MIRACLE
SOFTWARE SYSTEMS

Implementation of Open Authorization(OAuth) in DataPower

Keerthana Dhagam

Nithendra Santosh Nandyala

(EDGE APIM Team)

Contents

Introduction

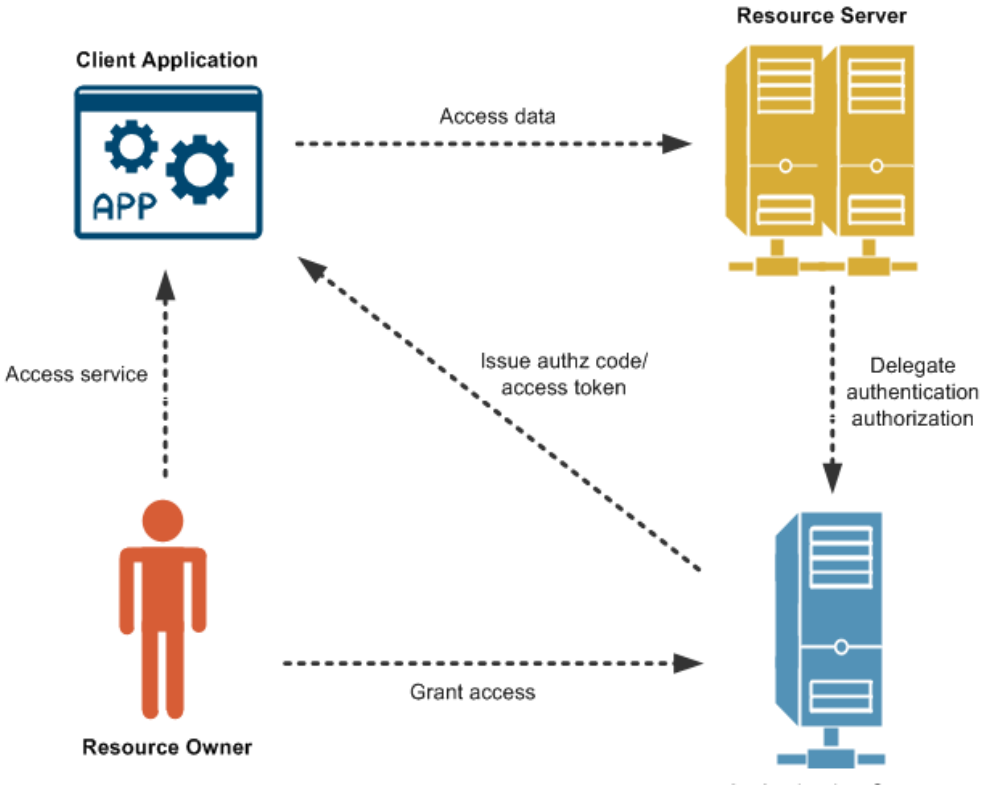
OAuth is an authorization protocol or a set of rules that allows a third-party application to access a user's data without the user needing to share their login credentials. The protocol defines a process that allows limited access to resources hosted by web-based services accessed over HTTP. For example, facebook provides APIs that extend their applications by providing content sharing capabilities.

Protected resources are accessed by providing credentials that can be authenticated and used to authorize access to the resource. If a resource owner wishes to grant access to a third-party, credentials must be provided, authenticated, and access authorized by the authorization service that protects the resources. In general, passwords are vulnerable; a "man in the middle attack" or a "shoulder attack" could obtain user's credentials and gain access to the resources. Also, a resource owner has to share his password with the entity that needs access. If a password is changed, the entity no longer has access to the resource. In such cases, authentication and authorization are all under the control of the resource owner, thereby limiting the sharing of authentication information. OAuth eliminates the need for sharing the user's credential with a third party entity.

The following are the different terms and components we use in OAuth:

- **Resource owner**: An entity that is used to grant access to a protected resource. This entity is generally a person interacting via a web browser.
- **Resource server**: A resource server has the resources and it is capable of accepting and responding to resource requests and resolving access tokens.
- **Authorization server**: A process that is capable of authenticating a resource owner and issuing access tokens. It takes the request from the client and issues tokens accordingly.

- **Client**: An entity that makes a request for a resource upon approval of the resource owner. After the client receives the token from the authorization server, it will request the resource server for the needed resources. In OAuth terminology, the client is the client of the resource server.



