

[**Editor's Note:** The following excerpt is from Chapter 2 of the free eBook *The Definitive Guide to Identity Management* (Realtimerepublishers.com) written by Archie Reed and available at <http://www.rainbow.com/insights/ebooks.asp>.]

Chapter 2: Identity Management and Security

Chapter 1 discussed the basics of Identity Management. As observed, Identity Management is not a simple off-the-shelf solution—comprised of both technology components and business strategies and policies, it is essential to research and understand the many aspects of Identity Management to provide a true Identity Management solution. That is the goal of the following chapters: to delve deeper into the components required and available to support an Identity Management initiative.

Identity Management helps meet the key security management requirements that most organizations have today. The security requirements of an enterprise that provides access to employees is different than those of an Internet-based consumer site. In the case of the enterprise, regardless of its size, it is important that there exist some record of each employee—the employee's role and access levels across systems. In the case of the consumer Internet sites, the same information exists, but it is obtained with less concern for accuracy, and the information has different attributes and contexts and may be more readily shared across other sites. In this case, there may be more reliance on validating information through credit card companies, which is something an employer generally will not utilize. Despite these differences, there is a great deal of commonality across definitions of Identity Management frameworks. In this chapter, we will evaluate how an Identity Management solution is defined within the security infrastructure of these scenarios, exploring each of the components that have a vital role in protecting a system.

Risk and Trust

When evaluating solutions that enforce security, there is always compromise. Within the security space, this compromise is considered risk management. The reason is that generally the more security you put into place, the less usable the system. In line with that consideration is the acceptance that a system has a value to the organization, which must be secured.

Often the only way to calculate the risk is to use a qualified actuarial representative—essentially a statistician who computes risks and premiums, generally for insurance policies. Given that this resource is beyond the reach or reality of most organizations, the calculations are done by internal staff as they attempt to define ROI for a project. Calculating such value is different from organization to organization, and project to project, and can involve basic concepts such as the impact of having employees unable to work overtime due to system security breaches, the potential impact of customers (for example, a boycott), or even legal action against the company. Although calculating the value of more physical considerations is fairly easy, determining the cost of service abuse relative to corporate reputation and similar intangible assets can be very difficult. The point is that as you begin to assess the value of the assets that you are trying to protect, it is important that you utilize representatives from across the organization.

Perhaps a more appropriate baseline to begin assessing risk is to ask more generic questions that can be changed as appropriate for your specific situation, along the following lines:

- How secure is my infrastructure? Is sensitive data protected if disgruntled employees gain access to restricted systems or resources?
- How secure are my connections beyond my infrastructure? Can you ensure that your high-value online transactions are binding?
- How secure are my communications within and external to my infrastructure? Are confidential emails and files protected from interception by unauthorized employees, competitors, and malicious parties?
- Is there a plan to improve security over time?
- Is security actually improving over time?
- Can I transfer risk using different solutions (for example, outsourcing)?
- How does my security compare with that of similar companies in the industry?
- How will I respond to a breach in security?

The important thing to note is that understanding the level of risk you are prepared to accept relates directly to the level of trust you have in your systems and your relationship with other organizations and their systems. This concept of trust becomes very important due to the fine line drawn between fully securing a system such that it is unusable and securing it enough such that risk is mitigated but the system can be used to actually perform the task for which it was implemented.

Beyond this matter, you need to consider impact to your company or “brand”. In the case of any organization, there is risk if untrusted parties with whom you have no legally binding or enforceable agreements gain access to confidential business data, in particular customer or employee data.

The cost of securing a system, let alone many systems and their integrated applications, can be astronomical. Consider that the more complex a security solution

- The higher the potential cost of implementation.
- The higher the potential cost of administration and maintenance.
- The more chance that services or data will be unavailable when needed and someone will not be able to do his or her job.
- The more chance that security configuration will be overlooked or someone will not be able to do his or her job.

These truths eventually increase the risks rather than decrease them. This is called the law of diminishing returns.

Identity Management solutions help increase security and minimize the risk of systems by helping manage the lifecycle of a single identity and mapping that identity across multiple systems, principally reducing complexity and cost. A project such as Identity Management that plans to manage information about people and store organizational knowledge will have many security requirements and potential restrictions. As such, you should ensure that your organization has appropriate security policies, and that you are applying the appropriate level of security to the project itself. The optimal scenario for any directory services project is to involve security as part of the core team that will deliver the directory services solution. In this way, you not only gain a representative, or team, who understands the policy and can apply it to the project but also the potential to have the policy changed if there are any issues that are not met, cannot be met, or should not be met. The involvement of security should also be tempered with full interaction with the business representatives to ensure that functionality aspects are not overlooked.

The core nature of Identity Management is to support other networked services and applications. In isolation, an Identity Management solution is useless. As a result, the focus of any security analysis requires a great deal of investigation into the ancillary services to ensure that all potential risks are identified and dealt with in the networked environment.

As we discussed, security is largely about risk management. Once you have assessed your situation, there is a traditional trio of steps to be dealt with when defining and maintaining a security solution that will deal with possible threats to a system. These steps are known as PDR:

1. **Protect**—In this step, you define your levels of security around the system and the way in which you will implement them.
2. **Detect**—Despite all the attempts you make to protect a system, there will likely be a way around it, so you need to ensure that you implement monitoring and intrusion-detection into your solution.
3. **Respond**—To avoid a panic response to a security breach and to successfully deal with breaches of security, you must ensure that there is a step-by-step approach that those involved in the process can easily follow.

These steps define the traditional high-level approach to system security. Within each of the steps are a number of focused activities. To protect a system as well as have any chance of detecting a problem with it, you must understand the system. Seems logical right? Well, unfortunately, many installations of technical solutions are done without considering the aspects of security. It is important that you create a regular review of your environment, especially during times of change.

Let's turn to the specifics of each core security component of the applications managed by an Identity Management solution. To do so, we will begin with an exploration of authentication.

Authentication

Authentication is the core concept of Identity Management. Authentication lies on the outer perimeter of a security infrastructure and can take a number of forms related to who or what is trying to gain access to the system. As you can see in Figure 2.1, most security systems traditionally use the silo approach—each application maintains “identity” information about users, their rights, and some form of control to manage access to the application.

