White paper

So You Want Single Sign-On
Table of Contents

1. Introduction................................................................................................................... 1
2. Setting a Baseline for Requirements .............................................................................. 1
   2.1 What Does “Single” Mean?....................................................................................... 1
   2.2 What Does “Sign-On” Mean?................................................................................... 2
   2.3 SSO Design = f(Stakeholder ) = “Single” + “Sign-On”............................................. 3
3. Designing for Benefits ................................................................................................. 4
   3.1 Sources of Value..................................................................................................... 4
   3.2 Generating Value Requires a Comprehensive Design........................................... 5
4. Bounding the Design ................................................................................................. 10
   4.1 Common SSO Technical Approaches ..................................................................... 10
      4.1.1 Password Synchronization ............................................................................ 10
      4.1.2 Centralized Directory .................................................................................... 11
      4.1.3 Intercepted Requests...................................................................................... 13
      4.1.4 Middleware/Translator .................................................................................. 15
   4.2 Sample System Context ......................................................................................... 17
   4.3 So What’s the “So-What” ..................................................................................... 19
5. Lessons Learned........................................................................................................ 20

Table of Tables

Table 1. Primary User Group Interest in SSO Design Issues............................................. 3
Table 2. Single Sign-On Value Drivers .............................................................................. 4
Table 3. SSO Design Considerations ................................................................................ 5
Table 4. Overview of Password Synchronization............................................................. 11
Table 5. Overview of Centralized Directory .................................................................... 13
Table 6. Overview of Intercepted Request Proxy ............................................................... 14
Table 7. Overview of Middleware/Translator .................................................................... 17

Table of Figures

Figure 1. Password Synchronization ............................................................................... 10
Figure 2. Centralized Directory ....................................................................................... 12
Figure 3. Intercepted Request Proxy ............................................................................... 13
Figure 4. Middleware/Translator .................................................................................... 16
Figure 5. SSO System Context ....................................................................................... 18
1. Introduction

If client interest is any indication, Single Sign-On (SSO) is today’s Holy Grail of technology services. Users want it. IT is hunting for it. Vendors say they have it. Whoever does it will be a hero. What’s more, the benefits are clear: reduced complexity for end-users, leading to improved productivity, reduced help desk demand and simplified administration. But, SSO remains more of a technology wish than a technical reality. So what is the problem?

BT Syntegra believes that the SSO expectation has not been met for four reasons.

♦ First and foremost, there is no established definition of SSO that is shared by end-users, security officers, and administrators. Depending on who you are and your needs, requirements differ.

♦ Second, no single product does all of what is required. Users may gain access to the network via the desktop/LAN, web, and/or a remote access service. Once on the network, they may require access to applications, transactions, devices, and/or special services. Products exist that solve subsets of the problem. However, current products typically support only one way a user may access the network and cannot be configured to control access to all and only the resources a user is entitled to use.

♦ Third, end users underestimate the complexity of the problem. Business and security requirements must be translated into electronic rules and processes that allow a common identity to be shared across the variety of ways a user may enter the network. Moreover, users do not appreciate that identity and privileges cannot be separated from each other and retain security control.

♦ Fourth, IT has not produced a user-centric, role-based management and administration design to serve as the guiding light to integrate products that contribute to a comprehensive solution for a given user community.

Given these conditions, this paper will answer these questions:

♦ What do “single” and “sign-on” imply for baseline requirements given the range of users and technical environments users intend to access?

♦ What does it mean to design for benefits; what are design considerations to extract that value?

♦ What are common design approaches and what do they mean for a comprehensive system context?

♦ What are some lessons BT Syntegra has learned?

2. Setting a Baseline for Requirements

2.1 What Does “Single” Mean?

BT Syntegra finds that “single” is a matter of user expectation. Expectations vary given the users and their organizational/commercial relationship. That said, there appears to be three universal truths: Users want it easy. Security officers want it safe. Administrators want it controlled.

These differing perspectives lead to seven views of “single”:

♦ Single identity and credential

♦ Single session logon

♦ Every single resource use is authorized
Every single transaction is authorized and its delivery ensured
♦ Single point of enforcement
♦ Single process to confirm identity then issue accounts, credentials, and rights
♦ Single repository of users, identity, credentials, and rights

Each major constituency (user, security office, and administrator) describes success differently. From the point of view of the user, SSO is a single identity, authenticated with a single credential, entered one time that provides access to all applications and services. From the perspective of the security officer, SSO is granular validation of every request based on rights authorized by information owners and on a recent authentication of the user, performed in a consistent manner at a central enforcement point. For the administrator, SSO is a consistent process that enables users to request access to electronic resources and ensures appropriate authorities have granted access based on business need.

