Mobility OAuth

Introduction

OAuth is an open standard for access delegation, commonly used as a way to grant applications access to exposed APIs on other API Gateway but without providing the passwords. Depending on the use case to be implemented different processes (OAuth authorization flows) can be designed and put in place.

In this case we will take into account two of those OAuth Authorization flows:

- Client Credentials Grant
- Resource Owner Password Credentials Grant

Purpose

The purpose of OAuth is to provide:

- Security: a secure delegated access to server resources on behalf of an API resource owner.
- HTTP based: OAuth allows access tokens to be issued to third-party clients by the authorization server, with the approval of the resource owner. The third party then uses the access token to access the protected resources hosted by the resource server.
- Scope based access: OAuth allows to implement the different authorization flows with different levels of fine-grained rights, so the resource owner can specify the level of access granted to a third party application.

Scope

Principal purpose for Mobility OAuth implementation is to expose a third party Authentication Server that allows the authorisation of different applications to consume a mobility APIs.

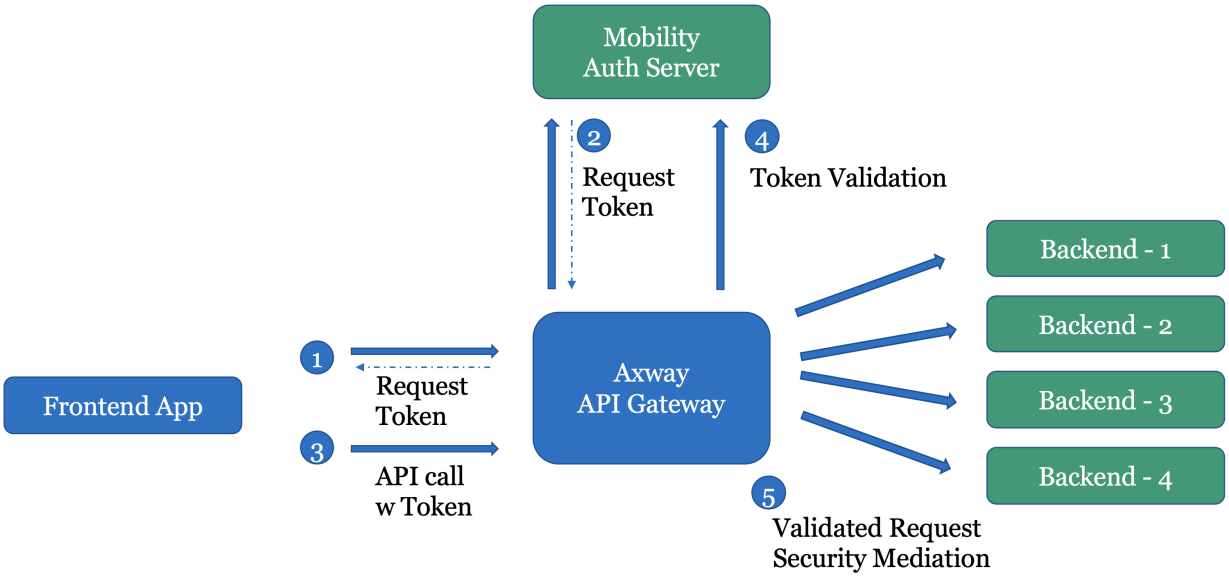
The second purpose is the API Gateway implementation to properly apply security policies to the incoming requests, so the routed requests to the backends are authenticated, authorised and valid. In this sense the scope for the OAuth integration is to check the incoming requests, for the OAuth configured APIs, and validate that the authorisation token is present and valid, this eliminates the need for the backends for checking the token validity. To validate the presented token the API Gateway will need to submit a request to the OAuth servers, responsible to emit those tokens, and inspect the response to allow/block the incoming request.

A third use case is the generation of certain security mechanisms accepted by the backends, and the transformation of the incoming request to accommodate to the backend needs.

