Identity, Access Management and Single Sign-On Web-based Solutions

SIHAM RHERMINI

Master of Science Thesis
Stockholm, Sweden 2012
TRITA-ICT-EX-2012:11
Abstract

The systems' complexity growth over wide networks has increased the need for a centralized access control and strong authentication methods. And because systems have become increasingly distributed, additional need for single sign on solutions have become even more relevant.

The aim of this study is to draw the state of the art on strong authentication solutions for protecting resources over networks and establish an evaluation summary of all the solutions presented. The study also presents the single sign on functionality and different architectures to implement it.

This study contains a market review of available commercial solutions that implement strong authentication and single sign on. One of the solution is selected for implementing the authentication service for a payroll management system for one of Accenture clients. The technical details regarding this product are presented including architecture, installation and tests.

The tested product needed custom developments to successfully implement the certificate authentication combined with an advanced certificate validation service. The single sign on functionality requires custom development to be integrated with all the system components.
Foreword

The thesis is subject to confidentiality constraints. For this reason, the client name and the client project name will not appear in this thesis.

This is a 20 points thesis in the area of Information and Communication Technology, in the master Information and Communication System Security at KTH in Stockholm. The thesis has been conducted on site at Accenture in Paris, France during the summer 2010.

Acknowledgments

It is my pleasure to express my sincerest regards to my supervisor, Mr. Bertrand HELFRE, for giving me the opportunity and privilege to be part of the security team and for providing his precious guidance and advices to me.

I owe my deepest gratitude to Mr. Aurélien PERROT DE-LAHOUSSE, who patiently answered my questions and generously shared his expertise and knowledge with me. A big thank you for his welcoming, availability and supervision.

Also, I would like to thank all members of Accenture team for offering me a friendly and enjoyable working environment all along the project period, specially Mr. Salah BEGHOUL.

Last but not least, I express my warmest thanks to my examiner, Sead MUFTIC, who contributed to the completion of this work.
Table of Contents

Abstract ........................................................................................................... 3
Foreword .......................................................................................................... 5
Acknowledgments .......................................................................................... 5
Useful contacts ................................................................................................. 14
Chapter 1 : Introduction ............................................................................... 16
  1.1 Problem statement and Motivation ........................................................... 16
  1.2 Purpose .................................................................................................... 16
  1.3 Context ................................................................................................... 16
  1.4 Thesis organization .................................................................................. 17
  1.5 Abbreviations ........................................................................................ 18
Chapter 2 : Background .................................................................................. 19
  2.1 Cryptography ........................................................................................... 19
    2.1.1 Symmetric cryptography ...................................................................... 19
    2.1.2 Asymmetric cryptography ................................................................... 20
  2.2 Public key Infrastructure ........................................................................ 20
  2.3 Authentication bases .............................................................................. 21
  2.4 Strong authentication ............................................................................. 21
Chapter 3 : State of the art of Strong authentication .................................. 22
  3.1 One Time Password ................................................................................ 22
    3.1.1 Presentation ....................................................................................... 22
    3.1.2 Illustration of the protocol ................................................................... 22
    3.1.3 Scratch list ......................................................................................... 24
      3.1.3.1 Description ..................................................................................... 24
      3.1.3.2 Advantages .................................................................................... 24

Access Management Solution 7/72
5.4.2 Plug-in SSO agent........................................................................................................47
5.4.3 Helper application agent................................................................................................47
5.4.4 Reverse proxy agent.....................................................................................................48

Chapter 6: Commercial products survey for SSO...............................................................49

6.1 EMC: RSA Access Manager..........................................................................................49
6.2 CA: SiteMinder................................................................................................................49
6.3 Evidian: Web Access Manager......................................................................................49
6.4 Entrust: GetAccess..........................................................................................................49

Chapter 7: Implementation....................................................................................................51

7.1 The selected authentication method..............................................................................51
7.2 The Selected product: RSA Access Manager.................................................................52
  7.2.1 Description................................................................................................................52
  7.2.2 Supported platforms..................................................................................................52
  7.2.3 RSA Access Manager architecture........................................................................53

7.3 Implementation in the project context..........................................................................55
  7.3.1 Global architecture....................................................................................................55
  7.3.2 Security Architecture...............................................................................................56
  7.3.3 The authentication architecture................................................................................57
  7.3.4 Installation................................................................................................................58
  7.3.5 Integration and custom developments.....................................................................59
    7.3.5.1 Design................................................................................................................59
    7.3.5.2 Implementation...................................................................................................60
    7.3.5.3 Technical details .................................................................................................60
    7.3.5.4 Test and validation...............................................................................................61
  7.3.6 Demonstration...........................................................................................................61

Chapter 8: Project Context..................................................................................................64
8.1 About Accenture................................................................................................................. 64
  8.1.1 Overview....................................................................................................................... 64
  8.1.2 Organization.................................................................................................................. 64
  8.1.3 Accenture Security Offer.............................................................................................. 65
8.2 Project organization............................................................................................................ 67
  8.2.1 The security team mission............................................................................................ 67
  8.2.2 My mission.................................................................................................................... 67
  8.2.3 The project management............................................................................................... 67
    8.2.3.1 Ticketing tool: ClearQuest....................................................................................... 68
    8.2.3.2 Versioning tool: ClearCase..................................................................................... 68
    8.2.3.3 Documents sharing: Alfresco.................................................................................. 68
    8.2.3.4 Meetings................................................................................................................... 68
    8.2.3.5 Planning..................................................................................................................... 69
  8.2.4 Packaged Development Methodology........................................................................... 69
  8.2.5 Team integration......................................................................................................... 69
Chapter 9 : Conclusion............................................................................................................. 71
  9.1 Personal feedback............................................................................................................. 71
  9.2 Perspectives..................................................................................................................... 71
Illustration Index

Illustration 1: Abbreviations.................................................................16
Illustration 2: Symmetric cryptography process....................................17
Illustration 3: Asymmetric encryption process......................................18
Illustration 4: Example of OTP authentication protocol..........................21
Illustration 5: Example of scratch list...................................................22
Illustration 6: Example of matrix card authentication scenario..................24
Illustration 7: Example of SMS OTP......................................................25
Illustration 8: Authentication protocol using challenge–response OTP........30
Illustration 9: Basic hardware token.....................................................31
Illustration 10: Advanced hardware token...........................................31
Illustration 11: Example of software token...........................................32
Illustration 12: Certificate authentication protocol..................................35
Illustration 13: Example smart card and smart card reader........................36
Illustration 14: Example of software certificate in a browser......................38
Illustration 15: Examples of biometrics..................................................39
Illustration 16: Strong authentication solutions evaluation summary..........43
Illustration 17: Plugin SSO agent architecture.........................................45
Illustration 18: Helper application agent architecture...............................46
Illustration 19: Reverse proxy agent architecture.....................................46
Illustration 20: Authentication solution justifications...............................49
Illustration 21: RSA Access Manager architecture..................................52
Illustration 22: Global architecture of the system.....................................53
Illustration 23: The security architecture................................................54
Illustration 24: The authentication chain...............................................55
Illustration 25: High level description of the SSO modules algorithm..........57
Illustration 26: Certificate use in the browser.............................................................59
Illustration 27: Access denied page.............................................................................60
Illustration 28: RSA administrative GUI.....................................................................61
Illustration 29: Accenture organization.......................................................................63
Illustration 30: Security entity in Accenture organization..............................................63
Illustration 31: Planning...............................................................................................67
Useful contacts

**Student:**

Siham RHERMINI  
10 rue Castiglione  
75001 Paris  
Tél : +33 (0)6 19 06 19 41  
siham@kth.se  
rhermini.siham@gmail.com

**Company tutor:**

Bertrand HEFLRE  
118-122 Avenue de France  
75013 Paris  
Phone : +33 (0)1 53 23 49 35  
bertrand.helfre@accenture.com

**University tutor:**

Sed MUFTIC  
Kungliga Tekniska Högskolan  
Stockholm  
Phone : +46 8 790 4130  
sead@kth.se
1 Introduction

1.1 Problem statement and Motivation

Information systems are becoming indispensable in the modern life, growing bigger every day, getting more complex and interconnected through wide world networks. In the same time, needs to make these systems securely accessible for different users, with different authorization rights increase.

To achieve these objectives, access management solutions were designed to implement user authentication methods and optimize authenticator management thanks to solutions like single sign on.

