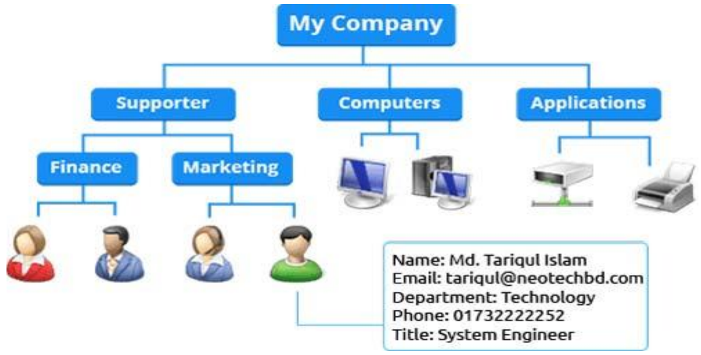
Integration of ADFS OAuth 2.0 with Web Application

Chaitanya V. V. S. K

Introduction

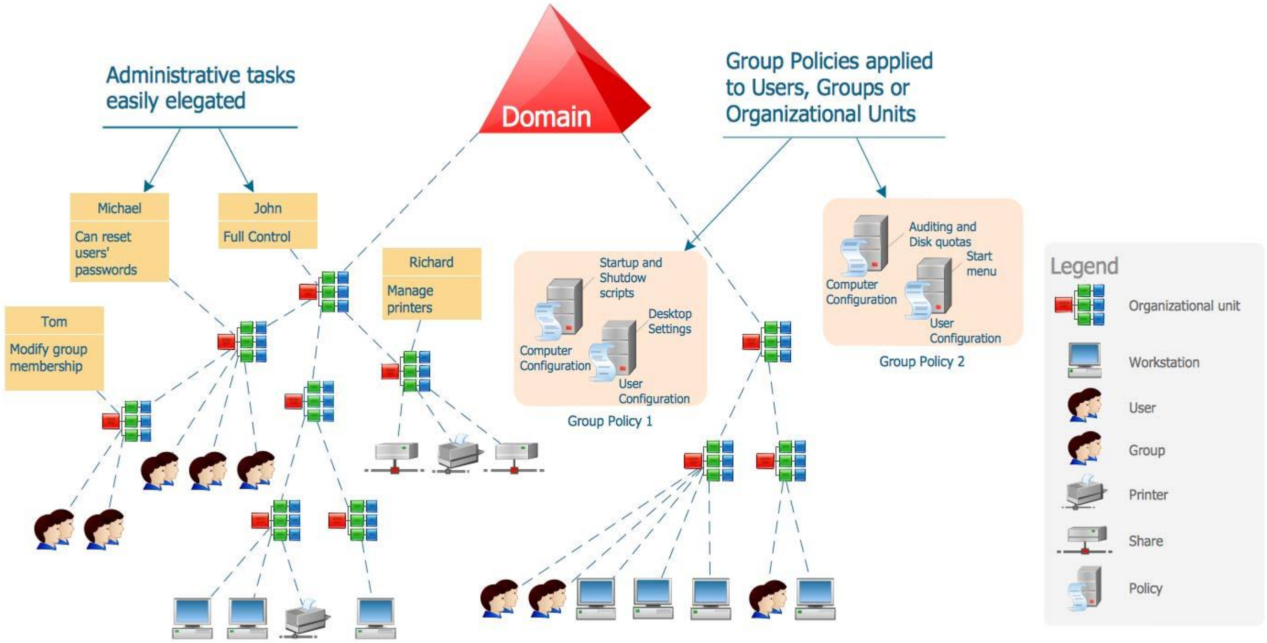
Active Directory

Active Directory (AD) is a directory service that Microsoft developed for Windows domain networks. It is included in most Windows Server operating systems as a set of processes and services. Initially, Active Directory was only in charge of centralized domain management, later it has been enhanced to provide AD FS, AD CS, LDS



