Enterprise Digital Identity Architecture Roadmap

Technical White Paper

Author:
Radovan Semančík

Date:
April 2005 (updated September 2005)

Version:
1.2

Abstract:
This document describes the available digital identity technologies and their role in the enterprise identity management. Front-end and back-end identity management integration approaches are shortly described and the recommendation for phasing of digital identity projects is given.

nLight, s.r.o.
Súľovská 34
Bratislava, Slovakia
www.nLight.sk

1 Introduction

The beginning of knowledge is the discovery
of something we do not understand.
-- Frank Herbert

From the very beginning of computer security, password was always the key. Many people still remember times when they had one password that had to be strong and they must change it every other month. Then another system was deployed that needed another password. And another. And then the Internet came ...

Now it becomes clear that the situation in the enterprise information systems as well as on the Internet is becoming critical. People have too many access accounts, too many passwords, cards, access calculators or other types of credentials and attributes. It is difficult to keep a track of these by the holder himself, not to mention poor security officer, that has to keep an eye on thousands of users.

The idea to consolidate user security management is almost as old as the first computer network, but only recently it got enough public attention. Many different technologies sprung to existence to address the aspects of this identity management crisis. This document describes the most interesting of these technologies and provides an overview of how they fit together. Also, few architectural recommendations are given to guide a successful deployments of the digital identity infrastructure.

2 Digital Identity Technologies

It is a capital mistake to theorize before one has data.

-- Arthur Conan Doyle

Many partial solutions to the identity-related problems were proposed during last years. Following paragraphs describe each of these technologies:

Directory Service is a structured database that stores data in more or less standardized fashion. The most today's directory services are accessible by Lightweight Directory Access Protocol (LDAP)[1] and follows the RFC 2798 (inetOrgPerson) schema. The directory services have many useful features, they usually implement fine-grained access control mechanisms and are sometimes used as a simple authentication servers. But as the directory service is only a database and it's primary reason for existence is a data storage, it's use for other purposes is limited.

Metadirectory is a data synchronization service, that keeps in sync several instances of directory databases. The metadirectory synchronization engine copies the relevant data from one directory server instance to the other, when the data modification is detected. The synchronization may be controlled by the set of simple rules, that in most cases only limit the scope of replication. Some metadirectories can even synchronize data with relational database tables and files on filesystem, but the data transformation features are usually very limited. Metadirectories allow simple merging of data from several similar sources and presenting a consolidated view of the data. Metadirectories are usually much faster than user provisioning systems and are used in situations with large amount of data in stored in similar formats.

Virtual Directory is essentially a protocol translator that makes directory data available to other applications (e.g. relational databases) and vice versa. Virtual directories allow directory-aware applications to access legacy database table as a directory service, or it may allow legacy application to see directory service as a table in remote relational database. Some virtual directory systems allow basic data transformation and caching. By using direct access to data instead of replication, virtual directories allow to overcome inconsistency problems that can be experienced with metadirectories and user provisioning systems.

User Provisioning System is an active tool that remotely manages user databases on many heterogeneous systems. User provisioning systems are usually quite complex and featureful, allows complex data synchronization and transformation, management and auditing. Most of the user provisioning systems implements rich workflow engine to control the data flows. Some systems use central internal database, other use data virtualization techniques similar to virtual directories. Some provisioning systems use agents on target systems, some are agentless and use only available remote interfaces. The metadirectories and virtual directories may show much better performance characteristics, but they cannot handle anything else but the most simple situations. The user provisioning systems were designed to make and maintain an order in chaotic user databases of existing information system's deployments, and are well suited for the task.

Access Management System contains agents, that are deployed on end systems and control user interactions with the end system. For example an agent may control UNIX login process or the access to the web server. The access management module may reject the user accessing the end system, may require additional authentication, etc. The access management system may implement security policies that are not implemented by the end

1 The LDAP-based directories are so common, that the abbreviation “LDAP” became a synonym for directory services.

system itself, which none of the previously described systems can. But the use of agents that have hooks deep in the end system has many drawbacks. The agents are usually bound the the specific version of end system, which may upgrades very painful. Also many applications are not tested with the agents and the presence of agent on the end system may void the application warranty and/or support requirements. The access management systems may be beneficial in some cases, but are not suitable to be deployed enterprise-wide.

