2

Selection of secure single sign-on solutions for heterogeneous computing environments

C.P.Louwrens and S.H. von Solms
Department of Computer Science, Rand Afrikaans University,
P.O. Box 524, Auckland Park, Johannesburg, 2006, South Africa.
Telephone: +27 011 489 2843
Fax: +27 011 489 2138
E-mail : buks@icon.co.za, basie@rkw.rau.ac.za

Abstract

Secure Single Sign-on (SSSO) is the concept of minimizing the number of different userids and passwords required to access various host systems in a distributed computing environment, while providing a consistently secure environment which also provides confidentiality and integrity services. In its purest form, Single Sign-on (SSO) allows a user to sign -on once to the enterprise computing environment and be granted access to participating host systems across the enterprise. In a wider context, extending the concept to SSSO, it impacts on the enforcement of security policies, security management and administration, security services, and overall productivity. Selecting and implementing SSSO solutions may present interesting challenges, and may lead to increased risk, if not done carefully and properly.

This paper discusses the concepts of SSSO, user requirements, and presents a reference framework for selection of Secure Single Sign-on solutions in heterogeneous computing environments, which can assist in SSSO requirements specification and product evaluation.

Keywords

Secure Single Sign-on, Authentication, Access Control, Integrity, Confidentiality, Security Management, Heterogeneous Environments.

1 INTRODUCTION

In today's heterogeneous computing environments, end users frequently need to access applications and network resources running on multiple platforms to perform their day-to day responsibilities. This typically requires that end users use different sign-on routines, userids and passwords creating a cumbersome management problem for themselves as well as systems administrators and security managers. The same end users often depend on the note-posting technique, trivial passwords or password sharing to contend with multiple sign-on procedures and passwords. (Computer Associates, 1996)

Whilst it is vital to ensure that data remains secure, traditional approaches can make systems unusable, requiring users to learn and navigate through different layers of passwords and log-on routines. Current research estimates that usability issues cost the average organization some 10% of the potential productivity gains enabled by IT systems. (ICL, 1996)

1.1 Impact of Single Sign-on

Gaining access to disparate systems, without single-sign-on impacts businesses in three ways :

- **Dissatisfied users.** Users experience security as a burden and foster an attitude of security being an impediment to performing day-to-day business activities.
- **Reduced efficiency**. Users can lose significant productive time by multiple sign-on's, changing and maintaining passwords and duplication of the administration effort.
- **Weakened security.** Faced with the need to remember a series of sign-on data, users are more likely to select passwords that are easily remembered, and thus easily guessed, share them or write them down. (Stanley, 1996)

1.2 Single Sign-on (SSO) versus Secure Single Sign-on (SSSO)

Single Sign-on (SSO) is a concept that provides the user with a single userid and password for access to all the resources on the enterprise network. The problem is, that in many cases, passwords and data are sent in the clear over the network, making it susceptible to interception and abuse. The concept of Single Sign-on must thus be extended to Secure Single Sign-on (SSSO) by also ensuring aspects of confidentiality and integrity. (Louwrens, 1996)

Secure Single Sign-On is thus defined as the ability to provide principals (users), after being authenticated once, with transparent access to a variety of services through a defined set of credentials from trustworthy certification authorities, via authorized applications, while maintaining end-to-end confidentiality, integrity and auditability.(Open Horizon, 1996; Louwrens, 1996)

SSSO, to be implemented successfully, requires a carefully architected security design, consistent security policy enforcement and a single view of security management and auditing. The challenge is to apply these requirements to heterogeneous and distributed computing environments.

When an organization is faced with the dilemma of selecting or building a solution for its SSSO requirements, there are very few, if any, standards to assist in making the right choice. Off -the -shelf products are generally immature and seldom cater for all circumstances. It is, therefore, essential to be able to measure products and in-house solutions against a common standard. The aim of this paper is to provide such a reference framework, against which SSSO solutions can be evaluated.

This paper is structured as follows: Section 2 gives an overview of the concept of Secure Single Sign-on; Section 3 sets out the requirements for SSSO; Section 4 introduces a Reference Framework for Evaluating SSSO solutions; and Section 5 the Conclusion.

