Directa:
Single Sign On

biomni™

Table of Contents

Confidentiality and Copyright 3

1.0 Introduction 4

2.0 Architecting Single Sign On 4

2.1 Domain Authentication, Directa Web Server located within the corporate Network 5

2.2 Domain Authentication, Directa Web Server located outside corporate network 5

2.3 Other System Authentication, Directa Web server located either in or outside corporate domain 5

3.0 Authentication Methods 6

3.1 Domain Authentication (Out of the Box Solution) 6

3.1.1 Requirements 7

3.1.2 Electronic Approval and Workflow using out of the box functionality 7

3.2 Domain Authentication (Bespoke Solution) 8

3.2.1 Directa User ID other than Windows ID, site within Customer's Domain 9

3.2.2 Directa User ID other than Windows ID, site outside Customer's Domain 10

3.3 Authentication by Third Party System 11

3.3.1 Pass through from Separate Application 11

3.3.2 Third Party Authentication Server 11

4.0 Pass through Authentication Methods 12

4.1 Generated Link Passes User Directly To Directa 12

4.1.1 Example 1: Encoded Querystring 12

4.1.2 Example 2: Encrypted form post 12

4.2 Trusted Third Party Authentication 13

5.0 Electronic Approval / Workflow Considerations 14

Version	Description of change	Updated by	Date
Version 0.9	Draft Release for Biomni comments	Neil Reynolds	29 April 2009
Version 0.91	Draft release to partners	Neil Reynolds	30 April 2009
Version 1.0	Release	Neil Reynolds	6 May 2009

Confidentiality and Copyright

All enquiries concerning this document should be addressed to:

Jon Hunt

Biomni Limited

York House

23 Kingsway

London

WC2B 6UJ

1.0 Introduction

In a typical implementation of Directa, users are authenticated by the system via a user ID and password entry form located on the front page.

However, where a user has already been authenticated against their Windows domain or perhaps by logging on to a portal application, it may be desirable to automatically log the users on to Directa without the need for re-authentication or for the user to log in manually again.

Options, both out of the box and bespoke are available for this. They can be tailored depending on the requirements of the end customer, infrastructure considerations, the type of authentication is taking place and the security requirements of the customer.

This document outlines the standard “out of the box” option available in Directa and discusses a number of different scenarios that can be used as the starting point for a bespoke development and implementation of Single Sign On if this method is not suitable.

2.0 Architecting Single Sign On

Two points need to be considered together when planning to implement Single Sign on in Directa:

1. **How is the user being authenticated?**
   Is the user being authenticated by logging on to a Windows domain? Or will they be accessing and logging on manually to some other system such as a portal site?

2. **Where is the Directa web server located in relation to the users accessing it?**
   Are users accessing the Directa site hosted outside their corporate network, such as over the internet or through a secure link to a managed service provider? Or is Directa hosted within the same network as the users?

If the user is to be authenticated on a Windows domain, then either the Directa web server or some other authenticating mechanism must be connected to the corporate domain.

If the user is to be authenticated by some other method, then the Directa web server can be located in any location relative to the user, providing the user can access it.

Having considered this, one of three scenarios is therefore possible.

2.1 Domain Authentication, Directa Web Server located within the corporate Network

If Directa is located within an organisations domain, and users will only be able to access Directa when authenticated in that domain, then the only “out of the box” solution to single sign on is available – NT Authentication (see section 3.1).

Alternatively if the pre-requisites for NT authentication do not fit the organisations other requirements then a custom module would need to be developed to authenticate the user using the necessary credentials.

2.2 Domain Authentication, Directa Web Server located outside corporate network

In this case, some other authenticating system would need to be leveraged and / or developed and installed within the company’s domain network. The user would access this system, it would authenticate the user against the domain and then redirect that user through to the Directa system using some form of link (in much the same way as described in section 2.3).

2.3 Other System Authentication, Directa Web server located either in or outside corporate domain

In this case it is something such as an intranet portal, which has required a user to log on and authenticate them, that will pass the user through to Directa via some mechanism (such as a form post or encrypted link). This will require development both on the side of the authenticating application and on the Directa application side.

