Authentication Concepts

Overview

The OAuth 2.0 (http://oauth.net/) protocol is based on the acquisition of an access token, which encapsulates the authorization that is granted to the client. In that context, IBM MobileFirst Platform Server serves as an authorization server and is able to generate such tokens. The client can then use these tokens to access resources on a resource server, which can be either MobileFirst Server itself or an external server. The resource server checks the validity of the token to make sure that the client can be granted access to the requested resource. This separation between resource server and authorization server in the new OAuth-based model allows you to enforce MobileFirst security on resources that are running outside MobileFirst Server.

This tutorial covers the following topics:

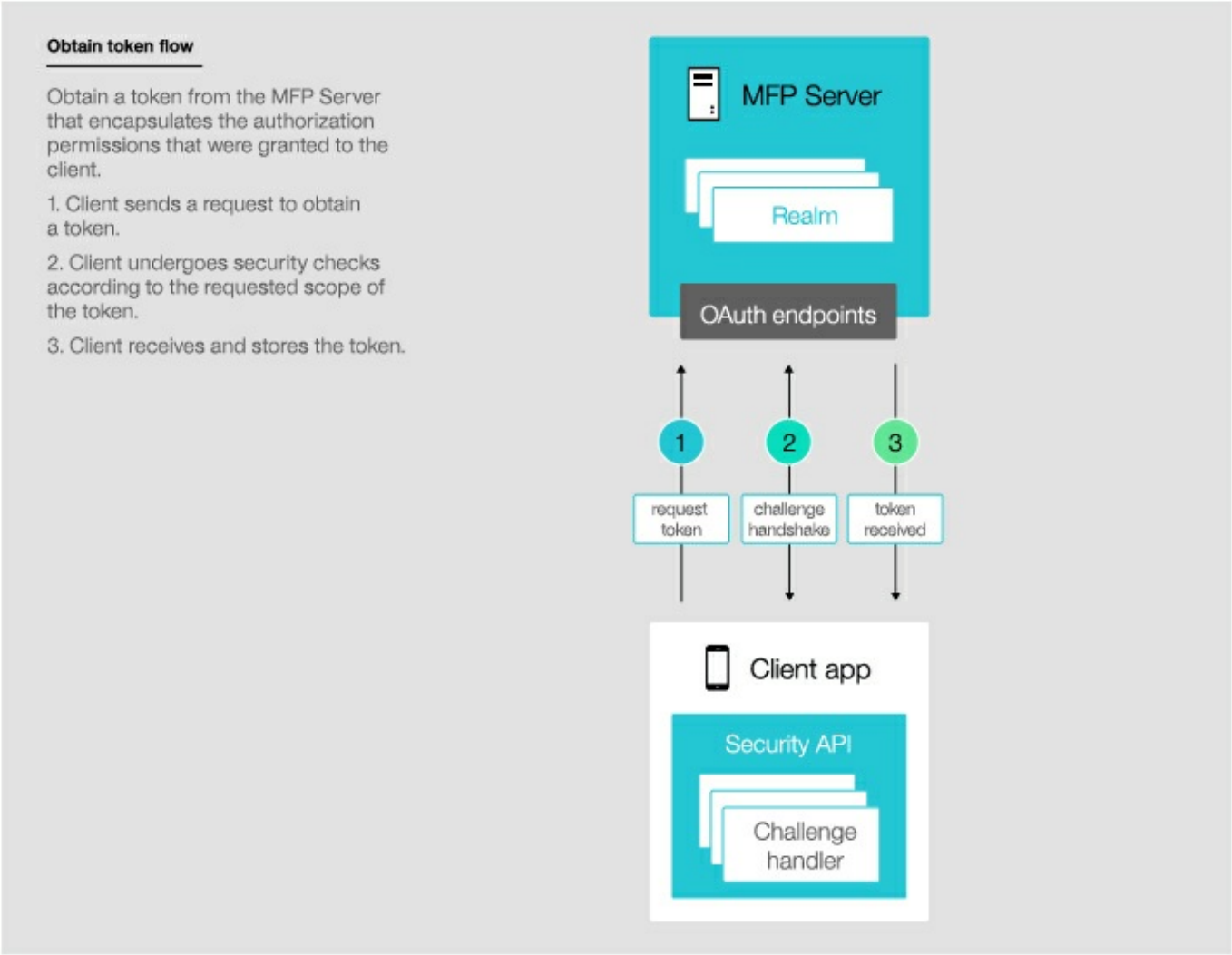
- Authorization flow
- Authorization entities
    - SecurityCheck
    - securityCheckDefinition
    - SecurityCheck implementation
    - SecurityCheckConfiguration
    - Built-in Security Checks
    - Scope
    - Scope Token
    - Challenge Handler
- Protecting resources
    - Java adapters
    - JavaScript adapters
    - External resources
- Configuring Authentication from the MobileFirst Console
- Further reading

Authorization flow

The new MobileFirst end-to-end authorization flow has two phases: the client acquires the token and then uses it to access a protected resource.

