OpenID Connect and OAuth2: a distributed trust scenario

These diagrams illustrate the use of OpenID Connect and OAuth2 working in tandem to allow a client to access a protected resource.

In this scenario, the Identity Provider (IdP), Authorization Server (AS), and Client are all in separate security domains. The Resource Server (RS) is in the same security domain as the AS to the extent that the RS can validate and trust a token generated by the AS.

The user must log in to the AS using credentials from their IdP, then authorize the client to access the RS for them. The AS is therefore a client of the IdP in addition to being an AS in its own right (for the RS).

These diagrams do not show discovery or registration.

Justin Richer, The MITRE Corporation

Identity Provider

Authorization Server

User

The user connects using a user agent at the client application which needs to access data at the resource server. Assumed: The client is registered and trusted at the AS and the AS is registered and trusted at the IdP.

Client

Resource Server

Out of Band
Front Channel
Back Channel

IdP

AS

User

Service URL

RS

Client

Client gets the service URL (out of band – configuration, through user)

Out of Band

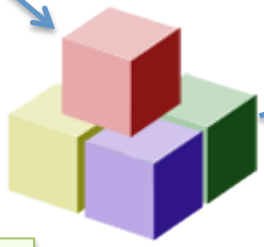
Front Channel

Back Channel

IdP

AS

User

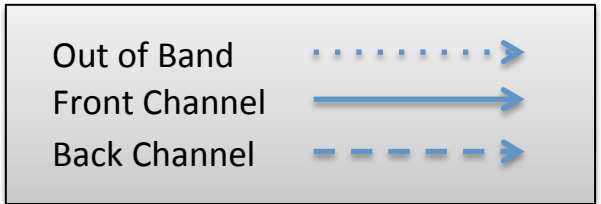


Client

GET

RS

Client fetches the service URL directly to access the data



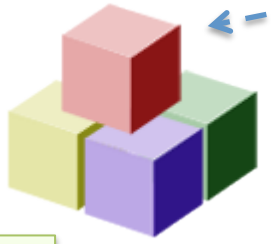
IdP

AS

User

401

RS



Client

Server sends back not authorized response, client must get authorization from user via OAuth

Out of Band
Front Channel
Back Channel

IdP

AS

Authorization
Request

User

302

Client

Client directs user to auth server

RS

Out of Band
Front Channel
Back Channel

IdP

AS

Authorization Request

302

User

RS

Client

Auth server redirects user to their IdP

Out of Band

Front Channel

Back Channel

IdP

AS

Authentication
& Authorization

User

RS

Client

User authenticates to IdP and authorizes auth server for login

Out of Band
Front Channel
Back Channel

IdP

AS

302

Authorization Code
Response

User

RS

Client

IdP directs user back to auth server with auth code

Out of Band
Front Channel
Back Channel

Authorization Code

IdP

AS

User

RS

Client

Auth server trades auth code for tokens

Out of Band
Front Channel
Back Channel

IdP

Access Token
ID Token

AS

User

RS

Client

Auth server trades auth code for tokens

Out of Band
Front Channel
Back Channel

Access Token

IdP

AS

User

RS

Client

Auth server trades access token for claims about the user

Out of Band
Front Channel
Back Channel

IdP

UserInfo
(Claims about user)

AS

User

RS

Client

Auth server trades access token for claims about the user

Out of Band
Front Channel
Back Channel

IdP

AS

User Authorizes
Client at AS

User

RS

Client

User is now authenticated at auth server, authorizes client to access resource server

Out of Band
Front Channel
Back Channel

IdP

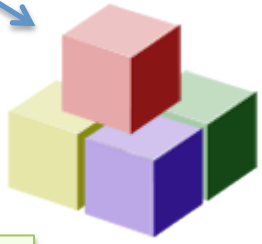
AS

302

User

Authorization Code

RS



Client

Auth server redirects user back to client with auth code

Out of Band
Front Channel
Back Channel

IdP

Authorization Code

AS

User

RS

Client

Client trades auth code for access token

Out of Band
Front Channel
Back Channel

IdP

AS

User

RS

Access Token

Client

Client trades auth code for access token

Out of Band
Front Channel
Back Channel

IdP

AS

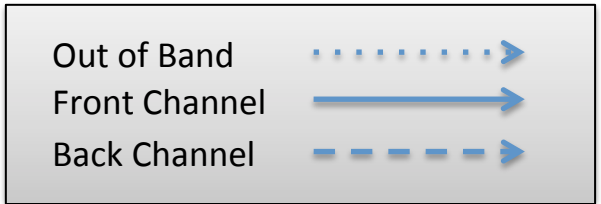
User

RS

Client

Access Token

Client presents access token to resource server

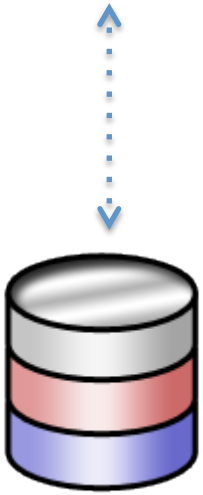


IdP

AS

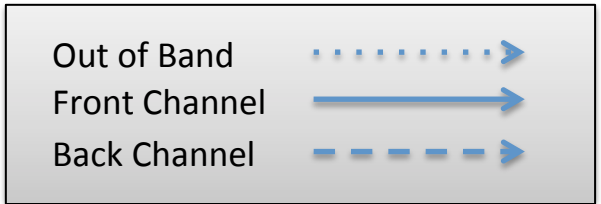
User

Client



RS

Resource server validates token with the authorization server (out of band: could be UMA, token introspection, or structured tokens)



IdP

AS

User

RS

Data

Client

Resource server returns requested data to client, which can now process it, display it, etc.

Out of Band
Front Channel
Back Channel

This documentation has been publicly released as part of the MITREid Connect project, case 11-4559.

https://github.com/mitreid-connect/

Justin Richer, The MITRE Corporation