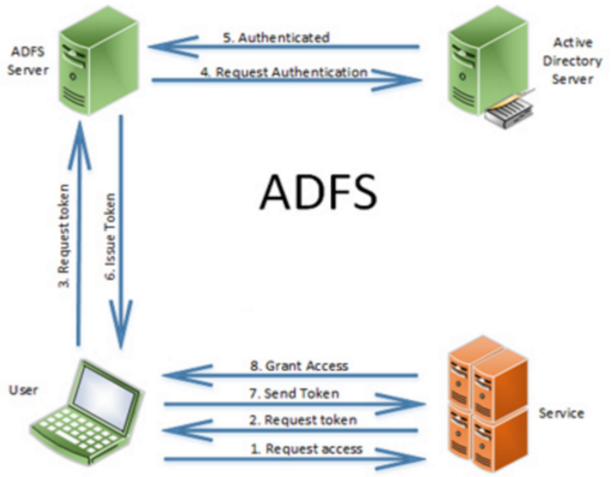
Active Directory Domain Services

Active Directory Domain Services (AD DS) is a server role in Active Directory that allows admins to manage and store information about resources from a network, as well as application data, in a distributed database.



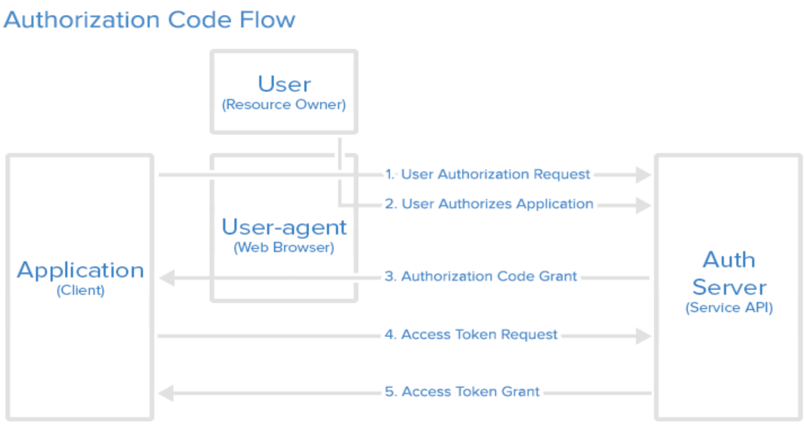
Active Directory Federation Services

Active Directory Federated Services (ADFS) is software designed by Microsoft for the Windows operating system that provides users with a single sign-in for all access points and applications throughout the organization. It follows a claim-based access that allows the user full access with a single sign-in while maintaining security and federated identity.



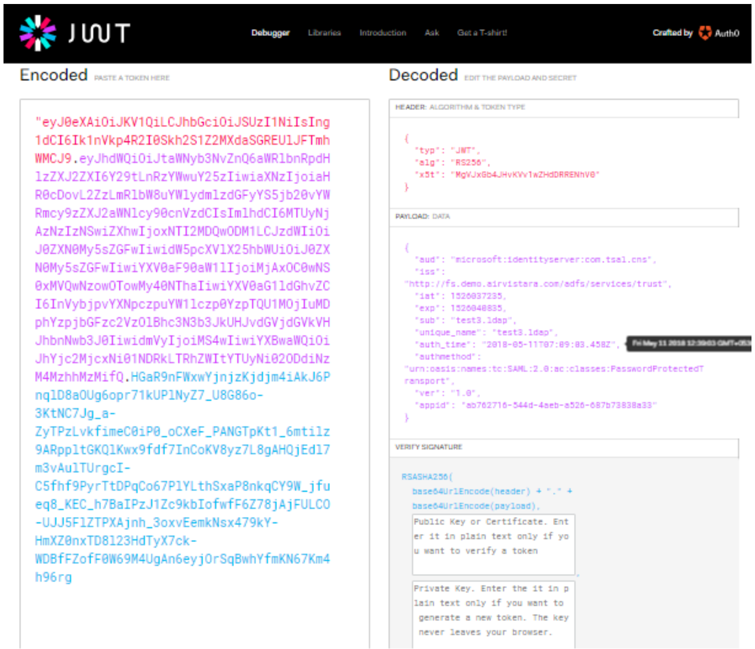
OAuth 2.0

OAuth 2 is an authorization framework that enables applications to obtain limited access to user accounts.OAuth 2.0 focuses on client developer simplicity while providing specific authorization flows such as Authorization Code, Implicit, Client Credentials, Password Credentials Flow for web, desktop and mobile apps.