1.2 Purpose

The purpose of this project thesis is to draw a non-exhaustive state of the art about strong authentication enterprise solutions used in industry. The aim is also to establish a set of evaluation criteria in order to elaborate an evaluation summary of the presented authentication solutions. The evaluation can help security architects choose the appropriate solution matching their requirements.

The thesis also presents single sign on solutions, usually combined with authentication solutions and a list of commercial products that implement this functionality.

1.3 Context

To be best prepared to integrate the working environment, Ensimag and KTH students perform practical trainings in parallel with the courses. These trainings end with the end-of-studies project.

My master project corresponds to my end of studies project at Ensimag and to the double degree master program “Information and Communication Systems Security” at KTH.

I joined Accenture security team for a mission for a prestigious client in the public services domain.

The team has the responsibility to design and deploy the appropriate authentication management solution as essential component of a complex payroll management system, including strong authentication and single sign on.

This was an opportunity for me to participate to all the project phases in an industrial and real context, from design, through implementation, to test and evaluation of the system put in place.

My purposes were to:

- Put into practice all the theoretical knowledge I acquired during my studies.
- Gain professional and personal experiences working in an international company.
- Develop expertise about new technologies and industrial products used in the security field.

As I have spent my last year abroad, project lasts for 8 months, from June 16th 2010 to January 21st 2011.
1.4 Thesis organization

Chapter 1 introduces the project thesis. Motivation, objective and context are laid down. The abbreviations used are also defined.

Technical background information is provided to the reader in chapter 2. In this chapter, technologies that are infrastructure components to authentication and single sign on solutions are briefly presented.

In chapter 3, the state of the art of strong authentication is drawn. This includes methods using one time password, dynamic token, digital certificates and biometrics.

Chapter 4 presents an evaluation of the strong authentication methods previously studied along with the criteria used to establish this evaluation.

Chapter 5 describes single sign on mechanism and architectures.

In chapter 6, a products survey will present some of single sign on enterprise commercial solutions.

In chapter 7, the details of the implementation of the chosen authentication and single sign on solution are laid down. My contribution to the design, implementation and test project phases is detailed.

Chapter 8 is dedicated to the organization of the project and different project management tasks and tools.

Finally, in chapter 9, conclusions and feedbacks of the project are made.
1.5 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Signification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Certification Authority</td>
</tr>
<tr>
<td>OTP</td>
<td>One Time Password</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>PKC</td>
<td>Public Key Certificate</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Socket Layer</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign On</td>
</tr>
<tr>
<td>TAN</td>
<td>Transaction Authentication Number</td>
</tr>
<tr>
<td>TFA</td>
<td>Two-Factor Authentication</td>
</tr>
<tr>
<td>OATH</td>
<td>Open Authentication</td>
</tr>
<tr>
<td>OCSP</td>
<td>Online Certificate Status Protocol</td>
</tr>
<tr>
<td>EAL</td>
<td>Evaluation Assurance Level</td>
</tr>
</tbody>
</table>

*Illustration 1: Abbreviations*
2 Background

In this chapter, technologies that play a significant role in building and operating authentication systems are described. These include introduction to cryptography, public-key infrastructure, authentication bases and strong authentication presentation.

2.1 Cryptography

Cryptography plays a central role in modern authentication and single sign on systems. It is used in various ways from generating certificates, signatures, protecting data and traffic to storing credentials locally in a secure manner. The most common cryptographic transformations are briefly explained below.

2.1.1 Symmetric cryptography

In symmetric cryptography, the cryptographic function uses the same key for encryption of the plain text and decryption of the cipher text as visualized in Illustration 2: Symmetric cryptography process, and thus it is said to be symmetric. Symmetric cryptography is sometimes called shared-secret or secret key cryptography because the encryption/decryption key has to be shared between all parties authorized to access the enciphered data.[1]

Symmetric cryptography is used for the encryption of bulk traffic because it is less demanding on computer resources than public-key based cryptography. Symmetric cryptography is utilized for all traffic encryption tasks over internet and to encipher credentials as single sign on tickets.

The downside of symmetric cryptography is the problem of key management, and the trustworthiness of those involved with the shared secret. Symmetric cryptography also requires a set of protocols for key distribution.

The most common symmetric algorithms are:

• Data Encryption Standard (DES)
• Advanced Encryption Standard (AES)
### 2.1.2 Asymmetric cryptography

Asymmetric cryptography, also known as public-key cryptography, is fundamentally different from symmetric cryptography. The difference between these two is in the level of secrecy of the encryption keys.

In symmetric cryptography, it is of primary concern to keep the encryption key secret. In asymmetric cryptography, one has two different keys, the public key, which can be revealed to anyone and the secret key that is personal and must be kept secret as seen in Illustration 3: Asymmetric encryption process, which presents the encryption process using public keys.[1]

The main difference is that anyone can now send securely encrypted material, which can be decrypted only with the receiver private key – the public key is not able to decrypt the encrypted data – only encrypt it.

The asymmetric cryptography can also be used to authenticate the sender. He encrypts material using his private key. The receiver tries to decrypt the sent data using the public key of the sender. If it succeeds then the sender is the one who has the identity bind to this public key, since he is the only one having the private key associate with the public key.

The RSA algorithm is the most popular and widely used public-key algorithm on the market, named after its inventors Ron Rivest, Adi Shamir and Leonard Adleman.

### 2.2 Public key Infrastructure

The Public-Key Infrastructure (PKI) is the collection of services needed for successful deployment and use of digital certificates stored in smart cards, tokens or file containers. It is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

PKI allows to bind public keys with respective user identities by means of a certificate authority (CA). The binding is established through the registration and issuance process, which, depending on the level of assurance the binding has, may be carried out by software at a CA, or under human supervision. For each user, the user identity, the public key, their binding, validity conditions and other attributes are made unforgettable in public key certificates issued by the CA.
A Public-Key Certificate (PKC) is primarily used for generation of digital signatures, and identification based on a cryptographic challenge-response authentication protocol, as SSL protocol. [1]

2.3 Authentication bases

Authentication establishes a level of confidence that the identity provided (e.g. username) is authentic. Most authentication methods today rely on one or more of the following factors:

- Something you know
- Something you have
- Something you are

A password is an example of “something you know”. To establish a level of trust it must be something that ONLY the user knows. Dependent on the level of risk identified, this form of authentication on its own may no longer be sufficient. More innovative use of the “something you know” factor, such as selection of password/PIN characters from a drop-down list or image selection may help to address part of the problem, such as key logging, but still rely on the same basic principle.

The “something you have” factor requires the user to be in possession of something. This “something” is usually referred to as a token. A token is typically, but not necessarily, a hardware device that has been issued to the user for the purposes of authentication. There are various forms of token which are described later.

The “something you are” factor refers to some form of biometric authentication, based on a measurement of some personal characteristics (which may or may not be physical). Multiple forms of biometric authentication are available, all of which may enhance the level of confidence in the authentication process. At present the applicability of biometric authentication to the on-line community may be limited, with it being better suited to a more closed community. Not all biometric authentication methods require dedicated or expensive hardware; however all do require some initial measurement to be taken during registration. [10]

2.4 Strong authentication

Strong authentication is associated with two-factor authentication or more generally multi-factor authentication.

Two-factor authentication (TFA) means using two independent authentication methods based on two independent authentication factors (e.g. password + value from physical token). This is in order to increase the assurance that the user has been authorized to access secure systems and to improve protection in a way that the compromise of either one of the factors on its own would not be sufficient for an attacker to gain access.

The chapter State of the art of Strong authentication lists strong authentication methods with the assumption that these methods are also combined with simple authentication method, for example user/password or user/PIN authentication. [12]
3 State of the art of Strong authentication

The chapter State of the art is dedicated to draw a non exhaustive picture of the solutions deployed in industry for strong authentication for users, mainly over web interactions.

3.1 One Time Password

3.1.1 Presentation

The principle of this authentication method is to use a password that is valid for only one session or one transaction. Thus this password is called One Time Password (OTP).

The OTP is transmitted by a chosen way to the user in order to represent a “something you have” authentication information. Combined with another authentication method, it is considered as strong two-factor authentication.

Unlike the basic password authentication, OTP method is not vulnerable to replay attacks. This means that, if a potential intruder manages to record an OTP that was already used to log into a service or to conduct a transaction, he or she will not be able to abuse it since it will be no longer valid.