2.2 What Does “Sign-On” Mean?

While the notion of “single” is challenging on its own, the range of technical sessions to which a user may “sign-on” compounds the problem. BT Syntegra divides electronic requests and privileges into the five following service areas (sample services are offered to clarify the grouping):
♦ LAN services: NOS accounts (UNIX, Microsoft Windows NT, Lotus Notes, NDS), email, Microsoft Windows applications
♦ Remote services: VPN, RAS, device (network address) access, transaction service/protocol
♦ Web services: URLs
♦ Special services: PKI, CTI, RACF, knowledge management, building access
♦ Application/transaction services: ERP systems, legacy applications

These services are distinguished by the points at which user identity is authenticated and rights are authorized. LAN access is authenticated when the user launches a desktop session. A server creating a remote connection authenticates a user’s access to the network. When required, web access is authenticated at the web server during an HTTP session. Special accounts and legacy applications may have their own method of confirming identity and authorizing access to all or parts of their functionality and data.

As if the variety of where sign-on occurs is not enough, developers have been especially creative in building different—if not situation-specific—sign-on mechanisms. Mechanisms vary by protocol, authentication credential, and rules to identify the user, just to name a few. Is it any wonder users want to pass through a single entry with a single entry key to unlock everything they need?

Sign-on is a security process. It exists to protect the network and its services/and applications from unauthorized, unintended, or malicious use. As a security service, SSO must not compromise electronic asset protection to meet end user expectations for ease of use. The security office function must ensure that SSO deployment does not introduce new threats, such as persistent connections, inadequate confirmation of user identity, misplaced faith in the social habits of users (will they protect themselves?), and/or reduced granularity of control.
2.3 SSO Design = f(Stakeholder) = “Single” + “Sign-On”

Pardon the mathematical notation, but it is accurate; SSO design is a function of or depends on the needs and interests of all stakeholders. Stakeholder interest can be organized by subgroups within user communities. Table 1 summarizes expectations and typical access requirements for the range of users that an SSO solution may need to support. An asterisk indicates that the user community subgroup has a vested interest in that SSO design issue.

### Table 1. Primary User Group Interest in SSO Design Issues

<table>
<thead>
<tr>
<th>Stakeholder / User Community</th>
<th>Expectations (Single …)</th>
<th>Access (Sign-On)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identity/credential</td>
<td>Session</td>
</tr>
<tr>
<td>Employees:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Nomadic/remote</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Security officer</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Administrator</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Information owner</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Contractors:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Substitute employee</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Off-site developer</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Service provider</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Trading partner:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Supplier</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Public:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recurring consumer</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Information seeker</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

The following observations pertain to Table 1:

- Who you are defines SSO functional requirements. When all potential users are considered, a comprehensive solution is required.
- There is significant value for network access and Web support. They should be part of an initial deployment plan.
- When a user has limited access (such as off-site developers, service providers and the public), there is no value in SSO from the single identity/credential perspective, but there is still a need to manage access to resources. Therefore, SSO must support both individual authentication and fine-grained authorization.
♦ If end user benefits are emphasized at the expense of administrative simplification and quality of security service, a comprehensive solution design will not emerge.

BT Syntegra believes that an SSO system design should be based on the generation of benefits to the enterprise. This ensures that the needs of all stakeholders (users, security officers and administrators) are heard. In fact, developers must design for benefits.

3. Designing for Benefits

3.1 Sources of Value

Users, administrators, and security officers are all stakeholders in an SSO solution. It is not in the best interest of the enterprise to provide an end user facility that is difficult to administer or threatens the security of the network. Therefore, combined needs and interests define the design objectives from which benefits flow. Table 2 summarizes cash and quality value drivers.

<table>
<thead>
<tr>
<th>Financial</th>
<th>Productivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ Reduce knowledge worker labor</td>
<td>♦ Eliminate time to sign-on</td>
</tr>
<tr>
<td>♦ Reduce administrative/help desk labor</td>
<td>♦ Reduce effort to maintain user lists, provision accounts/electronic identity/access rights and maintain authentication credentials</td>
</tr>
<tr>
<td>♦ Reduce exposure to security breaches and the associated recovery, regulatory and/or litigation expense</td>
<td>♦ Enable full productivity of new hire/transferee at job start</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Satisfaction</th>
<th>Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ Easier for user; single identity and credential</td>
<td>♦ Security practices and rules universally applied across enterprise</td>
</tr>
<tr>
<td>♦ Rapid response to user requests</td>
<td>♦ Organizational changes tied to access privileges</td>
</tr>
<tr>
<td>♦ Tie to physical asset provisioning</td>
<td>♦ Network changes tied to security rules</td>
</tr>
<tr>
<td>♦ Support self-service and widely distributed approval and /control</td>
<td>♦ Logging, query and reporting facility compliant with expected regulations for data access and use</td>
</tr>
<tr>
<td>♦ Reduce complexity of security rule development and maintenance</td>
<td>♦ Security practice management</td>
</tr>
<tr>
<td>♦ Reuse authoritative data</td>
<td></td>
</tr>
</tbody>
</table>

SSO yields cash returns. The marketplace has emphasized the productivity effect on end users that must access numerous systems to perform their duties. Much has been said about the non-productive effort spent logging on to multiple systems every day. Not as much has been said about the administrative support needed to deal with forgotten passwords or the administrative complexity to provision and maintain accounts, passwords, and rights. Nor has much been said about the impact on the productivity of every new employee/transferee at job start. A well-designed system will consider all aspects of the user and administrator experience if it is to reduce cost, improve the quality of security, and reduce exposure to security breaches.
A well-designed system will improve the quality of security services. The satisfaction of all stakeholders should be considered. Complexity should be reduced. While it is true that the critical success factor for SSO is end user satisfaction afforded by a single identity and credential, operations within the security office and administrative centers should be simplified as well.

