Perceptive Experience Single Sign-On Solutions

Technical Guide

Version: 2.x

Written by: Product Knowledge, R&D

Date: January 2016

Hyland™

Table of Contents

Overview

Nearly all organizations today have multiple applications that need to operate together. One of the challenges present in all integration scenarios is user password management. Disparate systems for user accounts and passwords cause a number of side effects. For example, side effects can include lost time due to forgotten passwords, increased maintenance due to multiple passwords per user, and a lack of seamless transitions between the applications that need to interact.

Single sign-on (SSO) technology is designed to simplify user management. There are many solution options available in the marketplace today. While they vary in scope, many provide both authentication and authorization. Authentication identifies who the logged in user is while Authorization restricts what the user can access and what actions they can take. These solutions also serve a variety of application types. Windows applications and browser applications are two main categories. Solutions that are focused on browser applications are often referred to as web-based SSO functionality.

SSO for Perceptive Experience

Perceptive Experience provides the ability to utilize SSO integration in addition to the authentication options already available in Perceptive Content Server. URL integration is frequently used to launch Perceptive Experience from a link embedded in another application so it can benefit from the use of SSO in these scenarios. Currently, only authentication can be delegated to the SSO provider. Authorization for Perceptive Experience is still managed from Perceptive Experience Management Console to determine user access privileges. Also, the user names must exist in both Perceptive Content Server and in the SSO provider. A user replication agent may be useful in synchronizing user lists. Perceptive Experience is designed to integrate with SSO providers that are installed and maintained by the customer.

Because authentication is handled by an external entity, a stateful session is introduced outside of the Perceptive Content and Perceptive Experience system. The SSO provider manages the lifecycle of this session, determines if the user is logged in, and manages timeouts for the session. When SSO mode is enabled for Perceptive Experience, there are a few resulting behavioral changes. The first is that there is a programmatic hook at load time in Perceptive Experience and Integration Server to read the name of the authenticated user from the SSO provider. Secondly, normal login into Perceptive Experience is disabled since authentication has been delegated. This includes the login screen and URL parameters.

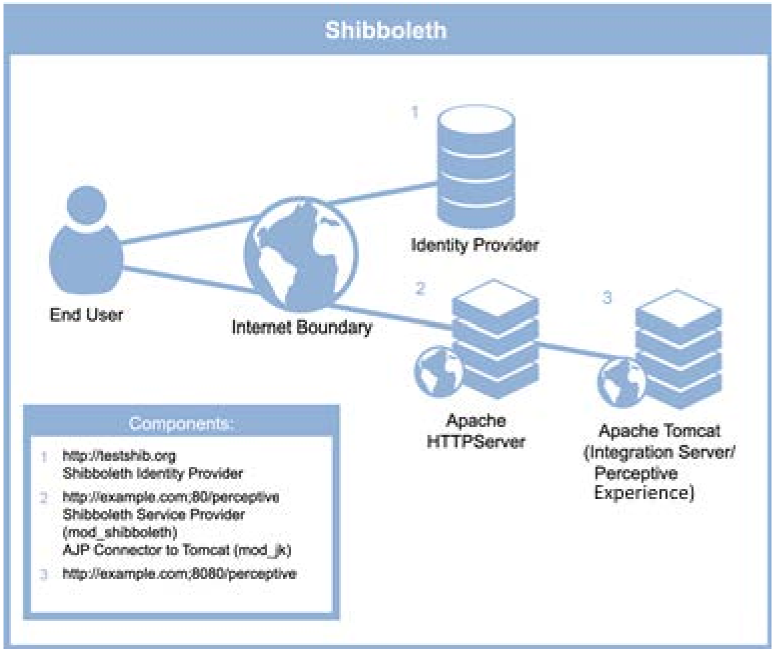
While solutions vary, the basic user story follows this general pattern.

1. User requests a URL resource (http://example.com/perceptiveexperience). An SSO provider intercepts the non-authenticated request and either prompts or redirects the user for credentials.
2. An Identity Provider (IdP) authenticates the credentials against the user store. If the user authenticates successfully, the IdP directs the user to the originally requested resource.
3. Perceptive Experience loads and reads the pre-authenticated user name from the SSO provider. This is typically communicated through an HTTP header.

Web-based SSO solutions

Shibboleth is a web-based, SSO solution. SiteMinder is another web-based SSO solution. There are many others that share a similar architecture. Web-based SSO solutions are configured to gate access to resources so that authentication is required to access the content. Perceptive Experience requires two gated resources when configuring SSO. The first is the sso directory of the /perceptive application (…/perceptive/sso). The other is the Integration Server application (…/integrationserver). To ensure proper operation of Perceptive Experience, only the sso directory should be gated, not the entire …/perceptive application.

