Estudo da Extensão de Métodos de Autenticação em um Middleware de Nuvens Híbridas

Renan Dembogurski[1], Antônio Tadeu[2], Bruno José Dembogurski[3], Edelberto Franco[1]

[1]Departamento de Ciência da Computação – Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora – MG – Brasil

[2]Laboratório Nacional de Ciência da Computação (LNCC) - Petrópolis - RJ - Brasil

[3]Departamento de Ciência da Computação
Universidade Federal Rural do Rio de Janeiro (UFRRJ) – Nova Iguaçú, RJ – Brasil

ad.renan@gmail.com, atagomes@lncc.br

brunodembogurski@ufrrj.br, edelberto@ice.ufjf.br

Abstract. *Cloud computing has attracted the attention of research, teaching institutions and the market in recent years. Within this context, Cloud Federations, which, in the large area of Identity and Access Management (IAM), have as their main role providing the necessary infrastructure for the application of this technology. Thus, the objective of this work is to present the challenges and experiences of developing, implementing and validating the Identity Management (GId) model whose main solution is to integrate the authentication methods for OpenID Connect (OIDC) with the middleware of the Identity Management project for the Cloud Computing Innovation Center (CICN), Fogbow.*

Resumo. *A Computação em Nuvem (Cloud Computing) vem atraindo a atenção de instituições de ensino e pesquisa e também do mercado nos últimos anos. Dentro desse contexto compreendem-se as federações de nuvens (Cloud Federation), que, na grande área de gestão de identidade e acesso (Identity and Access Management - IAM), têm como principal papel a oferta da infraestrutura necessária à aplicação dessa tecnologia. Assim, o objetivo deste trabalho é apresentar os desafios e experiências de desenvolver, implementar e validar o modelo de Gestão de Identidade (GId) tendo como solução principal a extensão dos métodos de autenticação para suporte ao OpenID Connect pelo middleware do projeto Gestão de Identidade para o Centro de Inovação em Computação em Nuvem (CICN), o Fogbow.*

1. Introdução

A Computação em Nuvem, do inglês *Cloud Computing* [Toosi et al. 2014], vem atraindo a atenção de instituições de ensino e pesquisa e também do mercado nos últimos anos. Suas perspectivas vão desde a utilização para armazenamento em larga escala, distribuído e resiliente, até o processamento em larga escala de grande massa de dados.

Dentro desse contexto, compreendem-se as federações de nuvens (*Cloud Federation*), que, na grande área de gestão de identidade e acesso (*Identity and Access Management* - IAM) [Silva et al. 2018] [Wangham et al. 2010], têm como principal papel a

oferta da infraestrutura necessária à aplicação dessa tecnologia. Essas federações são importantes num contexto onde se percebe a necessidade do uso de recursos computacionais pertencentes a mais de uma nuvem, muitas vezes utilizando tecnologias distintas.

Outro ponto relevante durante o desenvolvimento de pesquisas na área de computação em nuvem e IAM é a privacidade. Pode-se destacar a privacidade voltada para os dados armazenados nas nuvens, já que, além do espaço de armazenamento ser por essência compartilhado, a plataforma tem sua interconexão entre os provedores de serviço de armazenamento em nuvem baseada na infraestrutura da Internet. Nesse cenário, mecanismos de IAM são importantes não só para protegerem os dados dos usuários, mas também para restringirem e garantirem a confiabilidade no acesso aos dados armazenados.

Outra forma que a IAM pode prover benefícios a esse cenário é através da utilização de federações de identidade. Federações de identidade garantem que as entidades que a compõem ofertem um conjunto mínimo de atributos de seus usuários com certo nível de confiança a serviços das mesmas entidades parceiras. Exemplificando os benefícios de uma federação de identidade de forma sucinta, ela permite que um usuário registrado em uma instituição se autentique e utilize serviços de quaisquer instituições parceiras. Em uma federação de identidade os papéis das entidades que armazenam e ofertam os atributos dos usuários e daqueles que provêm serviços a esses usuários é bem definida. Eles são chamados de provedores de identidade e provedores de serviços, respectivamente. Além disso, provedores de identidade podem dispor dos mais diversos mecanismos de autenticação, como certificados digitais, *smartcards*, *tokens* de acesso, biometria, autenticação multi-fator, entre outros, a fim de incrementar a confiança na identidade do usuário para acesso aos provedores de serviços.

Num ambiente de nuvens federado, uma característica particular é a de que um mesmo usuário pode ser representado por entidades distintas, possuindo credenciais distintas, em cada uma das nuvens que compõem a federação. Faz sentido, então, pensar no uso de federações de identidade nesse caso para simplificar o acesso dos usuários com o recurso de *single sign-on* (SSO) [Silva et al. 2018], onde o usuário possui apenas as credenciais do seu provedor de identidade e com elas consegue acessar todos os sistemas de nuvens da federação.

1.1. Objetivo

Este trabalho surgiu como demanda do projeto CICN - Centro de Inovação em Computação em Nuvem - que envolve as seguintes instituições: Telecomunicações Brasileiras S/A (TELEBRAS), Laboratório Nacional de Computação Científica (LNCC), Serviço Federal de Processamento de Dados (SERPRO), Empresa de Tecnologia e Informações da Previdência Social (DATAPREV), sendo coordenado pelo LNCC.

O CICN tem por objetivo desenvolver atividades de pesquisa, desenvolvimento, absorção e transferência de tecnologias em computação em nuvem, estimulando sua adoção pelo setor público. O CICN propõe a definição de uma arquitetura de referência para computação em nuvem para governo eletrônico (e-Gov) e, a partir dela, soluções que permitam a criação de serviços de e-Gov em nuvem.