3.0 Authentication Methods

Several authentication methods are possible. One solution is available out of the box, subject to certain requirements being met. Other solutions have to be developed on a case by case basis and may likely require development on the customer as well as the Directa application side.

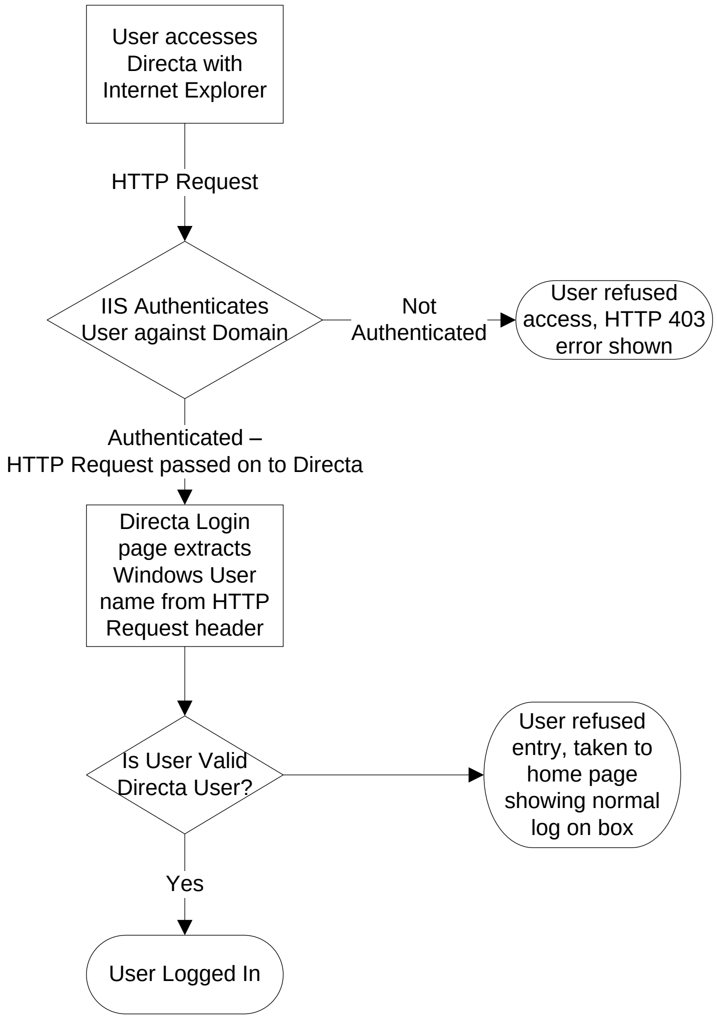
Directa has a number of built in software hooks that can be exposed to developers to facilitate the development of bespoke single sign on solutions.

3.1 Domain Authentication (Out of the Box Solution)

Directa ships with one method of Single Sign on implemented.

This method utilizes the security capability of Windows Internet Information Services (IIS) to authenticate the user before they are allowed access to the Directa web site. Once authenticated, Directa is then able to extract the user’s Windows user name from a property in the call to the web site and then uses that to log the user in to the system.

A normal log in box is still available for use of the system administrator login.


User accesses Directa with Internet Explorer
HTTP Request
IIS Authenticates User against Domain
Not Authenticated
User refused access, HTTP 403 error shown
Authenticated – HTTP Request passed on to Directa
Directa Login page extracts Windows User name from HTTP Request header
Is User Valid Directa User?
User refused entry, taken to home page showing normal log on box
Yes
User Logged In

3.1.1 Requirements

In order to use this functionality, the following conditions must be met.

1. The User ID of Directa users must exactly match either the Windows name or a concatenation of the user's domain and windows name. In other words, the Directa user name should be either:

   John.Smith
   or
   COMPANY-UK/John.Smith

2. The web site hosting Directa must be configured to enable Integrated Windows Authentication.

3. If using Firefox the web server must be added to the network.automatic-ntlm-auth.trusted-uris setting

3.1.2 Electronic Approval and Workflow using out of the box functionality

When a user receives an email requesting that they action a request as part of the workflow in that request, they may directly access the request in question by clicking on a link in the email notification. The system will authenticate as described above and forward the user to the specific request page.