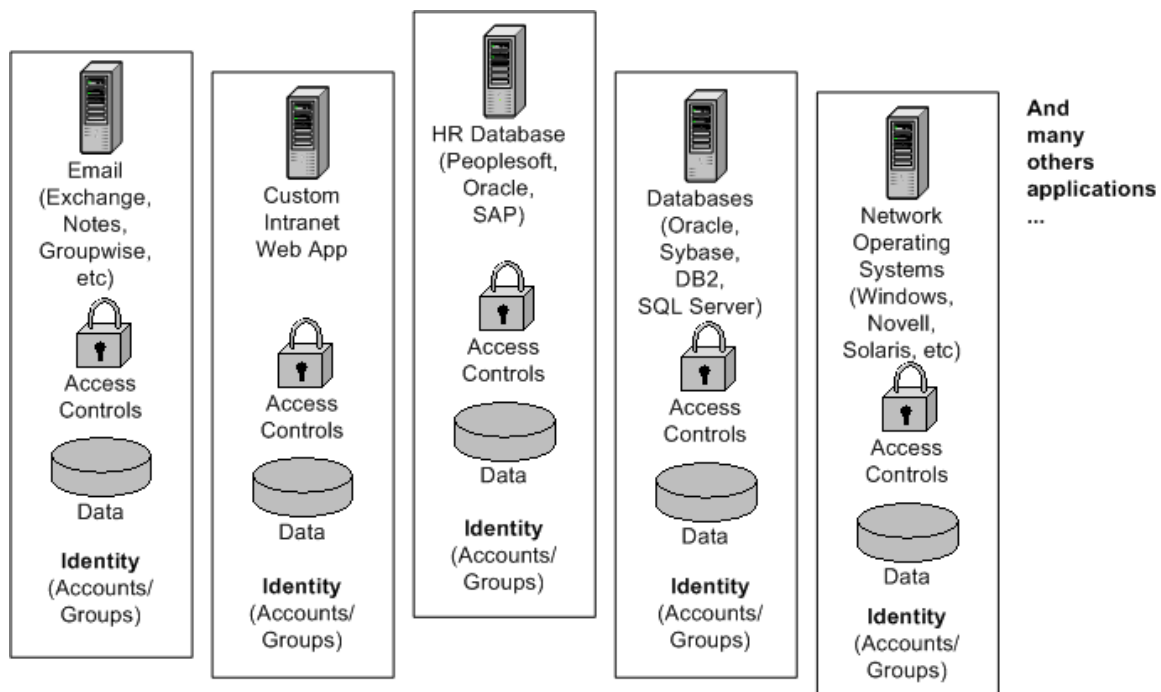


Figure 2.1: Silos of security and access information.

Authentication is the first thing that happens with any system, proving who the user is. The boundary of the system is the important thing to keep in mind as we move through this discussion.

Authentication can happen in numerous ways and requires some form of identification through the presentation of credentials to be validated by the system. At a conceptual level, authentication to a system for individuals can occur in a number of ways:

- Something you know—Traditionally, username and password, but may require the answer to specifically pre-arranged questions
- Something you have—Such as a time-based key device or token (for example, Rainbow Technologies’ iKey and RSA Security’s SecurID)
- Something you are—Considered to be primarily biometric measurements, such as retinal scans, fingerprints, and so on

Although these types of solutions are often used in specific situations due to complexity or cost or appropriateness, a system might enforce authentication to a specific service via several of the listed options. Indeed, the context of the authentication request becomes important. For example, accessing internal company resources may require simple password-based authentication from within the local network; however, when dialing in, authentication may only be possible using both the password and an additional key device control. Biometric solutions are potentially the greatest opportunity of identifying an individual; however, they are rarely be used in isolation to ensure security. For example, several vendors such as Rainbow Technologies and Digital Persona provide fingerprint scanners that act as tokens but require passwords or PINs to complete the authentication cycle. Chapter 3 will review specific implementations of vendor solutions, and Chapter 4 will cover potential issues with specific implementations. Two key drivers of Identity Management projects that deserve discussion here are password policy management and single sign-on (SSO).

Password Policy Management

Password management, which we discussed in Chapter 1, needs more explanation before we review password management applications in Chapter 3, specifically around the capability of password policy management. Password policy management is vital if that is all you are using to secure access to your network. Most enterprises enforce an account-naming standard. This common approach means that the account name is easy to derive and is therefore a minimal component of the authentication security.

That means that the password is the only real security you have. The traditional, almost comical approach to dealing with complex and multiple password management by individuals is to create a simple password that is easy to guess and/or to write down this password somewhere (for example, the ever popular sticky note on the monitor).

Password policy management becomes a significant issue without some form of top-down solution. Identity Management solutions generally provide for this type of thing, including the ability to define a consistent password policy across systems by coalescing the minimum and maximum capabilities of each system that is being managed, then providing a combination syntax and lifecycle constraints. Syntax deals with the format or composition of the password and ensures that it can be consistent across systems, using criteria including:

- Minimum and maximum password length
- Minimum and maximum alphabetic character counts
- Minimum and maximum numeric character counts
- Minimum and maximum punctuation character counts
- Exclusions (for example, specific words, variations of the account name)
- Consecutive character types
- Instances of any character
- Password uniqueness (for example, similarity check with previous passwords)

- Sequential character checking

Password Lifecycle

- Minimum and maximum days between password changes
- Password history count to enforce the number of times before the same password can be used again
- Number of failed logon attempts allowed before lockout
- Lockout duration
- Password reset questions (for example, what is your favorite color?)
- Any role-specific requirements (systems administrators have a different set of requirements than back office staff)

Remember that there are several goals to meet. Initially, the goal is to create a system in which the password-access mechanism meets a policy level that you are comfortable with such that attacks can be mitigated. Beyond this list of capabilities is the common requirement for auditing for specific events throughout the password policy management system. This requirement sounds simple but can be quite complex and is often not dealt with effectively by many systems. (This is related to the audit requirements discussed shortly.) For example, if the Identity Management system is primarily implemented to support a password management requirement, it is likely that the solution does not maintain each application access control layer OR its log collection (that is, each application is still accessed via its own UI and checks its own local security layer and reports failures to a local log). In this case, the Identity Management system may only enforce standard password changes into the local system without being able to access the applications' security logs. Thus, the Identity Management system will not identify times when local access fails causing a lockout on that system. Dependent on the solution, a SSO component would be expected to deal with the access layer at least.

In evaluating systems, you will find that some do not deal well with international characteristics on both technical and cultural levels. For example, technically some do not readily support internationalization and if they do, they have not been localized to all the countries that you may need to support. Culturally there may be preset questions that make no sense. For example, asking for the last four digits of a Social Security number makes sense only in the USA.

