

[**Editor’s Note:** The following excerpt is from the free eBook *The Administrator Shortcut Guide to User Management and Provisioning* (Realtimerepublishers.com) written by Dave Kearns and available from a link at <http://www.abridean.com/ebook>.]

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 than 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 than 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.

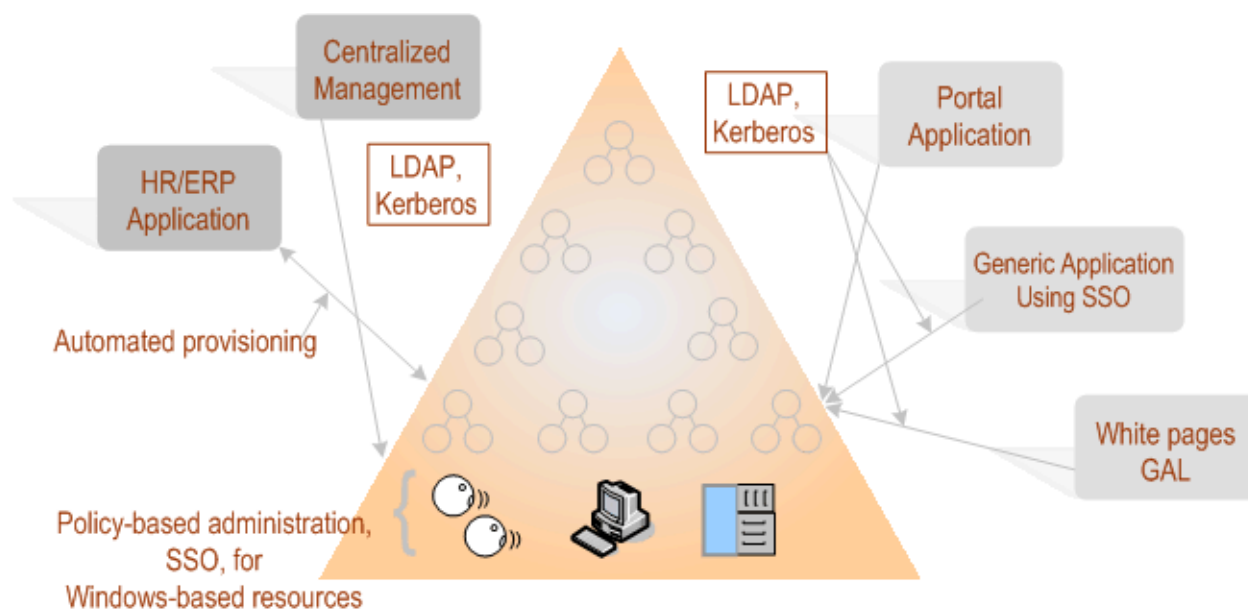


Figure 1.1: The hub-and-spoke architecture of enterprise directory services, meta directories, and virtual directories.

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.

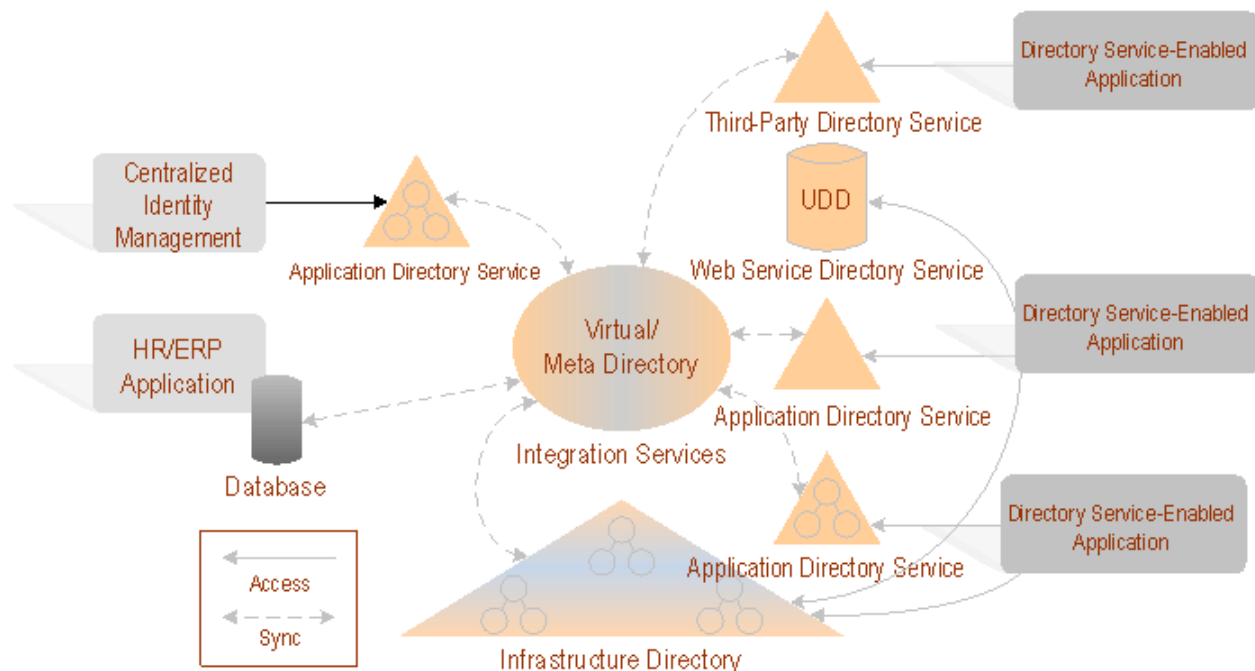


Figure 1.2: Virtual and meta directory services.


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 LDDs. 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 possibly 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.

[Editor's Note: This content was excerpted from the free eBook *The Administrator Shortcut Guide to User Management and Provisioning* (Realtimerepublishers.com) written by Dave Kearns and available from a link at <http://www.abridean.com/ebook>.]