If a user forwards this email to another user and they click on the link, that user will be unable to action that workflow stage as the system will recognise that the user attempting to approve the request is not the expected one.

Electronic approval considerations for bespoke single sign on implementations are discussed in section 5.0.

3.2 Domain Authentication (Bespoke Solution)

When authentication is taking place against a domain, the standard single sign on method may not be suitable. For example, customer requirements may dictate that the Directa user ID cannot be the same as the Windows name (or domain \ Windows name) - it may need to be the user's email address or company ID, for example. Or perhaps the Directa site is hosted outside the user's domain, and so the IIS server hosting Directa cannot authenticate the user against the domain.

In this case a bespoke method will have to be developed to enable the single sign on. This section describes some possible scenarios for this.

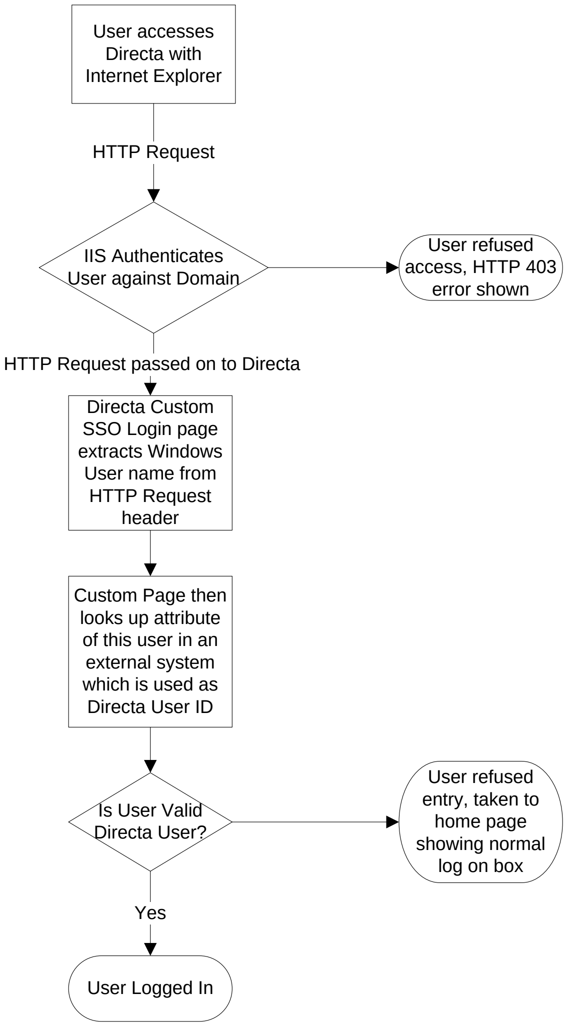
3.2.1 Directa User ID other than Windows ID, site within Customer's Domain

In this case the security features of IIS can still be used to authenticate the user when accessing the web server as with the standard functionality.

A custom web page would need to be developed and placed on the Directa web server. This custom web page would extract the Windows User name as before. However, in this case this ID could then be used to look up the Directa user name for this user in some other system.

