Universidade Federal Fluminense

EDELBERTO FRANCO SILVA

ACROSS: um Framework de Autenticação e Autorização Baseado em Políticas e Atributos para Organizações Virtuais

NITERÓI

2016

Universidade Federal Fluminense

EDELBERTO FRANCO SILVA

ACROSS: um Framework de Autenticação e Autorização Baseado em Políticas e Atributos para Organizações Virtuais

Tese de Doutorado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor. Área de concentração: Sistemas de Computação.

Orientadora: Profa. Dra. Débora Christina Muchaluat Saade

Co-orientadora: Profa. Dra. Natalia Castro Fernandes

NITERÓI

2016

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

S586 Silva, Edelberto Franco
ACROSS : um Framework de autenticação e autorização baseado em políticas e atributos para organizações virtuais / Edelberto Franco Silva. – Niterói, RJ : [s.n.], 2016.
107 f.

Tese (Doutorado em Computação) - Universidade Federal Fluminense, 2016.
Orientadores: Débora Christina Muchaluat Saade, Natalia Castro Fernandes.

1. Sistemas paralelos e distribuídos. 2. Framework (Programa de computador). 3. Gestão de identidade. I. Título.

CDD 004.35

ACROSS: um Framework de Autenticação e Autorização Baseado em Atributos e Políticas Distribuídas para Organizações Virtuais

EDELBERTO FRANCO SILVA

Tese de Doutorado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor. Área de concentração: Sistemas de Computação.

Aprovada por:

Profa. Débora Christina Muchaluat Saade, Dra. / UFF
(Orientadora)

Profa. Natalia Castro Fernandes, Dra. / UFF
(Co-orientadora)

Prof. Célio Vinicius Neves de Albuquerque, PhD / UFF

Prof. Igor Monteiro Moraes, Dr. / UFF

Profa. Noemi de La Rocque Rodriguez, Dra. / PUC-Rio

Prof. Alexandre Sztajnberg, Dr. / UERJ

Prof. Antônio Jorge Gomes Abelém, Dr. / UFPA

Niterói, 28 de Julho de 2016.

"A gente tem que sonhar, senão as coisas não acontecem."
Oscar Niemeyer

À minha amada Mãe.

Agradecimentos

Agradeço a Deus pelo dom da vida e por Ele estar presente em todos os momentos. À UFF (Universidade Federal Fluminense), em especial no laboratório MídiaCom e sua equipe, responsáveis por proporcionar diversas experiências e enriquecedoras oportunidades. A todos aqueles envolvidos nessa segunda casa, minha eterna gratidão e respeito ao profissionalismo, companheirismo e sensibilidade sempre demonstrado.

Em especial agradeço à Profa. Débora Muchaluat-Saade por aceitar me orientar e fazê-lo sempre com muita dedicação, e também pela sua compreensão nos mais diversos momentos. Agradeço à minha co-orientadora Profa. Natalia Castro Fernandes, que sempre se mostrou disposta a ajudar e uma pesquisadora entusiasta. Agradeço ao Prof. Célio Vinicius N. Albuquerque por sempre estar presente, desde o Mestrado neste instituto, apoiando e acreditando no meu trabalho. Na pessoa da Marister (nossa segunda mãe) estendo meus agradecimentos a cada colega que compartilha deste ambiente tão agradável de trabalho que é o MídiaCom.

Agradeço ao amigo Prof. Eduardo Pagani Julio, por todos esses anos de companheirismo. Minha gratidão a toda ajuda que me foi dada. E, aos amigos da época de república, Bruno Dembogurski, Gustavo Semaan, Carlos Henrique Sant'Anna, Lucas Lattari, Renan Dembogurski e Felipe Gomes.

À minha família, pais, Lindolfo e Leonor, e meu irmão Leonardo Franco, sua esposa Paula Cristina e o pequeno sobrinho Rafael Fernando, meu muito obrigado por sempre estarem presentes e disponíveis nos mais diversos momentos, e, principalmente, por compreenderem a escassez de tempo para partilhar os momentos em família e ainda se orgulharem da dedicação que a escolha pela vida acadêmica impõe.

À CAPES pelo apoio financeiro através de bolsa de estudos. À RNP, ao projeto FIBRE e todos os envolvidos, com merecido destaque aos colaboradores Profa. Noemi Rodriguez da PUC-RIO e Loïc Baron da UPMC (Sorbonne/Paris 6).

Resumo

Ao longo dos últimos anos, acompanha-se um crescente interesse, tanto no âmbito acadêmico como empresarial, de soluções para criação de ambientes federados. Tais federações visam facilitar o acesso de usuários a serviços compartilhados entre as diversas entidades parceiras. Em alguns casos, um serviço é oferecido por diferentes instituições de maneira compartilhada e deve-se permitir o acesso apenas a determinados membros dessas ou outras instituições, como por exemplo em um projeto interinstitucional. Esses cenários definem o conceito de organização virtual. Tais ambientes possuem requisitos particulares a cada instituição parceira e também genéricos à organização virtual como um todo. Desta forma, é interessante permitir que funcionalidades comuns nesse ambiente, envolvendo, principalmente, questões sobre autenticação e autorização, possam ser facilmente integradas a uma organização virtual.

Esta tese contribui ao estado da arte em gerência de organizações virtuais, propondo um novo *framework* que facilita, tanto a entrada de instituições em uma organização virtual (OV) quanto a criação de uma nova OV, colaborando para a solução de problemas importantes na gestão de identidade e acesso. Esse *framework* tem o nome de ACROSS, ***A**ttribute-based access **C**ont**RO**l and di**S**tributed policie**S***.

Além de propor a especificação, documentação e implementação de um *framework* que integra um conjunto mais amplo de funcionalidades do que aqueles presentes na literatura, o ACROSS permite a gerência e integração de uma organização virtual a soluções de gestão de identidade e acesso amplamente difundidas, como é o caso das federações de identidade e o conceito de controle de acesso baseado em atributos. Diferentemente dos demais trabalhos propostos na literatura, o ACROSS utiliza uma ampla e diversificada gama de conceitos de gestão de identidade e acesso em um único *framework* modularizado, integrado e extensível. Além de suportar a autenticação por federações de identidade e outros métodos, facilita a gerência de atributos específicos a uma organização virtual, através do conceito de provedores de atributos adicionais. Permite a transposição de credenciais entre o ambiente da federação de identidade e a organização virtual de forma simples e ainda realiza controle de acesso utilizando

políticas distribuídas e padrões baseados em papel e atributos. Além disso, permite que quaisquer que sejam os tipos de recursos compartilhados pela organização virtual, esses sejam facilmente integrados ao *framework* para a aplicação do controle de acesso.

Comparado a outros trabalhos, o ACROSS apresenta uma solução genérica de gestão de identidade focada em autenticação e autorização para organizações virtuais, uma vez que foi proposto para ser utilizado em ambientes com quaisquer tipos de recursos distribuídos. Outro resultado interessante é o auxílio na integração a qualquer ambiente de organização virtual, independentemente de características particulares, como tipos específicos de credenciais ou mensagens de gerência de recursos.

O *framework* ACROSS foi implementado em um espelho do GIdLab, o laboratório de experimentação em gestão de identidade real suportado pela RNP. Sua implementação e utilização foi validada por uma organização virtual hipotética que oferece acesso a recursos distribuídos.

Palavras-chave: ACROSS, Framework, Organização Virtual, Gestão de Identidade, Autenticação, Autorização

Abstract

Research interests about creating and using federations have recently increased in academic and business environments. Those federations have as main purpose to facilitate user access to shared-resource environments. In some cases, one specific service is maintained by many partners, and only specific users from different institutions may access those distributed resources, in order to participate in a common research initiative, as in an inter-institutional project. These scenarios are called virtual organizations and have particular requirements for partner institutions and common requirements for the virtual organizations itself. Thus, it is interesting to allow that common features involving authentication and authorization can be easily integrated into a virtual organization.

This thesis contributes to the state of the art in virtual organization management, proposing a new framework to facilitates both the entry of new institutions to an existing virtual organization (VO), as the creation of a new VO, contributing to the solution of important issues about identity and access management. This framework is called ACROSS, ***A**ttribute-based access **C**ont**RO**l and di**S**tributed policie**S***.

Besides proposing the specification, documentation and implementation of a framework with more integrated features than those in the literature, ACROSS enables the management and integration of a virtual organization to widespread identity and access management solutions, as the case of identity federations and the concept of attribute-based access control. Unlike other literature proposals, ACROSS uses wide and diversified concepts of identity and access management in a single modularized, integrated and extensible framework. Besides supporting identity federation authentication and other methods, it facilitates managing virtual organization specific attributes through the concept of additional attribute providers. It allows credential translation between the identity federation environment and the virtual organization and supports access control using distributed policies and mechanisms based on user roles and attributes. In addition, it allows easy integration of different kinds of shared-resources into the framework for access control purposes.

Compared to other works, ACROSS presents a generic identity management solution focusing on authentication and authorization for virtual organizations, since it has been proposed for use in environments with different kinds of distributed resources. Another interesting result is the help for integration into any virtual organization environment, regardless of particular characteristics, such as specific credential types or resource management messages.

The ACROSS framework was implemented in a GIdLab mirror. GIdLab is a real open lab for identity management experimentations supported by RNP. Its implementation and use has been validated by a hypothetical virtual organization that provides access to distributed resources.

Keywords: ACROSS, Framework, Virtual Organization, Identity Management, Authentication, Authorization

Abreviações

A&A Autenticação e Autorização

AA Attribute Authority

AAA Authentication, Authorization and Accounting

ABAC Attribute-Based Access Control

AC Attribute Certificates

ACL Access Control List

ACROSS Attribute-based access ContROl and diStributed policieS

ADF Access Control Decision Functions

AEF Access Control Enforcement Functions

API Application Programming Interface

CAFe Comunidade Acadêmica Federada

CAS Community Authorization Service

CAS2 Central Authentication Service

DAC Discretionary Access Control

DN Distinguished Name

DS Discovery Service

EDUROAM Education Roaming

FI Future Internet

FIBRE Future Internet Testbeds Experimentation Between Brazil and Europe

FIM Federated Identity Management

FIRE Future Internet Research and Experimentation

GENI Global Environment for Network Innovations

GIdLab Laboratório de Gestão de Identidade

GId Gestão de Identidade

GSI Globus Toolkit's Grid Security Infrastructure

HTTP HyperText Transfer Protocol

IAM Identity and Access Management

ICP Infraestrutura de Chave Pública

IAM Identity and Access Management

IdM Identity Management

IdP Identity Provider

IP Internet Protocol

JISC Joint Information Systems Committee

LDAP Lightweight Directory Access Protocol

LOA Level of Assurance

MAC Mandatory Access Control

NGS UK National Grid Service

NSF National Science Foundation

OASIS Organization for the Advancement of Structured Information Standards

OFELIA OpenFlow in Europe: Linking Infrastructure and Applications

OV Organizações Virtuais

PAP Policy Administration Point

PEP Policy Enforcement Point

PERMIS PrivilEge and Role Management Infrastructure Standard

PIP Policy Information Point

PKI Public Key Infrastructure

PMI Privilege Management Infrastructure

QoS Quality of Service

RBAC Role-based Access Control

REST Representational State Transfer

RNP Rede Nacional de Ensino e Pesquisa

RP Resource Provider

Rspec Resource Specification

SAAM Shibboleth and Apache Authorization Module

SAML Security Assertion Markup Language

SARONGS Shibboleth Access to Resources on the NGS

SCIM System for Cross-domain Identity Management

SDN Software Defined Networks

SFA Slice-Based Federation Architecture

SHEBANGS Shibboleth Enabled Bridge to Access the National Grid Service

SOA Source of Authority

SOAP Simple Object Access Protocol

SP Service Provider

SSH Secure Shell

SSO Single Sign-On

STCFed Serviço para Transposição de Credenciais de Autenticação Federada

UML Unified Modeling Language

VM Virtual Machine

VO Virtual Organization

VOMS Virtual Organization Membership Service

VOOT Virtual Organisation Orthogonal Technology

WAYF Where Are You From

XACML eXtensible Access Control Markup Language

XML EXtensible Markup Language

Sumário

Lista de Figuras xvii

Lista de Tabelas xx

1 Introdução 1

1.1 Motivação 4

1.2 Objetivos 6

1.3 Contribuições 6

1.4 Organização do texto 8

2 Embasamento Teórico 9

2.1 Organizações Virtuais 9

2.2 Federações de Identidade e SAML 12

2.3 Controle de Acesso 13

2.4 Resumo dos Conceitos Básicos 17

2.4.1 Autenticação 17

2.4.2 Agregação de Atributos 18

2.4.3 Transposição de Credenciais 18

2.4.4 Controle de Acesso 18

3 Trabalhos Relacionados 20

3.1 Frameworks de Autenticação e Controle de Acesso 20

3.2 Gestão de Identidade e Controle de Acesso 23

4 Proposta do framework ACROSS **27**

4.1 Detalhamento da proposta 28

4.2 Utilização do ACROSS 35

4.2.1 *Identity Federation Module* 35

4.2.2 *Attribute Module* 35

4.2.3 *Access Control Module* 37

4.3 Diagrama de Classes 48

4.4 ACROSS e o estado-da-arte 49

4.5 Sumarização da Comparação 52

5 Casos de Uso do framework ACROSS **55**

5.1 FIBRE 55

5.1.1 Cenário de Validação 55

5.1.2 Identity Federation Module 56

5.1.3 Attribute Module 57

5.1.3.1 Agregação de Atributos 58

5.1.3.2 Transposição de Credenciais 59

5.1.4 Listagem de Recursos 61

5.1.5 Access Control Module 61

5.2 Organização Virtual Hipotética 62

5.2.1 Cenário de Validação 62

5.2.2 Configuração de Políticas Globais do ACROSS 63

5.2.3 Configuração de Políticas Locais do ACROSS 65

5.2.4 Exemplo de Controle de Acesso de Usuário 66

6 Conclusões **73**

6.1 Contribuições 74

6.2 Trabalhos Futuros 76

Referências Bibliográficas **78**

Apêndices **86**

Apêndice A – Documentação Complementar **87**

A.1 *Identity Federation Module* 88

A.2 *Attribute Module* 91

A.3 *Access Control Module* 93

A.4 *VO Manager* 94

Apêndice B – Documentação XML **96**

B.1 ACROSS Global 96

B.1.1 XML da Configuração da OV 96

B.1.2 XML da Configuração para Classificação em Níveis 97

B.2 Configuração das Políticas 99

B.2.1 XML para Configuração de Recursos 99

B.2.2 XML para Configuração de Políticas 100

Apêndice C – Documentação da API **102**

C.1 *VO Manager* 102

C.1.1 *Resource* 103

C.2 *SP Module* 103

C.3 *Identity Federation Module* 104

C.3.1 *Credential Translation Module* 104

C.4 *Attribute Module* 104

C.4.1 *Attribute Aggregation Module* 104

C.4.2 *Attribute Provider Module* 105

C.5 *Access Control Module* 105

C.5.1 *Score* 105

C.5.2 *User Level* 107

C.5.3 *Global/Local Policy* 107

Lista de Figuras

1.1 Cenários de exemplo para ambientes de recursos distribuídos. 2

1.2 Organização Virtual formada por membros das instituições A e B para acesso a um ambiente de experimentação. 3

2.1 Exemplo de uma organização virtual e seus membros. 10

2.2 A FIBRENet com o FIBRE e suas ilhas [FIBRE 2016]. 11

2.3 Visão geral do *framework* de controle de acesso do padrão X.812 - ISO/IEC 10181-3 [ISO 2011]. 15

2.4 Componentes XACML [Moses 2005]. 16

4.1 Visão geral da arquitetura de componentes do ACROSS. 29

4.2 Framework ACROSS com seus módulos. 30

4.3 Diagrama de atividades para o usuário da OV usando ACROSS para se autenticar. 33

4.4 Diagrama de atividades para o usuário da OV usando ACROSS para alocar recursos. 34

4.5 Casos de uso para usuário da OV. 35

4.6 Autenticação do usuário utilizando o Módulo de Federação de Identidade. 36

4.7 Diagrama de sequência de acesso do usuário com atributos adicionais. . 37

4.8 Diagrama de casos de uso para a configuração do *Access Control Module*. 38

4.9 Configuração da pontuação para atributos. 39

4.10 Configuração dos níveis. 43

4.11 Configuração de recursos. 45

4.12 Diagrama de sequência para a configuração de políticas. 47

4.13 Diagrama de classes do ACROSS. 49

5.1 Cenário do caso de uso do ACROSS para FI no projeto FIBRE. 56

5.2 Portal FIBRE com acesso ao LDAP federado (a) ou CAFe Expresso (b). . 57

5.3 Passos de autenticação utilizando a CAFe. 58

5.4 Resultado da autenticação e agregação de atributos para o usuário FIBRE. 59

5.5 Credencial do usuário. 60

5.6 Usuário autenticado no portal do FIBRE com listagem de recursos (*slices*). 61

5.7 Cenário de validação do ACROSS. 63

5.8 Acesso ao ACROSS Global. 64

5.9 Resumo da configuração do ACROSS Global. 65

5.10 Interface de gerenciamento do ACROSS Global. 65

5.11 Resumo da configuração de uma instituição (Inst1). 66

5.12 Autenticação federada pela CAFe Expresso. 67

5.13 *Screenshots* da interface do usuário. 67

5.14 Interface do usuário após autenticação bem sucedida. 68

5.15 Interface do usuário. 69

5.16 *Screenshot* da interface do usuário no ACROSS após requisitar lista de recursos. 70

5.17 Usuário requisitando recursos para diferentes instituições. 70

5.18 Lista de recursos reservados. 71

5.19 Usuário requisitando recursos de diferentes instituições. 71

5.20 Mensagem de retorno informando que a requisição do usuário foi negada. 72

A.1 Diagrama de classes do ACROSS. 88

A.2 Casos de uso para administrador da OV. 89

A.3 Diagrama de sequência para instalação e configuração do *Identity Federation Module*. 90

A.4 Diagrama de casos de uso para o *Attribute Module*. 91

A.5 Instalação e configuração do Provedor de Atributos (*Attribute Provider*). 92

A.6 Instalação e configuração do Agregador de Atributos (*Attribute Aggregation*)). 93

A.7 Diagrama de casos de uso para a instalação do suporte ao *Access Control Module*. 94

A.8 Diagrama de sequência para a instalação do suporte ao *Access Control Module*. 94

A.9 Diagrama de casos de uso para a instalação do portal de administração para a OV e local. 94

A.10 Diagrama de sequência para a instalação do portal de administração para a OV e local. 95

Lista de Tabelas

4.1 Comparação entre X.812, ABAC e ACROSS. 32
4.2 Descrição das variáveis do Algoritmo 1. 41
4.3 Um exemplo de pontuação para atributos do usuário 43
4.4 Um exemplo de definição de nível baseado em pontuação. 44
4.5 Descrição das variáveis do Algoritmo 2. 45
4.6 Políticas de controle de acesso em níveis para máquinas virtuais. 46
4.7 Descrição dos símbolos 52
4.8 Comparação entre as propostas apresentadas 53

Capítulo 1

Introdução

A área de redes de computadores e sistemas distribuídos está em constante evolução, e um de seus principais motivos se dá à característica de cada vez mais dispositivos serem integrados ao sistema. Um dos aspectos mais interessantes desse cenário é a possibilidade de fazer uso da natureza distribuída desses sistemas para melhorar a eficiência e eficácia das soluções. Cenários que merecem destaque são os de *grid computing* (computação em grade) [Gagliardi 2004, Romberg 2002, Meirosu et al. 2005, IBM 2016, Crawford et al. 2003], *cloud computing* (computação em nuvem) [Amazon 2016, Microsoft 2016], e o de ambientes, ou *testbeds*, para experimentação em redes [Berman et al. 2014, Köpsel and Woesner 2011, Vandenberghe et al. 2013, Sallent et al. 2012].

A computação em grade é um exemplo maduro do uso da rede em larga escala para o desenvolvimento de projetos e soluções, tanto na área científica [Gagliardi 2004, Romberg 2002, Meirosu et al. 2005], conhecido como *e-Science* [Gomes et al. 2015], como na área comercial [IBM 2016, Crawford et al. 2003]. A computação em nuvem mostra uma nova maneira de se usar os ambientes distribuídos de redes, concentrando-se mais no lado do cliente como objetivo final, onde naturalmente encontram-se mais aplicações comerciais do que em computação em grade [Amazon 2016, Microsoft 2016]. Os ambientes de experimentação, ou *testbeds*, são voltados para o desenvolvimento e validação de novas soluções, e no caso das redes de computadores existem *testbeds* como aqueles voltados para o desenvolvimento da Internet do Futuro (FI – *Future Internet*) [Sallent et al. 2012]. Esse cenário é, também, um excelente exemplo da utilização de um ambiente distribuído para o desenvolvimento das redes de computadores, e apresenta, como nos dois cenários citados anteriormente, características como a heterogeneidade dos recursos e a necessidade da integração entre eles. A Figura 1.1 ilustra os três cenários citados.

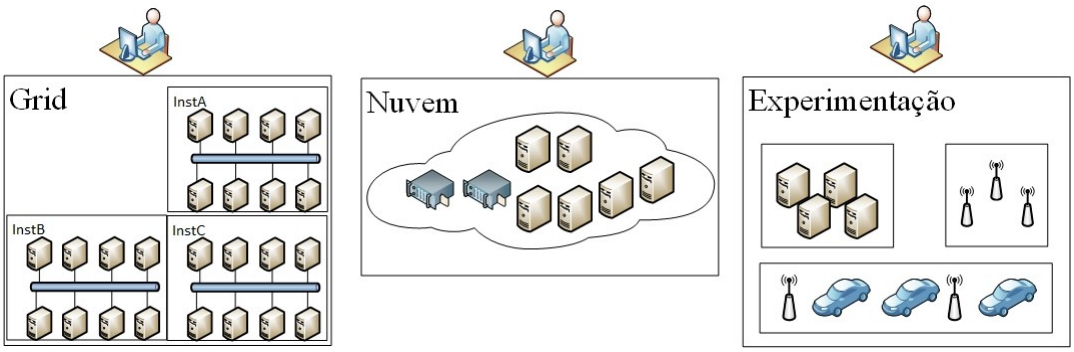

Figura 1.1: Cenários de exemplo para ambientes de recursos distribuídos.

Em todos esses ambientes, a gestão de identidade e acesso (IAM - *Identity and Access Management*) [Sturrus and Kulikova 2016, Gaedke et al. 2005], que inclui as funcionalidades de autenticação e autorização, se mostra essencial, sendo responsável por propor diretivas desde o acesso do usuário até a utilização dos recursos [Gunter et al. 2011].

Para entender os benefícios e a importância da IAM nos cenários citados, destaca-se o que naturalmente acontece na utilização de um ambiente com recursos distribuídos e compartilhados. Comumente, são criadas relações entre grupos envolvendo diferentes parceiros de diferentes instituições, os quais acessam um mesmo conjunto de recursos distribuídos, de forma compartilhada, recursos esses mantidos por diferentes administradores. Esses ambientes promovem a possibilidade de se criar as chamadas Organizações Virtuais (OVs – *Virtual Organizations*) [Foster et al. 2001a, Camarinha-Matos and Afsarmanesh 2005]. As OVs são como uma nova organização formada por alguns dos membros (e não todos) das entidades parceiras. As OVs podem ser exemplificadas pela Figura 1.2, onde tem-se uma OV formada por membros das instituições A e B para acesso a um ambiente heterogêneo e distribuído de recursos para experimentação. Nesse exemplo, será necessário tratar as questões de IAM desde a identificação e forma de acesso dos membros da OV até o controle sobre o que é possível acessar no ambiente de experimentação, respeitando as premissas, necessidades, características e políticas de controle que regem a OV e as instituições participantes.

Tanto a área de pesquisa [Gaedke et al. 2005, Thomas and Chandrasekaran 2016, Sharma et al. 2016] quanto comercial [ama 2016, goo 2016] têm apresentado enorme interesse nos processos de IAM para gerência de acesso aos recursos distribuídos. Os processos de IAM compreendem a gestão de identidade e o controle de acesso, onde a gestão (ou gerência) de identidade (IdM – *Identity Management*) pode ser entendida como o conjunto de processos e tecnologias usados para garantir a identidade de uma

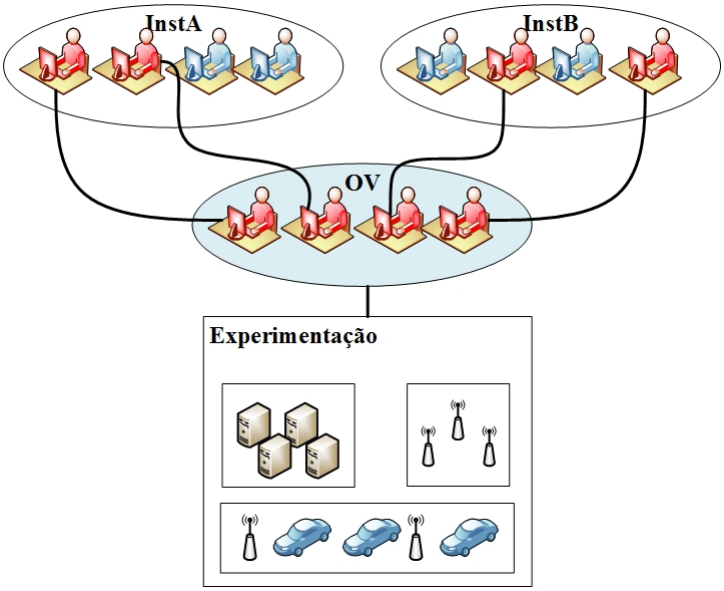

Figura 1.2: Organização Virtual formada por membros das instituições A e B para acesso a um ambiente de experimentação.

entidade, a qualidade das informações de identidade (identificadores, credenciais e atributos) e utilizar essas garantias em procedimentos de autenticação, autorização e auditoria (AAA – *Authentication, Authorization and Accounting*) [ITU-T 2009, Wangham et al. 2010]. Procedimentos de *autenticação* dizem respeito à confirmação da identidade de uma entidade, isto é, à verificação de que uma entidade é quem ela afirma ser. Já mecanismos de *autorização* definem os direitos de acesso a recursos associados a uma identidade já determinada. Procedimentos de autorização são utilizados tanto para descrever esses direitos de acesso como para garantir que eles sejam cumpridos, um conceito intimamente ligado ao controle de acesso. Finalmente, a *auditoria* se refere aos processos que permitem verificar o funcionamento correto dos procedimentos anteriores. Ainda nesse contexto, as federações de identidade têm ganho grande relevância [Torres et al. 2013, Pérez-Méndez et al. 2014]. Um de seus principais benefícios é a utilização de uma única credencial para cada usuário. Criando uma identidade somente em uma instituição a qual o usuário está vinculado, é possível que a autenticação do usuário seja feita por qualquer serviço sempre consultando essa base de origem do usuário. Assim, as federações de identidade solucionam o problema da autenticação única para diferentes serviços. Uma vez que são pautadas no respeito a políticas, padrões e práticas comuns a todos os membros da federação, as federações de identidades são consideradas fontes confiáveis para a autenticação.

1.1 Motivação

Tomando como principal motivação para o estudo dos ambientes de OV, esta tese destaca o cenário de FI. Seus principais requisitos como uma OV de recursos distribuídos foram identificados como:

- **ambiente heterogêneo:** uma vez que essa OV tem como característica a utilização de recursos de tipos diferentes, como nós sem fio, máquinas virtuais, equipamentos de roteamento definidos por software etc;
- **OV formada por diversas entidades parceiras:** já que o ambiente de experimentação deve ser geograficamente distribuído, e isso só é possível por meio da interconexão entre diversos parceiros, que incorporam seus recursos ao ambiente;
- **alocação dinâmica dos recursos distribuídos:** com os recursos sendo utilizados hora por certo usuário, hora por outro, devendo ser escalonado esse processo de alocação conforme os pedidos dos participantes da OV;
- **usuário pode assumir funções diferentes em instantes diferentes de tempo:** já que em um dado momento o mesmo usuário pode assumir a função de gerente de um projeto, e em outro apenas um integrante de um outro grupo de trabalho;
- **autonomia para políticas de controle de acesso:** onde cada entidade parceira que oferece acesso a seus recursos, pode descrever seu controle de acesso de forma independente sobre seus recursos oferecidos, respeitando a política geral da OV.

Ao propor um *framework* para A&A para esse ambiente, tais requisitos deverão ser atendidos. Em linhas gerais, esse *framework* deve suportar a heterogeneidade dos recursos do ambiente, suportar as políticas distribuídas e hierárquicas da OV, permitir a reserva de recursos de forma dinâmica e independente a partir do resgate de atributos do usuário de forma dinâmica para a visão atual do ambiente. Deve ainda, permitir que as políticas de controle de acesso a serem descritas sejam independentes entre as entidades que oferecem recursos, mas que respeitem as políticas gerais da OV. Os pontos destacados, são identificados em diversos ambientes de OV, e podem também ser aplicados a eles, mas fica clara a motivação do desenvolvimento deste trabalho no escopo do cenário de um ambiente de FI.

Alguns pontos importantes com relação à proposição da gerência de políticas de controle de acesso para OVs foram identificados como motivação para essa tese. Nos cenários de OVs, há tanto o interesse na criação de políticas de controle de acesso particulares a um certo ambiente, como a necessidade de manutenção de políticas comuns a todas as instituições parceiras. Principalmente por apresentarem características distribuídas e independentes, a administração desses ambientes e de suas políticas para controle de acesso se torna um grande desafio, despertando interesse e atenção no desenvolvimento de soluções.

Já como forma de solucionar a confiabilidade da autenticação, que é, em termos gerais, a verificação da identidade do usuário, as federações de identidade podem ser adotadas como essa fonte confiável de credenciais e atributos. Nesse caso, as federações de identidade validam e liberam os atributos do usuário que deseja utilizar o ambiente da OV. Porém, em geral, é necessário adicionar atributos ao usuário específicos ao ambiente. Essa adição visa manter um dos princípios da federação de identidade, que é a padronização dos atributos disponíveis dentro da federação de identidade, mas permitir a inclusão de novos atributos particulares à OV.

Na literatura, encontram-se diversas propostas para facilitar a gestão de identidade e o controle de acesso em ambientes de recursos distribuídos, especialmente nos cenários já comentados de computação em grade [Foster et al. 2001b, Spence et al. 2006, Barton et al. 2006], computação em nuvem [Zhu et al. 2014, Wen et al. 2012, Dhungana et al. 2013] e *testbeds* de FI [Sallent et al. 2012, one 2013, Köpsel and Woesner 2011, Szegedi et al. 2009, Friedman and Gavras 2011]. Entretanto, essas soluções apresentam características específicas a esses cenários particulares, o que torna o processo de adaptação a cenários diferentes dispendioso. Um fator complicador é que, em geral, administradores de uma OV costumam ter grande experiência na gerência do recurso a ser compartilhado, mas pouca experiência na área de gestão de identidades. Como conseqüência, por exemplo, um *framework* para computação em grade é raramente utilizado em um cenário de computação em nuvem, e se utilizado, demanda um grande esforço de adaptação ao novo ambiente [Lee et al. 2014, Garcia et al. 2013]. Muitas das vezes, até mesmo a utilização em cenários com características semelhantes às quais o *framework* foi desenvolvido é custosa [Vinicius G. Pinheiro 2015].

Sendo assim, percebe-se a necessidade de propor uma solução atual, integrada, que facilite a adesão por meio das OVs e que responda, principalmente, as seguintes questões:

- como armazenar, de forma confiável, atributos dos usuários específicos a OV e que não são mantidos pela federação de identidade.
- como realizar a transposição das credenciais do usuário advindo da Federação de Identidade para o ambiente de recursos da OV
- como realizar o controle de acesso no ambiente distribuído da OV.

1.2 Objetivos

Este trabalho tem o objetivo de criar uma solução de IAM para ambientes de recursos compartilhados, que aborde conceitos de gestão de identidade e controle de acesso e que possa ser facilmente adotada ou integrada a uma OV ou entidade participante dessa OV. Essa solução deve ser facilmente adaptável às características do ambiente de recursos distribuídos ao qual se aplica, oferecendo mecanismos de configuração adaptáveis ao cenário aplicado. Além disso, deverá satisfazer os seguintes requisitos baseados em [Afsarmanesh and Camarinha-Matos 2005]:

- ser integrada a uma solução de autenticação com suporte à federação de identidade e não exigir nenhuma mudança nessa federação;
- realizar autorização baseada em atributos e políticas;
- permitir a transposição de credenciais para diferentes tipos de ambiente de recursos;
- respeitar a privacidade dos usuários e
- ser aplicável a diferentes federações, ou ambientes, de recursos distribuídos.

Este trabalho tem como objetivo a especificação e implementação de um *framework*, que ofereça as funcionalidades de autenticação e autorização para diferentes OVs.