The key with password-based solutions is a reliance on encrypting the password in the store, securing access to that store, and ensuring that a validation request is also secured. Commonly, solutions use a hash of the password and a random challenge across the validation connection. Essentially, a system issues a challenge, the authentication service then hashes the challenge with the password and responds to the system so that it can validate the response against the stored password. There is an implicit trust in the security store, which has become standard practice within the enterprise. Moreover, since the beginning of dialup services through to fully featured Internet Service Providers (ISPs), users have trusted the providers with their password information.

There are various password authentication services available, such as

- Password Authentication Protocol—PAP is defined in Request for Comments (RFC) 1334 and is a rather simple and insecure solution that is rapidly decreasing in popularity.
- Challenge Handshake Authentication Protocol—CHAP is defined in RFC 1994 and has various derivatives; although usage of this protocol is also decreasing, CHAP still has reasonable use.
- Remote Authentication Dial-In User Service—RADIUS is a much more scalable solution, as is Terminal Access Controller Access Control System (TACACS). Both find their existence in dial-up solutions, especially in ISP environments.

Although it is unlikely that an SSO solution would support these protocols on the front-end (that is, for the initial authentication), there may be a requirement to support them against back-end systems.

SSO and Related Solutions

It is important at this stage to consider the desire for SSO as it is the basis for many of the Identity Management projects. Most businesses today are still attempting to implement some form of SSO, which is not a simple undertaking.

Surprisingly for many companies, employees trying to be productive in their daily work waste a huge amount of time in just such a state because of the new productivity services and systems being introduced that utilize separate user and security services from those already existing. Many employees spend large amounts of time obtaining access to services and applications. A number of studies exist around what the costs are associated with the silo approach to applications in today's businesses.

Studies suggest that the average user spends as much as 44 hours per year performing logon tasks. That is just over a week's effort for average employees, and given that it represents the time of only one employee, the problem is grossly inefficient when correlated across the number of employees in an average organization. Given an organization of more than 50 employees in this situation, a year's worth of productivity is lost just through access controls. Of course, there is always going to be time spent authenticating to systems; however, the promise of a SSO solution would cut down on such waste—based on the study numbers, the wasted time would be cut by a factor of four.

Similar issues exist for commercial Internet solutions, whereby a user is forced to manage numerous user names and passwords for different sites. In addition, users are often forced to re-enter credit card details and similar information to enable them to conduct business on those sites. A single solution to enable customers to move from site to site and continue to conduct business without having to bother with usernames and passwords is compelling to most commercial operators.

In this arena is where the potential of SSO begins to show. The combination of maintaining a single identifier and related password that allows access to multiple resources is powerful. There is also the factor that SSO helps in minimizing the need to remember multiple account names and related password and thus the security risks that occur when users are forced to write down that information in order to remember it.

As we begin to look at SSO, you should also note that whilst previous security risks are minimized, new risks become more prominent and need to be dealt with through a thorough review with your security group. The primary risk that is increased is the fact that there is now a single gate or access mechanism that needs to be broken to gain access to resources. Mitigation of this risk and other risks is discussed in the implementation discussion later in this chapter. However, to give a basic example, consider how you use Automated Teller Machines (ATM). To gain access to your account, you are required to present your card to the machine, which then prompts you for your PIN. This security solution is when dealing with simple account/password issues. By adding the requirement that the person accessing the system actually hold something physically, there is a combined protection mechanism.

SSO is also a common solution for remote access. Often remote access solutions are more complex than the simple scenario we painted for ATM access, employing smart cards or tokens such as iKey and SecurID, but also utilizing intelligence in the card or token itself. You might already use such solutions in your work, for example, if you use your building security pass not just to get into your office but also to access to your computer or your office remotely. Of course, if you forget your badge, this solution seems a little limiting.

SSO Basics

Each application requires a different logon account and password, and is generally managed through a separate administration interface. Effort is spent by each staff member in recalling this information, actually entering the logon information into each system, correcting mistakes, and attempting to maintain some sort of synchronicity between those systems. As a result, rather than matching a security desire for complex secure passwords, users are forced into trying to short-circuit the requirement by trying to select easy-to-remember passwords. Worse still, when mechanisms are installed to enforce secure password selection, users generally resort to using notes taped to the side of their computer screen listing the accounts and passwords they use. Doing so breaks multiple security tenets and makes the organization vulnerable to costly security breaches from both internal and visiting persons.

The same is true for administrators who must configure application services to utilize multiple security services to operate. The potential of an SSO solution is to allow for a comprehensive foundation to support enterprise-wide access for users to infrastructure services, while at the same time, enhancing the ability to provide *n-tier application solutions*. Although many are familiar with the client/server as a 2-tier architecture, many applications go well beyond this setup with many possible services and proxies, or additional tiers, between the user and an actual server or service the user wants to use. The importance of Identity Management is to ensure as much as possible that the right account and access information is in the right place across such architectures. This solution is possible by creating an environment in which all security and policy information is related through standards to a directory-based infrastructure supporting multiple clients, servers, applications, and services. Further, through a proper Identity Management solution, the administration of that account is much simpler when issues do arise. The costs of support for forgotten passwords can be immense, although hidden. Recent studies suggest that almost half of all service or Help desk calls are related to password or account lockout issues, and cost an average of \$80 each.

A number of products offer the promise of single logon. In the common security schemes, an identifier, such as user or application name, and a password is used to authenticate to a system. The reality of most SSO solutions today is that they offer a synchronization of user names and passwords across a defined group of systems.

A true SSO solution requires that all applications and systems stipulate a single source for security services. This includes the commonly referred to services of authentication, authorization, encryption, connectivity, and management. This is an extremely tall order for many application developers. Realistically, such a solution will not happen in the near future; however, by enforcing and representing such a model internally, vendors will be coerced into supporting such centralized services.

A common security threat to organizations occurs when users are given too many identifiers and passwords to remember—they often find the only way to retain the combinations is to write them down somewhere. As I have mentioned more than once, this occurrence is a significant security risk and provides one of the soft benefits for an Identity Management solution and related SSO requirements. Although you could place a value on some of the information, until there is an actual breach of security, many companies are not considering their exposure in these situations.

