirements that HIPPA brings to your IT environment is the need to perform regular audits of IT activities to verify that personal health information has been protected from unauthorized disclosure and/or illegitimate usage. In addition to establishing baseline security requirements for patient information, HIPPA requires that the information that *is* used or disclosed be kept to the least amount of information necessary in order to perform the required action (the intended use, request, or disclosure activity). In other words, not only do you have to prevent the information from unauthorized access or use, you must also make sure that when it is used, the minimum amount of information is disclosed.

To comply with these requirements, organizations that are considered covered entities must have security policies and procedures in place to ensure that appropriate levels of access control are implemented and that the disclosure and/or use of the personal health information is audited. Ensuring that the right users have the correct level of access, and that all use of the personal healthcare information is consistently audited, is a daunting task if done manually. This challenge becomes much easier if user account creation and the assignment of access rights is automated and controlled by policies established to enforce HIPAA compliance throughout the enterprise.

Thus, this area is one in which integration of the provisioning software with your HR database (which identify who has access to personal healthcare information) becomes far more than merely a cost-reduction measure and security improvement—it becomes a liability-minimizing mechanism for your organization. Not only will the appropriate access rights and permissions be assigned to the appropriate users, but auditing of that provisioning will also be automatic.

It is in regulation-compliance situations that a provisioning solution becomes more than merely advantageous—such a solution becomes a technology essential to implement within your enterprise. Tracking the requisite information without an automated process is not only laborious but also mistake prone—people can misunderstand the policy, incorrectly assign a user's access rights, or forget to disable a user account when the user leaves the company or moves to a position in which the user no longer needs access to personal healthcare information. Automated provisioning and deprovisioning avoids these kinds of errors, and consequently avoids the concomitant risks and liabilities. As an added bonus, the now-mandatory detailed auditing of who has access to what personal healthcare information (as well as how and when was it accessed, and in what capacity it was used) are part of the feature set that most provisioning solutions provide.

> 🕮 For more information regarding HIPAA, go to http://www.hhs.gov/ocr/hipaa/ and http://www.wedi.org/snip/public/articles/index%7E6.htm.

The Sarbanes-Oxley Act

Another key new federal regulation is the Sarbanes-Oxley Act, which requires extensive and rather stringent control and monitoring over the handling of financial information and financial reporting. The corporate executives (chief financial officers—CFOs—and chief executive officers—CEOs) of businesses regulated under the Sarbanes-Oxley Act are now required to assess their corporate reports and personally certify the accuracy and of the report contents. Serious penalties (including criminal charges) can be enforced against corporate offices for infractions of rules in the Sarbanes-Oxley Act.

Companies that are affected by this act must oversee the implementation of internal controls for financial systems and subsidiary systems and applications that are used in the generation or reporting of that financial information. These subsidiary systems should be considered in depth—monitoring and controlling requirements of Sarbanes-Oxley apply not only to the specific financial applications that contain the information or produce the reports but also to the entire IT infrastructure that supports the financial operations and applications. These systems frequently cross business, functional, and geographical boundaries, as well as span networks, services, applications, and IT divisions.

Adherence to Sarbanes-Oxley requirements must be documented, explicitly detailing the internal procedures used by the enterprise, and must demonstrate compliance with the all of the Sarbanes-Oxley requirements. Yet, in order to do so, these executive officers must have access to far more detailed procedural and auditing information that shows every internal procedure used to track financial information and demonstrate control over that information and the processes that produce it.

In this arena, provisioning can lend a very large helping hand, by establishing access controls that limit who has access to the financial information and what information they have access to. All of this responsibility can be automated using a provisioning solution, enabling company executives to set policies that render company operations in compliance with the Sarbanes-Oxley requirements. Importantly, provisioning solutions also manage the deprovisioning of user accounts and access to sensitive information, not only improving overall security but also helping you meet the strict access control demands specified in the Sarbanes-Oxley regulations.

> 🕮 For further clarification of the Sarbanes-Oxley Act of 2002, review the information at http://www.sarbanes-oxley.com/ and http://www.sec.gov/spotlight/sarbanes-oxley.htm.

The Graham Leach Bliley Act

The Graham Leach Bliley Act of 1999 is a federal regulation that addresses a range of issues affecting how banks, insurance firms, and other companies handling financial information operate, and additionally specifies new rules for the management and protection of identity information in financial transactions. This act defines a set of regulations that requires companies to handle customer financial information with new security and privacy constraints in mind.

To begin with, companies must strictly control how their employees access customers' financial information, limiting access to the specific customer financial data that is needed to perform the tasks related to their job, and not allowing employees to globally access information that falls outside of the purview of their job. An employee at a bank who is responsible for processing credit-card applications for customers, for instance, is not allowed to access mortgage or loan-related information for those same customers.

In addition to specifying access controls and limitations, the Graham Leach Bliley Act regulates how companies create and store customer financial information, how long the information must be stored, and how access to the information is monitored and audited. There are also secondary considerations, such as the response of major accounting and auditing firms to the regulations presented in the Graham Leach Bliley Act, that require companies that use, store, or access such customer financial information to pass information security audits.

Once again, provisioning comes to the rescue of beleaguered CEOs, CFOs, and IT administrators—the features and functionality of automated user management and provisioning can also assist companies in managing their user accounts; enabling them to set policies that control access to sensitive information and supplying the auditing capabilities necessary to document compliance with this act and other federal regulations.

By carefully assessing your company's responsibilities under these new federal regulations and integrating your business and IT operational requirements into the design of policies that are implemented by your provisioning application, you can be assured that each user is granted access to only the appropriate systems and information. Additionally, when someone leaves the company, that user's access is automatically deprovisioned, protecting you from a range of liabilities. This ongoing auditing included in the provisioning and deprovisioning operations that assist in demonstrating compliance with these federal regulations is a benefit not to be overlooked.

> 🕮 For more information regarding the Graham Leach Bliley Act of 1999, check out the details at http://banking.senate.gov/conf/ and http://www.keytlaw.com/Links/glbact.htm.

Summary

Throughout this chapter, we've looked at the fundamentals of automated provisioning systems and how they are employed. In addition, we have evaluated operational aspects of the provisioning applications in the enterprise environment. Recognizing that each business and its IT environment is different, various strategies for assessing and selecting provisioning solutions were presented. Which strategy to use—integrated suite or best of breed—and whether to develop provisioning yourself, hire consultants to build it for you, or to buy it off the shelf, fundamentally comes down to your assessment of what works best for your company.

In all of the provisioning applications, an underlying directory service provides infrastructure support for the identity management, authentication, and access control operations that must be performed. Accordingly, the various directory architectures were reviewed in the context of implementing provisioning solutions.

Application-specific factors involved with the integration of provisioning with common enterprise applications, such as HR software, messaging systems, portals, and content delivery systems highlighted the significance of provisioning as technology that crosses platform and application boundaries.

Support for provisioning is also being seen in some emerging technologies and standards, where developments in XML-based specifications are supporting provisioning operations. Web Services technologies are supplying cross-business integration of services and provisioning efforts are buoyed by the inter-business Web-based authentication and authorization mechanisms developed by the Liberty Alliance consortium.

With the advent of several new federal regulations requiring companies to control, monitor, and audit information access much more closely, provisioning is shown to play a key role in enabling companies to meet the new demands of the HIPAA, Sarbanes-Oxley, and Gramm Leach Bliley regulations.