Desenvolvendo Aplicações com Autenticação OAuth2 no Fiware

Gabriela Cavalcante Silva, Carlos Eduardo da Silva
Instituto Metrópole Digital
Universidade Federal do Rio Grande do Norte
Natal, RN, Brasil
Email: gabicavalcantesilva@gmail.com, kaduardo@imd.ufrn.br

***Resumo*—A busca por soluções de segurança da informação no desenvolvimento de aplicação vem se tornando uma prioridade cada vez maior. A plataforma Fiware oferece um conjunto de componentes reutilizáveis que facilita o desenvolvimento de aplicações para Cidades Inteligentes. Este artigo apresenta um estudo sobre os mecanismos de autenticação da plataforma Fiware. Como estudo de caso, foi desenvolvida uma aplicação Web Java que demonstra a utilização dos referidos mecanismos.**

***Keywords*-autenticação; gerencia de identidade; segurança;**

I. Introdução

Sabemos que é altamente desejado que informações, aplicativos, ou softwares sejam acessados somente por aqueles autorizados para tal. Para isso, existem vários mecanismos que lidam com diferentes aspectos de segurança da informação, tais como autenticação, políticas de segurança, entre outros. A plataforma Fiware foi projetada para considerar estes aspectos em uma abordagem *secure by design*, onde mecanismos de segurança são fornecidos através de um componentes, também chamados de *Generic Enablers* (GEs), que podem ser utilizados por aplicações ou outros componentes da plataforma [1].

Neste contexto, este trabalho apresenta um estudo sobre a plataforma FIWARE com foco no *Identity Management* GE (IdM), componente responsável pelo gerenciamento de identidades de usuários, organizações e aplicações, assim como pela autenticação dessas entidades. Nosso objetivo é demonstrar como desenvolver aplicações seguras baseada em REST com suporte a autenticação através do componente Fiware IdM GE. Para tal, desenvolvemos uma aplicação Web Java que será utilizada como estudo de caso.

II. GE Fiware para Gestão de Identidades

A gestão de identidades engloba aspectos ligados ao acesso de usuários a redes, serviços e aplicações, incluindo autenticação segura e privada de usuários para dispositivos, redes e serviços [2]. Nesse contexto, a plataforma Fiware oferece o componente Keyrock como sistema para gestão de identidades. Este GE considera requisitos para a gestão de identidades de "coisas" (IoT), e atende às especificações para o gerenciamento de identidades sobre diferentes domínios, permitindo ainda a gestão de autorização, com suporte a definição de papéis e permissões específicas de uma organização ou de uma aplicação.

A Figura 1 apresenta a arquitetura de alto nível do Keyrock. Desenvolvedores interagem com este componente através de um Portal de Desenvolvedor de Aplicações (*Application Developer Portal*) para registrarem suas aplicações no IdM. Usuários finais usam um Portal de Usuário Final (End User Portal) para se registrarem, e gerenciar seus respectivos perfis e suas organizações, enquanto que todas as entidades clientes de aplicações utilizam o IdM para fins de autenticação [1].

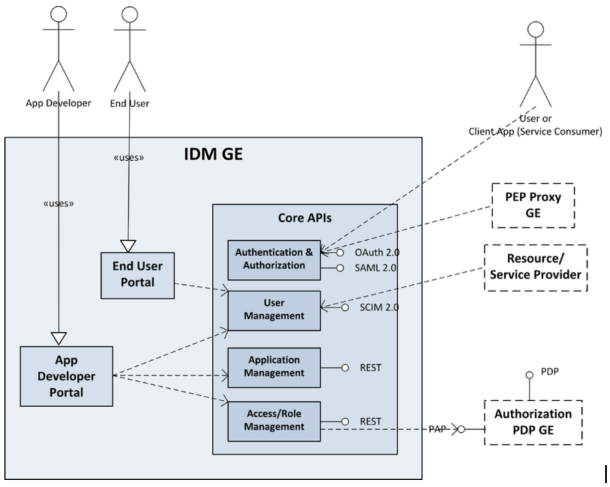

Figura 1. Identity Management GE - Arquitetura em alto-nível [3].

O Keyrock oferece três APIs voltadas para o gerenciamento de usuários, aplicações e políticas de controle de acesso, e uma API para fins de autenticação e autorização. A API que será o foco do nosso estudo, é a **API de Autenticação e Autorização (*Authentication & Authorization*)**, que segue as especificações dos padrões SAML 2.0[1] e OAuth2[2], suportando cenário de SSO e de Federação de Identidades. Mais especificamente, demonstraremos como realizar a autenticação em uma aplicação REST utilizando o protocolo OAuth2.

[1]http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

[2]http://oauth.net/2/

III. Aplicação de Exemplo

Para demonstrar a utilização dos mecanismos de autenticação do Keyrock, desenvolvemos uma aplicação de exemplo que será utilizada ao longo deste trabalho. O exemplo consiste em uma aplicação Web Java, formada por um componente *back-end* e um componente *front-end*. Sua arquitetura é apresentada na Figura 2. O *front-end* consiste em uma interface Web desenvolvida em JSF, que realiza chamadas a um serviço Web REST (nosso *back-end*).