As I have previously mentioned, an SSO solution creates a potential security risk because it sets up a single point of entry relating to security. By having only one gate to guard, an organization might feel that it has less of a security risk. Such is not the case. What it means is that an organization can now dedicate its security resources to dealing with more complex problems rather than worrying about the yellow sticky note syndrome.

One of the many duties that engage administration staff is in defining and implementing security policies. In defining an SSO service, the obvious benefit is for the end user or client of the service. There is, however, cost savings to be gained through a comprehensive implementation of the SSO security model across a wide heterogeneous environment. The simpler and more standardized security model makes the administration model simpler and more consistent for administrators. More time can be spent on correctly defining policy and procedure than is spent on dealing with system limitations across multiple application and security environments.

Another common approach to SSO as well as the integration of policy controls is to utilize an enterprise security administration product such as IBM's Tivoli and Computer Associates' Unicenter. Such applications allow an organization to coalesce their various platforms and applications into a single managed interface, define rules to be applied across the enterprise, then distribute administration to selected staff in charge of application security.

The issue of a common mechanism to define and manage access control across systems and solutions is a long way from being solved. As a result, there will be, in the meantime, a need to utilize the best of breed solutions to implement SSO across existing platforms. In the short term, it is unlikely that there will be an enterprise-wide solution for SSO that works across legacy platforms as well as the new environments being introduced. This is protracted by the fact that many common application vendors are still waiting for a standard to emerge in order to implement their common sign-on solutions. Large-scale enterprise planning should look for solutions that minimize the number of logon identifiers and passwords that a user has.

In the early 1990's the Distributed Common Environment (DCE) received quite a lot of attention, and one of its offerings included a common security infrastructure. Despite the level of effort that went into engineering the solution, it did not gain the market penetration that IBM had hoped for. It is still around today in a number of large installations. Out of this effort, came an appreciation of work done on solutions such as Kerberos, SAML, and beyond. We discuss these solutions in detail in Chapter 5; however, for now organizations might want to push vendors into using this type of solution for an SSO environment.

Figure 2.2 shows the standard way in which a user signs onto a network operating system (NOS) to access other applications supported by that NOS.

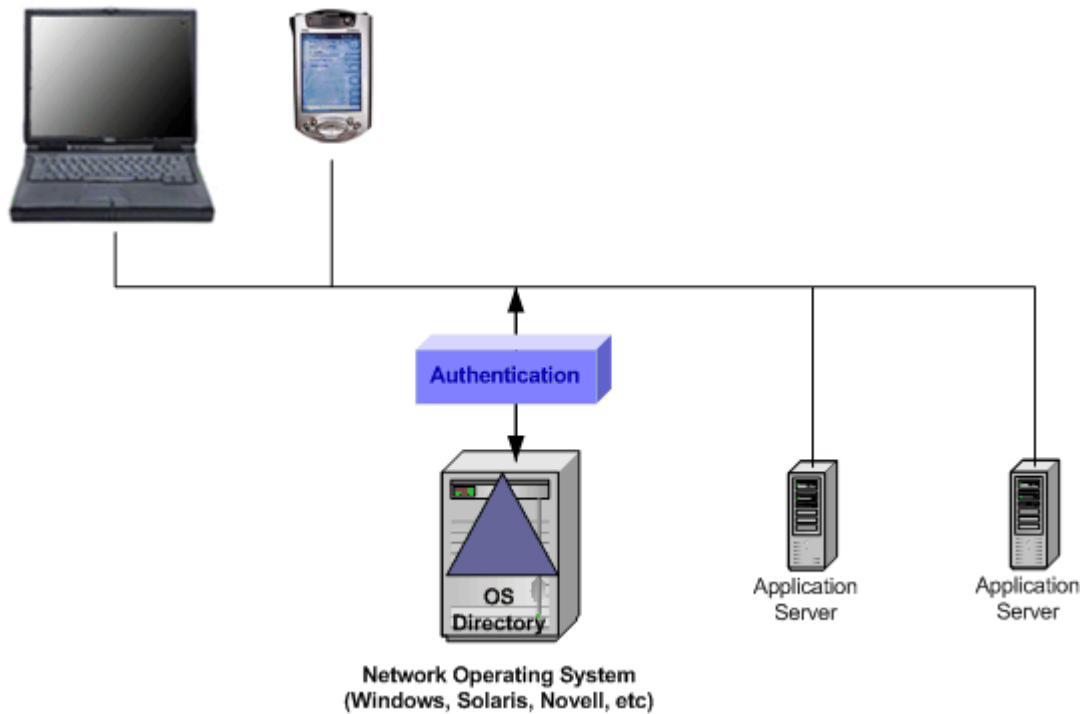


Figure 2.2: A NOS-based SSO solution.

Figure 2.3 shows a Web-based access control solution in which the Web-based authentication manager (WAM) authenticates users or defers authentication to a directory (or other source) and enables a session for multiple, heterogeneous applications.

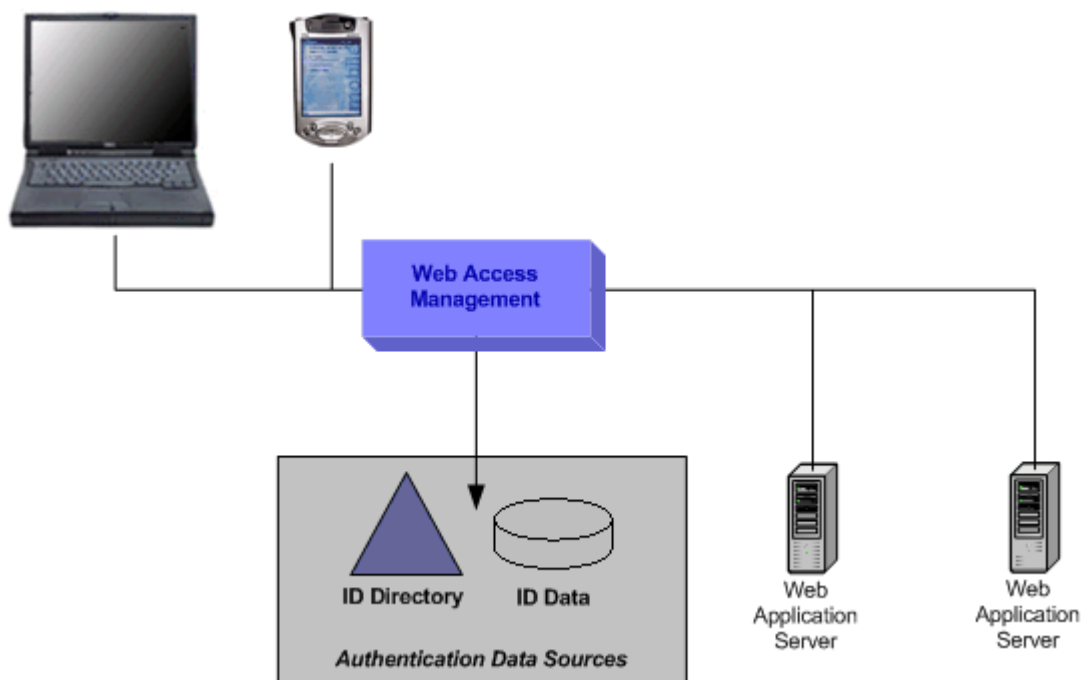


Figure 2.3: Web-based access manager SSO.