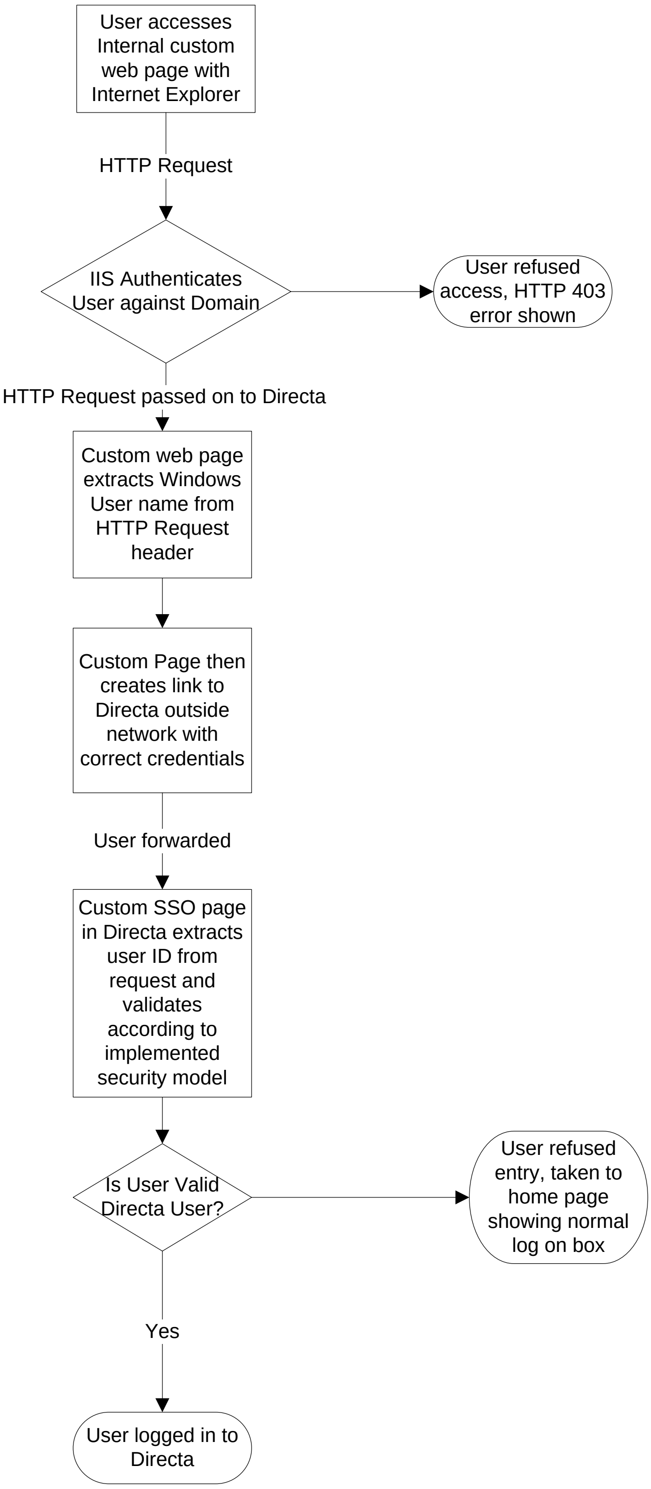
This look up could be querying an attribute of the user in their Active Directory definition or perhaps looking up information against their user record in an HR system.

Once the Directa user ID is resolved, the Single Sign module within Directa would be called by the custom page and the user logged in to Directa


User accesses Directa with Internet Explorer
HTTP Request
IIS Authenticates User against Domain
User refused access, HTTP 403 error shown
HTTP Request passed on to Directa
Directa Custom SSO Login page extracts Windows User name from HTTP Request header
Custom Page then looks up attribute of this user in an external system which is used as Directa User ID
Is User Valid Directa User?
User refused entry, taken to home page showing normal log on box
Yes
User Logged In

3.2.2 Directa User ID other than Windows ID, site outside Customer's Domain

In this case the initial authentication could not happen on the Directa web server but on some other internal system. This could still be a web page located within the domain of the user that extracts the user name and looks up the Directa ID. However, once this authentication is done the user would then need to be forwarded on to Directa using one of the methods outlined in section 3.3 Pass Through Authentication.


User accesses Internal custom web page with Internet Explorer
HTTP Request
IIS Authenticates User against Domain
User refused access, HTTP 403 error shown
HTTP Request passed on to Directa
Custom web page extracts Windows User name from HTTP Request header
Custom Page then creates link to Directa outside network with correct credentials
User forwarded
Custom SSO page in Directa extracts user ID from request and validates according to implemented security model
Is User Valid Directa User?
User refused entry, taken to home page showing normal log on box
Yes
User logged in to Directa

3.3 Authentication by Third Party System

In this case a separate system is used to authenticate the user. A user may have already logged on to a separate application, such as an intranet portal, which then presents that user with a direct link to Directa. Or perhaps the authentication is performed by a third party single sign on authentication server deployed across the organisation.