2 OVERVIEW OF SSSO

2.1 Security Services Required for SSSO

In order to implement SSSO, as previously defined, the total or partial integration of the following security services into the solution is essential: Authentication, Authorization/Logical Access Control, Security Management and Administration, Auditing, Cryptographic services, Key management, Integrity, Confidentiality and Availability. (Louwrens, 1996) The required components of a comprehensive SSSO solution as defined by Pfleeger (1989), are briefly discussed below:

- **Authentication.** This requirement is essential to confirm the identity of a communicating party, ensuring that only authorized people are allowed access. It is also essential that authentication happens on an individual level, i.e. any action can be uniquely linked to a specific subject or object, enforcing total accountability.
- **Authorization and Logical Access Control.** Authorization is enforced by logical access control. **Logical access control** ensures that only **authorized** users (subjects) get access to those resources (objects) they are authorized to access.
- **Integrity.** Data Integrity means that assets can be modified only by authorized parties. This is implemented using message authentication codes (MAC's), to prevent it from being undetectably tampered with.
- **Confidentiality.** Confidentiality or secrecy means that the assets of a computing system are accessible only by authorized parties. This is usually implemented through encryption.
- **Availability.** Availability means that assets are available to authorized parties. An authorized party should not be prevented from accessing those objects to which he or she or it has legitimate access.

- **Non- repudiation.** This means proof that a message received was not fabricated by someone other than the declared sender. This is implemented using digital signatures.
- **Security Management.** Effective management is the basis of any Information Security system. Full management facilities are needed. These include:
- **Administration.** User friendly administrative tools to administer logical access control through profiles.
- **Auditing.** For effective management of Information Security, full and proper auditing facilities should be available on all levels of the Environment. These auditing facilities include the logging of all relevant (selected) actions, and the proper tools to investigate the audit logs. On-line exception reporting should be possible.
- **Cryptographic Services and Key Management.** To implement Confidentiality and Non-repudiation and Strong Authentication, cryptographic services are needed.

2.2 New vulnerabilities introduced by SSSO

While single sign-on reduces the likelihood of users compromising their passwords, it can also introduce new vulnerabilities. These are:

- **Single-point-of-failure.** Some single sign-on solutions rely on dedicated authentication servers to support users. Multiple users may be inconvenienced if a server goes down. This can be addressed by introducing back-up (fail-over) servers, as well as alternate access paths to these servers. (Deloitte & Touche, 1996)
- **Multiplied access.** The risk of unauthorized access may be increased rather than reduced where single sign-on solutions are introduced, which if a user's password is disclosed, permit unauthorized access to all systems accessible to the user. The use of secure physical tokens for authentication can mitigate this vulnerability.
- **Insecure storage.** Sign-on data enabling access to multiple systems is exposed to unauthorized disclosure if stored insecurely by target systems, workstations or servers. Critical sign-on data should be stored in encrypted format where possible.
- **Insecure transmission.** Sign-on data is exposed to unauthorized interception if transmitted in clear, particularly when transmission is across networks using broadcast protocols which expose sign-on data to interception at all network nodes. Sign-on data should be encrypted when transmitted over networks. (Stanley, 1996)

2.3 Obstacles and Pitfalls to be considered

Without a proper design, implementing a SSSO system can create pitfalls for users and administrators. There are several obstacles and pitfalls to be considered:

- **Immaturity of Products.** The latest generation SSSO products are generally immature. No product as yet offers a perfect solution and there are obvious dangers in installing fast-moving technology which may be subject to bugs or unforeseen limitations, or obsoleted by further advances.
- **Lack of experience.** The limited number of successful SSSO implementations, plus lack of first-hand experience leaves information security managers, IS strategic planners and system developers uncertain about when and how to introduce single

sign-on, and which solutions to specify. Uncertainties are compounded because the capabilities of single sign-on products, and the ease with which they can be introduced , are frequently oversold.