Figure 2.4 shows how enterprise SSO tools manage SSO.

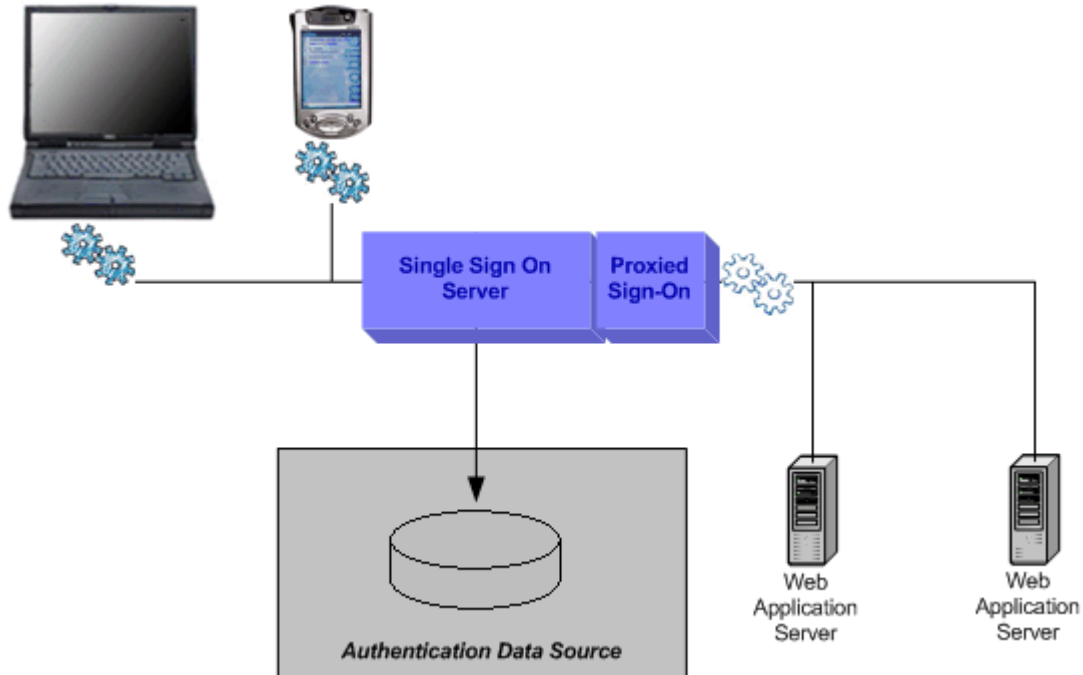


Figure 2.4: Enterprise SSO tools.

Although such solutions are popular, there are a number of issues:

- SSO tools require a “secret store” (encrypted, we hope) for passwords or other credentials in the network
- SSO tools require plug-ins in the applications to support pass-through authentication
- Can result in complex, proprietary solutions that few enterprises have deployed broadly

Figure 2.5 shows a client-side SSO solution that relies on proxy based sign-on.

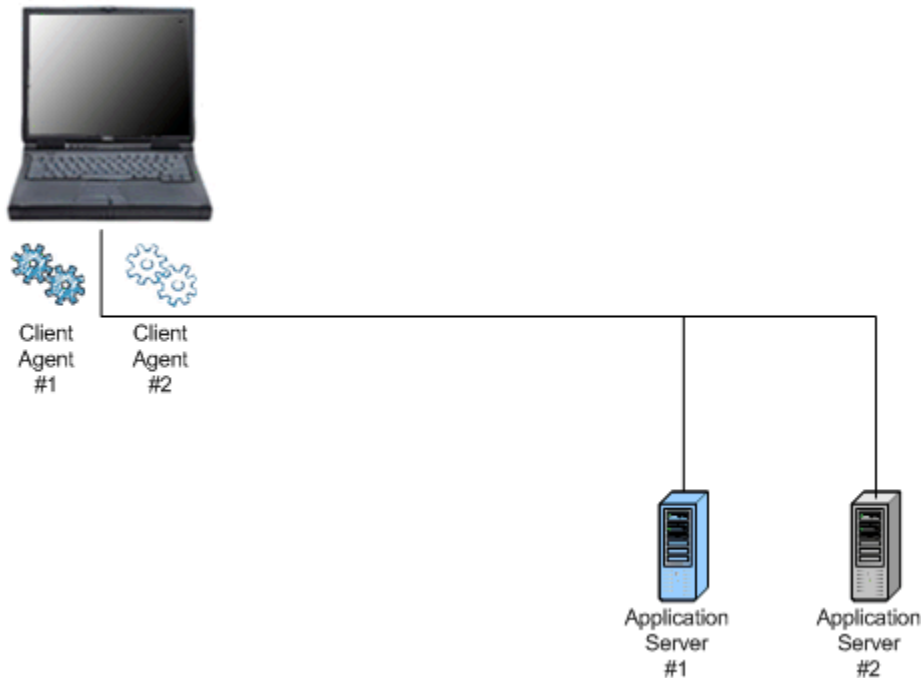


Figure 2.5: Client-side SSO.

Using client-side password capture software, the client program proxies user sign-on to NOSs, mainframes, and client/server applications. The considerations for this scenario are that it

- Requires a “secret store” for credentials on the client
- Requires significant scripting to implement
- Is best used in low-security situations or combined with stringent desktop security measures

Figure 2.6 shows a password synchronization solution.