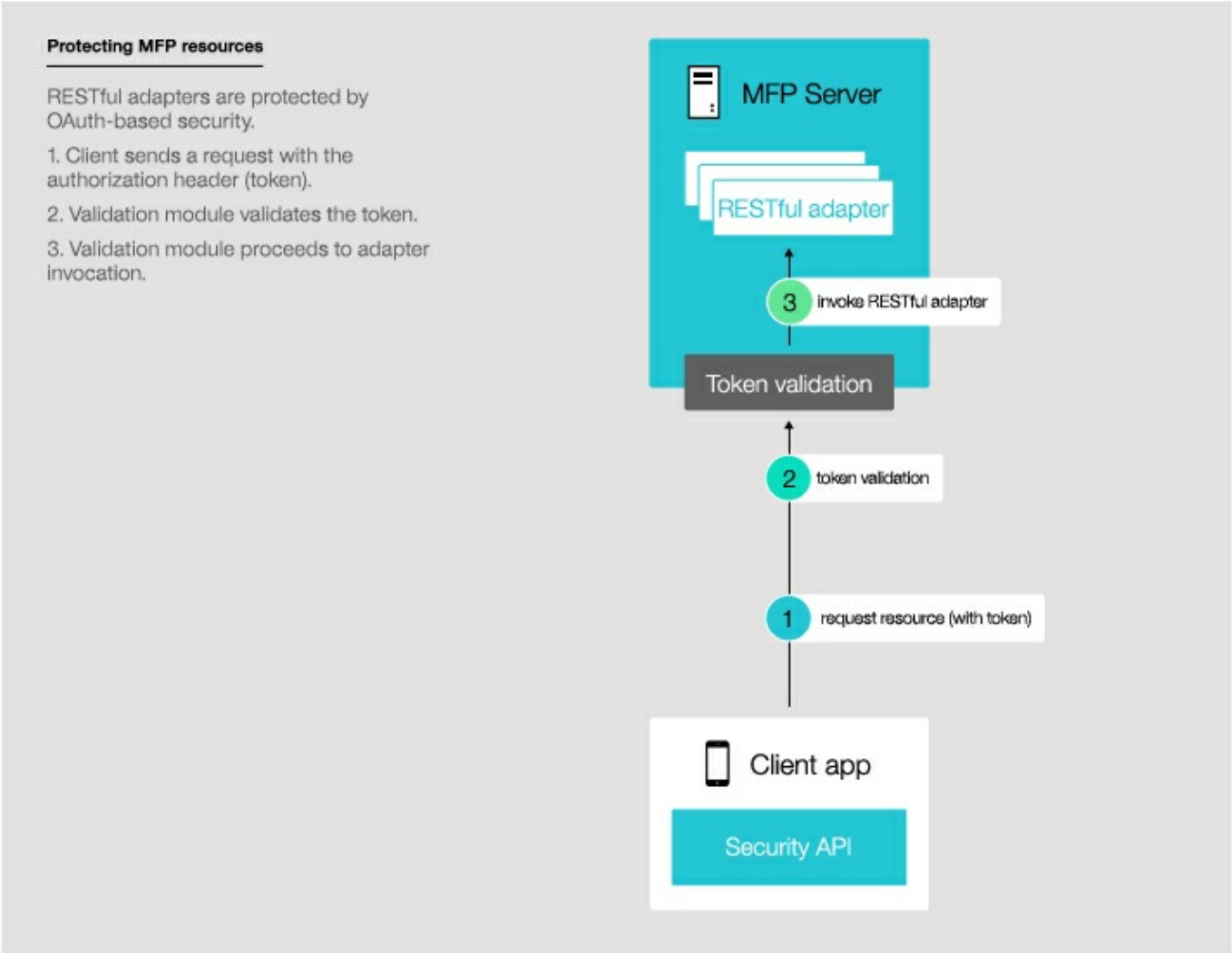
Acquiring a token

In this phase, the client undergoes security checks in order to receive an access token. These security checks use authorization entities, which are described in the next section.



Using a token to access a protected resource

It is possible to enforce MobileFirst security both on resources that run on MobileFirst Server, as shown in this diagram, and on resources that run on any external resource server as explained in tutorial Using MobileFirst Server to authenticate external resources (../../using-mobilefirst-server-authenticate-external-resources/).



Authorization entities

You can protect resources such as adapters from unauthorized access by specifying a **scope** or **scope token** that contains zero or more **SecurityCheck**.

A **SecurityCheck** defines the process to be used to authenticate users. It is often associated with a **SecurityCheckConfiguration** that defines properties to be used by the SecurityCheck.

SecurityChecks are instantiated by **Security Adapters**.

The same SecurityCheck can be used to protect several resources.

The client application needs to implement a **challenge handler** to handle challenges sent by the SecurityCheck.

SecurityCheck

A **SecurityCheck** is an object responsible for obtaining credentials from a client and validate them.

securityCheckDefinition

Security checks are defined inside adapters. Any adapter can theoretically define a SecurityCheck. An adapter can either be a *resource* adapter (meaning it serves resources, content, to send to the client), a *SecurityCheck* adapter, or **both**. However it is recommended to define your *SecurityCheck* in a seprate adapter.

SecurityCheck implementation

SecurityCheckConfiguration

Built-in Security Checks

Scope

Scope Token

Protecting resources

Java adapters

JavaScript adapters

External resources

Configuring Authentication from the MobileFirst Console

Further Reading