- **Uncertainty about Costs.** The relative immaturity of the field as a whole and the pace of change mean that there is widespread uncertainty about the costs and benefits of single sign-on. Costs are difficult to assess without detailed study. They vary depending on the nature of the single sign-on solution selected, method of implementation, number of users and the number and diversity of target platforms and applications, and can be substantial for implementations supporting many users. (Stanley, 1996)
- **Scalability.** Scalability of the solution seems to be another major concern, both in capability of handling peak demands, such as concurrent sign-on's in the mornings and manageability across the enterprise.
- **Dependence on Architecture.** SSSO solutions are dependent on the enterprise systems architecture, e.g. dumb terminal - host, client- server, two- or three-tier architectures, etc. Therefore, not every SSSO solution would be compatible with a given enterprise architecture.
- **Catering for future requirements.** There are few, if any, software solutions that accommodate all of the top operating system environments. So tailoring the right mix of solutions to the enterprise's information technology architecture and strategic direction is essential. Achieving technical integration across multiple platforms and applications is a major challenge, particularly when target systems and applications are themselves subject to constant change. Not all SSSO solutions are capable of meeting this challenge. Some entail significant change to platforms, applications and overall system architectures.
- **Cost of features not required.** Most SSSO software packages include more features than just SSO, thus the price paid for SSO includes other capabilities, which may be redundant with existing controls and management tools.
- **Key Management.** Good Key management is essential to ensure secure key generation and distribution. This should be done from a trusted key management center or Certification Authority.
- **Establishing Single Userids.** Establishing single userids in an enterprise is not a trivial management and administrative task. It is essential that implementation can be done in a phased manner. (Deloitte & Touche, 1996)

2.4 SSO Solution Types

There are several software products on the market that facilitate the implementation of single sign-on strategies. Available solutions fall into five main types:

- **Synchronization solutions.** These set a user's sign-on data to a consistent value on all target systems which he or she is entitled to access. (Stanley, 1996)
- **Scripting solutions.** Another technique for implementing single sign-on is scripting. This does not require changes to a user's existing sign-on data. A script is a string of commands and values that would normally be entered into the system. The script organizes these commands and values into a single module. So instead of executing

each command individually, the script is executed by the SSO server to provide the user with the requested access. (Deloitte & Touche, 1996)

- **Proxies and Trusted Hosts.** Another technique, using Proxies and Trusted Hosts, does not require any additional software. By setting up trust-relationships between hosts, and using proxy mechanisms, trusted users are logged on to any host in the trust-relationship without having to enter a userid or password. (Gregory, 1994)
- **Trusted Authentication Server solutions.** These provide a more secure, encryption-based authentication. With trusted authentication servers, a common database is built containing a list of users and cross-references to valid host systems, userids and passwords. When a user accesses the network, they sign -on through the trusted authentication server and are granted access to the host systems. This type of solution normally requires applications and systems to be specially adapted to enabled the security features to be utilized, i.e. implementation of DCE, or Kerberos.
- **Hybrid solutions.** These combine a trusted authentication server solution with one or more of the other types to allow single sign-on to be achieved across both specially adapted and unadapted systems. This allows new systems to utilize the benefits of trusted authentication, while using scripting for legacy applications. (Stanley, 1996)

Of the above, only **Trusted Authentication Server** solutions fit squarely into the SSSO concept. **Hybrid solutions** contain all of the SSSO functionality, but the extension of functionality with other methods, may actually reduce the level of security to some applications and systems.

3 REQUIREMENTS FOR SSSO

3.1 General Considerations

When selecting a SSSO solution, numerous requirements can be considered. These must include functional requirements as well as other requirements like product maturity, installed base, supplier stability, level of support available, introduction of new vulnerabilities and, obviously, cost. Only functional requirements and the introduction of vulnerabilities are considered in the list below.

The list of requirements was compiled from various sources, including the list compiled by the Georgia RACF User's Group (1995), and added to by the authors, from practical experience in assisting with the selection of a SSSO solution for a major bank. This list is not exhaustive and some variation could be expected for specific computing environments. For brevity's sake, 'Nice to Have' features have been omitted.

Requirements are grouped according to the security services required as identified in 2.2 above. Furthermore requirements are ranked as **Essential** or **Recommended** and are uniquely identified by a code which indicates the type of security service it provides. (Refer to section 4).