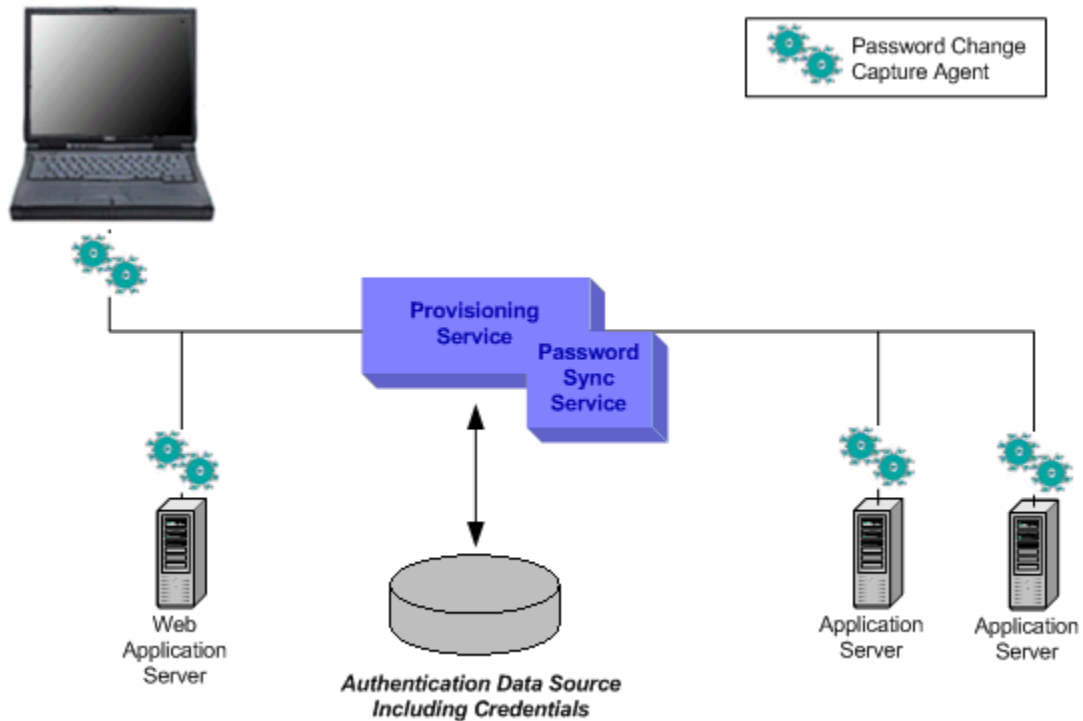


Figure 2.6: Password synchronization option.

The scenario illustrated in Figure 2.6 shows a password synchronization solution that provides a relatively transparent user experience wherein passwords created in any supported system are propagated to others. This is sometimes a custom project or dedicated software solution with the following issues:

- Password sync requires a “secret store” and plug-ins to supported systems
- Often unable to manage a consistent password policy across applications.

A similar solution is provided through password reset tools (see Figure 2.7), which propagate password changes made at a Web interface to supported systems. These solutions can manage password recovery, history, and expiration consistently. The only downside to this scenario is that users must be trained to use the password reset interface.

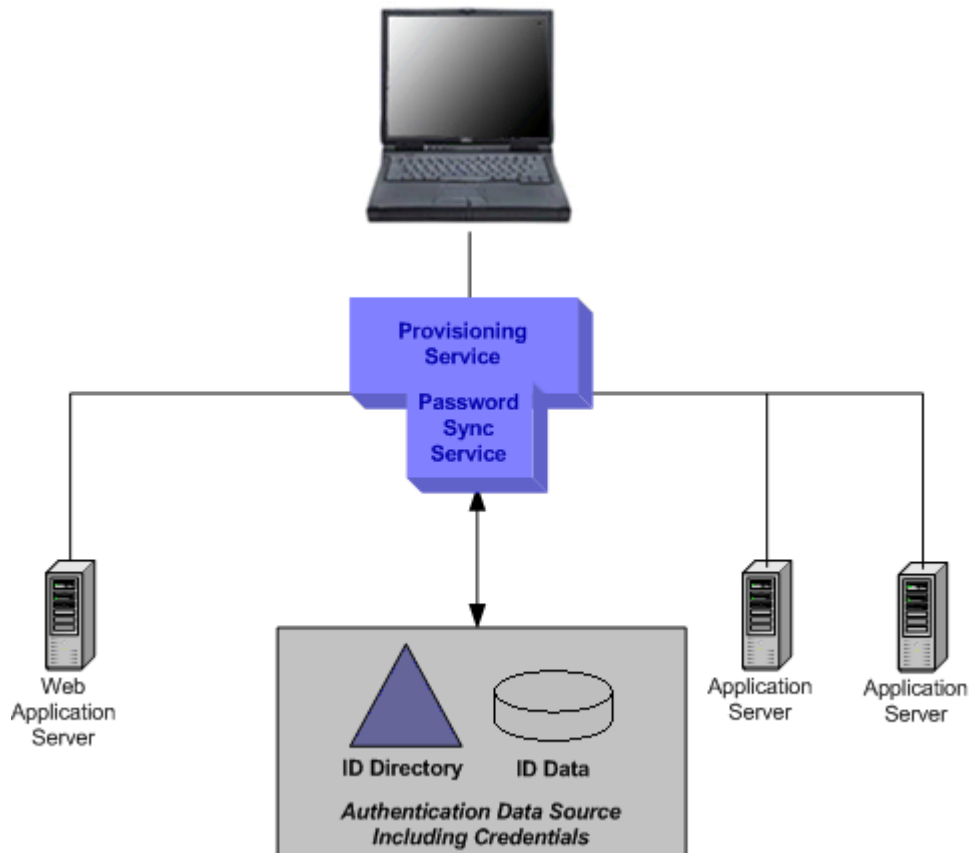



Figure 2.7: Password synchronization option 2.

The best alternative to this scenario is a meta-directory that can allow for the rules and policies to be easily defined within the provisioning system space. Given that all of these options are available, the choice of which solution to implement can be confusion. However, keep in mind that the goal is to create the right solution for your organization.

 Chapter 4 is about implementing Identity Management, and I will cover the methods to get to the right solution in more detail then.

Understanding these options is important in respect to understanding where the enforcement is done within your environment. In most of the examples provided, you will find that individual systems are still making authentication and authorization decisions within their own scope. This idea leads directly to understanding policy evaluation, and what it means to an Identity Management solution.

Policy Evaluation

It is important to understand where your Identity Management system will evaluate and enforce security policies within your system. The quicker you can evaluate the identity of who is trying to gain access to the system and determine the level of access allowable, if any, the quicker you can exclude that request from chewing up resources in the system. If enforcement can ensure that only valid access is gained at the authorization layer, the system will not expend extra effort. Another dimension to consider is how often credentials and access needs to be re-evaluated.