Public Key Infrastructure (PKI) is an infrastructure, that uses digital identity certificates secured by public key cryptography. There exists several systems that use similar principle, but most common commercially used systems are built on X.509 standard. These systems use digital certificates to bind user identifier to his public key by the trusted third party (certificate authority) signature. The user identifier in X.509 public key certificates was originally supposed to be a directory distinguished name, but that meaning evolved to be a couple of unstructured attributes. The X.509 certificates are now used in several protocols (SSL, TLS, IPsec, etc.), but the subjects that holds these certificates are usually computers or devices, not people. PKI can be used to implement single sign-on, but for this to work all participating applications has to be modified to support PKI (both server and client). The use of X.509 certificates has also a privacy concern, as all the attributes in the certificate are revealed when the certificate is presented. This was partially addressed by *Privilege Management Infrastructure* (PMI) that uses short-lived attribute certificates in addition to the long-lived public key certificates. Both PKI and PMI has many drawbacks, the deployments are complicated and the systems tend to be quite inflexible. Nevertheless, the PKI is still the only commonly used system that can provide practical non-repudiation (under special circumstances).

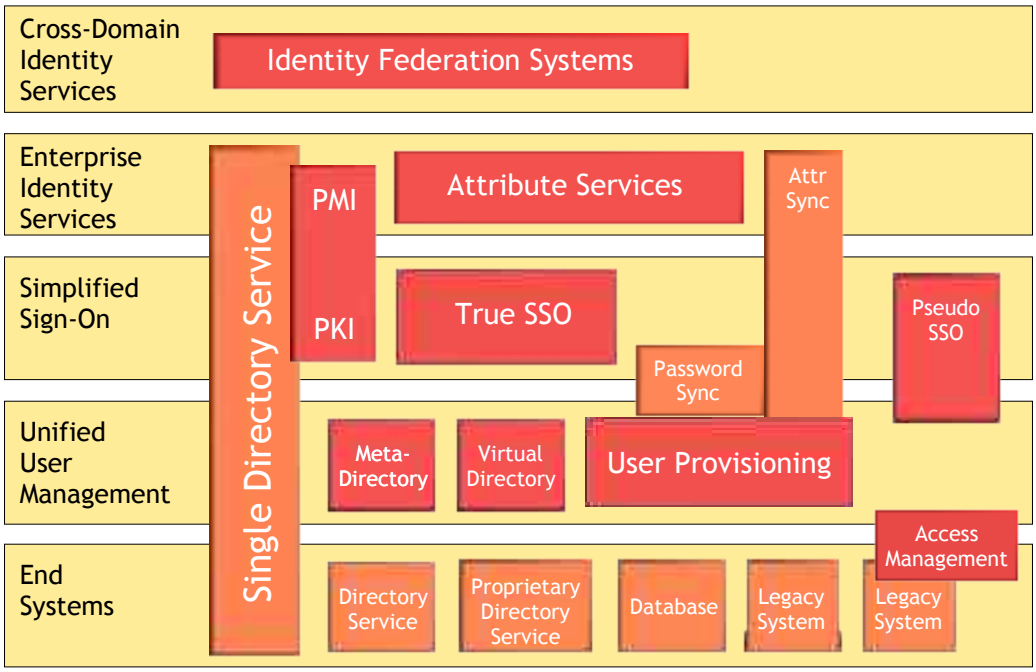
Pseudo Single Sign-On System is a tool, that stores user credentials for different systems. This tool is usually deployed on the end user workstation and waits for a login window to appear. When it appears, it automatically fill in user login and password (or other credentials) and submits the window. This method makes an illusion of single sign-on, while the user is essentially still signing-on to each system. The drawback of pseudo single sign-on system is limited user mobility and, in some cases, the risk of exposing the password database.