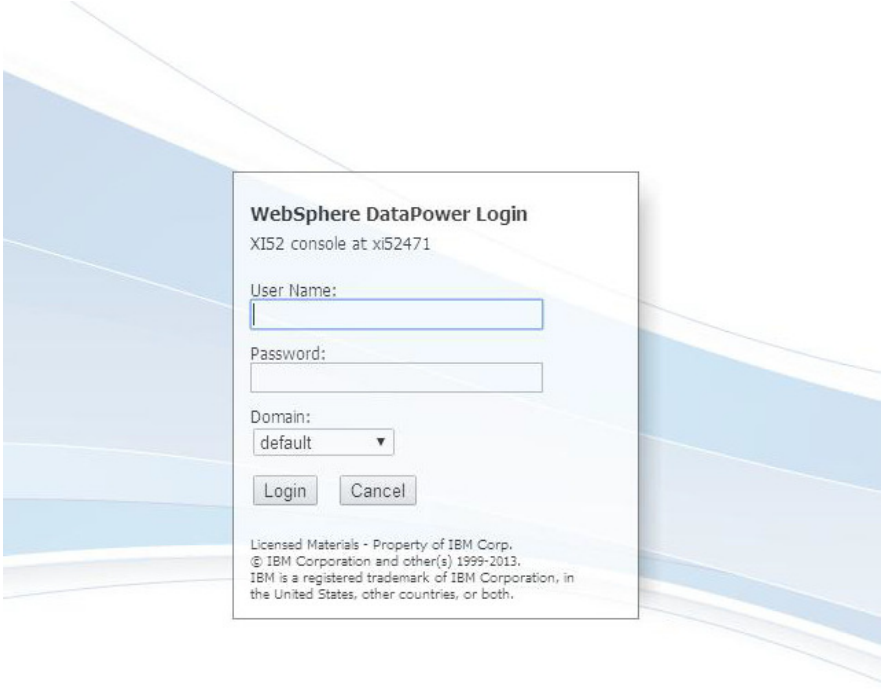
The basics steps involved in the whole process are:

1. Create an OAuth client profile.
2. Create an OAuth client group object.
3. Create AAA policies for authentication and authorization.
4. Create an authorization server using the Web Token Service.
5. Create a resource server enforcement point using the Multi-Protocol Gateway.

1. Create an OAuth client profile

A confidential client is an application that is capable of keeping a client password confidential to the world. This client password is assigned to the client app by the authorization server. This password is used to identify the client to the authorization server, to avoid fraud.

- Login into Web GUI by entering username, password and domain.



- Create OAuth client profile by navigating to Objects > Crypto Configuration > OAuth Client Profile.
- Add a new OAuth client profile and name it "account-application". This becomes the client identifier.
- The Supported Authorization Grant Type is "Client Credentials." This grant type is the focus of this article.

- The Client Secret is "passw0rd". This is used to authenticate the client with the authorization server. You need to uncheck the Generate Client Secret checkbox to specify the secret. The checkbox disappears once you populate the Client Secret field. To get the checkbox back, remove the content from the Client Secret field.
- The Scope is "updateCustomerInfo". The scope value specified in the access request is checked against values defined here.
- The Access Token Lifetime defines the lifetime in seconds of the access token.
- The Shared Secret protects tokens used by the OAuth protocol. Click the "+" button next to Shared Secret to create a shared secret object. This object references the sharedSecretKey.txt file uploaded during the preparation section. It is used to encrypt and decrypt the access token generated by DataPower.
- Click **Apply** for the OAuth Client Profile.

Main | Advanced

OAuth Client Profile: account-application [up]

Apply | Cancel | Delete | Undo

General

Administrative state: (•) enabled ( ) disabled

Comments:

Customized OAuth: [ ]

OAuth Role:
[x] Authorization and Token Endpoints
[x] Enforcement Point for Resource Server
*

Supported Authorization Grant Type:
[ ] Authorization Code
[ ] Implicit Grant
[ ] Resource Owner Password Credential
[x] Client Credentials
*

Authentication Method: Secret ▾ *

Client Secret: paasw0rd *

Customized Scope Check: [ ]

Scope: updateCustomerInfo *

Shared Secret: sslSecretkey ▾ + ... *

Enforcement Point for Resource Server

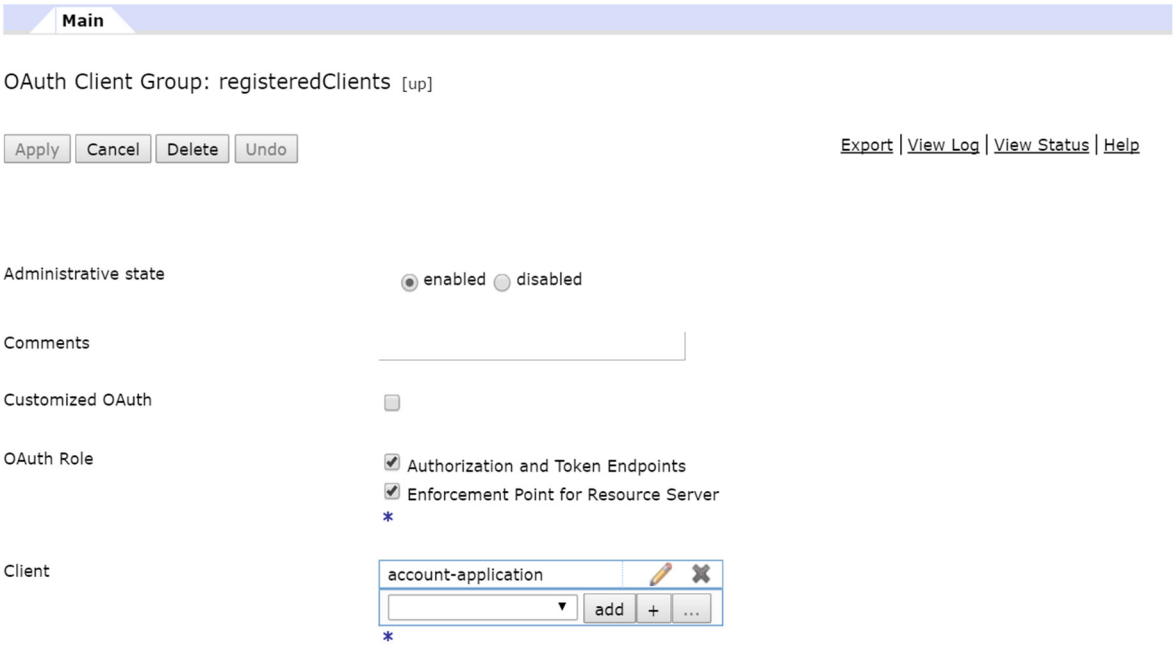
Verify Client Credential: [ ]