1.3 Contribuições

Esta tese propõe, documenta, implementa e valida um *framework* genérico chamado ACROSS (**A**ttribute-based access **C**ont**RO**l and di**S**tributed policie**S**) com o objetivo de facilitar a gestão de identidade e acesso em ambientes de OVs [Silva et al.

2015a, Silva et al. 2015b]. O ACROSS é um *framework* para controle de acesso distribuído baseado em políticas e atributos para VOs que respeita o padrão de controle de acesso "X.812 | ISO / IEC 10181 – 3 : 1996" [ISO 2011]. O ACROSS pode ser utilizado em diferentes cenários de recursos distribuídos.

O ACROSS trata tanto da autenticação como da autorização em OVs, possibilitando a criação de políticas locais e globais para uso de seus recursos. O ingresso de qualquer entidade em uma OV é facilitado a partir de uma federação de identidade, sendo a federação de identidade responsável por manter os principais atributos de um usuário de forma confiável. Além disso, a partir da utilização do ACROSS, esta OV pode armazenar atributos adicionais, ou particulares ao seu ambiente. Através de uma solução de agregação de atributos proposta em [Silva et al. 2015a], o ACROSS possibilita a geração de uma identidade de usuário completa, sem a necessidade de qualquer alteração dos dados do usuário na federação de identidade, nem duplicação de atributos dessa federação de identidade na organização virtual. O *framework* auxilia na instalação e configuração de todos os serviços de uma forma simplificada para quem deseja gerir uma OV, criando uma abstração de detalhes da gestão de identidade que não interessam diretamente ao gerente da OV. Todas essas funcionalidades do *framework* proposto são detalhadas através da modelagem em *Unified Modeling Language* (UML) [Alanen and Porres 2003].

No ACROSS, é proposta uma nova forma de abstração e generalização de pontuação de atributos para classificação de usuários em níveis, que são usados na definição de políticas de acesso. Com este conceito, o ACROSS implementa o *Attribute-Based Access Control* (ABAC) [Hu et al. 2013] e esconde aspectos mais custosos de configuração e gerenciamento dos atributos dos usuários, facilitando a gerência de acesso em OVs.

O ACROSS se mostra uma solução de autenticação e autorização flexível e adaptável para OV, suportando diversos conceitos de gestão de identidade e recursos. Entre os benefícios da utilização do ACROSS, destacam-se as contribuições científicas:

- criação de um *framework* de A&A com controle de acesso baseado em atributos e políticas distribuídas para organizações virtuais;
- criação de um agregador de atributos que respeita a privacidade do usuário através da utilização de um atributo opaco único de criação flexível;
- proposta de um novo modelo de classificação de usuário em níveis a partir de seus atributos;

O *framework* ACROSS também oferece as seguintes facilidades:

- desenvolvimento baseado na utilização de padrões, uma vez que se baseia em X.812 [ISO 2011], ABAC [Hu et al. 2013], XACML (*eXtensible Access Control Markup Language*) [Moses 2005], SAML (*Security Assertion Markup Language*) [OASIS 2005], SOAP (*Simple Object Access Protocol*) [W3C 2016], UML, etc, que facilitam a extensão e integração com outras soluções;
- autenticação baseada em SAML, herdando todos os benefícios das federações de identidade e facilitando a entrada de novos membros;
- especificação e implementação de forma modularizada, onde cada módulo tem características específicas, facilitando sua extensão e uso;
- instalação e configuração facilitada através de guias (*wizards*);
- facilidade de gestão das instituições, dos atributos de usuário, políticas e recursos, através de uma *interface web*.

1.4 Organização do texto

O trabalho está organizado em um total de sete capítulos. No Capítulo 2, o embasamento teórico sobre os principais conceitos e padrões de autenticação e autorização utilizados em organizações virtuais são apresentados. No Capítulo 3, são discutidos os trabalhos relacionados. Já o Capítulo 4 apresenta o *framework* ACROSS, proposta desta tese e compara-o aos principais trabalhos relacionados. O Capítulo 5 apresenta a implementação e uso do ACROSS em uma OV. A tese é concluída com a apresentação das considerações finais no Capítulo 6, realçando as contribuições e apontando trabalhos futuros.

Capítulo 2

Embasamento Teórico

Gestão de Identidade (GId) ou *Identity Management* (IdM) é o conjunto de processos e tecnologias utilizadas para garantir a identidade de uma entidade. A IdM garante a qualidade das informações da identidade, tais como identificadores, credenciais e atributos e usa essas garantias para realizar autenticação, autorização e processos de contabilidade/auditoria [Jensen 2012]. A gestão de identidade e o controle de acesso, compreendidos no conceito de *Identity and Access Management* (IAM) [Sturrus and Kulikova 2016, Gaedke et al. 2005], são necessários para as organizações virtuais criadas entre entidades parceiras que desejam compartilhar recursos. Sendo assim, este capítulo apresenta alguns conceitos básicos e tecnologias relacionadas com a gestão de identidade e acesso. Além disso, são descritas soluções para federações de identidade, protocolos de autenticação e mecanismos de controle de acesso.

2.1 Organizações Virtuais

Como introduzido no início dessa tese, as OVs – em especial aquelas da área acadêmica – têm requisitos especiais para o compartilhamento de recursos, levando em conta a necessidade de colaboração e integração criada por projetos, intercâmbios, cursos à distância, e outras atividades. Há alguns anos atrás, o conceito de identidade federada foi criado [Chadwick 2009, Rountree 2012], permitindo acesso a recursos para todos os membros de uma instituição ou toda a comunidade. Por exemplo, usando uma federação de identidade, um usuário pode acessar repositórios de periódicos sem a necessidade de duplicar informações e bancos de dados. Mesmo que as federações de identidade tenham se tornado uma importante solução para a confiança na integração do ambiente acadêmico, elas não são suficientes para o controle de acesso. Em certos casos, os recursos distribuídos devem ser compartilhados apenas por determi-

nados membros de diferentes instituições, como os participantes de um projeto inter-institucional ou estudantes do mesmo curso em diferentes instituições. Estes grupos são chamados de organizações virtuais ou OVs [Foster et al. 2001a, Camarinha-Matos and Afsarmanesh 2005].

Para ilustrar uma OV, a Figura 2.1 mostra duas instituições A e B, onde apenas alguns membros dessas instituições participam de um projeto, e juntos formam uma (OV).

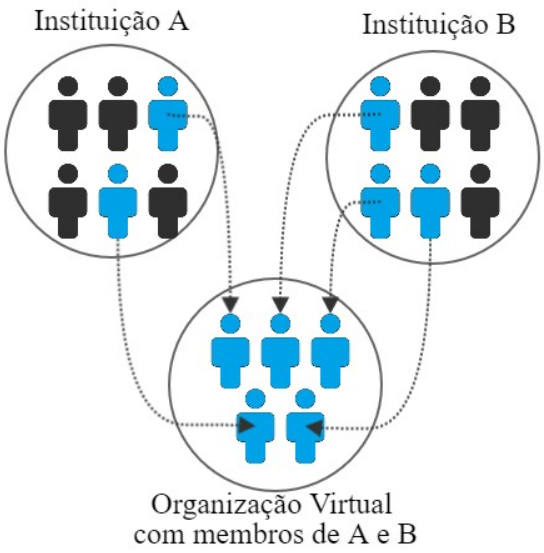

Figura 2.1: Exemplo de uma organização virtual e seus membros.

Um exemplo prático de uma OV é o projeto FIBRE (*Future Internet Testbeds Experimentation Between Brazil and Europe*) [Sallent et al. 2012], criado para testar e validar protocolos e soluções para a Internet do Futuro (*Future Internet* – FI). O FIBRE é composto por várias instituições em diferentes países. Seu principal objetivo é a interligação de ambientes de teste geograficamente distribuídos, chamados de *testbeds*, a fim de apoiar a experimentação em redes de nova geração, criando um ambiente experimental em larga escala utilizando uma vasta gama de dispositivos compartilhados. Nesse ambiente, um pesquisador pode realizar seu experimento reservando recursos de diferentes *testbeds* em diferentes instituições. Portanto, é necessário suportar o compartilhamento de recursos, de modo que um pesquisador possa ver quais recursos estão disponíveis em todos os *testbeds*, bem como serem aplicadas soluções de autenticação e autorização (A&A) para o uso desses recursos. Nesse caso, um usuário autenticado tem permissão para acessar os recursos de qualquer outro *testbed* na federação de recursos, desde que o usuário atenda às políticas de controle de acesso local de cada um

dos ambientes (instituições/*testbeds*) envolvidos e também às políticas globais da OV.

Para ilustrar o cenário atual do FIBRE, a Figura 2.2 apresenta as ilhas que compõem sua rede de experimentação, a FIBRENet. São 11 as ilhas (instituições) parceiras em 7 estados, envolvidas no projeto, tendo ainda a instalação e configuração em andamento de mais 4 novas ilhas em 3 diferentes estados do Brasil. Das 11 ilhas em funcionamento, 1 está localizada fora do Brasil, e é representada pela AMPATH. É possível ver, como indicado pelo ícones e legenda da figura, os equipamentos de cada ilha, sejam de experimentação sem fio (*Wireless Nodes*), ou recursos de redes definidas por software (*OpenFlow resources*).

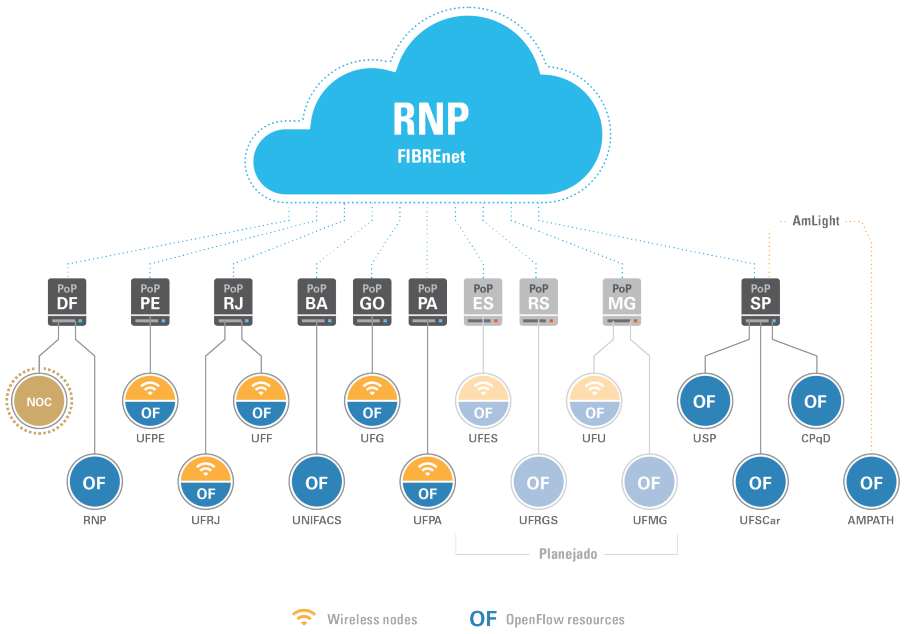

Figura 2.2: A FIBRENet com o FIBRE e suas ilhas [FIBRE 2016].

É importante destacar o ambiente do Projeto FIBRE, pois ele foi motivador para iniciar as pesquisas desta tese de doutorado. A partir desse cenário é que foi possível expandir os estudos para outros casos de uso.

Desta forma, outro exemplo de organização virtual é o de um ambiente de computação em nuvem. Pode-se modelar uma nuvem como uma OV e aplicar conceitos de gestão de identidade tanto sob a ótica de usuários finais quanto de pesquisadores. Por exemplo, se arquivos são compartilhados por grupos de usuários de diferentes do-

mínios, tem-se um problema de gestão de identidade no ambiente da OV. Já quando os recursos da nuvem são utilizados para realizar experimentação de novas soluções, surge um novo cenário, mais focado na área de pesquisa. Com infraestrutura semelhante à dos *testbeds* de redes, as nuvens experimentais consistem em ambientes distribuídos onde as instituições ou organizações parceiras, distribuídas geograficamente, criam uma OV [Berman et al. 2014] e necessitam de todos os requisitos de IAM.

No entanto, apesar da forte demanda por organizações virtuais, a criação de tais organizações é complexa. Além de questões práticas específicas de cada OV, sobre a gestão de recursos ou cooperação, como por exemplo, é exigido que um gerente da OV tenha grandes conhecimentos sobre gestão de identidade para implantar formas de autenticação e autorização e atender a todos os requisitos da OV e de cada instituição participante. Em geral, os administradores só têm conhecimento sobre um ambiente específico e não têm tempo ou interesse em se envolver nos mecanismos de gestão de identidades complexos. Desta forma, um framework genérico para IAM em OVs, tal como o ACROSS proposto nesta tese, pode simplificar a incorporação de gestão de identidade nesses ambientes.

2.2 Federações de Identidade e SAML

Como dito no Capítulo 1, as federações de identidade têm ganho grande relevância nos últimos anos [Torres et al. 2013, Pérez-Méndez et al. 2014]. Elas solucionam o problema da autenticação única para diferentes serviços. Baseada em padrões de comunicação e trocas de mensagens, como o *Security Assertion Markup Language* (SAML) [OASIS 2005], um dos principais motivadores para sua adoção é a utilização em seu escopo de uma única credencial para cada usuário. Com uma identidade criada somente na instituição de origem do usuário, a federação de identidade permite que a autenticação do usuário seja feita por qualquer serviço sempre usando a base de dados de usuários da instituição. Uma federação de identidade permite também que seja aplicado o conceito de *Single Sign-On* (SSO), onde o usuário, uma vez autenticado e autorizado, não necessita autenticar-se novamente em outros serviços.

Diversos são os exemplos de federações de identidade, tal como a Comunidade Acadêmica Federada (CAFe) [RNP 016s]. A CAFe é a federação de identidade acadêmica brasileira. Integrada à federação de federações de identidade européiaeuropeia, eduGAIN [edu 2016a], ajuda o usuário brasileiro a ter acesso aos serviços fornecidos

pelas instituições europeias. Além disso, auxilia na autenticação de diversos serviços, como o eduroam (*Education Roaming*) [edu 2016b], que permite acesso a rede sem fio por meio da autenticação em sua instituição de origem em qualquer parte do mundo.

O SAML [OASIS 2005] é um padrão aberto para federações de identidades para autenticação e autorização *web*. Ele apresenta um conjunto de especificações para definir uma infraestrutura para troca de informações de forma segura entre seus parceiros (*e.g.* instituições). Nesse padrão, são definidos os papéis das entidades, mensagens e protocolos de transporte suportados. Baseado no formato *EXtensible Markup Language* (XML) [W3C 2008], o SAML tem como foco a requisição e recuperação de atributos (formato por pares nome-valor), que poderão ser utilizados na aplicação que os requisita.

Desenvolvido pela *Organization for the Advancement of Structured Information Standards* (OASIS) [OASIS 2005], o SAML atualmente se encontra na versão 2.0 e as duas principais entidades que compõem o ambiente de Autenticação e Autorização (A&A) são:

- o Provedor de Identidade (IdP – *Identity Provider*), responsável por armazenar e fornecer informações sobre os usuários e sua autenticação;
- o Provedor de Serviço (SP – *Service Provider*), responsável por oferecer um ou mais serviços ou recursos.

Um exemplo de solução baseada em SAML é o Shibboleth [Scavo 2005], que implementa o SAML e permite que aplicações *web* aproveitem as facilidades oferecidas pelo modelo de federação de identidade, suportando, por exemplo, o conceito de SSO (*Single Sign-On*) [Bhargav-Spantzel et al. 2007] .

2.3 Controle de Acesso

Mecanismos de controle de acesso são responsáveis por associar direitos de acesso a recursos a um usuário. Portanto, uma federação de identidade é responsável por ajudar nessa etapa, validando a credencial do usuário e sendo a entrada para os procedimentos de autorização.

Mecanismos de controle de acesso são fundamentais para proteger um recurso contra o acesso não autorizado. Especificamente, uma política de controle de acesso define

as condições em que o acesso aos recursos pode ser concedido e para quem. Com a crescente complexidade de sistemas de computação, os métodos de controle de acesso evoluíram a partir do *Discretionary Access Control* (DAC) [Jin et al. 2012], *Mandatory Access Control* (MAC) [Sandhu 1993], e *Role-Based Access Control* (RBAC) [Ferraiolo et al. 2001], até o *Attribute-Based Access Control* (ABAC) [Hu et al. 2013].

O DAC tem muito uso em sistemas operacionais, onde o usuário está associado a certas permissões sobre um dado arquivo. Um exemplo prático de sistema operacional que utiliza DAC são os baseados em UNIX. Já o MAC se apresenta mais complexo e flexível que o DAC. Ambos pensam, em sua implementação original, em recursos como arquivos. Para o DAC, o usuário tem controle sobre as permissões de seus arquivos. Já para o MAC, o sistema pode impor o controle de acesso sobre arquivos. Diferentemente do DAC, que funciona como uma matriz de correlação, o MAC é como um grafo acíclico, que utiliza hierarquia e regras associadas aos nós para a aplicação do controle de acesso.

O RBAC é um dos mais utilizados mecanismos de controle de acesso. Ele cria uma forma simples de associação entre o usuário e um papel no sistema. Este papel é usado para a realização do controle de acesso. Um papel em RBAC pode ser, por exemplo, o cargo ao qual o usuário está associado, ou um grupo de usuários. Com a evolução dos mecanismos de controle de acesso, foi proposto o ABAC, que, resumidamente, permite que os atributos, tanto do usuário quanto do recurso e do ambiente como um todo, sejam avaliados para a tomada de decisão sobre a permissão ou não de concessão de acesso do usuário ao recurso. É o método mais flexível atualmente e permite uma enorme combinação de estratégias na aplicação do controle de acesso.

Como complemento aos métodos simples de autorização suportados pelo SAML, tem-se um padrão muito mais completo para esse fim, o *eXtensible Access Control Markup Language* (XACML) [Moses 2005]. O padrão XACML é uma linguagem para declaração de políticas de segurança definida pela OASIS, baseada em XML e que implementa o ISO/IEC 10181-3 [ISO 2011] e a recomendação ITU X.812 para um *framework* de autorização. A Figura 2.3 mostra uma visão geral do modelo X.812. Na figura, vê-se dois componentes essenciais: o *Access Control Enforcement Functions* (AEF) e o *Access Control Decision Functions* (ADF). A ideia principal do *framework* é que o AEF assegure que todas as requisições de acesso passem pelo ADF, onde o ADF decide, baseado em um conjunto de regras e políticas, se o acesso é permitido ou não. Em XACML, esses componentes são renomeados: AEF é equivalente ao *Policy Enforcement Point* (PEP) e

o ADF é equivalente ao *Policy Decision Point* (PDP). Em XACML, existem dois componentes a mais, o *Policy Administration Point* (PAP) e o *Policy Information Point* (PIP), como pode ser visto na Figura 2.4. O PAP é o repositório de políticas de acesso, e permite ao administrador adicionar ou editar políticas. O PIP é responsável por armazenar atributos adicionais para recursos (*resources*), usuários (*subjects*)[1], e o ambiente como um todo.

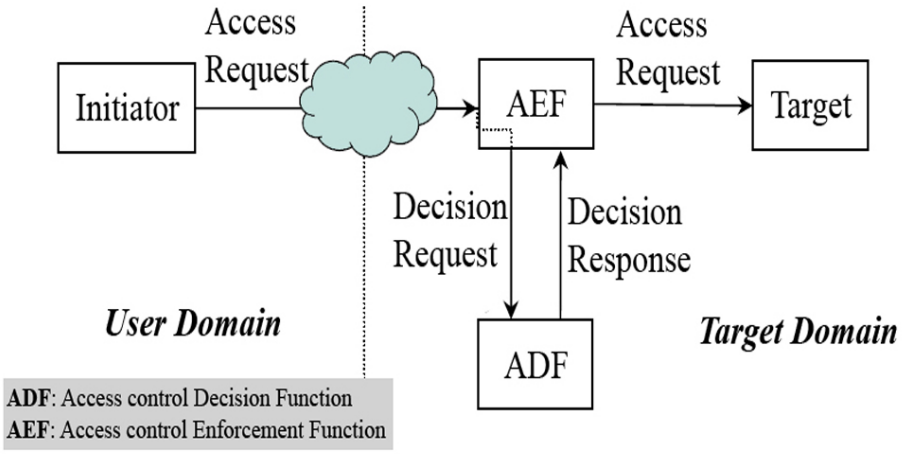

Figura 2.3: Visão geral do *framework* de controle de acesso do padrão X.812 - ISO/IEC 10181-3 [ISO 2011].

A Figura 2.4 mostra o processo de requisição de acesso em um ambiente XACML, onde as etapas são enumeradas de 1 a 10. Na etapa 1, tem-se a requisição de acesso ao PEP; o PEP por sua vez encaminha a requisição ao PDP para decidir se o acesso será permitido ou negado na etapa 2; as etapas 3 e 4 referem-se à consulta sobre qual política será aplicada a essa requisição; as etapas 5 e 7 são requisições a atributos adicionais feitas ao PIP, onde as etapas 6a, 6b e 6c representam o processo de registrar esses atributos; na etapa 8, o recurso informa seu status, e o PDP decide sobre o acesso requisitado, encaminhando a resposta ao PEP; para finalizar, o PEP verifica as chamadas *obligations*, que são referentes a, por exemplo, realizar o registro (*accounting*) da requisição para posterior auditoria, e conclui a verificação XACML.

[1] *Subjects* são citados como exemplo de usuários, porém representam qualquer entidade que envie uma solicitação de acesso e esteja, assim, subordinada aos mecanismos de controle de acesso.

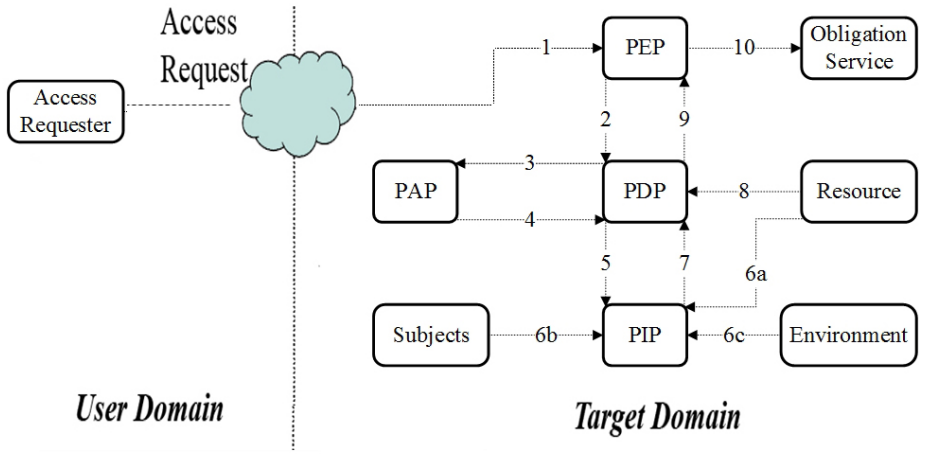

Figura 2.4: Componentes XACML [Moses 2005].

Além de ser uma linguagem para políticas de controle de acesso, a XACML define também um formato para mensagens de pedido e resposta. No padrão XACML, são definidos os formatos de troca de pedidos e respostas entre as entidades PDP e PEP, onde o último efetua realmente o processamento e aplicação da política. Já para garantir a interoperabilidade entre aplicações, o XACML faz uso de uma camada de abstração entre o ambiente de aplicação e o chamado Contexto XACML. Um Contexto XACML nada mais é que a definição XML representando canonicamente as entradas e saídas do PDP.

No XACML, o suporte à hierarquia de políticas é bem definido. O XACML prevê alguns algoritmos de combinação de regras (*rule-combining algorithm*) que facilitam a aplicação das regras de forma hierarquica. Algoritmos de combinação de regras ou políticas podem definir, por exemplo, que se uma das regras não for atendida, todo o acesso é negado (*deny-overrides*) ou vice-versa (*allow-overrides*). Os algoritmos estão bem definidos em [Moses 2005], assim como o formato das mensagens XACML a serem trocadas e a descrição para políticas. Vale ressaltar que o XACML prevê quatro tipos de retorno ao comparar atributos às políticas criadas. São eles: (1) *Permit*: aplicação da política de acesso permitido; (2) *Deny*: aplicação da política de acesso negado; (3) *Indeterminate*: quando um erro ocorre ou um valor requerido de um atributo não é encontrado e não é possível aplicar política alguma e (4) *Not Applicable*: a requisição

não pode ser respondida ou não é suportada pelo serviço.

O XACML pode ser visto como uma implementação para o ABAC [Hu et al. 2013], que como comentado neste capítulo, é um mecanismo de controle de acesso onde o usuário (*subject*) que deseja realizar operações sobre recursos é avaliado com base em seus atributos, nos atributos do recurso, nos atributos do ambiente e em um conjunto de políticas que são especificadas em termos desses atributos e condições. O ABAC provê mecanismos flexíveis para se aplicar políticas distribuídas e controle de acesso baseada nos atributos do usuário e dos recursos.

O *framework* ACROSS, proposto neste trabalho, é baseado em X.812, implementa o padrão XACML e suporta os mecanismos de controle de acesso RBAC e ABAC.

2.4 Resumo dos Conceitos Básicos

Como conclusão deste capítulo, são descritos conceitos básicos que envolvem as funcionalidades de um *framework* completo de A&A para OVs. São eles:

- a autenticação por federação de identidade;
- a agregação de atributos;
- a transposição de credenciais;
- o controle de acesso aos recursos oferecidos pela OV.

2.4.1 Autenticação

A autenticação foi abordada neste capítulo, levando em consideração as federações de identidade e o padrão SAML. É importante deixar claro que, neste trabalho, o método de autenticação herda os benefícios e facilidades de se usar ambientes federados de identidade, onde atributos de usuário são fornecidos da forma mais padronizada e confiável possível. Porém, é relevante dizer que, para a proposta desta tese, quaisquer que sejam as técnicas de autenticação, ela pode ser considerada, desde que seja suficientemente confiável e emita os atributos mínimos do usuário para a utilização da OV.

2.4.2 Agregação de Atributos

A agregação de atributos é utilizada como forma de facilitar a utilização de quaisquer que sejam os métodos de autenticação, uma vez que permite que apenas um conjunto mínimo de atributos do usuário seja exigido para a utilização da OV.

Como exemplo, pode-se imaginar um usuário autenticado por meio de uma federação de identidade, onde apenas o atributo email desse usuário foi liberado para a OV, mas deveria existir também um atributo que informasse qual o cargo daquele usuário. A agregação de atributos surge exatamente para facilitar essa escassez de atributos, já que permite que atributos complementares para o usuário sejam inseridos em um provedor de atributos adicionais específico para aquela OV. E, sendo assim, é facilmente compreendido que o atributo que informa o cargo daquele usuário pode ser criado no âmbito da OV e agregado aos atributos da federação de identidade para então formar a identidade completa do usuário.

2.4.3 Transposição de Credenciais

O conceito de transposição de credenciais é utilizado pelo fato de que credenciais para acesso aos recursos da OV, em geral, são específicas. Quando se diz específicas, refere-se aos padrões e formatos de credenciais. Como exemplo, no ambiente de experimentação para FI do FIBRE utilizam-se credenciais em formato de certificados X.509 aplicados ao ambiente SFA (*Slice-Based Facility Architecture*) [Peterson et al. 2008] de gerência de recursos distribuídos. Sendo assim, a partir dos atributos do usuário pode-se realizar a transposição/transformação/geração do certificado X.509 para o usuário. A partir de sua identidade completa (que são seus atributos advindos da federação de identidade – ou qualquer outra fonte de autenticação – mais os atributos adicionais fornecidos pela agregação de atributos) essa transposição é, então, possível.

Lembrando que, nesta tese, a transposição de credenciais é facilitada a partir da disponibilização dos atributos do usuário para uma ferramenta externa, que deverá consultar, resgatar, esses atributos e gerar, então a credencial específica do ambiente.

2.4.4 Controle de Acesso

O controle de acesso foi apresentado neste capítulo, comentando os principais métodos existentes, e enfatizando o modelo RBAC e ABAC. Nesta tese os atributos da

identidade completa do usuário são usados para realizar o controle de acesso. A proposta de controle de acesso desta tese é baseada em ABAC.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, são apresentados os principais trabalhos relacionados com foco em gerência de recursos distribuídos em OVs. Com a intenção de deixar mais clara a classificação dos trabalhos relacionados, foram criadas duas seções. Cada seção corresponde a uma categoria, sendo a primeira a de *frameworks* de autenticação e controle de acesso. Esta seção corresponde a soluções consideradas completas, e que deverão ser diretamente comparadas, ainda nessa tese, ao ACROSS. Já a outra seção é a de gestão de identidade e acesso, onde são apresentadas soluções relacionadas de escopo reduzido ou voltados especificamente para a gestão de identidade ou controle de acesso.

3.1 Frameworks de Autenticação e Controle de Acesso

O *Virtual Organization Membership Service* (VOMS) [Alfieri et al. 2003] é um *framework* para autenticação e controle de acesso desenvolvido, inicialmente, para o contexto de *grid computing* (computação em grade). O VOMS é baseado em papéis, que por sua vez são expressos por grupos e subgrupos. A ideia é ter uma hierarquia representada como um grafo em que haja uma raiz e, abaixo dela, os vértices sejam grupos e subgrupos. Nesse grafo, as arestas orientadas representam a herança de regras dos níveis superiores, como em um grafo acíclico.

No VOMS, o usuário está associado a um grupo e é representado por papéis (*roles*) e recursos (*capabilities*), onde um papel está intimamente relacionado ao usuário naquele grupo, ou seja, ao papel que ele possui dentro daquele grupo. Esse papel é utilizado para a tomada de decisão. Já os recursos são descritos de forma textual livre, indicando características particulares. Essas características poderão ser utilizadas no controle de acesso local, por exemplo, utilizando listas de controle de acesso (ACL - *Access Control List*).

As credenciais VOMS são certificados X.509, como:

/Fred/replicator/optimisation/Role=Admin,

onde o usuário da OV é *Fred*, que pertencente ao subgrupo *optimisation* do grupo *replicator*, tendo como papel *Admin*.

O VOMS se baseia no *Globus Toolkit's Grid Security Infrastructure* (GSI) [Foster and Kesselman 1996, Foster et al. 1998]. O GSI é uma infraestrutura de A&A, na qual é criado um certificado do tipo X.509 *proxy* [Farrell and Housley 2002] como credencial do usuário. Esse certificado tem um tempo de vida curto, o que o torna seguro por não ser persistente. Atributos de usuário, como grupos aos quais ele pertence, em geral, são armazenados em um diretório LDAP e, periodicamente, tais atributos são transformados em credenciais da OV.

No VOMS, existe uma política global que se refere à política da OV, que é uma política geral de autorização. Essa política global avalia se o usuário tem credenciais válidas ou pertence a um certo grupo. Além disso, existe também a política local, que é controlada pelo *Resource Provider* (RP), que é o responsável por oferecer o recurso em si. Para utilizar o VOMS, o usuário deve, primeiramente, obter credenciais globais junto à OV. Na sequência, o usuário deve apresentar suas credenciais ao RP (local) junto com a autorização pré-concedida pela OV (global). A ideia é que o usuário, mesmo sendo autorizado pela OV, possa ter seu acesso restringido localmente.

Outra proposta de *framework* é o *Community Authorization Service* (CAS) [Pearlman et al. 2002], que tem como ideia principal a inserção de um certo nível de controle de acesso global a recursos locais. Assim, parte das políticas locais podem ser definidas pelo administrador da OV, a fim de permitir que este administrador aplique políticas gerais da OV no ambiente distribuído. Essa gerência de controle de acesso é realizada pelo administrador da OV através do servidor CAS, que funciona de forma intermediária entre o requisitante do acesso e o recurso em si. Seu modelo de autorização é baseado em papéis, associando aos papeis o tipo de acesso a cada tipo de recurso.

É possível traçar um paralelo entre política global e local no CAS, onde o controle de acesso de nível global existe quando o usuário apresenta suas credenciais e é verificado junto à base de políticas globais da OV, associando a ele uma certa permissão conforme seu papel na OV. Já a política local pode ser vista no acesso, quando a instituição dona do recurso aplica uma política restritiva conforme o papel do usuário. O CAS é integrado ao *middleware* para *grids* computacionais *Globus*, GSI,[Foster and

Kesselman 1996], assim como o VOMS, e utiliza seus serviços para acesso aos recursos, permitindo gerar credenciais no formato de certificados X.509 *proxy* e delegar credenciais.

A diferença básica entre o CAS e o VOMS é que, no VOMS, o usuário pertence a um grupo que, por sua vez, é mapeado em um direito somente no domínio do recurso (na instituição, por exemplo). Já o CAS envia os direitos baseados nos papéis diretamente através do acesso à OV. De fato, as diferenças entre VOMS e CAS são sutis, pois ambos se apoiam no GSI.