In this example, Apache HTTPServer runs the Shibboleth service provider module that offers gated access to …/perceptiveexperience/sso and …/integrationserver. The actual content for …/perceptiveexperience and …/integrationserver is served to the user through a proxy from the Tomcat server using an AJP connector. When the user first requests http://example.com/perceptiveexperience, Shibboleth redirects the user to another site (http://testshib.org) that serves as the authority for the valid account names and passwords. There, users are prompted for their credentials. When a user authenticates successfully, the identity-authority site redirects the user back to http://example.com/perceptiveexperience with some information about the now logged-on user attached to the request.

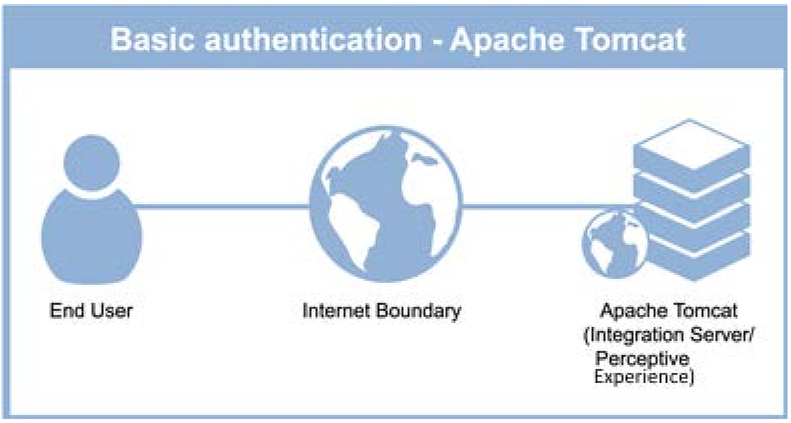

Shibboleth
1
Identity Provider
End User
Internet Boundary
2
3
Apache HTTPServer
Apache Tomcat (Integration Server/ Perceptive Experience)
Components:
1 http://testshib.org
Shibboleth Identity Provider
2 http://example.com:80/perceptive
Shibboleth Service Provider
(mod_shibboleth)
AJP Connector to Tomcat (mod_jk)
3 http://example.com:8080/perceptive

Typically, this information is communicated between these layers using an XML-based protocol called Security Assertion Markup Language (SAML). SAML relies on the same key-pair encryption technology employed by HTTPS to establish trust and to secure communications between the Identity Provider and Apache HTTPServer, even though they do not connect to each other directly. Their only contact with each other is through the information embedded along with the HTTP requests that are redirected.

When the new request has been passed to http://example.com/perceptiveexperience, the Shibboleth module again takes over and the following actions occur.

- Validates the authentication information embedded into the request.
- Approves access.
- Embeds the user name into the request as the REMOTE_USER header for informational use (configurable).
- Apache HTTP server uses the configured AJP connector to request the content from …/perceptiveexperience from the Tomcat server.
- Perceptive Experience loads on the Tomcat server and reads the name of the pre-authenticated user from the REMOTE_USER header.
- Content is sent back to the end user’s browser via reverse proxy through the AJP connector.

To the end user, it appears the Perceptive Experience content comes from the Apache HTTP server component.


Basic authentication - Apache Tomcat
End User
Internet Boundary
Apache Tomcat (Integration Server/ Perceptive Experience)

SSO integration options

The configuration steps required for Perceptive Content Server and Perceptive Experience can be found in the *Perceptive Content Server Installation and Setup Guide* and the *Perceptive Content Client Installation and Setup Guide*.

The following steps are used to determine whether an SSO provider should be integrated with Perceptive Experience.

- The SSO provider must detect any unauthorized request to Perceptive Experience's sso directory and Integration Server, and authenticate the user to the user store before granting access.
- The SSO provider must communicate information about the logged in user to Perceptive Experience and Integration Server through an HTTP header.

All access to Perceptive Experience's sso directory and Integration Server must be protected by the SSO provider. Since the authentication step occurs before Perceptive Experience loads, it implies a level of trust between Perceptive Experience and the SSO provider. Because of this, you need to ensure that only SSO provider-approved requests are granted access to either Perceptive Experience or Integration Server to prevent spoofing. Each SSO implementation has its own means of security against spoofing. Direct access to the Apache Tomcat application should be limited to connections from the HTTP server component.

The location of the sso folder that requires protection can be found at rootpath/sso or rootpath/packages/core/sso.