JSON Web Token (JWT)

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the **HMAC** algorithm) or a public/private key pair using **RSA**.



Encoded PASTE A TOKEN HERE

```
"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng
1dCI6Ik1nVkp4R2I0Skh2S1Z2MXdaSGREUlJFTmh
WMCJ9.eyJhdWQiOiJtaWNyb3NvZnQ6aWRlbnRpdH
lzZXJ2ZXI6Y29tLnRzYWwuY25zIiwiaXNzIjoiaH
R0cDovL2ZzLmRlbW8uYWlydmlzdGFyYS5jb20vYW
Rmcy9zZXJ2aWNlcy90cnVzdCIsImlhdCI6MTUyNj
AzNzIzNSwiZXhwIjoxNTI2MDQwODM1LCJzdWIiOi
J0ZXN0My5sZGFwIiwidW5pcXVlX25hbWUiOiJ0ZX
N0My5sZGFwIiwiYXV0aF90aW1lIjoiMjAxOC0wNS
0xMVQwNzowOTowMy40NThaIiwiYXV0aG1ldGhvZC
I6InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMD
phYzpjbGFzc2VzOlBhc3N3b3JkUHJvdGVjdGVkVH
JhbnNwb3J0IiwidmVyIjoiMS4wIiwiYXBwaWQiOi
JhYjc2MjcxNi01NDRkLTRhZWItYTUyNi02ODdiNz
M4MzhhMzMifQ.HGaR9nFWxwYjnjzKjdjm4iAkJ6P
nqlD8aOUg6opr71kUPlNyZ7_U8G86o-
3KtNC7Jg_a-
ZyTPzLvkfimeC0iP0_oCXeF_PANGTpKt1_6mtilz
9ARppltGKQlKwx9fdf7InCoKV8yz7L8gAHQjEdl7
m3vAulTUrgcI-
C5fhf9PyrTtDPqCo67PlYLthSxaP8nkqCY9W_jfu
eq8_KEC_h7BaIPzJ1Zc9kbIofwfF6Z78jAjFULCO
-UJJ5FlZTPXAjnh_3oxvEemkNsx479kY-
HmXZ0nxTD8l23HdTyX7ck-
WDBfFZofF0W69M4UgAn6eyjOrSqBwhYfmKN67Km4
h96rg
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "RS256",
  "x5t": "MgVJxGb4JHvKVv1wZHdDRRENhV0"
}
```

PAYLOAD: DATA

```
{
  "aud": "microsoft:identityserver:com.tsal.cns",
  "iss":
"http://fs.demo.airvistara.com/adfs/services/trust",
  "iat": 1526037235,
  "exp": 1526040835,
  "sub": "test3.ldap",
  "unique_name": "test3.ldap",
  "auth_time": "2018-05-11T07:09:03.458Z",
  "authmethod":
"urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedT
ransport",
  "ver": "1.0",
  "appid": "ab762716-544d-4aeb-a526-687b73838a33"
}
```