A demanda de gestão de identidade neste contexto surge como uma forma de estender e validar métodos de autenticação em seu gerente de nuvens híbridas. Para tanto,

deve-se propor e validar um novo método federado de autenticação, no caso o OpenID Connect, no *middleware* de código-fonte aberto e gratuito para gestão de nuvens híbridas inserido no contexto do projeto, chamado Fogbow [CICN Project 2018a]. Em resumo, este trabalho tem como principal objetivo apresentar os métodos de autenticação suportados e relatar os esforços para alcançar o objetivo apresentado anteriormente.

1.2. Motivação

As motivações para introdução de um novo método de autenticação são várias, pois o OpenID Connect (OIDC) provê algumas facilidades que são importantes para o ambiente em questão. Inicialmente, é importante ressaltar o uso do padrão OAuth2 [IETF OAuth Working Group 2018] que, atualmente, é critico para dispositivos móveis, internet das coisas (IoT) e segurança web. Alinhando-se ao OAuth2, é possível oferecer suporte aos desafios de gerenciamento de acesso, incluindo o login único para dispositivos móveis, a autenticação adaptativa de vários níveis e o acesso federado a recursos de nuvem de terceiros, sendo este último o qual é o mais importante neste contexto. Isto expande as possibilidades do Fogbow a aceitar cenários fora do projeto CICN, dando suporte a outras federações que não são acadêmicas.

1.3. Organização do Trabalho

O trabalho está organizado como segue: na Seção 2 são apresentados as principais formas de autenticação tanto federados quanto locais; já a Seção 3 descreve os conceitos e a arquitetura inerentes ao cenário estudado; a Seção 4 descreve a proposta para suporte ao novo método de autenticação no ambiente de nuvens híbridas, concluindo na Seção 5 com um estudo de caso sobre a implementação; por fim, conclui-se o trabalho e descreve-se as perspectivas futuras na Seção 6.

2. Fontes de Autenticação

Além da forma mais tradicional de autenticação baseada apenas em um par usuário e senha, há diversos outros métodos. Dentre esses, esta seção aborda os modelos e padrões de autenticação com maior destaque na literatura e que têm relação com o *middleware* de gerência a ser apresentado neste trabalho. São eles: X.509, Kerberos, CAS, VOMS, OIDC [OpenID Connect 2018] e Security Assertion Markup Language (SAML) [OASIS 2018].

2.1. X.509

Na criptografia, o X.509 é um padrão que define o formato dos certificados digitais. Os certificados X.509 são usados em diversos protocolos da Internet, incluindo o TLS/SSL, os quais são a base do HTTPS [Rescorla and Schiffman 2018], o protocolo seguro para navegar na web. Eles também são usados em aplicativos off-line, como assinaturas eletrônicas. Um certificado X.509 contém uma chave pública e uma identidade (ou um nome de host, ou uma organização, ou um indivíduo). Tais informações são assinadas para formar o certificado. O certificado é dito autoassinado quando ele é assinado pela chave privada correspondente à sua própria chave pública, em vez de ser assinado por uma autoridade de certificadora.. Quando um certificado é assinado por uma autoridade certificadora confiável, ou ainda validado por outros meios, alguém que detém esse certificado pode confiar na chave pública que ele contém para estabelecer comunicações seguras com

outra parte e para validar documentos assinados digitalmente pela chave privada correspondente.

Além do formato dos próprios certificados, o X.509 especifica listas de revogação de certificado como meio de distribuir informações sobre certificados que não são mais válidos e um algoritmo de validação do caminho de certificação, que permite que certificados sejam assinados por certificados CA intermediários, que são por sua vez, assinados por outros certificados, eventualmente, atingindo uma âncora de confiança.

O X.509 é definido pelo setor de Padronização da União Internacional de Telecomunicações (ITU-T) e é baseado no ASN.1, outro padrão ITU-T.

2.2. Kerberos

O Kerberos é muito mais um protocolo de autenticação, que tem seu surgimento na década de 1980. Em constante evolução, atualmente, suporta também a autenticação SSO. Por ser de código-fonte aberto, disponibilizado gratuitamente, e existir há longa data, o Kerberos conta com amplo suporte de diversas plataformas, como Windows, Linux, Solaris, AIX e z/OS. Originalmente criado pelo *Massachusetts Institute of Technology* (MIT), O Kerberos é composto por três partes: um cliente, um servidor e um terceiro confiável conhecido como Kerberos *Key Distribution Center* (KDC). Este fornece serviços de autenticação e concessão de registro.

O KDC pode ser visto como um mantenedor do repositório de contas de usuário, e onde esse repositório pode ser baseado, por exemplo, em LDAP. Nele, é armazenada uma chave de longo prazo para cada usuário (ou como chamado nesse ambiente, proprietário) em seu repositório de contas. Essa chave deriva-se da senha do proprietário, e somente o KDC e o usuário deverão saber qual é a chave de longo prazo e a senha associada.

Encontra-se a proposta e utilização de autenticação SSO com Kerberos para clientes Java, aproveitando o *cache* de credenciais gerenciado pelo Kerberos na autenticação.