Já o *PrivilEge and Role Management Infrastructure Standard* (PERMIS) [Chadwick et al. 2003] é uma *Privilege Management Infrastructure* (PMI) baseada em X.509. Análoga a uma Infraestrutura de Chave Pública (ICP) ou *Public Key Infrastructure* (PKI) [ITU-T 2012], o PMI apresenta uma *Attribute Authority* (AA), responsável por emitir certificados de atributos X.509 (*Attribute Certificates* – ACs) [Farrell and Housley 2002] para os usuários. Além disso, uma Autoridade Certificadora (AC) é responsável por armazenar a ligação entre a credencial do usuário, seu *Distinguished Name* (DN) e os atributos de privilégios do usuário. A raiz de confiança do PMI é chamada de *Source of Authority* (SOA).

Diferentemente do CAS, o PERMIS utiliza o ABAC a partir dos certificados de atributos do usuário. Sabemos que o PERMIS é uma solução madura, difundida e bem estruturada no padrão X.509, assim como o VOMS. Porém, essas soluções acabam por se tornar um tanto quanto engessadas na estrutura de grade computacional, principalmente pelo tipo de credencial suportada, além de ambas terem como herança o foco na integração do *Globus Toolkit*, o que pode dificultar a adoção destes *frameworks* ou exigir um grande esforço de desenvolvimento para a integração com outros ambientes de autenticação ou que utilizem credenciais de acesso ao ambiente de recursos diferentes de certificados X.509.

A arquitetura do PERMIS apresenta as entidades necessárias ao ABAC e respeita o *framework* genérico ISO/IEC 10181-3.

O Akenti [Lab 2016] é similar ao PERMIS na infraestrutura de autorização, já que ambos têm uma entidade principal responsável por realizar o controle de acesso do usuário aos recursos. Essa entidade principal, que serve como entrada na utilização do *framework* e controle de acesso no Akenti é chamado de *resource gateway* e desempenha o papel do PEP nessa infraestrutura.

Desenvolvido originalmente com o foco em recursos *web*, o Akenti foi posteriormente estendido para suportar outros tipos de recursos, como os utilizados em *grids* e OVs. Assim como no PERMIS, as políticas são escritas em XML. Tanto o PERMIS quanto o Akenti utilizam diretórios LDAP para armazenar suas políticas e credenciais de usuário como certificados.

A autorização no Akenti é realizada em nível de recurso. Isso quer dizer que a identidade do usuário é verificada no momento do acesso ao recurso. Essa infraestrutura permite políticas distribuídas, onde entidades conhecidas como *stakeholders* são responsáveis por verificar hierarquicamente a permissão de acesso de forma independente. Desta forma, pode-se afirmar que existe um suporte aos conceitos de políticas globais e locais.

Ao final dessa seção, vê-se o quanto essas soluções têm correlação. Apesar de propostas, a princípio, para ambientes com características particulares de recursos distribuídos, todas elas apresentam semelhantes. E, como todas elas apresentam soluções desde a autenticação à autorização, colaboram para o estudo do estado-da-arte dessa tese e deverão ser comparadas, ao final desse trabalho, ao ACROSS.

3.2 Gestão de Identidade e Controle de Acesso

Nesta seção, serão vistos os principais trabalhos que apresentam correlação com o tema de estudo dessa tese, mas realizam contribuições mais pontuais. Os trabalhos dessa seção são importantes, também, para se entender como esforços na área de gestão de identidade e acesso ainda é um tema relevante.

O mesmo acrônimo CAS apresentado anteriormente se aplica ao *Central Authentication Service* [Apereo 2016], que vamos chamar neste texto de CAS2. Porém, com relação ao CAS (*Community Authorization Service*), que tem um foco maior na autorização e é um *framework* mais voltado para OVs, o CAS2 é um protocolo SSO para *web* [1]. Criado primeiramente na Universidade de Yale, o CAS2 passou por vários grupos até ser atualmente suportado pela organização Apereo. É importante notar que faz mais sentido comparar o CAS2 ao SAML, por ser um protocolo que prevê, como o SAML, autenticação e autorização. O CAS2 apresenta, basicamente, as mesmas entidades do SAML, o SP e IdP. Suporta diversos métodos de autenticação como LDAP, RADIUS, banco de dados etc. É difícil não comparar a implementação CAS2 à implementação

[1] https://github.com/apereo/cas/tree/master/cas-server-documentation

do Shibboleth, por conta dos objetivos e datas de criação. O SAML é proposto por volta de 2001 e implementado pela Internet2 como Shibboleth a partir de 2003, e o CAS2 é proposto no mesmo ano que SAML, em 2001, e tem sua versão 1.0 por volta também de 2003.

Com foco não só na área acadêmica e de pesquisa, o CAS2 tem diversos *plugins* para autenticação e autorização, como integração ao OpenID [Rae et al. 2012] e ao oAuth [Hardt 2012]. Para se integrar o CAS2 com uma federação SAML, como é o caso da CAFe, seria necessário "shibboletizar" o SP CAS2, porém é necessário levar em consideração a versão de cada solução e suas compatibilidades. Uma vez que CAS2 também trabalha com o conceito de atributos, é possível, a partir dos atributos liberados, realizar autorização, porém no nível da aplicação que gerencia o serviço oferecido pela OV. A autorização não é internamente suportada pelo protocolo do sistema, como no VOMS ou no PERMIS. Esse tipo de autorização é suportado também pela implementação Shibboleth, e, basicamente, permite ou nega o acesso a certo usuário no ato da sua autenticação com base em algum atributo.

Vale apresentar também o *System for Cross-domain Identity Management* (SCIM), como iniciativa de gestão de identidades. O SCIM é um padrão proposto pelas RFCs 7643 e 7644 [Hunt et al. 2015b, Hunt et al. 2015a], criado para simplificar a gerência de usuários, grupos e recursos, com foco na autorização. O SCIM define um *schema*, responsável pela definição dos atributos suportados e um protocolo de gestão de identidade baseado em funcionalidades HTTP *Representational State Transfer* (REST) [Fielding 2000]. Há uma semelhança entre o LDAP e o SCIM, porém o SCIM não tem a intenção de ser um substituto ao LDAP. Com o SCIM, o que se propõe é uma forma mais simples de gerenciar grupos e usuários em ambientes distribuídos, podendo ser aplicado a ambientes específicos, como nuvem [Hunt et al. 2015b], e em soluções como *Virtual Organisation Orthogonal Technology* (VOOT) [VOOT API/Internet2 2016] da GÉANT e *Grouper* [gro 2016] da Internet2.

O VOOT é uma extensão do SCIM, que objetiva ser um protocolo simples para acesso somente leitura de dados sobre usuários em domínios compartilhados (*cross-domain*). Dados, neste caso, são, em geral, relacionados às informações sobre a composição de membros de um grupo ou se um usuário faz parte de certo grupo.

O Grouper [gro 2016] é uma solução da Internet2. O Grouper, assim como VOOT ou qualquer outra solução baseada em SCIM, não é considerado um *framework* para OVs, mas sim, um *middleware* que trabalha como uma solução complementar para

autorização em OVs. Foi desenvolvido para gerência de acesso em ambientes amplamente distribuídos e genéricos, os quais são comuns no ambiente acadêmico e de pesquisa. É muito utilizado no controle de acesso a aplicações, listas, wikis, calendários etc. Esse sistema trabalha com o conceito de grupos, papéis e permissões específicas para tomada de decisão no controle de acesso.

Outros trabalhos relacionados merecem ser destacados como esforços pontuais para a criação de soluções para OVs, sendo divididos em três categorias. A primeira compreende soluções para a integração com federações de identidade; a segunda é a agregação de atributos e a transposição de credenciais; por último, o controle de acesso.

Na integração de federações de identidade, tem-se diversos trabalhos na literatura. Os mais utilizados e que visam, principalmente, a autenticação, são as implementações do SAML, destacando o Shibboleth e simpleSAMLphp [sim 2016]. Nessa tese, foca-se na implementação do SAML2 pelo Shibboleth/Internet2.

Já para o passo da transposição de credenciais, destacam-se os ambientes de federação em grade. Geralmente, nesses ambientes, é necessário acessar os recursos utilizando credenciais em formatos específicos. Esse formato, em geral, é de certificados X.509, o que exige que as credenciais (ou atributos) de origem, sejam traduzidas (ou transpostas).

Pode-se destacar diversos trabalhos dentro do cenário de grades computacionais para a transposição de credenciais integrados ao ambiente SAML, como os apoiados pelo JISC (*Joint Information Systems Committee*)[2]: ShibGrid, SHEBANGS (*Shibboleth Enabled Bridge to Access the National Grid Service*) e SARoNGS (*Shibboleth Access to Resources on the NGS*); e projetos apoiados pela NSF (*National Science Foundation*)[3], como o ShibGrid, CILogon e o go.teragrid [Spence et al. 2006, Barton et al. 2006].

Dentre os trabalhos relacionados em grade, destacam-se os trabalhos ShibGrid [Spence et al. 2006] e SHEBANGS [Wang et al. 2009]. Desenvolvidos especialmente para o *UK National Grid Service* (NGS), sua infraestrutura é baseada em certificados X.509 e em uma base para armazenamento desses certificados. Essa base funciona como um *proxy* na comunicação entre o usuário e o serviço e é chamada de MyProxy [Basney et al. 2005]. No caso do SHEBANGS, o usuário, após se autenticar via Shibboleth, tem seus atributos recebidos pelo serviço de transposição de credenciais, que gera e armazena o certificado X.509 no MyProxy. Já o ShibGrid utiliza o MyProxy para

[2]http://www.jisc.ac.uk/
[3]http://www.nsf.gov/

associar uma identidade advinda do Shibboleth com um certificado X.509 já existente, a partir do *Distinguished Name* (DN) do certificado do usuário, realizando assim uma associação da credencial do usuário Shibboleth com um certificado X.509 a cada autenticação.

Merece destaque também o Serviço para Transposição de Credenciais de Autenticação Federada (STCFed) [de Mello et al. 2009], uma iniciativa brasileira que realiza a transposição de credenciais SAML, advindas da federação CAFe, para aplicações ou ambientes que não sejam baseados em *web*. Ou seja, torna-se possível que uma aplicação, por exemplo, *standalone* (*desktop*), passe a ter os benefícios de fazer parte da federação CAFe através da transposição de credenciais. Nessa tese é desenvolvida uma interface para facilitar a transposição de credenciais para quaisquer tipos.

Para a agregação de atributos, há diversos modelos propostos por [Chadwick and Inman 2009]. Uma solução prática da utilização de agregação de atributos em IdPs diversos a partir de um identificador único está na federação acadêmica da NREN suíça, SWITCH[4], que utiliza o conceito de *Simple Aggregation* do Shibboleth[5] para a obtenção de múltiplos atributos através de múltiplas fontes. Nessa tese, é criado um método de agregação de atributos baseado em provedores de atributos adicionais, os quais guardam atributos específicos do ambiente da OV.

[4]https://www.switch.ch/aai/support/tools/vo-concept/
[5]https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAttributeResolver

Capítulo 4

Proposta do framework ACROSS

O *framework* ACROSS (**A**ttribute-based access **C**ont**RO**l and di**S**tributed policie**S**) foi projetado para ajudar instituições a criarem, ou participarem, de uma OV que oferece acesso a recursos distribuídos. Fornecendo soluções para instalação e configuração da gestão de identidade em OVs, o ACROSS oferece assistentes para configuração dos serviços básicos para criação ou participação em uma OV. Esses assistentes incluem a instalação e configuração de um módulo no provedor de serviços para ingresso a uma federação de identidade baseada em Shibboleth, configuração e informações para controle de acesso e políticas distribuídas, suporte a mecanismos de controle de acesso como RBAC e ABAC e autenticação baseada em SAML com a finalidade de facilitar a integração com a federação de identidade.

Outro conceito importante considerado na proposta do ACROSS é o de provedores de atributos adicionais e de um agregador de atributos. Utilizando esse conceito, o ACROSS é capaz de respeitar as premissas de uma federação de identidade, onde atributos do usuário são armazenados apenas em sua instituição de origem, e não devem ser modificados. Com provedores de atributos adicionais, o ACROSS permite a criação, gerência e armazenamento de atributos adicionais, específicos da OV. Além disso, o *framework* prevê a transposição de credenciais, facilitando a geração de credenciais específicas ao ambiente de recursos distribuídos em questão.

O ACROSS suporta, ainda, políticas de controle de acesso distribuídas e hierárquicas. Com políticas distribuídas, é possível dividir a gerência de políticas em dois níveis: um global e outro local. O nível local é responsável pelas políticas específicas de uma instituição e o nível global é responsável pelas políticas globais da OV.

Para facilitar a utilização do ACROSS, foram desenvolvidos assistentes de instalação e configuração para todos seus módulos. A partir de seus assistentes, é possível,

além de instalar e configurar todos os serviços, configurar e gerenciar políticas globais, locais, recursos etc. através de uma interface web. Essa interface permite a troca de mensagens com os componentes do ACROSS (e.g. agregador de atributos) utilizando *Simple Object Access Protocol* (SOAP) [W3C 2016] e arquivos XML, facilitando a interoperabilidade e respeitando padrões internacionais. Essas facilidades são vitais ao sucesso das OVs, especialmente porque, em geral, gerentes de OVs nem sempre têm conhecimentos avançados em IAM para realizar as etapas de instalação e configuração manualmente.

Uma visão geral dos componentes da arquitetura do ACROSS podem ser vistos na Figura 4.1. No topo, tem-se a Federação de Identidade (*Identity Federation*), responsável pela autenticação dos usuários. No meio, estão os componentes do ACROSS, onde cada módulo possui funções específicas. O *Identity Federation Module* é responsável pela comunicação com a Federação de Identidade. O *Attribute Module* é responsável por gerenciar, requisitar e recuperar atributos específicos da OV a partir dos provedores de atributos. O *Access Control Module* é responsável pela autorização. O *SP Module* é responsável pela comunicação, ou troca de mensagens, com todos os módulos do ACROSS. Por último, o *VO Manager* oferece uma interface web com facilidades de gerência para o administrador da OV. Anterior a todos, responsável ainda pela preparação do ambiente, instalando todos os serviços, pacotes, dependências de software e configurando esses parâmetros básicos, e iniciais, está o *ACROSS Wizards*. Na parte inferior da figura, está representada a Federação de Recursos (*Resource Federation*), responsável pelo compartilhamento dos recursos das instituições.

4.1 Detalhamento da proposta

Cada módulo do ACROSS representa uma camada de serviço, como mostra a Figura 4.2.

O *Identity Federation Module* tem dois submódulos específicos, o *SAML Module* e o *Credential Translation Module*. O *SAML Module* é responsável pela comunicação e troca de dados com a federação de identidades. Usando este módulo, é possível autenticar um usuário e receber todos os atributos advindos de sua instituição de origem.

O *Credential Translation Module* [Silva et al. 2014a] é responsável por gerar credenciais de tipos específicos para um dado ambiente de OV. A credencial é composta por dados obtidos a partir do *SAML Module* e do *Attribute Provider Module*. A tradução de

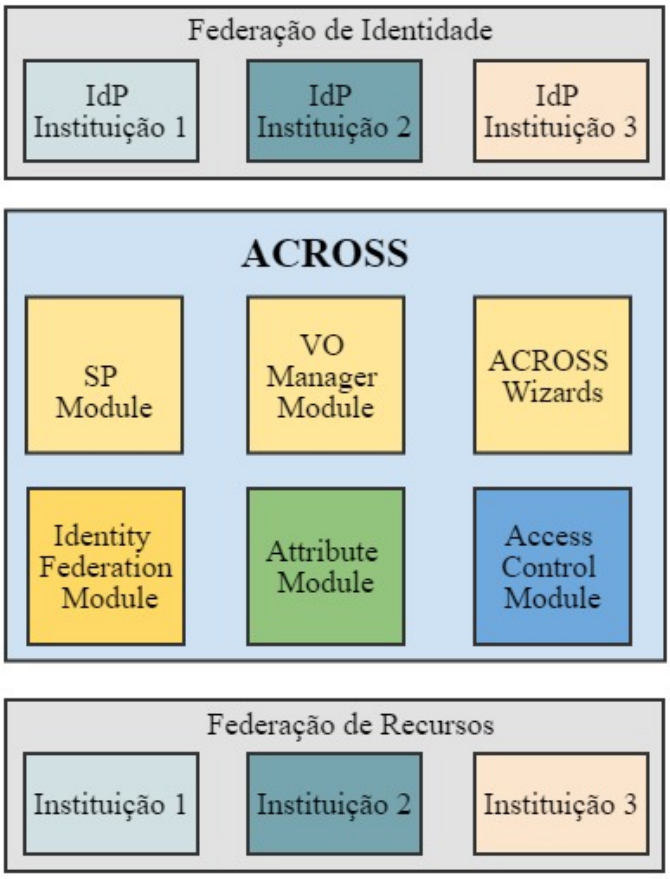

Figura 4.1: Visão geral da arquitetura de componentes do ACROSS.

credenciais é um requisito necessário para a integração de ambientes de organizações virtuais. Cada OV define credenciais de acesso específicas ao seu ambiente, por exemplo, podendo ser um certificado X.509 [ITU-T 2012]. Em geral, a credencial obtida da federação de identidade não é compatível com a requisitada pela OV. Sendo assim, é necessário utilizar o *Credential Translation Module*, que, além de acessar os atributos recebidos pelo *SAML Module*, requisita os atributos adicionais e específicos da OV e gera a credencial de acesso. Note que a implementação deste módulo é específica a cada OV. Uma vez que o ACROSS disponibiliza todos os atributos do usuário agregados, um consumidor desses atributos, disponibilizados em arquivos XML, pode ser desenvolvido, ou integrado, para a geração da credencial específica do ambiente da OV.

O *Attribute Module* é responsável por armazenar os atributos adicionais de cada usuário de uma OV. Usualmente, a federação de identidade não provê todos os atributos necessários a todas as funcionalidades da OV, necessitando complementar esses atributos com o cadastro local de atributos específicos para aquele ambiente. Um

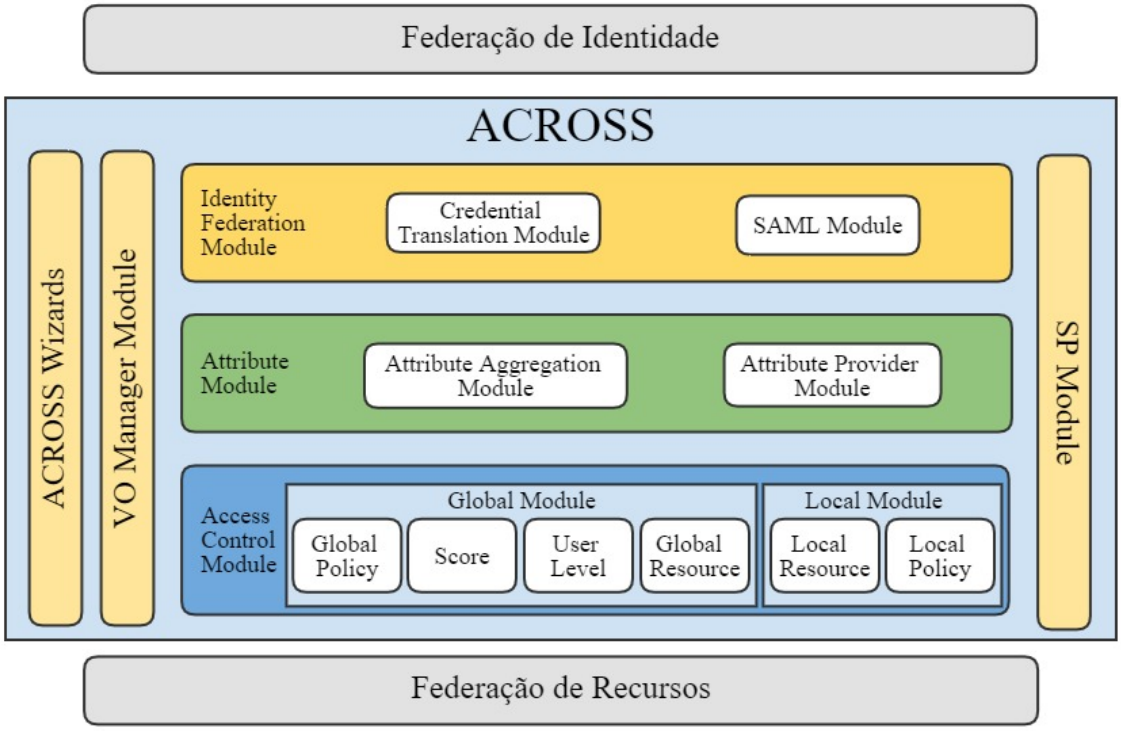

Figura 4.2: Framework ACROSS com seus módulos.

exemplo prático da necessidade de atributos adicionais é o do serviço de periódicos da CAPES [1], acessado através da autenticação de um usuário pela federação de identidade CAFe. Como a CAPES necessita de atributos adicionais não disponíveis nos provedores de identidade da CAFe, é feito um novo cadastro, que guarda, além da vinculação da CAFe com essa conta local, novos atributos específicos do ambiente. Porém, em geral, isso acontece duplicando-se atributos do usuário vindos da federação de identidade. Se o ACROSS fosse utilizado neste caso, somente os atributos adicionais seriam armazenados no provedor de atributos da CAPES, através do *Attribute Module* do ACROSS.

O *Attribute Module* é relativamente independente dos outros módulos se forem considerados seus aspectos de instalação e configuração. O *Attribute Module* tem dois submódulos, o *Attribute Provider* e o *Attribute Aggregation Module*.

O *Attribute Provider* é quem armazena os atributos adicionais, especificados pelo administrador da VO. Este módulo cria uma árvore de diretórios LDAP [Group 2006] para armazenar esses atributos. O *Attribute Aggregation Module* recebe a requisição de agregação do SP Module e então requisita os atributos adicionais do usuário ao *Attribute Provider*. Uma vez que o *SP Module* tenha todos os atributos do usuário, é possível realizar a transposição de credenciais, e iniciar efetivamente o controle de

[1] http://periodicos.capes.gov.br/

acesso do usuário e a etapa de requisição de recursos.

O terceiro módulo principal é o *Access Control Module*. Ele é baseado no modelo ABAC e permite que a OV ou o administrador da instituição estabeleça políticas globais ou locais para controle de acesso. Seu modelo é distribuído, o que significa que é executado em dois níveis: nível global da OV (*Global Module*) e nível local (Módulo Local – *Local Module*) a cada instituição que integra a OV. O *Access Control Module* tem quatro submódulos no *Global Module* e dois outros submódulos no Módulo Local.

No *Global Module*, existem submódulos para classificar usuários em níveis de acesso, que são os submódulos de Pontuação (*Score*) e Nível de Usuário (*User Level*). Além disso, ACROSS tem o submódulo *Global Policy*, responsável por armazenar as políticas de controle de acesso, e o submódulo *Global Resource*, usado para armazenar os tipos de recursos suportados pela OV.

No *Local Module*, presente em cada instituição da OV, há os submódulos *Local Policy* e *Local Resource*. Similar aos módulos de *Global Policy*, o *Local Policy* é responsável por armazenar as políticas locais criadas por um gestor local da instituição. O submódulo *Local Resource* registra os recursos em uma instituição, de acordo com os tipos de recursos suportados e registrados no submódulo *Global Resource*.

Como proposto em [Silva et al. 2015a], o administrador da OV é capaz de configurar pesos e pontos aos atributos de usuários (*Score Module*) e essa pontuação será responsável por associar um usuário a um certo nível (*User Level Module*), que será utilizado para o controle de acesso, considerando as políticas globais e locais.

Uma vez que cada ambiente de recursos distribuídos tem sua forma particular de requisitar e reservar recursos (usualmente através de arquivos baseados em XML), é necessário adaptar o ACROSS a cada ambiente. Por isso, o *Access Control Module* possui uma interface que facilita o desenvolvimento de *wrappers* para a comunicação entre ACROSS e o ambiente da federação de recursos específicos.

O ACROSS tem dois outros módulos, o *SP Module* e *VO Manager*. O *SP Module* é responsável pela interação entre o usuário e os outros módulos do ACROSS. Essa interação é realizada pelo usuário através da interface de gerência ou acesso do ambiente da OV, onde, por sua vez, o *SP Module* se integra. O *VO Manager Module* provê uma interface de gerenciamento para o administrador da OV. Essa gerência é relativa às configurações de parâmetros de atributos, políticas etc. toda a parte de administração posterior à instalação de todos os módulos através dessa interface. O último módulo é

o *ACROSS-Wizards*, responsável pela instalação de serviços e dependências de *software* e configuração dos parâmetros iniciais do ACROSS.

Uma comparação entre X.812 - ISO/IEC 10181-3, ABAC, e os módulos do ACROSS é detalhada na Tabela 4.1. Esta tabela ajuda a identificar quais entidades são responsáveis pela execução de cada função nesses *frameworks*. Em todos os três modelos, estão presentes entidades referentes à tomada de decisão (*Decision Point*) e a entidade de aplicação dessa ação (Action Point), mas entidades referentes à gestão de políticas e atributos aparecem apenas no ABAC e no ACROSS.

Tabela 4.1: Comparação entre X.812, ABAC e ACROSS.

	Action Point	**Decision Point**	**Policy Manager**	**Attribute Repository**
X.812	AEF	ADF	–	–
ABAC	PEP	PDP	PAP	PIP
ACROSS	SP Module	Access Control Module	VO Manager Module	Attribute Module

Como mencionado, o ABAC é um mecanismo de controle de acesso formalizado pelo NIST em [Hu et al. 2013] e o ACROSS é baseado nesse guia e no X812 - ISO/IEC 10181-3. Assim, o ACROSS herda as vantagens dessas iniciativas e estende suas funcionalidades com conceitos como provedores de atributos, agregação de atributos, controle de acesso baseado em políticas, níveis de controle de acesso e generalização de recursos.

Para mostrar as etapas e funcionalidades de um pedido de autenticação de um usuário para acesso aos recursos de uma OV, um diagrama de atividades é apresentado na Figura 4.3, considerando os módulos do ACROSS, a federação de identidade e a federação de recursos.

As etapas mostradas no diagrama da Figura 4.3 são, primeiramente, a autenticação do usuário, onde o usuário requisita acesso e é encaminhado ao seu IdP (*Identity Provider*) na federação de identidade através do *Identity Federation Module* com seu *SAML Module*. Então, as credenciais do usuário são requisitadas por seu IdP. Depois disso, suas credenciais são validadas e seus atributos recebidos pelo *SAML Module*, que é responsável por disponibilizá-los ao SP Module. Então, o *SP Module* requisita os atributos adicionais ao *Attribute Aggregation Module*. O *Attribute Aggregation Module* recebe os

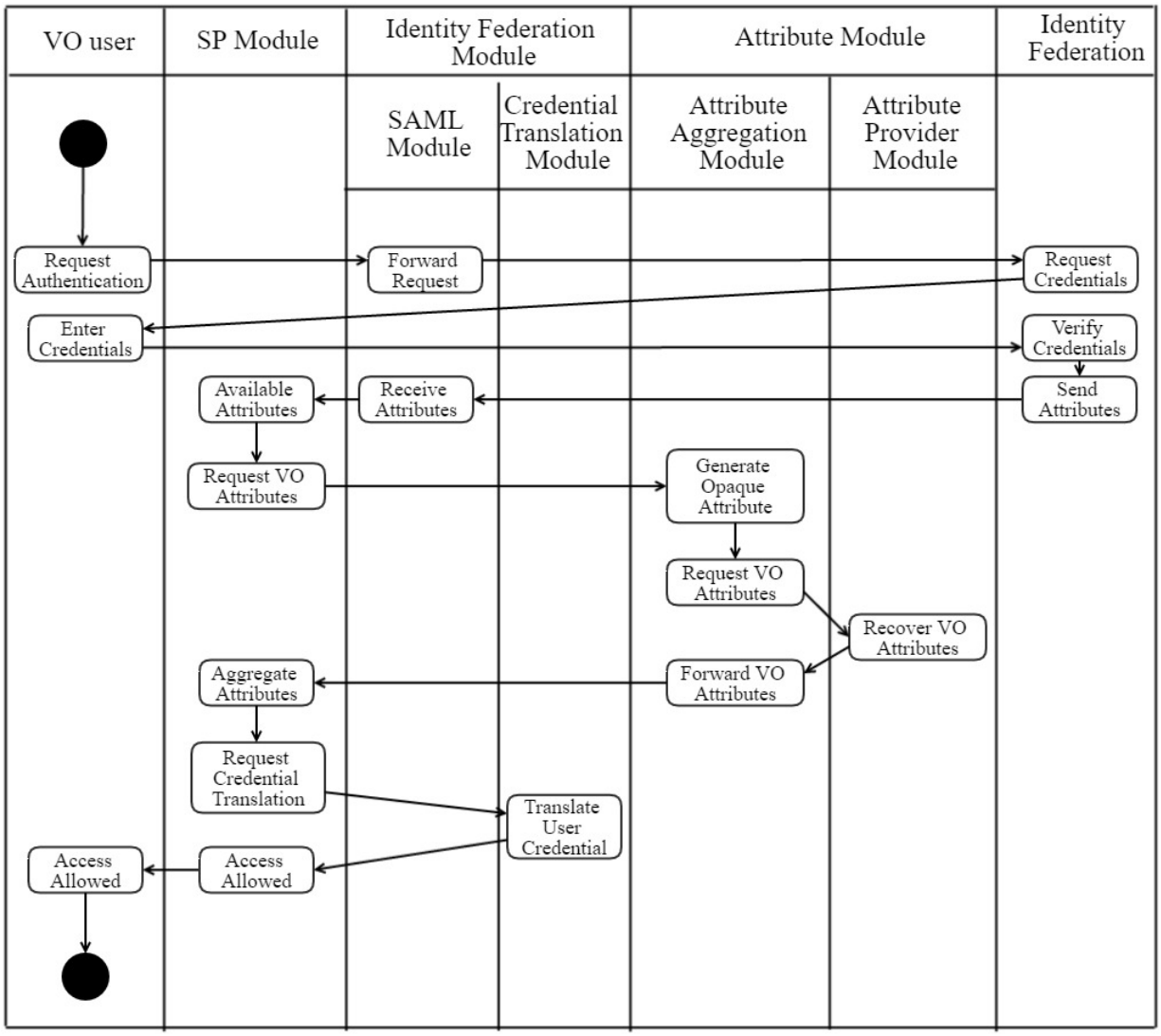

Figura 4.3: Diagrama de atividades para o usuário da OV usando ACROSS para se autenticar.

atributos do usuário necessários para gerar seu identificador (ou atributo opaco) para busca nos provedores de atributos adicionais. O atributo opaco é quem identifica o usuário e é utilizado na recuperação de atributos do usuário pelo *Attribute Provider*. Depois de requisitar e receber os atributos adicionais do usuário, o *Attribute Aggregation Module* encaminha esses atributos ao *SP Module* que os agrega com os atributos do usuário que vieram da federação de identidade. Após esses passos, O *SP Module* é capaz de disponibilizar todos os atributos do usuário para o *Credential Translation Module*, responsável por gerar a credencial específica para a OV. O *SP Module* guarda todos os atributos do usuário temporariamente em uma sessão SAML.