On the downside, OTPs are difficult for human beings to memorize. Therefore they require additional mechanisms in order to work.

To insure the uniqueness of the used OTP, they are generated using one of the following methods:

- Randomly, the set of random generated passwords is known by the authentication server. Every time a password is used, it is dropped off from the authorized passwords set on the authentication server.
- Based on time synchronization between the authentication server and the client providing the password (OTPs are valid only for a short time period).
- Using a challenge-response mechanism.

The present section details different basic technologies and mechanisms used in industry to implement random generated OTP authentication.

Section Dynamic token describes token authentication mechanisms using OTPs generated based on a challenge-response input or time synchronization.

3.1.2 Illustration of the protocol

The following figure illustrates an authentication protocol using OTP.
Illustration 4: Example of OTP authentication protocol
3.1.3 Scratch list

3.1.3.1 Description

This authentication method is based on a list of randomly generated passwords, named scratch list and issued to each customer. The list is kept either on the authentication server side and on the user side.

Usually this method is used in a multi factor authentication process, the user first enters his or her username and password in the established manner. Assuming the information is entered correctly, the user will then be asked to input, as a second authentication factor, one of the passwords contained in the scratch list provided to him. The server will validate the provided password by verifying that it is present in the database. If the verification successes, the used password will be removed from the server side.

By this mean, if the user or an attacker tries to reuse the password, the server will not recognize it.

The same principal of scratch list was used in the phone credits recharging mechanisms: the client buys a credit recharging card, he scratches the hidden password and discovers the recharging code. Dialing this code allows the server to transfer credit to the user account with the card value.

In the banking environment, the method is called TAN list (Transaction Authentication Number).

The scratch list is commonly printed and distributed to the customers on physical mediums: sheet, books, cards...

3.1.3.2 Advantages

- Low cost of deployment mechanisms.
- Simple usage of the list implies no need to train users.
• The solution is very simple and does not need any complex training.

3.1.3.3 Disadvantages

• Easily compromised since the scratch list is not well protected (usually clear text or hidden area to scratch).
• This method offers only user authentication and none of the other security services (e.g. non-repudiation).
• The current implementations of this method are starting to be obsolete. Thus the maintenance and support are not guaranteed.
• The scratch list has a relatively short life time.
3.1.4 Matrix card

3.1.4.1 Description

This authentication method is based on an OTP extracted from a matrix card, also called grid card. The OTP corresponds to the coordinates sent by the authentication server to the user. The authentication matrix is unique and the numbers and letters position is totally random. The matrix can either be:

- Downloaded and stored on the user workstation for the authentication.
- Distributed to the user on physical medium, on the employee badges for example.

A scenario of matrix card authentication can be as follows:

Before he attempts to login, each user is provided with a personalized matrix of numbers and letters. The authenticating server knows the characters on a given user's matrix. During authentication, the user is challenged to input entries from particular cells on his matrix, corresponding to the coordinates sent as challenge. If these responses match the known and assigned matrix entries, the user is authenticated.

Illustration 6: Example of matrix card authentication scenario

3.1.4.2 Advantages

- Reduced cost of the solution and high flexibility.
- Easy to use for the users and thus need no training.
- Offers an acceptable security level compared to the deployment cost.
- Offers more possibilities than the scratch list and then more flexible expiry policy.

3.1.4.3 Disadvantages

- Because the matrix card is deployed on physical mediums, the method suffers from the same vulnerabilities as the scratch list in a way that the authentication is easily compromised.
- Solution for user authentication only and doesn't offer non repudiation service.
3.1.5 SMS OTPs

3.1.5.1 Description

An appealing alternative to deploying tokens is to use something the customer already has, their mobile phone. In this case, the authentication server generates the OTP and sends it to the customer as SMS message. The customer returns the OTP to the authentication server through their web browser. [9]

![Illustration 7: Example of SMS OTP](image)

This method is being used in the banking environment for user authentication and also for transaction authentication. [8]

3.1.5.2 Advantages

- The SMS can include a description of the operation processed; for electronic payment for example it can include the amount the user is paying. In authentication scenario the SMS, can include the identity of the website the user is authenticating to and thus the client could be able to detect fraudulently modified operation.
- SMS is a user-friendly, simple and popular communication channel. Using text messaging to transmit OTP information means users receive information in a format which they are comfortable with, as it is convenient and can be delivered to customers anytime, from anywhere.
- It is now possible to check a customer’s location and mobile subscriber identity by querying value-added network information directly from the GSM network, which enhances anti-fraud procedures.

3.1.5.3 Disadvantages

- Not guaranteed availability for all customers: the cell phone is a pre-requisite to all the users which cannot be guaranteed.
- The solution depends on the GSM network, thus is dependent of the availability of the coverage and the network load and delays.
• Additional costs of the outgoing SMS messages depending on the deal with the telecommunication operators.
• Customers management is expensive, since this approach relies on the authentication management server maintaining the current details of the customer's phone number and the customer being able to receive messages at the particular moment of logon.
3.1.6 Email OTPs

3.1.6.1 Description

The OTP used for the authentication can also be sent via a common used communication channel: email. As for the SMS method, the OTP is generated by the authentication server and then sent to the user email box.

3.1.6.2 Advantages

• Unlike the previous physical mediums that provide authentication OTP, the email OTP is not vulnerable to physical damages.
• Low cost solution, benefiting from an ecosystem that is already deployed.

3.1.6.3 Disadvantages

• Email is not considered as a secure channel to transmit the authenticator on.
• The solution includes the cost of an SMTP server to provide the email service.
• The method provides only authentication.
• Additional user details management should be ensured to keep the users email and contacts updated.
3.1.7 Conclusion

Combined with other simple authentication methods to constitute a two-factor authentication, using OTP certainly enhance the security level to an affordable and relatively acceptable level.

Although, one of the main disadvantage of using OTP is that it only provides authentication and none of the other security services: non repudiation, confidentiality ...
The method requires also the possession of OTP generation software/hardware and access to a secondary channel for OTP transmission.
3.2 Dynamic token

3.2.1 Presentation

The purpose of this section is to present authentication mechanisms which process special types of OTPs to authenticate users, challenge response and time-based OTPs, generated by special devices. These devices possessed by the user, are then considered as a “Something you have” authentication element. Semantically, the devices dedicated to the generation of OTPs are commonly called “token” and so they will be designated in this section.

The OTPs generated by the token are either:

- Synchronized password using time as a common denominator. The OTP changes at regular intervals, for example once a minute. The authentication server and the tokens are synchronized to Universal Time. To log into the system, users enter their password and the unique value generated by their token. They will be granted access to the system if their OTP and the one generated on the server side are the same.

  Problem which can arise with time-synchronized tokens is that the token and the central authentication server can get out of sync. If the token’s clock drifts significantly ahead of or behind the server’s clock, the authentication server may be vulnerable to a cryptographic attack.

- Challenge-response password, based on a challenge sent by a central server (such as a random string of numbers). The user would enter the challenge into the token. The token then calculates the response that serves as a one-time numeric password that is entered into the system. If the response from the user is the same as the response expected by the server, the user will be granted access to the system. In some implementations, the user must enter a PIN before the server will generate a challenge.

Most vendors employ proprietary algorithms for their token to generate the OTPs. However, the Initiative for Open Authentication (OATH) is an industry consortium promoting standardization and interoperability and discouraging security through obscurity.

The following sections will discuss two types of tokens: hardware and software.

3.2.2 Illustration of the protocol

During deployment phase, the users of the systems receive their token and are registered in the system with the associated token ID. The figure below illustrates the authentication protocol using challenge response OTP.
Illustration 8: Authentication protocol using challenge-response OTP
3.2.3 Hardware token

3.2.3.1 Description

Hardware tokens are physical devices offering the functionality of OTP generation. The token model defines the type of OTP that can be generated. The following picture illustrate different token models. The token hardware is designed to be tamper-resistant to deter reverse engineering of the token.

Many vendors offer a wide range of OTP-generating tokens of different shapes and sizes, and many offer custom branding options to print for example the company logo on the token. The simplest tokens are suitable for user authentication only with a simple button to start a generating routine with some display capability to show the generated key number. More advanced tokens incorporate keypad to allow entry of PIN code or challenge, making them suitable for transaction authentication. [11]

3.2.3.2 Advantages

- Users don't need to remember complex passwords.
- The solution can be used both for login and transaction authentication.
- The solution is computing platform and application independent.
- The solution offers portability advantage so the users can perform authentication from remote location, the small token size allowing it to be carried everywhere.