2.3. Community Authorization Service

Criado na Universidade de Yale, o *Community Authorization Service* (CAS) [Pearlman et al. 2002] passou por vários grupos até ser, atualmente, suportado pela organização Apereo. O CAS apresenta, basicamente, as entidades comuns ao ambiente de IAM, o SP e IdP. Suportando diversas fontes de autenticação como LDAP, RADIUS, banco de dados, certificados X.509, dentre vários outros. Existe uma forte semelhança entre o CAS e o SAML, e como nota de curiosidade observa-se que o SAML foi proposto por volta de 2001 e implementado pela Internet2 como Shibboleth a partir de 2003, e o CAS foi proposto no mesmo ano que SAML, em 2001, e tem sua versão 1.0 liberada também por volta de 2003.

Além da integração com diversos métodos de autenticação, o CAS tem suporte a protocolos de autorização como OAuth [IETF 2018] e também pode aplicar conceitos como a autorização por papéis, grupos ou atributos, vistos a frente nesse relatório.

2.4. Virtual Organization Membership Service

Virtual Organization Membership Service (VOMS) [Alfieri et al. 2003] é um framework para autenticação e controle de acesso desenvolvido inicialmente para o contexto de **grade**

computacional. O VOMS utiliza políticas e autorizações baseadas em papeis. A política global está relacionada ao nível de política de VO, que é uma política de autorização geral, avaliando se um usuário possui uma credencial válida ou pertence a um determinado grupo. Já a local, é conduzida pelo RP (Provedor de Recursos), responsável por fornecer o recurso em si. Para utilizá-lo, o usuário deve reenviar suas credenciais para o RP (local) junto com uma pré-autorização concedida pelo VO (global). A idéia principal é que o usuário, mesmo que autorizado pelo VO, possa ter seu acesso restrito localmente.

2.5. Security Assertion Markup Language

A *Security Assertion Markup Language (SAML)* é um framework baseado em XML que define uma infraestrutura para troca de informações seguras da autenticação do usuário, seus direitos e atributos. Essa infraestrutura é responsável por possibilitar a comunicação entre os parceiros. A sua especificação é elaborada pelo *Security Services Technical Committee* (SSTC) que faz parte da *Advancing Open Standards for the Information Society* (OASIS) [OASIS 2018]. Também são apresentadas informações de segurança na forma de asserções (declarações), definindo as regras e a sintaxe para geração, requisição, transferências e uso dessas asserções.

Atualmente na versão 2.0, a especificação SAML é um dos padrões mais adotados pelo modelo de identidades federadas. Este consiste de alguns componentes que funcionam como blocos que podem ser combinados em configurações diferentes para suportar implementações de cenários diferentes. Os componentes primeiramente permitem transferência de identidade, autenticação, atributos e informações de autorização entre provedores de identidades e de serviços que possuem uma relação de confiança estabelecida. O núcleo da especificação SAML define a estrutura e o conteúdo das asserções e mensagens de protocolo usado para transferir essas informações.

Dois conceitos bastante comuns quando se trata de ambientes SAML, são o metadata e o contexto de autenticação. O primeiro define como informar e compartilhar informações entre entidades SAML e papéis (como provedor de identidade, provedor de serviço, etc.), onde o metadata é o arquivo que contém informações sobre ligações SAML, identificadores de identidade, protocolos de transportes suportados, certificados digitais e chaves criptográficas; já o contexto de autenticação surge quando, por exemplo, um provedor de serviço precisa ter acesso à informações detalhadas referentes ao mecanismo de autenticação que é empregado pelo provedor de identidade do usuário. O contexto de autenticação SAML é usado justamente na comunicação entre o provedor de serviços e o de identidades, permitindo ao primeiro solicitar uma forma específica de autenticação e ao segundo permitir o acesso do usuário em seus serviços.

O SAML permite a autenticação SSO e também a integração com mecanismos de autorização para prover, através dos atributos do usuário, formas interessantes de controle de acesso.

2.6. OpenID Connect

O OIDC, OpenID Connect, é um protocolo de autenticação única SSO que permite que os usuários se autentiquem em sites (provedor de serviços), utilizando o identificador OpenID (conta) que desejarem. Isto permite ao usuário controlar as informações que serão compartilhadas com as aplicações [OpenID Connect 2018]. Neste protocolo, quando um

usuário fornece o seu identificador ele é, imediatamente, redirecionado para o seu provedor OpenID, que realiza a autenticação utilizando o método de autenticação, suportado no provedor OpenID indicado. Após a confirmação dos dados, o usuário é redirecionado para o provedor de serviços, junto com seus atributos.

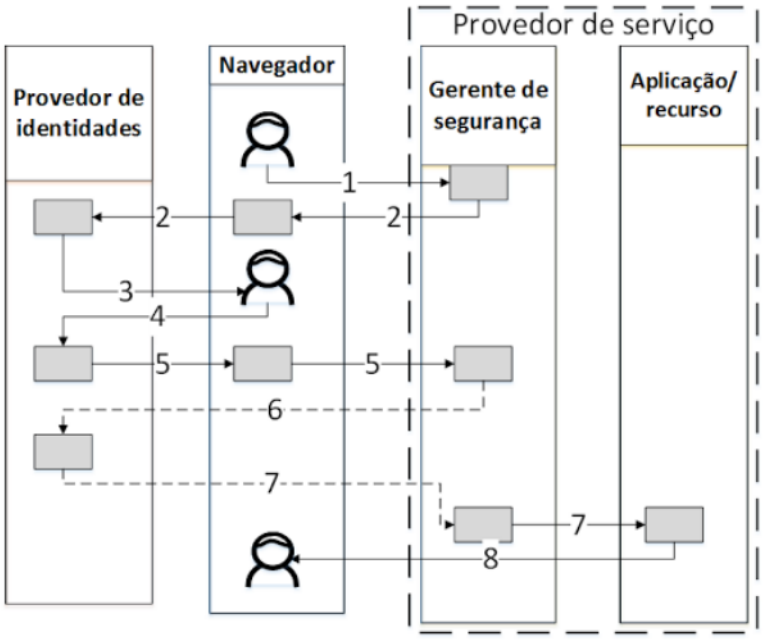

Figura 1. Fluxo da interação entre Usuário, IdP e SP no OpenID Connect [Recordon and Reed 2006].