3.1.1 Authentication

- **Single Sign-on (AUTH E01).**The product should enable authentication by a single logon to all enterprise resources, by a single userid and password, or token/biometric plus password. Re-authentication should only be required if considered necessary for a enhanced level of security.
- **Support common Password Rules (AUTH E02).** All common password rules should be supported.
- **Support a Standard Primary USERID Format (AUTH E03).** All common USERID syntax rules should be defined by the administrator. The product should include features to translate unlike USERIDs from different platforms so that they can be serviced.
- **Auto Revoke after a number of invalid Attempts (AUTH E04).** Users should be revoked from system access after a specified number of invalid attempts. This threshold should be set by the administrator.
- **Capture Point of Origin Information (AUTH E05).** The product should be able to capture telephone caller ID or phone number for dial-in access information if needed.
- **Support Sign -on's from Variety of Sources (AUTH E06).** The product should support signons from a variety of sources, like LAN/WAN, workstations, Laptops/Notebooks, Dial-in, and Dumb terminals without compromising the level of security.
- **Ensure USERID Uniqueness (AUTH E07).** The product should ensure that all USERIDs are unique, so that no two USERIDs can be the same.
- **Authentication Server should be Portable (AUTH R08).** The product should provide for the authentication server to reside on any platform that the product can control to ensure portability.
- **Support Public/Private Key technology (AUTH R09).** The product should support asymmetric encryption technologies such as RSA. This can be used for strong authentication and non-repudiation services.
- **Support Tokens/Biometrics (AUTH R10).** The product should support the use of security tokens such as smart cards, challenge-response tokens and biometrical devices to enable their use on any platform.

3.1.2 Access control and Authorization

- **Differentiated administration Privileges (ACL E01).** The product should support differentiated administration privileges at the different levels of control.
- **Default Protection unless specified (ACL E02).** The product should provide for the protection of all resources and entities as the default, unless the opposite protection for only those resources is specified.
- **Ability to support Scripting (ACL E03).** The product should support the use of scripting for legacy applications and systems.

- **Physical Terminal/Node/Address Control (ACL R04).** The product should have the ability to restrict or control access on the basis of a terminal, node, or network address.
- **Single Point of Authorization (ACL R05).** All authorizations should be made via a single point, i.e. an authentication server. This provides not only a single point of administration for the product, but also reduced network security traffic.
- **Support Standard Ticket/Certificate Technologies (ACL R06).** The product should support standard ticket or certificate technologies such as IBM's RACF Pass Tickets, Kerberos certificates or SESAME Privilege Attribute Certificates (PAC's), ensuring that the product can reside in an environment using ticket / certificate technology to provide security authentication and authorization. (IBM, 1994; SESAME, 1996)
- **Support Masking/ Generics (ACL R07).** The product should support security profiles containing generic characters that enable the product to make security decisions based on groups of resources as opposed to individual security profiles.
- **Allow Delegation Within Power of Authority (ACL R08).** The product should allow an administrator to delegate security administration authority to others at the discretion of the administrator within his/her span of authority.

3.1.3 Data Integrity/Confidentiality/Availability

- **No Clear-text Passwords (DICA E01).** At no time should any password be available on the network or in the security database in clear, human- readable form. The only exception is the use of dumb terminals where the terminal does not support encryption techniques. Where dumb terminals have to be used, 'one-time' passwords should be considered, possibly together with challenge-response tokens.
- **Integrity of Security DB(s) (DICA E02).** The database used by the product to store security information and parameters should be protected from changes via any source other than the product itself.
- **Failsoft Ability (DICA E03).** The product should have the ability to perform at a degraded degree without access to the security database. This enables the product to at least work in a degraded mode in emergency in such a fashion that security is not compromised.
- **Inactive User Time -out (DICA R04).** All users who are inactive for a set period of time during a session should be timed out and signed off all sessions.
- **Commercial Standard Encryption (DICA R05).** The encryption used in the product should be standard.
- **Option for Single or Distributed Security Databases (DICA R06).** The product should support the option of having a single security database or several distributed security databases on different platforms.
- **Inactive User Revoke (DICA R07).** All users who have not signed on within a set period of time should be revoked. The period should be configurable by the administrator.

- **Optional Application Data Encryption (DICA R08).** The product should provide the optional ability to interface to encrypted application data if the encryption techniques are provided.
- **Key Management (DICA R09).** A Trusted Key Management Center / Certificate Authority is essential when dealing with cryptographic keys. This is especially true if asymmetric encryption is to be used.