3.2.3.3 Disadvantages

- Involves additional costs, such as the cost of the tokens and replacement fees (batteries ...).
- Users need to always carry the token with them, which is an additional unusual device besides mobile phones, laptops, ...
- Users need multiple tokens for multiple Web sites and devices.
- The solution does not protect fully from man-in-the-middle attacks (i.e., attacks where an intruder intercepts a user's session and steals the user's credentials by acting as a proxy between the user and the authentication device without the user's knowledge)
3.2.4 Software Token

3.2.4.1 Description

Also called virtual tokens, these are software implementations of the hardware token functionality designed for installation on computers (hard drive), USB devices, PDA’s and mobile devices such as BlackBerry and iPhone.

Virtual tokens work by sharing the token generation process between the internet website and the user’s computer.

Depending upon hosting device, virtual tokens can offer:

• Equivalent manipulation as the hardware token, this means require the user to launch the software token application, generate an OTP and then cut-and-paste this into the password field.
• Integration with web application, browsers, VPN clients for near transparent authentication. In this case, users only need to provide their PIN when prompted and if the PIN is correct, the software token will automatically provide the necessary OTP in direct communication with the authentication server, without user intervention.

Software tokens can be issued and revoked as often as required, without additional cost or need to remove or recover the token from the client machine.

Illustration 11: Example of software token

3.2.4.2 Advantages

• Virtual token can be held on multiple media, such as a hard disk, floppy disk, and flash drive.
• The solution does not require the distribution and use of additional hardware, so the solution can have more user acceptance.
• If not explicitly revoked, software token does not expire for example because of batteries.
• Lower cost compared to hardware token.
• Virtual Token cannot be lost or stolen, so users experience no account downtime.
• More resistance to Man-in-the-middle Attack: with hardware tokens, fraudsters do not need to steal the actual token devices to compromise a user’s account. They only need to convince the user to disclose their token number to the fraudster’s fictitious website and then pass these values on to the genuine websites to access the accounts. By contrast, the keys and token numbers produced by the virtual token process have been “localized” and bind to the user’s specific device, eliminating their use by fraudsters who would be supplying these stolen values from their own (different) devices.

### 3.2.4.3 Disadvantages

• Software token requires some amount of user training.
• Virtual token need to be deployed in controlled environment.
• The solution requires reinstallation and configuration in case there is an operating system corruption or problem.
• It needs a protected and personal environment (e.g., should not be used from a public workstation).
• The virtual token is platform and application dependent (OS installed on, the web application it can be integrated with).
3.3 Certificate authentication

3.3.1 Presentation

Certificate authentication refers to the process of identifying the end-users in a transaction using digital certificates. In the web-based interactions, it is commonly used for server authentication, user authentication and mutual server-client authentication. This section will emphasize on the use of certificate for user authentication.

The certificate authentication process identifies users by virtue of their issued certificates, and is utilized whenever a secure transaction is initiated between a principle and a recipient, such as a client request to access a secure site.

Upon initial request and in parallel of an eventual server authentication, the user will present its digital certificate to the server which includes at a minimum the Distinguish Name (DN) and the associated public key.

The certificate is signed digitally by a trusted third party known as Certificate Authority (CA). The CA vouches for the authenticity of the certificate holder.

The most important element of the authentication process is the signature of the issuing CA. This signature is what the server will validate (using OCSP for example) to determine the recognized certificates, trusted certificate path and chain and the library of certificate authorities.

When the authentication server verifies digital certificates, it is checking to see if the certificate has been signed by a trusted CA. Before a certificate will be issued and signed by a CA, the user must be registered and his credentialing information must be verified. Thus, the certificate becomes a unique and unchangeable document that is suitable only for its holder.

If accepted, then the server and the user perform some cryptographic operations that prove the user possesses the private key corresponding to the sent public key. After success, the certificate authentication is done.

If the issuing CA is not recognized, then the certificate is not authenticated and instead, the user is denied access to the resource.

The most common used format of the digital certificate is PKCS#12 that stores the user private key with accompanying public key certificate, protected with a password-based symmetric key. During authentication process, the user presents his extracted public key and uses his private key for the cryptographic process.

For the management and deployment of the certificates, a public key infrastructure PKI should be deployed.

3.3.2 Illustration of the protocol

The following figure illustrates the authentication protocol using X.509 digital certificate.
Illustration 12: Certificate authentication protocol

1. User provides X.509 certificate protected by a PIN.
2. Client (browser) prompts for user X.509 certificate.
3. Authentication server sends user X.509 certificate.
5. Authentication server checks correct responses to the cryptographic operations.
7. Certificate validation server sends certificate validation OK.
8. Authentication OK for both User and Certificate validation server.

Illustration 12: Certificate authentication protocol
3.3.3 Smart card certificate

3.3.3.1 Description

To be securely kept, digital certificates (private key and public key) can be stored on hardware device, with cryptographic capabilities, called smart card. Physically, it can be in a shape of credit card or built on USB token.

For the authentication process, the smart card is inserted into a card reader to be connected to the workstation. The user enters his PIN to authenticate himself to the card. The user's public-key certificate is retrieved from the card through a secure process and verified to be valid and from a trusted issuer. During this process, a challenge-response based on the public key contained in the certificate is presented to the card. This challenge verifies that the card is in possession of and can successfully use the corresponding private key using its cryptographic capabilities.

Illustration 13: Example smart card and smart card reader

The smart card is proven to be tamper resistance preventing non-authorized user from accessing the smart card data.

3.3.3.2 Advantages

- Difficult for non-authorized users to extract the private key when stored on a smart card.
- The solution provide non-repudiation service (i.e., the act of denying an action after the fact). For instance, once someone uses a digital certificate and private key, the user cannot deny his action, because the private key resides with the user only.
- A third-party CA authenticates and distributes the digital certificates, thus helping to increase user trust.
- This method offers transaction authentication and digital signature service.
3.3.3.3 Disadvantages

- Requires user training for card reader use.
- Needs multiple certificates for different sites or devices.
- Involves additional cost, such as the certifying authority's subscription cost for issuing the digital certificates and card reader deployment cost.
- Requires users to carry an additional smart card reading device or USB so that they can log in.
3.3.4 Software certificate

3.3.4.1 Description

Rather than storing PKI private keys and digital certificates on a physical device, they can also be stored on the user workstation directly, protected by software. The most common example is storing the certificate in a PKCS#12 format in the user web browser. The certificate is imported into the browser using a PIN. When prompted, the browser present the user certificate to be authenticated by the servers.

Illustration 14: Example of software certificate in a browser

3.3.4.2 Advantages

- The solution provides transaction and user authentication.
- It allows highly secure digital signature.
- No need of additional device (card or USB token) to be carried to perform authentication.
- Cost saving in distribution and maintenance.

3.3.4.3 Disadvantages

- Low user mobility since the authentication cannot be perform unless the user is connected from his personal workstation where the certificate is installed.
- Vulnerable to PC Trojans and generally to the workstation security vulnerabilities.
- High integration and support costs.
3.4 Biometric authentication

3.4.1 The method

Biometric authentication is an authentication method based on biological or behavioral features that uniquely identifies a person. When a user is given an account, the system administration takes a set of biometric measurements that identify user to an acceptable degree of error. Whenever the user accesses the system, if he provides a good enough sample of the measured biometrics, which closely match the previous recorded biometric templates, the identity of the person authenticated is guaranteed.

Common used biological features are:
- Fingerprints based on the position of curves and junctions of a fingerprint.
- Face recognition.
- Eyes characteristics using the pigment structure of the iris and the retina.

Common used behavioral features are:
- Voiceprint involves recognition of a speaker's voice characteristics or verbal information verification.
- Typing rhythm based on keystroke intervals, keystroke pressure and keystroke duration.

3.4.2 Advantages

- No risk to lose or forget the authenticator, it is permanently present and accessible for the user.
- Cost saving regarding authenticator management and administration.
• Very user-friendly, it is more evident for a user to pass his finger on a scan than to re-
member, type and re-type his password.
• Biometric credentials are extremely difficult to duplicate.
• Biometrics takes care of non-repudiation because they are specific to the user only.
• Different options are available, such as fingerprint, iris or retina authentication.