Use Cases

- Consumer Application

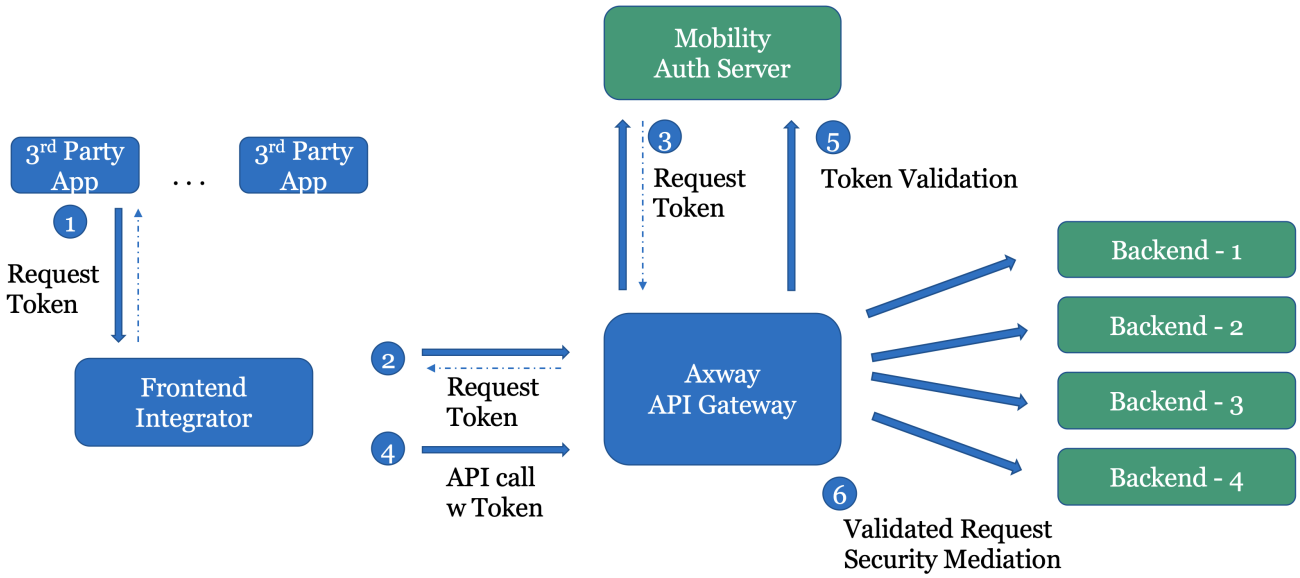
Mobility IdP– Consumer Application



1. The Frontend App, will request an access token to the Axway gateway. The app validation and parameters check will be performed.
2. Axway gateway will request a token to the Mobility Authorization Server will provide it, as soon as the backend application is registered and authorized to use the service in scope.
3. The Frontend App, upon obtention of the Access Token, will request the API Gateway on the needed APIs (provided by the services in the scope of the token requested).
4. API Gateway will check the validity of the token with the Token owner (Mobility Authorization Server).
5. Once the Mobility Authorization Server guarantees that the access token is still valid, the API Gateway sends a request to the Backend that exposes the business need.

- Integrator & 3rd Parties

Mobility Idp – Integrator & 3rd Parties



1. The 3rd party App, will request an access token to the Frontend Integrator. Security mediation and request forging will be performed, and is out of scope of this document.
2. Frontend Integrator will request a token to Axway API Gateway using their own received credentials from Axway, and the 3rd party credentials created in Axway. The app validation, frontend integrator and parameters check will be performed.
3. Axway gateway will request a token to the Mobility Authorization Server will provide it, as soon as the backend application is registered and authorized to use the service in scope.
4. The Frontend App, upon obtention of the Access Token, will request the API Gateway on the needed APIs (provided by the services in the scope of the token requested).
5. API Gateway will check the validity of the token with the Token owner (Mobility Authorization Server).
6. Once the Mobility Authorization Server guarantees that the access token is still valid, the API Gateway sends a request to the Backend that exposes the business need. In this case, special integration mediation will be performed to inform the backend about the 3rd party application (finality).