maytech
GLOBAL DATA TRANSFER

FTP-Stream

Integrating Active Directory Federation Services

1 | Overview

Active Directory Federation Services (ADFS) is a standards-based service that allows the secure sharing of identity information between trusted business partners (known as a federation) across an extranet. When a user needs to access a Web application from one of its federation partners, the user's own organisation is responsible for authenticating the user and providing identity information in the form of "claims" to the partner that hosts the Web application. The hosting partner uses its trust policy to map the incoming claims to claims that are understood by its Web application, which uses the claims to make authorisation decisions.

IT organisations can use identity federations to make decisions based on identity data from other organisations, while also sharing selected information about their own users' identities. You can think of a federation as an agreement between two organisations with some common purpose, often structured so that each partner retains the management of its own internal affairs. In this context, identity is defined by a set of statements or claims about a subject. So the common purpose of an identity federation is the sharing of identity information and identity authentication responsibilities. ADFS enables this decentralised identity sharing by implementing the WS-Federation protocol along with standards such as WS-Trust and Security Assertion Markup Language (SAML).

ADFS Integration of Maytech allows the provision and authentication of users against Corporate Identification Service such as Active Directory as well as implementation of Single Sign-on Service.

2 | Major Benefits

The following is a brief list of the major benefits to using ADFS:

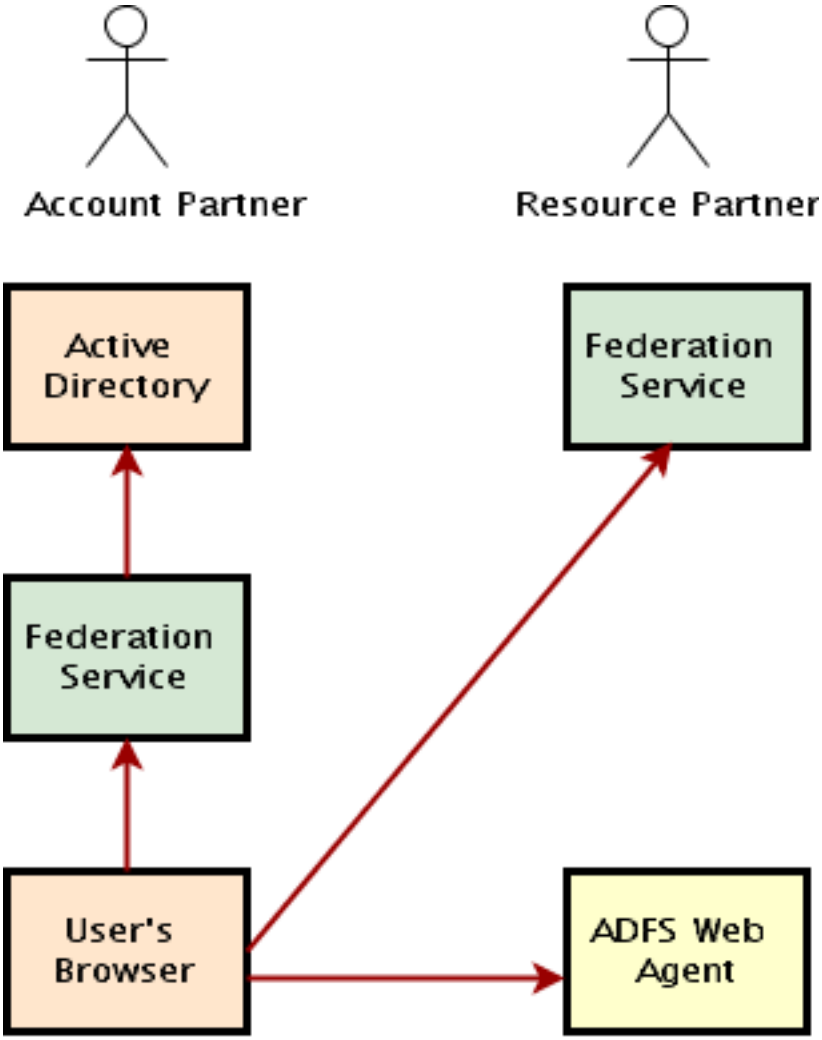
- Web single sign-on (SSO)
- ADFS provides Web SSO to federated partners outside your organisation, which enables their users to have a SSO experience when they access your organisation's Web-based applications.
- Web Services (WS)-* interoperability
- ADFS provides a federated identity management solution that interoperates with other security products that support the WS-* Web Services Architecture. ADFS follows the WS-Federation specification (for passive clients; that is, browsers), which makes it possible for environments that do not use the Windows identity model to federate with Windows environments.
- Partner user account management not required
- The federated partner's Identity Provider (IdP) sends claims that reflect its users' identity, groups, and attribute data. Therefore, your organisation no longer needs to revoke, change, or reset the credentials for the partner's users, since the credentials are managed by the partner organisation. Additionally, if a partnership needs to be terminated, it can be performed with a single trust policy change. Without ADFS, individual accounts for each partner user would need to be deactivated
- Claim mapping
- Claims are defined in terms that each partner understands and appropriately mapped in the ADFS trust policy for exchange between federation partners.
- Centralised federated partner management
- Extensible architecture
- ADFS provides an extensible architecture for claim augmentation, for example, adding or modifying claims using custom business logic during claims processing. Organisations can use this extensibility to modify ADFS to finely support their business policies.