Um outro diagrama de atividades, agora para as etapas de autorização, requisição

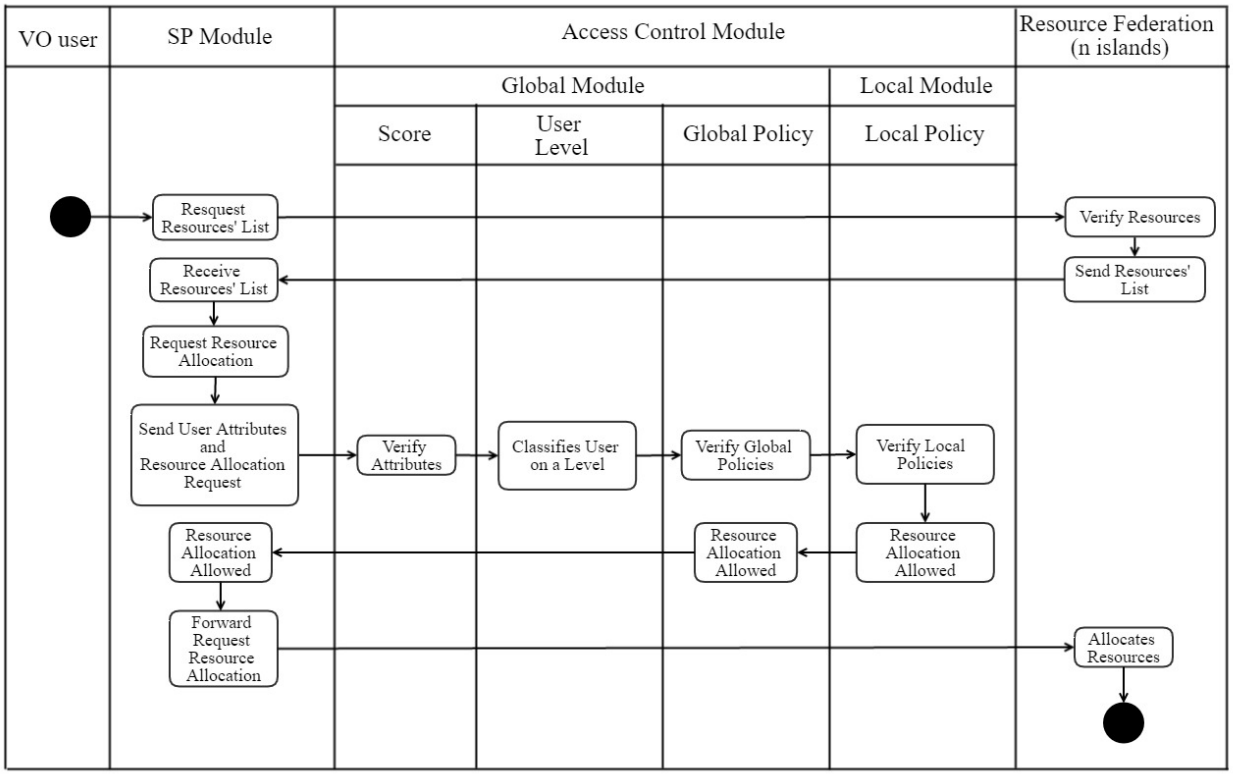

Figura 4.4: Diagrama de atividades para o usuário da OV usando ACROSS para alocar recursos.

e alocação de recursos é apresentado na Figura 4.4. Primeiro, uma lista de recursos disponíveis é requisitada à federação de recursos. Uma vez que essa informação é recebida, a federação de recursos disponibiliza a lista ao usuário, que poderá selecionar quais deles deseja requisitar. Todas as interações entre o usuário e o ambiente de recursos são realizadas *pelo SP Module*. Então, o *SP Module* fica responsável por enviar ao *Access Control Module* todos os atributos do usuário, e requisitar a reserva dos recursos. O *Access Control Module*, através do *Global Module*, tem a tarefa de verificar os atributos do usuário e associar a esse usuário uma pontuação. Baseado nessa pontuação, o usuário será classificado em um nível específico, usando o *User Level Module* [Silva et al. 2015a]. Uma vez o usuário tendo seu nível associado, políticas globais são verificadas para avaliar se a requisição do usuário é autorizada pelo submódulo *Global Policy*. Se o usuário satisfaz todos os requisitos, a requisição é enviada às instituições da federação de recursos envolvidas no pedido. As instituições, então, verificam suas políticas locais, através do submódulo *Local Policy*, pertencente ao *Local Module*. Se o usuário é autorizado, os recursos são reservados e a atividade termina.

4.2 Utilização do ACROSS

4.2.1 Identity Federation Module

O diagrama de casos de uso da Figura 4.5 representa o caso de uso para a utilização do *Identity Federation Module*. O usuário autentica-se e realiza as funcionalidades da OV, conforme o diagrama de casos de uso da Figura 4.5, podendo, então, requisitar as listas de recursos das instituições e requisitar a reserva desses recursos. Lembrando que, todos os passos, desde a autenticação até a alocação de um recurso puderam ser vistos no diagrama de atividades das Figuras 4.3 e 4.4.

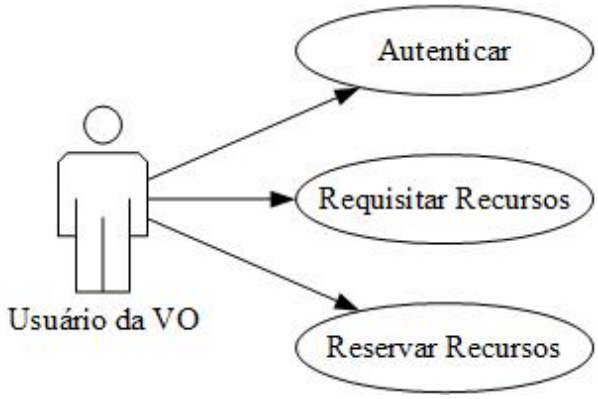

Figura 4.5: Casos de uso para usuário da OV.

A Figura 4.6 mostra o diagrama de sequência para uma autenticação do usuário da OV. Essa autenticação é baseada em SAML, onde o usuário é autenticado por uma federação de identidade. Neste diagrama, os usuários da OV realizam o procedimento de autenticação através do *SAML Module*, onde sua solicitação de login é encaminhada para a federação de identidade. Depois disso, o usuário da OV pode continuar o processo de autenticação diretamente em sua instituição de origem, com suas credenciais originais. Após a validação da credencial do usuário, o *SP Module* recebe todos os atributos do usuário, verifica-os e cria uma sessão SAML. Essas etapas são padrão para autenticação utilizando SAML.

4.2.2 Attribute Module

O *Attribute Module* permite a agregação de atributos pelo atributo *Attribute Aggregation Module* e suporta provedores de atributos adicionais pelo *Attribute Provider*. O *Attribute Provider* é uma árvore de diretórios LDAP adicional [Group 2006], que armazena atributos específicos da OV, inexistentes na federação de identidade. O *Attribute*

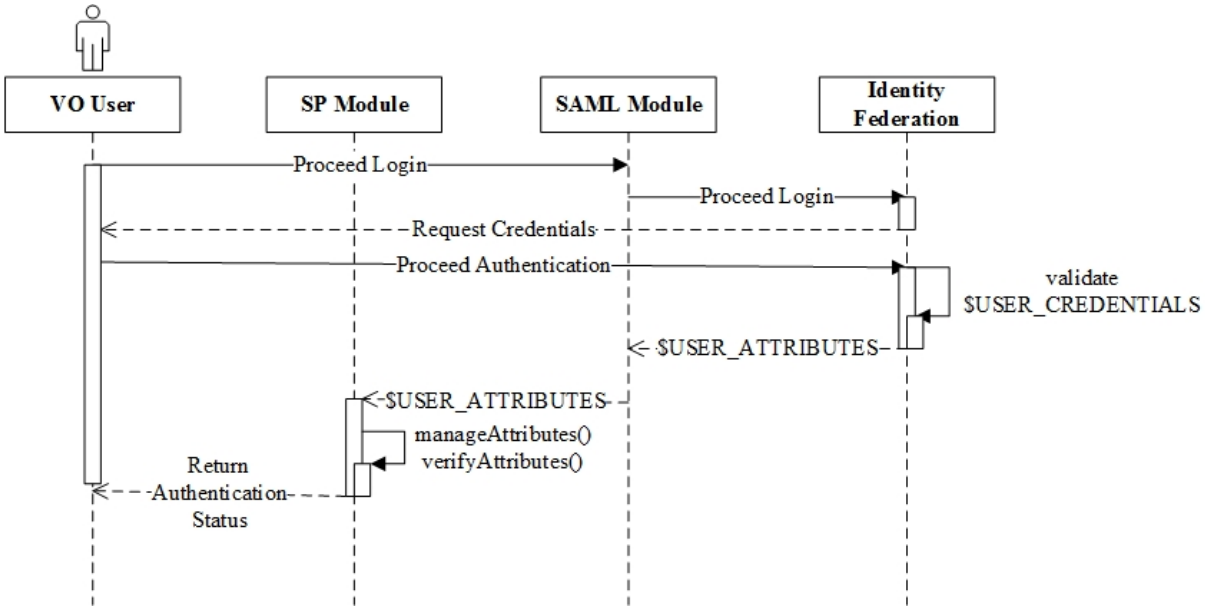

Figura 4.6: Autenticação do usuário utilizando o Módulo de Federação de Identidade.

Aggregation Module é responsável por agregar todos os atributos da federação de identidade com os adicionais dos provedores de atributos.

No *Attribute Module*, uma vez que o usuário está autenticado pelo *Identity Federation Module*, um atributo opaco é gerado pelo *Attribute Aggregation Module*, como mostra a Figura 4.7. Esse diagrama aborda a autenticação do usuário, com a funcionalidade adicional da requisição, recuperação e agregação dos atributos adicionais a partir dos provedores de atributos da OV (utilizando o atributo opaco), disponibilizando à OV através do SP Module.

O atributo opaco [Silva et al. 2015a] tem como objetivo ser um identificador único do usuário no universo dos provedores de atributos adicionais, além de preservar a privacidade do usuário na OV, uma vez que ofusca através de um *hash* a real identidade do usuário e quais atributos adicionais pertencem a ele. Um exemplo da geração de atributo opaco utilizando como entrada o atributo *mail*[2] e um número aleatório gerado na instalação do ACROSS é descrito a seguir.

$$\delta \leftarrow Attr_u(mail) \cup \$salt \tag{4.1}$$

$$Attr_U(opaque) \leftarrow hash(\delta) \tag{4.2}$$

[2] $Attr_u(mail)$ depende do schema inetOrgPerson.

Observe que o método de geração do atributo opaco é particular a cada OV, dependendo do atributo de entrada escolhido. No exemplo, foi utilizado o atributo de e-mail do usuário.

Na Figura 4.7, o usuário realiza as mesmas etapas vistas na Figura 4.6, mas agora adicionando a funcionalidade de agregação de atributos realizada pelo SP Module. O *SP Module* é responsável por enviar todos os atributos necessários para a geração do atributo opaco e requisitar os atributos específicos da OV ao *Attribute Aggregation Module*, que é responsável, por sua vez, por requisitar os atributos adicionais ao *Attribute Provider*.

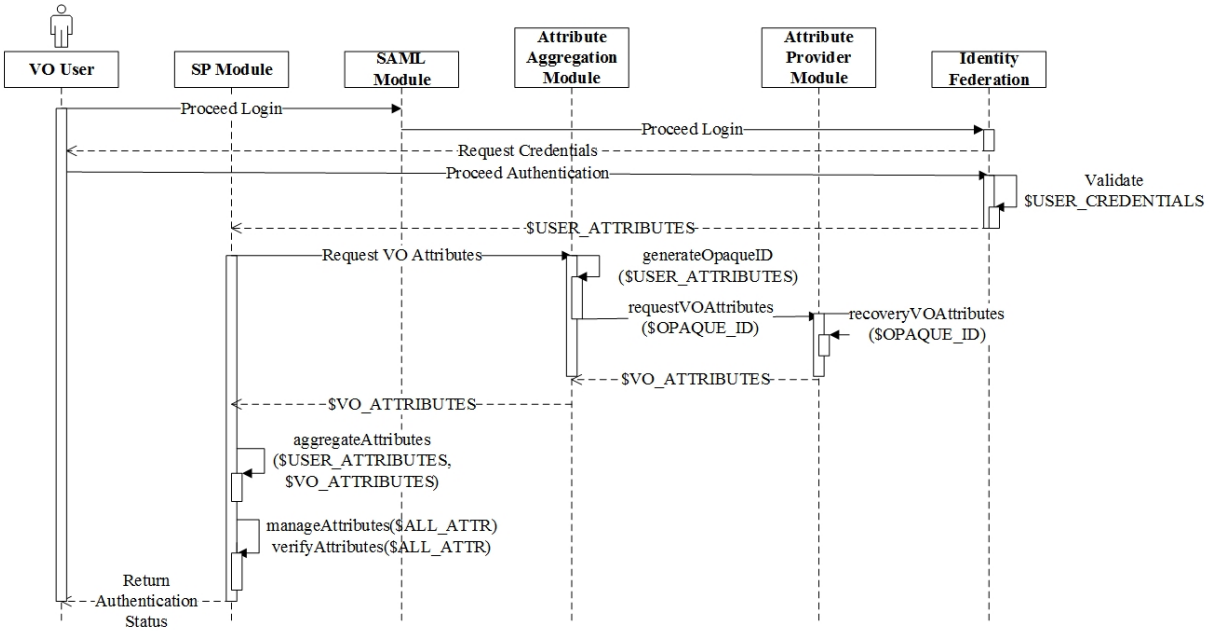

Figura 4.7: Diagrama de sequência de acesso do usuário com atributos adicionais.

4.2.3 Access Control Module

O *Access Control Module* é o ponto chave do ACROSS. Este módulo faz o controle de acesso, através do ABAC ou, até mesmo, do RBAC, aplicando políticas baseadas nos atributos e na classificação do usuário. Este módulo tem quatro submódulos no nível global e dois no nível local, como mostrado na Figura 4.2. O *VO Manager* permite ao administrador da OV configurar o *Access Control Module* conforme as necessidades da OV.

Através do *Global Module*, o administrador da OV pode configurar os pontos e pesos para os atributos do usuário, assim como os níveis aos quais cada pontuação equivale. Pontos e pesos podem ser configurados a cada par de valores dos atributos do usuário

(nome do atributo, valor do atributo). O conjunto de atributos do usuário compreende os atributos do IdP e o dos provedores de atributos da OV, ou seja, o conjunto agregado de atributos.

A seguir são apresentados os diagramas de sequência para a configuração dos parâmetros de utilização do *Access Control Module*, realizada pelo administrador da OV através da interface de gerência do Módulo Gerente da OV. Mas, antes, a Figura 4.8 mostra o diagrama de casos de uso para o módulo. Nele está representada a configuração de cada um dos submódulos já comentados nesta seção.

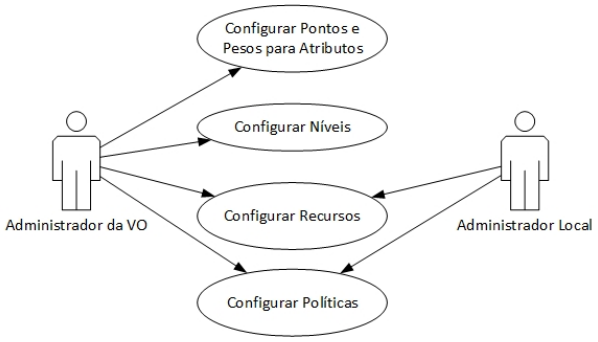

Figura 4.8: Diagrama de casos de uso para a configuração do *Access Control Module*.

O administrador da OV irá configurar primeiramente a pontuação (*score*) para todos os atributos considerados no controle de acesso. A Figura 4.9 mostra que o administrador irá informar ao assistente de configuração as credenciais para acesso ao agregador de atributos. Este passo permite que seja recuperada uma lista com todos os atributos suportados pela OV. Feito isso, a lista de atributos é disponibilizada ao administrador da OV, que associará ao atributo (representado por $ATTR_ID), um comparador ($COMPARE), um valor de atributo a ser comparado ($VALUE), pontos ($POINT) e um peso ($WEIGHT). Isso é feito para cada atributo. Ao final da configuração, o valor máximo da pontuação de cada atributo é conhecido e pode ser normalizado.

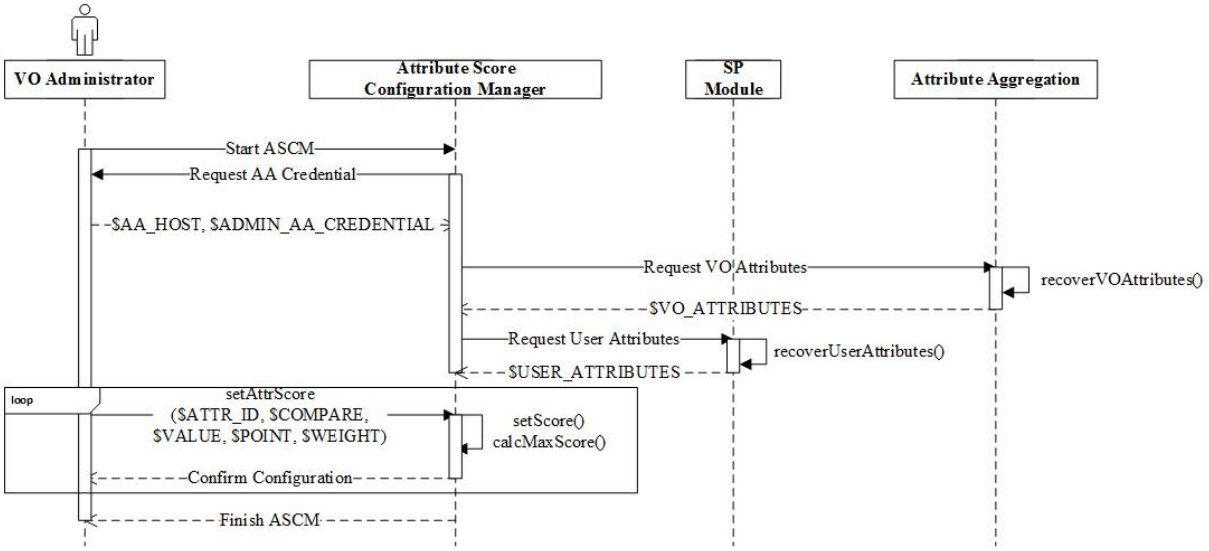

Figura 4.9: Configuração da pontuação para atributos.

Para a comparação de atributos e valores, o ACROSS disponibiliza os seguintes operadores:

- == compara se o valor do atributo do usuário é idêntico ao cadastrado pelo administrador da OV;
- >= compara se o valor do atributo do usuário é maior ou igual ao informado pelo cadastrado pelo administrador da OV;
- <= compara se o valor do atributo do usuário é menor ou igual ao informado pelo cadastrado pelo administrador da OV;
- > compara se o valor do atributo do usuário é maior ao informado pelo cadastrado pelo administrador da OV;
- < compara se o valor do atributo do usuário é menor ao informado pelo cadastrado pelo administrador da OV;
- ! = compara se o valor do atributo do usuário é diferente ao informado pelo cadastrado pelo administrador da OV;

Um atributo padrão do ACROSS, criado para contabilizar o número de vezes que o usuário utilizou o sistema, é criado logo que o módulo de controle de acesso é instalado. Ele é chamado de *howManyTimes* e é do tipo inteiro. Este atributo pode assumir a forma de um atributo dinâmico, que aplica distintas pontuações (pontos e pesos) conforme seu valor atual . A intenção de introduzir esse atributo é demonstrar como um

atributo dinâmico pode ser utilizado no controle de acesso, e ser aplicado para beneficiar dado usuário conforme utilização do sistema. Por exemplo, uma certa pontuação e peso pode ser atribuída ao usuário quando menos de 5 acessos foram realizados, mas a partir de 20 acessos, essa pontuação é maior. Vejamos os parâmetros para ilustrar esses casos:

Para número de acessos do usuário ao sistema menor ou igual a 5:

- Atributo ($ATTR_ID): howManyTimes;
- Comparador ($COMPARE): <=;
- Valor ($VALUE): 5;
- Pontos ($POINT): 20;
- Peso ($WEIGHT): 1.

Para número de acessos do usuário ao sistema maior ou igual a 20:

- Atributo ($ATTR_ID): howManyTimes;
- Comparador ($COMPARE): >=;
- Valor ($VALUE): 20;
- Pontos ($POINT): 30;
- Peso ($WEIGHT): 2.

Para facilitar a classificação dos usuários e a administração de ambientes de recursos distribuídos através de políticas, este trabalho propõe um novo método de generalização de valores, tanto para atributos de usuários como de recursos. Essa abordagem genérica é aplicada a cenários que aplicam o mecanismo de controle ABAC. Primeiramente, atributos são associados a pontos e pesos e uma pontuação é computada para, então, classificar o usuário em um nível. Os pontos e pesos de cada atributo são determinados pelo administrador da OV [Silva et al. 2015a]. O Algoritmo 1 mostra o procedimento de cálculo da pontuação dos atributos de um usuário. É possível acompanhar que o algoritmo tem como entrada todos os atributos do usuário, ALL_ATTR, esses atributos serão comparados a uma lista de atributos cadastrados pelo administrador da

OV, com suas respectivas pontuações (pontos e pesos), $SCORE\_ATTRIBUTE\_LIST$. As listas são usadas para comparação, onde, se o atributo do usuário existe na lista de atributos cadastrados, e respeita a operação de comparação < $COMPARE$ >, a pontuação é então atribuída ao usuário. Ao final, essa pontuação do usuário deverá ser normalizada. Responsável por essa normalização são as variáveis MIN e MAX, que correspondem aos valores de pontuação total mínima e máxima para os atributos do usuário, sendo previamente calculados, baseando-se nos pesos, pontos e valores para cada um dos atributos registrados pelo administrador da OV. A normalização simples para essa pontuação obriga que os valores estejam entre 0 e 1, independentemente dos pontos e pesos que o administrador da OV havia informado para cada atributo.

As variáveis utilizadas no algoritmo estão explicadas na Tabela 4.2.

Tabela 4.2: Descrição das variáveis do Algoritmo 1.

Variáveis		**Descrição**
SCORE_ATTRIBUTE_LIST		Atributos cadastrados pelo administrador da OV
SCORE_ATTRIBUTE_LIST	.NAME	Nome do atributo
	.VALUE	Valor do atributo
	.POINT	Ponto do atributo
	.WEIGHT	Peso do atributo
ALL_ATTR		Todos atributos do usuário agregados
ALL_ATTR	.NAME	Nome do atributo
	.VALUE	Valor do Atributo
MIN		Pontuação mínima possível
MAX		Pontuação máximo possível
COMPARE		Comparador
Total		Pontuação do usuário normalizada
total_score_list, total_all_attr, i and x		Variáveis locais

Uma vez configurada a pontuação de cada atributo é possível configurar os níveis de acesso. A Figura 4.10 mostra os passos para a configuração dos níveis. Esta etapa é simples, onde o administrador deverá informar o valor mínimo $MIN_SCORE e o $LEVEL correspondente a ser associado. Iniciar a atribuição de valores e níveis informando valores mínimos permite que valores não se sobreponham e insere confiança à configuração.

Algoritmo 1: Pontuação de atributos e normalização.

Entrada: ALL_ATTR, SCORE_ATTRIBUTES_LIST, COMPARE and MIN and MAX possible values from score attributes
Saída: Total user score normalized

```
// Initializing local variables
i, x, Total ← 0;
// Getting the number of attributes in user and score lists
total_all_attr ← count(ALL_ATTR);
total_score_list ← count(SCORE_ATTRIBUTES_LIST);
// Calculating score for all attributes from attribute list
   configured by VO manager
para i < total_all_attr faça
    para x < total_score_list faça
        // Compare if user attribute exists in score attribute list
           configured by VO manager
        se ALL_ATTR.NAME[i] == SCORE_ATTRIBUTES_LIST.NAME[x]
         então
            // Uses comparison operator to verify if values of user
               attribute and score attribute list are respected.
            // COMPARE can assume >, <, >=, <=, == or !== values
            se ALL_ATTR.VALUE[i] < COMPARE >
             SCORE_ATTRIBUTES_LIST.VALUE[x] então
                // Updating total value of user score
                Total ← Total + SCORE_ATTRIBUTES_LIST.POINT[x] *
                 SCORE_ATTRIBUTES_LIST.WEIGHT[x];
            fim
        fim
        // Increment index x
        x ← x + 1;
    fim
    // Increment index i
    i ← i + 1;
fim
// Normalization
Total ← [(Total − MIN)/(MAX − MIN)];
Resultado: Total
;
```

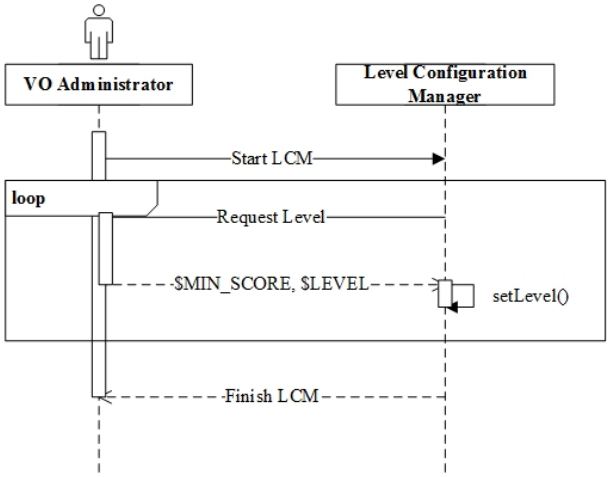

Figura 4.10: Configuração dos níveis.

Depois de configurar pontos de atributos e pesos, o administrador da OV deve definir os níveis de usuário, especificando para cada nível a pontuação mínima do usuário. Quando um usuário está associado a um nível, uma generalização pode ser usada, porque a política de controle de acesso não precisa saber exatamente quais atributos um usuário possui.

Para ilustrar a proposta de generalização baseada nos valores de atributos, um exemplo é dado na Tabela 4.3. Neste exemplo, o valor máximo é igual a 80 e o mínimo é 0, onde os valores normalizados variam entre [0-1], como segue:

Tabela 4.3. Um exemplo de pontuação para atributos do usuário

Um exemplo de pontuação do usuário.					
Attribute Name	**Valor do atributo**	**ponto**	**peso**	**pontuação**	**normalizada**
brEduAffiliationType	student	10	3	30	
omfAdmin	TRUE	10	2	20	
institution	uff	8	1	8	
			User Score	**58**	**0,725**

No exemplo da Tabela 4.3 o usuário tem pontuação de 0,725 , após normalização. No modelo proposto, o administrador da OV deve determinar um conjunto de níveis $Set_L = \{N_1, ..., N_L\}$, onde L é o número de níveis. Adicionalmente, o administrador da OV define um intervalo de valores para cada nível N_i, sendo que $l_i < N_i \leq l_{i+1}$, onde l_i é equivalente ao valor mínimo do nível N_i e l_{i+1} é o mínimo do próximo nível

N_{i+1}. Um usuário está no nível N_i, onde $1 \leq i \leq L$. No exemplo da Tabela 4.4, o administrador da OV criou 3 níveis, onde, no exemplo, o usuário da Tabela 4.3 está associado ao nível 2. O Algoritmo 2 mostra como é feita a classificação do usuário em níveis. Nesse algoritmo, é levada em consideração a pontuação do usuário, os níveis pré-configurados e seus intervalos de valores. Detalhes sobre as variáveis são descritos pela Tabela 4.5.

Tabela 4.4. Um exemplo de definição de nível baseado em pontuação.

Definição de nível	
Score	**Nível**
$0 \leq N < 0,5$	1
$0,5 \leq N < 0,75$	2
$0,75 \leq N \leq 1$	3

Algoritmo 2: Classificação de usuários em níveis.

Entrada: User total score normalized

Saída: User level classification

```
// Initializing local variables
```

1 $i \leftarrow 0$;

```
// Getting the number of levels
```

2 $total\_levels\_list \leftarrow count(LEVELS\_LIST)$;

```
// Verying levels configured by VO manager
```

3 **para** $i \leq total\_levels\_list$ **faça**

```
    // Verifying if user score is on range of level
```

4 **se** $LEVELS\_LIST.MIN\_SCORE[i] < User.Total \leq LEVELS\_LIST.MAX\_SCORE[i]$ **então**

```
        // If it is true, classify user in this level
```

5 $User.Level \leftarrow LEVELS\_LIST.NUMBER[i]$;

```
        // If it is not true, try next level
```

6 **fim**

```
    // Increment i
```

7 $i \leftarrow i + 1$;

8 **fim**

9 **retorna** $User.Level$;

Tabela 4.5. Descrição das variáveis do Algoritmo 2.

Atributos		**Descrição**
LEVELS_LIST		Níveis cadastrados pelo administrador da OV. Com um intervalo de valores associado.
LEVELS_LIST	.MIN_SCORE	Valor mínimo de pontuação
	.MAX_SCORE	Valor máximo de pontuação
	.NUMBER	Número do nível
User	.Total	Pontuação do usuário normalizada
	.Level	Nível que usuário foi classificado
i, total_levels_list		Variáveis locais

Uma vez concluída a etapa de configuração de pontuação de atributos e níveis de usuário, o administrador da OV deve também registrar os tipos de registros suportados pela OV, usando o módulo *Global Resource*. Níveis de usuários e tipos de recursos são utilizados para definir políticas globais e locais. Sendo assim, é apresentado o diagrama sequência para o cadastro de recursos. A Figura 4.11 mostra que é solicitado ao administrador global da OV cadastrar um recurso apenas com os dados de tipo ($TYPE) e uma descrição em texto livre ($DESCRIPTION).

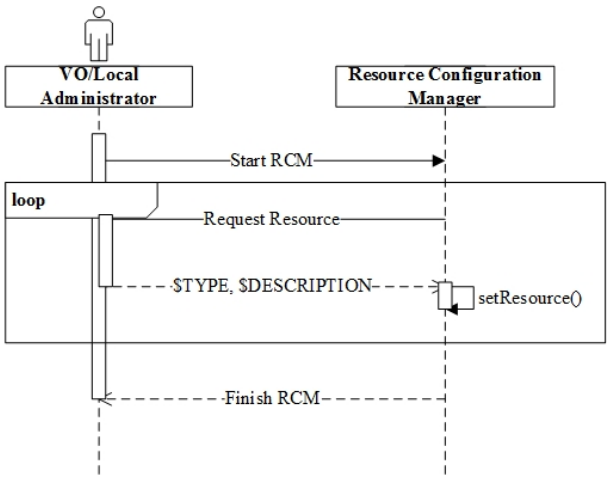

Figura 4.11. Configuração de recursos.

O administrador da OV pode criar políticas globais através do módulo *Global Po-*

licy. Um administrador local é responsável por informar o número de recursos de cada tipo oferecido por sua instituição, usando o módulo *Local Resource*, e por criar políticas locais através do módulo *Local Policy*.

A fim de definir as políticas, um administrador (OV ou local) deve definir o número de recursos que um usuário em um determinado nível pode solicitar. Por exemplo, suponha que máquinas virtuais (*Virtual Machines* – VM) são registradas como tipos de recursos suportados. A Tabela 4.6 mostra um exemplo considerando três níveis de usuário. Um usuário de nível 2 podem solicitar até 15 máquinas virtuais.

Tabela 4.6. Políticas de controle de acesso em níveis para máquinas virtuais.

Access Control policies based on scores for virtual machines	
Level	**VMs**
1	$Num\_VM \leq 5$
2	$Num\_VM \leq 15$
3	$Num\_VM \leq 20$

A Figura 4.12 mostra as etapas de configuração de políticas. O administrador começa pela Gerência de Configuração de Políticas (*Policy Configuration Manager*), que por sua vez irá consultar os níveis cadastrados através do Gerência de Níveis de Usuário (*User Level Manager*), e os recursos em Gerência de Recursos (*Resource Manager*). Então, o administrador informa ao *Policy Configuration Manager*, para cada tipo de recurso ($RESOURCE_ID), um nível de acesso ($LEVEL_ID), uma descrição da política ($POLICY_DESCRIPTION), e um número de recursos a que a política é associada ($NUMBER_OF_RESOURCES). O ACROSS, então, cria um arquivo XACML [Moses 2005] descrevendo cada política. Essa etapa pode ser repetida várias vezes.

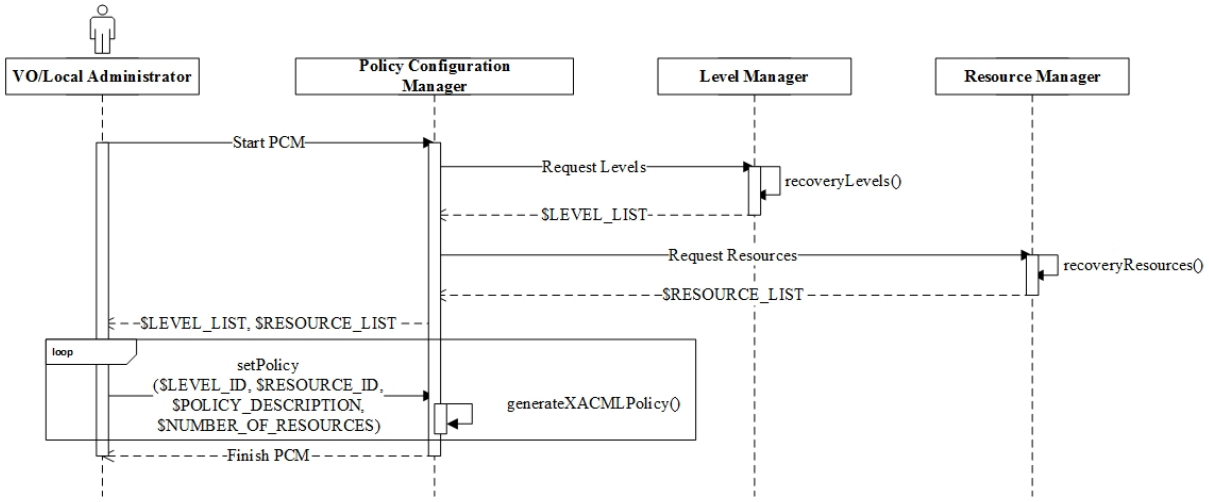

Figura 4.12. Diagrama de sequência para a configuração de políticas.