Fri May 11 2018 12:39:03 GMT+053

VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  Public Key or Certificate. Enter it in plain text only if you want to verify a token
  ,
  Private Key. Enter the it in plain text only if you want to generate a new token. The key never leaves your browser.
```

ADFS Prerequisites

1. Windows Server 2012 R2 / Windows Server 2016

Hardware Requirement	Minimum	Recommended
CPU	1.4 GHz, 64 bit	Quad Core , 2 GHz
RAM	1 GB	4 GB
Free Space	32 GB	100 GB
Software Requirement	IIS, .Net Framework 4.5	

Complete Requirement List

2. Installing Active Directory Services.

   References :

   https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/deploy/install-a-new-windows-server-2012-active-directory-forest--level-200-

   https://support.rackspace.com/how-to/installing-active-directory-on-windows-server-2012/

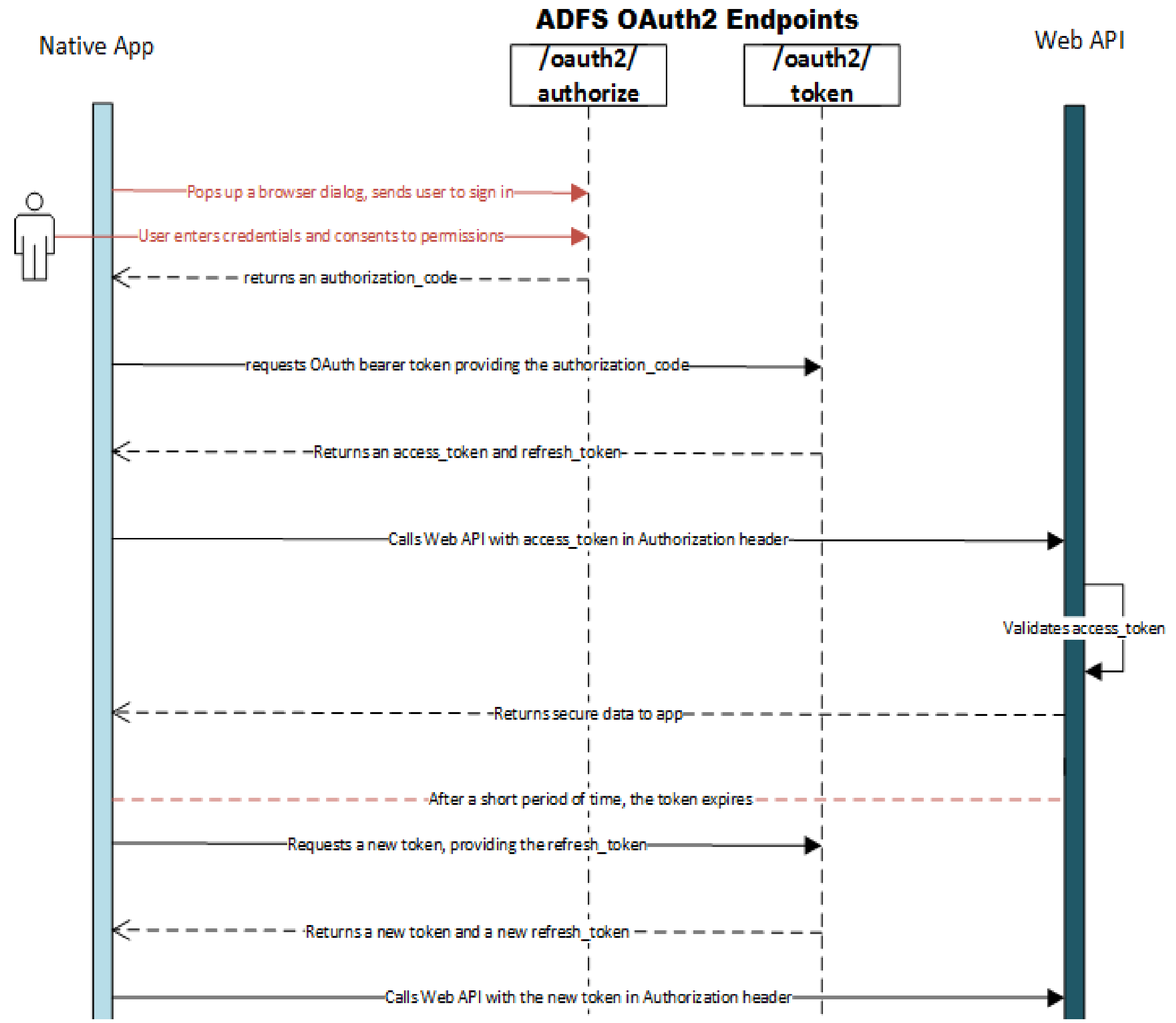
   https://social.technet.microsoft.com/wiki/contents/articles/12370.windows-server-2012-set-up-your-first-domain-controller-step-by-step.aspx

   https://blogs.technet.microsoft.com/canitpro/2017/02/22/step-by-step-setting-up-active-directory-in-w

indows-server-2016/
https://technet.microsoft.com/en-us/library/cc302671.aspx
https://www.youtube.com/watch?v=y3sPX6T9W28&list=PL1l78n6W8zyon7NlRbmvhiTcWQL-QBkmK&index=6

3. Installing Active Directory Federation Services
   References :
   https://msdn.microsoft.com/en-us/library/azure/dn528857.aspx
   https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/windows-server-2012-r2-ad-fs-deployment-guide
   https://www.youtube.com/watch?v=tAQ2n-bJ6Vs&list=PL1l78n6W8zyon7NlRbmvhiTcWQL-QBkmK&index=7

Architecture of ADFS OAuth 2.0(Authorization Grant Flow)


ADFS OAuth2 Endpoints
Native App
/oauth2/ authorize
/oauth2/ token
Web API
Pops up a browser dialog, sends user to sign in
User enters credentials and consents to permissions
returns an authorization_code
requests OAuth bearer token providing the authorization_code
Returns an access_token and refresh_token
Calls Web API with access_token in Authorization header
Validates access_token
Returns secure data to app
After a short period of time, the token expires
Requests a new token, providing the refresh_token
Returns a new token and a new refresh_token
Calls Web API with the new token in Authorization header

- https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-protocols-oauth-code
- https://blogs.technet.microsoft.com/maheshu/2015/04/28/oauth-2-0-support-in-adfs-on-windows-server-2012-r2/

ADFS Configuration

Server Manager in Windows server provides GUI for complete administration , development and configuration management of ADFS (ease for business users).

Powershell in Windows Server also supports CLI for ADFS configuration management (ease for developers).

Adding ADFS Client

An ADFS Client with following parameters needs to be added to the ADFS

1. Name *
   A Client Name which specifies the client(application name)
2. ClientId *
   A unique ID to be distinguished from other ADFS clients
3. RedirectUri *
   Used by OAuth 2.0 for redirection when the client requests authorization to access a resource secured by AD FS

Powershell