Use Validation URL: [ ]

2.Create the OAuth client group

An OAuth client group contains the profiles of the OAuth clients that the DataPower appliance accepts requests from. Hence we create an OAuth Client Group.

- Navigate to Objects > Crypto Configuration > OAuth Client Group. An OAuth Client Group manages the registration of OAuth clients for a single OAuth authorization server endpoint.
- Add a new client group and then enter details for the new group.
- The properties of the Client group are as follows:
    - Name: registeredClients
    - Client: Select the account-application and click the Add button.



- Click Apply.

3.Creating a AAA Policy for authentication

By having a AAA policy, you define the authentication, authorization, and auditing stages on a DataPower device. The AAA policy is invoked by using a AAA action of a processing rule for dedicated service objects such as XML firewalls, multiprotocol gateways, or Web Service Proxies.

- Navigating to Objects > XML Processing > AAA Policy.
- Click Add to create an AAA policy.
- Set Name to clientcred-aaa.
- Select the Identity Extraction tab.

Configure AAA Policy

Main | Identity extraction | Authentication | Credential mapping | Resource extraction | Resource mapping | Authorizatio

AAA Policy: clientcred-aaa [up]

Apply | Cancel | Delete | Undo

Export | View Log | View Status | Help
Flush cache

Administrative state: (•) enabled ( ) disabled

Comments:

Authorized counter: (none) + ...

Rejected counter: (none) Reject Counter Tool

SAML signature validation credentials: (none) + ...

SAML message signing key: (none) + ...

Methods

- [ ] HTTP Authentication header
- [ ] Password-carrying UsernameToken element from WS-Security header
- [ ] Derived-key UsernameToken element from WS-Security header
- [ ] BinarySecurityToken element from WS-Security header
- [ ] WS-SecureConversation identifier
- [ ] WS-Trust Base or Supporting token
- [ ] Kerberos AP-REQ from WS-Security header
- [ ] Kerberos AP-REQ from SPNEGO token
- [ ] Subject DN of SSL certificate from connection peer
- [ ] Name from SAML Attribute assertion
- [ ] Name from SAML Authentication assertion
- [ ] SAML Artifact
- [ ] Client IP address
- [ ] Subject DN from certificate in message signature
- [ ] Token extracted from message
- [ ] Token extracted as cookie value
- [ ] LTPA token
- [ ] Processing metadata
- [ ] Custom style sheet
- [ ] HTML forms-based authentication
- [x] OAuth

*

Registered OAuth clients

registeredClients ▼ + ... *

Configure AAA Policy

Main | Identity extraction | **Authentication** | Credential mapping | Resource extraction | Resource mapping | Authorizat

AAA Policy: clientcred-aaa [up]

Apply | Cancel | Delete | Undo

Export | View Log | View Status | Help
Flush cache

Method: Pass identity token to authorization phase *

Cache authentication results: Absolute *

Cache lifetime: 3 Seconds

Configure AAA Policy

Main | Identity extraction | Authentication | Credential mapping | **Resource extraction** | Resource mapping | Authorize

AAA Policy: clientcred-aaa [up]