Um exemplo de política XACML criada pelo administrador da OV usando ACROSS é mostrado na Listagem 4.1, onde pode-se ver que a política é aplicada ao nível 3 (*level 03*). Na linha 1, observa-se o algoritmo de combinação utilizado para esta política, neste caso o *deny-overrides*, que representa uma política mais restritiva, onde, se uma das políticas tem reposta igual a *deny*, todas as respostas às demais políticas são negadas. Na linha 7, observa-se o valor de nível do usuário, e na linha 8 observa-se a referência a esse atributo como aquele ao qual a política verificará. Na linha 16, tem-se a saída da política, onde, se o usuário tem nível igual a 3, a política tem como resposta *permit*, ou seja, permitido.

Listagem 4.1. Exemplo de configuração de política em XACML.

```
<PolicySet PolicySetId="policies-0001" PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-
    algorithm:deny-overrides">
  <Description> </Description>
  <Target>
    <Subjects>
      <Subject>
        <SubjectMatch MatchId="urn:oasis:names:tc:xacml:2.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string"> 03 </AttributeValue>
              <SubjectAttributeDesignator> level </SubjectAttributeDesignator>
        </SubjectMatch>
      </Subject>
    </Subjects>
  </Target>
  <PolicyIdReference> across-policy-0001 </PolicyIdReference>
  <Policy PolicyId="across-policy-0001" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-
      algorithm:deny-overrides">
    <Target/>
    <Rule RuleId="across-rule-0001" Effect="Permit"/>
  </Policy>
</PolicySet>
```

4.3 Diagrama de Classes

Como complemento à documentação do ACROSS são apresentados seus diagramas de classes. O diagrama apresentado na Figura 4.13, representa todos os módulos do ACROSS, exceto o ACROSS Wizards. A especificação e detalhamento do ACROSS Wizards são apresentados no Apêndice A. É interessante destacar também que a documentações complementares sobre as funções e arquivos gerados pela implementação do ACROSS, podem ser vistas nos Apêndices B e C.

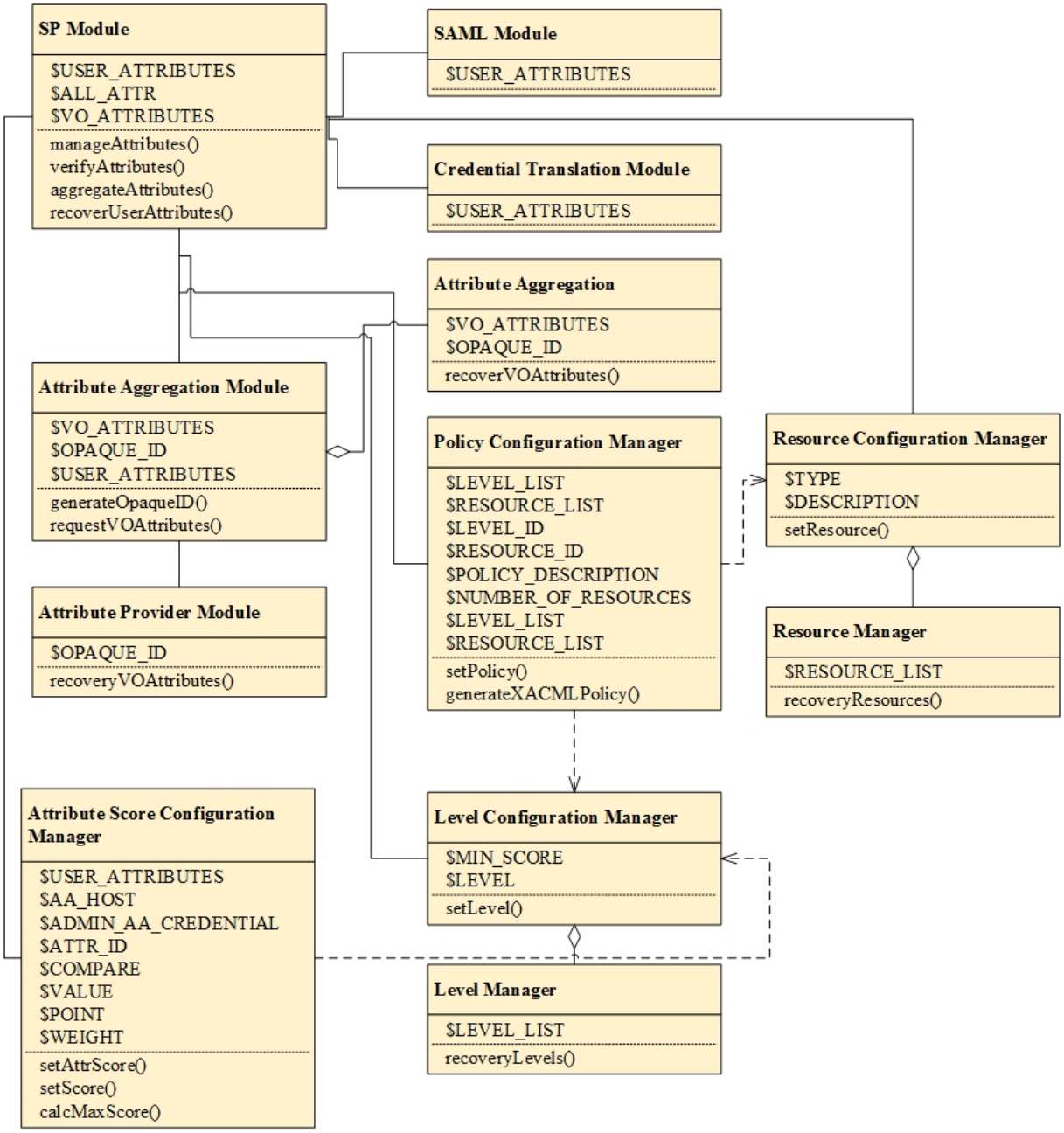

Figura 4.13. Diagrama de classes do ACROSS.

Sendo assim, complementando o exposto em cada seção desse capítulo, os diagramas de classes mostram de forma objetiva a associação e dependência entre os módulos, seguindo o que foi visto pelos diagramas de sequência anteriormente.

4.4 ACROSS e o estado-da-arte

Como forma de comparação entre a proposta introduzida nesta tese, destacam-se os principais pontos todos os *frameworks* do Capítulo 3 e o ACROSS.

Em comparação com o VOMS, o ACROSS tem bem claro o papel das políticas globais e locais, funcionando, então, de forma bem diferente. Para o ACROSS, o ambiente de uma VO é como uma federação de recursos, seja esse recurso de que tipo for, diferentemente da origem do VOMS, onde recursos são nós de uma *grid* computacional. No ACROSS as políticas globais validam e classificam o usuário com base em seus atributos, classificando-o em um nível específico de acordo com a configuração realizada pelo administrador global VO. E, no nível local, apenas o nível de acesso do usuário é necessário para verificar a solicitação do usuário a um recurso. O ACROSS também garante a privacidade dos atributos dos usuários, já que apenas um atributo genérico (*i.e.* o nível do usuário) é enviado aos pontos de decisão locais, mantendo os valores de atributos e cálculos de pontuação em nível global. Diferentemente do VOMS, que necessita que o usuário reapresente suas credenciais no nível local, mesmo que já as tenha apresentado no nível global.

O ACROSS visa facilitar a criação de uma VO ou o ingresso a uma VO preexistente. Diferentemente do VOMS, o ACROSS visa, neste primeiro momento, o ambiente de uma VO. O VOMS, por sua vez, permite, através da sua exigência de padronização das credenciais, a integração entre VOs. Porém, o ACROSS simplifica a gestão de atributos específicos à VO, e a geração de credenciais a ambientes diversos de recursos distribuídos, sem a necessidade de adaptar o ambiente a um determinado tipo de credencial, como é o caso do VOMS.

Comparando o ACROSS ao CAS, esse último apresenta políticas globais e locais mais simples e limitadas. Apesar de, para a época de sua proposta, atender às premissas de políticas hierárquicas e distribuídas, é, hoje, uma versão simples baseada em RBAC. O ACROSS tem sua estrutura de controle de acesso baseado em ABAC e XACML. No CAS, o papel do usuário e a permissão associada a ele é processada no nível global da VO, assim que o usuário ingressa pelo *CAS Server*. A partir daí, as políticas serão aplicadas conforme essa credencial (certificado proxy) gerado. No ACROSS, essa verificação é feita tanto logo após a autenticação, como no momento do pedido de acesso a um recurso, pelo *Access Control Module*, com base nos atributos do usuário e sua classificação em um nível. O ACROSS oferece uma abordagem mais flexível, uma vez que a cada interação com a federação de recursos a classificação do usuário pode mudar. Ao mesmo tempo, o ACROSS verifica atributos não estáticos, que podem ter valores diferentes em momentos distintos, uma característica da utilização do ABAC no ACROSS.

Já o PERMIS, diferentemente do CAS e do VOMS, mas assim como o ACROSS, suporta o ABAC. No caso do PERMIS, os atributos considerados estão armazenados em certificados de atributos. As principais diferenças entre PERMIS e ACROSS estão na facilidade de instalação, configuração e integração oferecidos pelo ACROSS às VOs. Sabe-se que o PERMIS é uma estrutura madura, disseminada e bem estruturada, e que se apoia em infraestrutura de chaves públicas com certificados do tipo X.509, assim como o VOMS e o CAS. No entanto, PERMIS, VOMS e CAS herdaram, ou tinham a intenção de se integrar, a ambientes que utilizam o *Globus Toolkit*, principalmente, em grades computacionais. A adaptação desses *frameworks* para utilização em outros cenários pode ser bastante dispendiosa, uma vez que, apesar da utilização de certificados X.509 e certificados proxy ser um ponto forte de segurança ao ambiente, essa exigência acaba por atrelar essa infraestrutura à da VO. O ACROSS, neste caso, tenta ser mais genérico, facilitando a integração com qualquer ambiente de recursos a partir do *Credential Translation Module*, que libera os atributos para serem utilizados para gerar quaisquer tipos de credenciais (certificados X.509, *tokens* oAuth etc). Além disso, o ACROSS prevê outras funcionalidades, como o suporte a atributos adicionais e a adição do controle de acesso baseado em papéis e atributos.

O Akenti, assim como o ACROSS, o CAS e o PERMIS, tem suporte a políticas distribuídas, onde as suas entidades (chamadas de *stakeholders* e responsáveis por um recurso, ou um conjunto deles) são responsáveis por verificar a permissão de acesso de um usuário a um recurso. Isso faz com que Akenti possa implementar políticas globais e locais. Porém, como o Akenti foi pensado primeiramente para acesso a recursos individuais, sua escalabilidade aparece como um ponto fraco. Como o acesso é feito com o suporte da infraestrutura de chaves públicas, cada *stakeholder* deve conhecer e confiar na CA de cada potencial usuário. Essa característica gera um esforço adicional de manutenção e limita a escalabilidade da solução. Diferentemente, o ACROSS apresenta uma hierarquia clara de políticas baseadas em XACML e um ponto central de administração das configurações globais da VO.

No ACROSS, os mesmos conceitos do *System for Cross-domain Identity Management* (SCIM), Grouper e *Virtual Organisation Orthogonal Technology* (VOOT) podem ser usados, uma vez que usuários são classificados em níveis de acordo com seus atributos. Sendo assim, pode-se fazer uma analogia com a classificação por grupos dos trabalhos citados. Usuários que tenham o mesmo valor para seus atributos têm a mesma pontuação e são alocados no mesmo nível. Neste caso, conforme a configuração do administrador da VO, é possível criar uma equivalência entre um nível e um grupo

específico. A interface web do SCIM VO facilita a gerência de atributos e grupos, mas a maior diferença entre ACROSS e soluções baseadas em SCIM é que as soluções baseadas em SCIM são específicas para a gerência de usuários e grupos. O ACROSS pensa em uma solução muito mais flexível. Assim, atributos do ACROSS podem determinar grupos do SCIM, mas não o contrário.

4.5 Sumarização da Comparação

Para sumarizar a comparação, foram levantados os principais pontos de interesse atualmente para a proposta de um *framework* para VOs:

- para a **autenticação**, o suporte à federação de identidade (SAML/Shibboleth), o provedor de atributos adicionais e a agregação de atributos;
- para a **autorização**, o suporte aos mecanismos RBAC e ABAC, o uso de políticas distribuídas, globais e locais e o suporte ao XACML.

Os símbolos usados na comparação são apresentados pela Tabela 4.7. O *suporte nativo* é intuitivo; o *suporte parcial* é explicado nos próximos parágrafos para aqueles que têm ocorrência dessa característica. As *soluções de terceiros* são baseadas em ferramentas criadas por colaboradores ou outros projetos, adaptadas para funcionar com o *framework* em questão.

Tabela 4.7. Descrição dos símbolos

Símbolo	Significado
✓	Suporte nativo
□	Suporte parcial
■	Soluções de terceiros
	Sem suporte

Tabela 4.8. Comparação entre as propostas apresentadas

	ACROSS	**CAS**	**VOMS**	**PERMIS**	**Akenti**
Suporte à Federação de Identidade	✓	✓	■	✓	■
Suporte à Provedor de Atributos	✓	✓	✓	✓	✓
Suporte à Agregação de Atributos	✓				
Suporte ao RBAC	✓	✓	✓	✓	✓
Suporte ao ABAC	✓		□	□	□
Políticas Distribuídas	✓	✓	□	✓	✓
Suporte ao XACML	✓				

O ACROSS é mais do que apenas um *framework* que visa aplicação em um ambiente específico, ele aborda diversos conceitos de IAM. Na tabela, observa-se que o ACROSS suporta as diferentes funcionalidades necessárias a IAM. A integração com a federação de identidade é realizada pelo *Identity Federation Module*, conectando esse *framework* com uma federação baseada em Shibboleth (em nossa validação à CAFe Expresso). O provedor de atributos adicionais e a agregação de atributos são suportados pelo *Attribute Module* e o suporte ao RBAC e ABAC, assim como as políticas distribuídas, pelo *Access Control Module*. O ACROSS ainda pode ser facilmente estendido a partir das chamadas via SOAP e troca de arquivos XML. Além disso, seus módulos são independentes e podem ser facilmente modificados, estendidos ou substituídos. O ACROSS ainda apresenta um suporte à instalação e configuração facilitada através de assistentes.

O suporte à federação de identidade é nativo ao ACROSS e também ao PERMIS com o *Shibboleth and Apache Authorization Module* (SAAM) [Xu et al. 2005], e o CAS, com seu conceito natural de autorização e suporte às instituições distribuídas. Os de-

mais sistemas têm soluções de terceiros. No VOMS, assim como para o Akenti, a solução desenvolvida por terceiros foi o ShibGrid [Spence et al. 2006], desenvolvida no contexto do *UK National Grid Service*.

Todos os *frameworks* apresentam suporte ao provedor de atributos. O CAS, o VOMS, o PERMIS e o Akenti suportam esses atributos adicionais em certificados de atributos, respeitando sua infraestrutura de chaves públicas e certificados herdados do Globus Toolkit. Porém, apenas o ACROSS apresenta suporte ao agregador de atributos, que, além de agregar os atributos da federação de identidade com os atributos do provedor de atributos adicionais, tem uma característica escalável, suportando diversos provedores de atributos adicionais. O ACROSS ainda provê a privacidade do usuário, através do atributo opaco.

Focando na autorização, o ACROSS apresenta dois mecanismos que merecem destaque: o RBAC e ABAC. O RBAC é suportado por todos os *frameworks* através de papéis. O ABAC é suportado nativamente pelo ACROSS e parcialmente pelo VOMS, PERMIS e Akenti. Diz-se que este suporte é parcial porque estes *frameworks* têm suporte a certificados de atributos, porém a análise desses atributos é sempre realizada na forma de papéis estáticos, associando-os a perfis pré-estabelecidos. Ou seja, esses *frameworks* poderiam ser estendidos para suportar o ABAC, porém teriam que levar em consideração todas as características das entidades desse mecanismo, como a adoção de políticas distribuídas e hierárquicas, a avaliação de atributos dinâmicos etc.

Para prover políticas distribuídas, diz-se que o *framework* deve suportar políticas globais e locais. Conforme visto neste capítulo, o ACROSS tem um suporte nativo e de simples entendimento, se apoiando no XACML e X.812. O CAS, o PERMIS e o Akenti foram considerados como suporte nativo. Contudo, destaca-se, por exemplo, que o Akenti tem limitações quanto ao suporte à escalabilidade de suas políticas. Já o VOMS tem apenas suporte parcial, por ter sua verificação apenas no nível global. Por ser o mais atual *framework* proposto dentre os comparados, o ACROSS é o único com suporte ao XACML, herdando todos seus benefícios discutidos no Capítulo 2.

Capítulo 5

Casos de Uso do framework ACROSS

Neste capítulo, são apresentados dois casos de uso do *framework* ACROSS. Dois casos de uso são apresentados. O primeiro aborda o caso de uso da proposta inicial do ACROSS, naquele momento, focado para o ambiente de FI. Este primeiro caso de uso, portanto, se apoia no projeto FIBRE. Já o segundo estudo de caso, valida o ACROSS em seu estado atual, proposto por esta tese, em uma OV hipotética de recursos distribuídos.

5.1 FIBRE

Neste caso de uso, o ACROSS é chamado de ACROSS-FI, e introduz uma validação aos módulos do ACROSS no cenário de FI, tendo o FIBRE como cenário. Essa etapa foi validada em [Silva et al. 2015a], utilizando o Laboratório de Gestão de Identidade da RNP, o GIdLab [Wangham et al. 2013], instalando o portal do FIBRE [MySlice 2013] e criando duas ilhas de experimentação com recursos de VM (*Virtual Machines* – Máquinas Virtuais).

Primeiramente, o *Identity Federation Module* é validado, logo em seguida o *Attribute Module* com a agregação de atributos adicionais e a transposição de credencias. Por fim, a generalização dos atributos por meio da conversão de atributos em pontuação e níveis, para assim validar o *Access Control Module* nesse ambiente.

5.1.1 Cenário de Validação

A Figura 5.1 mostra o cenário do caso de uso do ACROSS para o projeto FIBRE e todos os componentes envolvidos, a partir da autenticação até a autorização. É possível ver, na parte superior, a federação de identidade CAFe Expresso e o LDAP Federado

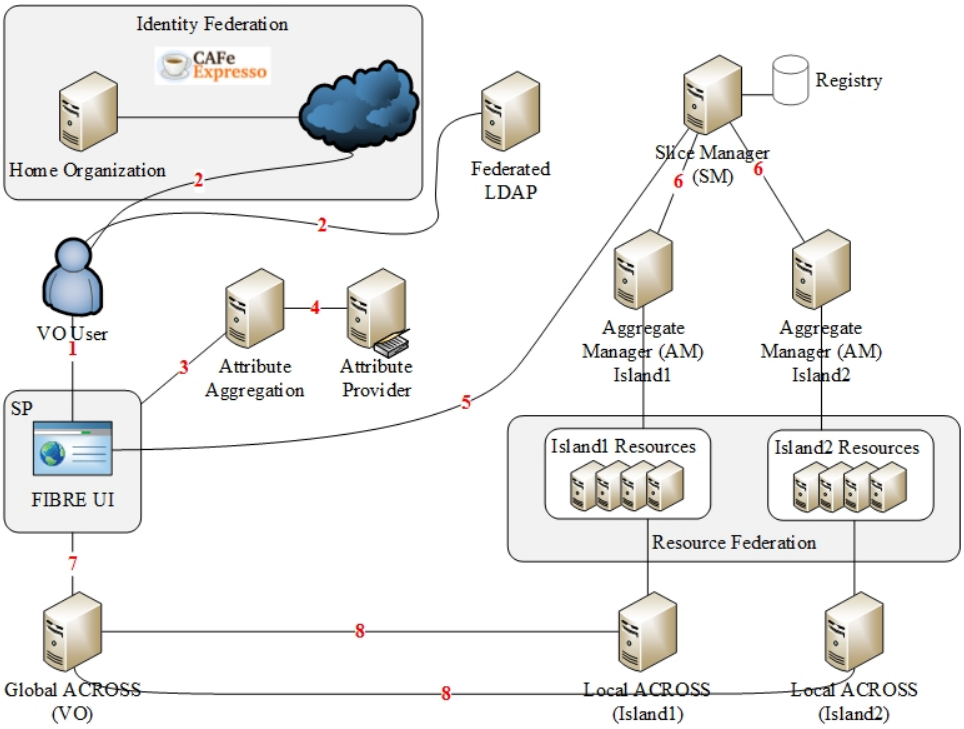

Figura 5.1. Cenário do caso de uso do ACROSS para FI no projeto FIBRE.

do FIBRE, onde o processo de autenticação ocorre. O LDAP Federado é uma base de credenciais para usuário de instituições parceiras que não estão na CAFe. Essa etapa de comunicação entre as entidades é realizada pelo *Identity Federation Module* do ACROSS, configurado no SP. É possível ver também, além do próprio portal do FIBRE no SP, a indicação dos serviços de agregação de atributos (*Attribute Aggregation*) e provedor de atributos (*Attribute Provider*), que compõem o *Attribute Module*. Além disso, o controle de acesso ABAC por pontuação do *Access Control Module* é realizado pelo *Global ACROSS* e os *Local ACROSS* das ilhas de experimentação.

Nesse caso de uso, estão definidos os passos para cada atividade realizada nesse cenário. Esses passos estão compreendidos nas subseções seguintes e serão descritos dentro de cada módulo ao qual pertencem. Os passos descritos nesse caso de uso são os mesmos apresentados de forma geral pelos digramas das Figuras 4.3 e 4.4.

5.1.2 Identity Federation Module

No **passo 1**, o usuário acessa o SP, que o encaminha para autenticação (**passo 2**), que pode ser feita pela federação de identidade CAFe, ou então pelo diretório LDAP federado do FIBRE. O LDAP federado do FIBRE é uma árvore de diretórios que gerencia as credenciais de todas as instituições que não integram a CAFe, mas participam dessa

OV. Lembrando que, as etapas de autenticação descritas são as etapas tradicionais para a criação de uma sessão SAML [OASIS 2005], desde a requisição de acesso pelo usuário ao SP, sendo redirecionado ao WAYF (*Where Are You From*) até seu IdP para autenticação, e a liberação dos atributos entre o IdP e o SP. Esse passo é realizado pelo *SAML Module* do *Identity Federation Module*. A Figura 5.2 mostra a tela de autenticação do portal do FIBRE.

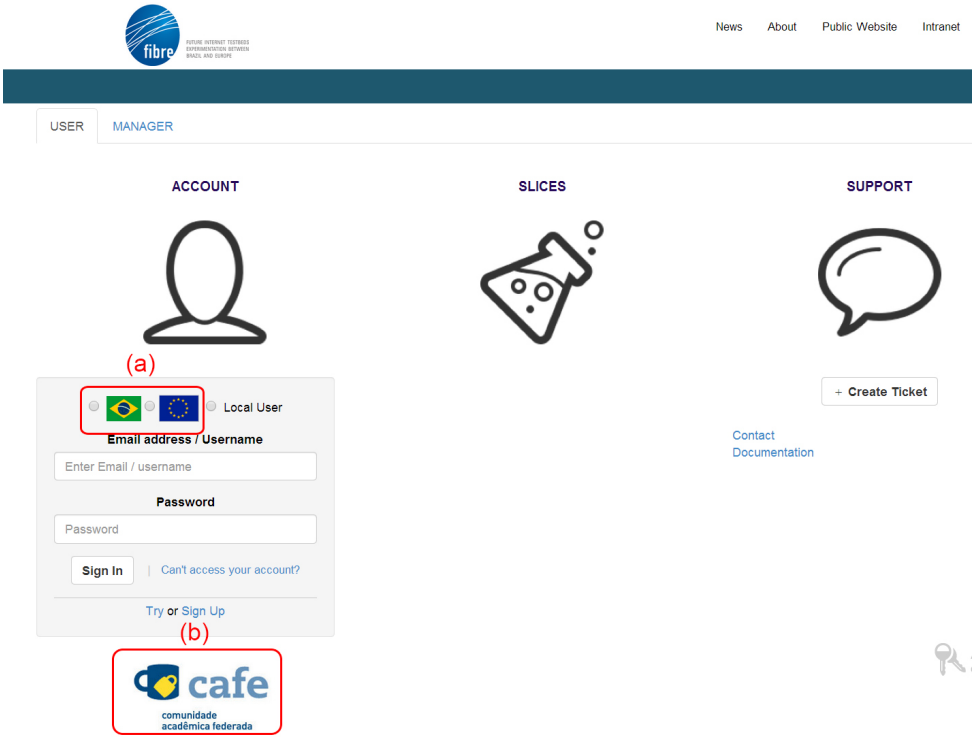

Figura 5.2. Portal FIBRE com acesso ao LDAP federado (a) ou CAFe Expresso (b).

Nas Figuras 5.3a e 5.3b ilustra-se a autenticação do usuário, com a escolha de seu IdP para autenticação, no WAYF, e a autenticação em si, utilizando suas credenciais da federação de identidade.

Uma vez validada a autenticação do usuário em sua instituição de origem, o usuário será redirecionado ao SP, que por sua vez, realizará os passos a seguir.

5.1.3 Attribute Module

Neste momento, o usuário já encontra-se autenticado pela federação de identidade e será realizada a agregação dos atributos adicionais da OV aos atributos recebidos da federação de identidade. Após esses passos, será gerada a credencial do usuário para o ambiente FIBRE a partir desses atributos, utilizando a transposição de credenciais, o *Credential Translation Module* do ACROSS.

(a) Escolha do IdP da instituição pelo usuário.

(b) Autenticação na CAFe com uso das credenciais.

Figura 5.3. Passos de autenticação utilizando a CAFe.

5.1.3.1 Agregação de Atributos

No **passo 3** o *SP Module*, localizado no SP, envia os atributos necessários à criação do atributo opaco pelo *Attribute Aggregation Module*. Então, os atributos adicionais do FIBRE são recuperados do *Attribute Provider* no **passo 4** e agregados, sendo assim, disponibilizados ao SP.

No caso de uso do FIBRE, foi feita uma implementação simples do atributo opaco e agregação, onde, para o usuário com os atributos *uid* $\rightarrow$ *esilva@uff* com *uidNumber* $\rightarrow$ *1223*, concatenando esses dois atributos foi gerado o *hash* MD5 $af2ec12ce73cc910358ddb400f4abb74$ como atributo opaco. Outros *hashes* criptográficos, assim como a proposta desta tese no Capítulo 4, com adição de um *salt*, podem ser usados em um ambiente de produção.

Após as etapas descritas na seção do *Identity Federation Module* desse caso de uso, a Figura 5.4 ilustra todos os atributos do usuário, tanto aqueles da federação de identidade CAFe Expresso como os atributos adicionais do *Attribute Provider* (destacados

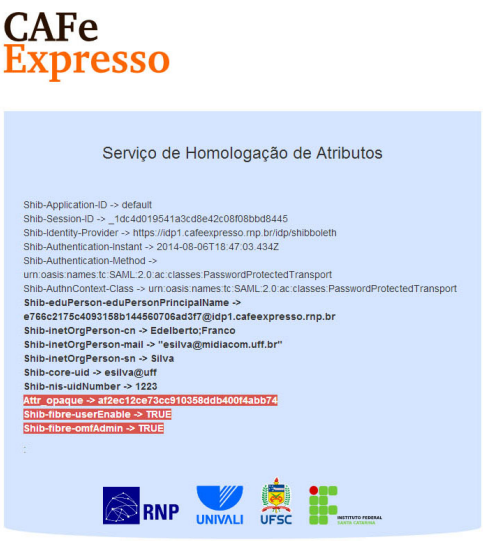

Figura 5.4. Resultado da autenticação e agregação de atributos para o usuário FIBRE.

em vermelho). Os atributos destacados, são específicos ao ambiente da OV FIBRE, i.e., *Shib-fibre-userEnable*, indica se o usuário está ativo no testbed, *Shib-fibre-omfAdmin*, se o usuário é administrador OMF (*cOntrol and Management Framework*) [Rakotoarivelo et al. 2010] e o atributo opaco do usuário, *Attr_opaque*.

5.1.3.2 Transposição de Credenciais

Os atributos do usuário são disponibilizados para a transposição de credenciais pelo *Credential Translation Module*, que, por sua vez, permitem a geração da credencial SFA (*Slice-Based Federation Architecture*) [Peterson et al. 2008] do usuário.

Como demonstração da transposição de credencial para o ambiente do FIBRE, tem-se o conteúdo da credencial do usuário em SFA, sendo um certificado do tipo X.509, como pode ser visto na Figura 5.5.