3 | ADFS Architecture

There are various parts of ADFS. In any given federation relationship, one side supplies the users (accounts) and the other side supplies the applications (resources). When you install ADFS, you'll configure its trust policy using the ADFS administration snap-in, which shows up in your Administrative Tools folder, to indicate the list of partners with which you want to federate. Users from the account partner will be accessing ADFS-enabled applications in the resource partner.

On either side sits what is known as a federation service. Each federation service exposes a Web application, and it's expected that the user's browser will be redirected to these applications in order to establish a login. The federation service resolves the impedance mismatch between the account and resource partners, so much so that the partners don't have to be using the same identity or operating system technology. Your app doesn't need to worry much about the federation service; the ADFS team provides a Web agent that runs in the ASP.NET pipeline as an Http Module that will do the heavy lifting for you. All you need to do is configure your application to use the Web agent and supply some configuration settings so it knows where to find your company's federation service. Figure 1 shows the basic structure of ADFS and how the user's browser interacts with the various components. This shows what things would look like if ADFS were on both sides of the wire, but it's easy enough to imagine the account partner, for example, being implemented in WebSphere with an IBM directory service behind it, exposing a Java-language federation service.

3 | ADFS Architecture Image



4 | Workflow

While a deep understanding of the process behind the delivery of an ADFS authenticated user to the Web application is not necessary, an overview of the process can be helpful. The following scenario is a simplified example of what occurs behind the scenes when a user from a federated partner accesses a federation-aware application.

Consider a user from one partner (known as the "account" partner) who attempts to access an ADFS enabled Web application hosted by another partner (known as the "resource" partner). The ADFS Web Agent on the resource partner's Web server intercepts the request and checks whether the user has an appropriate ADFS cookie. If so, the user is given access to the Web application. If not, the user is redirected to the resource partner's ADFS server.

The resource partner's ADFS checks the request for a security (SAML) token from the account partner and, if the token is not found, orchestrates "home realm" discovery. Home realm discovery is the process of determining the federation server associated with the user either implicitly or by presenting the user with a Web page to make an explicit choice. Once determined, a self-identifier is added to the request so that the account partner's federation server knows who is requesting a security token and the user is redirected to the account partner's federation server.

There, the user is asked to authenticate, either implicitly if the account partner is using ADFS with integrated Windows authentication, or through some other explicit manner depending on the federation service's implementation. It is important to recognise that the user is authenticated by the user's own organisation. Once authenticated, the account partner's federation service issues a security token containing identity information (in the form of "claims") (see Appendix 1) and redirects the user back to the resource partner's ADFS server.