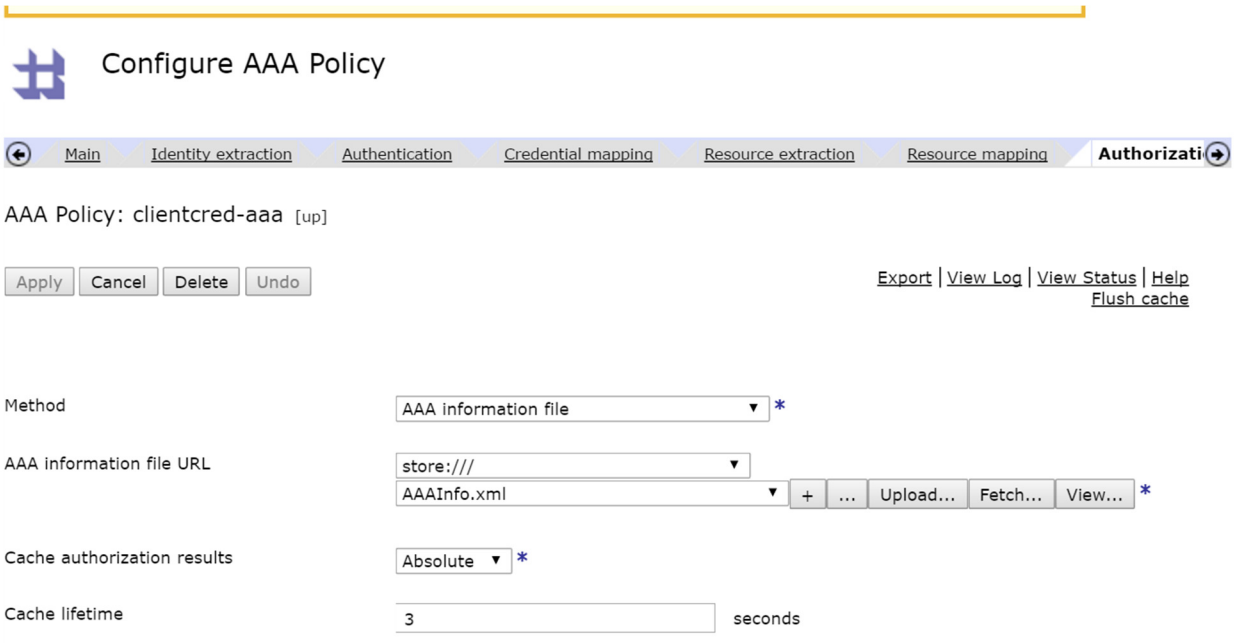
Apply | Cancel | Delete | Undo

Export | View Log | View Status | Help
Flush cache

Resource information:
- [ ] URL sent to back end
- [x] URL sent by client
- [ ] URI of top level element in message
- [ ] Local name of request element
- [ ] HTTP operation (GET or POST)
- [ ] XPath expression
- [x] Processing metadata

*

Processing metadata items: oauth-scope-metadata + ...

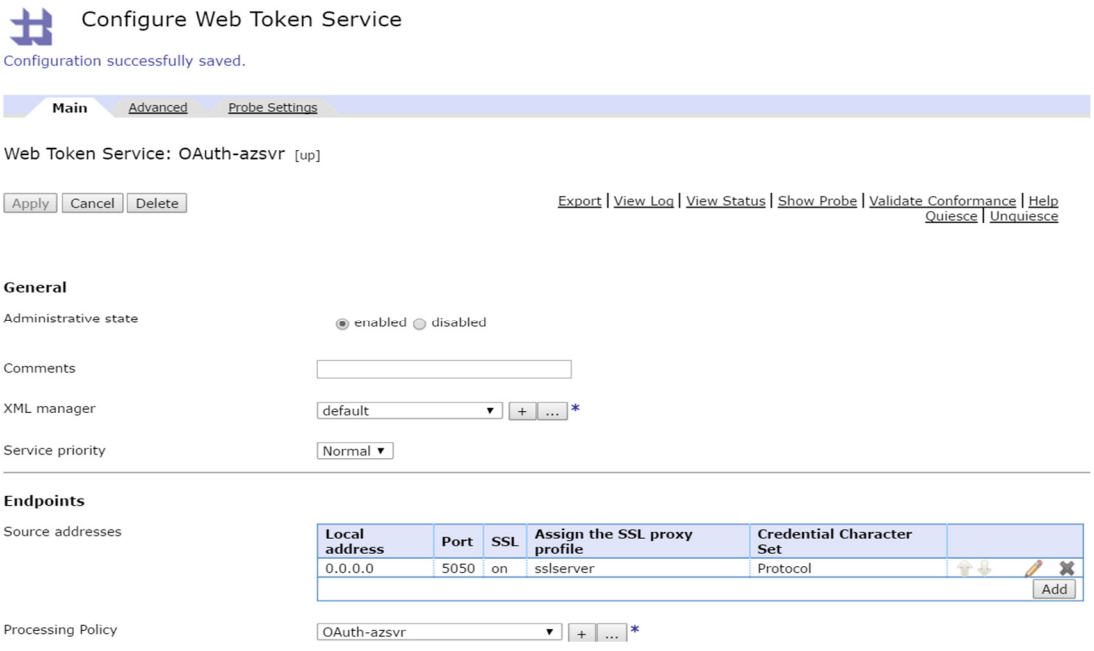


- Select OAuth as the method and choose the registeredClients OAuth Client Group.
- Select the Authentication tab. Select Pass Identity Token to the Authorize Step.
- Select the Resource extraction tab. Select Processing Metadata and choose oauth-scope-metadata for processing metadata items.
- Select the Authorization tab. Select AAA information file as the method and specify the file as local:///AAAInfo.xml.
- Click Apply.

4.Creating a Web Token Service (Authorization Server)