To improve performance of many systems, there is usually a policy to define how often the validity of authentication needs to be evaluated, and potentially for the user to be challenged again. Sometimes this is after a period of inactivity, but in highly secure environments, there is a need to ensure that the account used for access is still valid.

Access Control

Managing the means of access as well as providing clear capabilities based on attributes assigned to an individual or account is known as role based access control (RBAC). There are several fairly new standards to choose from, however, there are minimal implementations or compatibility with the standards to really make them useful. The National Institute of Standards and Technology (NIST) offers an RBAC reference model and The Organization for the Advancement of Structured Information Standards (OASIS) offers eXtensible Access Control Markup Language (XACML), an XML specification for expressing policies for information access over the Internet.

It is also important to note that few identity management players support a complete RBAC solution. The primary issue is that unlike a simple authentication model specific to an application, RBAC requires significant effort up front to design a model that works for an environment that is specific but flexible enough for an organization to live with. This can create significant changes to how an organization functions.

The initial goal of an organization is to deliver some form of a delegated administration capability. If you plan to deliver this, you must consider the ROI of creating a real RBAC solution. Chapter 4 will discuss the ROI of Identity Management including RBAC.

The NIST core RBAC offers a good overview of what is desired in an RBAC solution at <http://csrc.nist.gov/publications/nistbul/cs195-12.txt>. The NIST component defines five basic data elements:

- Users
- Roles
- Permissions
- Objects
- Operations

The whole model is defined in terms of individual users and permissions being assigned to roles. Within an Identity Management solution, these objects can be considered as follows:

- Users—An entity that uses the system
- Roles—A job function within the context of an organization
- Permissions—Approval to perform an operation on one or more objects.
- Object—Can be many things; for example, an entry in a target system (such as an account), a network resource (a printer), an application (a procurement), a policy (password policies), and so on
- Operations—Various and unbounded but including customer-defined workflow processes such as a password reset, the addition, modification, or removal (deletion) of user accounts, and specific data about those accounts; importantly, it should be possible to delegate these operations to other users

Delegated administration allows a chain of approvers to be identified to securely delegate capabilities (even roles) such as account provisioning to appropriate parties in the environment. Delegated administration allows the offloading of the responsibility for user management to those who know their users best, increasing administrative efficiencies and reducing the level of staff required at the central site. If an administrator has a delegable right on a user profile, he or she should be able to delegate that authority to another administrator. Similarly, the ultimate in delegated administration is to allow individual users to perform certain administrative tasks on their own accounts, the most finite example being password management. Delegated administration makes it possible to

- Decentralize administration by breaking down responsibilities among administrators by geographic location or area of expertise (account creation, password management, security, and so on).
- Delegation of responsibilities to authorized users (managers) who are best suited to assign and monitor the responsibilities of the users that they work with.

In general, when looking at Identity Management solutions, it should be possible to delegate all permissions, including the ability to

- View, create, modify, and delete users
- Change passwords
- Add or delete a user in a security group
- Approve or reject requests

Hierarchical RBAC

Hierarchical RBAC requires the support of role hierarchies, whereby senior roles acquire the permissions of their juniors, and junior roles acquire the user membership of their seniors. The NIST standard recognizes two types of role hierarchies.

- **General Hierarchical RBAC**—Arbitrary orders and relationships between roles serve as the role hierarchy.
- **Limited Hierarchical RBAC**—Restrictions are placed on the role hierarchy. Typically hierarchies are limited to simple structures such as trees or inverted trees.

Although General Hierarchical RBAC introduces potential problems of hierarchy loop detection and/or prevention, it is seen as the most useful.

In an RBAC solution, consider that occupants of the same roles at different locations in an organization will need access to different underlying systems. This allows the same role (for example, Development Engineer) to be given access to different systems based on differing values in the role occupant's profile. So while all development engineers need access to source control, it is likely that those in one office or working on one product may need access to a different source control system from those in another office or working on a different project. To solve this problem, a parameterized permission object can be used. A single permission Source Control Access might be used. However the mappings from that object into the connected systems (that is, source control systems) would vary based on a user's location attribute or on the project attribute.

The alternative to parameterized roles is to use scoped roles. In this approach, the roles are defined with a direct scope. If we consider our previous example, instead of having a single role of Development Engineer, there would be locally scoped versions of this role (for example, Application Development Engineer and Interface Development Engineer), and these locally scoped roles would be linked to locally scoped permission objects (for example, Application Source Control Access and Interface Source Control Access).

The tradeoff between the two schemes is that in the scoped roles model there will be more roles to be defined whereas the parameterized roles model has fewer roles but is more complex to develop and potentially more complex to administer (depending on how good the provided administrative tools are). It is worth noting that the two models are not mutually exclusive. That is, both can be supported at the same time and used in a complementary fashion. Also, there are no functional limitations associated with either model—both can be used to get the job done. So, when evaluating options, obviously focus on what best fits with your requirements.

Auditing

Reporting and audit controls are an important part of Identity Management. For example, HIPAA, discussed in Chapter 1, requires organizations that deal with personal data to track all access to that data. Thus, not only does a record need to be made of each access to a record system but also any data transfer, change, and deletion. Financial Services requires tracking and produces information that can be considered for forensic research.

Forensics

Forensics is the next step following auditing, wherein once something has been audited, there may be a need to dig deeper and potentially recover or reproduce events as they happened across systems. Identity Management solutions can compound the problems that they were intended to solve unless careful thought is put into the requirements of the system and the specifications. There is an expectation that once a system goes in that it will be able to consolidate and view activities across the applications it manages. This collides with the goals of monitoring solutions and a recent crowd of solutions with the goal of providing intrusion detection.

Identity Management solutions can only help if they are able to extract and divine operational information about various activities through audit and reporting capabilities. Aside from that, forensic data collection must span all relevant boundaries, especially in cases of federated identity.

Accounting

A final component of security is accounting. When charges are made based on access to resources, usually in commercial environments such as ISPs, ASPs, and so on, a mechanism is required that can track usage and feed that into some form of billing database. This type of requirement fades in and out in many enterprises who decide to charge-back internal system usage to cost centers or business departments in order to effectively manage costs. This should be a separate database from the audit reporting.