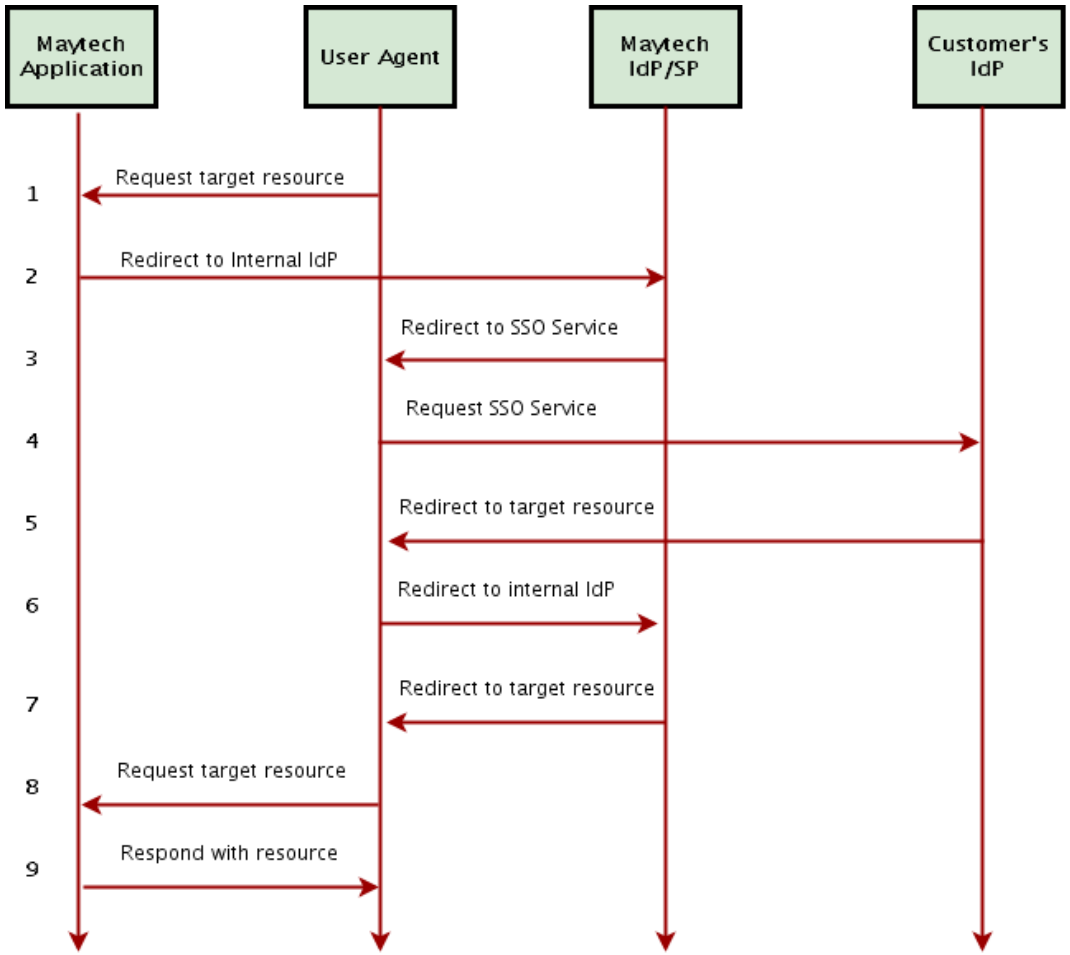
The resource partner's ADFS now finds the security token and verifies it is from one of its trusted account partners. ADFS uses its trust policy to map the account partner claims to claims that are understood by its Web application. Then ADFS issues a second SAML token that contains the resource partner claims, writes the contents to a cookie on the user's computer, and redirects the user back to the Web application.

The ADFS Web Agent discovers the cookie, parses the SAML token, and allows access to the Web application. The Web application instantiates a Single-Sign-On-Identity object, which contains the claims parsed from the SAML token, and uses these claims to make authorisation decisions.

5 | Maytech SAML Use Case

The primary SAML use case is called Web Browser Single Sign-On (SSO). A user wielding a user agent (usually a web browser) requests a web resource protected by a SAML service provider. The service provider, wishing to know the identity of the requesting user, issues an authentication request to a SAML identity provider through the user agent. The resulting protocol flow is illustrated in the following diagram.

5.1 | SAML Web Browser SSO



1. Request target resource. The principal (via an HTTP user agent) requests a target resource at the service provider:

https://acme.ftpstream.com/

2. Redirect to internal IdP. Maytech service provider determines the user's preferred identity provider:

https://acme.ftpstream.com/saml

3. Redirect to SSO Service. Maytech service provider redirects the user agent to the SSO Service at the identity provider.