3.1.4 Security Administration Management and Auditing

- **Single point of Administration (SAMA E01)**. The product should provide for full administration from a single point, if required.
- **Role - profile based (SAMA E02).** The product should enable the grouping of like Subjects (users) and Objects into role based profiles, using Discretionary Access Control. This will enable more efficient administration of access authority.
- **Full Audit Trail (SAMA E03).** All changes, modifications, additions, and deletions to the security database should be logged. The audit trail for all systems should be configurable to suit different security requirements and reduce overhead. The degree of logging should be controlled by the administrator.
- **Single Revoke/Resume for All Platforms (SAMA E04).** The product should support a single revoke or resume of a USERID regardless of the platform.
- **Ability to Enforce Enterprise Security Rules (SAMA E05).** The product should provide the ability to enforce security rules over the entire enterprise regardless of platform. This will ensure the implementation of a single security policy and consistent security over resources on all protected platforms.
- **Ability to Trace Access (SAMA E06).** The product should enable the administrator to be able to trace access to systems regardless of system or platform.
- **Scoping and Decentralization of Control (SAMA E07).** The product should be able to support the creation of spans of control so that administrators can be excluded from or included in certain security control areas within the overall security setup.
- **Synchronization Across all Entities (SAMA E08).** The product should synchronize security data across all entities and all platforms. This ensures that all security decisions are made with up-to- date security information.
- **Real-Time and Batch Update (SAMA E09).** All changes should be made on-line /real-time. The ability to batch changes together is also important to enable easy loading or changing of large numbers of security resources or users.
- **Customize in Real -time (SAMA E10).** The ability to customize or make changes to those features which are customizable without re-initializing the product, is important.
- **User Defined Fields (SAMA E11).** The product should have a number of user customizable/ user-defined fields.
- **Support Customized Reporting (SAMA E12).** The product should have the ability to create customized reports using SQL query or similar reporting tools to produce security setup reports/queries.

- **Support User Exits/Options (SAMA R13).** The product should support the addition of user exits/options that could be attached to the base product at strategically identified points of operation.
- **Customizable Messages (SAMA R14) .** The product should support the use of customized security messages.
- **Common Control Language Across All Platforms (SAMA R15).** The product should feature a common control language across all serviced platforms so that administrators do not have to learn and use different commands on different platforms.
- **Ability to Recreate from Logged Information (SAMA R16).** Information logged by the system should be able to be used to "backout" changes to the security system. Example: used to recreate deleted resources or users. This enables mass changes to be "backed out" of production or enables mass additions to be made based on logged information.
- **Administration for multiple Platforms (SAMA R17).** The product should provide for the administration of the product for any of the supported platforms.
- **Ability to Create Security Extract Files (SAMA R18).** The product should have the feature to produce an extract file of the security structure and the logging/violation records.
- **Test Facility (SAMA R19).** The product should include a test facility to enable administrators to test security changes before placing then into production.

3.1.5 General Functionality

- **Backward Compatible (GFR E01).** All releases of the product should be backward compatible or release independent. Features of new releases should co-exist with current features and not require a total reinstallation of the product.
- **Conformance to Standards (GFR E02).** The product should be able to interface with existing application, database, or network security by way of standard security interfaces. This will ensure that the product will mesh with existing security products installed. Where possible, the product must conform to the known and accepted international standards. This will go a long way in ensuring that the product is flexible and "future proof".
- **Phased Implementation (GFR E03).** The product should be able to be selectively implemented for individual users, systems or resources to enable ease of implementation and migration for legacy systems. This will also allow the product to be 'phased in'.
- **Consistent User Interface (GFR R04).** The product should have a common and familiar procedure for users to gain access to their systems and applications.
- **Ease of Use (GFR R05).** The product should make use of a standard GUI interface that is both consistent and intuitive to use. The interface may vary slightly between platforms (i.e. Windows, OS/2, Xwindows, etc.) but should retain the same functionality. This ensures operating consistency and lowers training needs.(CKS, 1996)

- **Flexible Cost (GFR R06).** The cost of the product should be reasonable. Several cost scenarios should be considered such as per seat, CPU, site licensing and MIPS pricing. Pricing should include disaster recovery scenarios.
- **Certification (GFR R07).** The product should be certified in terms of acknowledged international standards, i.e. ITSEC E2 Level , C2 level of the US Orange Book. This will give a more accurate measurement of the security level obtainable if the product is properly installed and configured.
- **One Single Product (GFR R08).** The product should be a single product, not a compendium of several associated products. Modularity for the sake of platform-to platform compatibility is acceptable and favored.
- **Software Release Distribution (GFR R09)**. New releases of the product should be distributed via the network from a single distribution server of the administrator's choice. This enables an administrator to upgrade the product on any platform without physically moving from platform to platform.