3.4.3 Disadvantages

• Cannot be revoked or re-issued: if the biometrics has been forged or copied, all the au-
thentication system is compromised.
• This authenticator does not respect two main security principals: the “principle of
least common mechanism” since the metric is used as an authenticator in many sys-
tems and the “principle of separation of privilege” because when the biometrics are
stolen it would permit to access other system using these biometrics.
• Biometric identification systems undermine privacy by making identity theft more
likely.
• The uniqueness of the biometric features is till now an assumption not formally
proved.
• Biometrics involve additional hardware costs such as scanners and costs for support
and maintenance, beside high deployment costs.
• The method may not be suitable for mass-consumer deployment.
4 Evaluation of strong authentication solutions

The purpose of this section is to compare the studied strong authentication solutions and present an evaluation summary which can help customers choose the most convenient and appropriate authentication method for their needs and constraints. The evaluation is based on criteria detailed in the first part. The second part presents the evaluation results.

4.1 Evaluation criteria

The criteria used for the evaluation of the studied strong authentication solutions have been chosen to match customers requirements regarding cost, aimed security level, ergonomic and constraints related to legal legislation in the customers country.

4.1.1 Acceptability

This criterion evaluates how the system is easy to use for customer with different background knowledge.

People will simply not use system they find difficult, even if it provides great security level. Positive customer response is essential to the success of any deployment in a way that solution complexity should not be an obstacle against applying security.

Portability of the authenticator can be included in the acceptability factor since the user would be very pleased to be able to authenticate from everywhere and to be independent to a maximum from any workstation or device. An example is the fingerprint authenticator that the user obviously does not need to think about carrying it with him less than mobile phone authenticator which is, in his turn, more portable than software certificate installed onto workstation.

Considering all this factors, simplicity of use should not affect the reliability of the system: the security architect should find the right balance between acceptability and effectiveness.

4.1.2 Effectiveness

This criterion evaluate how effective is the solution against a wide range of simple and advanced attack scenarios, which means the security level it guarantees.

Reliability is also included in the effectiveness evaluation of the authentication solution. As an example, consider two authentication systems for workstation activation: one using webcam to scan eye retina and another using fingerprint reader with a PIN. Even if the retina scan system is easier for user utilization, it presents higher false rejection rate, when fingerprint is less intuitive but are more reliable.

The effectiveness also measures the ability of the system to integrate new eventual evolutions (system growth), to be technically compatible with the wide range of existing products and software developed and tested and to be capable of offering different services (for example authentication and electronic signature).
4.1.3 Cost

This is an important criterion to consider the full costs, and to devise a strategy to extract the greatest return on the necessary investment. It includes:

- Initial purchase costs, related to deployment infrastructure that must be scalable, highly reliable, and securely protected, to the packaging and the delivery of the solution, to the price of authentication products (devices and software) added to licenses fees.
- Deployment costs, including both internal and outsourced resources to plan, install and configure the solution, as well as costs to set up and train end-users.
- Support costs to maintain the solution deployment as well as end-user support and help desk
- Lifetime and replacement costs comprise maintenance of the system and reissue of new token in case of failure, annual support subscription and license renewal.

Naturally, the lower is the cost of the authentication solution, the higher is the solution score for the criteria.

4.1.4 Client constraints

This criterion includes all external subjective constraints the client is subject to, that can be related to:

- Legal legislation of the country where the solution is deployed specially the public it is designed for, the cryptographic algorithm it uses. This is particularly true for solutions using biometrics.
- Technical constraints regarding pre-requisites or existing infrastructures.

This criterion is not included in the evaluation summary because it is closely related to local characteristics of the project, but it highly motivates the client choice.
4.2 Evaluation summary

The following table draws a summary evaluation of the strong authentication solutions presented:

<table>
<thead>
<tr>
<th></th>
<th>Acceptability</th>
<th>Effectiveness</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scratch List</td>
<td>✭✭✭</td>
<td>✭</td>
<td>✭✭✭✭✭✭</td>
</tr>
<tr>
<td>Matrix card</td>
<td>✭✭✭</td>
<td>✭✭</td>
<td>✭✭✭✭✭✭</td>
</tr>
<tr>
<td>SMS OTPs</td>
<td>✭✭✭✭</td>
<td>✭✭✭</td>
<td>✭✭✭✭✭✭</td>
</tr>
<tr>
<td>Email OTPs</td>
<td>✭✭✭✭</td>
<td>✭✭✭</td>
<td>✭✭✭✭✭✭</td>
</tr>
<tr>
<td>Hardware token</td>
<td>✭✭✭</td>
<td>✭✭✭✭✭</td>
<td>✭✭✭</td>
</tr>
<tr>
<td>Software token</td>
<td>✭✭</td>
<td>✭✭✭✭</td>
<td>✭✭✭✭✭</td>
</tr>
<tr>
<td>Smart card certificate</td>
<td>✭✭✭</td>
<td>✭✭✭✭✭</td>
<td>✭✭</td>
</tr>
<tr>
<td>Software certificate</td>
<td>✭✭</td>
<td>✭✭✭✭</td>
<td>✭✭✭</td>
</tr>
<tr>
<td>Biometrics</td>
<td>✭✭✭✭✭✭</td>
<td>✭✭✭✭</td>
<td>✭✭</td>
</tr>
</tbody>
</table>

Illustration 16: Strong authentication solutions evaluation summary

As the Table shows, none of the solutions responds ideally to all the criteria. Thus the system architects are in charge of defining all the system requirements and pre-requisites, list all the possible solutions that match their objectives and then choose the one which offers the right balance between all the criteria.
5 Single Sign On

5.1 Presentation

Single Sign On (SSO), is a property of access control of multiple related but independent systems, according to a defined security policy, using login process only once and granting access to all the other systems with the previously performed authentication.

The fundamental problem a single sign on solution solves is the problem of forwarding the authentication credential from one service to another in a secure manner. This should hold true even though the client is regarded entirely untrustworthy.

Since the internet is stateless, this means that the single sign on solution must check every request to see if there is an authentication policy pertaining to the resource or application the user is trying to access.

5.2 Advantages

• Reduce administrative costs. SSO can reduce help desk costs associated with password resets since the user uses only one password to access the system and then SSO functionality relays his identity to the other systems.
• Increase security. SSO eliminates the need to remember multiple passwords and thus removes the risk of users engaging in unsafe password practices.
• Bolster employee productivity. SSO provides access to multiple resources with a single password to help bolster productivity.

5.3 Disadvantages

• Integration problem. The SSO cannot be proven to be integrated with all the system components and could require complex design to do so.
• System evolution. Adding new components to the system can introduce incompatibility with the SSO features.

5.4 Single sign on architectures

This part attempts to clarify the different ways a SSO solution can be built emphasizing on three main architectures: plug-in SSO agent, helper application agent and reverse proxy agent architecture. Before, it details the client-agent-server architecture with the responsibilities of each component in the SSO process.

5.4.1 Client-agent-server architecture

The client is a distrusted component, capable of completing the authentication protocols and storing the access control ticket issued to it by the server if the authentication was successful. Commonly it is an encrypted cookie. A typical client would be using a web browser acting on behalf of the user as a user-agent.
The agent is situated near the protected service acting as a gatekeeper, consulting the server for authentication and authorization decisions as well as supplying it with audit-data. The agent is a small piece of code that effectively can say ‘yes’ or ‘no’ to resource requests based on the authorization information provided by the server, and forward the acceptable ones to the service and the responses back to the client.

The heart of the system is the server, which provides the back-end processing capabilities with support for different authentication methods, user databases, policy evaluation capabilities and audit logging and processing functions. It is the server who validates the cookies.

### 5.4.2 Plug-in SSO agent

In this model, the protected resource, i.e. the software service, is modern enough to support natively authentication method plugs-in. This enables the addition of the SSO agent service into the software infrastructure.

![Illustration 17: Plugin SSO agent architecture](image)

### 5.4.3 Helper application agent

In this model, the agent is situated in the same physical machine, but it is not as integrated into the application as a plug-in. It has to intercept the data connections before they reach the application. After interception, the agent checks if the accessed resource is protected. In this case, either:

- The agent is able to retrieve user valid credential if previous authentication has been made and grant access to the user.
- Or forward the request to the authentication service to perform authentication.
The main problematic of this architecture is to guaranty the capability to pass authentication information from the agent to the application.