4. Request SSO Service. The user agent issues a GET request to the SSO service at Customer's identity provider. The SSO service processes the AuthnRequest and performs a security check. If the user does not have a valid security context, the identity provider identifies the user with a usual login form.

https://adfs.customer.com/

5. Redirect to target resource. The assertion consumer service processes the response, creates a security context at the service provider and redirects the user agent to the target resource.

6. Redirect to internal IdP. Maytech service provider determines the user's preferred identity provider:

https://acme.ftpstream.com/saml

7. Redirect to target resource. The assertion consumer service processes the response, creates a security context at the service provider and redirects the user agent to the target resource.

8. Request target resource. The user agent requests the target resource at the service provider:

https://acme.ftpstream.com/

9. Respond with requested resource. As a security context exists, the service provider returns the resource to the user agent.

6 | ADFS SSO Integration with FTP-Stream

In order to complete the connection between identity provider and FTP-Stream service provider you should exchange the metadata between SP and IdP. You can obtain FTP-Stream's SP SAML 2.0 metadata by requesting metadata XML file from Maytech Service Desk. As soon as you receive the required metadata, you should provide us with IdP metadata XML file.
When Single Sign-on turned on in the FTP-Stream account, FTP-Stream Web service will display **SignOn with Identity Provider** button for authentication via identity provider.

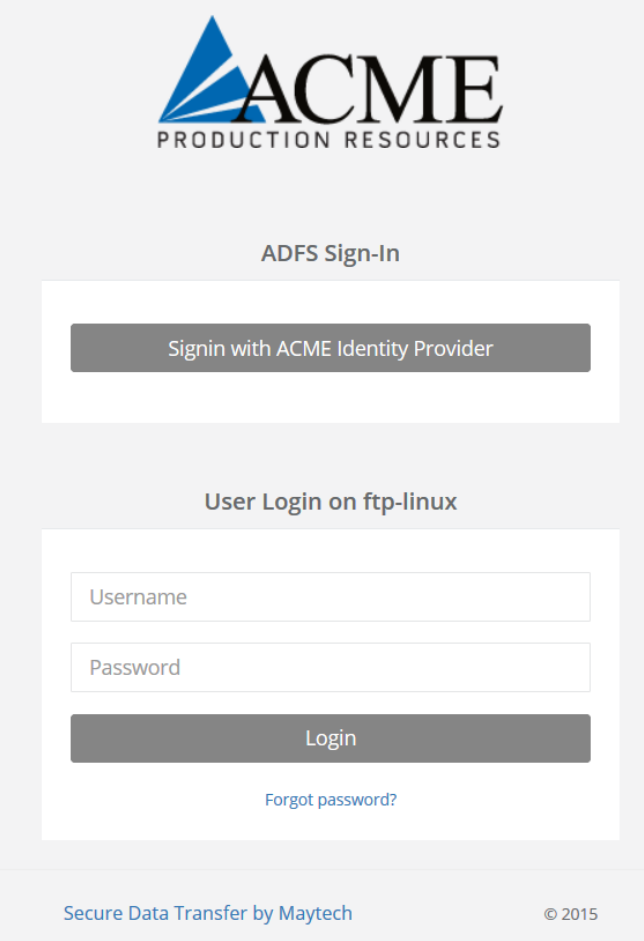


Figure 4: User login on ACME Corporation

Appendix 1: Types of claims used in Maytech ADFS Integration

The following example displays a number of fields that are received by FTP-Stream SP in order to dynamically sign in a user.

Field	Status	Description	Example
Login	Required	3-20 upper/lower ASCII characters or numbers	john
Home	Required	Up to 255 upper/lower ASCII characters or numbers.	/project11/john
Name	Optional	Text sting up to 255 characters	John Smith
Email	Optional	Valid email address	john@acme.com
Quota	Optional	Integer value representing bytes	1073741824
SSH Key List	Optional		
Group Membership List	Optional	Permission group membership.	grp_no_download, grp_no_upload, grp_no_delete, grp_no_filelist, grp_no_overwrite

Fields can be adjusted based on customer requirements.