A Figura 1 traz a interação entre o provedor de identidade (IdP) e o provedor de serviço (SP) no OIDC[1]. As entidades presentes na figura são: o IdP, o navegador (browser) do usuário, o SP composto por um gerente de segurança, responsável por controlar o acesso ao recurso, e o recurso ou aplicação em si. Descrevendo os passos, temos: no passo 1 o usuário requisita acesso ao recurso através do gerente de segurança do SP. Este então requisita, no passo 2, a autenticação do usuário através do navegador, a autenticação propriamente é realizada após o encaminhamento do usuário através do navegador ao seu IdP. Essa autenticação ocorre desde o passo 2 até os passos 3, 4 e 5, onde, no passo 4 o usuário consente ou não a liberação de atributos ao SP. No passo 5 o IdP envia ao SP o que é chamado de prova de autenticação, e após essa ser recebida pelo SP é validada no passo 6. No passo 7 o SP efetivamente recebe os atributos do usuário vindos do IdP, e decide no passo 8 se o acesso do usuário ao recurso está liberado ou não.

OIDC é um protocolo que permite a autenticação SSO web, e também oferece a autorização com o auxílio de um outro protocolo específico a isso, o OAuth. Então, quando se vê passos referentes à autorização, como liberação de atributos na Figura 2, quem oferta essa funcionalidade é, na verdade, o protocolo OAuth. Atualmente na versão 2.0 é possível ver como a camada adicional de autorização criada pelo OAuth funciona em [Hardt 2012].

[1]Neste trabalho as entidades RP e OP são tratados como equivalentes ao SP e IdP de uma federação de identidade, respectivamente.

3. Fogbow

O Fogbow [CICN Project 2018a] é um *middleware* que federa nuvens privadas. Dizer que essa solução federa soluções em nuvens é afirmar que realiza a comunicação e união entre sistemas diferentes de gestão em nuvem para a sua gestão interna. Por exemplo, é possível que uma das nuvens participante esteja apoiada sobre a solução OpenStack, e outro participante utilize o OpenNebula.

O acesso à federação de recursos em nuvem é possível porque o Fogbow oferta uma API comum aos participantes, padronizando a forma de comunicação entre as gerências das soluções em nuvem e a visão global da federação de nuvens ofertada pelo FogBow. Sendo assim, sua principal idéia é fornecer as funcionalidades de uma federação de recursos distribuídos em nuvem em um nível superior, através da implementação de um *middleware* cujo único propósito é suportar operações na federação. O FogBow é implantado no topo do orquestrador de nuvens do IaaS em cada membro da federação, e para deixar mais clara essa visão, este trabalho apresenta sua arquitetura e como a autenticação e autorização acontece no ambiente

3.1. Arquitetura

A arquitetura de uma federação Fogbow de nuvens privadas baseia-se em dois componentes principais: o gerente de membros (*Membership manager*) e o gerente de alocação (*Allocation Manager*).

O gerente de membros é responsável por determinar quais membros da federação estão atualmente ativos. Ele implementa um protocolo de sincronização *gossip-style* para descobrir os endereços dos gerenciadores de alocação conhecidos por outros gerentes de membros. Os gerentes de membros trocam informações periódicas entre si, a fim de manter suas informações de membros atualizadas.

Da mesma forma, periodicamente, o gerenciador de alocação é executado em um determinado site, e interage com seu gerenciador de membros associado - que pode ser executado no mesmo site ou em um site remoto - para notificar sua disponibilidade atual e recuperar a lista de gerentes de alocação atualmente ativos. A Figura 2 mostra como a federação Fogbow é composta, suas entidades e componentes, divididos em cada um de seus sites particulares (*i.e.* Site A, Site B etc). Lembrando que, em cada site pode haver uma solução de orquestrador diferente para a nuvem privada.

3.2. Orquestradores

Os orquestradores são responsáveis por efetivamente realizar ações sobre os recursos, como alocação e liberação dos mesmo. Atualmente aqueles suportados pelo *middleware* são: CloudStack [Apache CloudStack 2017] OpenNebula [OpenNebula 2018], OpenStack [OpenStack 2018] e Azure [Microsoft 2018], assim com o ambiente do Amazon EC2 [Amazon EC2 2018] [Toosi et al. 2014].