### 5.4.4 Reverse proxy agent

In this model, the functionality of the agent is situated in an external machine that routes traffic from the public network to the private network, which is shared between the proxy and the service. This solution is required if the protected resource is not able to use either plug-ins or there does not exist an agent that will run on the OS platform of the resource. In this case, the agent is running in its own computing environment.
6 Commercial products survey for SSO

In this chapter, some authentication management products that implement SSO functionality currently available on the market are enumerated. This part of the study is based entirely on marketing data from the vendors’ web sites, so no guarantees as to the accuracy of the claims can be given by the author.

6.1 EMC: RSA Access Manager

RSA Access Manager is a web access management software that serves as the foundation of any identity management system. It enables organizations to provide secure access to web applications within intranets, extranets, portals and exchange infrastructures-with transparent, SSO access for a positive user experience. Enabling such access can improve communication and facilitate business with customers, strengthen partner relationships and bolster employee productivity. Equally important, organizations can reduce administrative burdens and lower costs. It is supported on Windows 2003/2008, HP-UX, Solaris and AIX platforms.

6.2 CA: SiteMinder

SiteMinder is a centralized Web access management system that enables user authentication and single sign on, policy-based authorization, identity federation, and auditing of access to Web applications and portals. It offers centralized Web single sign on, authentication management, policy-based authorization, identity federation, and audit services as part of a proven enterprise security system that is highly manageable, reliable, and scalable. It is supported on UNIX, Linux, and Windows, as well as mainframes, application servers, Web servers, ERP, CRM, RDBMS, directory, and other systems.

6.3 Evidian: Web Access Manager

Evidian Web Access Manager is a secure web single sign on solution that connects your remote users to their web applications through a secure portal. It enforces your security policy over both desktop and web access. Evidian Web Access Manager requires a Solaris or Red Hat Enterprise Linux server. Evidian Web Access Manager provides single sign on to HTML and Java applications. In addition, the companion Evidian Enterprise SSO module provides single sign on to Windows, Citrix and terminal-based applications.

6.4 Entrust: GetAccess

GetAccess is a high performance, scalable Web access control solution. It centrally manages access to multiple applications through a single portal, providing users with single sign on to the applications and content they are authorized to see. With the broadest support for user authentication methods, flexible roles and rules-based access control, self-service features, and proven performance to millions of users, Entrust GetAccess enables organizations
to reduce administration costs while driving more services through their Web portal. It provides single sign on support with both basic and strong authentication for web clients. It is available for Windows 2000/2003, Solaris, Linux, HP-UX and AIX on the servers and any web browser for the client. [5]
7 Implementation

7.1 The selected authentication method

Based on the security requirements identified for the system and the authentication methods evaluated in the previous chapter, security architects on the project, in collaboration with the client, established the following result studies:

<table>
<thead>
<tr>
<th>Solution</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scratch List</td>
<td>The existing enterprise solutions use uncommon components whose maintenance and evolution are not guaranteed nor supported</td>
</tr>
<tr>
<td>Matrix card</td>
<td>The existing enterprise solutions are not compliant with the client infrastructure.</td>
</tr>
<tr>
<td>SMS OTPs</td>
<td>The client cannot require from his users to possess a mobile phone as a pre-requisite to access the system</td>
</tr>
<tr>
<td>Email OTPs</td>
<td>The security level of the solution is not acceptable</td>
</tr>
<tr>
<td>Hardware token</td>
<td>The proposed enterprise solution are not compliant with the client infrastructure</td>
</tr>
<tr>
<td>Software token</td>
<td></td>
</tr>
<tr>
<td>Smart card certificate</td>
<td></td>
</tr>
<tr>
<td>Software certificate</td>
<td></td>
</tr>
<tr>
<td>Biometrics</td>
<td>France has legal constraints regarding the use of biometric system: the usage should be justified and authorized by the National Commission for Informatics and Liberty (CNIL), which is not the case for the project</td>
</tr>
</tbody>
</table>

Illustration 20: Authentication solution justifications

As shown in the above table, the strong authentication system will use digital certificate.

The reason why two solutions were designated as winning is that the certificate deployment is not of the responsibility of Accenture team. It is the client who is in charge of implementing and providing the PKI that will issue the user certificates, and thus responsible of the used medium to distribute the certificate (smart card or software installation on the users' workstation), which is not yet defined in this phase of the project. Because of these circumstances, the deployment of the selected product is based on the assumption that certificate is already available in a software format on the user workstation.
7.2 The Selected product: RSA Access Manager

The security product selected to implement the certificate authentication and the SSO feature is RSA Access Manager along with developed custom modules for advanced functionalities and integration. The product has been chosen in the proposal phase, regarding the product cost, the product certification and established partnerships.

7.2.1 Description

RSA Access Manager enables organizations to manage authentication and authorization policies for a large numbers of users from a central location. This ensures that only authorized users can access sensitive data within web-based systems – ensuring the right people have the right access at the right time. Access privileges can be determined by selected attributes, such as job function and responsibilities, and can be readily turned off if a person is terminated or takes on a new job role that does not require access to certain systems. Access Manager is Common Criteria EAL3+ certified. [2]

7.2.2 Supported platforms

RSA Access Manager supports different operating systems, data stores, application servers and web servers. It is detailed as follows [7]:

- Operating systems:
  - Microsoft Windows Server 2003 R2 SP2 x86 64 bit, 2003 SP2 x86 32 bit and 64 bit, 2008 x86 64 bit, 2008 SP2 x86 32 bit and 64 bit
  - Solaris 9 SPARC, 10 SPARC 64 bit and 10 Zones
  - Red Hat Enterprise Linux (RHEL) 4.x x86 (32 bit and 64 bit), 5 ES x86 64 bit
  - SuSE Linux Enterprise Server 9 x86 32 bit, 10 x86 64 bit
  - IBM AIX 5.3, 6.1
  - VMWare ESX, Workstation

- Data Stores:
  - Microsoft Active Directory
  - Sun Java System Directory Server 5.2, 6.3, 7.0
  - Oracle 9i, 10g RDBMS, 10g R2 RDBMS, 10g R2 RAC, 11g
  - Novell eDirectory 8.8.1
  - Adaptive Enterprise Server 15.5

- Application servers:
  - Oracle WebLogic Server 9.2, 10.0
  - IBM WebSphere Application Server 6.0.2, 6.1, 7.0

- Web servers:
  - Microsoft Internet Information Server (IIS) 6, 7.0
  - Sun Java Systems Web Server 6.0, 6.1, 7.0
  - Apache HTTP Server 2.0x, 2.2x
7.2.3 RSA Access Manager architecture

RSA Access Manager consists of a few major components [2]. These are:

- RSA web agent
- Authorization server
- Entitlement server
- Dispatcher
- Data stores for storing users and RSA Access Manager configuration
- Administrative GUI to manage users, resources and policies

The components interactions are illustrated in figure Illustration 21: RSA Access Manager architecture.

When a user tries to access the protected resource on a web server, he actually communicates with RSA web agent. The web agent takes care of checking if the user possesses the needed authenticator (cookie) to access the protected resource. If not, the agent redirects the user to perform the authentication process corresponding to the authentication method chosen. In project implementation, it was a certificate authentication. If the user has an authenticator (cookie), the agent validates it against the authorization server designated by the dispatcher. The dispatcher plays actually a load balancer role when multiple authorization servers are used. He then forwards the agent request to the less charged server. The authorization server accesses the database for verifications.

The administrative GUI interacts with the entitlement server to set up groups configurations, authorization rules and policies definitions, stored in RSA Access Manager data store.
Illustration 21: RSA Access Manager architecture
7.3 Implementation in the project context

This section will present RSA Access Manager, as it was used in the project context I worked in, integrated into the global information system dedicated to payroll management and how it implements the security services desired. The global architecture of the system is first presented and all the functionalities it implements. Next, the security architecture is described including the security services it is designed for. The authentication architecture designed to achieve strong authentication is zoomed out. RSA Access Manager integration is detailed and all the custom developments needed.

7.3.1 Global architecture

The global information system is dedicated to HR management and offers advanced functionalities for payroll management. The system is build around a central unit, called “pay calculator” that implements all the payroll functionalities, the principal purpose of the system. The end-users and all the external components access the system through a unique access point. This component interacts with internal and external elements which perform all the other services as illustrated in the following figure.