True Single Sign-On System authenticates user only once and transfer the information that the user has authenticated to the other systems. Majority of the true single sign-on systems are based on tickets, that are issued by the authentication server and consumed by the services. Older SSO systems were based on symmetric cryptography (e.g. Kerberos), newer systems use XML and public key cryptography. Some true SSO systems may also be extended to authenticate the user to non-interactive services, e.g. the Attribute Services.

Attribute Services are used by the end system to read and manage user attributes. In this case, end systems do not need to store the attributes in its databases, thus eliminating user accounts on the end systems. In combination with true single sign-on systems, this approach may enable deployment of “pure” services that do not need to authenticate user directly and are still capable to work with user attributes. The attribute services are expected to be a part of broader architectural concept: the Service Oriented Architecture (SOA).

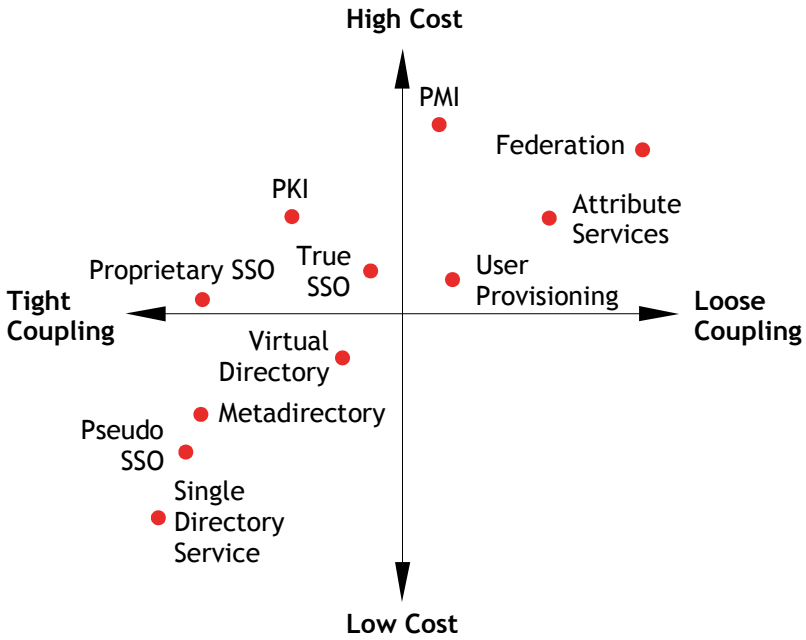
Identity Federation is a business concept, that enables several organizations to share a relevant parts of user identities and authentication status in a secure manner. Federations allow to link user accounts on different service provider sites or to directly use services without a need for an account. This mechanism is designed to work across organizational boundaries and to keep the privacy of users. The federation concept is not yet completed, it is still evolving.

None of these technologies can be regarded as a complete solution, but together they may form a functional identity management system. The different identity management technologies can be assigned to identity management architectural layers, as illustrated on the next figure.



There are many combinations of technologies that can yield good partial results, but in a successful digital identity implementation each of these layers must be appropriately implemented.

The most apparent differences in digital identity technologies are related to the decoupling that the systems allows and the cost of the deployment. The estimate of these aspects are graphically illustrated on next figure.



Systems that allows tight coupling may be good for small or relatively simple deployments. In large deployments these system will tend to be inflexible and the number of supported end systems will be much less than 40% in medium to large deployments. The deployment of tight coupled systems is relatively inexpensive, but their limited scope may render the entire deployed system unusable. Tight coupled system will not be able to provide federation or any sound cross-domain services.

On the other hand, systems that support loose coupling may be more complex and more expensive, but they are usually able to roll out on 80% or more of the end systems. The deployments will provide more flexibility for changing business requirements and better overall return on investment.

3 Implementing Digital Identity

We can't solve problems by using the same kind of thinking we use when we created them.

-- Albert Einstein

The environment in which the identity management deployment starts is far from the ideal. The user management procedures are frequently designed and maintained ad-hoc. The account usernames for a single user are different in several systems without any deterministic link to the user. The organizational structure is fuzzy, the only available automatically maintained organizational tree is an “official” one, that is different from practical competency structure. The end systems lack robust user management interfaces and each of them mandates own user record structure. And all that is without sound documentation of any kind.