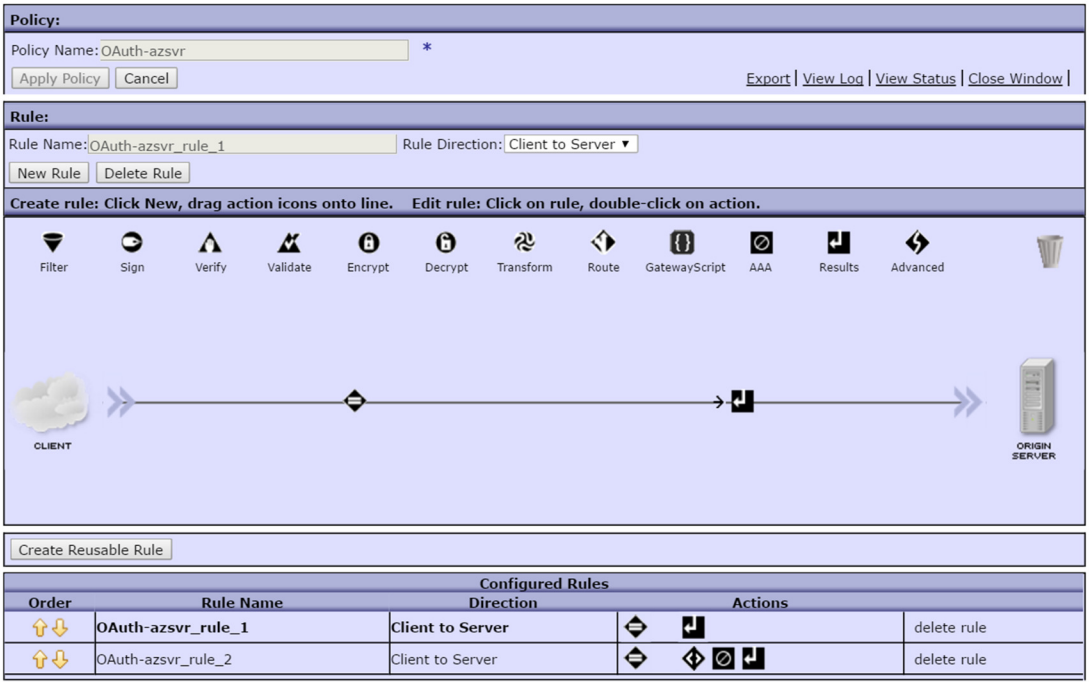
A token *service* is a service that authenticates an entity and issues a security token. It negotiates trust between client applications and services and removes the need for a direct relationship between them. The WTS introduced in DataPower provides a dedicated and simplified service to act as the token service using the existing functionality of the AAA object and other actions of a processing policy. It is a loopback service (a service object without a backend) that can be configured as both the authorization and token endpoint of an authorization server.

- Browse for Web Token Service in search bar and click on “Add New Web Token Service”
- Name it as OAuth-azsvr.
- Scroll down to add endpoints and SSL proxy profile.
- Set the port number to 5050 and make sure that SSL is on.
- Click Add to add the source address record and then click Next.
- For AAA Policy, select clientcred-aaa that we created in the previous section, and click Next.
- Click the Commit button.
- Click the View Web Token Service button to view the results.

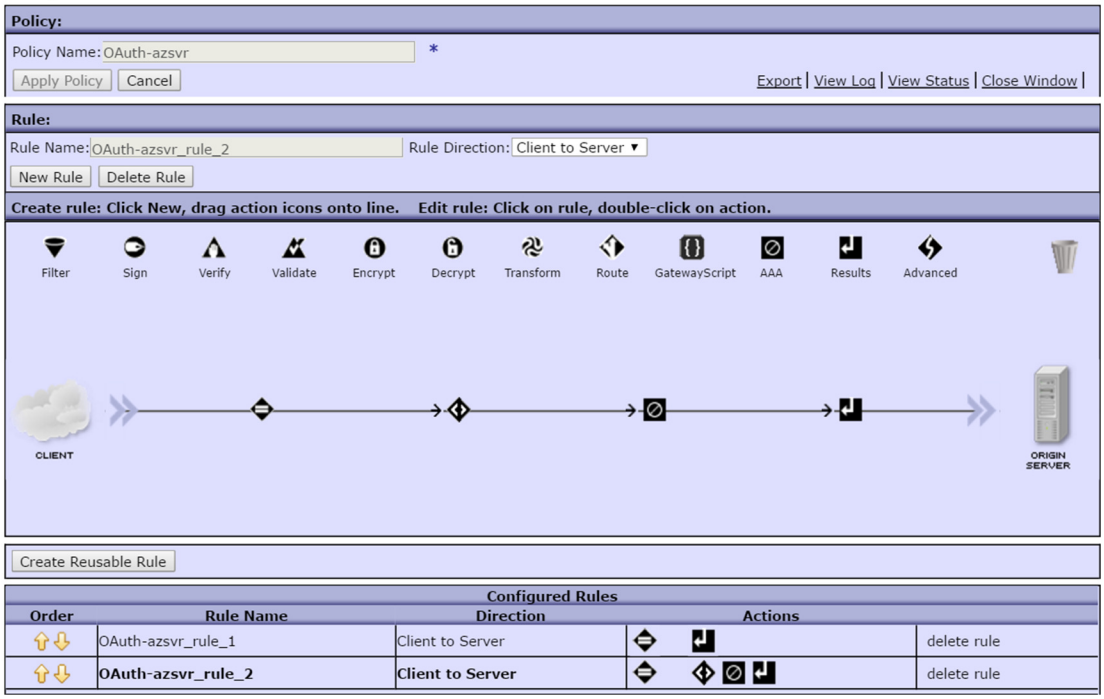


- Create a policy in Web Token Service:
    - For Rule 1, configure the match action and drag the results action.
    - For rule 2, drag the match, advanced, AAA and results action. Select the clientcred-aaa AAA policy that we initially created.