Assim como os outros orquestrados apresentados até aqui, o Apache CloudStack é uma plataforma open-source voltado para a criação de uma infraestrutura de nuvens públicas, privadas e híbrida como serviço (*Infrastructure as a Service* - IaaS). Mais detalhes podem ser encontrados em [Apache CloudStack 2017]. Já o Azure, pode ser visto como um IaaS que permite que nós Windows e nós Linux sejam facilmente provisionados na nuvem da Azure da Microsoft. Algum dos motivadores à adoção do suporte ao

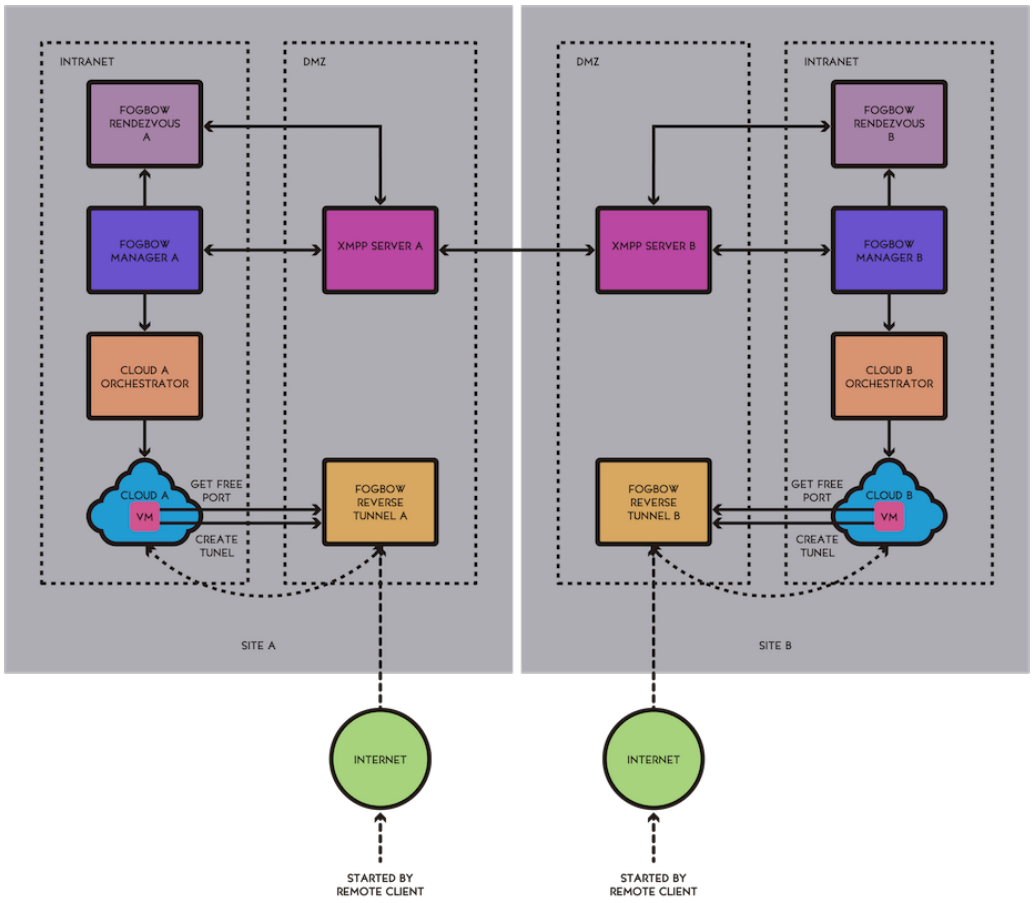

Figura 2. Arquitetura da federação Fogbow com seus gerentes de membros e alocação [CICN Project 2018a].

Azure neste projeto é o suporte de uma estratégia de nuvem híbrida para IaaS, lembrando que o orquestrador comercial oferece também serviços de *Software as a Service* (SaaS) e *Platform as a Service* (PaaS). Com uma linha próxima ao provisionamento comercial de recursos de infraestrutura do Azure tem-se a Amazon EC2. Do mesmo princípio e conceito de suporte a nuvens híbridas o Fogbow suporta essa IaaS, provendo acesso através de seu *middleware*.

É importante ressaltar que o *middleware* Fogbow já se encontra bem desenvolvido e documentado quanto ao suporte a esses orquestradores. Toda a documentação referente, inclusive à sua configuração, pode ser encontrada na página oficial [CICN Project 2018a], mais especificamente em [CICN Project 2018c].

3.3. Plugin de Identidade

A implementação atual do Fogbow fornece diversos plugins que lidam com tipos diferentes de credenciais. Lidar com credenciais diferentes, transformando-as em um formato específico de um outro ambiente é conhecido em gestão de identidade como tradução de credenciais. O Fogbow tem suporte por meio de plugins aos métodos de

autenticação: Shibboleth (SAML), VOMS, entre outros que podem ser consultados em [CICN Project 2018b].

Se a credencial for autêntica, o FM (*Fogbow Manager*) verificará se o usuário da federação autenticada está autorizado a executar a solicitação de serviço. Isso, por sua vez, é realizado por um plugin de Autorização de Usuário da Federação. Existem implementações deste plugin que funcionam em conjunto com cada um dos plugins de autenticação. O plugin de autorização do VOMS verifica o nome da Organização Virtual (OV) e as funções que o usuário pode ter, que estão listadas no proxy fornecido, para decidir se o usuário da federação deve ter acesso ao serviço solicitado. Ao usar Shibboleth para autenticação, a autorização é feita com base no conjunto de atributos contidos na asserção de autenticação SAML. O Fogbow fornece um plugin de autorização com base em uma *White List*. Esta lista pode ser configurada para autorizar usuários que tenham um atributo específico que corresponda aos valores listados no arquivo de configuração do plugin, no caso do Shibboleth, e, como comentado, também pode ser utilizado como uma verificação da OV de origem do usuário (quando utilizado para o VOMS), restringindo ou não o acesso do usuário.

Finalmente, também existem plugins nativos de autorização de interoperabilidade que são capazes de interagir com o mecanismo de autorização do orquestrador da nuvem.