![Global architecture of the system](Illustration 22: Global architecture of the system)

The system is composed of different internal components, each accomplishing a service:
- The transactional service ensures the coherence of all the transactions and calculations.
- The archiving service carries out the storage of the business documents of the system.
- The exchange service provides an architecture to secure the communication channel for exchanging documents between the IS and the external components.
- The security service, based on different software and hardware components, offers different security functions essential to protect the system.

7.3.2 Security Architecture

The purpose of the security architecture is to implement the security service in the system, to fulfill the 6 standard security requirements:
- Confidentiality
- Authentication
- Authorization
- Integrity
- Non-repudiation
- Availability

The following figure illustrate all the components of the security architecture and how each of it meets the standard security services:

Illustration 23: The security architecture
• The digital signature component meets the integrity and non repudiation requirement in a way that every document exchange with Payroll management IS is signed to guarantee the issuer identity and the integrity of all the data contained in the file.
• The authorization component defines the roles and habilitations of every user of the IS.
• The authentication component ensures the confidentiality and authentication services by allowing access to the system only through one unique access point through encrypted secure, and identified network flows. RSA Access Manager product implements the authentication service helped by certificate validation software.

7.3.3 The authentication architecture

The packaged software used to implement the authentication architecture are as follows:
• RSA Access Manager server and agent which functionalities and architecture are detailed previously.
• VeriCert which is a solution for electronic certificate validation.
The goals of the authentication is to ensure the following services:
• Authenticate all accesses to the system. The solution adopted is to use strong authentication based on electronic certificates. The user certificates are validated using the validation service.
• For user acceptance, regardless of the different components of the system, the architecture should be transparent. In order to achieve this goal, the user should not be prompted to authentication every time he accesses a different component. The solution is to integrate SSO mechanism based on the cookie issued in the authentication phase.

The authentication chain

The architecture that allows to implement the authentication service is detailed in the next figure:
The steps of a basic authentication scenario are as follows:

1. The unique access point hosting the reverse proxy for the system access intercepts any incoming request. RSA Access Manager Agent hosted also on the unique access point checks if the request accesses a protected resource. If so, the agent redirects the request to the authentication module.

2. The request is passed to the authentication module in charge of extracting the user certificate from the request headers. This module interacts with the Certificate Validation Service and the RSA Access Manager Server as detailed in the steps 3 and 4.

3. The Certificate Validation Service validates the user certificate. This service allows to determine the CA accepted and the certificate chains allowed.

4. The authentication module checks the user identity against the RSA Access Manager Server using the subject DN of the user certificate. If the check succeeded, the server issued an encrypted SSO cookie for user. This cookie will allow the user authenticate to the other system components.

5. The request is then forwarded to the component the user is accessing. The SSO module of the component extracts the SSO cookie of the user from the headers, decrypts it using RSA Access Manager Server. If the decryption succeeded and the user identity extracted, the module issues an application cookie for the component and the user can access the target.

**7.3.4 Installation**

The technical details of the installation of RSA Access Manager on the project are:
• RSA server installed on RHEL 5.4 x86_64 bit
• RSA agent installed on front Apache HTTP server 2.2.15
• RSA administrative GUI installed on Tomcat Apache 5.5.27
• RSA servers are connected to a LDAP store Sun Java System Directory Server 7.0

7.3.5 Integration and custom developments

During the design phase, the first prototypes revealed that the selected products do not automatically implement all the requirements. Thus custom modules have been added to guarantee a perfect integration.

Two main custom developments, illustrated in Illustration 24: The authentication chain, were required and added:

• A custom authentication module between the RSA web agent and RSA server. This module was introduced to perform an advanced validation of the user digital certificate, against a dedicated component, the certificate validation service. This module was developed as J2EE web application installed on Apache Tomcat 5.5.27 which interacts with RSA servers through editor APIs.
• As the global system was composed of different others components, the identity of the user had to be propagated to them using SSO model, i.e. to be transmitted transparently to the component so the user would not be prompted for new authentication. This is done by adding custom developed SSO modules integrated into the security framework of each component.

7.3.5.1 Design

One of the main design tasks I performed is the specification of the login modules to be integrated with each components of the payroll management system. This task was very challenging as it requires to priorly understand the security model of each application. For this, I had to read the security documentation of the components, open support tickets with the application editors and profit from any previous experience of other Accenture teams on the same product.
The efforts soon return profits as I successfully succeeded to design and specify all SSO modules of all the components.

In a high level understanding, algorithm the SSO modules implements is presented in Illustration 25: High level description of the SSO modules algorithm. The role of these modules is to extract the authenticator (the cookie) from the requests, decrypt and validate it against the authorization server and then inject the user identity. These modules were developed in native Java and used editors APIs to call cryptographic methods to manipulate the encrypted cookie.

7.3.5.2 Implementation

Once the design and specifications of the single sign on modules were finished and validated with the team, we had to prototype and test them. We proceeded as follows:

- Installation and configuration of the components: using the installation documents and the expert advices, I installed the RSA Access Manager solution and configured it to perfectly integrate the custom modules.
- Prototype the module: working with a developer, I participated to the implementation of the SSO module of each application. This task required good team relationship skills. I have to work hand in hand with the developer and explain all the design and specifications details as he joined the team lately.

7.3.5.3 Technical details

The SSO modules were developed in JAVA using Eclipse development environment and Maven compiling architecture. The module part of the security architecture, the implementation should be compliant to the security standards regarding the code (variables, used API, initialization …), the logging files and the parameter files. We also performed peer review to guaranty the code quality.
7.3.5.4 Test and validation

As our architecture interacts with all the components of the system, we were very dependent on the other teams. To avoid being blocked and to be able to test the implementation, I had to set up a test environment to simulate the entire authentication chain. For this purpose, I configured SSL termination on reverse proxy hosted on the Unique Access Point, using SSL configuration on Apache.

As the PKI is centralized, getting the certificate signed by the official Certificate Authority is very complex and requires many administrative procedures. To avoid delays and be able to test our implementations, I used an open source SSL tool to generate self-signed certificate simulating the CA. The user test certificates were signed by the previous CA certificate.

To test the implemented modules, I designed a testing plan to test all the possible authentication scenarios, including the entire authentication chain, the error cases and the exception causes. I have then to report the bugs to the developer, who fixed them. I had to verify that all the bugs are fixed and to perform also non-regression tests.

Testing the module was challenging, as the tests were performed on the development platform shared by all the teams. So I had the responsibility to guarantee that the integration of the modules would not affect the platform and inform all the team of the operations planned on the platform.

I also had the responsibility to write the installation guides for all the implemented modules and the user guide to describe how to use the strong authentication:

• Request the user certificate and provide all the information needed to issue a certificate.
• Install the certificate on the browser using the PIN (in the test phase).
• Describe excepted results.
• Report abnormal behaviors.

I was very proud when we invited the client to see the realized successful prototype during a demonstration. The client was very satisfied with the results.

7.3.6 Demonstration

From the user point of view, the access to protected resource follows the scenario:

• User opens his browser and accesses the URL of the protected resource.
• The user is prompted to select the digital certificate to authenticate, as shown in figure Illustration 26: Certificate use in the browser.
• Transparently to the client, the authentication process described in section 6.3.3 is performed.
• If the authentication is successful, the user access the resource and he is issued a SSO cookie as authenticator to access other resources.
• If the authentication fails, the access denied page, illustrated in Illustration 27: Access denied page is displayed.
Illustration 26: Certificate use in the browser

Illustration 27: Access denied page
The administrative GUI of RSA Access Manager is illustrated in Illustration 28: RSA administrative GUI. It allows:

- Define resource to protect.
- Define access policies and rules.
- Create and manage users and groups.

Illustration 28: RSA administrative GUI
8 Project Context

8.1 About Accenture

8.1.1 Overview

Accenture is a global management consulting, technology services and outsourcing company. With approximately 211 000 people in 200 locations, Accenture serves clients in more than 120 countries, among the world's leading companies and governments including 94 of the Fortune Global 100. Its most prestigious clients in France are: Total, EDF, Gaz de France, Cegetel, Societe Generale, Airbus, Alcatel, Carrefour, Air Liquide and the French Ministry of Finance. For the last fiscal year 2010, the firm generated net revenues of US$21.6 billion. [6]

Committed to help them achieve high performance and improve efficiently their businesses, the partnership between Accenture and its clients is carried out at different levels:

- Analysis of the client's situation, its position and its needs in terms of technological evolution, in order to provide appropriate advice to meet its objectives.
- Closed collaboration with the client in order to provide innovative proposals, comprehensive and tailored to every need.
- Implementation of end-to-end solution, from design to development, and from validation to change management.
- Outsourcing of certain functions of the client company, allowing him to refocus on his heart business and to reduce costs.