```
Add-AdfsClient -Name "Test1 ADFS OAuth2.0 Application" -ClientId
"xy9597367-880z-8gif-a967-7356880aisa" -RedirectUri
"http://mosaicuatelb-360193052.ap-southeast-1.elb.amazonaws.com/cns" -Description "OAuth 2.0 ADFS
test client application1"
```

References :

https://docs.microsoft.com/en-us/powershell/module/adfs/add-adfsclient

https://docs.microsoft.com/en-us/powershell/module/adfs/get-adfsclient

https://docs.microsoft.com/en-us/powershell/module/adfs/set-adfsclient

https://docs.microsoft.com/en-us/powershell/module/adfs/enable-adfsclient

https://docs.microsoft.com/en-us/powershell/module/adfs/disable-adfsclient

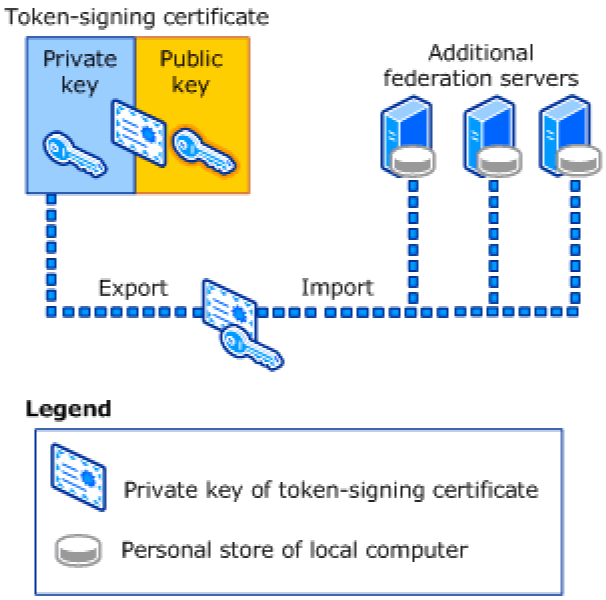
https://docs.microsoft.com/en-us/powershell/module/adfs/remove-adfsclient

Adding Relying Trust Party

Adding Relying Trust Party needs the following

1. Resource (Relying Trust Party Identifier)
   A unique Identifier (basically application url) needs to be chosen for creating the trust between ADFS and the application
2. Token Signing Certificate
   This is a standard X509 certificate that is used for securely signing all tokens that the federation server

issues. A token-signing certificate can be obtained by requesting one from an enterprise CA or a public CA or by creating a self-signed certificate. The private/public key pairing that is used with token-signing certificates is the most important validation mechanism of any federated partnership.


Token-signing certificate
Private key
Public key
Additional federation servers
Export
Import
Legend
Private key of token-signing certificate
Personal store of local computer

3. Token Encryption / Decryption Certificate

   This is a standard X509 certificate that is used to decrypt/encrypt any incoming tokens. It is also published in federation metadata. This can also be generated in the same way like Token-signing certificate.
4. Setting Claim Rules for resource

   Required Claims to be configured as LDAP parameters / by custom claim rule which will be sent to the application for authorising the user accordingly.
5. Tokens Issuance properties

   Issuance of id token, access token, refresh token and tokens life time can be configured in the relying trust party using powershell.

Powershell