4. Proposta

A arquitetura de componentes integrantes de todo o fluxo de autenticação OIDC no Fogbow pode ser visto na Figura 3. Nela é possível identificar o usuário e os componentes que formam a federação OpenID Connect, responsável por conter o provedor de identidades com as credenciais e atributos do usuário[2], o módulo de autenticação OIDC, que deve ser composto por um portal web (chamado de portal de autenticação) e um módulo de autenticação OIDC integrado ao servidor web Apache. O módulo é responsável por permitir a troca de atributos entre tokens entre o lado da aplicação web de serviço e o provedor de identidade. Por último o Fogbow pode ser visto com seus dois principais componentes para a autenticação, o portal de consulta e gerência de recursos (Dashboard) e o gerenciador de recursos em si, representado pelo *Fogbow Manager*.

Na Figura 3, vê-se um usuário que deseja se autenticar no ambiente Fogbow utilizando o método OIDC e realizar operações sobre Fogbow através do Dashboard. Cada um dos passos pode ser descritos como: **(1)** usuário solicita autenticação OIDC ao portal de autenticação; **(2)** o portal redireciona o pedido ao módulo OIDC suportado pelo Apache; **(3)** por sua vez o módulo OIDC encaminha o pedido ao Provedor de Identidade OIDC que recebe as credenciais do usuário **(4)**. Os passos **(3)**, **(4)** e **(5)** são uma abstração dos passos descritos pela Figura 1. Sendo assim, o passo **(5)** ao retornar os atributos do usuário os repassa ao portal de autenticação **(6)**. O portal por sua vez pode encaminhar os atributos do usuário ao Plugin de Identidade do Fogbow **(7)**, localizado no lado Fogbow, mais precisamente no Dashboard. O usuário desta forma já está habilitado a utilizar o Fogbow e gerenciar recursos através da interface nativa do *middleware* **(8)**, que efetivamente realizará ações sobre os recursos através do Fogbow Manager **(9)**.

A proposta, portanto, está embasada na integração entre uma federação de identidade OIDC e o Fogbow através de um módulo Apache OIDC. E, como visto, para o

[2]O atributo *eppn* é o mais importante neste cenário, e serve como identificador do usuário no Fogbow.

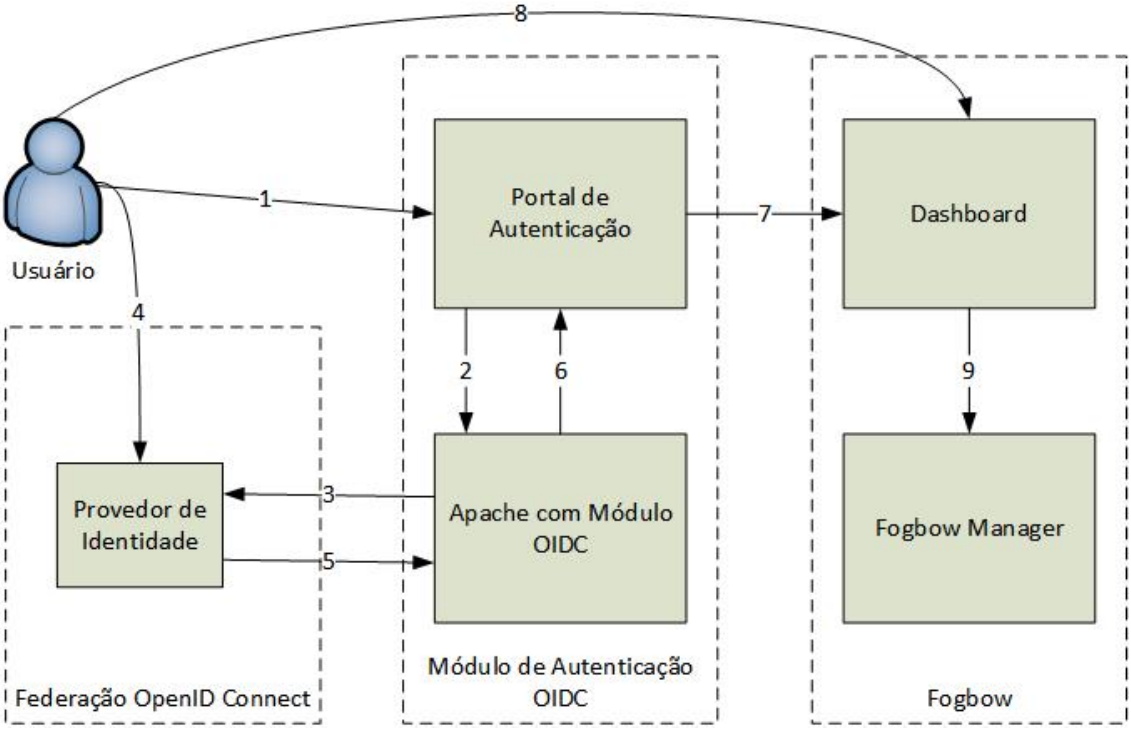

Figura 3. Passos para a autenticação OIDC no ambiente Fogbow.

suporte de comunicação e, principalmente, a interpretação dos atributos do usuário vindos desse módulo, o Plugin de Identidade tem papel fundamental nesse contexto. Ele deve ser capaz de identificar, compreender e tratar os atributos do usuário, transpondo a credencial OIDC no formato Fogbow. Desta forma, foi desenvolvido um Plugin de Identidade OIDC para o Fogbow, com base nos demais plugins existentes (*i.e.* SAML, VOMS, LDAP etc).