The system should ensure timely response to ad hoc requests and planned organizational changes. It should be easy for security officers and administrators to perform their functions. It should promote self-service, reuse existing data to avoid entry errors, automate the process where possible and widely distribute the administrative burden to information owners and decision makers where human control is mandatory. Ideally, the SSO system should be integrated with the process that provides physical resources such as laptops, telephone connections, and building access. In short, the SSO system should focus on the end user but provide cost-effective tools to all the players involved in making users effective.

Moreover, the quality of the SSO solution should be judged in terms of the execution of business-driven security policies. Rarely does SSO marketing collateral discuss the consistency and reliability of administrative processes, or the timeliness and accuracy of the data on which enforcement depends. Yet, security administration exists to translate the security policy and business intentions into enforcement rules.

Data used by security technologies to make enforcement decisions represent the single greatest threat to electronic security. Therefore, the SSO system should promote consistent and universal administrative practices across the enterprise. The SSO system should detect changes in databases that contain authoritative lists of users and their business relationship, and to databases that contain authoritative lists of network resources. The system then should take appropriate action, be it adding/suspending rights, adjusting access rules, or notifying the administrator for manual intervention. It should provide the tools to determine which users read or modify what data and when. And it should enable the security office to oversee actions in a distributed environment, ensuring compliance with best practice.

### 3.2 Generating Value Requires a Comprehensive Design

If an SSO investment is to yield expected benefits, designers must make a conscious effort to link design requirements with benefit generation. Table 3 summarizes design considerations BT Syntegra has found to be important in business-driven SSO systems.

<table>
<thead>
<tr>
<th>Design Factor</th>
<th>Assessment Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security and business policy representation</td>
<td>Access roles – user groups and access rights</td>
</tr>
<tr>
<td></td>
<td>Workflow/approval rules</td>
</tr>
<tr>
<td></td>
<td>Quality of security services</td>
</tr>
<tr>
<td>Registration and renewal support</td>
<td>Physical versus electronic identity</td>
</tr>
<tr>
<td></td>
<td>Ensuring uniqueness of identity</td>
</tr>
<tr>
<td></td>
<td>Ubiquity</td>
</tr>
<tr>
<td>Reinvented administration</td>
<td>Prompt action</td>
</tr>
<tr>
<td></td>
<td>Widely distributed administration</td>
</tr>
</tbody>
</table>
The eleven design factors are amplified below. A checklist has been provided for each factor to focus design discussions. Please review this list again after reading Section 4.

**Security and Business Policy Representation**

- Access roles are required. Roles are based on authorized privilege defined by:
  - User needs – status, organizational fit (location, department, work group, rank), functional / commercial responsibility (job code, service provided)
    - User groupings – User community, organizational fit, job function, and commercial relationship (recurring user, one-time shopper, strategic partner)
  - Mapping between user and resource groups
  - 1-offs for special, limited access – executive/contractor, special process
- Workflow rules are required:
  - Administrative roles/separation of duties – super user, security officer, information owner, data entry/administration, user, support/help desk
  - Span of control/dual approval of rule changes
  - Organizational responsibility for account/access request approvals
  - Escalation process if request not handled promptly
- Rules for quality of security include:
  - Multi-part authentication
  - Frequency for renewing credentials
  - Conditions under which accounts/rights are terminated or suspended
  - Standardization across the enterprise
  - Detection of account changes not compliant with policy

**Registration and Renewal Process**

- Linking physical and electronic identity requires:

---

<table>
<thead>
<tr>
<th>Design Factor</th>
<th>Assessment Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools promote quality</td>
<td>of security services</td>
</tr>
<tr>
<td>Dealing with change</td>
<td>Systemic monitoring tools</td>
</tr>
<tr>
<td>Personalization</td>
<td>User profile</td>
</tr>
<tr>
<td>Real-time enforcement</td>
<td>Authentication, authorization, audit services</td>
</tr>
<tr>
<td>Post-event reporting</td>
<td>Report and query services</td>
</tr>
<tr>
<td>Securing security</td>
<td>Protecting administrative and data-handling architectures</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Flexibility through open standards and APIs</td>
</tr>
<tr>
<td>Complexity</td>
<td>Scope</td>
</tr>
<tr>
<td></td>
<td>Solution configuration / specialization</td>
</tr>
</tbody>
</table>
Manual validation of user-supplied source (authoritative) data
User-supplied data verified for authenticity
Authoritative data reused based on an existing shared secret

Identity must be unique:
Target environment validates proposed identity/account has not been assigned
Target environment validates that user does not already exist when registering but does exist when renewing
Authoritative data reused based on an existing shared secret
Identity/authentication credential creation and provisioning processes integrated

Tools are capable of supporting range of users and their access requests:
Diversity of user communities and number of users
Web tools, email URL transport