Para que a transposição de credencial fosse possível nesse ambiente, houve um esforço do gestor do FIBRE, já que os atributos do usuário foram disponibilizados pelo *Credential Translation Module*. Houve a necessidade de armazenamento dessa credencial transposta no *Registry* do SFA, que é a base de dados responsável por todos os registros de controle no FIBRE, desde credenciais do usuário até informações sobre recursos e reservas.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 3 (0x3)
    Signature Algorithm: md5WithRSAEncryption
        Issuer: CN=fibre
        Validity
            Not Before: Nov  8 01:08:21 2013 GMT
            Not After : Nov  7 01:08:21 2018 GMT
        Subject: CN=fibre.dummy.esilva@uff
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d7:76:ed:c8:c9:e2:e2:9e:a5:65:90:be:1c:5a:
                    16:ca:09:53:a2:f7:ca:84:e8:e4:f2:22:98:2b:af:
                    42:de:6a:e2:4d:bf:d6:ed:af:58:58:b9:67:6a:9a:
                    98:10:11:b4:a0:30:b5:cf:a1:f5:aa:3b:58:5e:c5:
                    7b:af:30:c2:f4:b1:20:ee:df:b4:cc:ab:40:4e:02:
                    60:8c:4b:5f:7e:9e:ee:92:a2:5c:16:8e:9f:2a:f3:
                    1b:c1:6a:c3:84:06:74:0f:53:32:ed:9c:64:38:e5:
                    15:33:73:48:f9:03:5d:b1:7c:c6:47:00:bd:1a:96:
                    e4:01:84:f7:bc:7e:a4:f7:cd:d9:b3:fc:60:6c:df:
                    c0:bf:1b:28:d5:24:e6:00:ae:07:4e:a4:d1:2f:ae:
                    ab:a3:ea:1d:42:30:88:a0:2c:ef:9f:c2:2f:17:4a:
                    7a:79:c6:b1:0b:c2:f4:f5:19:ed:14:c0:c4:9b:d1:
                    aa:55:9e:60:5b:ec:cc:44:45:2f:33:b5:43:df:5b:
                    70:cf:07:79:63:a9:d3:07:9f:ce:d6:78:e3:10:c4:
                    50:c4:17:15:2f:73:c6:bc:72:40:11:10:8c:6b:7b:
                    9b:57:3c:e8:dc:4e:aa:82:70:34:c3:ca:b0:d8:e8:
                    73:a1:f2:ca:76:06:7a:1d:a9:98:4c:20:4e:77:78:
                    5f:eb
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
         7f:8c:06:bd:27:60:92:a4:9c:aa:6a:91:2a:4b:ff:19:a6:d9:
         9f:e9:cf:66:87:73:a3:35:e3:72:6e:45:e3:e6:f4:1f:ef:59:
         fb:bf:f4:9c:6b:74:b2:29:f5:c3:d8:a9:0a:cb:27:d0:c6:2d:
         d4:b4:39:fd:67:37:81:90:93:5c:31:a9:e3:95:7e:5a:24:9e:
         c3:bb:f1:1f:27:0a:63:b4:63:7e:2a:dd:a5:60:d1:27:cf:20:
         1b:45:b2:07:ff:90:67:9c:d9:ea:d1:22:ea:43:ed:d7:f2:15:
         11:68:79:ec:ec:8a:37:44:7c:45:3b:69:77:e4:e5:da:77:ff:
         b2:96:65:eb:82:d4:74:c6:09:ac:fd:83:66:15:35:93:69:67:
         51:c5:98:8a:70:8d:03:79:95:4d:04:51:af:c5:0c:66:3f:88:
         2e:11:7f:4a:66:c5:e9:10:eb:af:11:ab:ae:5b:01:64:18:00:
         eb:ac:e9:cc:64:0e:e8:b2:4c:f5:b9:9a:b6:7f:17:7a:01:4c:
         09:13:df:1c:91:07:b3:94:3c:19:4a:de:0b:ea:34:78:ad:23:
         5e:d7:b0:ea:47:4a:2a:99:4d:69:a4:e2:c0:0e:eb:1a:ae:48:
         b9:dc:d6:de:d4:d5:09:7f:d9:5c:11:3e:a3:ff:06:6f:2f:63:
         6c:5a:82:94
```

Figura 5.5. Credencial do usuário.

5.1.4 Listagem de Recursos

No **passo 5** ocorre uma etapa específica do ambiente do projeto FIBRE. O usuário autenticado requisita uma lista de recursos disponíveis nas ilhas do FIBRE. Essa requisição utiliza o SFA [Peterson et al. 2010] como protocolo de comunicação na federação de recursos. Nessa etapa, o SFA é responsável por se comunicar com o SM (*Slice Manager*), responsável pela criação da gerência da fatia de recurso alocado ao usuário, e que tem a comunicação com os AM (*Aggregate Managers*), que gerenciam localmente os recursos em cada ilha (**passo 6**). E então, os recursos disponíveis são listados através de um arquivo baseado em XML, chamado RSpecs (*Resource Specification*) e retornados ao usuário, habilitando o usuário a requisitar a reserva de recursos.

É possível ver a listagem de recursos reservados ao usuário já autenticado no portal do FIBRE pela Figura 5.6.

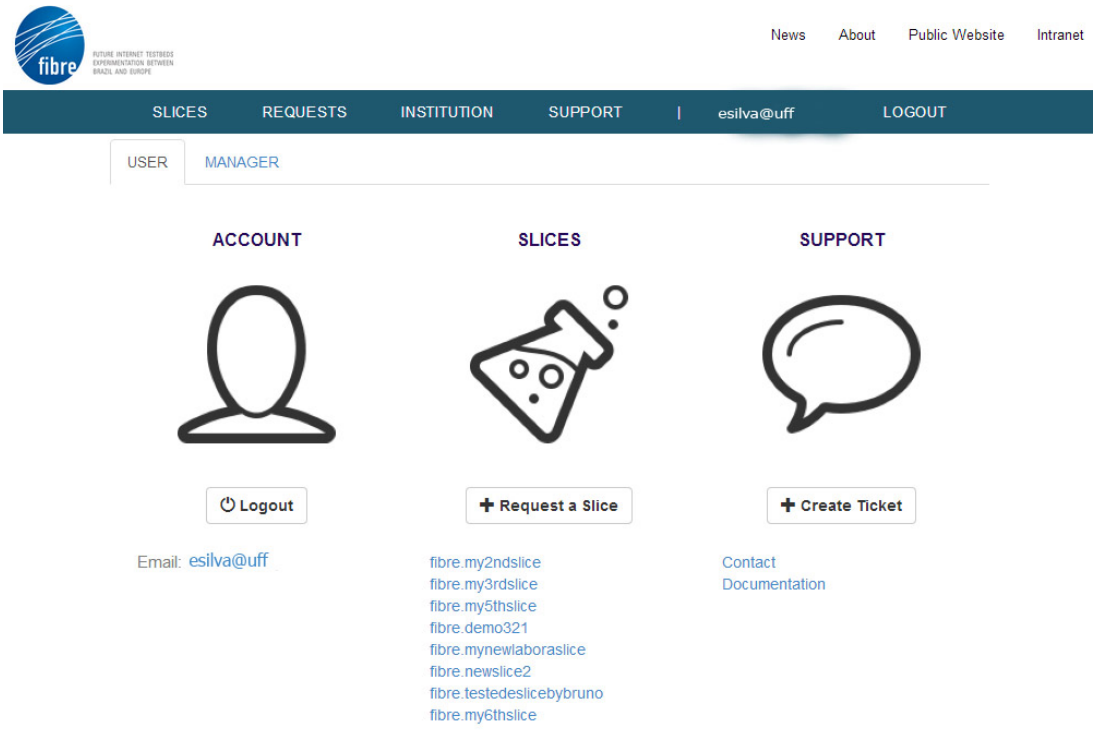

Figura 5.6. Usuário autenticado no portal do FIBRE com listagem de recursos (*slices*).

5.1.5 Access Control Module

Uma vez autenticado no portal do FIBRE e com sua credencial para a utilização da federação de recursos, o usuário é capaz de requisitar a listagem de recursos disponíveis. Selecionando os recursos, são encaminhados os atributos ao *Global ACROSS*, que

verificará se o usuário atende ao controle de acesso global. Lembrando, que para isso, será computada a pontuação do usuário, através de seus atributos, e também realizada a classificação dele em um nível, para assim ser verificada a política XACML conforme o nível do usuário e os recursos requisitados, como visto no Capítulo 4.

As políticas de cada ilha são verificadas através do *Local ACROSS*, caso o usuário atenda à política global. Uma vez que as políticas sejam atendidas pelo nível do usuário e recursos requisitados, o *SP Module* recebe essa resposta e envia o RSpec de reserva dos recursos para as ilhas, que efetivamente reservam tais recursos.

5.2 Organização Virtual Hipotética

5.2.1 Cenário de Validação

Para confirmar a motivação de propor um novo, flexível, e genérico *framework* de A&A, o ACROSS é validado em um cenário hipotético de OV. O cenário escolhido emula uma OV composta por três diferentes instituições, chamadas de Instituição 1 (Inst1), Instituição 2 (Inst2) e Instituição 3 (Inst3). Nessa OV, o objetivo principal é que as instituições ofereçam recursos compartilhados (VMs – *virtual machines*) para seus membros. A federação de identidade é baseada em SAML, implementando o Shibboleth versão 2 em um espelhamento da federação de identidade brasileira, a CAFe Expresso. Foi criada uma federação local para os experimentos, exportando máquinas virtuais do GIdLab [Wangham et al. 2013]. O GIdLab é um ambiente de experimentação da RNP[1] . O ambiente de validação é composto por VMs responsáveis por hospedar o SP, todos os módulos do ACROSS e as entidades da CAFe Expresso, o *Discovery Service* (DS) e o IdP.

A Figura 5.7 mostra o cenário de validação, onde um usuário acessa a OV através de uma interface web e autentica-se na federação de identidade. Um usuário da OV pode requisitar e reservar recursos na federação de recursos usando essa interface web, que representa seu SP, e portanto, tem o *SP Module* integrado. Outro ator nesse cenário é o administrador da OV, que é responsável por configurar os componentes globais do ACROSS através da interface de gerência do módulo *VO Manager*. O componente do *Local Module* do ACROSS se comunica com os módulos de controle de acesso e o agregador de atributos, além dos componentes locais, para tomar suas decisões sobre controle de acesso. Os componentes do *Local Module* do ACROSS representam a gestão

[1] http://www.rnp.br/

local de recursos e políticas. Outros atores neste cenário são os administradores locais, ou administradores das instituições, responsáveis por configurar os componentes em nível local. Outra entidade neste cenário é o gerenciador de recursos, responsável por solicitar e reservar recursos na federação de recurso. Esta entidade não faz parte do *framework* ACROSS.

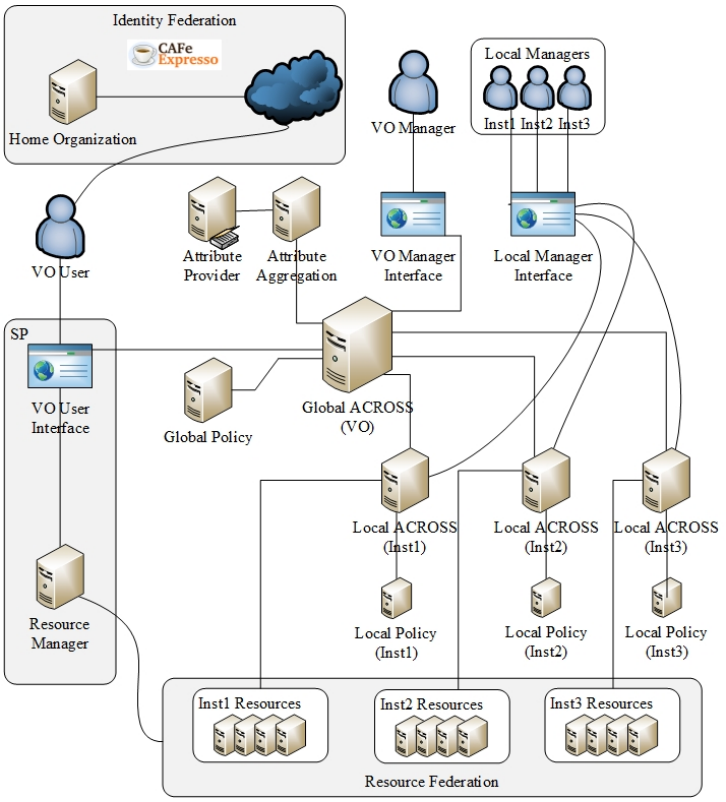

Figura 5.7. Cenário de validação do ACROSS.

5.2.2 Configuração de Políticas Globais do ACROSS

Usando a interface de gerência da OV, o administrador da OV pode configurar os parâmetros do ACROSS de nível global. Como mencionado, no nível global, o administrador da OV tem a capacidade de configurar a pontuação dos atributos, criar níveis de acesso, políticas e recursos suportados pela VO. Após acessar o sistema, como na Figura 5.8, o administrador é encaminhado para a interface do *VO Manager*, representado pela Figura 5.9. Nessa figura, pode-se ver as opções de configurações do nível global, *i.e.* pontuação de atributos (*Score Attributes*), níveis (*Levels*), recursos (*Resources*), políticas (*Policies*), e usuários da OV (*VO Users*). A figura mostra uma configuração da pontuação para atributos. Selecionando o atributo suportado pelo ACROSS, o administrador da OV pode definir seu operador de comparação (como visto no Capítulo 4), atribuir

seu valor, ponto e peso. Neste exemplo, a OV tem dois atributos adicionais, chamados *admin* e *position*. O atributo *admin* pode receber um valor de atributo (*true* ou *false*), e o atributo *position* pode receber valores como: *student*, *faculty*, etc. Além disso, a OV está configurada para suportar dois atributos da federação de identidade, chamado *Shib-eduPerson-eduPersonPrimaryAffiliation* e o *Shib-brEduPerson-brEntranceDate* . Além disso, é possível também usar o atributo interno *howManyTimes*, que corresponde às vezes que o usuário acesso a interface.

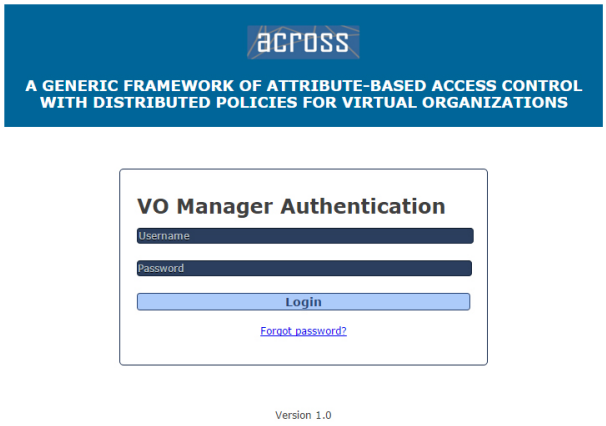

Figura 5.8. Acesso ao ACROSS Global.

Na primeira vez que um usuário da OV é autenticado através da federação de identidade, sua entrada é criada no provedor de atributos como um usuário desativado. Em seguida, o administrador da OV pode permitir seu acesso, gerenciando seus atributos adicionais através da opção *VO Users*.

Como é mostrado na Figura 5.10, a interface de gerenciamento global do ACROSS mostra um resumo da configuração, onde se vê todas as informações da OV configurada em nível global. Nessa OV, existem quatro pontuações configuradas para atributos, onde o valor máximo que um usuário pode alcançar é 220 pontos. É importante ressaltar que os valores de mínimo e o máximo que um usuário pode receber são calculados automaticamente com base na pontuação dos atributos configurados pelo administrador da OV e esses valores são necessários para realizar a normalização da pontuação do usuário.

Neste exemplo, uma máquina virtual é configurada como um recurso suportado pela plataforma do ACROSS. Três diferentes níveis de usuário são criados e três diferentes políticas globais, uma para cada nível de acesso. Por exemplo, se um usuário é classificado como nível 3, ele está autorizado a reservar até 10 VMs.

Um vídeo demonstrativo da configuração global do ACROSS pode ser visto em:

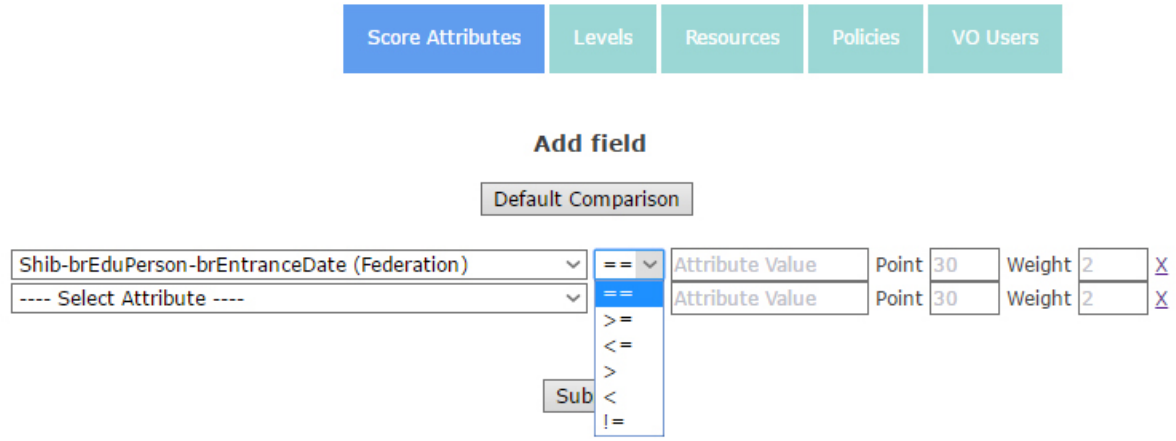

Figura 5.9. Resumo da configuração do ACROSS Global.

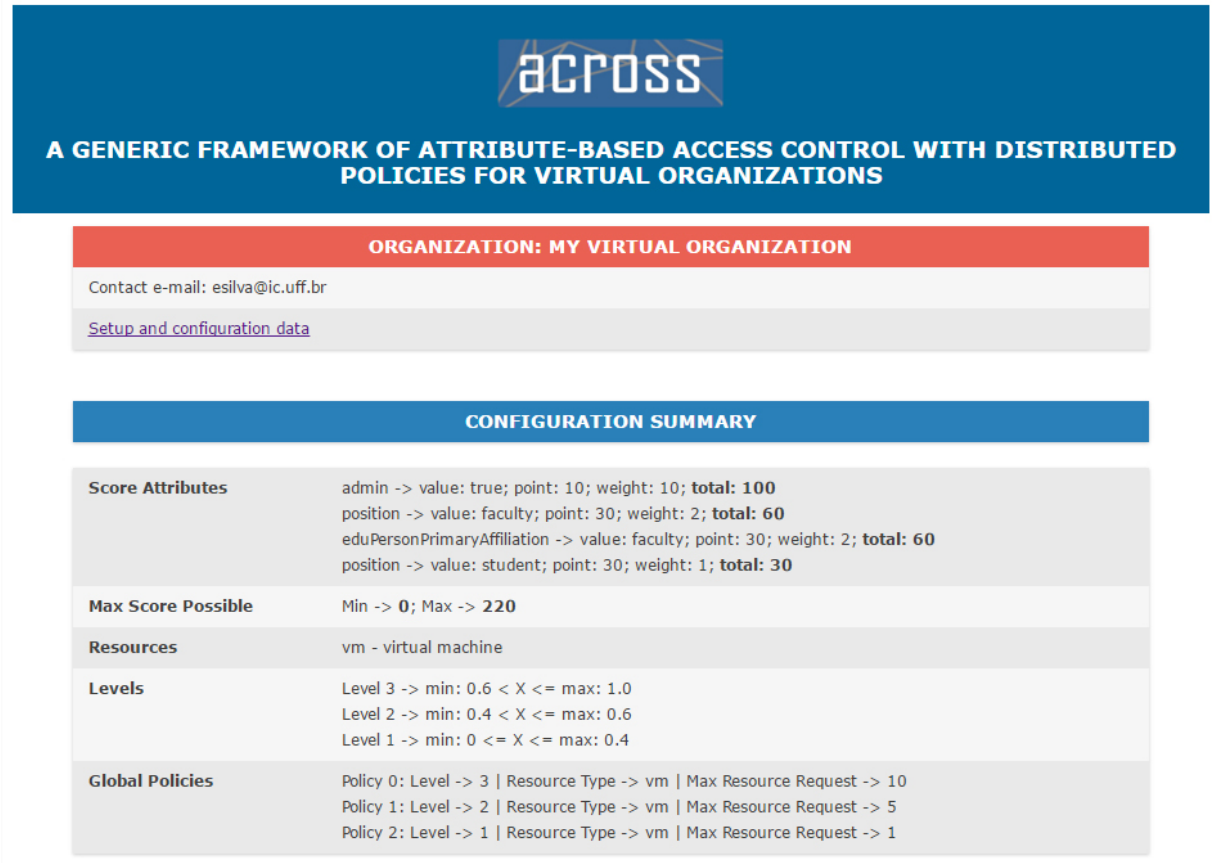

Figura 5.10. Interface de gerenciamento do ACROSS Global.

`http://www.midiacom.uff.br/across/tese/#global`

5.2.3 Configuração de Políticas Locais do ACROSS

Depois de introduzir a configuração do ACROSS em nível global, é demonstrado como um administrador local pode configurar seu ambiente. A Figura 5.11 mostra

um *screenshot* da interface de configuração para a Inst1 (o mesmo tipo de configuração pode ser realizada nas instituições Inst2 e Inst3). O nome da instituição, o tipo e número de recursos disponíveis são mostrados, além das políticas para controle de acesso que foram configuradas localmente. Neste exemplo, Inst1 tem três VMs para compartilhar e usuários com nível igual a 2 podem requisitar somente 2 VMs.

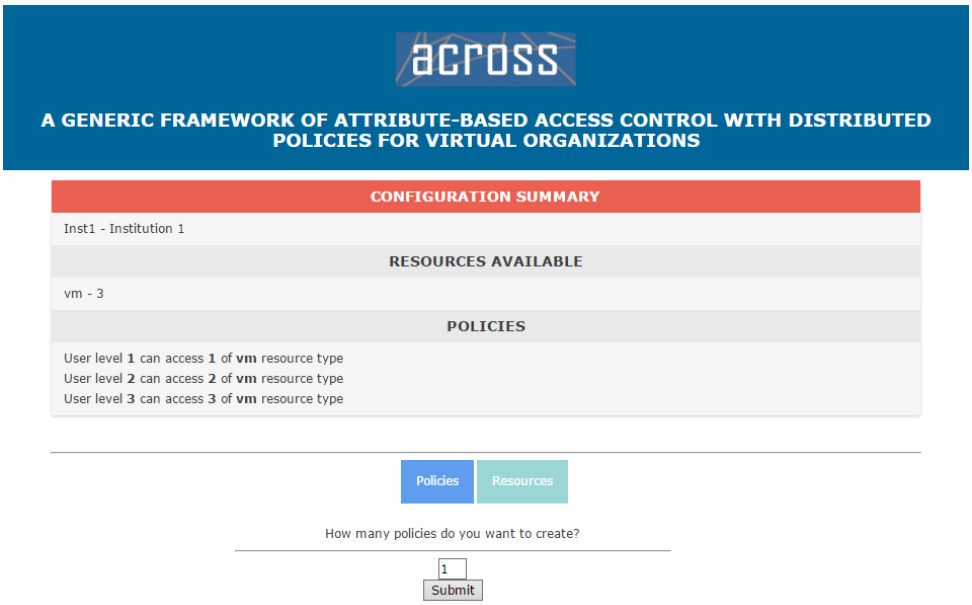

Figura 5.11. Resumo da configuração de uma instituição (Inst1).

Um vídeo demonstrativo da configuração de políticas locais do ACROSS pode ser visto em:
`http://www.midiacom.uff.br/across/tese/#local`

5.2.4 Exemplo de Controle de Acesso de Usuário

Nesta seção, um caso de uso será detalhado, baseado em uma interface de usuário da OV hipotética, implementado especialmente para validar as etapas de autenticação e autorização para reservar um determinado número de recursos. Esta demonstração apresenta cinco passos para a validação do ACROSS:

1. o usuário se autentica através da federação de identidade CAFe Expresso, Figura 5.12 e 5.13;

2. após autenticado, os atributos adicionais do usuário são recebidos e agregados pelo SP, classificando-o em um nível específico;

3. o usuário requisita a lista de recursos disponíveis nas instituições participantes da OV;

4. após receber a lista de recursos, o usuário informa o número de recursos que deseja requisitar;

5. para validar sua autorização, o nível do usuário é avaliado pela OV (nível global) através de suas políticas e, depois, nas instituições (nível local), para verificar se o usuário pode ou não reservar tais recursos.

Figura 5.12. Autenticação federada pela CAFe Expresso.

(a) WAYF na CAFe Expresso. (b) Autenticação do usuário no IdP.

Figura 5.13. *Screenshots* da interface do usuário.

O usuário, clicando no botão da CAFe Expresso, é, então, encaminhado pelo *SP Module* ao *SAML Module* (de forma transparente) e ao serviço WAYF (*Where Are You From*) para selecionar sua instituição de origem. Neste exemplo, conforme a Figura 5.13(a), Institution2 é a escolha do usuário como sua instituição de origem para a validação das credenciais.

Uma vez redirecionado para sua instituição de origem, o usuário informa suas credenciais, como pode ser visto na Figura 5.13(b). Se as credenciais estiverem corretas,

uma sessão SAML é criada, e seus atributos (requisitados pelo *SP Module*) são enviados ao módulo da interface do usuário da OV.

Caso o administrador da OV tenha habilitado o usuário para o acesso à OV, ele poderá ver seus atributos agregados, sua pontuação, e em qual nível foi classificado, como mostra a Figura 5.14.

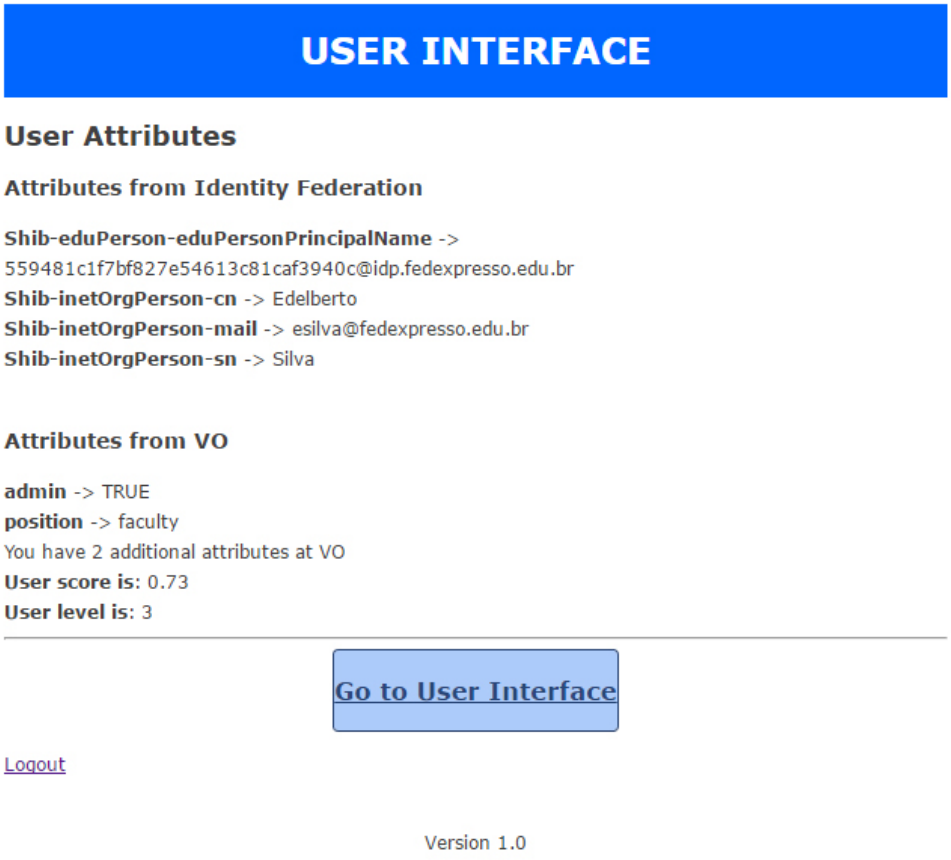

Figura 5.14. Interface do usuário após autenticação bem sucedida.

Após clicar em "Go to User Interface", o usuário pode ver um resumo de suas informações, como, seu nível e quantos recursos pode reservar, levando em consideração apenas as políticas globais. Ele pode requisitar uma lista de recursos para todas as instituições participantes da federação de recursos conforme o tipo de recurso desejado, clicando no botão "Request Resource List", como mostrado na Figura 5.15. Nesse ponto, o usuário ainda não realizou a requisição da lista de recursos e não tem nenhum recuso reservado a ele.

Na Figura 5.16, que fica disponível após a requisição da lista de recursos, há uma opção sobre a reserva de recursos, mostrando a lista de recursos de cada instituição. Neste exemplo, Inst1 tem 3 VMs, Inst2 tem 10 VMs, e Inst3 tem 2 VMs. Neste momento, o usuário pode informar quantos recursos ele quer e enviar sua solicitação ao

ACROSS.

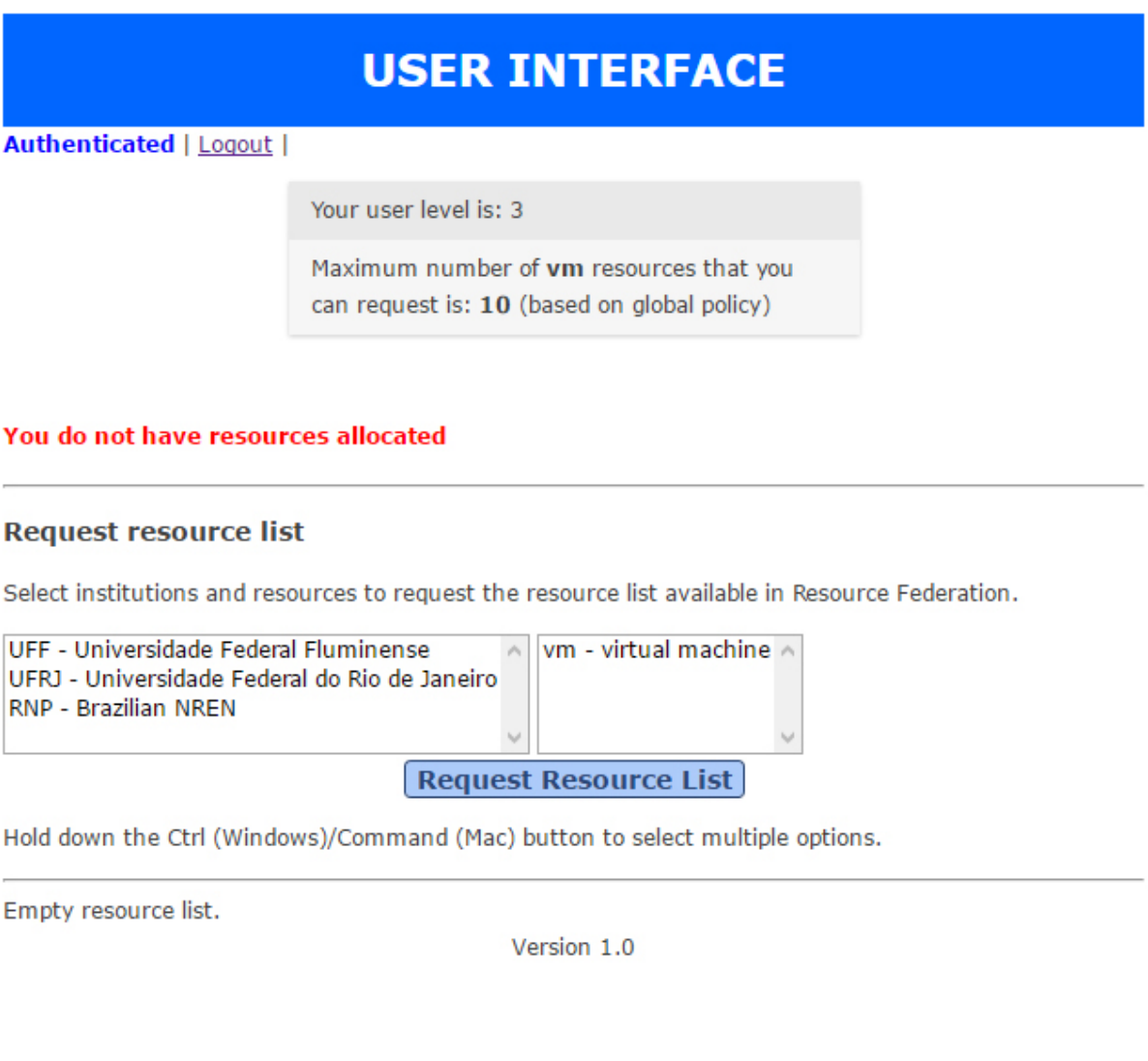

Figura 5.15. Interface do usuário.

Para permitir essa validação, foi criado um gerenciador de recursos central para listar, solicitar e reservar recursos. O *Resource Manager*, como é chamado, é responsável por armazenar os dados sobre todos os recursos de todas as instituições parceiras na federação de recursos, simulando a funcionalidade de uma federação de recursos real.

O usuário requisita recursos às instituições Inst1, Inst2, and Inst3, como é mostrado na Figura 5.17. Foi requisitada 1 VM à Inst1, 3 VMs à Inst2, e 2 VMs à Inst3, totalizando 6 VMs.

Neste exemplo, o usuário satisfaz todas as políticas globais e locais, o que, em seguida, gera uma resposta de reserva bem sucedida, como mostrado pela Figura 5.18. Uma vez que o usuário deseje liberar os recursos reservados, ele pode fazê-lo clicando em “Free resources reserved”.

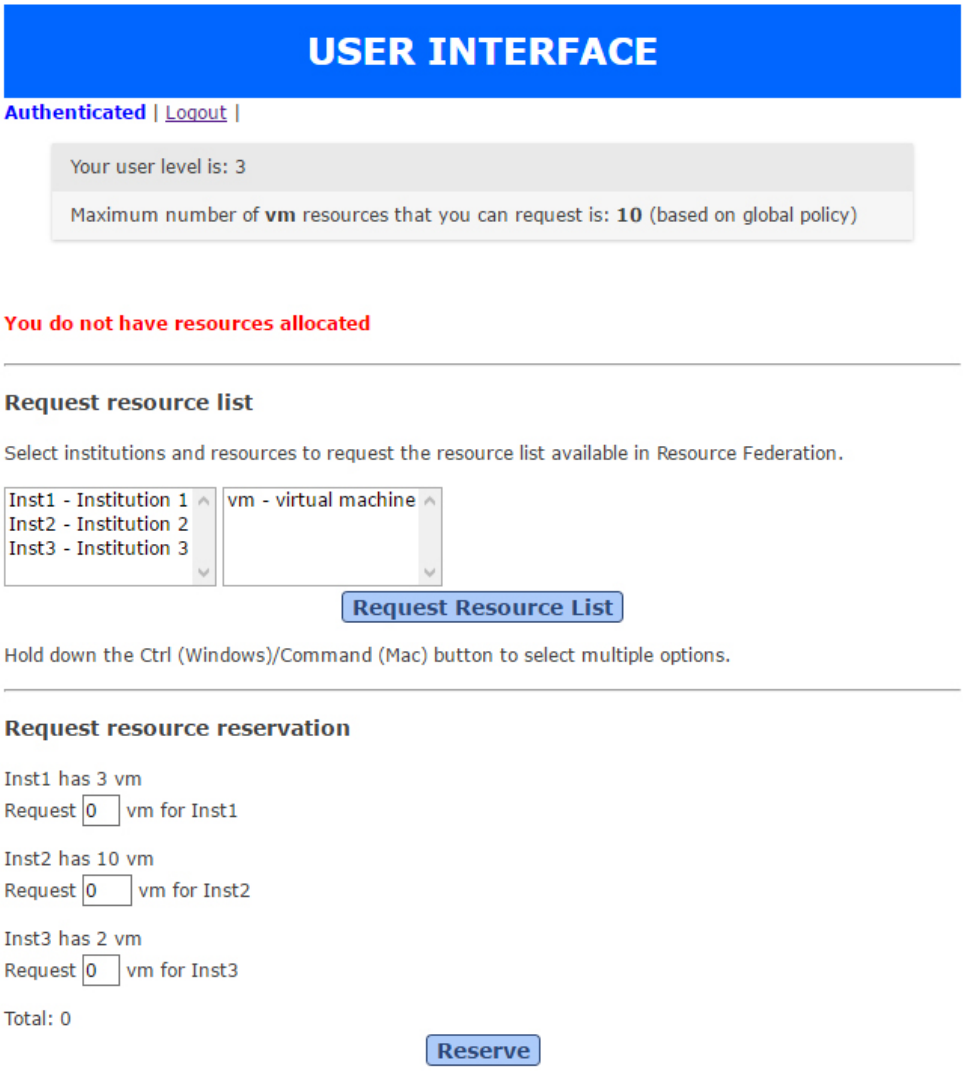

Figura 5.16. *Screenshot* da interface do usuário no ACROSS após requisitar lista de recursos.

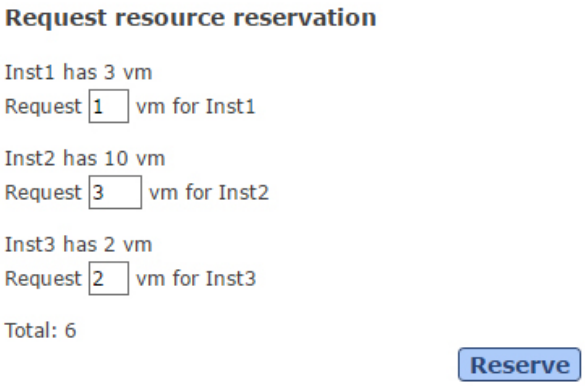

Figura 5.17. Usuário requisitando recursos para diferentes instituições.

Para completar nossa validação, é dado outro exemplo de requisição que não é bem sucedida. Na Figura 5.19, o usuário deseja requisitar 3 VMs à Inst1, 6 à Inst2 e 2 à Inst3, com um total de 11 VMs. De acordo com as políticas globais, configuradas pelo

Figura 5.18. Lista de recursos reservados.

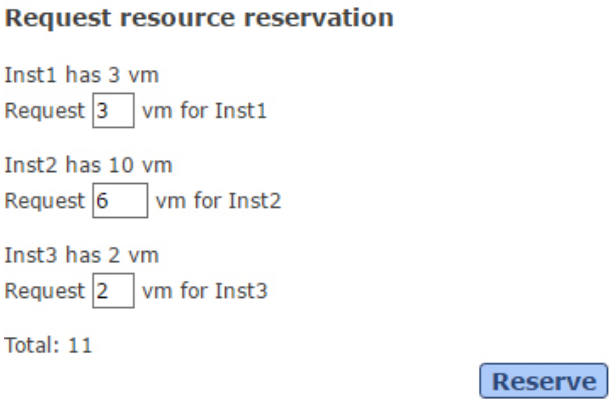

Figura 5.19. Usuário requisitando recursos de diferentes instituições.

administrador da OV neste exemplo, um usuário nível 3 pode requisitar no máximo 10 VMs. Então, uma vez feita a requisição, ela é negada já nas políticas globais, como podemos ver pela Figura 5.20.

Um vídeo demonstrativo para a reserva de recursos e o controle de acesso do usuário pode ser visto em:
`http://www.midiacom.uff.br/across/tese/#user`

A partir da implementação de uma OV hipotética, esta seção mostrou uma validação do ACROSS e sua implementação, ilustrando as interfaces de gerenciamento locais e global, a autenticação do usuário, a solicitação de acesso de recursos e reservas, provando que ACROSS funciona e pode ser facilmente utilizado como solução de A&A distribuída com base em atributos para organizações virtuais.

Figura 5.20. Mensagem de retorno informando que a requisição do usuário foi negada.

Capítulo 6

Conclusões

Esta tese propôs um novo *framework* para a gestão de identidade e acesso. Sua maior contribuição é facilitar tanto a entrada de entidades em uma organização virtual já existente, quanto a criação de uma nova OV. O ACROSS, ***A**ttribute-based access **C**ont**RO**l and di**S**tributed policie**S***, como foi chamado, contribuiu também ao estado da arte em gerência de organizações virtuais.

Diferentemente dos demais trabalhos propostos na literatura, o ACROSS se baseia em diversos conceitos de gestão de identidade e acesso para a construção de um *framework* integrado. Podemos destacar como principais facilidades do ACROSS o suporte à autenticação por federações de identidade, a facilidade a gerência de atributos específicos a uma organização virtual, através do conceito de provedores de atributos adicionais, a facilidade de transposição de credenciais entre o ambiente da federação de identidade e a organização virtual, e o controle de acesso utilizando políticas distribuídas e padrões baseados em papel e atributos. Outra facilidade é permitir o acesso a diferentes tipos de recursos compartilhados por uma organização virtual.

Comparado a outros trabalhos, ACROSS oferece mais funcionalidades relacionadas à autenticação e autorização para organizações virtuais. O framework também auxilia a entrada de uma nova instituição em uma organização virtual, independentemente de características particulares, como tipos específicos de credenciais ou mensagens de gerência de recursos.

A validação do ACROSS foi realizada através da implementação em um laboratório de experimentação em gestão de identidade real suportado pela RNP, o GIdLab, aplicando-o a uma organização virtual hipotética de recursos distribuídos.

6.1 Contribuições

É possível classificar, conforme os conceitos de IAM, as contribuições alcançadas pelo desenvolvimento do ACROSS. Para se tornar um *framework* completo de autenticação e autorização, foram necessários o desenvolvimento de cada um de seus módulos separadamente, a partir da experiência adquirida com trabalhos relacionados a A&A para redes de experimentação em Internet do Futuro.

Primeiramente, foi levada em consideração a integração da solução com uma federação de identidade. Assim, foram criadas soluções particulares, a princípio, para a federação CAFe, e depois desenvolvido o *Identity Federation Module* e o assistente de instalação e configuração para a integração de uma instituição a uma federação de identidade SAML/Shibboleth, o que agiliza a inclusão de uma VO, ou serviço. As publicações [Fernandes et al. 2013, Silva et al. 2013, Silva et al. 2014b] foram resultados deste estudo.

Após essa etapa, percebeu-se que no ambiente da OV, nem sempre todos os atributos necessários são disponibilizados por um IdP do usuário na federação de identidade, e que, com isso, ou se rompia o princípio da padronização dos atributos da federação de identidade e suas políticas (adicionando atributos ao IdP na federação), ou criavam-se novos cadastros, duplicando dados dos usuários no ambiente da OV, para poder adicionar esses novos atributos. Sendo assim, foi proposto o desenvolvimento do suporte a provedores de atributos adicionais, e um agregador de atributos, que unisse todos os atributos, tanto aqueles da federação de identidade, quanto os específicos da OV. Para isso, foi levado em consideração, com base na literatura, a necessidade também de se preservar a identidade do usuários nesses provedores adicionais, além de se ter um atributo persistente para identificação do usuário nesses provedores de atributos adicionais. Decidiu-se então, não apenas utilizar um atributo vindo da federação de identidade como identificador que correlacionasse o usuário nos dois ambientes, mas sim propor um atributo flexível ao administrador da OV e que ofuscasse aos demais a que usuário pertencem os atributos adicionais. Com isso, foi criado um atributo opaco, que é gerado a partir da combinação de um ou mais atributos da federação de identidade e uma sequência numérica particular à OV . Para facilitar a utilização e integração na OV, foi desenvolvido o *Attribute Module*, esse módulo incorporou todas essas funcionalidades e ainda possui um assistente para a instalação e configuração tanto de provedores de atributos adicionais como do agregador de atributos. A publicação [Silva et al. 2015a] é fruto deste estudo.

Outro conceito importante em uma OV é que o ambiente pode ser formado por diferentes tipos de recursos, e então, utilizar uma credencial atrelada a um tipo específico não é o ideal. Desta forma, foi proposto o *Credential Translation Module*. Esse módulo permite a geração de quaisquer tipos de credenciais, sejam elas quais forem (*e.g.* certificados X.509, *tokens* etc). O *Credential Translation Module* é, na verdade, um submódulo do *Identity Federation Module*, pois ele depende dos atributos advindos da federação de identidade e também daqueles consultados pelo *Identity Federation Module* ao *Attribute Module*. Com a disponibilidade dos atributos agregados do usuário de uma forma transparente, fica mais simples desenvolver qualquer tradutor de credenciais específicas a qualquer ambiente. Como resultado, foi publicado o trabalho [Silva et al. 2014a], e foi esse mesmo trabalho que motivou o desenvolvimento deste módulo.

O último módulo, que conclui a contribuição do ACROSS como um *framework* robusto para OV, é o de controle de acesso. Notou-se que a autenticação já havia sido bem trabalhada, e eram necessários avanços nas funções de autorização. Com isso, foram estudados os principais mecanismos de controle de acesso propostos na literatura, padronizados (ou formalmente propostos), e aplicados em diversos ambientes. Percebeu-se que era necessário aplicar os conceitos de políticas distribuídas, a fim de permitir que a administração seja menos onerosa à OV e às instituições que participantes. Sendo assim, foi proposto um modelo de políticas distribuídas em [Silva et al. 2014c], que focou em uma proposta para um ambiente de recursos distribuídos de testbeds para Internet do Futuro em [Silva et al. 2015a]. Neste último trabalho, foram aplicados a esse ambiente a integração com a federação de identidade CAFe, a agregação de atributos e as políticas distribuídas por meio do ABAC utilizando uma implementação de XACML. Então, o *Access Control Module* foi integrado à arquitetura do *framework* ACROSS. Para tanto, esse módulo deveria permitir a edição de políticas independentes, mas que respeitassem a hierarquia do ambiente de uma OV, criando assim os módulos global e local, onde no módulo global, o administrador da OV é o responsável, e pode, além de criar suas políticas globais, realizar um cadastro global de recursos e atributos. Isso é feito com a proposta de atribuir pontuação a cada atributo do usuário, níveis a essa pontuação e criar as políticas associadas a esses atributos. Com isso, o usuário é classificado em um nível conforme seus atributos, o que permite uma gerência facilitada do controle de acesso, além de permitir que, no nível local, a instituição não precise gerenciar ou realizar qualquer operação sobre os atributos do usuário, mas apenas criar políticas para os níveis existentes na OV. Essa forma de tratar atributos também aplicou conceitos de privacidade aos atributos do usuário, já que

somente no nível global da OV é que esses são conhecidos.

Com isso, foi então publicado o trabalho [Silva et al. 2015b], onde a modelagem do ACROSS foi inicialmente apresentada. Nesta tese, essa modelagem foi estendida e documentada por completo.

Comparado aos demais trabalhos relacionados, o ACROSS possui mais funcionalidades relacionadas à autenticação e autorização. Além de aplicar os conceitos mais atuais de IAM, ACROSS foi pensado para ser flexível e permitir sua extensão futura. Além disso, oferece funcionalidades adicionais como a instalação e configuração por meio de assistentes, o que agiliza e facilita a adesão de uma instituição a uma OV.

6.2 Trabalhos Futuros

Como próximos passos estão a utilização do ACROSS em um ambiente real de recursos compartilhados e distribuídos. O primeiro ambiente deverá ser o do projeto FIBRE. Esperamos utilizar o ACROSS ainda em outros ambientes, como em aplicações de arquivos compartilhados e distribuídos em *cloud*. Outra aplicação real em que desejamos utilizar o ACROSS é na integração com serviços em gerência de *cloud*, como soluções baseadas em OpenStack[1]. Ainda esperamos desenvolver uma plataforma com a documentação disponibilizada *online*, assim como o código fonte *open-source*.

Em outro trabalho futuro deseja-se considerar também atributos de recursos para a criação das políticas de controle de acesso, além de atributos dos usuário. Essa possibilidade é prevista pelo ABAC na entidade PIP, e deverá contribuir para uma maior granularidade de controle de acesso ao ambiente. Além disso, espera-se que um módulo de contabilidade (*i.e. accounting*) seja desenvolvido e integrado ao ACROSS. Também deseja-se a integração com outros métodos de autenticação, permitindo que usuários não pertencentes às federações acadêmicas baseadas em SAML/Shibboleth também possam utilizar o ambiente da OV. Entre esses métodos podemos citar o OpenID[2]. Porém, sabemos que será necessário levar em consideração o nível de confiança aos atributos fornecidos por essas federações, o LoA, e as políticas da OV.

Outra linha de pesquisa interessante é relacionada à hierarquia de políticas, realizando um estudo mais aprofundado sobre seu impacto. Atualmente o ACROSS utiliza a política mais restritiva, onde, caso seja negado o acesso ao recurso em alguma etapa

[1] https://www.openstack.org/
[2] http://openid.net/

da requisição, todas as demais políticas posteriores são negadas. Este algoritmo de combinação de políticas é utilizado pelo XACML, e conhecido como *deny-overrides*. Apesar de, no contexto atual do ACROSS, ser levado em consideração apenas o atributo de nível do usuário, se combinada esta proposta com a citada anteriormente, sobre a consideração dos atributos dos recursos, é possível gerar novas avaliações no nível da aplicação do XACML. Desta forma, é possível empregar políticas de acesso mais complexas e realizar a comparação da interferência dos resultados na hierarquia de políticas [Li et al. 2009, Ramli 2015].

A avaliação do desempenho e escalabilidade do XACML é também relevante para a verificação da escalabilidade do *framework*. O que é possível confirmar através de alguns trabalhos, como [Turkmen and Crispo 2008, Ulltveit-Moe and Oleshchuk 2012, Butler et al. 2011, Ilhan et al. 2015]. É interessante levar em consideração, durante o desenvolvimento desse estudo, também, a verificação do ganho com a utilização do conceito de *cache* para as políticas no ambiente da OV.

Deseja-se estudar qual o custo para a integração da autenticação utilizando outras federações de identidade da eduGAIN. A ideia é que, como já há suporte à autenticação CAFe, seja simples esta entrada como serviço na federação de federação, a eduGAIN. Como o ACROSS prevê suporte à transposição de credenciais e seu controle de acesso independe do tipo de recurso, acredita-se que ele possa ser adotado como um *framework* pra serviços em ambientes heterogêneos em projetos de escala global. Para tanto, será necessário avaliar o impacto para essa integração entre o ACROSS e a eduGAIN, verificando as políticas das diversas federações de identidade, identificando pontos em comum e requisitos básicos de atributos, credenciais, políticas de acesso e gerência de recursos.