5. Resultados

Esta seção apresenta os resultados obtidos por este trabalho, com suas telas de autenticação e validação da proposta.

O ambiente de validação e desenvolvimento tem o apoio do serviço GIdLab da RNP, o laboratório de experimentação em gestão de identidade da Rede Nacional de Ensino e Pesquisa. Neste ambiente é possível validar a integração com os métodos de autenticação federada sobre o OpenID Connect a partir da infraestrutura do ambiente MITRE ID já existente. Durante a validação também foram testados a autenticação com SAML/Shibboleth CAFe na CAFe Expresso e com uma base de usuários LDAP.

Uma máquina virtual (VM) foi criada no GIdLab, respondendo pelo endereço `http://cicn.gidlab.rnp.br`. Está VM tem a seguinte configuração: Ubuntu 16.04 LTS, com 2 GB de RAM, 50GB de HD e uma CPU de 2.6GHz. Para a validação da solução não se faz necessário realizar alocações reais de recursos em orquestradores da nuvem[3]. Portanto, foi utilizada a solução de *NoCloud*.

[3]Essa afirmação se deve ao fato de que uma vez o usuário acessando o Dashboard, este já tem permissões para gerenciar recursos sob o Fogbow Manager. Os detalhes para a realização de alocação de recursos e a

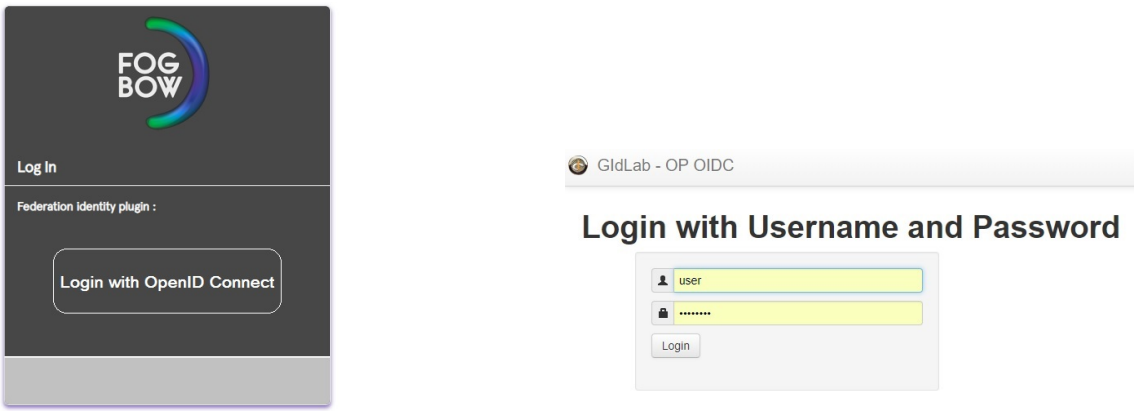

Figura 4. (a) Tela de autenticação no Fogbow. (b) Tela de autenticação no provedor de identidade do OIDC.

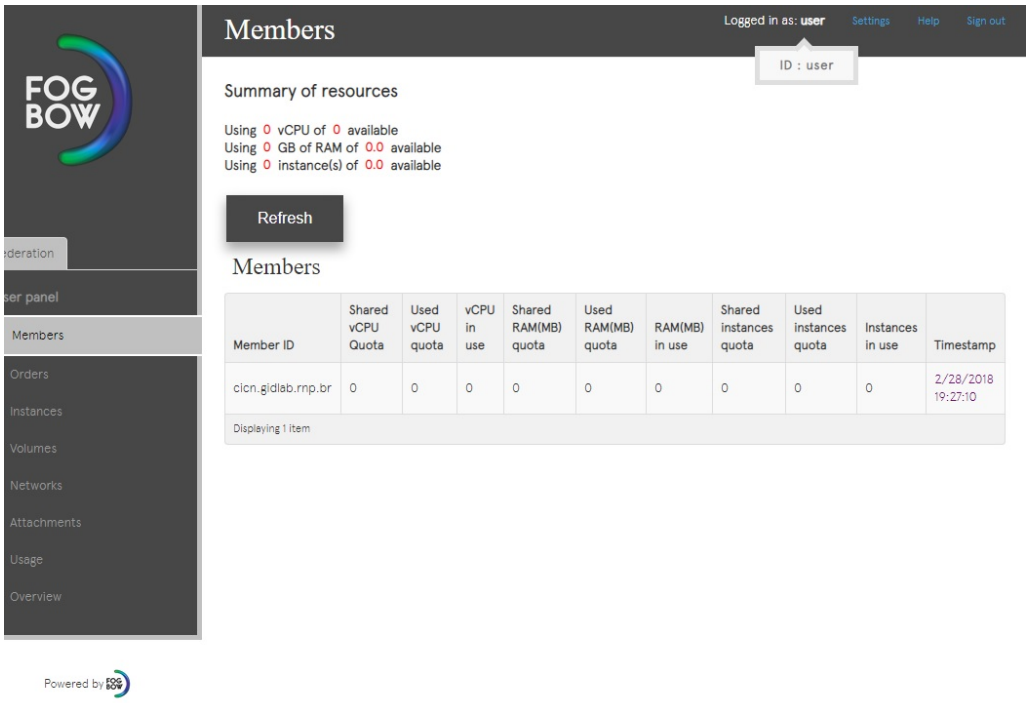

Figura 5. Tela do Dashboard do Fogbow.