8.1.2 Organization

Accenture has a matrix structure. The first axis represents the “operating groups” which are large entities oriented to client business. This division into "Operating groups" enables Accenture develop an appropriate approach towards the customers, depending on the context and specificities of each of these industries. [6]

The "operating groups", in order of size in France, are as follows:

- Products, serving companies in industries from automotive, capital goods, distribution, land transport and consumer goods like Carrefour, SNCF.
- Financial Services, operates in the banking domain, capital markets and insurance. Most known clients are: Societe Generale, Calyon, AXA, Groupama, Caisse d'Epargne.
- Communication and high technology, designed for the telecommunications, media and aerospace sector. Cegetel, Orange, Airbus, France Telecom are some of the prestigious clients of this operating group.
- Resources, operates on the sectors of chemicals, energy, utilities, metals and mining. Most known clients are: Total, EDF, Gaz de France, Rhodia.
Health and Public services, serves companies in areas of defense, education and in general governmental services. Key customers: The Ministry of Economy and Finance; ANPE, LaPoste.

The second axis deals with the growth platforms, which are the range of services that can be provided to assist customers on very specific aspects, such as managing customer relations. The growth platforms act across operating groups to meet transverse business needs and mobilizes highly skilled people experts on key products such as Siebel, Peoplesoft, SAP. The growth platforms are:

- Management Consulting
- Systems Integration and Technology
- Outsourcing Services
8.1.3 Accenture Security Offer

The diagram below specifies the positioning of the security entity in Accenture organization:

- Security risk management
- Enterprise security Application
- Identity & Access Management
- Infrastructure Security
- Business continuity management

During my project, I worked on the offer Identity & Access Management.
8.2 Project organization

8.2.1 The security team mission

For my end-of-studies project at Accenture, I joined the security team within an information system project dedicated to payroll management. The project intended to perform secure transactions and to manipulate sensitive data, the security team was one of the important entities on the project in charge of design, implementation, test and deployment of the security service.

The tasks and responsibilities the team carries out are:

- Insure a global security coherence on the project by assisting and collaborating with all other teams to fulfill all the client security requirements.
- Design and implement a perimeter security including network security (Firewalls, Antivirus) and infrastructure security (Hardening)
- Set up a security architecture for electronic approval including digital signature and operation logging
- Design and implement an Identity and Access Management solution to guarantee secure access to the system based on strong authentication, roles and capabilities

8.2.2 My mission

My mission was mainly related to the Identity and Access Management solution, working closely with the consultant in charge of this part. Two other consultants were responsible for the electronic approval architecture.

My objectives were:

1. Understand the context of the whole project and the Identity and Access Management solution. In this phase, I had to develop skills related to the products used, the environments that constituted the system. I also had to identify the teams I will be in close interaction with.
2. Understand the global architecture and services offered by the system.
3. Participate to the design of the strong authentication solution.
4. Participate to the design of the SSO solution which should be integrated to all other services.
5. Build a test platform for the solution prototype
6. Design the prototype solution
7. Test and evaluate the solution

8.2.3 The project management

There were about 150 consultants working on the project. Thus, an efficient project management tools were needed to keep all these people coordinated and to allow them honor the responsibilities towards the client and towards the reputation of their company.

To achieve this, many tools and procedures have been deployed. In fact, I needed some time to get used to those tools, but soon realized the efficiency, and time saving it offers.
8.2.3.1 Ticketing tool: ClearQuest

As we were divided into many teams, sending emails or calling to fix or to request a technical support seem to be a very prehistorical and not efficient solution. For this purpose, a ticketing software, named ClearQuest, is available to subscribe technical requests to the appropriate team.

The workflow of the ticketing is as follows:

- Log on the application, register the problem and provide a general description of it. It is also possible to attach documents (log files, scripts) to help solve the problem.
- The manager of the responsible team attributes the task to the appropriate person.
- The person solves the problem or contacts you for more details and works together with you to find the fix.
- When the problem is solved, we close the ticket.

This system allows keeping track of all the changes made, identifying the cause of any problem, and rolling back in case of incident.

8.2.3.2 Versioning tool: ClearCase

The project is constituted of different platforms: from development platform where we test develop to production platform. To ensure coherence between all the platforms, all the configuration files, tested developments on the development platform are delivered to the other platforms through an automatic process to avoid manual procedure. The used tool is called ClearCase.

A versioning tool is deployed for this purpose. It also offers to perform code compilation with a standardized build platform to keep track of all the code versions.

8.2.3.3 Documents sharing: Alfresco

All documents relevant to the project are kept on a shared repository and available to everybody on the project. Restrictions are applied on sensitive documents for example security documents were accessible only by people habilitated to do so.

This allows us also to keep all the version of all the documents and to be safe in case of a local disk crash.

8.2.3.4 Meetings

We used to organize a weekly team meeting on Monday. In this meeting, the manager presents an overview of the evolution of the project, the milestones validated and those planned for the next week.

Each member of the team presents the work he did during the past week, exposes problems he faced (internal or because of interaction with other teams). It is also used to give the manager feedback and to inform him if there is any blocking points or any questioning.

After every delivery, we organize a meeting with the client to expose the design we adopted, to ensure we meet the requirements and the expected objectives.
8.2.3.5 Planning

The global planning of the project is as follows:

Illustration 31: Planning

8.2.4 Packaged Development Methodology

Accenture has designed and tested methodologies to carry out different types of projects, to respond and optimize the effort and results.

For the security architecture, the team adopted the packaged development methodology. This methodology has been designed to support the implementation of packaged software. The design is based on using packaged products developed by specialized editors in the target technology and to integrate it, configure it to meet our specific requirements.

The challenge using this methodology is to success to:

- Understand all the functionalities offered by the products in a minimum amount of time.
- Perfectly integrate all the packaged software and avoid any conflicts with the other component.
- Develop custom modules to guarantee the integration.
- Test every component independently before integration.
- Test and configure the integration into the whole system.

8.2.5 Team integration

When I joined the team in the 16th of June, I started reading the welcoming kit. It included the general organization of the project, the different teams and interlocutors I will be in interaction with.

I have to read and understand the global design of system and the different components it is composed of.

I then zoomed on the security architecture.
A risk analysis phase has been completed by the previous team in charge on the security architecture on the project. The analysis leads to security chart classifying all the risks and vulnerabilities the system is exposed to. It also specified all the security requirements our design should fulfill. Reading this analysis allowed me getting familiar with the security context the system will be deployed in. I also gained many benefits from understanding the application of the theoretical risk analysis in industrial environments.

As described in the section 7.3.2, the security architecture includes the authentication and authorization solution RSA Access Manager. As it was a new product for me and the team member, so we had the opportunity to receive a training for the product. A product expert came from UK to Paris to provide us this training which lasts for 2 days. He explained all the functionalities the product offers, the best practices to optimize the performances and the most suitable configurations for our case. The expert also give us advices beyond our needs which we could exploit in others projects.

As a new team joiner, a member of the team took time to explained to me how all the platforms and tools of the project work.
9 Conclusion

9.1 Personal feedback

The project is a great opportunity to put in practice all the theoretical knowledge we acquired, to involve into a professional working team and to develop many professional and personal qualities.
Willing to integrate the working environment after my graduation, I chose to perform my project in industry. The project gives me the chance to discover the consulting job.
The very stimulating, challenging and dynamic work environment in a big project for a prestigious client made this project very interesting and very instructive.
Technically, the project allows me to discover industrial products I do not deal with in the academic studies. Thanks to the training, I was able to develop an expertise on the authentication products we used on the solution.
The interaction with many teams on the project and different software editors abroad allows me to develop a professional communication skills, writing email or formulating requested to the right people.

9.2 Perspectives

For the last part of the project, I m planning to continue the implementation phase working hand in hand with a developer. I will also participate to the integration phase with the other systems, which need preparation of test cases to guaranty a perfect integration of all the components of the information system.
Bibliography


[12] Offering SIM Strong Authentication to Internet Services, 3GSM WORLD CONGRESS, BARCELONA 13 - 16 FEBRUARY 2006