Referências Bibliográficas

[gro 2016] (2016). Grouper – groups management toolkit. http://internet2.edu/grouper. Acessado em: 15 de Julho 2016.

[sim 2016] (2016). simpleSAMLphp. https://simplesamlphp.org/. Acessado em: 15 de Julho 2016.

[edu 2016a] (Acesso em Março de 2016a). *eduGAIN*. http://www.edugain.org.

[edu 2016b] (Acesso em Março de 2016b). *eduroam - EDUcation ROAMing*. http://www.eduroam.org.

[ama 2016] (Junho, 2016). *AWS Identity and Access Management (IAM)*. https://aws.amazon.com/pt/iam/.

[goo 2016] (Junho, 2016). *Google Cloud Identity & Access Management (IAM)*. https://cloud.google.com/iam/.

[one 2013] (MarÃ§o, 2013). *OneLab - Future Internet Testbeds*. http://www.onelab.eu/.

[Afsarmanesh and Camarinha-Matos 2005] Afsarmanesh, H. and Camarinha-Matos, L. M. (2005). *A Framework for Management of Virtual Organization Breeding Environments*, pages 35–48. Springer US, Boston, MA.

[Alanen and Porres 2003] Alanen, M. and Porres, I. (2003). *UML 2003 - The Unified Modeling Language*, volume 2863 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, Berlin, Heidelberg.

[Alfieri et al. 2003] Alfieri, R. et al. (2003). Managing dynamic user communities in a grid of autonomous resources. *CoRR*, cs.DC/0306004.

[Amazon 2016] Amazon (2016). Amazon Web Services. https://aws.amazon.com. Acessado em: 15 de Julho 2016.

[Apereo 2016] Apereo (2016). CAS – Central Authentication Service. https://apereo.github.io/cas/4.2.x/index.html. Acessado em: 10 de Julho 2016.

[Barton et al. 2006] Barton, T., Basney, J., Freeman, T., Scavo, T., Siebenlist, F., Welch, V., Ananthakrishnan, R., Baker, B., Goode, M., and Keahey, K. (2006). Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, Gridshib, and MyProxy. In *5th Annual PKI R&D Workshop*.

[Basney et al. 2005] Basney, J., Humphrey, M., and Welch, V. (2005). The myproxy online credential repository. *Softw., Pract. Exper.*, 35(9):801–816.

[Berman et al. 2014] Berman, M., Chase, J. S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., Ricci, R., and Seskar, I. (2014). Geni: A federated testbed for innovative network experiments. *Comput. Netw.*, 61:5–23.

[Bhargav-Spantzel et al. 2007] Bhargav-Spantzel, A., Camenisch, J., Gross, T., and Sommer, D. (2007). User centricity: A taxonomy and open issues. *J. Comput. Secur.*, 15(5):493–527.

[Butler et al. 2011] Butler, B., Jennings, B., and Botvich, D. (2011). An experimental testbed to predict the performance of xacml policy decision points. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 353–360.

[Camarinha-Matos and Afsarmanesh 2005] Camarinha-Matos, L. M. and Afsarmanesh, H. (2005). Collaborative networks: a new scientific discipline. *Journal of intelligent manufacturing*, 16(4-5):439–452.

[Chadwick and Inman 2009] Chadwick, D. and Inman, G. (2009). Attribute aggregation in federated identity management. *Computer*, 42(5):33–40.

[Chadwick et al. 2003] Chadwick, D., Otenko, A., and Ball, E. (2003). Role-based access control with x.509 attribute certificates. *Internet Computing, IEEE*, 7(2):62–69.

[Chadwick 2009] Chadwick, D. W. (2009). Foundations of security analysis and design v. chapter Federated Identity Management, pages 96–120. Springer-Verlag, Berlin, Heidelberg.

[Crawford et al. 2003] Crawford, C. H., Dias, D. M., Iyengar, A. K., Novaes, M., Zhang, L., Crawford, C., Dias, D., Iyengar, A., and Novaes, M. (2003). Commercial applications of grid computing.

[de Mello et al. 2009] de Mello, E. R. et al. (2009). A model for authentication credentials translation in service oriented architecture. *Transactions on Computational Science*, 4:68–86.

[Dhungana et al. 2013] Dhungana, R., Mohammad, A., Sharma, A., and Schoen, I. (2013). Identity management framework for cloud networking infrastructure. In *Innovations in Information Technology (IIT), 2013 9th International Conference on*, pages 13–17.

[Farrell and Housley 2002] Farrell, S. and Housley, R. (2002). An Internet Attribute Certificate Profile for Authorization. RFC 3281 (Proposed Standard). Obsoleted by RFC 5755.

[Fernandes et al. 2013] Fernandes, N. C., Silva, E., Muchaluat-Saade, D., and Magalhaes, L. (2013). Gestão de identidade em testbeds brasileiros para a internet do futuro. In *SBRC 2013 - WPEIF*, Brasilia.

[Ferraiolo et al. 2001] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R. (2001). Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274.

[FIBRE 2016] FIBRE (2016). Future internet brazilian environment for experimentation. https://fibre.org.br/. Acessado em: 10 de Agosto de 2016.

[Fielding 2000] Fielding, R. (2000). Representational state transfer. *Architectural Styles and the Design of Netowork-based Software Architecture*, pages 76–85.

[Foster and Kesselman 1996] Foster, I. and Kesselman, C. (1996). Globus: A metacomputing infrastructure toolkit. *International Journal of Supercomputer Applications*, 11:115–128.

[Foster et al. 1998] Foster, I., Kesselman, C., Tsudik, G., and Tuecke, S. (1998). A security architecture for computational grids. In *Proceedings of the 5th ACM Conference on Computer and Communications Security*, CCS '98, pages 83–92, New York, NY, USA. ACM.

[Foster et al. 2001a] Foster, I., Kesselman, C., and Tuecke, S. (2001a). The anatomy of the grid - enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15:2001.

[Foster et al. 2001b] Foster, I., Kesselman, C., and Tuecke, S. (2001b). The anatomy of the grid: Enabling scalable virtual organizations. *Int. J. High Perform. Comput. Appl.*, 15(3):200–222.

[Friedman and Gavras 2011] Friedman, T. and Gavras, A. (2011). Fire openlab ip testbed and tool demo. In *Proceedings of the 4th European conference on Towards a service-based internet*, ServiceWave'11, pages 323–324, Berlin, Heidelberg. Springer-Verlag.

[Gaedke et al. 2005] Gaedke, M., Meinecke, J., and Nussbaumer, M. (2005). A modeling approach to federated identity and access management. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1156–1157. ACM.

[Gagliardi 2004] Gagliardi, F. (2004). The egee european grid infrastructure project. In *High Performance Computing for Computational Science-VECPAR 2004*, pages 194–203. Springer.

[Garcia et al. 2013] Garcia, A. L., del Castillo, E. F., and Puel, M. (2013). Identity federation with voms in cloud infrastructures. In *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, volume 1, pages 42–48.

[Gomes et al. 2015] Gomes, A. T. A., Bastos, B. F., Medeiros, V., and Moreira, V. M. (2015). Experiences of the brazilian national high-performance computing network on the rapid prototyping of science gateways. *Concurrency and Computation: Practice and Experience*, 27(2):271–289.

[Group 2006] Group, N. W. (2006). Comment on rfc 4516 - lightweight directory access protocol (ldap).

[Gunter et al. 2011] Gunter, C. A., Liebovitz, D., and Malin, B. (2011). Experience-based access management: A life-cycle framework for identity and access management systems. *IEEE security & privacy*, 9(5):48.

[Hardt 2012] Hardt, D. (2012). The OAuth 2.0 Authorization Framework. RFC 6749 (Proposed Standard).

[Hu et al. 2013] Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., Hu, V. C., Ferraiolo, D., Kuhn, R., Friedman, A. R., Lang, A. J., Cogdell, M. M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., and Cybersecurity, S. (2013). Guide to attribute based access control (abac) definition and considerations (draft).

[Hunt et al. 2015a] Hunt, P., Grizzle, K., Ansari, M., Wahlstroem, E., and Mortimore, C. (2015a). System for Cross-domain Identity Management: Protocol. RFC 7644 (Proposed Standard).

[Hunt et al. 2015b] Hunt, P., Grizzle, K., Wahlstroem, E., and Mortimore, C. (2015b). System for Cross-domain Identity Management: Core Schema. RFC 7643 (Proposed Standard).

[IBM 2016] IBM (2016). High Performance Distributed Scheduling with IBM Platform Symphony. `http://www-03.ibm.com/systems/platformcomputing/products/symphony/`. Acessado em: 15 de Julho 2016.

[Ilhan et al. 2015] Ilhan, . M., Thatmann, D., and KÃ¼pper, A. (2015). A performance analysis of the xacml decision process and the impact of caching. In *2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, pages 216–223.

[ISO 2011] ISO (2011). ISO/IEC 10181-3:1996 - Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework.

[ITU-T 2009] ITU-T (2009). NGN identity management framework. Recommendation Y.2720.

[ITU-T 2012] ITU-T (2012). The directory: Public-key and attribute certificate frameworks. Technical Report X.509, ITU-T SG17, Geneva, Switzerland.

[Jensen 2012] Jensen, J. (2012). Federated identity management challenges. In *Availability, Reliability and Security (ARES), 2012 Seventh International Conference on*, pages 230–235.

[Jin et al. 2012] Jin, X., Krishnan, R., and Sandhu, R. (2012). A unified attribute-based access control model covering dac, mac and rbac. In *Proceedings of the 26th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy*, DBSec'12, pages 41–55, Berlin, Heidelberg. Springer-Verlag.

[Köpsel and Woesner 2011] Köpsel, A. and Woesner, H. (2011). Ofelia: pan-european test facility for openflow experimentation. In *Proceedings of the 4th European conference on Towards a service-based internet*, ServiceWave'11, pages 311–312, Berlin, Heidelberg. Springer-Verlag.

[Lab 2016] Lab, L. B. N. (2016). Akenti – distributed pki-based authorization system. `http://dst.lbl.gov/ACSSoftware/Akenti/`. Acessado em: 15 de Julho 2016.

[Lee et al. 2014] Lee, C. A., Desai, N., and Brethorst, A. (2014). A keystone-based virtual organization management system. In *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, pages 727–730.

[Li et al. 2009] Li, N., Wang, Q., Qardaji, W., Bertino, E., Rao, P., Lobo, J., and Lin, D. (2009). Access control policy combining: theory meets practice. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 135–144. ACM.

[Meirosu et al. 2005] Meirosu, C., Golonka, P., Hirstius, A., Stancu, S., Dobinson, B., Radius, E., Antony, A., Dijkstra, F., Blom, J., and de Laat, C. (2005). Native 10gigabit ethernet experiments over long distances. *Future Generation Computer Systems*, 21(4):457–468.

[Microsoft 2016] Microsoft (2016). Microsoft Azure. `https://azure.microsoft.com`. Acessado em: 15 de Julho 2016.

[Moses 2005] Moses, T. (2005). eXtensible Access Control Markup Language TC v2.0 (XACML).

[MySlice 2013] MySlice (2013). `http://www.myslice.info`.

[OASIS 2005] OASIS (2005). Security assertion markup language (saml) v2.0.

[Pearlman et al. 2002] 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.

[Pérez-Méndez et al. 2014] Pérez-Méndez, A., Pereńiguez-GarcÃa, F., Marín-López, R., López-Millán, G., and Howlett, J. (2014). Identity federations beyond the web: A survey. *IEEE Communications Surveys Tutorials*, 16(4):2125–2141.

[Peterson et al. 2008] Peterson, L., Ricci, R., Falk, A., and Chase, J. (2008). Slice-based facility architecture. Technical report.

[Peterson et al. 2010] Peterson, L., Ricci, R., Falk, A., and Chase, J. (2010). Slice-based federation architecture. Technical report.

[Rae et al. 2012] Rae, L., Recordon, D., and Messina, C. (2012). *OpenID: The Definitive Guide*. Oreilly & Associates Inc, 1st edition.

[Rakotoarivelo et al. 2010] Rakotoarivelo, T., Ott, M., Jourjon, G., and Seskar, I. (2010). OMF: a control and management framework for networking testbeds. *SIGOPS Oper. Syst. Rev.*, 43(4):54–59.

[Ramli 2015] Ramli, C. D. P. K. (2015). Detecting incompleteness, conflicting and unreachability xacml policies using answer set programming. *arXiv preprint arXiv:1503.02732*.

[RNP 016s] RNP (2016s). CAFe - Federated Academic Community. `http://portal.rnp.br/web/servicos/cafe-en`.

[Romberg 2002] Romberg, M. (2002). The unicore grid infrastructure. *Scientific Programming*, 10(2):149–157.

[Rountree 2012] Rountree, D. (2012). *Federated Identity Primer*. Elsevier Science.

[Sallent et al. 2012] Sallent, S., Abelem, A., Machado, I., Bergesio, L., Fdida, S., Rezende, J., Simeonidou, D., Salvador, M., Ciuffo, L., Tassiulas, L., and Bermudo, C. (2012). FIBRE project: Brazil and Europe unite forces and testbeds for the Internet of the future. In *Proceedings of TridentCom 2012*.

[Sandhu 1993] Sandhu, R. S. (1993). Lattice-based access control models. *Computer*, 26(11):9–19.

[Scavo 2005] Scavo, T. e Cantor, S. (2005). Shibboleth architecture. Technical report.

[Sharma et al. 2016] Sharma, D. H., Dhote, C., and Potey, M. M. (2016). Identity and access management as security-as-a-service from clouds. *Procedia Computer Science*, 79:170–174.

[Silva et al. 2014a] Silva, E., Fernandes, N., Rodriguez, N., and Muchaluat-Saade, D. (2014a). Credential translations in future internet testbeds federation. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–6.

[Silva et al. 2015a] Silva, E., Fernandes, N. C., and Muchaluat-Saade, D. (2015a). Across-fi: Attribute-based access control with distributed policies for future internet testbeds. In *14th International Conference on Networking*, pages 198–204, Barcelona/Spain.

[Silva et al. 2014b] Silva, E., Fernandes, N. C., Rodriguez, N., and Muchaluat-Saade, D. (2014b). Gestão de identidade em testbeds de internet do futuro baseada em federacoes a&a academicas. In *CSBC 2014 - SEMISH ()*, Brasilia, Brazil.

[Silva et al. 2013] Silva, E., Muchaluat-Saade, D., and Fernandes, N. C. (2013). Transposicão de credenciais para uso de testbeds para a internet do futuro. In *SBSeg 2013 - WGID*, Manaus.

[Silva et al. 2014c] Silva, E., Muchaluat-Saade, D., and Fernandes, N. C. (2014c). Controle de acesso baseado em políticas e atributos para federaćões de recursos. In *SBSeg 2014 - WGID*, Belo Horizonte.

[Silva et al. 2015b] Silva, E., Muchaluat-Saade, D., and Fernandes, N. C. (2015b). Modelagem do across: Um arcaboućo de a&a baseado em politicas e atributos para organizacoes virtuais. In *SBSeg 2015 WGID ()*, Florianopolis - SC.

[Spence et al. 2006] Spence, D., Geddes, N., Jensen, J., Richards, A., Viljoen, M., Martin, A., Dovey, M., Norman, M., Tang, K., Trefethen, A., Wallom, D., Allan, R., and Meredith, D. (2006). ShibGrid: Shibboleth access for the UK National Grid Service. In *eScience 2006, Amsterdam*.

[Sturrus and Kulikova 2016] Sturrus, E. and Kulikova, O. (2016). Identity and access management. *Encyclopedia of Cloud Computing*, pages 396–405.

[Szegedi et al. 2009] Szegedi, P., Figuerola, S., Campanella, M., Maglaris, V., and Cervelló-Pastor, C. (2009). With evolution for revolution: managing federica for future internet research. *Comm. Mag.*, 47(7):34–39.

[Thomas and Chandrasekaran 2016] Thomas, M. V. and Chandrasekaran, K. (2016). Identity and access management in the cloud computing environments. *Developing Interoperable and Federated Cloud Architecture*, page 61.

[Torres et al. 2013] Torres, J., Nogueira, M., and Pujolle, G. (2013). A survey on identity management for the future network. *IEEE Communications Surveys Tutorials*, 15(2):787–802.

[Turkmen and Crispo 2008] Turkmen, F. and Crispo, B. (2008). Performance evaluation of xacml pdp implementations. In *Proceedings of the 2008 ACM Workshop on Secure Web Services*, SWS '08, pages 37–44, New York, NY, USA. ACM.

[Ulltveit-Moe and Oleshchuk 2012] Ulltveit-Moe, N. and Oleshchuk, V. (2012). Decision-cache based {XACML} authorisation and anonymisation for {XML} documents. *Computer Standards & Interfaces*, 34(6):527 – 534. Intelligent DAQ's, Advanced Computing and Interfacing Systems.

[Vandenberghe et al. 2013] Vandenberghe, W., Vermeulen, B., Demeester, P., Willner, A., Papavassiliou, S., Gavras, A., Sioutis, M., Quereilhac, A., Al-Hazmi, Y., Lobillo, F., Schreiner, F., Velayos, C., Vico-Oton, A., Androulidakis, G., Papagianni, C., Ntofon, O., and Boniface, M. (2013). Architecture for the heterogeneous federation of Future Internet experimentation facilities. In Fiedler, M., Marques, P., Simeonidou, D., Vanelli, A., and Vermesan, O., editors, *Future Network and Mobile Summit, 2013*, Los Alamitos, CA, USA. IEEE.

[Vinicius G. Pinheiro 2015] Vinicius G. Pinheiro, Maria Julia de Lima, R. M. e. N. R. (2015). Suporte a organizaćões virtuais para autenticaćão e controle de acesso no csgrid. In *SBSeg 2015 - WGID*, Florianopolis.

[VOOT API/Internet2 2016] VOOT API/Internet2 (2016). Virtual Organisation Orthogonal Technology. `https://spaces.internet2.edu/display/COmanage/VOOT+API`. Acessado em: 05 de Julho 2016.

[W3C 2008] W3C (2008). Extensible markup language (xml) 1.0 (fifth edition).

[W3C 2016] W3C (2016). SOAP - Simple Object Access Protocol. https://www.w3.org/TR/soap/. Acessado em: 15 de Julho 2016.

[Wang et al. 2009] Wang, X. D., Jones, M., Jensen, J., Richards, A., Wallom, D., Ma, T., Frank, R., Spence, D., Young, S., Devereux, C., and Geddes, N. (2009). Shibboleth access for resources on the national grid service (sarongs). In *Information Assurance and Security, 2009. IAS '09. Fifth International Conference on*, volume 2, pages 338–341.

[Wangham et al. 2010] 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.

[Wangham et al. 2013] Wangham, M. S., Mello, E. R., Souza, M. C., and Coelho, H. (2013). Gidlab: Laboratorio de experimentacao em gestao de identidades. *Anais XIII Simposio Brasileiro em Seguranca da Informacao e de Sistemas Computacionais*, page 481â486.

[Wen et al. 2012] Wen, X., Gu, G., Li, Q., Gao, Y., and Zhang, X. (2012). Comparison of open-source cloud management platforms: Openstack and opennebula. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 2457–2461.

[Xu et al. 2005] Xu, W., Chadwick, D., and Otenko, S. (2005). Development of a flexible permis authorisation module for shibboleth and apache server. In Chadwick, D. and Zhao, G., editors, *Public Key Infrastructure*, volume 3545 of *Lecture Notes in Computer Science*, pages 162–179. Springer Berlin Heidelberg.

[Zhu et al. 2014] Zhu, Y., Huang, D., Hu, C., and Wang, X. (2014). From rbac to abac: Constructing flexible data access control for cloud storage services. *Services Computing, IEEE Transactions on*, PP(99):1–1.

Apêndices

APÊNDICE A – Documentação Complementar

A instalação e configuração é uma etapa muito importante na criação ou adesão de uma instituição a uma organização virtual. O ACROSS oferece suporte à instalação e configuração através de assistentes que facilitam a entrada de uma OV em uma federação de identidade, instalando e configurando todos os softwares e dependências necessárias para o ingresso em uma federação baseada em SAML. Os assistentes do ACROSS, também facilitam a criação de provedores de atributos da OV, a configuração desses provedores de atributos e do agregador de atributos. Chamado de *ACROSS Wizards*, esses assistentes podem ser identificados na Figura 4.2. O ACROSS também fornece uma ferramenta web para o gerenciamento das políticas, já equivalente ao *VO Manager*.

Na Figura A.1 é possível ver o diagrama de classes para a instalação e configuração dos pacotes, módulos e serviços, equivalente ao *ACROSS Wizards*.

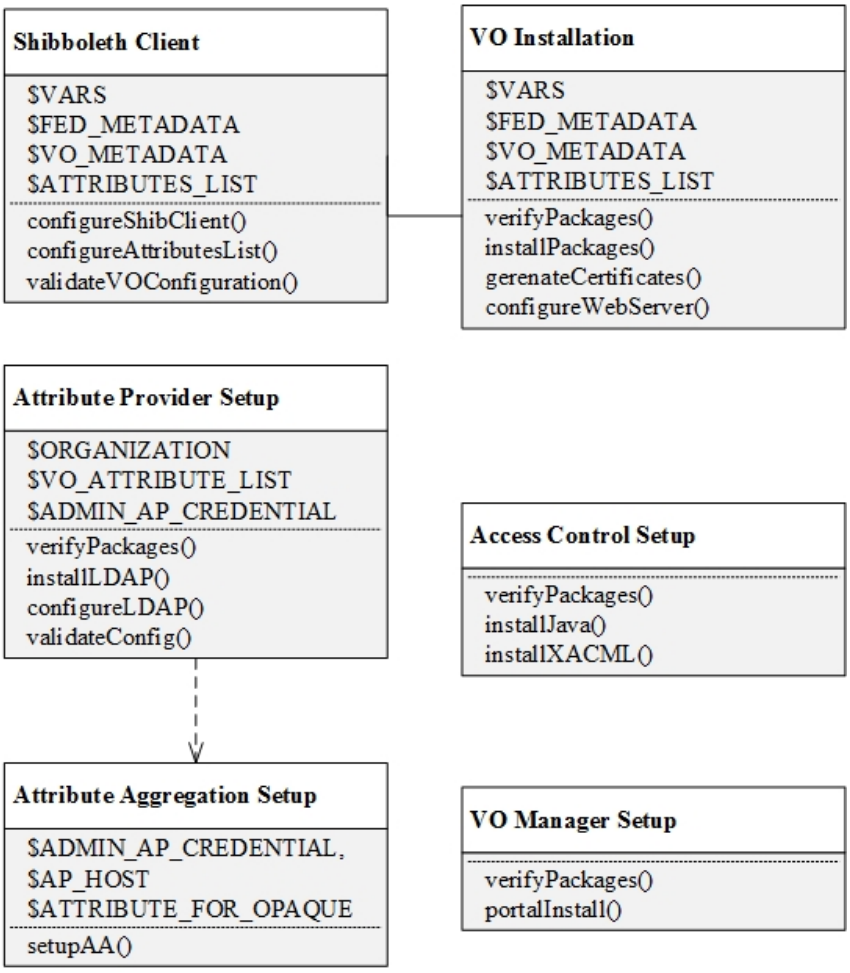

Figura A.1. Diagrama de classes do ACROSS.

A.1 Identity Federation Module

O diagrama de casos de uso da Figura A.2 mostra as possíveis tarefas do administrador da OV, onde tem-se representada a instalação e configuração dos serviços referentes à comunicação da OV com a federação de identidade e a autenticação através dessa federação, a fim de validar a configuração do módulo.

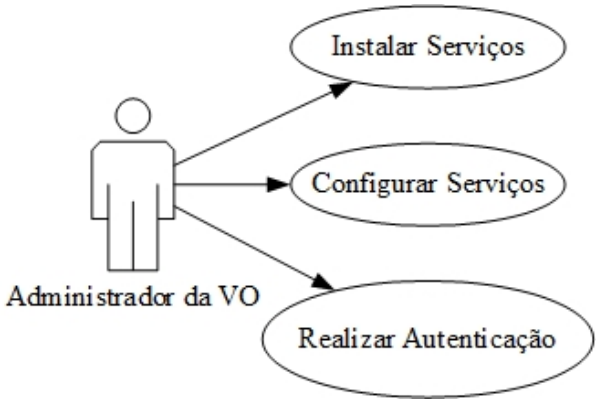

Figura A.2. Casos de uso para administrador da OV.

Já o diagrama de sequência para a instalação do *Identity Federation Module* é detalhado pela Figura A.2, onde há três componentes: o "VO Installation", responsável pela instalação e configuração do módulo de federação de identidade do *framework* ACROSS; o "Shibboleth Client", que deverá ser configurado como Provedor de Serviço (SP – *Service Provider*); e o componente "Identity Federation", que representa a federação baseada em Shibboleth. Neste diagrama, apresentam-se as atividades de instalação dos serviços básicos necessários a este módulo e também a sua configuração, realizando desde o suporte a atributos necessários para a OV, quanto gerando o metadado necessário para entrada deste serviço na federação de identidade.

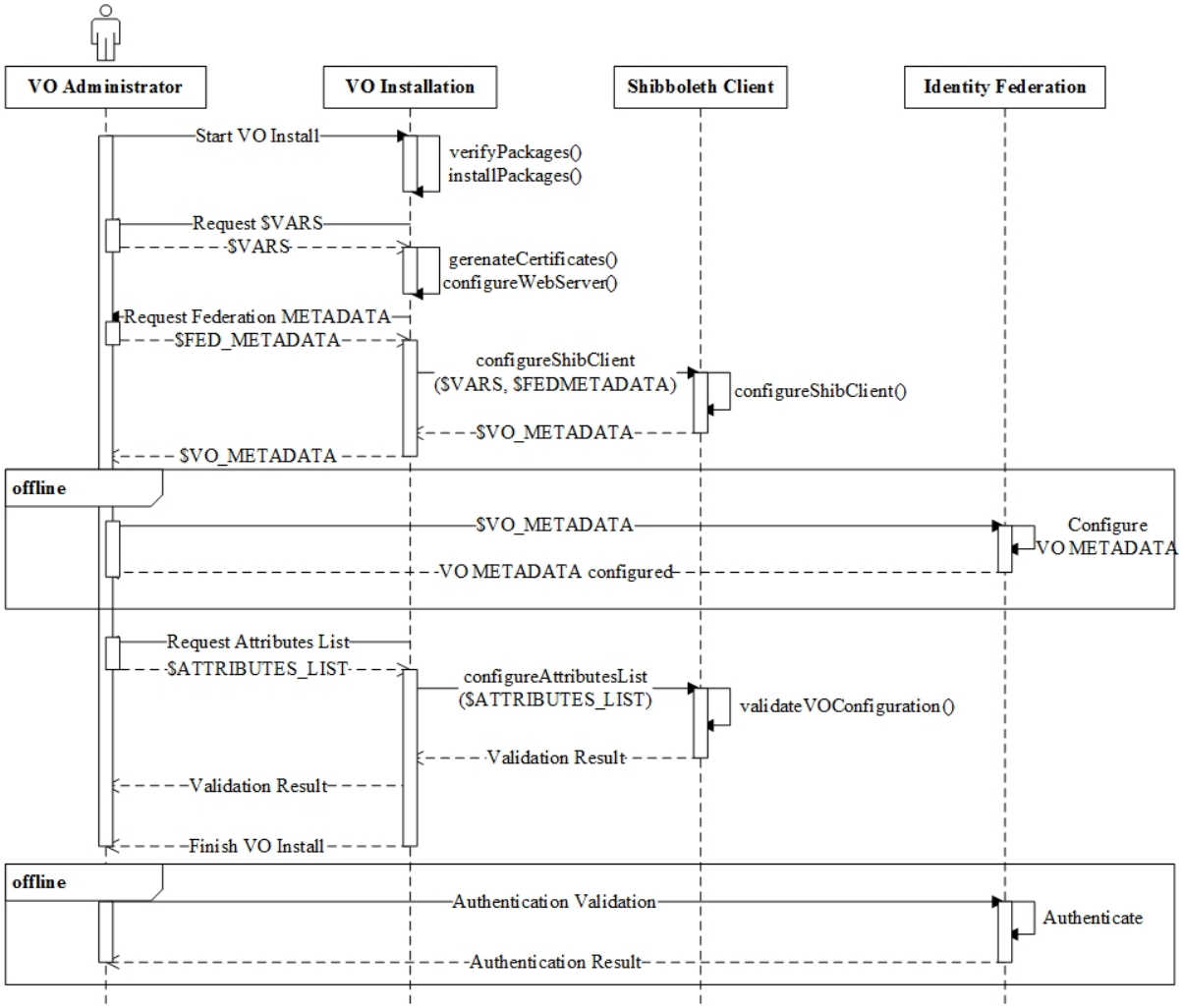

Figura A.3. Diagrama de sequência para instalação e configuração do *Identity Federation Module*.

É necessário, a princípio, ter o metadado[1] da federação de identidade e, seguindo o diagrama, tem-se: o início da instalação pelo administrador da OV, através do "Start VO Install", passando pela instalação dos pacotes necessários para os serviços de servidor web e cliente Shibboleth, quando o administrador deverá preencher algumas informações. Em seguida, é solicitado ao administrador o metadado da federação, que será configurado no cliente shibboleth do módulo e então gerado o metadado da OV, enviado ao administrador. Por sua vez, o administrador deve entrar em contato com a federação de identidade para inserir o metadado de sua OV, a fim de iniciar efetivamente sua participação naquela federação. Logo após realizados tais passos, o administrador deverá selecionar quais atributos, da lista de possíveis atributos, ele necessita suportar, e então é feita a sua configuração e confirmada ao administrador.

[1] Metadado é um termo genérico, mas em relação a SAML (e Shibboleth), refere-se à configuração dos dados utilizados para promover a comunicação entre um SP ou IdP uns com os outros. Neste caso, em formato XML.

Por fim, o administrador utiliza o módulo para realizar sua autenticação na federação de identidade, com o objetivo de validar a configuração realizada, e então finaliza a configuração.

O passo citado no diagrama da Figura A.3 como "*Authentication Validation*", é visto na Figura 4.6. Esta autenticação será descrita com detalhes ainda neste capítulo.

A.2 Attribute Module

O *Attribute Provider* provê um diretório adicional para armazenamento dos atributos específicos da OV. Este diretório é uma base LDAP. Há ainda o agregador de atributos, responsável pela união dos atributos vindos da federação de identidade com os atributos adicionais do provedor de atributos da OV.

Conforme se pode observar na Figura A.4, para o *Attribute Provider*, a interação direta é somente realizada pelo administrador da OV, quando realizada a instalação e configuração do módulo. Já na utilização em si do *framework* pelo usuário, a comunicação com este módulo é feita de forma transparente para o usuário.

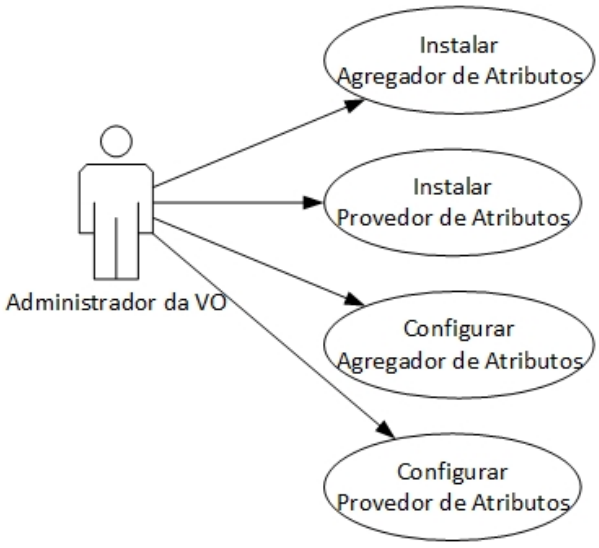

Figura A.4. Diagrama de casos de uso para o *Attribute Module*.

A fim de facilitar o entendimento da instalação e configuração do módulo pelo administrador da OV, existem dois diagramas de sequência, ilustrados pelas Figuras A.5 e A.6. A Figura A.5 ilustra, especificamente, a parte referente ao provedor de atributos adicionais e a Figura A.6 exibe as etapas para a criação do agregador de atributos.

O diagrama de sequência do provedor de atributos da Figura A.5 tem o seguinte passo-a-passo:

1. o administrador da OV inicializa a instalação do AP (*Attribute Provider*) a partir da chamada *Start AP Install*;

2. o módulo instala os pacotes básicos para o provedor de atributos, basicamente, o serviço LDAP (*slapd*);

3. o módulo requisita o nome da organização ($ORGANIZATION) e o administrador o informa;

4. administrador da OV informa quais atributos (adicionais) ($VO_ATTRIBUTES_LIST) serão inseridos no *schema* LDAP, seu tipo e nome, a fim de que sejam suportados;

5. módulo requisita as credenciais do administrador da OV. Neste momento, será requisitada a senha para o usuário "cn=Manager,dc=%$ORGANIZATION%,dc=across";

6. feitos esses passos, o assistente irá requisitar ao administrador que inicie o assistente do agregador de atributos, ilustrado na Figura A.6;

7. uma vez finalizada a etapa do assistente do agregador de atributos, um teste de validação é executado e a instalação é finalizada.

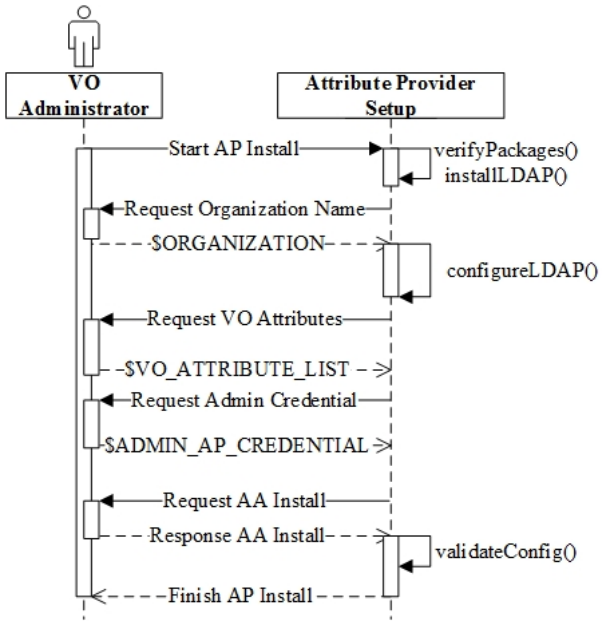

Figura A.5. Instalação e configuração do Provedor de Atributos (*Attribute Provider*).

Como parte da instalação/configuração do módulo provedor de atributos, o assistente de instalação/configuração do agregador de atributos deverá ser utilizado. Sendo assim, a Figura A.6 exibe o diagrama de sequência para esta etapa. Seguindo o diagrama, temos o passo-a-passo:

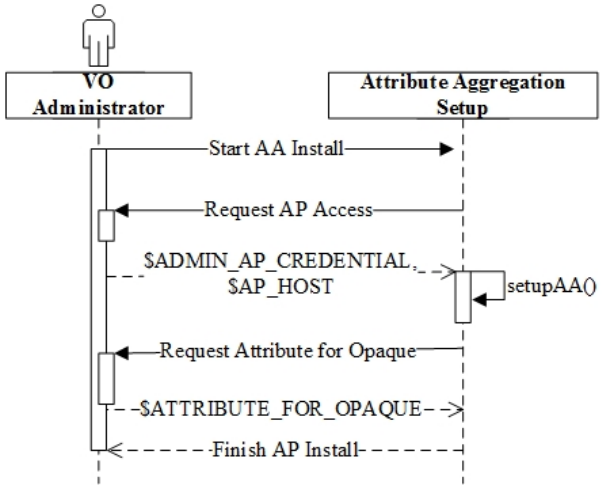

Figura A.6. Instalação e configuração do Agregador de Atributos (*Attribute Aggregation*)).

1. inicialização do assistente por meio da chamada *Start AA Install*;
2. módulo requisita ao administrador as credenciais para acesso ao LDAP do provedor de atributos;
3. o administrador informa os dados de acesso, endereço do IP ou FQDN do *host* e as credenciais (usuário e senha do administrador – ou usuário com permissão de leitura/consulta da base);
4. o administrador também é questionado sobre qual, ou quais, atributo(s) representam o atributo opaco;
5. o módulo então configura e valida as configurações locais e de acesso ao agregador de atributos;
6. o módulo confirma ao administrador da OV sucesso e finaliza o assistente, indicando que o administrador pode retornar ao passo de finalização do assistente do provedor de atributos.

A.3 Access Control Module

Para a instalação e configuração do *Access Control Module*, o administrador, tanto da OV quanto da instituição, deve iniciar o assistente de instalação do suporte ao XACML, como pode ser visto no diagrama de casos de uso da Figura A.7.

Figura A.7. Diagrama de casos de uso para a instalação do suporte ao *Access Control Module*.

Os passos desse assistente são simples, instalando apenas o suporte ao Java e, logo após, o pacote do XACML, baseado no WSO2 Balana[2]. Esses passos estão descritos no diagrama de sequência da Figura A.8.

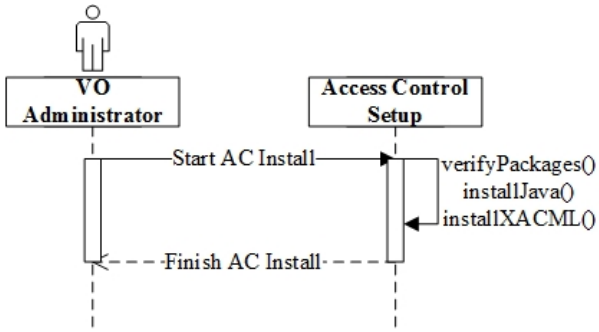

Figura A.8. Diagrama de sequência para a instalação do suporte ao *Access Control Module*.

A.4 VO Manager

O diagrama de casos de uso para o *VO Manager* é mostrado na Figura A.9, que compreende a instalação do portal tanto para a OV quanto para as instituições.

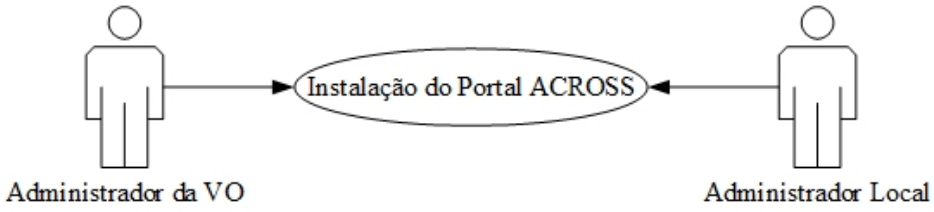

Figura A.9. Diagrama de casos de uso para a instalação do portal de administração para a OV e local.

O diagrama de sequência da atividade de instalação do portal do ACROSS, visto na Figura A.10, mostra a instalação desse portal de administração web e suas dependências.

[2]https://github.com/wso2/balana.

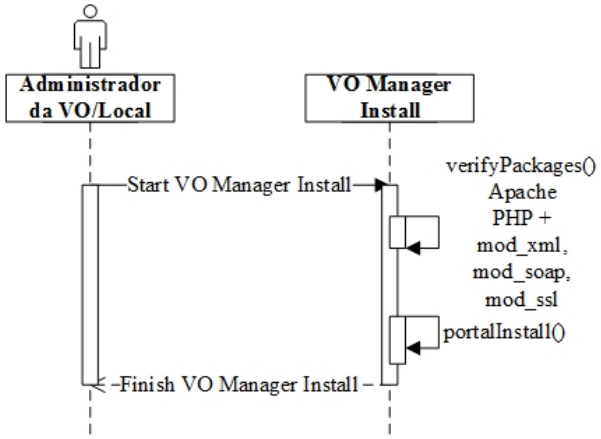

Figura A.10. Diagrama de sequência para a instalação do portal de administração para a OV e local.

APÊNDICE B – Documentação XML

Para o armazenamento e comunicação padronizada entre as entidades envolvidas no ACROSS, são utilizado arquivos XML.

Este apêndice apresenta exemplos para todos os arquivos XML gerados pelo ACROSS em seus respectivos módulos.

B.1 ACROSS Global

No nível global do ACROSS são configurados dados gerais da OV, assim como cadastradas as formas de comunicação entre o ACROSS Global e o Attribute Module, por exemplo. Também são armazenados os arquivos XML com informações referentes aos pontos (*score*), classificação em níveis (*level*) e políticas (*policies*) de nível global.

B.1.1 XML da Configuração da OV

A Listagem B.1 mostra um exemplo do XML de configuração com os dados da OV. Onde há *tags* referentes ao nome da VO, acrônimo, email de contato do responsável e coordenadas geográficas de localização da sede daquela OV.

Listagem B.1. Exemplo de configuração de dados da OV. Arquivo organization.xml.

