SAPinsider

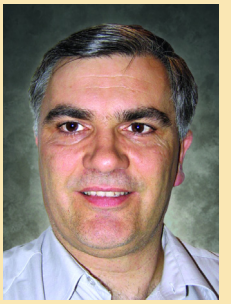
This article appeared in the Apr▪May▪Jun 2005 issue of *SAP Insider* and appears here with permission from the publisher, Wellesley Information Services (WIS), **www.WISpubs.com**.

WIS Wellesley Information Services

Are Your Single Sign-On Options Keeping Pace with Your SAP System?

Regular Feature

Security Strategies



Sarah Maidstone, SAP AG

Patrick Hildenbrand, SAP AG

The concept of single sign-on (SSO), and the benefits it can bring to an enterprise, is not new. Many SAP customers are already taking advantage of SSO to lower their administration costs and make life simpler and more secure for their users. However, as system landscapes become more diverse — expanding from maybe one or two core IT systems to a complex web of legacy, SAP, and point solutions — customers are more rapidly looking to SSO to help streamline administrative efforts and drive the most value out of the powerful technology already at work in their enterprise.

Whether you've already tinkered with SSO (perhaps through SAP logon tickets), are investigating SSO options, or are starting from scratch, the time is now to implement single sign-on. Doing so will enable you to streamline the complex systems in place at your company, relieve your end users from having to remember and continuously enter various user names and passwords, and cut down your administration time. In this article, we'll go beyond the classic SSO methods and look at a broader range of authentication and single sign-on options that are compatible with your SAP systems, focusing particularly on those available in the Java environment.

What's the Difference Between Authentication and Single Sign-On?

Authentication, or initial authentication, occurs when users first identify themselves to a system, and in turn this identification is verified. Initial authentication in SAP environments can take a number of different forms, ranging from anonymous or guest access to a Web site through the familiar user ID and password procedure, to using X.509 digital certificates.

Where **single sign-on** is in place, the user is issued credentials in one form or another following initial authentication. This allows the user to forego subsequent authentication steps when accessing further systems, offering not only convenience, but also increased security by limiting the number of times users enter sensitive information. This reduces the temptation for users to choose an easy-to-guess password. Single sign-on authenticates the user to access all the applications they have been given rights to in the SSO landscape, and eliminates future authentication prompts when the user switches applications during that particular session.

How Does SAP Support Single Sign-On?

The classic single sign-on options provided by SAP are X.509 certificates or SAP logon tickets, both of which we've explored in depth in previous "Security Strategies" columns.[1] SAP customers have also turned to external, third-party options such as biometric devices and smart cards that integrate with SAP systems via the Pluggable Authentication Service (PAS) interface.

[1] See "Single Sign-On with SAP Systems" by Juergen Schneider in the July-September 2001 issue of *SAP Insider* (www.SAPinsider.com).

We'll discuss PAS in more detail in a later section.

While SAP support for these classic SSO options is unwavering, this article familiarizes SAP customers with four additional methods of implementing single sign-on and reasons for their use:

- HTTP Header Variable Authentication
- Security Assertion Markup Language (SAML)
- Pluggable Authentication Service (PAS)
- Java Authentication and Authorization Service (JAAS)

> ✔ **Note!**
> Functionality or an application programming interface (API) for all four of these single sign-on options, to the extent that they are discussed in this article, is available with SAP NetWeaver '04.

By the end of this article, you'll have a stronger sense of the spectrum of SSO options available to SAP customers, and which method is best suited to your company's specific needs. For an overview of the four SSO methods we'll discuss here, **Figure 1** is a quick reference chart, offering a summary of the advantages and availability of each.

HTTP Header Variable Authentication

To use HTTP header variable authentication for single sign-on, you need an intermediary to act as an authentication authority between the user and the application server. In other words, initial authentication takes place outside of the SAP system, in an enterprise application management (EAM) solution, for example. Having authenticated the user, the intermediary enriches the request by adding identity information to the header, and forwards it to the application (see **Figure 2** on the next page). SAP provides a login module within the application server that interprets the information in the header.