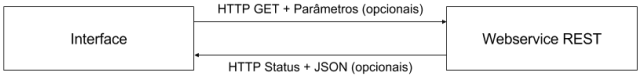

Figura 2. Arquitetura da aplicação de exemplo.

A fim de exemplificar a realização da autenticação da aplicação com o OAuth2, são descritas abaixo as modificações realizadas sobre a aplicação original.

O primeiro passo para se utilizar o serviço de autenticação do Keyrock é efetuar o registro da aplicação no IdM. Ao acessar o endereço do IdM e realizar o login é possível ver na tela inicial uma listagem de aplicações cadastradas e a opção para cadastro de uma nova aplicação. O cadastro é divido em três partes: o registro das informações da aplicação (nome, descrição, etc), a seleção de uma imagem de exibição, e o cadastro dos papéis e permissões associados a aplicação.

Parte das informações informadas inclui uma URL de *Callback*, que deve existir na aplicação. Após autenticação, o IdM redireciona a sessão de navegação para onde a Callback URL estará indicando. Ao finalizar o registro de uma aplicação, suas informações serão exibidas. Nesta tela, temos também um *Client Id* e *Client Secret* gerados para a aplicação. Essas informações serão utilizados no momento em que fizer requisições de autenticação para o Keyrock.

Uma vez registrada, é necessário alterar a aplicação para que a mesma delegue o processo de autenticação ao Keyrock. Para tal, foram inseridos elementos de segurança que só permitem o envio de requisição ao *back-end* após a autenticação do usuário. A nova arquitetura da aplicação é apresentada na Figura 3.

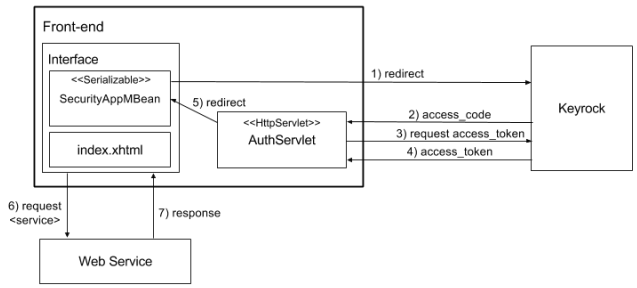

Figura 3. Arquitetura da aplicação de exemplo com autenticação.

O *SecurityAppMBeam* está associado à interface do usuário, e receberá a requisição feita quando o botão responsável por requisitar a autenticação for pressionado, redirecionando o usuário para a tela de login no Keyrock (passo 1). Esse redirecionamento precisa incluir o *Client Id* e *Client Secret* da aplicação. Uma vez autenticado, o Keyrock solicita uma autorização do usuário para a aplicação na qual ele deseja acessar. O Keyrock então retorna um *access-code* para a aplicação, utilizando o endereço cadastrado na URL de Callback (Passo 2). Em nosso exemplo, essa URL aponta para um *servlet* (AuthServlet), que é responsável por recuperar o *access-code* gerado pelo Keyrock. Em seguida, o *servlet* realiza uma requisição ao IdM para obter um *access-token* (passo 3). De posse do *access-token*, a aplicação pode agora seguir seu funcionamento normal, redirecionando o usuário para sua página principal.

Uma vez obtido o *access-token*, requisições podem ser realizadas ao IdM enviando como parâmetro para confirmar a autenticação o token recebido, como a solicitação das informações de um determinado usuário no Keyrock. Também é possível utilizar o token em chamadas para acessar outras GE Fiware, que fariam o papel de back-end e usariam o token para confirmar a autenticação do usuário.

A aplicação de exemplo, junto com outro exemplo desenvolvido em Python, está disponível no repositório do projeto Smart Metropolis[3], junto com instruções para sua utilização.

IV. Conclusão

Neste trabalho apresentamos um estudo sobre a utilização do componente Keyrock da plataforma Fiware para a autenticação de usuários. Desenvolvemos uma aplicação Web de exemplo que demonstra o processo de autenticação através do protocolo OAuth2. Esse processo nos permitiu adquirir um conhecimento mais aprofundado do Keyrock. Nossos próximos passos incluem o desenvolvimento de exemplos envolvendo cenários mais complexos de autenticação e autorização usando o Keyrock, e demonstrando o uso de políticas de controle de acesso na plataforma Fiware.

Referências

[1] Fiware, "Fiware security chapter architecture," Tech. Rep., 2014. [Online]. Available: https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/Security_Architecture

[2] A. Jøsang and S. Pope, "User centric identity management," *AusCERT Asia Pacific Information Technology Security Conference*, 2005.

[3] Fiware, "Fiware architecture description security identity management," Tech. Rep., 2016. [Online]. Available: https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/FIWARE.ArchitectureDescription.Security.IdentityManagement

[3] http://projetos.imd.ufrn.br/SmartMetropolis-InfraestruturaGroup/Example-Application-Security