Configure Web Token Service Style Policy



Configure Web Token Service Style Policy



5.Creating AAA policy

- Navigating to Objects > XML Processing > AAA Policy. Click Add to create a new AAA policy and name it rssvr-aaa.
- Select the Identity Extraction tab. Select OAuth as the method and choose the registeredClients OAuth Client Group.
- Select the Authentication tab. Select Pass Identity Token to the Authorize Step.
- Select the Resource extraction tab. Check URL sent by client and Processing metadata. Select oauth scope metadata for the dropdown. DataPower performs a comparison of the output from "URL sent by client" against the scope allowed by the access token.
- Select the Authorization tab. Select Allow Any authenticated client.
- Click Apply.

AAA Policy: rssvr-aaa [up]

Apply Cancel Delete Undo

Methods

- [ ] HTTP Authentication header
- [ ] Password-carrying UsernameToken element from WS-Security header
- [ ] Derived-key UsernameToken element from WS-Security header
- [ ] BinarySecurityToken element from WS-Security header
- [ ] WS-SecureConversation identifier
- [ ] WS-Trust Base or Supporting token
- [ ] Kerberos AP-REQ from WS-Security header
- [ ] Kerberos AP-REQ from SPNEGO token
- [ ] Subject DN of SSL certificate from connection peer
- [ ] Name from SAML Attribute assertion
- [ ] Name from SAML Authentication assertion
- [ ] SAML Artifact
- [ ] Client IP address
- [ ] Subject DN from certificate in message signature
- [ ] Token extracted from message
- [ ] Token extracted as cookie value
- [ ] LTPA token
- [ ] Processing metadata
- [ ] Custom style sheet
- [ ] HTML forms-based authentication
- [x] OAuth

*

Registered OAuth clients registeredClients + ... *

Configure AAA Policy

Main | Identity extraction | **Authentication** | Credential mapping | Resource extraction | Resource mapping | Authorizat

AAA Policy: rssvr-aaa [up]

Apply | Cancel | Delete | Undo

Export | View Log | View Status | Help
Flush cache

Method: Pass identity token to authorization phase *

Cache authentication results: Absolute *

Cache lifetime: 3 Seconds

Configure AAA Policy

Main | Identity extraction | Authentication | Credential mapping | **Resource extraction** | Resource mapping | Authoriz

AAA Policy: rssvr-aaa [up]

Apply | Cancel | Delete | Undo

Export | View Log | View Status | Help
Flush cache

Resource information *

- [ ] URL sent to back end
- [x] URL sent by client
- [ ] URI of top level element in message
- [ ] Local name of request element
- [ ] HTTP operation (GET or POST)
- [ ] XPath expression
- [x] Processing metadata

Processing metadata items: oauth-scope-metadata + ...

6.Create a Multi Protocol Gateway (Enforcement point)

A Multi Protocol Gateway will process the requests between various protocols.

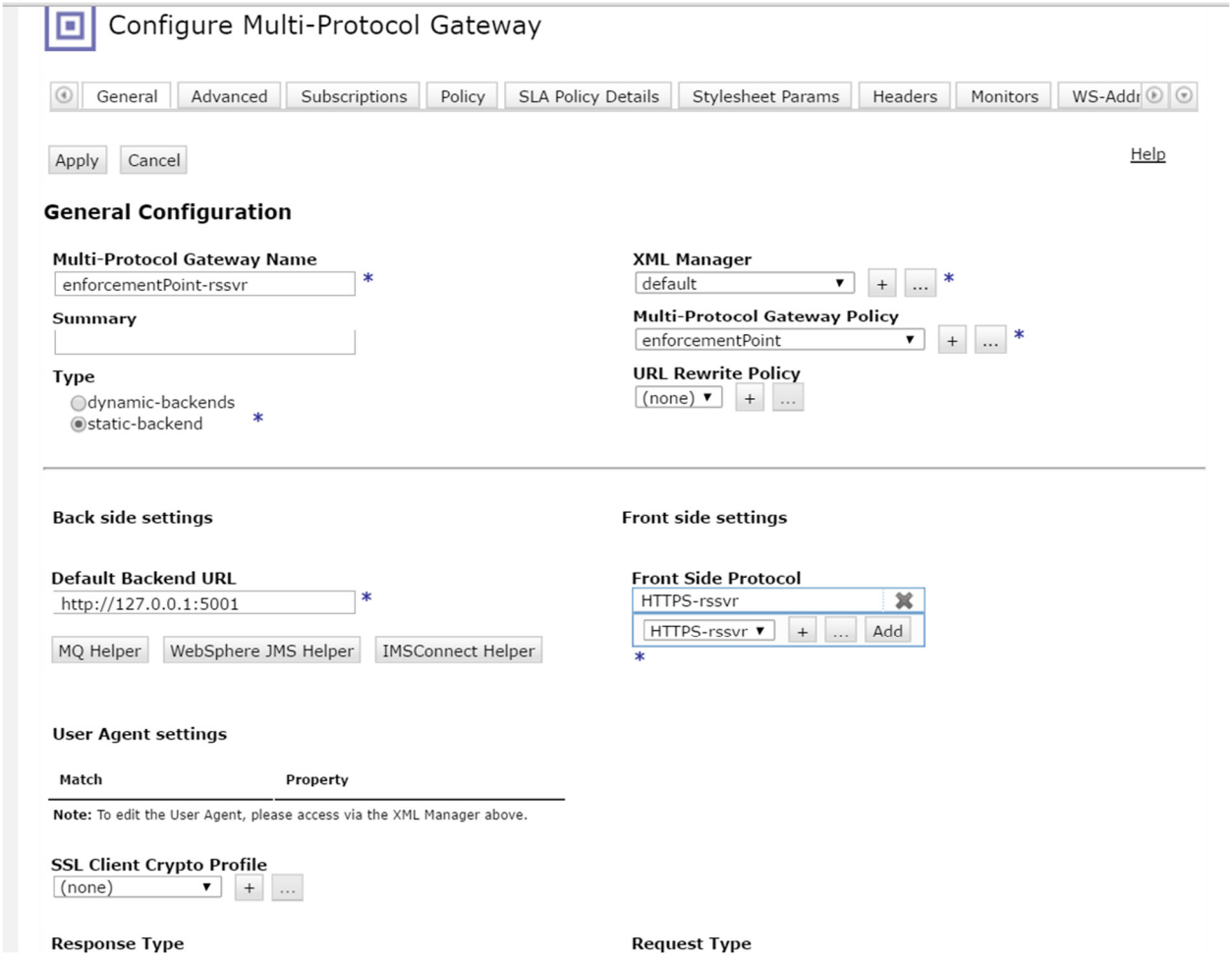
Steps to process request:

1. Receives Request
2. Process request with processing policy
3. Forward to remote server or back end

MPG will use front side handlers to manage client connections. A single MPG can have multiple front side handlers.

- The enforcement point is implemented with a MPGW. Navigate to the Control Panel, select Multi-Protocol Gateway, and click Add. Enter the following.

- Service Name: enforcementPoint-rssvr
- Backend URL: http://127.0.0.1:5001
- Request Type: JSON.
- Response Type: XML



- Create the HTTPS Front Side Protocol by clicking "+" next to Front Side Protocol field and selecting HTTPS Front Side Handler.
  - Name: HTTPS-rssvr

- Port: 5051
- Check GET method.
- SSL Proxy: Select sslserver. Upload the certificate and key which act as validation.
- Click Apply.

Configure Crypto Key

Main

Crypto Key: sslserver [up]

Apply | Cancel | Undo

Export | View Log | View Status | Help
Convert Crypto Key Object

Administrative state: (•) enabled ( ) disabled

File Name: cert:/// ▼
sslserver-privkey.pem ▼ Upload... Fetch... Edit... View... *

Password:

Password Alias: ( ) on (•) off

Configure Crypto Certificate

Main

Crypto Certificate: crypto_cert [up]

Apply | Cancel | Undo

Export | View Log | View Status | Help
Convert Crypto Certificate Object

Administrative state: (•) enabled ( ) disabled

File Name: cert:/// ▼
sslserver-sscert.pem ▼ Details... Upload... Fetch... *

Password:

Password Alias: ( ) on (•) off

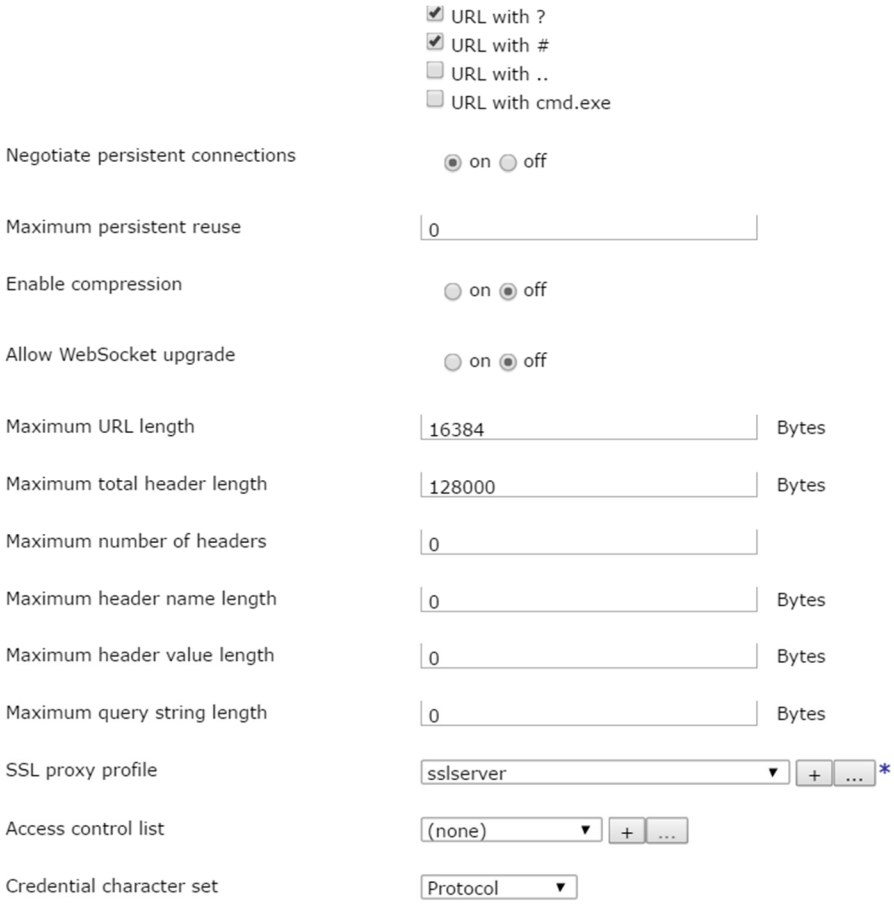
Ignore Expiration Dates: ( ) on (•) off

HTTPS Front Side Handler: HTTPS-rssvr [up]