There are two fundamental approaches how to address that situation:

Front-end integration is a technique, that tries to present users with a single unified application (usually portal-like) that will act as a “face” (or façade) for all available systems. This is not an easy task in just providing the unified user interface, leave alone the security of front-end to back-end communication. This approach usually employs pseudo SSO or proprietary SSO systems.

Back-end integration focus on the unification of user management procedures in different systems. This usually means the automation of user account creation and modification, centralization of user databases, etc. The user interface is usually left untouched, the most apparent effect for end user is unification of passwords and access privilege request process. The most apparent side-effect of back-end integration is the creation and clean-up of enterprise-wide user database.

It seems that front-end integration may not be feasible for complex medium-to-large systems without implementing back-end integration first. The reason is quite clear: there must be a functional central user database and service security infrastructure for front-end integration to be possible. None of these is frequently seen in today's enterprise information system architectures.

According to this, following steps for successful identity management deployment are recommended:

1. *Assess your identity information repositories.* Make a quick analysis of your information assets and resources: user databases, organizational structure, user management procedures, legislative regulations, etc. The results will help in defining the requirements and scope of following steps.
2. *Deploy provisioning system.* Automate and refine your user management procedures by implementing user provisioning system. Cleverly customized workflow engine that follows the dynamic organizational structure is usually the key here. Deployed provisioning system should provide tools for user database cleanup, the design and deployment of role-based access control mechanisms and end system monitoring.
3. *Create central user database.* Use the deployed provisioning system to create and maintain a central user database, usually as a replicated directory server. This database may act as an authoritative source for the systems in following steps. Alternatively, use virtual directory or metadirectory techniques to create the central user database. The central user database is not central in the way that *all* systems must use it. The database

is supposed to hold authoritative data that *some* important (usually infrastructure) services will use.

4. *Implement single sign-on.* Using the central database implement Single Sign-On as the authentication service. The implementation of SSO may be quite straightforward for web applications and very complex for legacy applications.
5. *Deploy attribute services* using central user database and authentication service. The simplest attribute service may be implemented by sharing parts of directory server with other applications. But as your Service Oriented Architecture evolve, more complex attribute services will be needed, most probably in the form of infrastructure web services. This will allow the deployment of relatively lightweight services, that needs not to maintain their own user databases.
6. *Federate.* Connect your digital identity services with similar services in other organizations by joining a federation relationship or network. Using a federation will allow a better degree of cross-organizational cooperation.

Each step of the digital identity deployment will provide additional benefits and may be deployed relatively independently. But for the project to succeed the final goal must be clear from the very beginning.

4 Conclusion

The only way of discovering the limits of the possible is to venture a little way past them into the impossible.
-- Arthur C. Clarke

The deployment of digital identity technologies will definitely be a major step forward. The early steps will bring a touch of order to the existing chaos. The higher automation level and cleaner design will reduce the cost of user management. This will bring better flexibility, service levels and therefore also a considerably higher workforce efficiency.

The personal information protection laws and other legislative regulations are getting more restrictive every day. The cost of compliance by manual means is getting unbearable. The identity management technologies, even in the early stages may reduce the recurring costs of legislative compliance.

The old monolithic, application-oriented software architectures reached their limits. It will not take long and they will be considered as obsolete as non-networked computer systems. The service-oriented architectures are at their rise. But for the distributed system to be deployed efficiently, vital infrastructure services must exist. The identity services are the critical part here, which only recently got sufficient attention. No complex service-based architecture can be deployed without solid identity services platform.

The services that cross organizational boundaries are the most challenging to architect and deploy. The use of different policies and procedures make it almost impossible to use the same mechanisms as are used intra-enterprise. The federation technologies must make loose coupling the basic operational principle, allowing broad flexibility and still maintaining interoperability. The identity federation and similar concepts will dramatically change the existing electronic business as we know it now.

The road to the age of digital identity will not be an easy one, but it is definitely a necessary one. And the potential benefits of the technology are so attractive, that it is a road worth taking.