You can access SAP Web Application Server Java, SAP Internet Transaction Server (SAP ITS), and the SAP Enterprise Portal in this way. Access is not restricted to only your SAP implementation either, as this method can also be used to access non-SAP systems that support HTTP header-based authentication.

Because of this, using header variables is the most flexible method of implementing single sign-on, allowing access to both SAP and non-SAP systems. It's simple to support, and integration is straightforward. However, it is also the least secure; another server can send a request directly to the SAP system, bypassing the intermediary and making the request appear as though it has been authenticated when in fact it has not. That is why we recommend that if you select this method, you protect the communication path between the intermediary and the application server using SSL

SSO Method	Description	Benefits	Availability
HTTP Header Variable Authentication	Initial authentication takes place outside the SAP system; an intermediary (for example, an enterprise application management solution) acts as an authentication authority between the user and the application server	Most flexible SSO method, with access to both SAP and non-SAP systems; simple to support	Java, SAP ITS
Security Assertion Markup Language (SAML)	Industry standard protocol for encoding security-related information into XML and exchanging these assertions via request and response	Most secure and future-proof, for forward-looking companies starting from scratch	Java 6.30, ABAP next major release
Pluggable Authentication Service (PAS)	Application programming interface (API) provided by SAP that allows you to plug in external authentication	Sensible option for taking advantage of the SSO functionality you already have in place	SAP ITS
Java Authentication and Authorization Service (JAAS)	J2SE standard, implemented by SAP to provide authentication in Java environments by using application-independent login modules	Robust and flexible in an enterprise context; allows for leveraging of external authentication mechanisms	Java 6.40 SP9

Figure 1 SAP-Supported Single Sign-On Options at a Glance

(Secure Sockets Layer) with mutual authentication.[2]

While the HTTP header variable authentication approach only works inside an individual network zone, it's well suited to companies that are looking to extend the reach of an EAM product they already have in place.

Security Assertion Markup Language

Security Assertion Markup Language (SAML) is an industry-standard protocol for encoding security-related information, or assertions, into XML and exchanging this information in the form of request and response. It is not an authentication mechanism in itself, but can be used to provide information about the identity and the authorizations of the user requesting access.

> **✔ Note!**
> As of SAP NetWeaver '04, SAP only supports authentication assertions.

As with HTTP header variable authentication, single sign-on using SAML involves a login module provided by SAP on the application server. But the process itself differs (see **Figure 3**). The user logs onto the authentication authority — in this case, the source Web site — stating which resource he or she wants to access ❶. The source site replies with a URL that includes an artifact[3] and a redirect URL to the resource or destination ❷. The destination sends the artifact directly to the source ❸ and receives assertions in return ❹. Based on these assertions ❺, the user is then granted access ❻.

Most of this communication can take place over an HTTP connection. However, pushing and pulling the assertion (❹ and ❺) requires an HTTPS connection, as SAP does not support digital signatures for SAML assertions. Using this kind of encrypted channel enables you to ensure that the assertions are not tampered with in any way.

Although using SAML for authentication and single sign-on does not offer the best performance times of the options discussed in this article, it does have two decisive advantages over all the others: First, it is the most secure, because an encrypted channel protects the data exchanged from tampering, and the call-back mechanism involves using a one-time token. And second, it's also

