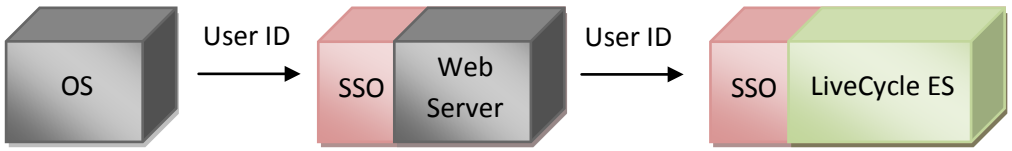
Building SSO with Mosaic and C-EMS

LiveCycle ES2 SSO

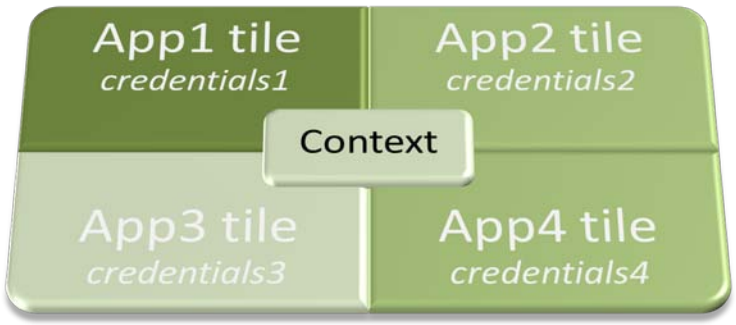
Adobe LiveCycle ES2 already provides Single Sign-On (SSO) functionality. This mostly allows users avoiding typing their login / password credentials if they have been already authenticated at the operating system level. Otherwise, they first have to identify themselves on the LiveCycle login page.

http://livedocs.adobe.com/livecycle/8.2/programLC/programmer/help/wwhelp/wwhimpl/common/html/wwhelp.htm?context=sdkHelp&file=001498.html



Composite application SSO

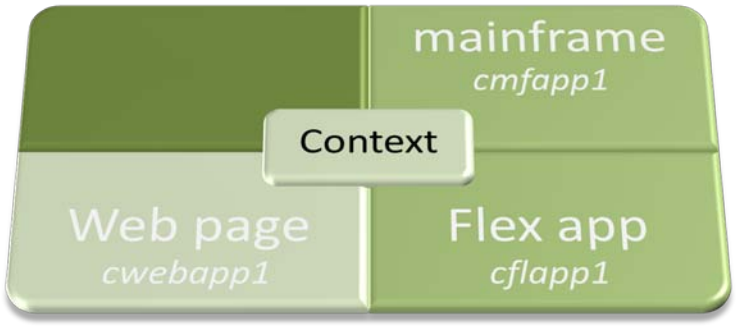
Single Sign-on (SSO) is especially important to composite applications, as they give access to multiple heterogeneous data sources and applications. When building and deploying composite applications with Mosaic, the problem is that each tile is connected to different applications, requiring different credentials:



Therefore, a well designed system requires storing and later retrieving user credentials for each of those external systems accessed.

Let's suppose a Mosaic composite application gives access to one green screen mainframe application (mfapp), one Web page (webapp), and one Flex application (flapp).

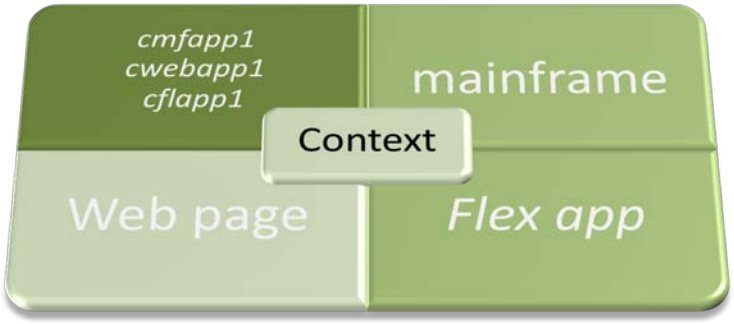
User u1 has co1 credentials on the operating system, cmfapp1 credentials on the mainframe application, cwebapp1 credentials on the Web application and cflapp1 credentials on the flex application.



Mosaic authentication and authorization system mostly relies on Spring Security (a robust security framework formerly known as Acegi). Regarding integration with enterprise SSO solutions Spring Security relies on Central Authentication Service (CAS) a well-know open source SSO framework.

Credential Vault Tile

In order to avoid user u1 supplying all his credentials for all 3 applications each time he is connecting to this Mosaic composite application, we can add a new "*Credential Vault*" tile:



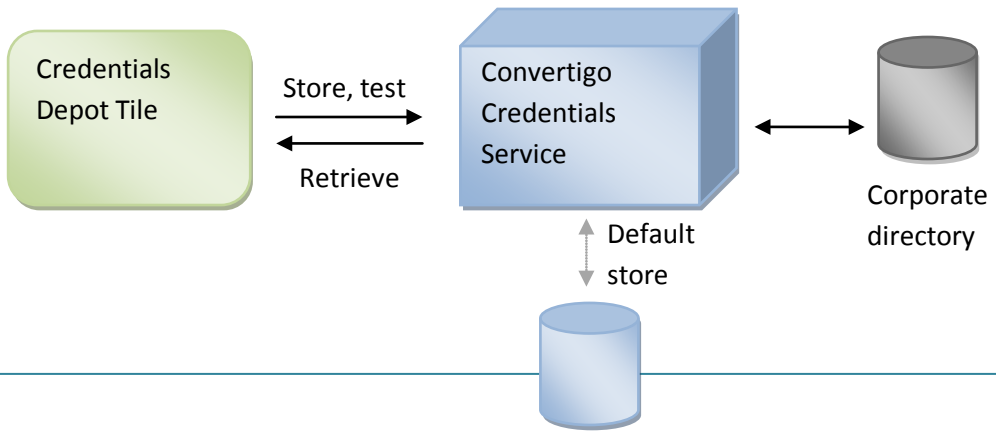
The *Credential Vault* tile will provide a way to input credentials, store, retrieve and test (if possible) them through a service. Owing to Mosaic view context, these credentials will be made available to the other tiles and will be used to connect to the different applications. At connection time, the *Credential Vault* tile could be hidden if all credentials are supplied.

NB: Mosaic provides an API in the tile SDK to retrieve a based-64 encoded SAML assertion from an authentication provider.

Credential Vault Service

The *Credential Vault* provides an interface (typically a Web service) to store, retrieve and test credentials. This service can easily be developed using Convertigo technology. A default implementation relies on the SQL module available with the C-EMS platform, but customers are able to easily reconfigure this in order to use their corporate credentials repository (LDAP server, Microsoft Active Directory…).

NB: for security reasons there is no local storage of credentials.



Conclusion

The proposed Mosaic SSO architecture described above is just an example of what can be achieved with C-EMS. Alternate implementations are possible, like, for instance, using Adobe LiveCycle DS for persistence of the *Credential Vault*.

Final implementation will finally rely on what security tools, products and policies are available at customer side.

Here are examples of more complete SSO implementations relying on the open source SSO framework CAS, JAAS and Kerberos:

http://www.ibm.com/developerworks/web/library/wa-singlesign/?Open&ca=daw-ps-news

http://www.ibm.com/developerworks/java/library/j-gss-sso/