As Figuras 4(a) e 4(b) e a Figura 5 mostram as telas para a autenticação no Fogbow utilizando o OIDC. Na Figura 4(a) é possível ver a tela de seleção do método de autenticação pelo usuário, que é então redirecionado para autenticação em seu IdP, como mostra a Figura 4(b). Já a Figura 5 mostra o Dashboard do Fogbow depois da autenticação ter sido realizada com sucesso no IdP OIDC e transposta ao amiente do Fogbow.

6. Conclusões e Trabalhos Futuros

Este trabalho apresentou, acima de tudo, uma solução de gestão de nuvem híbrida com suporte a diversos métodos de autenticação, inclusive federadas. Foi ainda proposto e implementado um novo plugin para um diferente método de autenticação não antes suportado

associação das credenciais do Fogbow com os orquestradores foge do escopo deste trabalho.

pelo *middleware*. Fica claro que com a utilização do Fogbow a gestão da autenticação em nuvens híbridas fica facilitado, uma vez que é transparente a transposição de credenciais entre este ponto único de autenticação e o orquestrador da nuvem em si.

A incorporação do método de autenticação federado OIDC ao Fogbow permite que seja suportado agora um novo modelo de autenticação, amplamente utilizado por outras soluções. Além disso, o trabalho confirma a facilidade da adição de novos métodos de autenticação para nuvem através do suporte dado pelo plugin de identidade do *middleware*.

Como trabalhos futuros, um próximo passo é a adição de uma autenticação do tipo multi-fator. Outra proposta e a adição do suporte a diferentes métodos de autorização daqueles hoje suportados pelos orquestradores de nuvem em geral (*i.e. Role-based Access Control* - RBAC). Aproveitando do benefício de se ter um *middleware* para a gerência de nuvens híbridas, está em estudo a utilização do módulo de autorização do framework ACROSS [Silva et al. 2018], que incorporado a este *middleware* facilite o suporte ao método ABAC para autorização no ambiente.

Referências

Alfieri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Gianoli, A., Spataro, F., Bonnassieux, F., Broadfoot, P. J., Lowe, G., Cornwall, L., Jensen, J., Kelsey, D. P., Frohner, Á., Groep, D. L., de Cerff, W. S., Steenbakkers, M., Venekamp, G., Kouril, D., McNab, A., Mulmo, O., Silander, M., Hahkala, J., and Lörentey, K. (2003). Managing dynamic user communities in a grid of autonomous resources. *CoRR*, cs.DC/0306004.

Amazon EC2 (2018). Amazon Web Services. Disponível em `https://aws.amazon.com/ec2/`. Acessado em: 06/09/2018.

Apache CloudStack (2017). The Apache Software Foundation. Disponível em `https://cloudstack.apache.org/`. Acessado em: 06/09/2018.

CICN Project (2018a). Fogbow. Disponível em `http://www.fogbowcloud.org/`. Acessado em: 06/09/2018.

CICN Project (2018b). Fogbow behavioral plugins. Disponível em `http://www.fogbowcloud.org/interoperability-behavioral-plugins`. Acessado em: 06/09/2018.

CICN Project (2018c). Fogbow manager configuration. Disponível em `http://www.fogbowcloud.org/install-configure-fogbow-manager`. Acessado em: 06/09/2018.

Hardt, D. (2012). The oauth 2.0 authorization framework. Disponível em `https://tools.ietf.org/html/rfc6749`. Acessado em: 06/09/2018.

IETF, O. W. G. (2018). Oauth. Disponível em `https://oauth.net/`. Acessado em: 06/09/2018.

IETF OAuth Working Group (2018). Oauth2. Disponível em `https://oauth.net/2/`. Acessado em: 06/09/2018.

Microsoft (2018). Microsoft azure. Disponível em `https://azure.microsoft.com`. Acessado em: 06/09/2018.

OASIS (2018). Security Assertion Markup Language - SAML. Disponível em https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security. Acessado em: 06/09/2018.

OpenID Connect (2018). OIDF The OpenID Foundation. Disponível em https://oauth.net/2/. Acessado em: 06/09/2018.

OpenNebula (2018). OpenNebula Project. Disponível em https://opennebula.org/. Acessado em: 06/09/2018.

OpenStack (2018). OpenStack Foundation. Disponível em https://www.openstack.org/. Acessado em: 06/09/2018.

Pearlman, L., Welch, V., Foster, I., Kesselman, C., and Tuecke, S. (2002). A community authorization service for group collaboration. In *Policies for Distributed Systems and Networks, 2002. Proceedings. Third International Workshop on*, pages 50–59.

Recordon, D. and Reed, D. (2006). Openid 2.0: A platform for user-centric identity management. In *Proceedings of the Second ACM Workshop on Digital Identity Management*, DIM '06, pages 11–16, New York, NY, USA. ACM.

Rescorla, E. and Schiffman, A. (2018). The secure hypertext transfer protocol. Disponível em https://tools.ietf.org/html/rfc2660. Acessado em: 06/09/2018.

Silva, E. F., Muchaluat-Saade, D. C., and Fernandes, N. C. (2018). Across: A generic framework for attribute-based access control with distributed policies for virtual organizations. *Future Generation Computer Systems*, 78:1 – 17.

Toosi, A. N., Calheiros, R. N., and Buyya, R. (2014). Interconnected cloud computing environments: Challenges, taxonomy, and survey. *ACM Comput. Surv.*, 47(1):7:1–7:47.

Wangham, M., Mello, E., Böger, D., Gurios, M., and Fraga, J. (2010). Gerenciamento de identidades federadas. In Porto, L., editor, *Livro de Minicursos do SBSEG*. SBC.