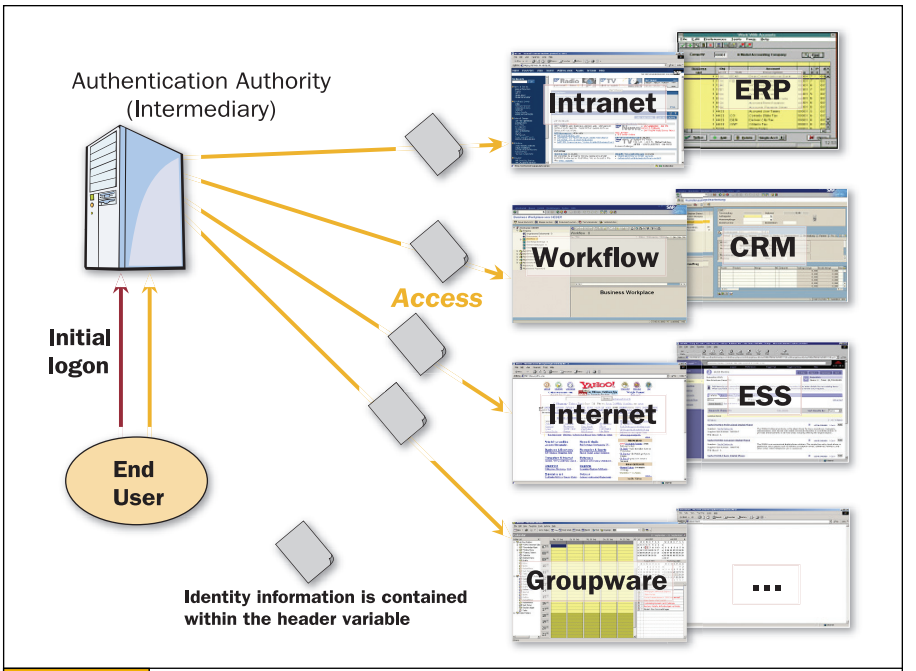


Figure 2 The SSO Process for HTTP Header Variable Authentication

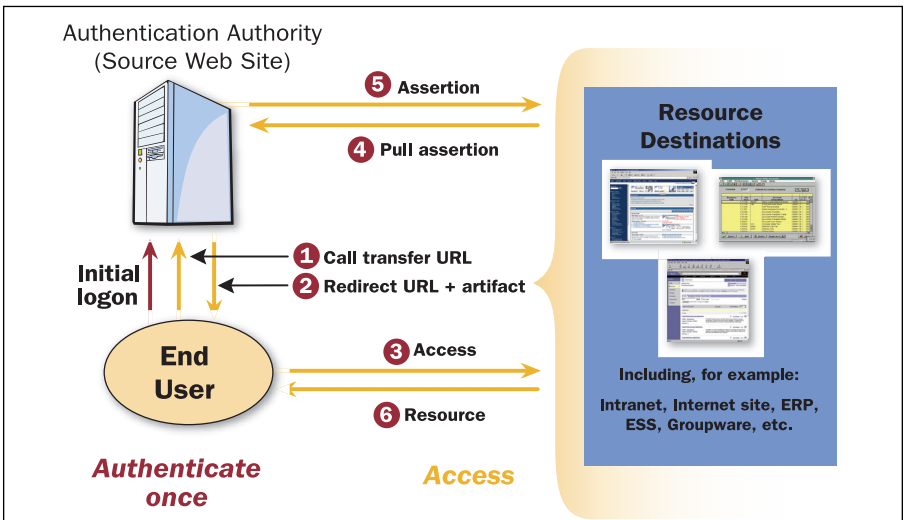


Figure 3 The SSO Process for Security Assertion Markup Language (SAML)

[2] For more information about SSL, consult "Security Features of the SAP Web Application Server" in the October-December 2001 issue of *SAP Insider* (www.SAPinsider.com), or access the "Security Basics" session from SAP TechEd '04 on the SAP Developer Network. (At www.sdn.sap.com, click on Events and navigate to the *SAP TechEd '04 Educational Sessions on SDN* link; the "Security Basics" session is under the "Security and Regulatory Compliance" track.)

[3] An SAML artifact is a small piece of data that identifies an assertion and a source site. It is carried as part of a URL query string and conveyed by a redirection to the destination site.

the most future-proof, since it provides authentication across different networks, and customers don't need to have all the pieces from the same vendor. These attributes will become increasingly important as the concept of federated identities begins to become a reality. SAML is an industry standard that is gaining prominence thanks to the work of OASIS and the Liberty Alliance,[4] and SAP's support for SAML will keep pace with improvements to the protocol.

SAML is a good choice for customers starting from scratch, or for those who already use SAML in some capacity in their enterprise. If security and standardization are key to your organization, then this option is the strongest of the ones discussed here.