```
Add-ADFSRelyingPartyTrust -Name  "Fabrikam" -MetadataURL
"https://fabrikam.com/federationmetadata/2007-06/federationmetadata.xml"
```

References :

https://docs.microsoft.com/en-us/powershell/module/adfs/add-adfsrelyingpartytrust

https://docs.microsoft.com/en-us/powershell/module/adfs/set-adfsrelyingpartytrust

https://docs.microsoft.com/en-us/powershell/module/adfs/remove-adfsrelyingpartytrust

https://docs.microsoft.com/en-us/powershell/module/adfs/get-adfsrelyingpartytrust

Web Application Configuration

For Web Application which uses RestFul Services / API , both client and server side needs to be configured with ADFS for successful token validation and authorization of the user.

Client Side Configuration

1. The login is directly redirected to the ADFS by using OAuth2 (/authorize) by passing the above configured clientId , resource ,redirect uri and grant type as authorization_code.
2. After the successful authentication , the adfs will redirect to the specified redirect uri with the authorization_code (or) error which needs to be handled by the client.
3. Using Client side javascript , the code needs to be captured and again needs to be sent to ADFS OAuth2 (/token) by passing clientId , resource , redirect uri and code.
4. An access token (JWT) will be sent to the client with the encoded claims of the user against the resource. Client needs to cache this access token in cookies with the provided expiration time.
5. There after client needs to send the access token in Authorization Header to all ajax calls to the API
6. If Client receives an Unauthorized response , again it should redirect to /authorize for user login.

Server Side / API Configuration

1. A Servlet Filter needs to be configured before sending the API request to controller
2. Filter will validate the token which is present in the authorization header of the request.
3. If the token is successfully validated, the request needs to be passed to controller else the filter will return Unauthorized Response detailing the reason.

Validation of the JWT

1. JWT is generated at the ADFS side using the configured certificates private and public keys by using some security algorithms such as RS256.
2. JWT contains three parts Header, Payload and Signature.
3. Any JWT dependency can be used for verifying the Signature and validating the token (Java JWT maven repo : com.auth0 / java-jwt / 3.3.0 , io.jsonwebtoken / jjwt / 0.9.0)
4. The public and private keys that are configured for the token signing certificate are to be made available at the API.

References :

https://jwt.io/ , https://github.com/auth0/java-jwt , https://github.com/jwtk/jjwt