There are no restrictions on the relative location of the authenticating application and Directa – these can be on different domains, as long as the user can access both systems via his web browser and the authentication mechanism supports communication between these locations.

3.3.1 Pass through from Separate Application

Here a user has logged on to another system. That system constructs a link, containing the credentials required to log the user into Directa and these

There are a myriad of possible ways of implementing this method of Single Sign on and some common options and their relative security considerations are discussed in section 4.0.

3.3.2 Third Party Authentication Server

If the organisation has deployed a third party centralised authentication server, then it may be required to integrate this authentication method into the Directa log on process.

Here, when accessing the web server, a call will be placed to the authentication server to confirm that the user has the right to access this resource and check if the user has an active authenticated session against the server. If the user is not currently authenticated they would be re-routed to that server to be authenticated before being returned to Directa

4.0 Pass through Authentication Methods

In this model, the user is already authenticated by some other system. This system then generates a link to Directa, most likely encrypting the link using an appropriate method depending on the security requirements of the end customer.

As well as development by Biomni to extend the Single Sign on Interface to allow this method of authentication, development would be required by the customer or vendor of the customer's system initiating the Single sign on connection.

Broadly speaking, this method can be broken into two types.

4.1 Generated Link Passes User Directly To Directa

In this case the calling application creates a link to Directa. This link could be a query string or form post to a custom page in Directa which processes the information passed and validates the user against the Directa user data.

The information passed in this link and how it is constructed will depend on the security requirements of the customer. Two examples of these are outlined below, but an actual implementation could use some combination of these or indeed a separate model altogether.

4.1.1 Example 1: Encoded Querystring

In this example the calling application simply takes the Directa User ID and encodes it with an expected parameter. For example, it could base 64 encode the string "userid=john.smith" and then append the code to a URL taking the user to the single sign on entry page in Directa.
http://www.directaserver.com/sso.ashx?ID=<base 64 encoded string>

Base 64 encoding is not especially secure, however and easily decoded by a knowledgeable malicious user. This could be acceptable from a security perspective where the calling application and Directa sites both sit within a companies' network and Directa not exposed to the internet. However, the base 64 encoding algorithm is easily implemented and custom development time on both sides would be minimal in this case.

4.1.2 Example 2: Encrypted form post

Here, more care is taken over both how the user ID is obfuscated and the details submitted.

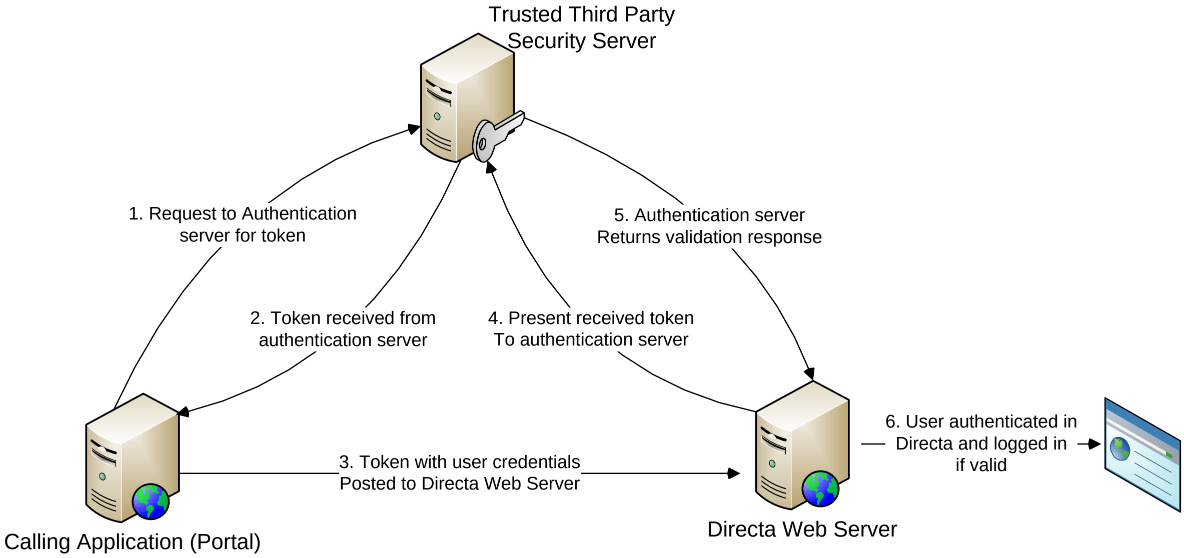
A hidden form on the calling application could store a number of parameters that would be passed to Directa. These parameters would be encrypted (rather than encoded) using an agreed algorithm; a security key would be shared between both the calling application and Directa. As well as the user name, a timestamp could be incorporated into the encrypted information. When receiving the call from the external application, Directa could not only decrypt the user name but also check that the timestamp was within an acceptable time from the point the request for single sign on was received.