Reinvented Administration
Prompt provisioning and adjustments are required:
Automated user-to-need and need-to-privilege assignment rules
Data is available to make decisions based on user type, organizational unit, job
Changes in user status and network resources update roles and rules
Requests/response conveyed immediately to/from decision maker
Requests have a deadline for action after which approval is escalated
Distributed actions monitored for timely disposition
Separation of user versus device, service, application-centric administration

Administrative control is widely distributed versus centralized
Users are empowered to make requests for accounts and specialized access
Information owners/manager approve special requests

Administration tools are capable of super-simplified support:
User/resource groups standardize/simplify rules and manual actions, and to reset passwords
On-line help for users of administrative tools
Administrative role limits view of user data and access to functions
Centralized command console/server reused regardless of administrative role
Pick lists, standardized data provided manually

Dealing With Change
Changes in users, employment/commercial relationship, and network resources detected systemically:
Business event linked to electronic rights – name, transfer, promotion
Authoritative user lists are integrated with the solution – HR, accounting (contractors/suppliers, customers, service providers)
Authoritative list of network resources exist and integrated to the same granularity as enforcement rules
Monitoring scripts detect changes

Appropriate administrator must be notified:
- Report of automated versus manual actions
- Issue security alert if change in administrator rights required

Personalization
- Data is retained in personalized user profile:
  - Commercial relationship
  - Identity and authentication credentials
  - Access rights/role and/or user group membership
  - Reuse user profile regardless of network entry point (desktop/LAN, Web, remote)
- User interface personalized based on authorized rights:
  - Display all and only links and/or desktop icons user has been authorized to use
  - Role-based control ties display to business event/relationship

User Interaction
- Specialized user tools are required:
  - Browser adequate for all services
  - Network authentication supports access to other network services and applications
  - Specialty desktop software required for remote connectivity/session protection
- User responsibilities
  - Credential is password/phrase/private key/token/biometrics
  - Credential is maintained/stored on server and transmission encrypted, or is stored on desktop and password protected, or is stored/generated at user site but external from desktop
  - Shared secret created/maintained by user for self-service and help desk support

Real-Time Enforcement
- AAA (authentication, authorization and audit) services are required:
  - Data support and enforcement control is centralized versus at each enforcement point
  - Password updates replicated to all systems on change in any system subject to application-specific password rules
  - Identity is authenticated at one point
  - Authorization services from a centralized repository for all services
  - Log all interactions centrally and externally from the application/service
  - Enforcement services and their control files integrated with solution
  - Threats detected by packet/log scanning; alerts issued
The user reauthenticated every time there is a risk the session may have been hijacked.

Granular enforcement is required:
- Same access rights for a user regardless of network entry (web, desktop, remote)
- Transaction-level access control
- Middleware service translate between web and legacy applications
- Externalized core application security

**Post-Event Reporting**
- Closed-looped security requires confirmation that policies are acting as intended:
  - Periodic reports - performance of security tools, administrative activities
- Comply with data access policy/regulations:
  - Query capability - who accessed/updated and when

**Securing Security**
- Protect the administrative infrastructure:
  - Multi-part administrator authentication
  - Limits to administrator functional access and organizational reach
  - Administrative transaction logging and review
  - Scripts and functional support for help desk, including caller authentication
  - Web kiosk rules for self-served password reset
- Protect the data-handling infrastructure:
  - Criticality of administrative data
  - Design requirements for service continuation
  - Integrity of administrative data in transit and at rest

**Interoperability**
- A flexible infrastructure is required:
  - LDAP-based
  - Open APIs to integrate non-standard solutions
- Integration/customization is likely:
  - Proprietary systems are to be included
  - An off-the-shelf product is (or is not) preferred over custom development

**Complexity**
- Scope is managed:
  - Stability of services
  - Breadth of services
  - Number of custom ODBC applications
  - Scale – user variation, granularity of control
- The configuration is very specialized:
  - Network connectivity, protocols
4. Bounding the Design

4.1 Common SSO Technical Approaches

4.1.1 Password Synchronization

A rudimentary form of SSO provisions then maintains a single identity credential, typically a password, for selected applications or services. Selected applications and services are defined to the administrative console and stored by a user in a directory within it. A template or connector of some sort shares the assigned credential with each supported application the user may access. The password synchronization design is summarized in Figure 1 and Table 4.

How “smart” the connector is that links the administrative console with supported applications varies by product. Some products systemically create accounts and ensure the uniqueness of identity (typically account number). They either create a default authentication credential that must be changed at next use; or they request the user or administrator to enter the credential to be used.

Both processes create concerns. Administrator-created credentials must be shared securely with the user. User-controlled password entry requires the administrative application to monitor password changes and push any detected change to all other applications. In the latter case, the connector must be bi-directional. Not only must it be capable of updating the application, it must be capable of updating the master credential in the administrative system for pushing to all other applications. Or, the administrative application must be aware of the internal password update cycles for all applications and demand the user update their credential based on the most restrictive cycle.

Furthermore, the process must be linked manually to the function that manages the list of authorized users, typically HR. As a rule, products in this space cannot ensure systemically that all and only authorized users have access to all and only the services they require. Their ability to quickly provision and terminate user accounts and credentials usually depends on manually provided organizational
input, then rapid response by administrators. At that, their ability is limited to the systems for which connectors exist.