4 FRAMEWORK FOR EVALUATING SSSO SOLUTIONS

4.1 Essential Functionality

Table 1 below indicates the functionality considered Essential for a Secure Single Sign-on solution. These are the requirements that **must** be satisfied.

Table 1 Essential Functionality for a Secure Single Sign-on solution.

Reference	*Essential Functionality*
AUTH	*Authentication*
AUTH E01	Single Sign-on
AUTH E02	Support common password rules
AUTH E03	Support a Standard Primary USERID Format
AUTH E04	Auto Revoke after a number of Invalid Attempts
AUTH E05	Capture Point of Origin Information
AUTH E06	Support Sign- on's from Variety of Sources
AUTH E07	Ensure USERID Uniqueness
ACL	*Authorization and Access Control*
ACL E01	Differentiated Administration Privileges
ACL E02	Default Protection unless specified
ACL E03	Ability to support scripting
DICA	*Data Integrity/Confidentiality/Availability*
DICA E01	No Clear Text Passwords
DICA E04	Integrity of Security DB(s)
DICA E05	Failsoft Ability
SAMA	*Security Administration Management and Auditing*
SAMA E01	Single point of Administration
SAMA E02	Role profile based
SAMA E03	Full Audit Trail
SAMA E04	Single Revoke/Resume for All Platforms

SAMA E05	Ability to Enforce Enterprise Security Rules
SAMA E06	Ability to Trace Access
SAMA E07	Scoping and Decentralization of Control
SAMA E08	Synchronization Across all Entities
SAMA E09	Real-Time and Batch Update
SAMA E10	Customize in Real-Time
SAMA E11	User Defined Fields
SAMA E12	Support Customized Reporting
GFR	*General Functionality*
GFR E01	Backward Compatible
GFR E02	Conformance to Standards
GFR E03	Phased Implementation

4.2 Additional Recommended Functionality.

Table 2 below lists the additional functionality recommended for a Secure Single Sign-on solution. Although these items are not essential, much value can be added to the eventual successful implementation of a solution.

Table 2 Additional Recommended Functional Requirements for a Secure Single Sign-on solution.

Reference	*Recommended Functionality*
AUTH	*Authentication*
AUTH R08	Authentication Server should be Portable
AUTH R09	Support Public/Private Key technology
AUTH R10	Support Tokens/Biometrics
ACL	*Authorization and Access Control*
ACL R04	Physical Terminal/Node/Address Control
ACL R05	Single Point of Authorization
ACL R06	Support Standard Ticket/Certificate Technologies
ACL R07	Support Masking/ Generics
ACL R08	Allow Delegation Within Power of Authority
DICA	*Data Integrity/Confidentiality/Availability*
DICA R02	Inactive User Time -out
DICA R03	Commercial Standard Encryption
DICA R06	Option for Single or Distributed Security Databases
DICA R07	Inactive User Revoke
DICA R08	Optional Application Data Encryption
DICA R09	Key Management
SAMA	*Security Administration Management and Auditing*
SAMA R13	Support User Exits/Options
SAMA R14	Customizable Messages
SAMA R15	Common Control Language Across All Platforms
SAMA R16	Ability to Recreate from Logged Information

SAMA R17	Administration for multiple Platforms
SAMA R18	Ability to Create Security Extract Files
SAMA R19	Test Facility
GFR	*General Functionality*
GFR R04	Consistent User Interface
GFR R05	Ease of Use
GFR R06	Flexible Cost
GFR R07	Certification
GFR R08	One Single Product
GFR R09	Software Release Distribution

For detailed explanations of the requirements listed above, refer to section 3 of this paper.

4.3 The SSSO Reference Framework

As stated previously, this paper does not concern itself with selection criteria other than the functional requirements for a Secure Single Sign-on solution. It assumes that the other related criteria like product maturity, installed base, supplier stability, level of support available, introduction of new vulnerabilities and cost, will be investigated.