```
1 <?xml version="1.0"?>
2 <organization>
3   <data>
4     <name>My Virtual Organization</name>
5     <acronym>TESTVO</acronym>
6     <mail>esilva@ic.uff.br</mail>
7     <coordinates>-22.9, -43.129999999999995</coordinates>
8   </data>
9 </organization>
```

Já a Listagem B.2 tem um exemplo de arquivo resultado da configuração no *Attribute Aggregation*, gerado durante a instalação e configuração do *Attribute Module*,

referente à comunicação entre este módulo e o *Attribute Provider*. Este arquivo apresenta o endereço IP *Attribute Provider* e provisiona um acesso por usuário e senha a ser armazenado nas respectivas *tags* de mesmo nome.

Listagem B.2. Exemplo de configuração do arquivo ap_address.xml

```
<attributeProvider>
  <data>
    <address>192.168.0.3</address>
    <username></username>
    <password></password>
  </data>
</attributeProvider>
```

Listagem B.3. Exemplo de configuração do arquivo AA.conf

```
manager;password;ldapAddr
cn=Manager,dc=uff,dc=across;1ed32211;192.168.0.3
```

Dados, como usuário do LDAP no *Attribute Provider*, senha e caminho na árvore de diretório, são armazenado pelo *Attribute Aggregation* em um arquivo de configuração com o formato do exemplo da Listagem B.3.

Outro arquivo de configuração gerado, e necessário para o entendimento do tipo de atributo armazenado pelo *Attribute Module*, é apresentado na Listagem B.4. Estas entradas se referem aos atributos adicionais.

Listagem B.4. Exemplo do arquivo APattr.conf.

```
admin;boolean
position;string
```

B.1.2 XML da Configuração para Classificação em Níveis

Um exemplo do arquivo XML resultado da configuração de pontuação (*score*) de atributos pode ser visto na Listagem B.5. A configuração deste arquivo é feita no nível global, a partir da interface de gerência *VO Manager*.

Com o elemento raíz <*score*>, para cada atributo com pontuação, há uma *tag* inicial <*attribute*> e uma *tag* final </*attribute*>, que possui os elementos <*name*>, referente ao nome do atributo, <*op*>, referente ao operador selecionado, <*point*>, refente aos pontos, <*weight*>, ao peso, e <*total*> é a multiplicação entre ponto e peso.

Listagem B.5. Exemplo do arquivo score.xml.

```
<?xml version="1.0"?>
<score>
  <attribute>
    <name>admin</name>
    <op>==</op>
    <value>true</value>
    <point>10</point>
    <weight>10</weight>
    <total>100</total>
  </attribute>
  <attribute>
    <name>position</name>
    <op>==</op>
    <value>faculty</value>
    <point>30</point>
    <weight>2</weight>
    <total>60</total>
  </attribute>
  <attribute>
    <name>Shib-eduPerson-eduPersonPrimaryAffiliation</name>
    <op>==</op>
    <value>faculty</value>
    <point>30</point>
    <weight>2</weight>
    <total>60</total>
  </attribute>
  <attribute>
    <name>position</name>
    <op>==</op>
    <value>student</value>
    <point>30</point>
    <weight>1</weight>
    <total>30</total>
  </attribute>
</score>
```

A partir do XML gerado na Listagem B.5, o máximo valor possível para combinação de atributos de um usuário é armazenado em outro XML, como mostra a Listagem B.6. Este arquivo é a entrada para o cálculo da normalização dos valores de pontuação dos atributos de um usuário.

Listagem B.6. Exemplo do arquivo max.xml.

```
<?xml version="1.0"?>
<normalization>
  <values>
    <min>0</min>
    <max>220</max>
  </values>
</normalization>
```

A configuração de níveis gera o arquivo XML da Listagem B.7, a partir do cadastro do Administrador da OV.

Listagem B.7. Exemplo do arquivo level.xml.

```
<?xml version="1.0"?>
<levels>
  <level>
    <number>3</number>
    <min>0.6</min>
    <max>1.0</max>
  </level>
  <level>
    <number>2</number>
    <min>0.4</min>
    <max>0.6</max>
  </level>
  <level>
    <number>1</number>
    <min>0</min>
    <max>0.4</max>
  </level>
</levels>
```

B.2 Configuração das Políticas

As políticas são configuradas tanto em nível global como local à instituição. Ambas devem cadastras recursos. Porém, no nível global é cadastrado somente o tipo de recurso, e no nível local das instituições, é cadastrada a quantidade daquele tipo de recurso.

B.2.1 XML para Configuração de Recursos

Sendo assim, o exemplo da Listagem B.8, mostra os tipos de recursos cadastrados e suportados pela OV, feito no nível gobal.

Listagem B.8. Exemplo do arquivo resources.xml.

```
<?xml version="1.0"?>
<resources>
  <resource>
    <id>5</id>
    <type>VM</type>
    <description>Virtual Machine</description>
  </resource>
</resources>
```

Já a Listagem B.9, mostra o cadastro da quantidade de recursos a instituição 1 possui.

Listagem B.9. Exemplo do arquivo resources.xml da Instituição.

```
<?xml version="1.0"?>
<institution_resources>
  <institution>
    <id>Inst1</id>
  </institution>
  <resources>
    <resource>
      <id>5</id>
      <type>VM</type>
      <total>3</total>
    </resource>
  </resources>
</institution_resources>
```

B.2.2 XML para Configuração de Políticas

A Listagem B.10 mostra o XML de políticas, que tem o mesmo formato tanto nos níveis Global quanto local.

Listagem B.10. Exemplo do arquivo policies.xml.

```
<?xml version="1.0"?>
<policies>
  <policy>
    <level>3</level>
    <resource>VM</resource>
    <total>10</total>
  </policy>
  <policy>
    <level>2</level>
    <resource>VM</resource>
    <total>5</total>
  </policy>
  <policy>
    <level>1</level>
    <resource>VM</resource>
    <total>1</total>
  </policy>
</policies>
```

Esse XML é a entrada para a geração do XACML já mostrado no texto desta tese.

Por fim, as consultas aos ACROSS locais são realizadas pelo ACROSS global seguindo o arquivo XML da Listagem B.11. Este XML contém o endereço IP de cada uma das instituições cadastradas pela interface do *VO Manager*, e serão consultadas

a partir desses dados quanto às políticas locais. Há a previsão da utilização de uma comunicação autenticada entre as entidades, por usuário e senha.

Listagem B.11. Exemplo do arquivo institutions.xml.

```
<?xml version="1.0"?>
<institutions>
  <institution>
    <address>192.168.0.4</address>
    <acronym>Inst1</acronym>
    <description>Institution 1</description>
    <username></username>
    <password></password>
  </institution>
  <institution>
    <address>192.168.0.5</address>
    <acronym>Inst2</acronym>
    <description>Institution 2</description>
    <username></username>
    <password></password>
  </institution>
  <institution>
    <address>192.168.0.6</address>
    <acronym>Inst3</acronym>
    <description>Institution 3</description>
    <username></username>
    <password></password>
  </institution>
</institutions>
```

APÊNDICE C – Documentação da API

Esta documentação complementar a esta tese, apresenta a API *Application Programming Interface* do ACROSS, com suas funções e retornos.

A documentação completa e atualizada pode ser acessada pelo endereço: `http://www.midiacom.uff.br/across/`.

C.1 VO Manager

Listagem C.1. função getVOData

```
function getVOData(xml) {
   return string
}

Função: getVOData
Parâmetro: xml
Retorno: string
```

Listagem C.2. função storeInstitutions.

```
function storeInstitutions(){
   return boolean
}

Função: storeInstitutions()
Parâmetro: vetor string
Retorno: boolean
```

C.1.1 Resource

Listagem C.3. função setResource.

```
function setResource(){
   return boolean
}

Função: setResource ()
Parâmetro: vetor string
Retorno: boolean
```

Listagem C.4. função getResource.

```
function getResource(){
   return vetor string
}

Função: getResource ()
Parâmetro: null
Retorno: vetor string
```

C.2 SP Module

Listagem C.5. função manageAttributes.

```
function manageAttributes(){
   return vetor string
}

Função: manageAttributes()
Parâmetro: vetor string
Retorno: vetor de string
```

Listagem C.6. função verifyAttributes()

```
function verifyAttributes(){
   return vetor string
}

Função: verifyAttributes()
Parâmetro: vetor string
Retorno: vetor string
```

Listagem C.7. função aggregateAttributes

```
function aggregateAttributes(){
   return vetor string
}

Função: aggregateAttributes()
Parâmetro: vetor string
Retorno: vetor string
```

Listagem C.8. função recoverUserAttributes

```
function recoverUserAttributes(){
   return vetor string
}

Função: recoverUserAttributes()
Parâmetro: vetor string
Retorno: vetor string
```

C.3 Identity Federation Module

Apenas a função que disponibiliza os atributos para a transposição de credenciais é exibida nesta seção.

C.3.1 Credential Translation Module

Listagem C.9. função credentialTranslation

```
function credentialTranslation(){
   return vetor string
}

Função: credentialTranslation()
Parâmetro: userID
Retorno: vetor string
```

C.4 Attribute Module

Funções para o *Attribute Module*, com seus módulos: *Attribute Aggregation Module* e *Attribute Provider Module*.

C.4.1 Attribute Aggregation Module

Listagem C.10. função generateOpaqueID

```
function generateOpaqueID(){
   return string
}

Função: generateOpaqueID()
Parâmetro: string
Retorno: string
```

Listagem C.11. função requestVOAttributes

```
function requestVOAttributes(){
   return vetor string
}

Função: requestVOAttributes()
Parâmetro: string
Retorno: vetor string
```

C.4.2 Attribute Provider Module

Listagem C.12. função recoveryVOAttributes

```
function recoveryVOAttributes(){
   return vetor string
}

Função: recoveryVOAttributes()
Parâmetro: string
Retorno: vetor string
```

C.5 Access Control Module

Funções para o *Access Control Module*, com seus submódulos: *Score, Level User* e *Global/Local Policy*.

C.5.1 Score

Listagem C.13. função setAttScore

```
function setAttScore(){
   return boolean
}

Função: setAttScore()
Parâmetro: vetor string
Retorno: boolean
```

Listagem C.14. função setScore

```
1 function setScore(){
2    return boolean
3 }
4
5 Função: setScore()
6 Parâmetro: vetor string
7 Retorno: boolean
```

Listagem C.15. função calcMaxScore

```
function calcMaxScore(){
   return integer
}

Função: calcMaxScore()
Parâmetro: vetor integer
Retorno: integer
```

C.5.2 User Level

Listagem C.16. função setLevel

```
function setLevel(){
   return integer
}

Função: setLevel()
Parâmetro: integer
Retorno: integer
```

Listagem C.17. função recoveryLevels

```
function recoveryLevels(){
   return vetor string
}

Função: recoveryLevels()
Parâmetro: null
Retorno: vector string
```

C.5.3 Global/Local Policy

Listagem C.18. função setPolicy

```
function setPolicy(){
   return boolean
}

Função: setPolicy()
Parâmetro: boolean
Retorno: vector string
```

Listagem C.19. função generateXACMLPolicy

```
1 function generateXACMLPolicy(){
2    return boolean
3 }
4
5 Função: generateXACMLPolicy()
6 Parâmetro: boolean
7 Retorno: vector string
```