4.2 Trusted Third Party Authentication

This is a secure method of Single Sign on where the authentication of the user is performed by an external authentication authority / server. This could be an already existing authentication server or a custom server created for the purposes of the particular project.

This potentially involves the deployment of components on both the calling application and Directa side. These modules would communicate with the Authentication server to authenticate the user. Communication would typically be using web services (especially if the two servers are not located on the same domain) but some other connection method could be deployed.

This process can be shown in the diagram below.


Trusted Third Party
Security Server
1. Request to Authentication server for token
2. Token received from authentication server
3. Token with user credentials Posted to Directa Web Server
4. Present received token To authentication server
5. Authentication server Returns validation response
6. User authenticated in Directa and logged in if valid
Calling Application (Portal)
Directa Web Server

1. The calling application generating the link to the single sign on connection sends a request to the authentication server passing the user name to be authenticated.
2. The authentication server returns an encrypted token containing the information required to later authenticate this user.
3. The calling application then posts the single sign on request to the web server with the token.
4. The Directa web server then sends the token to the authentication server.
5. The authentication server then verifies that the token was created by that server, is valid and has not expired then posts the “valid / invalid” response back to the Directa web server
6. The Directa web server then verifies the user is valid and logs the user in if so.

This method of pass through sign on is the most secure of those discussed. However the main disadvantage of this method is the need for the actual security server. If this does not exist in the organisation then this will be a significant cost, either to develop from scratch or use a third party solution. Development time will consequently be significantly greater than the other methods.

5.0 Electronic Approval / Workflow Considerations

In Directa, approvers and stagers of requests are notified by email when they have an action to perform. These emails, in a standard implementation, include a link which will take the user directly into the Directa request to action their workflow step.

Depending on a systems setting in the specific Directa implementation, one of two different things would happen at this point.

1. The user will be automatically logged on and taken into the request. After processing their workflow role in the request, the user remains logged in.

2. The user will be first presented with a log in box before being taken into the request. If the user tries to log in with the credentials of a different user, they will not be allowed to action the request.

In the first case, while this is more convenient to the end user, a security risk accompanies this convenience. If a user forwards their email on to another user, that user, by clicking on the link will be logged in as the original user, with all that user's access rights and powers, hence the need for option 2 if this is unacceptable for the customer.

But how should this work in a single sign implementation? As discussed in 3.1.2, standard out of the box Windows authentication will intercept users clicking on links originally sent to other users and prevent that user auctioning the request.

In a custom single sign on implementation, three separate methods would be possible, with increasing amount of development required to implement these.

1. As example 1 above, the user clicking on the link will be taken into the request as normal. The user would then be logged in as the original approver, not themselves. No additional development would be required for this option.

2. An alternative would be for the link in the email not to take the user directly to the request, but to sign the user in by the standard single sign on method, passing a parameter to take the user to the user's list of all requests awaiting action. In this way users would not be signed in as someone else, but would be signed in as themselves.

3. The single sign on method would need to be expanded to not only allow the user to be logged in via the implemented SSO method, but also be taken to the request in question providing the user signing in matches the expected user in the workflow.