The proposed reference framework for selection of Secure Single Sign-on solutions consist of two main aspects, namely:

- Firstly, considering the essential requirements. These requirements are not weighted. The decisions are binary and should the solution not conform to every one, it should not be considered further.

- Secondly, after the essential requirements have been satisfied, the additional recommended functionality should be considered. Unlike the essential requirements, these requirements should be weighted. The individual weights must reflect the individual needs of the enterprise environment.

Having completed the two steps above, an objective comparison of the available Secure Single Sign-on solutions can be made. The best candidate can then selected for piloting. Bearing in mind the possible pitfalls for the prospective buyer, it is prudent to pilot the SSSO solution with a selected group of users in a well defined computing environment, after which full implementation can follow. Refer to figure 1 below.

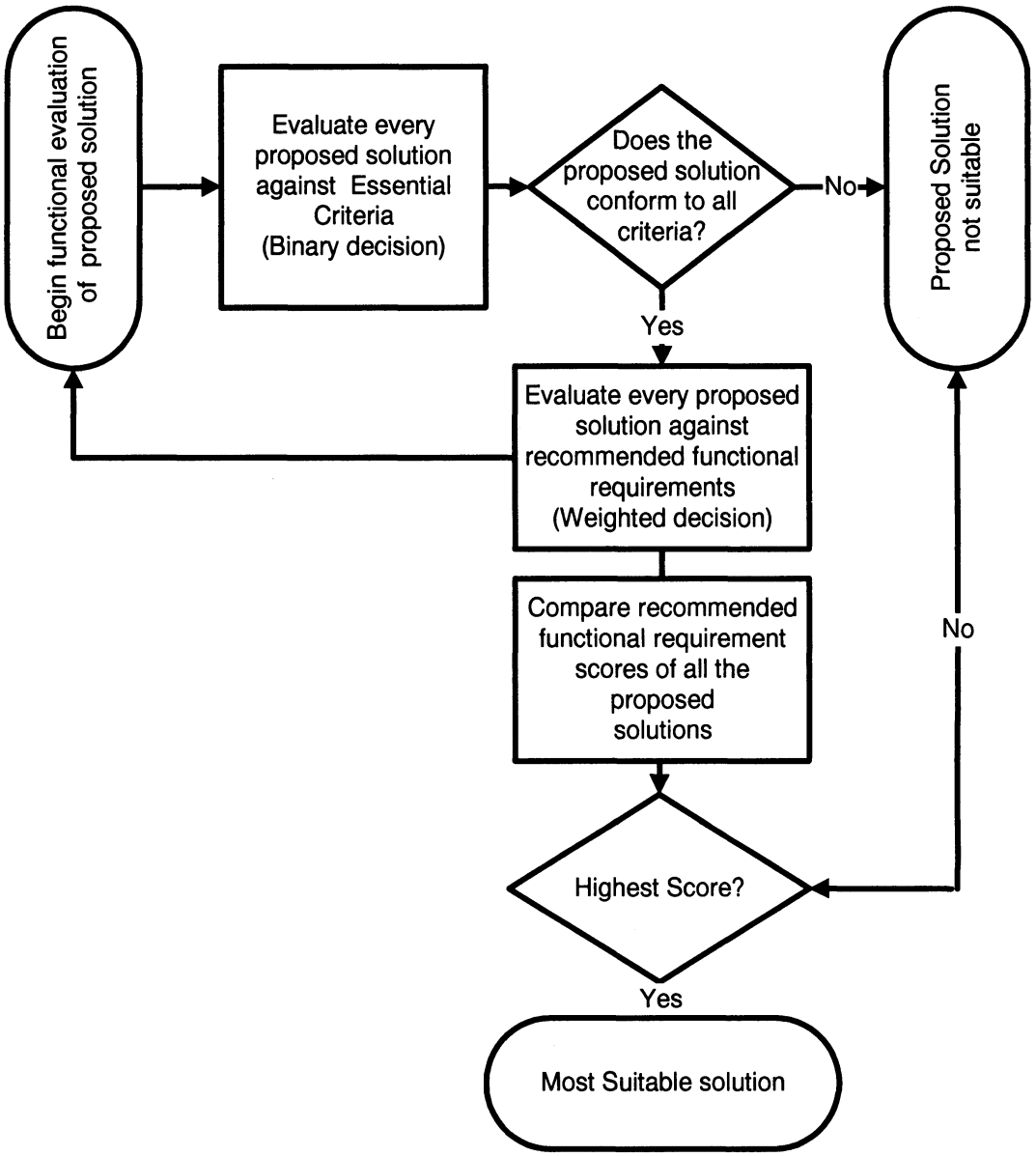

Figure 1 The SSSO Evaluation Flowchart.(Louwrens, 1996)