These products simplify account identity and authentication provisioning, but they are not platforms for complete application security, nor do they provide SSO per se. Generally, connectors or APIs do not support provisioning user-specific rights within a target application. They work best when all users have the same rights. Moreover, the user must log into each application individually, using the same authentication credential for each application. However, a user now has a standardized password and may have a standardized electronic identity. From both the user’s and administrator’s perspective, the process is simplified—but not single.

To make the solution appear to be SSO for the user, a script could be invoked to log into each application at session start. That script would be secured by the common credential. The script would create an application session for each supported application. There are significant weaknesses in this approach. If there are a large number of users with the same job start time, the network traffic would be significant. Further, to be effective, session status would have to be monitored to ensure the connection is maintained. Lastly, a large number of open sessions not being used would represent a security risk.

### Table 4. Overview of Password Synchronization

<table>
<thead>
<tr>
<th>Good News</th>
<th>Bad News</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ Users have only one password to remember for supported applications and services</td>
<td>♦ Connector for every supported application</td>
</tr>
<tr>
<td>♦ Connectors create account and authentication credential inside each application</td>
<td>♦ Log into each application; not SSO per se</td>
</tr>
<tr>
<td>♦ Password synched to each application</td>
<td>♦ Access rights within the application must be the same for all users or manually maintained through the application</td>
</tr>
<tr>
<td>♦ Supports common password reset</td>
<td>♦ Detecting a password change within an application then sharing with all other applications can be delayed, leading to multiple passwords for a time</td>
</tr>
<tr>
<td>♦ Enables more rapid response to organizational change for supported applications</td>
<td>♦ Coordinating password cycles</td>
</tr>
<tr>
<td></td>
<td>♦ Securely sharing credential if defined by an administrator</td>
</tr>
<tr>
<td></td>
<td>♦ Manually linking to the process that defines the authoritative lists of users</td>
</tr>
</tbody>
</table>

#### 4.1.2 Centralized Directory

Another rudimentary form of SSO provisions then maintains a single identity and credential, typically a password, in a common directory that is controlled centrally. Selected applications and services are defined to the administrative framework and stored by user. The application looks to the directory during the application request process to see if the user has access to this application. Users may authenticate to an environment such as their desktop that then validates application access requests. Or, the user may enter the credential to the application that looks to the directory for credential validation. In the latter case, a password control facility is required for the user to enter a password that
is used by all applications. The centralized directory design is summarized in Figure 2 and Table 5.

Microsoft Windows uses this general structure. Microsoft Active Directory maintains the list of users and their authentication credential. Users log into their desktop or network account. The network retains user identity throughout the session. This could represent a security risk if desktop-locking software is not deployed. Once a user has been authenticated, all Microsoft Windows application requests are checked against the directory to validate that the user has an access right. Note that reliance on Microsoft Windows 2000 or Microsoft Windows NT for support precludes support for non-Microsoft Windows applications without significant additional work. Further, it precludes all users not granted desktop accounts. However, software development partners do create non-desktop, LDAP-enabled packages that can look to the central directory for their authentication credential.

![Figure 2. Centralized Directory](image)

Generally, the process of creating a user defines their desktop account or user identity and the initial password. The user maintains future updates to the authentication credential. Only one password cycle exists: that for the desktop environment.

This solution has some similarities to the password synchronization process. Linkage with the authoritative list of users typically is ensured manually, although directory synchronization between HR and a central directory can substantially ensure prompt response to organization changes. In addition, there is no capability to provide user-specific rights within the applications. If required, a parallel administrative environment is necessary. Moreover, a password control/reset facility would be required to manage the password or credential retained in the directory if non-Microsoft Windows applications are supported.

The good news is that this solution does appear to be SSO from the perspective of the user, at least in the desktop world and products built to reuse desktop credentials. A user authenticates once per session at the desktop and gets access to all the applications for which they have access by merely clicking on a desktop icon displayed at session startup.
Table 5. Overview of Centralized Directory

<table>
<thead>
<tr>
<th>Good News</th>
<th>Bad News</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ Single user account/password</td>
<td>♦ Logon to each non-Microsoft Windows application</td>
</tr>
<tr>
<td>♦ Centralized file and control</td>
<td>♦ Directory call from every non-Microsoft Windows application; not all can be modified</td>
</tr>
<tr>
<td>♦ Aligns with Microsoft desktop/LAN architecture</td>
<td>♦ Access rights within the application must be the same for all users or manually maintained through the application</td>
</tr>
<tr>
<td>♦ SSO from user’s perspective in Windows</td>
<td>♦ Sessions remain active without added software</td>
</tr>
<tr>
<td>♦ No password synchronization problems or related delays</td>
<td>♦ Linking to the process that defines the authoritative lists of users</td>
</tr>
<tr>
<td></td>
<td>♦ Script to display authorized services at startup</td>
</tr>
</tbody>
</table>

In other than the desktop world, users would still have to log into each application that would authenticate users based on a directory call. Every application to be supported would require modification. Implementation would be less complex when applications are LDAP-enabled.