Apply | Cancel | Undo

Export | View Log | View Status | Help
Quiesce | Unquiesce

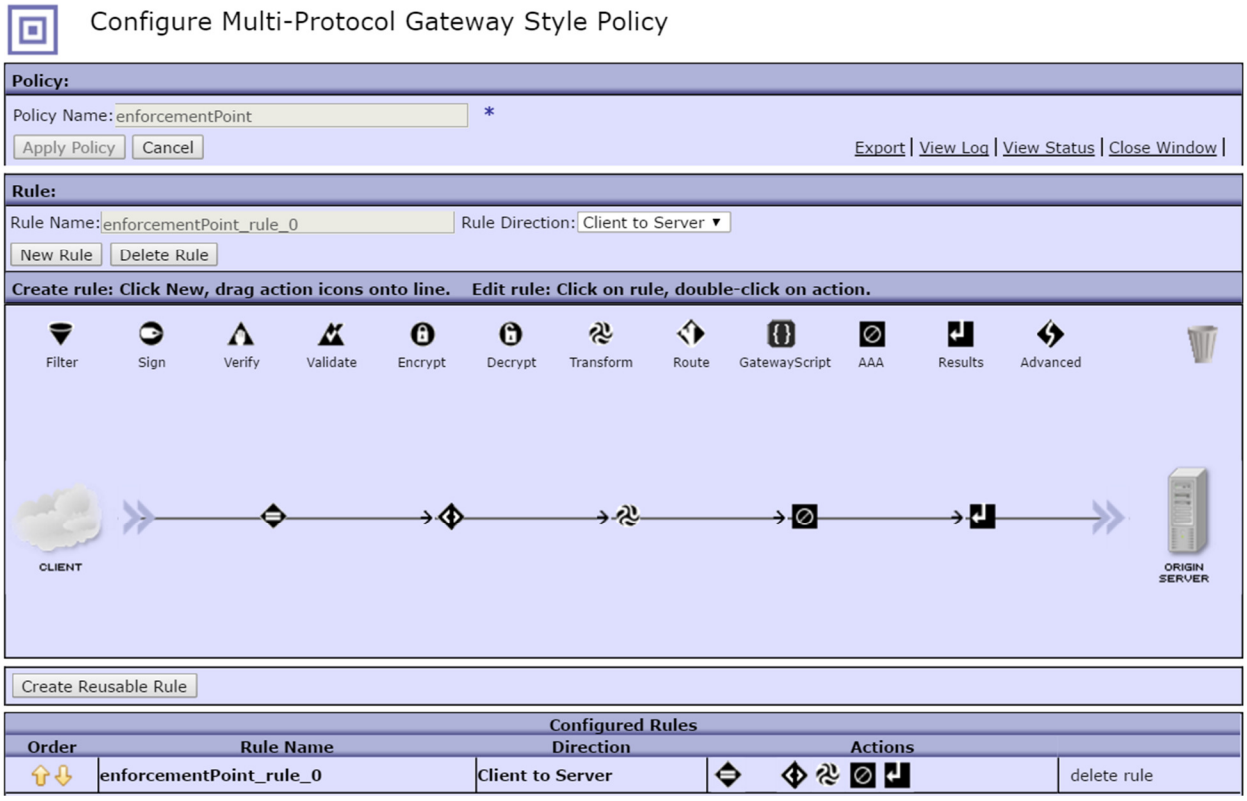
Administrative state	(•) enabled ( ) disabled
Comments	
Local IP address	0.0.0.0 Select Alias *
Port	5051 *
HTTP version to client	HTTP 1.1 ▾
Allowed methods and versions	[x] HTTP 1.0 [x] HTTP 1.1 [x] POST method [x] GET method [x] PUT method [ ] HEAD method [ ] OPTIONS [ ] TRACE method [ ] DELETE method [ ] Custom methods [x] URL with ? [x] URL with # [ ] URL with .. [ ] URL with cmd.exe



- Create the processing policy by clicking "+" next to Multi-Protocol Gateway Policy.
  - Policy Name: enforcementPoint
  - Click the New Rule button and change Rule Direction to Client to Server.
  - Double-click the Match action. Create a new match rule called MatchAll. Add a matching rule for all URLs in the Matching Rule tab.
  - Drag an Advanced action after the Match action. Double-click it and select Convert Query Params to XML. Click Next and then Done.
  - Drag and drop transform action and upload the JSONx_to_XML_OAuth_Request.xslt file and click Done.

- Drag an AAA action after the Convert action and double-click it. Select the rssvr-aaa AAA Policy from the dropdown menu.

- Click Apply Policy and close the window. Then click Apply for the MPGW.



The backend service of the MPG will act as a resource server. This will provide with the required resources.