5 CONCLUSION

Using traditional security approaches with today's heterogeneous computing environments can make systems unusable, lead to reduced productivity and potentially compromise the security of the systems. By introducing single sign-on functionality alone, some of the security issues, like sharing of passwords and enhance productivity

may be addressed, but it may introduce other vulnerabilities like single points of failure, multiplied access, insecure storage and insecure transmission of sign-on data. These vulnerabilities can only be addressed by taking a holistic view of the total enterprise security environment and implementing a properly architected Secure Single Sign-on (SSSO) solution. Of the five identified types of solutions, only Trusted Authentication Server and Hybrid solutions can provide Secure Single Sign-on if properly implemented. By following the approach as illustrated in the Reference framework for the Evaluation of a SSSO solution as presented in this paper, an objective decision on the most appropriate SSSO solution can be made.

Nevertheless, it is clear that a well designed and architected SSSO solution can provide added levels of security and substantially reduce security administration and security management workloads.

6 REFERENCES

Pfleeger, C.P. (1989) *Security in Computing*, Prentice-Hall.

Louwrens, C.P. (1996) *Single Sign-on in Heterogeneous Computing Environments*, MSc Dissertation, Rand Afrikaans University.

International Business Machines Corporation (1994),*Secured Single Signon in a Client/Server Environment*, International Support Organization, Poughkeepsie Center, NY 12601-5400.

Stanley, A. et al, (1996) *Position Paper Single Sign-on*, European Security Forum.

Georgia RACF Users' Group, *Single Signon Functional Requirements*, September 18, 1995, http://widow.mindspring.com/~ajc10/sso.html

Deloitte & Touche (1996) *Taking the Mystery out of .. Single Sign On*, http://www.dttus.com/dttus/publish/mystery/singsign.htm

ICL Access Manager Business Unit, *AccessManager*, Eskdale Road, Winnersh, Wokingham, Berkshire, RG11 5TT, http://www.icl.com/access

CKS (1996), *CKS MyNet, MyNet Concepts and Facilities*, Publication reference : MyCF0.01

Computer Associates (1996) *CA-Unicenter/Single Sign-On*, Concepts and Facilities v.1.0.

Gregory, N. (1994) *One Click, Many Services, Security- Single Signon using Proxies &Trusted Hosts*, ACO User Forum, http://www1.psi.ch/www_aco_hn/documentation/uf940525.html.

Open Horizon (1996) *Enterprise Client/Server Secure Single Sign-On*, Open Horizon White Paper, http://www.openhorizon.com:80/whtpaper/sso/sso0369.htm.

SESAME (1996), *Secure European System for Applications in a Multi-vendor Environment*, http://www.esat.kuleuven.ac.be.80:sesame3.html.

7 BIOGRAPHY

Cecil (Buks) Louwrens is currently an MSc student at the Department Of Computer Science of the Rand Afrikaans University in Johannesburg, South Africa. The contents of this paper forms part of his MSc studies on Single Sign-on in Heterogeneous Computing Environments.

Prof Sebastiaan (Basie) von Solms is Head of the Department of Computer Science at the Rand Afrikaans University in Johannesburg, South Africa. He is also the South African representative on Technical Committee 11 [Information Security] (TC11) of the International Federation for Information Processing (IFIP), and is present Chairman of TC11.

Prof von Solms has published numerous research papers on Information Security, and had spent 1995 on a 12 month industry sabbatical at IBM Development Laboratory at Hursley in the UK. He is presently also a consultant on Information Security to IBM South Africa. He is also a member of the Review Panel of the journal Computers and Security, as well as a member of the Editorial Board of the South African Computer Journal.