4.1.3 Intercepted Requests

An intercepted request AAA proxy server provides a true SSO solution. “AAA” implies capabilities to authenticate, authorize and account or log security requests. That is, every user request to access a new application or service is validated and the action taken (allow or deny) logged. Users provide their credentials only when a new session is established. A user may be requested to enter their password or token that authenticates them to the AAA server; or, if a PKI certificate is used, a user enters their password that protects the certificate on their desktop or smart card. The proxy server authenticates the user and validates that they have been authorized previously to the service or application requested. Session continuity is monitored and sessions terminated if not used in a reasonable time. The intercepted request proxy design is summarized in Figure 3 and Table 6.

![Figure 3. Intercepted Request Proxy](image-url)
Population and control of the directory that interacts with the proxy server is of critical concern. The administrative console should support the user registration process. To reduce the total cost of administration, the administrative console design should consider tactics for widely distributed data administration and self-service without jeopardizing security controls. For instance, the user could make account and access requests. The security function would define which information owners/managers must approve what requests, and which rights will be granted automatically once the user’s physical identity and commercial relationship/role has been confirmed.

User management services should be systemically linked to authoritative data. The stability of that data should be monitored to detect organizational changes (new hires, terminations, leaves and transfers) that affect access rights, and either alert the administrator to take appropriate action or take a predetermined automatic response. Moreover, given the flexibility of the directory, the proxy can invoke role-based rules enforcing fine-grained access to individual resources (devices, services/protocol, application servers, URLs) as defined by the security function. Rules of this type are adequate if all users authorized to the resource have the same rights to that resource.

The proxy server approach affords additional security capabilities. All packets can be encrypted between the user desktop and the proxy server. Since all transactions are logged, an audit trail independent of the application or service can be produced. In addition, application-level authentication and access authorization is not required at all if access is gated through the proxy server. Rather, the process of identity and privileges management is shifted to the administrative console and the directory.

Directory centricity affords additional capabilities as well. A centralized directory containing all users with electronic access can be reused to personalize the user experience. For instance, all and only authorized URLs and desktop links can be displayed. Doing so limits perceptions of what the user is empowered to do.

Despite its significant advantages, a proxy solution has its limitations. Desktop software may be needed (such as a VPN client) with the attendant challenges of software distribution and configuration (unless a web browser suffices). Or, authentication scripts may be LDAP-enabled. The criteria used to invoke rules must be limited to the data contained in packet headers. Therefore, transaction-level and user-specific rights within an application are very difficult to enforce. Moreover, the central nature of the proxy makes it a potential bottleneck, especially if encryption is being done. A bottleneck can be overcome with a cluster of proxy servers or with distributed points of access. Lastly, the range of supported protocols varies by product.

### Table 6. Overview of Intercepted Request Proxy

<table>
<thead>
<tr>
<th>Good News</th>
<th>Bad News</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ Right validated for every new resource request</td>
<td>♦ Desktop software may be required and/or authentication processes modified</td>
</tr>
<tr>
<td>♦ User-specific authentication and access control not required within applications and services</td>
<td>♦ Proxy may become a bottleneck unless clustered or distributed</td>
</tr>
<tr>
<td>♦ Credential request at new proxy session; session continuity is monitored</td>
<td>♦ Not all protocols may be supported</td>
</tr>
<tr>
<td></td>
<td>♦ Criteria used to invoke rules limited to packet header data; no transaction-level granularity</td>
</tr>
</tbody>
</table>
### Good News

- Centralized security control yet promotes self-service and widely distributed administration
- Consolidation point to audit/track requests by user
- Versatile, role-based rules
- Directory data can promote personalization
- Easy to provide encryption between the user desktop and the proxy server, internally or externally

### Bad News

#### 4.1.4 Middleware/Translator

A middleware solution can enhance the intercepted request proxy. Specifically, it can overcome the transaction-level granularity limitation. Most businesses have legacy applications that support users with different levels of access. If the business wants to allow a consumer to query an order status, for instance, a subset of the application functionality must be available. From a security perspective, it is not prudent to grant a user with such a limited interest and functional need access to the entire application. Nor is it reasonable (from an administrative overhead point of view) to give each user an account whose access right is limited to the desired use of the system.

Herein lies the value of a middleware layer. It translates a request (be it launched internally or externally) into a series of transactions that the target system can process. In kind, the middleware layer translates the responses, consolidates the results into a web-compatible view and delivers it to the user via the web server. The presentation GUI would be personalized by user based on their identity and previously authorized roles. Identity, authentication credential, and rights (or links) to which a user has been granted access through their roles would be retained in the directory produced by the administrative console. If transactions are business critical (that is, more than queries) where once-and-only-once assurance is required, a message management facility is required to ensure delivery and response. The middleware/translator (web) design is summarized in Figure 4 and Table 7.
In effect, the (web) server acts as the proxy server in the intercepted request solution. Identity and authentication would be required at every new web session. Every request would be authorized. Privacy can be ensured with SSL. For web services, only a browser is required eliminating software distribution. The data associated with a query and/or transaction can be logged outside of the transaction system to support sophisticated user and data handling reports and tracking. Individual accounts are not required in the target systems since the right of the user has been confirmed in advance.