Billing records may be generated on the fly or may be derived from provisioning and usage information. It is important to maintain a consistent method for achieving billing for customer records and dispute resolution, as well as any audit requirements.

Policy Management and Enforcement

Moving on from authentication, authorization, and access controls there is the issue of how these requirements are enforced throughout the system. Consider a system as a large office building. Access is granted at the front door based on an employee badge. Assume for the moment that the employee is who they say they are. What happens if the employee quits or is let go, and the employee card is not taken away? Does that ex-employee still have access to the building? Can the ex-employee still get into the building? What can the employee get access to while in the building?

Privacy

Within commercial enterprises, there is a significant contention between the desire for personalization and privacy. For example, consider portals such as Yahoo, AOL, and so on. To provide any degree of customization based on desires of the individual requires that preferences and personal information be stored. The trouble with maintaining that information is that it may be potentially traded with other parties.


Although the government may change the levels of compliance required over time, the HIPAA privacy regulations are in effect April 14, 2003. HIPAA is a good requirement to systematically analyze because it provides a key example of legal regulation in place to protect specific customer's data—in this case, for healthcare patients. In addition, HIPAA has a broad-reaching effect to any organization that might have a need, or obligation, to have access to such customer data. HIPAA requires that protected health information be neither used nor disclosed without permission of the individual customer, except for healthcare treatment and payment and when required by the healthcare operations of a group health plan.

The issue of trust is key in understanding why HIPAA came into being. Because access to healthcare data has become available through electronic systems and the security around that data was inconsistent to the point of causing harm to consumers, the government stepped in to provide a legal framework for that protection, in effect forcing a trust model into the healthcare industry in the USA.

The importance of the government regulation is an example that if organizations begin to share customer or employee data without managing the compliance issues in some way, either technically or manually, then the governments will step in. In fact, there may in some regulation cases already in place bent toward supporting or confining the exchange of user data inter- or even intra-organizationally in many countries.

Federation and Federated Identity

The concept of federation is used when disconnected systems or enterprises need to interoperate with each other's concepts of identity. This has become a more specific use case for the concept of creating secure extranets. But what is federation? The Burton Group in August 2002 provided the following definition "Federated Identity Management [is] the use of agreements, standards, and technologies to make identity and entitlements portable across autonomous identity domains." The goal of federation is to enable transparent and secure exchange of identity information to enable disparate systems to interoperate at the security level.

 Federation creates risk. Federation requires breaks in the organizational border and can diminish any content control you may have in order to facilitate the movement of identity data across those boundaries. This requires that policies be strenuously defined outside the traditional technical bounds, and legal and compliance issues be carefully addressed.

The obvious extension to what we might consider an enterprise solution to a truly federated solution is the need to work across disparate enterprise domains (as opposed to the Windows domain concept). The reality is that the cross-enterprise concept or “solution” is still evolving. The issue with federated identity is that this type of thing has been tried before with varying degrees of success. Why is this “federated identity” thing any different? History shows that very tightly defined solutions (that is, tightly coupled systems) have some success but are either not very flexible or not very simple or are too simple. As the industry works through the convergence issues, the loose links established so far will help enable flexible solutions to evolve.

The first example of this that has drawn a number of vendors together on the same road is the Liberty Alliance. According to the Liberty Alliance (Liberty Architecture Overview V1.0, 11 July 2002):

The Internet is now a prime vehicle for business, community, and personal interactions. The notion of identity is the crucial component of this vehicle. Today, one’s identity on the Internet is fragmented across various identity providers—employers, Internal portals, various communities, and business services. This fragmentation yields isolated, high-friction, one-to-one customer-to-business relationships and experiences.

Federated network identity is the key to reducing this friction and realizing new business taxonomies and opportunities, coupled with new economies of scale. In this new world of federated commerce, a user’s online identity, personal profile, personalized online configurations, buying habits and history, and shopping preferences will be administered by the user and securely shared with the organizations of the user’s choosing. A federated network identity model will ensure that critical private information is used by appropriate parties.

The key thing to realize about federation and the way in which the standards are evolving is that there is still a considerable amount of work that needs to happen “out of band.” The out-of-band requirements are essentially the agreements that are set up ahead of time around a specific goal and physically signed and executed to ensure that there are resolution processes in place should something unacceptable happen during federated interactions. For example, what if account data, such as spending limit, is not updated by one partner in the process, which causes a financial loss to another party, who allows a purchase to be made or a service to be used based on that out of date information. Who is responsible? This is not something dealt with by current standards such as SAML.

SAML uses the communications defined through Simple Object Access Protocol (SOAP) and XML to exchange authentication and authorization assertion tokens between domains. In a general sense, domain A can undertake an authentication and assign that user rights within its own system. From that, domain A may assert to another domain what it has done. In this case, domain B might be willing to accept that assertion and allow certain actions to take place for the user within its own environment. For example, an online travel agent can assert that a person has authenticated with them and has bought a ticket. An airline may take this assertion and derive that the user has rights within its own system to perform some actions, such as pick a seat. Essentially this is a variation of SSO, in a federated model. This is the basis of the work done by the Liberty Alliance Project.

The level of success that Liberty and similar alliances may have is based on concepts we discussed at the beginning of this chapter: trust and risk. How much do you trust the assertion being provided and what or how much are you willing to risk based on that assertion?

Industry groups such as credit card companies deal with this type of situation by setting up a framework defining standards and principles within which “members” operate. Importantly, this includes resolution processes in the event of some failure. For federation, this type of activity is the goal of the PingID Network (<http://www.pingid.com/>).



The goal of the PingID Network is that members instantly benefit from access to standardized business operating rules and regulations, privacy policies, and dispute resolution procedures.

Summary

Chapter 2 introduced the remaining concepts that you need to be familiar with Identity Management solutions. As stated at the beginning of Chapter 1, the goal was to define the concepts and terms used in the field of Identity Management, and this has continued through this chapter. The importance of this is to ensure that before introducing the bulk of the technical solutions and standards available for Identity Management solutions, a common vocabulary is understood.

This foundation sets the groundwork for the review of Identity Management applications in Chapter 3. Remember that some of the concepts, and more importantly standards, are still being formulated, so applications that claim they offer certain functionality deserve close scrutiny if you plan to rely on and deploy them.

[Editor’s Note: This content was excerpted from the free eBook *The Definitive Guide to Identity Management* (Realtimepublishers.com) written by Archie Reed and available at <http://www.rainbow.com/insights/ebooks.asp>.]