Pluggable Authentication Service

For those companies relying heavily on SAP ITS, the Pluggable Authentication Service (PAS) is an application programming interface (API) provided by SAP that enables the SAP ITS to use the result of an external authentication mechanism. In other words, PAS is an SAP-provided interface that allows you to plug in external authentication. Once authenticated, a user can be issued an SAP logon ticket so that further authentication is not required.

There are a number of types of authentication you can use in conjunction with PAS, depending on your system landscape and strategic priorities. For example, if you already have an LDAP directory server, you may choose to use LDAP bind for authentication via PAS. Other certified alternatives are:

- Windows NT LAN Manager protocol (NTLM)
- Verifying user ID and password on the Windows domain controller
- SSL and X.509 digital certificates
- A mechanism on the Web server or an intermediary that sets HTTP header variables
- SAP-certified partner products, mainly EAM solutions[5]

The Pluggable Authentication Service is more mature than the previous two options we've looked at. It has been available for a number of years, and is already used by many customers to leverage external authentication mechanisms. As such, it's a sensible option for taking advantage of the functionality you already have if you're using a standalone SAP ITS, for example, and are not starting from scratch.

Note that the external ITS is only available up to and including SAP Web Application Server (SAP Web AS) release 6.20. From SAP NetWeaver 2004 on, the ITS is an integral part of the SAP Web AS, and does not support the PAS interface.

Java Authentication and Authorization Service

The Java Authentication and Authorization Service (JAAS), a J2SE standard, is implemented by SAP to provide authentication in Java environments. It allows you to use multiple authentication technologies and simultaneously not have to change either existing login services or the applications and services to be accessed.

> ✔ **Note!**
> While Java does need to be present somewhere within your environment to use the Java Authentication and Authorization Service, you can use JAAS in an environment that is not exclusively Java-based.

Technically, the different authentication types are implemented as login modules,

[4] OASIS and the Liberty Alliance are international consortiums advocating Web services and federated identity management standards, respectively. For more information about OASIS, visit www.oasis-open.org, and to learn more about the Liberty Alliance, consult www.projectliberty.org.

[5] You can find a full list of SAP-certified security partners at http://service.sap.com/securitypartners.

which are application-independent. Login modules define authentication logic, and each implements a specific type of authentication technology, such as user ID and password, or digital certificates. Login modules can be bundled into stacks, which enable you to define a sequence of authentication logic for an application. Through the JAAS interface, the SAP Web Application Server Java can act as a bridge to ABAP-based applications by allowing you to use authentication methods not supported in an ABAP environment. To achieve single sign-on, you can use login modules in conjunction with, for example, SAP logon tickets, client certificates, Kerberos, or smart cards.

Using JAAS allows you to take an industry standards-based approach to your Java environment, while at the same time benefiting from SAP's expertise in the business software arena. Based on this expertise, SAP has extended the functionality defined in the J2SE standard to make it more robust and flexible in an enterprise context. It also enables you to leverage your existing investment in external authentication mechanisms by using them for the initial logon.

Conclusion

Whichever method or combination of methods for implementing SSO you choose, what's clear is that authentication and single sign-on technology is keeping pace with functional innovation. By offering a range of options, SAP enables customers to implement the most appropriate level of security for your company. Using and extending the reach of technologies you already have in place makes for a smooth transition.

To learn more about SAP support for single sign-on, visit **www.service.sap.com/security** or **www.sdn.sap.com/sdn/developerareas/security.sdn**. ■

Sarah Maidstone has been a security product manager since 2002, and speaks regularly on security at SAP conferences. She has over six years of experience in various roles at SAP and holds an MA(Hons) degree in English Language and German.

Patrick Hildenbrand has been a product manager for security at SAP AG in Walldorf, Germany since 2003. He focuses especially on authentication, infrastructure issues, and the SAP Web AS J2EE Engine. He has great knowledge about the operation of SAP and non-SAP infrastructures as a result of his experience in the design and support of the system infrastructure at SAP Hosting, where he worked for close to three years. Patrick holds an MS degree in Industrial and Computer Science.