A middleware layer plus intercepted request solution offers a simplified and standardized mechanism to offer flexible access to legacy applications. By associating access privileges with XML document objects that a user accesses via their browser, the user interface can become a powerful tool to promote both security and ease of access. However, security rules in the existing application that are based on submitted data elements may have to be transferred to the middleware process. A customer could be presented selected query options or the capability to submit an order and track progress but not gain access to the myriad of other tasks an order management system performs. In contrast, a trading partner could have access to transactions reporting on raw material delivery but not submit an order.

Given the fine-grained control middleware affords, simplification and flexibility must be designed into the administrative process to ensure little or no additional administrative effort for very large communities. The administrative tools in the intercepted request solution should be extensible to meet the middleware/translation requirement. The registration process can be simplified greatly for large communities by assigning default roles to such groups. Any user requiring additional access must request it and be authorized by the information owner. Monitoring the stability of authoritative data remains an important design condition.

Such web environments exist today. They do not, however, provide a framework to meet the SSO challenge. Generally, the administrative support facilities are not designed for integration with other forms of SSO. Nor are they designed with an eye toward a standardized architecture that can be leveraged for many application environments. A separate middleware process is required for LAN and remote access services.
Table 7. Overview of Middleware/Translator

<table>
<thead>
<tr>
<th>Good News</th>
<th>Bad News</th>
</tr>
</thead>
<tbody>
<tr>
<td>♦ Right validated for every request</td>
<td>♦ Middleware/translator required</td>
</tr>
<tr>
<td>♦ Credential request at new web session</td>
<td>♦ XML/form/translation rules for each transaction</td>
</tr>
<tr>
<td>♦ Centralized security control yet promotes self-service</td>
<td>♦ Existing security rules in applications transferred to translator process (to some degree)</td>
</tr>
<tr>
<td>♦ Consolidation point to audit/track data manipulation and access by user</td>
<td>♦ Transaction assurance requires message management if more than queries supported</td>
</tr>
<tr>
<td>♦ Transaction granularity</td>
<td></td>
</tr>
<tr>
<td>♦ Browser access to XML or web scripts</td>
<td></td>
</tr>
<tr>
<td>♦ Personalized session through role-based access</td>
<td></td>
</tr>
<tr>
<td>♦ Confidential sessions through SSL</td>
<td></td>
</tr>
<tr>
<td>♦ No user-specific accounts in the legacy system if entering via the web server</td>
<td></td>
</tr>
</tbody>
</table>

4.2 Sample System Context

Figure 5 proposes a context diagram for a comprehensive SSO system. A comprehensive design must consider all of the potential stakeholders to be supported and the benefits to be generated. That said, a comprehensive design is some blending of the four technical approaches described in Section 4.1. The system context (Figure 5) is described here in terms of its general functionality and the interactions of critical users with the system.

The target design has four primary components or functional modules. The specific requirements for each module come from the design considerations found in Section 3.2. The four components are:

1. Rules base: Interface to generate rules in conformance with security policy and business practices that control the user provisioning process.
2. Data administration: Interface to provision and administer users, culminating in a user profile that manages enforcement authentication and authorizations.
3. Enforcement: Session-level security tools supporting desktop, Web and remote access authentication/authorization services plus application/transaction control.
Figure 5 denotes four primary user functions interacting with the system. Operationally, the same persons may perform any or all of the functions, or many persons may be engaged in the same function.

The **security officer** function is responsible for representing business interests and security policy as electronic rules for identity management and access control. In the process, roles are defined to bundle groups of access privileges together for assignment to a group of users. Since the provisioning process may be executed electronically, rules and roles must be associated with descriptive user data. When a manual provisioning process is used, the workflow rules that specify required approvals and escalation must be defined. In either case, the **security officer** must ensure that the spirit and intent of the security policy is observed, review audit reports, limit the access of administrators and oversee processes.

The **data administrator** function may be automated or performed manually. Either way, the function exists to assign a unique electronic identity to a user and resolve access privileges. When the process is manual, **users** may be empowered to request access rights for disposition by the **decision maker** specified by the **security officer**. Or, the **data administrator** (or designee such as the manager of the user) may enter data on behalf of users and launch the manual workflow approval process. Often, authoritative data is integrated into the administrative process to minimize data entry effort and to ensure that the data describing the commercial relationship retained in the user profile is current and accurate. The **data administrator** may be alerted to changes in authoritative data for a **user** that may affect security rights. Common alerts that change security access privileges include an employee or contractor hired, promoted, terminated, or on leave, or changes in customer or supplier priority. The **data administrator** may be charged with responding to information requests pertaining to who accessed what systems or who read and/or modified what data elements and when. Query responses are generated from log entries accumulated during the administrative and/or enforcement process.
The decision maker function acts as a decision authority for manual process control. The workflow process alerts the decision authority that their approval is required. Approvals (or denials) may be required for a business change that affects authoritative data from which a provisioning rule is executed automatically, or to provision a non-electronic asset, or for a manual resource access request. An approval should cause a pending request to be activated in the user profile. Once a disposition has been reached, the user is notified of the results.

The user is a primary benefactor of the SSO system. Users may be employees, contractors, service providers, partners, suppliers, and/or customers. Ultimately, users require access to specific services, applications, and devices to complete their mission. By retaining identity and privileges in a user profile, authentication and authorization services can be performed in a consistent fashion across otherwise disparate technologies. That is, the same identity and authentication credential could be reused for desktop/LAN, web, and remote access mechanisms. In addition, all access requests are logged to support volume and audit reports. The enforcement process may translate data provided by the user into native transactions that can be processed by legacy applications. If that happens, data on which the transaction and response depend could be logged as well to support a query base that may be required by regulation.

Note that data bridges the modules. In combination, data and the administrative process become the integrating forces for SSO. The rules supplemented with authoritative data generate the user profile that is accessed during enforcement to authenticate users and validate their access—in turn, creating log entries as resources are accessed then summarized in periodic reports. This data life cycle defines a closed-looped environment in which authoritative data are translated by policy, and then policy informs rules, which informs user-specific rights, which informs enforcement, which informs policy review and change.

4.3 So What’s the “So-What”

Suppose a new person gets hired to start next Monday. The new hire entry in the HR system defines a business event. The SSO system detects that event and its associated organizational unit, location, and job function. Prior to Monday and based on the rules previously defined by the security officer, the system proposes instructions to create accounts and privileges, submitting them to the appropriate decision makers for action. When approved, access is granted, the user profile entry activated, the local data administrator notified and a hard copy letter sent to the user advising them of their identity and rights.

Needs for other physical assets would be handled in a similar fashion. Suppose the user needs a phone, workstation, building access card, and a pager. Instructions to obtain these resources would be proposed for approval by the user’s supervisor. Once approved, the appropriate action is taken.

On Monday, the user arrives. The physical assets are at the work site as is the letter with security instructions and rights. The user may be required to reset the SSO password. Other than that, the user goes to work with a single identity and credential governing access to all and only the electronic assets to which the user is entitled.
5. Lessons Learned

**Most organizations will have multiple sign-on mechanisms.** The current and foreseeable reality is that larger organizations already have numerous sign-on points, including web applications, desktop/LAN services, remote access, and legacy applications. It is unrealistic and risky to reinvent quickly the enforcement infrastructure. Rather, design principles should be developed to avoid compounding the problem, to standardize the minimum number of technical implementations required by each user community and to simplify user interaction. Then, converge on the target system.

**Multiple sign-on mechanisms do not preclude dramatic administrative improvements for cost reduction and improved security.** While current security features and functionality may be difficult to replace at one time, provisioning users and their credentials/rights within the local environments is not. Tools to support the security and administrative environments described in Section 4 are available today. Administrative simplicity and the quality of security services can be improved significantly today, plus many of the user-related value drivers can be addressed at the same time. Moreover, by focusing on data and administration as the integrating forces, the system design can be “future proofed.” As technology evolves, products can be replaced without jeopardizing quality control for the security process.

**A plan is required to guide migration and implementation.** SSO is a complex problem. A target architecture should be developed on which to converge over time. Then, given current conditions, connect where you are and where you want to be with a migration plan and signposts to initiate action. The migration plan should select a high-value area as a starting point to demonstrate efficacy and value. Proven starting points include administrative simplification, Web SSO and remote access. Filter disparate requests through the target architecture and plan to maximize the value of each manageable bite.

**SSO is an administrative process challenge as much as a technical enforcement challenge.** SSO is supposed to provide an easy way for users to get access to all and only the services to which they are entitled. Implied is a process that defines what users are entitled to access. The administrative design should describe tools to maintain – logically and easily – the rules that map users and their commercial relationship to a set of rights required to meet intended business objectives. Moreover, the tools should yield a security approval workflow process that does not abrogate responsibility. Rather, it should ensure that the appropriate persons approve requests in a timely manner.

**Distributed network-based directories (such as Microsoft Active Directory) will be a big player for LAN-based services, while centralized, federated directories will support centralized services.** The strengths of the various directory products should be leveraged. While there is no technical reason that a single enterprise directory cannot support all network entry and user authentication points, network traffic would increase, and performance would suffer especially for distributed services. The reverse is true for centralized services if all data were retained in a purely distributed environment. The integrating force should be the administrative infrastructure and its data management services controlling reuse of authoritative data, meta-services, synchronization and replication, not the directory products.
Consider the pitfalls. SSO means fewer points at which authentication and authorization can be confirmed. Penetrated single sign-on provides a perpetrator a single key to open the door to all services protected with the single credential. Therefore, securing the administrative technical and administrative infrastructures (people, process, and data) becomes mission-critical.
About BT Syntegra

BT Syntegra plays a key role in BT’s Information and Communications Technology (ICT) strategy as its expert in business transformation and change management.

It employs more than 5000 business and technical professionals worldwide and in the financial year ending 30 March 2003 reported revenues of around $1 billion.

BT Syntegra helps organizations transform the way they operate by applying business knowledge and technology to make possible new and better ways of working.

Its primary activity is the provision of consultancy and systems integration services including business consultancy, complex program management and custom systems design, development and operation.

To complement its own specialization BT Syntegra, where needed, is able to call upon skill and expertise from other parts of BT.

For further information visit www.btsyntegra.com.